FC9Y-B812

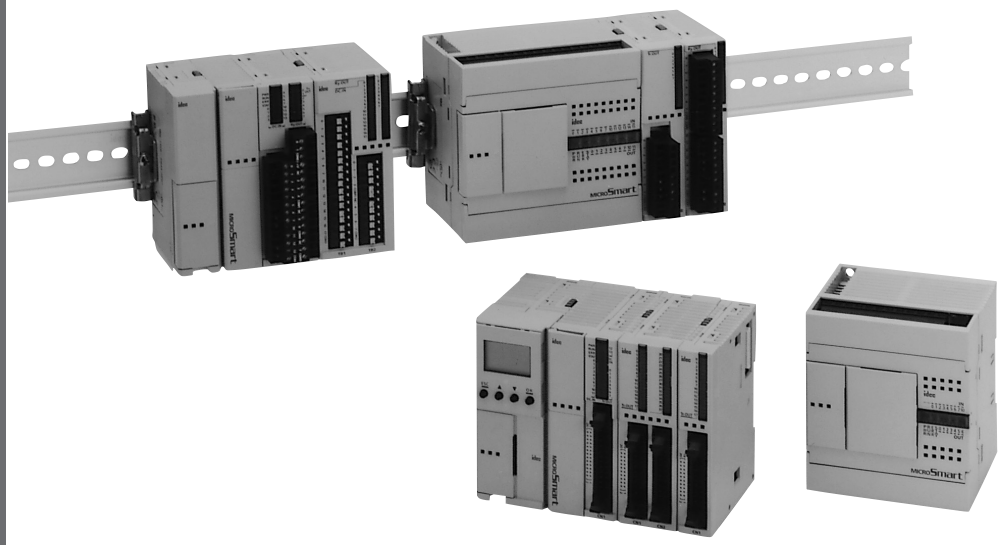
MICROSmart

IDEC

FC4A SERIES

Micro Programmable Logic Controller

User's Manual



IDEC CORPORATION

MICROSMART USER'S MANUAL UPDATE

Introduction

This manual includes additional descriptions of new modules and upgraded functionality of the FC4A MicroSmart CPU modules with system program version up to 210 in detail.

New Modules

Analog I/O Modules (Ladder Refresh Type)

Name	I/O Signal	I/O Points	Type No.
Analog Input Module	Voltage (0 to 10V DC) Current (4 to 20mA) Thermocouple (K, J, T) Resistance thermometer (Pt100, Pt1000, Ni100, Ni1000)	4 inputs	FC4A-J4CN1
	Voltage (0 to 10V DC) Current (4 to 20mA)	8 inputs	FC4A-J8C1
	Thermistor (NTC, PTC)	8 inputs	FC4A-J8AT1
Analog Output Module	Voltage (–10 to +10V DC) Current (4 to 20mA)	2 outputs	FC4A-K2C1

Upgraded Functionality

Twelve new functions have been implemented in the FC4A MicroSmart CPU modules. Availability of the twelve new functions depends on the model and system program version of the CPU modules as listed below:

<table>
<tr><th rowspan="2">CPU Module</th><th colspan="3">All-in-One Type</th><th colspan="2">Slim Type</th></tr>
<tr><th>FC4A-C10R2
FC4A-C10R2C</th><th>FC4A-C16R2
FC4A-C16R2C</th><th>FC4A-C24R2
FC4A-C24R2C</th><th>FC4A-D20K3
FC4A-D20S3</th><th>FC4A-D20RK1
FC4A-D20RS1
FC4A-D40K3
FC4A-D40S3</th></tr>
<tr><td>Analog I/O Modules (Ladder Refresh Type)</td><td>—</td><td>—</td><td>204 or higher</td><td>204 or higher</td><td>203 or higher</td></tr>
<tr><td>AS-Interface Master Module Compatibility</td><td rowspan="3">—</td><td rowspan="3">—</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="4">201 or higher</td></tr>
<tr><td>64KB Memory Cartridge Compatibility</td></tr>
<tr><td>PID Instruction Upgrade</td><td rowspan="2">202 or higher</td><td rowspan="2">202 or higher</td></tr>
<tr><td>HMI Module Initial Screen Selection</td><td>203 or higher</td><td>202 or higher</td></tr>
<tr><td>RS485 User Communication Compatibility</td><td rowspan="7">—</td><td rowspan="3">204 or higher</td><td rowspan="3">204 or higher</td><td rowspan="6">204 or higher</td><td rowspan="3">202 or higher</td></tr>
<tr><td>User Communication BCC Upgrade
(ADD-2comp, Modbus ASCII, and Modbus RTU)</td></tr>
<tr><td>Pulse Instructions Upgrade</td><td rowspan="3">—</td><td>—</td></tr>
<tr><td>Coordinate Conversion Instructions Upgrade</td><td rowspan="2">204 or higher</td><td rowspan="2">203 or higher</td></tr>
<tr><td>Intelligent Module Access Instructions</td></tr>
<tr><td>Downloading from Memory Cartridge to CPU Module</td><td rowspan="2">210 or higher</td><td rowspan="2">210 or higher</td><td rowspan="2">210 or higher</td><td rowspan="2">210 or higher</td></tr>
<tr><td>User Program Read Prohibit</td></tr>
</table>

To confirm the system program version of the MicroSmart CPU module, use WindLDR on a computer connected with the CPU module. Bring WindLDR into the online mode. The system program version is indicated on the PLC status dialog box. For details about the procedure, see page 29-1.

SAFETY PRECAUTIONS

- Read this user's manual to make sure of correct operation before starting installation, wiring, operation, maintenance, and inspection of the MicroSmart.
- All MicroSmart modules are manufactured under IDEC's rigorous quality control system, but users must add a backup or failsafe provision to the control system when using the MicroSmart in applications where heavy damage or personal injury may be caused in case the MicroSmart should fail.
- In this user's manual, safety precautions are categorized in order of importance to Warning and Caution:

⚠ Warning **Warning notices are used to emphasize that improper operation may cause severe personal injury or death.**

- Turn off power to the MicroSmart before starting installation, removal, wiring, maintenance, and inspection of the MicroSmart. Failure to turn power off may cause electrical shocks or fire hazard.
- Special expertise is required to install, wire, program, and operate the MicroSmart. People without such expertise must not use the MicroSmart.
- Emergency stop and interlocking circuits must be configured outside the MicroSmart. If such a circuit is configured inside the MicroSmart, failure of the MicroSmart may cause disorder of the control system, damage, or accidents.
- Install the MicroSmart according to the instructions described in this user's manual. Improper installation will result in falling, failure, or malfunction of the MicroSmart.

⚠ Caution **Caution notices are used where inattention might cause personal injury or damage to equipment.**

- The MicroSmart is designed for installation in a cabinet. Do not install the MicroSmart outside a cabinet.
- Install the MicroSmart in environments described in this user's manual. If the MicroSmart is used in places where the MicroSmart is subjected to high-temperature, high-humidity, condensation, corrosive gases, excessive vibrations, and excessive shocks, then electrical shocks, fire hazard, or malfunction will result.
- The environment for using the MicroSmart is "Pollution degree 2." Use the MicroSmart in environments of pollution degree 2 (according to IEC 60664-1).
- Prevent the MicroSmart from falling while moving or transporting the MicroSmart, otherwise damage or malfunction of the MicroSmart will result.
- Prevent metal fragments and pieces of wire from dropping inside the MicroSmart housing. Put a cover on the MicroSmart modules during installation and wiring. Ingress of such fragments and chips may cause fire hazard, damage, or malfunction.
- Use a power supply of the rated value. Use of a wrong power supply may cause fire hazard.
- Use an IEC 60127-approved fuse on the power line outside the MicroSmart. This is required when equipment containing the MicroSmart is destined for Europe.
- Use an IEC 60127-approved fuse on the output circuit. This is required when equipment containing the MicroSmart is destined for Europe.
- Use an EU-approved circuit breaker. This is required when equipment containing the MicroSmart is destined for Europe.
- Make sure of safety before starting and stopping the MicroSmart or when operating the MicroSmart to force outputs on or off. Incorrect operation on the MicroSmart may cause machine damage or accidents.
- If relays or transistors in the MicroSmart output modules should fail, outputs may remain on or off. For output signals which may cause heavy accidents, provide a monitor circuit outside the MicroSmart.
- Do not connect the ground wire directly to the MicroSmart. Connect a protective ground to the cabinet containing the MicroSmart using an M4 or larger screw. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not disassemble, repair, or modify the MicroSmart modules.
- Dispose of the battery in the MicroSmart modules when the battery is dead in accordance with pertaining regulations. When storing or disposing of the battery, use a proper container prepared for this purpose. This is required when equipment containing the MicroSmart is destined for Europe.
- When disposing of the MicroSmart, do so as an industrial waste.

About This Manual

This user's manual primarily describes entire functions, installation, and programming of the FC4A MicroSmart CPU and all other modules. Also included are powerful communications of the MicroSmart and troubleshooting procedures.

CHAPTER 1: GENERAL INFORMATION

General information about the MicroSmart, features, brief description on special functions, and various system setup configurations for communication.

CHAPTER 2: MODULE SPECIFICATIONS

Specifications of CPU, input, output, mixed I/O, analog I/O, and other optional modules.

CHAPTER 3: INSTALLATION AND WIRING

Methods and precautions for installing and wiring the MicroSmart modules.

CHAPTER 4: OPERATION BASICS

General information about setting up the basic MicroSmart system for programming, starting and stopping MicroSmart operation, and simple operating procedures from creating a user program using WindLDR on a PC to monitoring the MicroSmart operation.

CHAPTER 5: SPECIAL FUNCTIONS

Stop/reset inputs, run/stop selection at memory backup error, keep designation for internal relays, shift registers, counters, and data registers. Also included are high-speed counter, catch input, interrupt input, timer interrupt, input filter, user program read/write protection, constant scan time, partial program download, and many more special functions.

CHAPTER 6: ALLOCATION NUMBERS

Allocation numbers available for the MicroSmart CPU modules to program basic and advanced instructions. Special internal relays and special data registers are also described.

CHAPTER 7: BASIC INSTRUCTIONS

Programming of the basic instructions, available operands, and sample programs.

CHAPTER 8: ADVANCED INSTRUCTIONS

General rules of using advanced instructions, terms, data types, and formats used for advanced instructions.

CHAPTER 9 THROUGH CHAPTER 23:

Detailed descriptions on advanced instructions grouped into 15 chapters.

CHAPTER 24 THROUGH CHAPTER 28:

Analog I/O control and various communication functions such as data link, computer link, modem mode, and AS-Interface.

CHAPTER 29: TROUBLESHOOTING

Procedures to determine the cause of trouble and actions to be taken when any trouble occurs while operating the MicroSmart.

APPENDIX

Additional information about execution times for instructions, I/O delay time, and MicroSmart type list.

INDEX

Alphabetical listing of key words.

IMPORTANT INFORMATION

Under no circumstances shall IDEC Corporation be held liable or responsible for indirect or consequential damages resulting from the use of or the application of IDEC PLC components, individually or in combination with other equipment.

All persons using these components must be willing to accept responsibility for choosing the correct component to suit their application and for choosing an application appropriate for the component, individually or in combination with other equipment.

All diagrams and examples in this manual are for illustrative purposes only. In no way does including these diagrams and examples in this manual constitute a guarantee as to their suitability for any specific application. To test and approve all programs, prior to installation, is the responsibility of the end user.

Revision Record

The table below summarizes the changes to this manual since last printing of FC9Y-B812-0A in June, 2006.

Revision	Description of Change	Page
Analog I/O Modules (Ladder Refresh Type)	Four analog input and output modules are added.	2-43, 6-5, 24-1
AS-Interface Master Module Compatibility	AS-Interface master module is added.	2-58, 6-5, 28-1
RS485 User Communication Compatibility	These functions are now available for FC4A-C16R2, FC4A-C16R2C, FC4A-C24R2, FC4A-C24R2C, FC4A-D20K3, and FC4A-D20S3.	17-1
User Communication BCC Upgrade (ADD-2comp, Modbus ASCII, and Modbus RTU)		
Pulse Instructions Upgrade	These functions are now available for FC4A-D20K3 and FC4A-D20S3.	20-1
Coordinate Conversion Instructions Upgrade	These functions are now available for FC4A-C24R2C, FC4A-D20K3, and FC4A-D20S3.	19-1
Intelligent Module Access Instructions		23-1
Downloading from Memory Cartridge to CPU Module	A user program can be downloaded from a memory cartridge to the CPU module.	2-66
User Program Read Prohibit	Read protection is upgraded and this option prevents copying of the user program completely.	5-25
Analog I/O Modules Upgrade (Version 200 or higher)	Four END refresh type analog input and output modules are upgraded.	2-44

TABLE OF CONTENTS

CHAPTER 5: SPECIAL FUNCTIONS

CHAPTER 6: ALLOCATION NUMBERS

CHAPTER 7: BASIC INSTRUCTIONS

1: GENERAL INFORMATION

Introduction

This chapter describes general information for understanding the MicroSmart's powerful capabilities and system setups to use the MicroSmart in various ways of communication.

About the MicroSmart

IDEC's MicroSmart is a new family of micro programmable logic controllers available in two styles of CPU modules; all-in-one and slim types. The all-in-one type CPU module has 10, 16, or 24 I/O terminals and is equipped with a built-in universal power supply to operate on 100 to 240V AC or 24V DC. Using four 16-point I/O modules, the 24-I/O type CPU module can expand the I/O points up to a total of 88 points. The slim type CPU module has 20 or 40 I/O terminals and operates on 24V DC. The total I/O points can be expanded to a maximum of 264.

User programs for the MicroSmart can be edited using WindLDR on a Windows PC. Since WindLDR can load existing user programs made for IDEC's previous PLCs such as all FA series, MICRO-1, MICRO^3, MICRO^{3}C, and OpenNet Controller, your software assets can be used in the new control system.

Program capacity of the all-in-one type CPU modules is 4,800 bytes (800 steps) on the 10-I/O type CPU module, 15,000 bytes (2,500 steps) on the 16-I/O type, and 27,000 bytes (4,500 steps) on the 24-I/O type. Slim type CPU modules have a program capacity of 27,000 bytes (4,500 steps) or 31,200 bytes (5,200 steps). When using an optional 64KB memory cartridge on slim type CPU modules, the program capacity can be expanded up to 64,500 bytes (10,750 steps).

Features

Powerful Communication Functions

The MicroSmart features four powerful communication functions.

Maintenance Communication (Computer Link)	When a MicroSmart CPU module is connected to a computer, operating status and I/O status can be monitored on the computer, data in the CPU can be monitored or updated, and user programs can be downloaded and uploaded. All CPU modules (except the all-in-one 10-I/O type) can set up a 1:N computer link system to connect a maximum of 32 CPU modules to a computer.
User Communication	All MicroSmart CPU modules can be linked to external RS232C devices such as computers, printers, and barcode readers, using the user communication function. RS485 user communication is also available on upgraded CPU modules of slim 20-I/O relay output and 40-I/O types.
Modem Communication	All MicroSmart CPU modules (except the all-in-one 10-I/O type) can communicate through modems using the built-in modem protocol.
Data Link	All MicroSmart CPU modules (except the all-in-one 10-I/O type) can set up a data link system. One CPU module at the master station can communicate with 31 slave stations through an RS485 line to exchange data and perform distributed control effectively.

Communication Adapter (All-in-one 16- and 24-I/O type CPU modules)
Communication Module (Slim type CPU modules)

In addition to the standard RS232C port 1, the all-in-one 16- and 24-I/O type CPU modules feature a port 2 connector to install an optional RS232C or RS485 communication adapter. All slim type CPU modules can be used with an optional RS232C or RS485 communication module to add communication port 2. With an optional HMI base module mounted with a slim type CPU module, an optional RS232C or RS485 communication adapter can also be installed on the HMI base module.

RS232C Communication Adapter **RS232C Communication Module**	Used for computer link 1:1 communication, user communication, and modem communication.
RS485 Communication Adapter **RS485 Communication Module**	Available in mini DIN connector and terminal block styles. Used for computer link 1:1 or 1:N communication, user communication, and data link communication.

HMI Module (all CPU modules)

An optional HMI module can be installed on any all-in-one type CPU module, and also on the HMI base module mounted next to any slim type CPU module. The HMI module makes it possible to manipulate the RAM data in the CPU module without using the Online menu options in WindLDR.

HMI module functions include:

- Displaying timer/counter current values and changing timer/counter preset values
- Displaying and changing data register values
- Setting and resetting bit operand statuses, such as inputs, outputs, internal relays, and shift register bits
- Displaying and clearing error data
- Starting and stopping the PLC
- Displaying and changing calendar/clock data (only when using the clock cartridge)
- Confirming changed timer/counter preset values

Clock Cartridge (all CPU modules)

An optional clock cartridge can be installed on the CPU module to store real time calendar/clock data for use with advanced instructions to perform time-scheduled control.

Memory Cartridge (all CPU modules)

A user program can be stored on an optional memory cartridge using WindLDR. The memory cartridge can be installed on another CPU module to replace user programs without the need for connecting to a computer. The original user program in the CPU module is restored after removing the memory cartridge. The user program on the memory cartridge can also be downloaded to the CPU module. The download option is selected using WindLDR.

Analog I/O Modules (all CPU modules except the all-in-one 10- and 16-I/O types)

Analog I/O modules are available in 3-I/O types, 2-input type, and 1-output type. The input channel can accept either voltage (0 to 10V DC) and current (4 to 20 mA) signals or thermocouple (types K, J, and T) and resistance thermometer (Pt 100) signals. The output channel generates voltage (0 to 10V DC) and current (4 to 20 mA) signals.

AS-Interface Master Module (slim type 20-I/O relay output and 40-I/O types)

Four upgraded slim type CPU modules (FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3) with system program ver. 201 and higher can use the AS-Interface master module, and have additional internal relays and data registers to communicate with slaves, such as actuators and sensors, through the AS-Interface bus. For details about AS-Interface module and AS-Interface communication, see page 2-58 and chapter 28.

Special Functions

The MicroSmart features various special functions packed in the small housing as described below. For details about these functions, see the following chapters.

Stop and Reset Inputs

Any input terminal on the CPU module can be designated as a stop or reset input to control the MicroSmart operation.

RUN/STOP Selection at Startup when “Keep” Data is Broken

When data to be kept such as “keep” designated counter values are broken while the CPU is powered down, the user can select whether the CPU starts to run or not to prevent undesirable operation at the next startup.

“Keep” or “Clear” Designation of CPU Data

Internal relays, shift register bits, counter current values, and data register values can be designated to be kept or cleared when the CPU is powered down. All or a specified range of these operands can be designated as keep or clear types.

High-speed Counter

The MicroSmart has four built-in high-speed counters to make it possible to count up to 65,535 (FFFFh) high-speed pulses which cannot be counted by the normal user program processing. One high-speed counter (all-in-one type CPU modules) or two high-speed counters (slim type CPU modules) can be used as either two-phase or single-phase high-speed counters at a maximum count input frequency of 20 kHz. Three or two others are single-phase high-speed counters with a maximum counting frequency of 5 kHz. The high-speed counters can be used for simple positioning control and simple motor control.

Catch Input

Four inputs can be used as catch inputs. The catch input makes sure to receive short input pulses (rising pulse of 40 μs or falling pulse of 150 μs minimum) from sensors without regard to the scan time.

Interrupt Input

Four inputs can be used as interrupt inputs. When a quick response to an external input is required, such as positioning control, the interrupt input can call a subroutine to execute an interrupt program.

Timer Interrupt

In addition to the interrupt input, slim type CPU modules FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K1, and FC4A-D40S1 have a timer interrupt function. When a repetitive operation is required, the timer interrupt can be used to call a subroutine repeatedly at predetermined intervals of 10 through 140 ms.

Input Filter

The input filter can be adjusted for eight inputs to reject input noises. Selectable input filter values to pass input signals are 0 ms, and 3 through 15 ms in 1-ms increments. The input filter rejects inputs shorter than the selected input filter value minus 2 ms. This function is useful for eliminating input noises and chatter in limit switches.

User Program Read/Write Protection

The user program in the CPU module can be protected against reading and/or writing by including a password in the user program. Read protection without a password is also available to inhibit reading completely.

Constant Scan Time

The scan time may vary whether basic and advanced instructions are executed or not depending on input conditions to these instructions. When performing repetitive control, the scan time can be made constant by entering a required scan time value into a special data register reserved for constant scan time.

Partial Program Download

Normally, the CPU module has to be stopped before downloading a user program. All CPU modules (except the all-in-one 10-I/O type) have run-time program download capabilities to download a user program containing small changes while the CPU is running in either 1:1 or 1:N computer link system. This function is particularly useful to make small modifications to the user program and confirm the changes while the CPU is running.

Analog Potentiometer

All CPU modules have an analog potentiometer, except the all-in-one 24-I/O type CPU module has two analog potentiometers. The values (0 through 255) set with analog potentiometers 1 and 2 are stored to special data registers. The analog potentiometer can be used to change the preset value for a timer or counter.

Analog Voltage Input

Every slim type CPU module has an analog voltage input connector. When an analog voltage of 0 through 10V DC is applied to the analog voltage input connector, the signal is converted to a digital value of 0 through 255 and stored to a special data register. The data is updated in every scan.

Pulse Output

Slim type CPU modules have pulse output instructions to generate high-speed pulse outputs from transistor output terminals used for simple position control applications, illumination control, trapezoidal control, and zero-return control.

PID Control

All CPU modules (except the all-in-one 10- and 16-I/O types) have the PID instruction, which implements a PID (proportional, integral, and derivative) algorithm with built-in auto tuning to determine PID parameters. This instruction is primarily designed for use with an analog I/O module to read analog input data, and turns on and off a designated output to perform PID control in applications such as temperature control. In addition, the PID instruction can also generate an analog output using an analog I/O module.

Expansion Data Register

Slim type CPU modules FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3 have expansion data registers D2000 through D7999. Numerical data can be set to expansion data registers using WindLDR. When downloading the user program, the preset values of the expansion data registers are also downloaded to the EEPROM in the CPU module. Since the data in the EEPROM is non-volatile, the preset values of the expansion data registers are maintained semi-permanently and loaded to the RAM each time the CPU is powered up.

System Setup

This section illustrates system setup configurations for using powerful communication functions of the MicroSmart.

User Communication and Modem Communication System

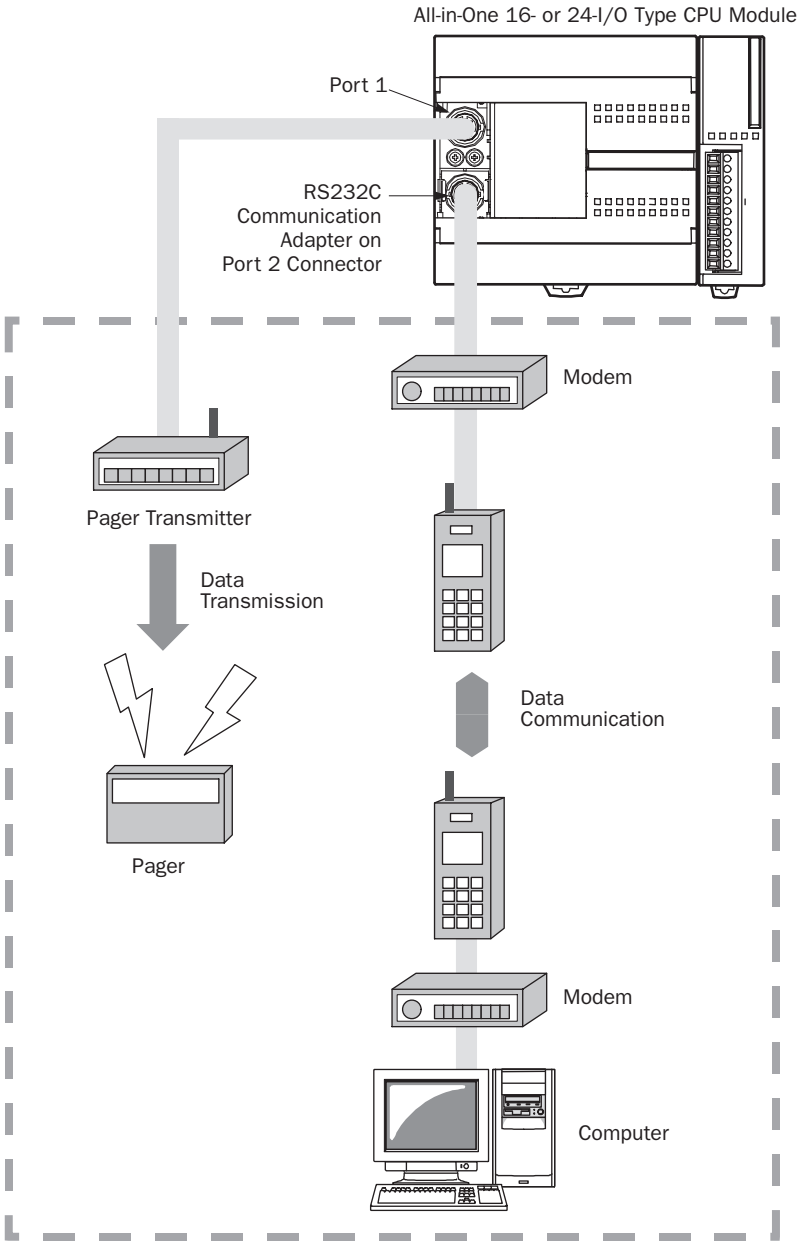
The all-in-one 16- and 24-I/O type MicroSmart CPU modules have port 1 for RS232C communication and port 2 connector. An optional RS232C or RS485 communication adapter can be installed on the port 2 connector. With an RS232C communication adapter installed on port 2, the 16- or 24-I/O type MicroSmart CPU module can communicate with two RS232C devices at the same time.

The figure below illustrates a system setup of user communication and modem communication. In this example, the operating status of a remote machine is monitored on a computer through modems connected to port 2 and the data is transferred through port 1 to a pager transmitter using the user communication.

The same system can be set up using any slim type CPU module and an optional RS232C communication module.

For details about the user communication, see page 17-1.

For details about the modem mode, see page 27-1.

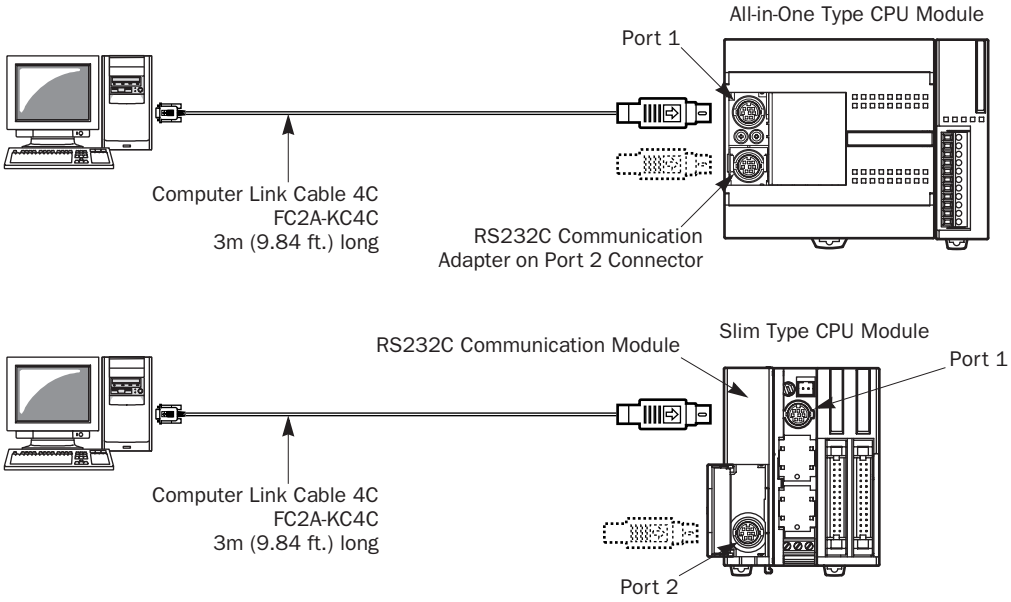


Computer Link System

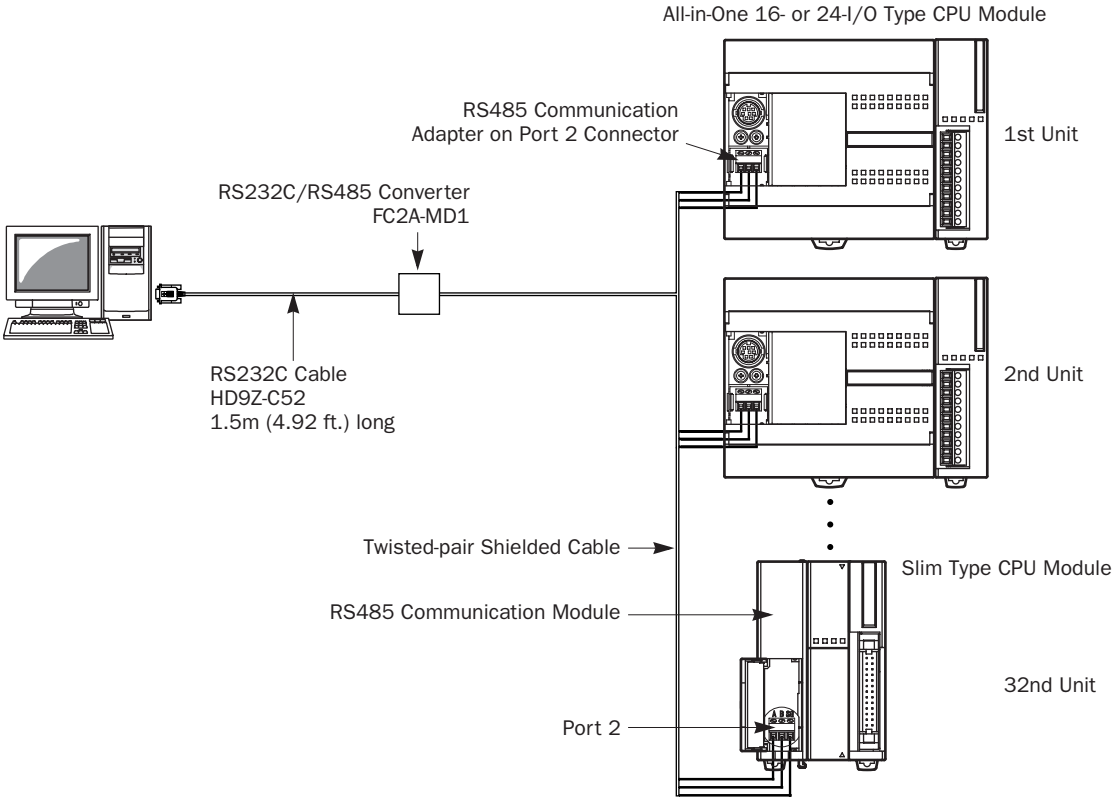
When the MicroSmart is connected to a computer, operating status and I/O status can be monitored on the computer, data in the CPU module can be monitored or updated, and user programs can be downloaded and uploaded. When an optional RS485 communication adapter is installed on the port 2 connector of the all-in-one 16- or 24-I/O type CPU modules or when an optional RS485 communication module is mounted with any slim type CPU modules, a maximum of 32 CPU modules can be connected to one computer in the 1:N computer link system.

For details about the computer link communication, see pages 4-1 and 26-1.

Computer Link 1:1 Communication



Computer Link 1:N Communication

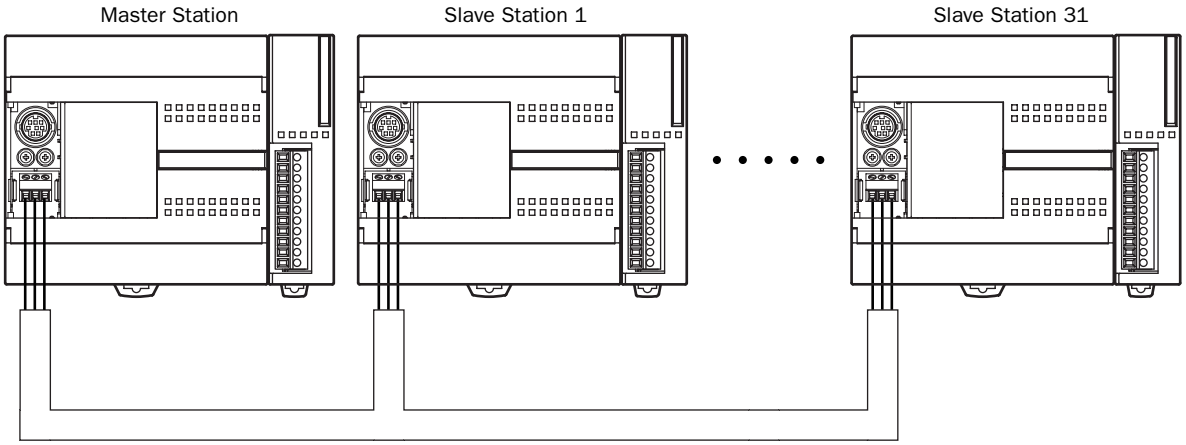


Data Link System

With an optional RS485 communication adapter installed on the port 2 connector, one 16- or 24-I/O type CPU module at the master station can communicate with 31 slave stations through the RS485 line to exchange data and perform distributed control effectively. The RS485 terminals are connected with each other using a 2-core twisted pair cable.

The same data link system can also be set up using any slim type CPU modules mounted with RS485 communication modules.

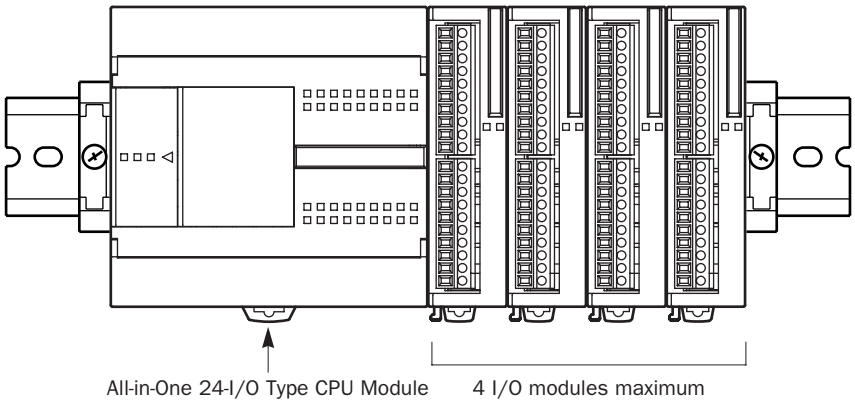
For details about the data link communication, see page 25-1.



Basic System

The all-in-one 10-I/O type CPU module has 6 input terminals and 4 output terminals. The 16-I/O type CPU module has 9 input terminals and 7 output terminals. The 24-I/O type CPU module has 14 input terminals and 10 output terminals. Only the 24-I/O type CPU module has an expansion connector to connect I/O modules. When four 16-point input or output modules are connected to the 24-I/O type CPU module, the I/O points can be expanded to a maximum of 88 points.

Any slim type CPU module can add a maximum of seven expansion I/O modules.

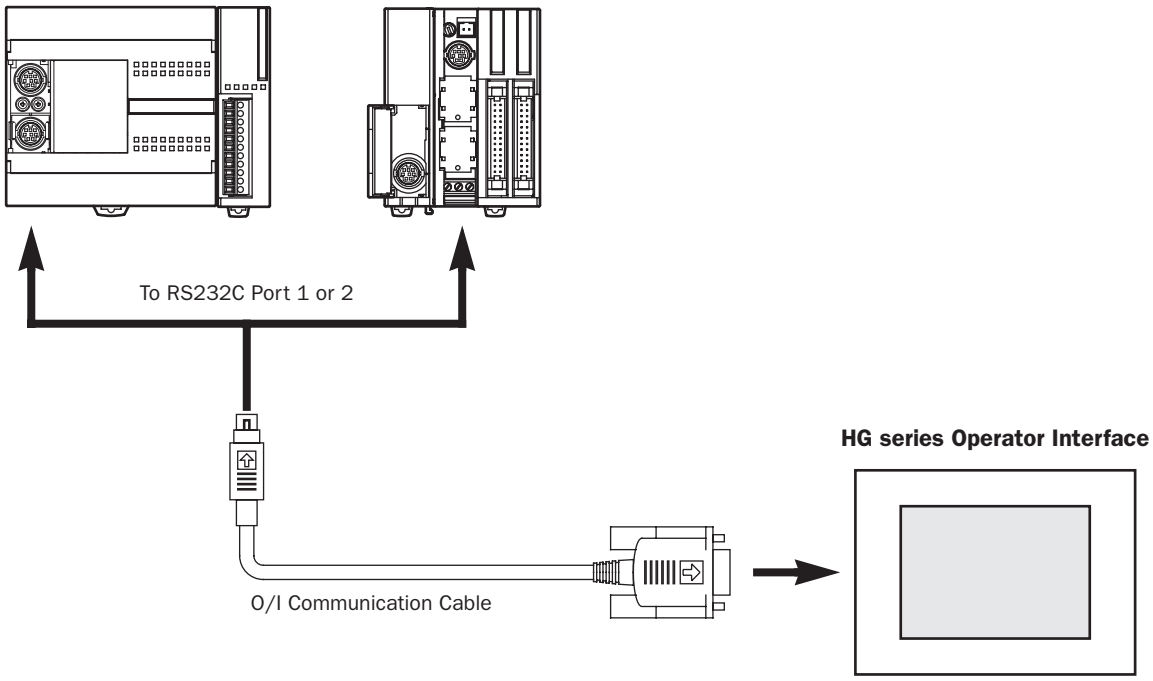


Operator Interface Communication System

The MicroSmart can communicate with IDEC's HG series operator interfaces through RS232C port 1 and port 2.

Optional cables are available for connection between the MicroSmart and HG series operator interfaces. When installing an optional RS232C communication adapter on the all-in-one type CPU module or an optional RS232C communication module on the slim type CPU module, two operator interfaces can be connected to one MicroSmart CPU module.

For details about communication settings, see the user's manual for the operator interface.



Applicable Cables to Operator Interfaces

Operator Interface	O/I Communication Cable	For Use on MicroSmart
HG1B, HG2A Series	FC4A-KC1C	RS232C port 1 and port 2
	HG9Z-XC183	Port 2 only
HG2F, HG3F, HG4F Series	FC4A-KC2C	RS232C port 1 and port 2
	HG9Z-3C125	Port 2 only

AS-Interface Network

Actuator-Sensor-Interface, abbreviated AS-Interface

The MicroSmart can be connected to the AS-Interface network using the AS-Interface master module (FC4A-AS62M).

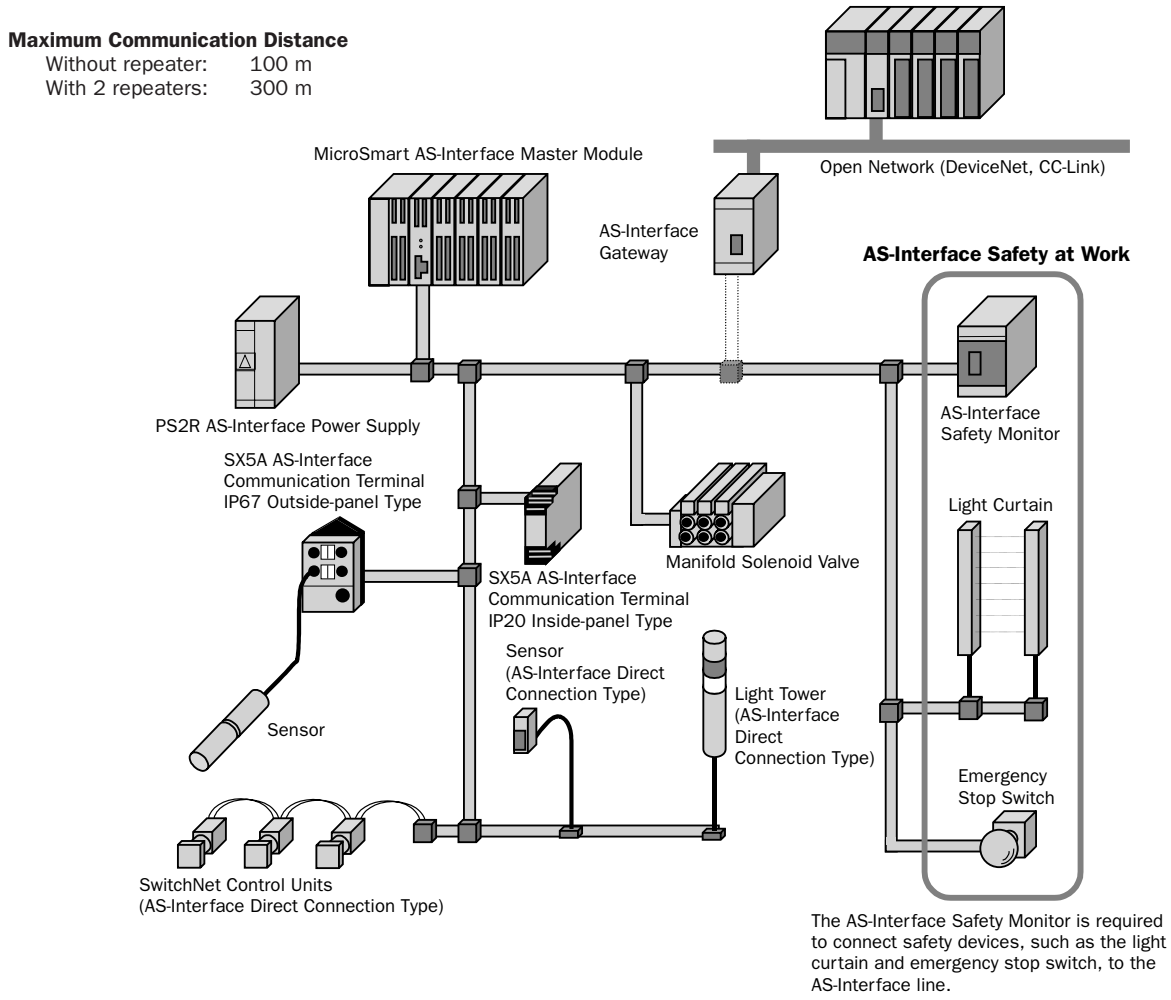
AS-Interface is a type of field bus that is primarily intended to be used to control sensors and actuators. AS-Interface is a network system that is compatible with the IEC62026 standard and is not proprietary to any one manufacturer. A master device can communicate with slave devices such as sensors, actuators, and remote I/Os, using digital and analog signals transmitted over the AS-Interface bus.

The AS-Interface system is comprised of the following three major components:

- One master, such as the MicroSmart AS-Interface master module
- One or more slave devices, such as sensors, actuators, switches, and indicators
- Dedicated 30V DC AS-Interface power supply (26.5 to 31.6V DC)

These components are connected using a two-core cable for both data transmission and AS-Interface power supply. AS-Interface employs a simple yet efficient wiring system and features automatic slave address assignment function, while installation and maintenance are also very easy.

For details about AS-Interface communication, see page 2-58 and chapter 28.



SwitchNet™ SwitchNet is an IDEC trademark for pushbuttons, pilot lights, and other control units capable of direct connection to the AS-Interface. SwitchNet devices are completely compatible with AS-Interface Ver. 2.1.

2: MODULE SPECIFICATIONS

Introduction

This chapter describes MicroSmart modules, parts names, and specifications of each module.

Available modules include all-in-one type and slim type CPU modules, digital input modules, digital output modules, mixed I/O modules, analog I/O modules, HMI module, HMI base module, communication adapters, communication modules, memory cartridge, and clock cartridge.

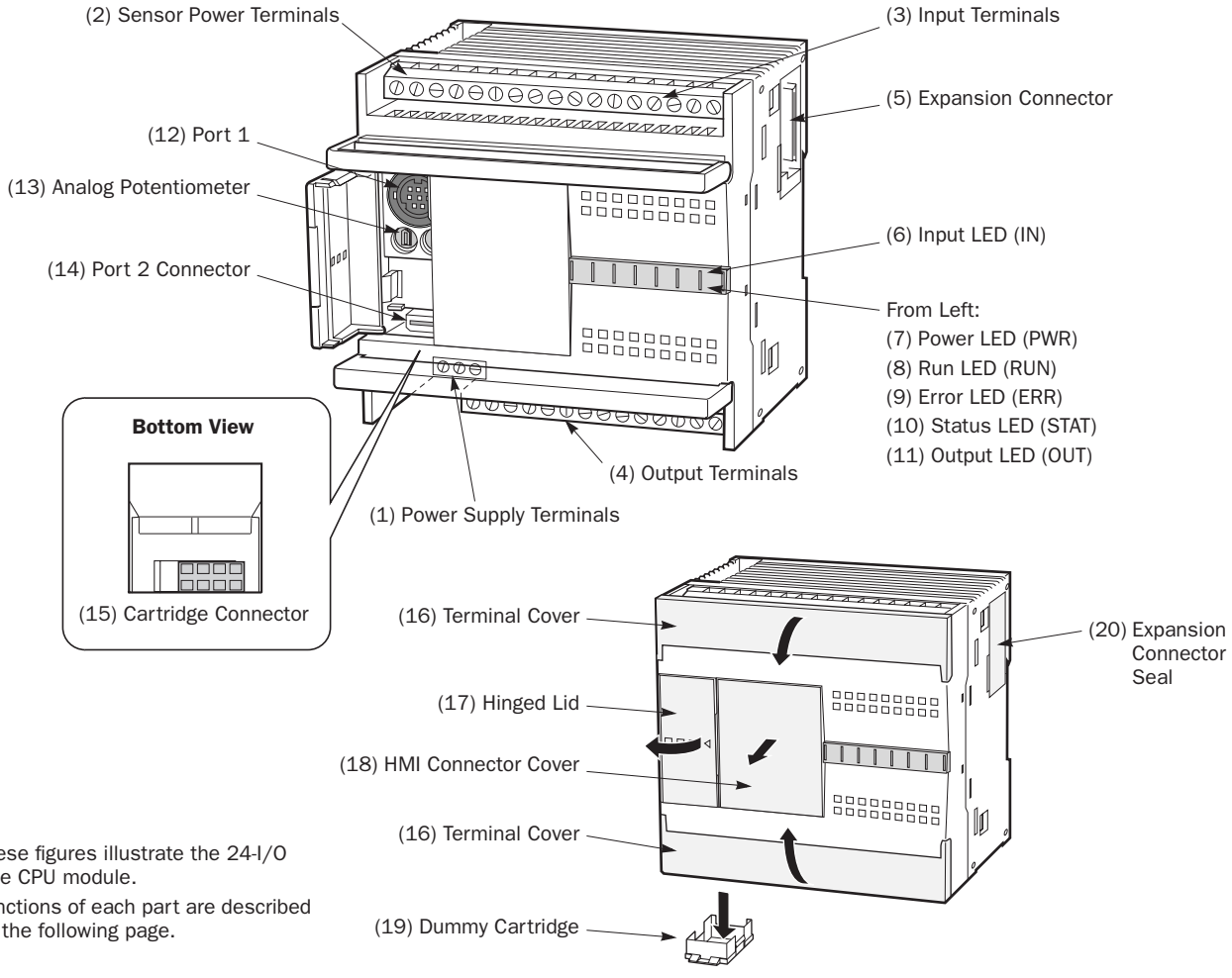
CPU Modules (All-in-One Type)

All-in-one type CPU modules are available in 10-, 16-, and 24-I/O types. The 10-I/O type has 6 input and 4 output terminals, the 16-I/O type 9 input and 7 output terminals, and the 24-I/O type 14 input and 10 output terminals. Every all-in-one type CPU module has communication port 1 for RS232C communication. In addition, the 16- and 24-I/O type CPU modules have port 2 connector to install an optional RS232C or RS485 communication adapter for 1:N computer link, modem communication, or data link communication. Every all-in-one type CPU module has a cartridge connector to install an optional memory cartridge or clock cartridge.

CPU Module Type Numbers (All-in-One Type)

Power Voltage	10-I/O Type	16-I/O Type	24-I/O Type
100 -240V AC (50/60 Hz)	FC4A-C10R2	FC4A-C16R2	FC4A-C24R2
24V DC	FC4A-C10R2C	FC4A-C16R2C	FC4A-C24R2C

Parts Description (All-in-One Type)



These figures illustrate the 24-I/O type CPU module.

Functions of each part are described on the following page.

(1) Power Supply Terminals
Connect power supply to these terminals. Power voltage 100-240V AC or 24V DC. See page 3-16.

(2) Sensor Power Terminals (AC power type only)
For supplying power to sensors (24V DC, 250mA). These terminals can be used for supplying power to input circuits. Use the sensor power supply only for supplying power to input devices connected to the MicroSmart.

(3) Input Terminals
For connecting input signals from input devices such as sensors, pushbuttons, and limit switches. The input terminals accept both sink and source DC input signals.

(4) Output Terminals
For connecting output signals to output devices such as electromechanical relays and solenoid valves. The internal output relay is rated at 240V AC/2A or 30V DC/2A.

(5) Expansion Connector (24-I/O type CPU module only)
For connecting digital and analog I/O modules to the 24-I/O type CPU module.

(6) Input LED (IN)
Turns on when a corresponding input is on.

(7) Power LED (PWR)
Turns on when power is supplied to the CPU module.

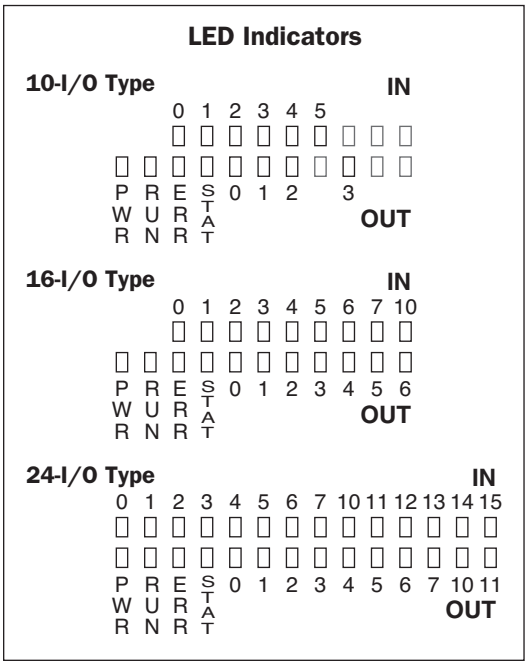
(8) Run LED (RUN)
Turns on when the CPU module is executing the user program.

(9) Error LED (ERR)
Turns on when an error has occurred in the CPU module.

(10) Status LED (STAT)
The status LED can be turned on or off using the user program to indicate a specified status.

(11) Output LED (OUT)
Turns on when a corresponding output is on.

(12) Port 1 (RS232C)
For connecting a computer to download a user program and monitor the PLC operation on a computer using WindLDR.



(13) Analog Potentiometer
Sets a value of 0 through 255 to a special data register. The 10- and 16-I/O types have one potentiometer. The 24-I/O type has two potentiometers. The analog potentiometer can be used to set a preset value for an analog timer.

(14) Port 2 Connector (16- and 24-I/O type CPU modules only)
For connecting an optional RS232C or RS485 communication adapter.

(15) Cartridge Connector
For connecting an optional memory cartridge or clock cartridge.

(16) Terminal Cover
For protecting the input and output terminals. When wiring the terminals, open the covers.

(17) Hinged Lid
Open the lid to gain access to the port 1, port 2 connector, and analog potentiometer.

(18) HMI Connector Cover
Remove the HMI connector cover when using an optional HMI module.

(19) Dummy Cartridge
Remove the dummy cartridge when using an optional memory cartridge or clock cartridge.

(20) Expansion Connector Seal (24-I/O type CPU module only)
Remove the expansion connector seal when connecting a digital or analog I/O module.

General Specifications (All-in-One Type CPU Module)

Normal Operating Conditions

CPU Module		FC4A-C10R2	FC4A-C16R2	FC4A-C24R2
	DC Power Type	FC4A-C10R2C	FC4A-C16R2C	FC4A-C24R2C
Operating Temperature		0 to 55°C (operating ambient temperature)		
Storage Temperature		–25 to +70°C		
Relative Humidity		10 to 95% (non-condensing)		
Pollution Degree		2 (IEC 60664-1)		
Degree of Protection		IP20 (IEC 60529)		
Corrosion Immunity		Atmosphere free from corrosive gases		
Altitude		Operation: 0 to 2,000m (0 to 6,565 feet) Transport: 0 to 3,000m (0 to 9,840 feet)		
Vibration Resistance		When mounted on a DIN rail or panel surface: 5 to 9 Hz amplitude 3.5 mm, 9 to 150 Hz acceleration 9.8 m/s^2 (1G) 2 hours per axis on each of three mutually perpendicular axes (IEC 61131-2)		
Shock Resistance		147 m/s^2 (15G), 11 ms duration, 3 shocks per axis on three mutually perpendicular axes (IEC 61131-2)		
ESD Immunity		Contact discharge: ±6 kV, Air discharge: ±8 kV (IEC 61000-4-2)		
Weight	**AC Power Type**	230g	250g	305g
	DC Power Type	240g	260g	310g

Power Supply

Rated Power Voltage		AC power type: 100 to 240V AC, DC power type: 24V DC		
Allowable Voltage Range		AC power type: 85 to 264V AC, DC power type: 20.4 to 28.8V DC (including ripple)		
Rated Power Frequency		AC power type: 50/60 Hz (47 to 63 Hz)		
Maximum Input Current		250 mA (85V AC) 160 mA (24V DC)	300 mA (85V AC) 190 mA (24V DC)	450 mA (85V AC) 360 mA (24V DC)
Maximum Power Consumption	**AC Power Type**	FC4A-C10R2: 30VA (264V AC), 20VA (100V AC) (CPU module*) FC4A-C16R2: 31VA (264V AC), 22VA (100V AC) (CPU module*) FC4A-C24R2: 40VA (264V AC), 33VA (100V AC) (CPU module* + 4 I/O modules) *The CPU module power consumption includes 250mA sensor power.		
	DC Power Type	FC4A-C10R2C: 3.9W (24V DC) (CPU module) FC4A-C16R2C: 4.6W (24V DC) (CPU module) FC4A-C24R2C: 8.7W (24V DC) (CPU module + 4 I/O modules)		
Allowable Momentary Power Interruption		10 ms (at the rated power voltage)		
Dielectric Strength		Between power and ⏚ or ⏚ terminals: 1,500V AC, 1 minute Between I/O and ⏚ or ⏚ terminals: 1,500V AC, 1 minute		
Insulation Resistance		Between power and ⏚ or ⏚ terminals: 10 MΩ minimum (500V DC megger) Between I/O and ⏚ or ⏚ terminals: 10 MΩ minimum (500V DC megger)		
Noise Resistance		AC or DC power terminals: 1.5 kV, 50 nsec to 1 µs I/O terminals (coupling clamp): 1.5 kV, 50 nsec to 1 µs		
Inrush Current		35A maximum	35A maximum	40A maximum
Grounding Wire		UL1007 AWG16		
Power Supply Wire		UL1015 AWG22, UL1007 AWG18		
Effect of Improper Power Supply Connection		Reverse polarity: Normal operation (AC), No operation, no damage (DC) Improper voltage or frequency: Permanent damage may be caused Improper lead connection: Permanent damage may be caused		

Note: The maximum number of relay outputs that can be turned on simultaneously is 33 points (AC power type CPU module) or 44 points (DC power type CPU module) including relay outputs on the CPU module.

Function Specifications (All-in-One Type CPU Module)

CPU Module Specifications

CPU Module		FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	
Program Capacity		4,800 bytes (800 steps)	15,000 bytes (2,500 steps)	27,000 bytes (4,500 steps)	
Expandable I/O Modules		—	—	4 modules	
I/O Points	Input	6	9	14	Expansion: 64
	Output	4	7	10	
User Program Storage		EEPROM			
RAM Backup	Backup Duration	Approx. 30 days (typical) at 25°C after backup battery fully charged			
	Backup Data	Internal relay, shift register, counter, data register			
	Battery	Lithium secondary battery			
	Charging Time	Approx. 24 hours for charging from 0% to 90% of full charge at 25°C			
	Battery Life	Approx. 1000 cycles of discharging down to 10% and full charging			
	Replaceability	Not possible to replace battery			
Control System		Stored program system			
Instruction Words		35 basic 38 advanced	35 basic 40 advanced	35 basic 48 advanced	
Processing Time	Basic instruction	1.65 ms (1000 steps) See page A-1.			
	END processing	0.64 ms (not including expansion I/O service, clock function processing, data link processing, and interrupt processing) See page A-2.			
Internal Relay		256	1024	1024	
Shift Register		64	128	128	
Data Register		400	1300	1300	
Counter (adding, dual pulse reversible, up/down selection reversible)		32	100	100	
Timer (1-sec, 100-ms, 10-ms, 1-ms)		32	100	100	
Input Filter		3 to 15 ms (selectable in increments of 1 ms)			
Catch Input Interrupt Input		Four inputs (I2 through I5) can be designated as catch inputs or interrupt inputs Minimum turn on pulse width: 40 µs maximum Minimum turn off pulse width: 150 µs maximum			
Self-diagnostic Function		Power failure, watchdog timer, data link connection, user program EEPROM sum check, timer/counter preset value sum check, user program RAM sum check, keep data, user program syntax, user program writing, CPU module, clock IC, I/O bus initialize, user program execution			
Start/Stop Method		Turning power on and off Start/stop command in WindLDR Turning start control special internal relay M8000 on and off Turning designated stop or reset input off and on			

System Statuses at Stop, Reset, and Restart

Mode	Output	Internal Relay, Shift Register, Counter, Data Register		Timer Current Value
		Keep Type	Clear Type	
Run	Operating	Operating	Operating	Operating
Stop (Stop input ON)	OFF	Unchanged	Unchanged	Unchanged
Reset (Reset input ON)	OFF	OFF/Reset to zero	OFF/Reset to zero	Reset to zero
Restart	Unchanged	Unchanged	OFF/Reset to zero	Reset to preset

Communication Function

Communication Port	Port 1 (RS232C)	Port 2 (RS232C) Communication Adapter	Port 2 (RS485) Communication Adapter
Standards	EIA RS232C	EIA RS232C	EIA RS485
Maximum Baud Rate	19,200 bps	19,200 bps	Computer link: 19,200 bps Data link: 38,400 bps
Maintenance Communication (Computer Link)	Possible	Possible	Possible
User Communication	Possible	Possible	Not possible
Modem Communication	Not possible	Possible	Not possible
Data Link Communication	Not possible	Not possible	Possible
Quantity of Slave Stations	—	—	31
Maximum Cable Length	Special cable	Special cable	200m *
Isolation between Internal Circuit and Communication Port	Not isolated	Not isolated	Not isolated

* Recommended cable for RS485: Twisted-pair shielded cable with a minimum core wire of 0.3 mm^2.
Conductor resistance 85 Ω/km maximum, shield resistance 20 Ω/km maximum.

Built-in Functions

High-speed Counter	Maximum Counting Frequency and High-speed Counter Points	Total 4 points Single/two-phase selectable: 20 kHz (1 point) Single-phase: 5 kHz (3 points)
	Counting Range	0 to 65535 (16 bits)
	Operation Mode	Rotary encoder mode and adding counter mode
Sensor Power Supply (AC power type only)	Output Voltage/Current	24V DC (+10% to –15%), 250 mA
	Overload Detection	Not available
	Isolation	Isolated from the internal circuit
Analog Potentiometer	Quantity	1 point (10- and 16-I/O type CPU) 2 points (24-I/O type CPU)
	Data Range	0 to 255

Memory Cartridge (Option)

Memory Type	EEPROM
Accessible Memory Capacity	32 KB
Hardware for Storing Data	CPU module
Software for Storing Data	WindLDR
Quantity of Stored Programs	One user program can be stored on one memory cartridge.
Program Execution Priority	When a memory cartridge is installed, the user program on the memory cartridge is executed.

Clock Cartridge (Option)

Accuracy	±30 sec/month (typical) at 25°C
Backup Duration	Approx. 30 days (typical) at 25°C after backup battery fully charged
Battery	Lithium secondary battery
Charging Time	Approx. 10 hours for charging from 0% to 90% of full charge
Battery Life	Approx. 100 recharge cycles after discharging down to 10% of full charge
Replaceability	Not possible to replace battery

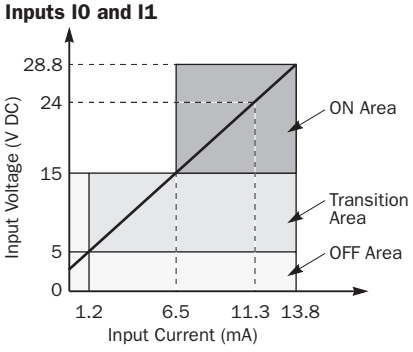
DC Input Specifications (All-in-One Type CPU Module)

CPU Module	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C
Input Points and Common Line	6 points in 1 common line	9 points in 1 common line	14 points in 1 common line
Terminal Arrangement	See CPU Module Terminal Arrangement on pages 2-8 and 2-9.		
Rated Input Voltage	24V DC sink/source input signal		
Input Voltage Range	20.4 to 28.8V DC		
Rated Input Current	I0 and I1: 11 mA I2 to I7, I10 to I15: 7 mA/point (24V DC)		
Input Impedance	I0 and I1: 2.1 kΩ I2 to I7, I10 to I15: 3.4 kΩ		
Turn ON Time	I0 to I5: 35 µs + filter value I6, I7, I10 to I15: 40 µs + filter value		
Turn OFF Time	I0 and I1: 45 µs + filter value I2 to I7, I10 to I15: 150 µs + filter value		
Isolation	Between input terminals: Not isolated Internal circuit: Photocoupler isolated		
Input Type	Type 1 (IEC 61131)		
External Load for I/O Interconnection	Not needed		
Signal Determination Method	Static		
Effect of Improper Input Connection	Both sinking and sourcing input signals can be connected. If any input exceeding the rated value is applied, permanent damage may be caused.		
Cable Length	3m (9.84 ft.) in compliance with electromagnetic immunity		

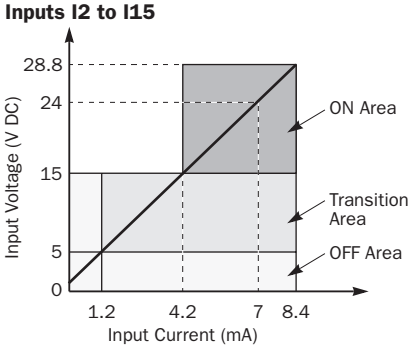
Input Operating Range

The input operating range of the Type 1 (IEC 61131-2) input module is shown below:

Inputs I0 and I1

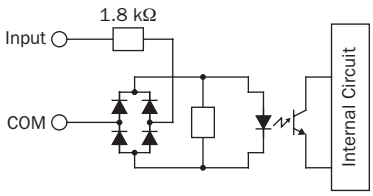


Inputs I2 to I15

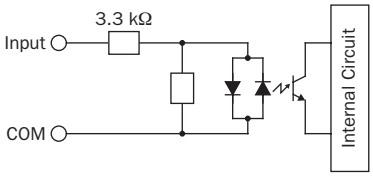


Input Internal Circuit

Inputs I0 and I1

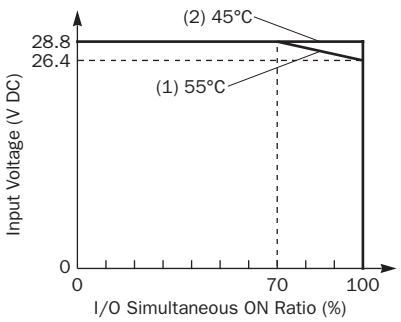


Inputs I2 to I15



I/O Usage Limits

When using the FC4A-C16R2/C or FC4A-C24R2/C at an ambient temperature of 55°C in the normal mounting direction, limit the inputs and outputs, respectively, which turn on simultaneously along line (1).



When using at 45°C, all I/Os can be turned on simultaneously at input voltage 28.8V DC as indicated with line (2).

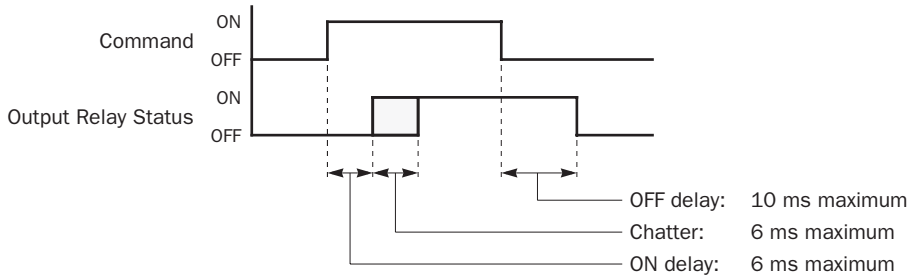
When using the FC4A-C10R2/C, all I/Os can be turned on simultaneously at 55°C, input voltage 28.8V DC.

For other possible mounting directions, see page 3-12.

Relay Output Specifications (All-in-One Type CPU Module)

CPU Module		FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C
No. of Outputs		4 points	7 points	10 points
Output Points per Common Line	**COM0**	3 NO contacts	4 NO contacts	4 NO contacts
	COM1	1 NO contact	2 NO contacts	4 NO contacts
	COM2	—	1 NO contact	1 NO contact
	COM3	—	—	1 NO contact
Terminal Arrangement		See CPU Module Terminal Arrangement on pages 2-8 and 2-9.		
Maximum Load Current		2A per point 8A per common line		
Minimum Switching Load		0.1 mA/0.1V DC (reference value)		
Initial Contact Resistance		30 mΩ maximum		
Electrical Life		100,000 operations minimum (rated load 1,800 operations/hour)		
Mechanical Life		20,000,000 operations minimum (no load 18,000 operations/hour)		
Rated Load (resistive/inductive)		240V AC/2A, 30V DC/2A		
Dielectric Strength		Between output and ⏚ or ⏚ terminals: 1,500V AC, 1 minute Between output terminal and internal circuit: 1,500V AC, 1 minute Between output terminals (COMs): 1,500V AC, 1 minute		
Contact Protection Circuit for Relay Output		See page 3-15.		

Output Delay



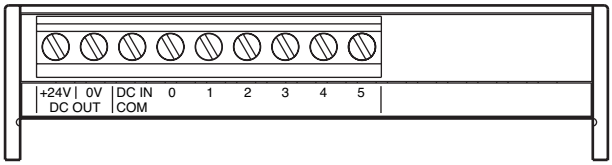
CPU Module Terminal Arrangement (All-in-One Type)

The input and output terminal arrangements of the all-in-one type CPU modules are shown below.

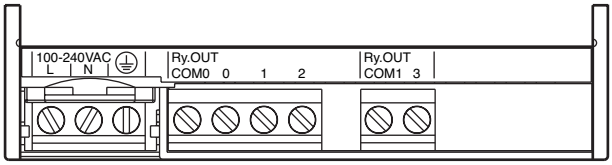
AC Power Type CPU Module

FC4A-C10R2

Sensor Power Terminals
Input Terminals

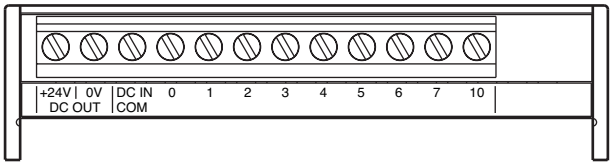


AC Power Terminals
Output Terminals

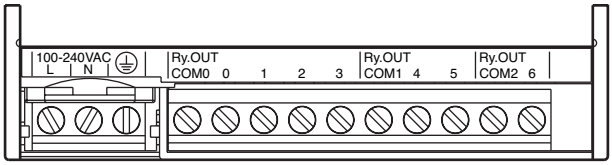


FC4A-C16R2

Sensor Power Terminals
Input Terminals

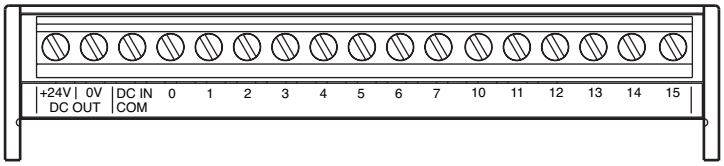


AC Power Terminals
Output Terminals

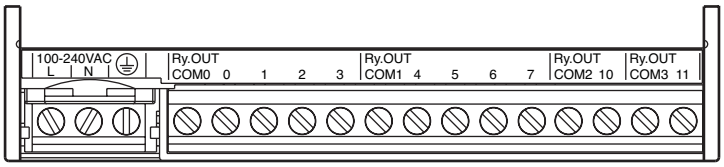


FC4A-C24R2

Sensor Power Terminals
Input Terminals



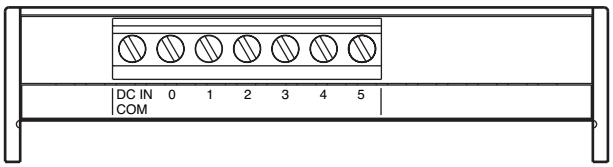
AC Power Terminals
Output Terminals



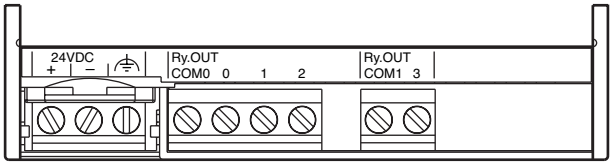
DC Power Type CPU Module

FC4A-C10R2C

Input Terminals

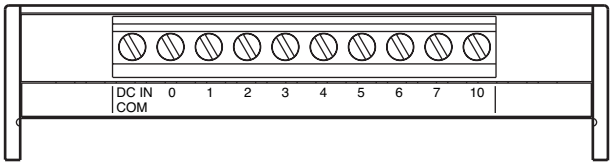


DC Power Terminals
Output Terminals

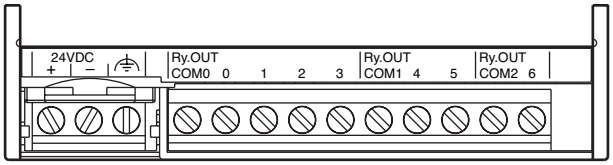


FC4A-C16R2C

Input Terminals

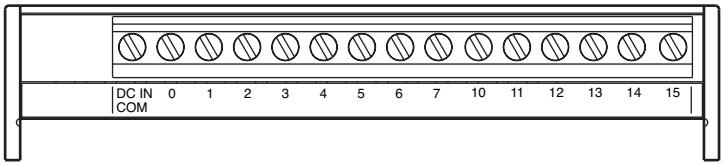


DC Power Terminals
Output Terminals

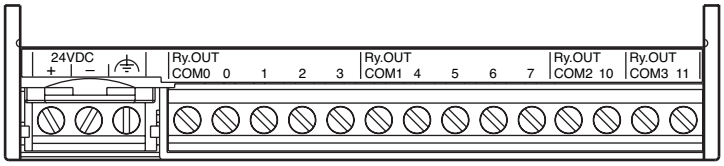


FC4A-C24R2C

Input Terminals

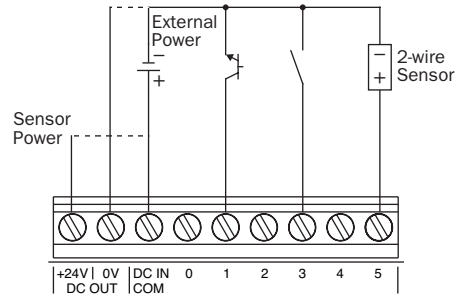
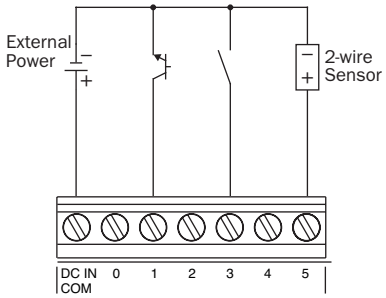
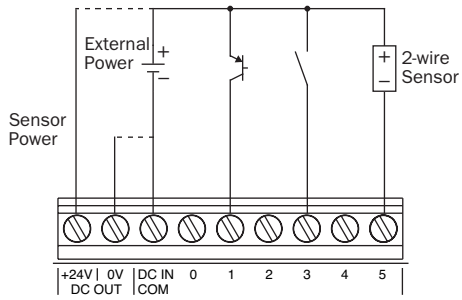
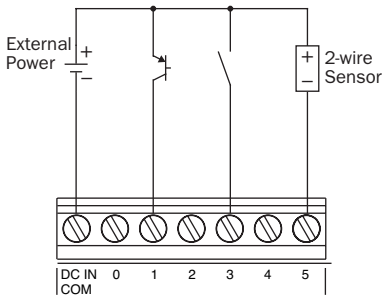
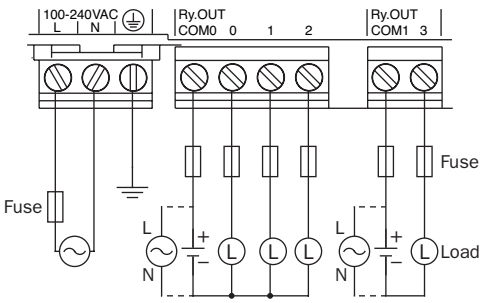
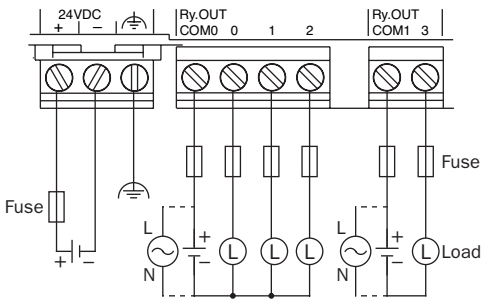


DC Power Terminals
Output Terminals



I/O Wiring Diagrams (All-in-One Type CPU Module)

The input and output wiring examples of the CPU modules are shown below. For wiring precautions, see pages 3-13 through 3-16.

AC Power Type CPU Module	DC Power Type CPU Module
DC Source Input Wiring	**DC Source Input Wiring**
	
DC Sink Input Wiring	**DC Sink Input Wiring**
	
AC Power and Relay Output Wiring	**DC Power and Relay Output Wiring**
	

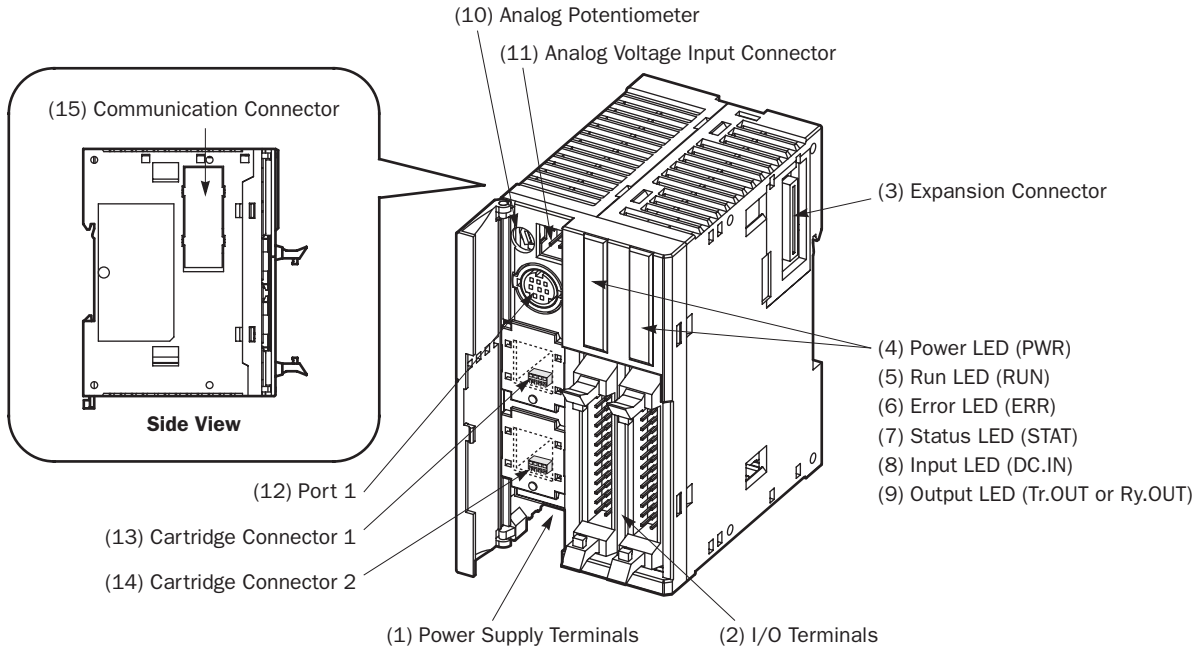
CPU Modules (Slim Type)

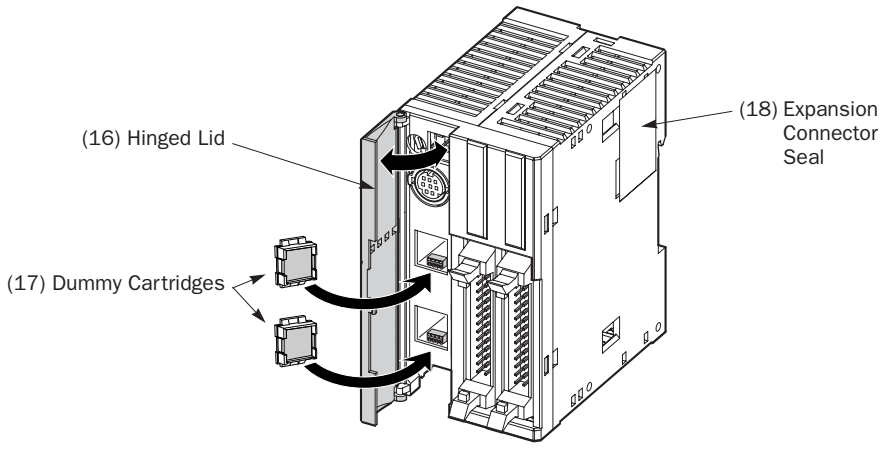
Slim type CPU modules are available in 20- and 40-I/O types. The 20-I/O type has 12 input and 8 output terminals, and the 40-I/O type has 24 input and 16 output terminals. The FC4A-D20RK1 and FC4A-D20RS1 have 2 transistor outputs used for high-speed outputs and pulse outputs in addition to 10 relay outputs. Every slim type CPU module has communication port 1 for RS232C communication, and can mount an optional RS232C or RS485 communication module for 1:N computer link, modem communication, and data link communication. The HMI base module can also be mounted to install an optional HMI module and a communication adapter. Every slim type CPU module has two cartridge connectors to install an optional memory cartridge and a clock cartridge.

CPU Module Type Numbers (Slim Type)

I/O Points	Output Type	High-speed Transistor Output (Q0 & Q1)	Type No.
20 (12 in / 8 out)	Transistor Sink Output 0.3A		FC4A-D20K3
	Transistor Source Output 0.3A		FC4A-D20S3
20 (12 in / 8 out)	Relay Output 240V AC/30V DC, 2A	Sink Output 0.3A	FC4A-D20RK1
		Source Output 0.3A	FC4A-D20RS1
40 (24 in / 16 out)	Transistor Sink Output 0.3A		FC4A-D40K3
	Transistor Source Output 0.3A		FC4A-D40S3

Parts Description (Slim Type)





These figures illustrate the 40-I/O type CPU module.

Functions of each part are described on the following page.

(1) Power Supply Terminals
Connect power supply to these terminals. Power voltage 24V DC. See page 3-17.

(2) I/O Terminals
For connecting input and output signals. The input terminals accept both sink and source 24V DC input signals. Transistor and relay output types are available. Transistor output type has MIL connectors and relay output type has removable screw connectors.

(3) Expansion Connector
For connecting digital and analog I/O modules.

(4) Power LED (PWR)
Turns on when power is supplied to the CPU module.

(5) Run LED (RUN)
Turns on when the CPU module is executing the user program.

(6) Error LED (ERR)
Turns on when an error occurs in the CPU module.

(7) Status LED (STAT)
The status LED can be turned on or off using the user program to indicate a specified status.

(8) Input LED (IN)
Turns on when a corresponding input is on.

(9) Output LED (Tr.OUT or Ry.OUT)
Turns on when a corresponding output is on.

(10) Analog Potentiometer
Sets a value of 0 through 255 to a special data register. All slim type CPU modules have one potentiometer, which can be used to set a preset value for an analog timer.

(11) Analog Voltage Input Connector
For connecting an analog voltage source of 0 through 10V DC. The analog voltage is converted to a value of 0 through 255 and stored to a special data register.

(12) Port 1 (RS232C)
For connecting a computer to download a user program and monitor the PLC operation on a computer using WindLDR.

(13) Cartridge Connector 1
For connecting an optional memory cartridge or clock cartridge.

(14) Cartridge Connector 2
For connecting an optional memory cartridge or clock cartridge.

(15) Communication Connector
For connecting an optional communication module or HMI base module. Remove the connector cover before connecting a module.

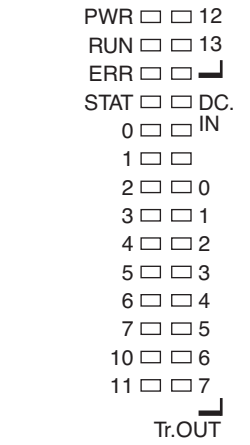
(16) Hinged Lid
Open the lid to gain access to the port 1, cartridge connectors 1 and 2, analog potentiometer, and analog voltage input connector.

(17) Dummy Cartridges
Remove the dummy cartridge when using an optional memory cartridge or clock cartridge.

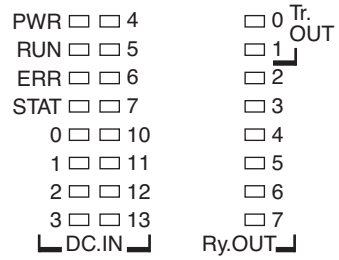
(18) Expansion Connector Seal
Remove the expansion connector seal when connecting a digital I/O or analog I/O module.

LED Indicators

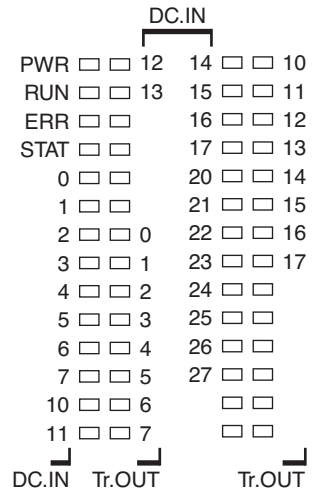
20-I/O Type (Transistor Output)



20-I/O Type (Relay Output)



40-I/O Type (Transistor Output)



General Specifications (Slim Type CPU Module)

Normal Operating Conditions

CPU Module	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1	FC4A-D40K3 FC4A-D40S3
Operating Temperature	0 to 55°C (operating ambient temperature)		
Storage Temperature	–25 to +70°C		
Relative Humidity	10 to 95% (non-condensing)		
Pollution Degree	2 (IEC 60664-1)		
Degree of Protection	IP20 (IEC 60529)		
Corrosion Immunity	Atmosphere free from corrosive gases		
Altitude	Operation: 0 to 2,000m (0 to 6,565 feet) Transport: 0 to 3,000m (0 to 9,840 feet)		
Vibration Resistance	When mounted on a DIN rail or panel surface: 5 to 9 Hz amplitude 3.5 mm, 9 to 150 Hz acceleration 9.8 m/s^2 (1G) 2 hours per axis on each of three mutually perpendicular axes (IEC 61131-2)		
Shock Resistance	147 m/s^2 (15G), 11 ms duration, 3 shocks per axis on three mutually perpendicular axes (IEC 61131-2)		
ESD Immunity	Contact discharge: ±6 kV, Air discharge: ±8 kV (IEC 61000-4-2)		
Weight	140g	185g	180g

Power Supply

Rated Power Voltage	24V DC		
Allowable Voltage Range	20.4 to 26.4V DC (including ripple)		
Maximum Input Current	560 mA (26.4V DC)	700 mA (26.4V DC)	700 mA (26.4V DC)
Maximum Power Consumption	CPU module + 7 I/O modules		
	14W (26.4V DC)	17W (26.4V DC)	17W (26.4V DC)
Allowable Momentary Power Interruption	10 ms (at 24V DC)		
Dielectric Strength	Between power and ⏚ terminals: 500V AC, 1 minute Between I/O and ⏚ terminals: 1,500V AC, 1 minute		
Insulation Resistance	Between power and ⏚ terminals: 10 MΩ minimum (500V DC megger) Between I/O and ⏚ terminals: 10 MΩ minimum (500V DC megger)		
Noise Resistance	DC power terminals: 1.0 kV, 50 nsec to 1 µs I/O terminals (coupling clamp): 1.5 kV, 50 nsec to 1 µs		
Inrush Current	50A maximum (24V DC)		
Grounding Wire	UL1015 AWG22, UL1007 AWG18		
Power Supply Wire	UL1015 AWG22, UL1007 AWG18		
Effect of Improper Power Supply Connection	Reverse polarity: No operation, no damage Improper voltage or frequency: Permanent damage may be caused Improper lead connection: Permanent damage may be caused		

Note: The maximum number of relay outputs that can be turned on simultaneously is 96 points including relay outputs on the CPU module.

Function Specifications (Slim Type CPU Module)

CPU Module Specifications

CPU Module		FC4A-D20K3 FC4A-D20S3		FC4A-D20RK1 FC4A-D20RS1		FC4A-D40K3 FC4A-D40S3	
Program Capacity		27,000 bytes (4,500 steps)		31,200 bytes (5,200 steps) 64,500 bytes (10,750 steps) (Note 1, Note 2)			
Expandable I/O Modules		7 modules					
I/O Points	Input	12	Expansion: 128	12	Expansion: 224	24	Expansion: 224
	Output	8		8		16	
User Program Storage		EEPROM					
RAM Backup	Backup Duration	Approx. 30 days (typical) at 25°C after backup battery fully charged					
	Backup Data	Internal relay, shift register, counter, data register, expansion data register					
	Battery	Lithium secondary battery					
	Charging Time	Approx. 24 hours for charging from 0% to 90% of full charge at 25°C					
	Battery Life	Approx. 1000 cycles of discharging down to 10% and full charging					
	Replaceability	Not possible to replace battery					
Control System		Stored program system					
Instruction Words		35 basic 53 advanced		35 basic 72 advanced			
Processing Time	Basic instruction	1.65 ms (1000 steps) See page A-1.					
	END processing	0.64 ms (not including expansion I/O service, clock function processing, data link processing, and interrupt processing) See page A-2.					
Internal Relay		1024		1024 + 560 for AS-Interface operands (Note 2)			
Shift Register		128					
Data Register		1300		1300 + 300 for AS-Interface operands (Note 2)			
Expansion Data Register		—		6,000			
Counter		100 (adding, dual pulse reversible, up/down selection reversible)					
Timer		100 (1-sec, 100-ms, 10-ms, 1-ms)					
Input Filter		3 to 15 ms (selectable in increments of 1 ms)					
Catch Input Interrupt Input		Four inputs (I2 through I5) can be designated as catch inputs or interrupt inputs Minimum turn on pulse width: 40 µs maximum Minimum turn off pulse width: 150 µs maximum					
Self-diagnostic Function		Power failure, watchdog timer, data link connection, user program EEPROM sum check, timer/counter preset value sum check, user program RAM sum check, keep data, user program syntax, user program writing, CPU module, clock IC, I/O bus initialize, user program execution					
Start/Stop Method		Turning power on and off Start/stop command in WindLDR Turning start control special internal relay M8000 on and off Turning designated stop or reset input off and on					

Note 1: When using a 64KB memory cartridge and WindLDR ver. 4.2 or higher.
Note 2: When using a CPU module with system program ver. 201 or higher and WindLDR ver. 4.2 or higher.

System Statuses at Stop, Reset, and Restart

Mode	Output	Internal Relay, Shift Register, Counter, Data Register, Expansion Data Register		Timer Current Value
		Keep Type	Clear Type	
Run	Operating	Operating	Operating	Operating
Stop (Stop input ON)	OFF	Unchanged	Unchanged	Unchanged
Reset (Reset input ON)	OFF	OFF/Reset to zero	OFF/Reset to zero	Reset to zero
Restart	Unchanged	Unchanged	OFF/Reset to zero	Reset to preset

Note: All expansion data registers are keep types. AS-Interface operands (M1300-M1977 and D1700-D1999) remain unchanged when the reset input is turned on.

Communication Function

Communication Port	Port 1 (RS232C)	Port 2 (RS232C) Communication Module Communication Adapter	Port 2 (RS485) Communication Module Communication Adapter
Standards	EIA RS232C	EIA RS232C	EIA RS485
Maximum Baud Rate	19,200 bps	19,200 bps	Computer link: 19,200 bps User comm.: 19,200 bps Data link: 38,400 bps
Maintenance Communication (Computer Link)	Possible	Possible	Possible
User Communication	Possible	Possible	Possible (Note 1)
Modem Communication	Not possible	Possible	Not possible
Data Link Communication	Not possible	Not possible	Possible
Quantity of Slave Stations	—	—	31
Maximum Cable Length	Special cable	Special cable	200m (Note 2)
Isolation between Internal Circuit and Communication Port	Not isolated	Not isolated	Not isolated

Note 1: RS485 user communication is available on upgraded CPU modules only, see page 17-1.
Note 2: Recommended cable for RS485: Twisted-pair shielded cable with a minimum core wire of 0.3 mm^2. Conductor resistance 85 Ω/km maximum, shield resistance 20 Ω/km maximum.

Built-in Functions

High-speed Counter	Maximum Counting Frequency and High-speed Counter Points	Total 4 points Single/two-phase selectable: 20 kHz (2 points) Single-phase: 5 kHz (2 points)
	Counting Range	0 to 65535 (16 bits)
	Operation Mode	Rotary encoder mode and adding counter mode
Analog Potentiometer	Quantity	1 point
	Data Range	0 to 255
Analog Voltage Input	Quantity	1 point
	Input Voltage Range	0 to 10V DC
	Input Impedance	Approx. 100 kΩ
	Data Range	0 to 255
Pulse Output	Quantity	2 points
	Maximum Frequency	20 kHz

Memory Cartridge (Option)

Memory Type	EEPROM
Accessible Memory Capacity	32 KB or 64 KB (64KB cartridge is for upgraded CPU modules only, see page 2-65)
Hardware for Storing Data	CPU module
Software for Storing Data	WindLDR
Quantity of Stored Programs	One user program can be stored on one memory cartridge.
Program Execution Priority	When a memory cartridge is installed, the user program on the memory cartridge is executed.

Clock Cartridge (Option)

Accuracy	±30 sec/month (typical) at 25°C
Backup Duration	Approx. 30 days (typical) at 25°C after backup battery fully charged
Battery	Lithium secondary battery
Charging Time	Approx. 10 hours for charging from 0% to 90% of full charge
Battery Life	Approx. 100 recharge cycles after discharging down to 10% of full charge
Replaceability	Not possible to replace battery

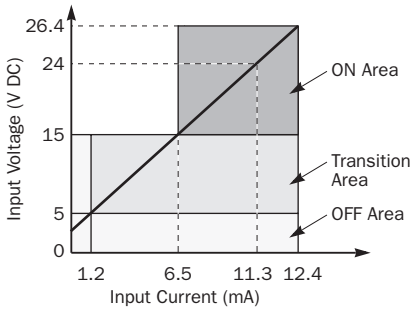
DC Input Specifications (Slim Type CPU Module)

CPU Module	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1	FC4A-D40K3 FC4A-D40S3
Input Points and Common Lines	12 points in 1 common line	12 points in 1 common line	24 points in 2 common lines
Terminal Arrangement	See CPU Module Terminal Arrangement on pages 2-19 through 2-22.		
Rated Input Voltage	24V DC sink/source input signal		
Input Voltage Range	20.4 to 26.4V DC		
Rated Input Current	I0, I1, I6, I7: 5 mA/point (24V DC) I2 to I5, I10 to I27: 7 mA/point (24V DC)		
Input Impedance	I0, I1, I6, I7: 5.7 kΩ I2 to I5, I10 to I27: 3.4 kΩ		
Turn ON Time	I0 to I7: 35 µs + filter value I10 to I27: 40 µs + filter value		
Turn OFF Time	I0, I1, I6, I7: 45 µs + filter value I2 to I5, I10 to I27: 150 µs + filter value		
Isolation	Between input terminals: Not isolated Internal circuit: Photocoupler isolated		
Input Type	Type 1 (IEC 61131)		
External Load for I/O Interconnection	Not needed		
Signal Determination Method	Static		
Effect of Improper Input Connection	Both sinking and sourcing input signals can be connected. If any input exceeding the rated value is applied, permanent damage may be caused.		
Cable Length	3m (9.84 ft.) in compliance with electromagnetic immunity		
Connector on Mother Board	FL26A2MA (Oki Electric Cable)	MC1.5/13-G-3.81BK (Phoenix Contact)	FL26A2MA (Oki Electric Cable)
Connector Insertion/Removal Durability	100 times minimum		

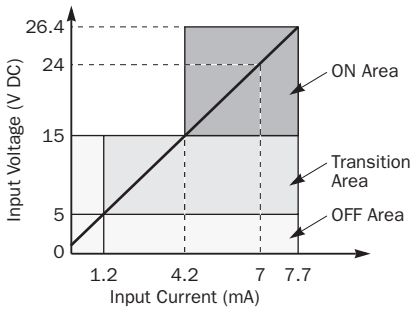
Input Operating Range

The input operating range of the Type 1 (IEC 61131-2) input module is shown below:

Inputs I0, I1, I6, and I7

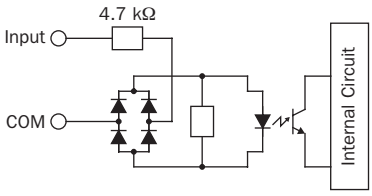


Inputs I2 to I5 and I10 to I27

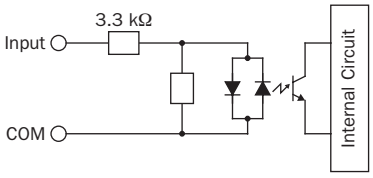


Input Internal Circuit

Inputs I0, I1, I6, and I7



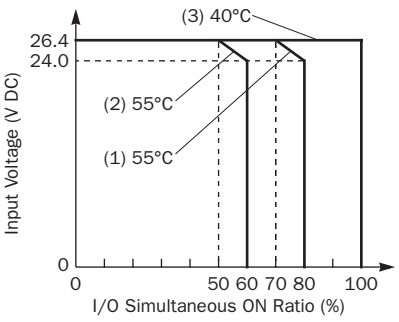
Inputs I2 to I5 and I10 to I27



I/O Usage Limits

When using the FC4A-D20K3/S3 at an ambient temperature of 55°C in the normal mounting direction, limit the inputs and outputs, respectively, which turn on simultaneously along line (1).

When using the FC4A-D40K3/S3, limit the inputs and outputs, respectively, which turn on simultaneously on each connector along line (2).



When using at 40°C, all I/Os can be turned on simultaneously at 26.4V DC as indicated with line (3).

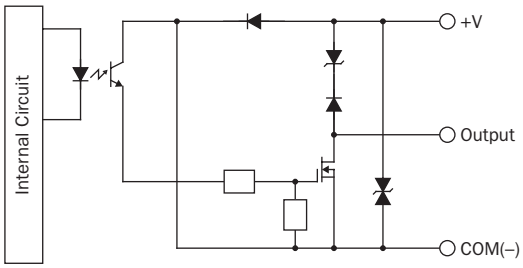
When using the FC4A-D20RK1/RS1, all I/Os can be turned on simultaneously at 55°C, input voltage 26.4V DC.

Transistor Sink and Source Output Specifications (Slim Type CPU Module)

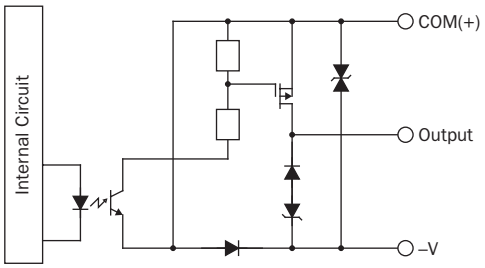
CPU Module		FC4A-D20K3 FC4A-D20RK1 FC4A-D40K3	FC4A-D20S3 FC4A-D20RS1 FC4A-D40S3
Output Type		Sink output	Source output
Output Points and Common Lines		FC4A-D20K3/S3: 8 points in 1 common line FC4A-D20RK1/RS1: 2 points in 1 common line FC4A-D40K3/S3: 16 points in 2 common lines	
Terminal Arrangement		See CPU Module Terminal Arrangement on pages 2-19 through 2-22.	
Rated Load Voltage		24V DC	
Operating Load Voltage Range		20.4 to 28.8V DC	
Rated Load Current		0.3A per output point	
Maximum Load Current		1A per common line	
Voltage Drop (ON Voltage)		1V maximum (voltage between COM and output terminals when output is on)	
Inrush Current		1A maximum	
Leakage Current		0.1 mA maximum	
Clamping Voltage		39V±1V	
Maximum Lamp Load		8W	
Inductive Load		L/R = 10 ms (28.8V DC, 1 Hz)	
External Current Draw		100 mA maximum, 24V DC (power voltage at the +V terminal)	100 mA maximum, 24V DC (power voltage at the –V terminal)
Isolation		Between output terminal and internal circuit: Photocoupler isolated Between output terminals: Not isolated	
Connector on Mother Board		FC4A-D20K3/S3: FL26A2MA (Oki Electric Cable) FC4A-D20RK1/RS1: MC1.5/16-G-3.81BK (Phoenix Contact) FC4A-D40K3/S3: FL26A2MA (Oki Electric Cable)	
Connector Insertion/Removal Durability		100 times minimum	
Output Delay	**Turn ON Time**	Q0, Q1: 5 µs maximum Q2 to Q17 300 µs maximum	
	Turn OFF Time	Q0, Q1: 5 µs maximum Q2 to Q17 300 µs maximum	

Output Internal Circuit

FC4A-D20K3, -D20RK1, and -D40K3 (Sink Output)



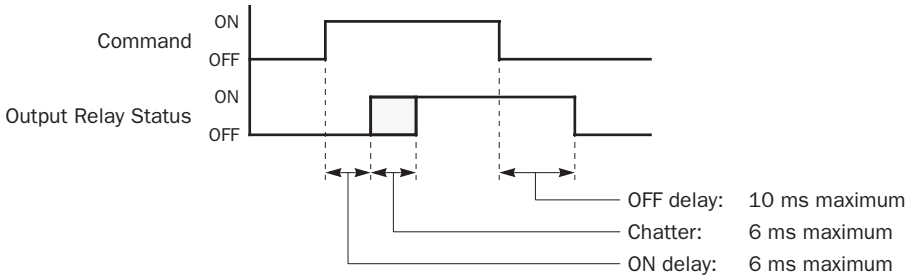
FC4A-D20S3, -D20RS1, and -D40S3 (Source Output)



Relay Output Specifications (Slim Type CPU Module)

CPU Module		FC4A-D20RK1	FC4A-D20RS1
No. of Outputs		8 points including 2 transistor output points	
Output Points per Common Line	**COM0**	(2 points transistor sink output)	(2 points transistor source output)
	COM1	3 NO contacts	
	COM2	2 NO contacts	
	COM3	1 NO contact	
Terminal Arrangement		See CPU Module Terminal Arrangement on page 2-20.	
Maximum Load Current		2A per point 8A per common line	
Minimum Switching Load		0.1 mA/0.1V DC (reference value)	
Initial Contact Resistance		30 mΩ maximum	
Electrical Life		100,000 operations minimum (rated load 1,800 operations/hour)	
Mechanical Life		20,000,000 operations minimum (no load 18,000 operations/hour)	
Rated Load (resistive/inductive)		240V AC/2A, 30V DC/2A	
Dielectric Strength		Between output and ⏚ terminals: 1,500V AC, 1 minute Between output terminal and internal circuit: 1,500V AC, 1 minute Between output terminals (COMs): 1,500V AC, 1 minute	
Connector on Mother Board		MC1.5/16-G-3.81BK (Phoenix Contact)	
Connector Insertion/Removal Durability		100 times minimum	
Contact Protection Circuit for Relay Output		See page 3-15.	

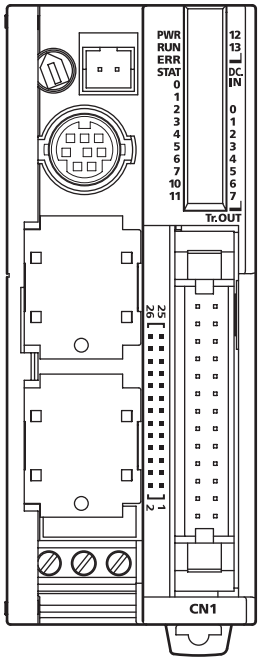
Output Delay



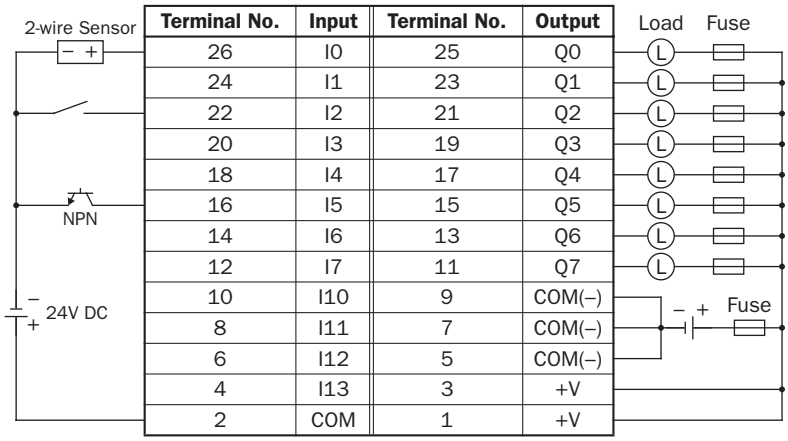
CPU Module Terminal Arrangement and I/O Wiring Diagrams (Slim Type)

FC4A-D20K3 (20-I/O Transistor Sink Output Type CPU Module)

Applicable Connector: FC4A-PMC26P (*not* supplied with the CPU module)



Source Input Wiring **Sink Output Wiring**

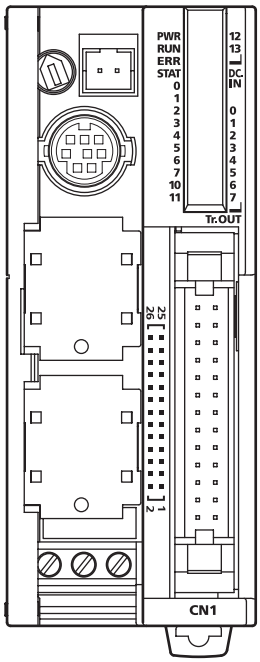


Terminal No.	Input	Terminal No.	Output
26	I0	25	Q0
24	I1	23	Q1
22	I2	21	Q2
20	I3	19	Q3
18	I4	17	Q4
16	I5	15	Q5
14	I6	13	Q6
12	I7	11	Q7
10	I10	9	COM(–)
8	I11	7	COM(–)
6	I12	5	COM(–)
4	I13	3	+V
2	COM	1	+V

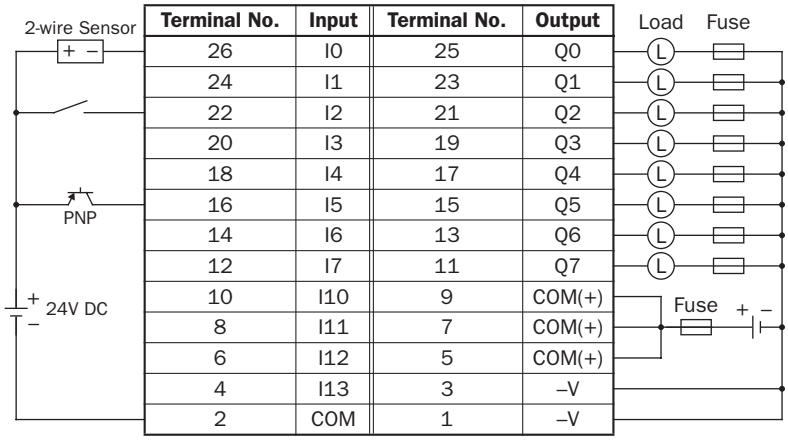
- COM(–) terminals are connected together internally.
- COM and COM(–) terminals are *not* connected together internally.
- +V terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 through 3-17.

FC4A-D20S3 (20-I/O Transistor Source Output Type CPU Module)

Applicable Connector: FC4A-PMC26P (*not* supplied with the CPU module)



Sink Input Wiring **Source Output Wiring**

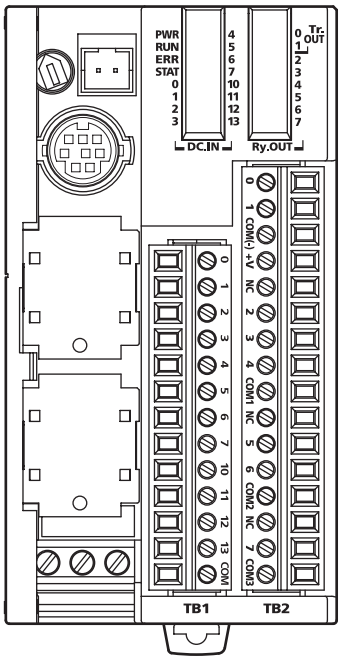


Terminal No.	Input	Terminal No.	Output
26	I0	25	Q0
24	I1	23	Q1
22	I2	21	Q2
20	I3	19	Q3
18	I4	17	Q4
16	I5	15	Q5
14	I6	13	Q6
12	I7	11	Q7
10	I10	9	COM(+)
8	I11	7	COM(+)
6	I12	5	COM(+)
4	I13	3	–V
2	COM	1	–V

- COM(+) terminals are connected together internally.
- COM and COM(+) terminals are *not* connected together internally.
- –V terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 through 3-17.

FC4A-D20RK1 (20-I/O Relay and Transistor Sink High-speed Output Type CPU Module)

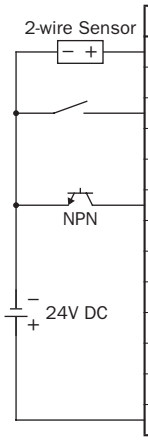
Applicable Terminal Blocks: **TB1 (Left Side)** **FC4A-PMT13P (supplied with the CPU module)**
TB2 (Right Side) **FC4A-PMTK16P (supplied with the CPU module)**



Source Input Wiring

TB1

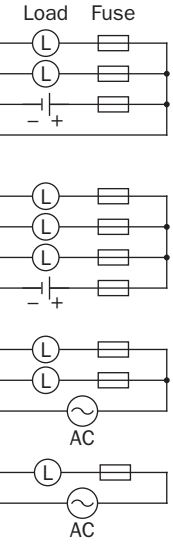
Terminal No.	Input
1	I0
2	I1
3	I2
4	I3
5	I4
6	I5
7	I6
8	I7
9	I10
10	I11
11	I12
12	I13
13	COM



Sink Output Wiring

TB2

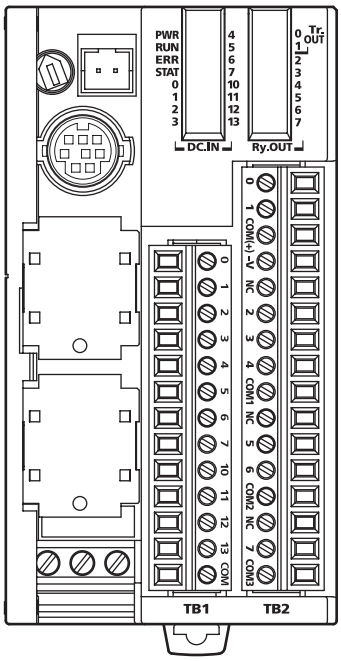
Terminal No.	Output
1	Q0
2	Q1
3	COM(–)
4	+V
5	NC
6	Q2
7	Q3
8	Q4
9	COM1
10	NC
11	Q5
12	Q6
13	COM2
14	NC
15	Q7
16	COM3



- Outputs Q0 and Q1 are transistor sink outputs; others are relay outputs.
- COM, COM(–), COM1, COM2, and COM3 terminals are *not* connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 through 3-17.

FC4A-D20RS1 (20-I/O Relay and Transistor Source High-speed Output Type CPU Module)

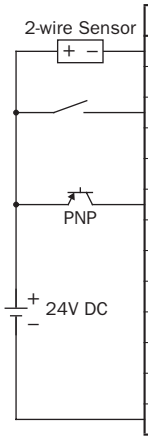
Applicable Terminal Blocks: **TB1 (Left Side)** **FC4A-PMT13P (supplied with the CPU module)**
TB2 (Right Side) **FC4A-PMTS16P (supplied with the CPU module)**



Sink Input Wiring

TB1

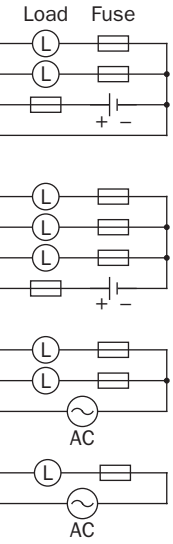
Terminal No.	Input
1	I0
2	I1
3	I2
4	I3
5	I4
6	I5
7	I6
8	I7
9	I10
10	I11
11	I12
12	I13
13	COM



Source Output Wiring

TB2

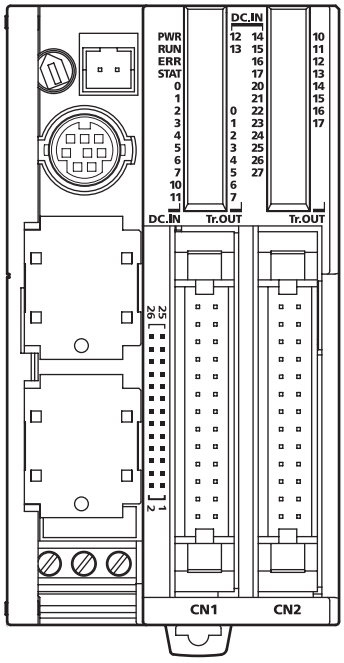
Terminal No.	Output
1	Q0
2	Q1
3	COM(+)
4	–V
5	NC
6	Q2
7	Q3
8	Q4
9	COM1
10	NC
11	Q5
12	Q6
13	COM2
14	NC
15	Q7
16	COM3



- Outputs Q0 and Q1 are transistor source outputs; others are relay outputs.
- COM, COM(+), COM1, COM2, and COM3 terminals are *not* connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 through 3-17.

FC4A-D40K3 (40-I/O Transistor Sink Output Type CPU Module)

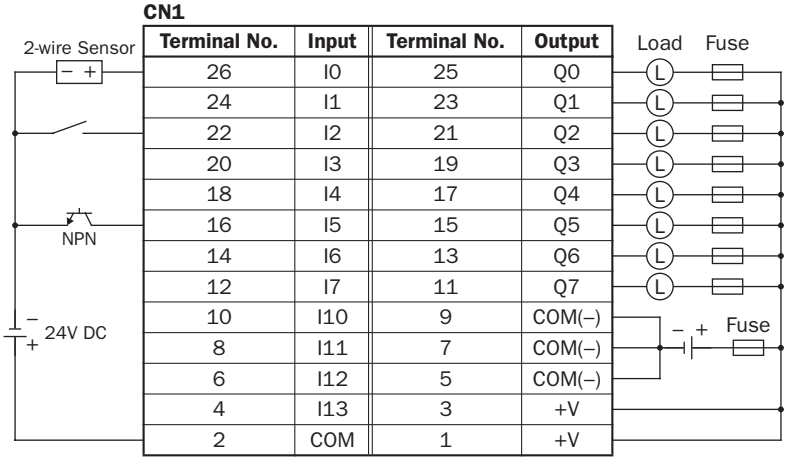
Applicable Connector: FC4A-PMC26P (*not* supplied with the CPU module)



Source Input Wiring **Sink Output Wiring**

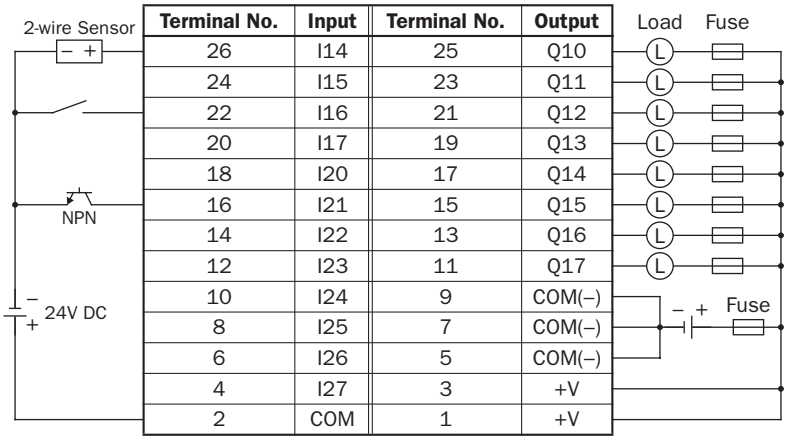
CN1

Terminal No.	Input	Terminal No.	Output
26	I0	25	Q0
24	I1	23	Q1
22	I2	21	Q2
20	I3	19	Q3
18	I4	17	Q4
16	I5	15	Q5
14	I6	13	Q6
12	I7	11	Q7
10	I10	9	COM(–)
8	I11	7	COM(–)
6	I12	5	COM(–)
4	I13	3	+V
2	COM	1	+V



CN2

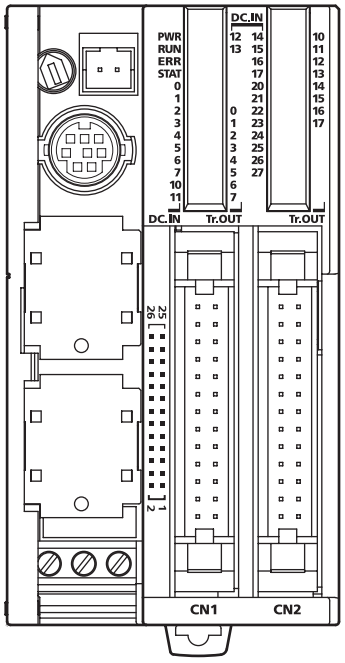
Terminal No.	Input	Terminal No.	Output
26	I14	25	Q10
24	I15	23	Q11
22	I16	21	Q12
20	I17	19	Q13
18	I20	17	Q14
16	I21	15	Q15
14	I22	13	Q16
12	I23	11	Q17
10	I24	9	COM(–)
8	I25	7	COM(–)
6	I26	5	COM(–)
4	I27	3	+V
2	COM	1	+V



- Terminals on CN1 and CN2 are *not* connected together internally.
- COM(–) terminals are connected together internally.
- COM and COM(–) terminals are *not* connected together internally.
- +V terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 through 3-17.

FC4A-D40S3 (40-I/O Transistor Source Output Type CPU Module)

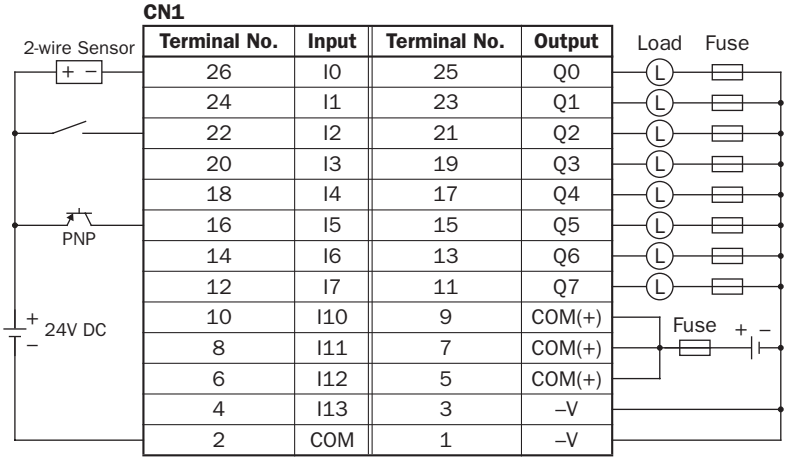
Applicable Connector: FC4A-PMC26P (*not* supplied with the CPU module)



Sink Input Wiring **Source Output Wiring**

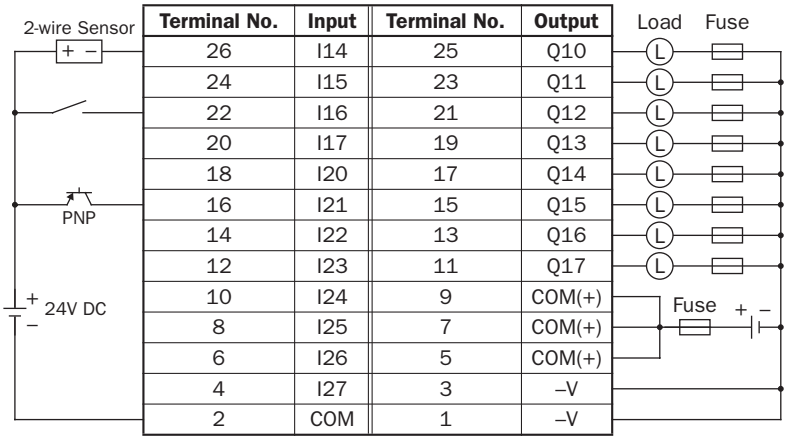
CN1

Terminal No.	Input	Terminal No.	Output
26	I0	25	Q0
24	I1	23	Q1
22	I2	21	Q2
20	I3	19	Q3
18	I4	17	Q4
16	I5	15	Q5
14	I6	13	Q6
12	I7	11	Q7
10	I10	9	COM(+)
8	I11	7	COM(+)
6	I12	5	COM(+)
4	I13	3	–V
2	COM	1	–V



CN2

Terminal No.	Input	Terminal No.	Output
26	I14	25	Q10
24	I15	23	Q11
22	I16	21	Q12
20	I17	19	Q13
18	I20	17	Q14
16	I21	15	Q15
14	I22	13	Q16
12	I23	11	Q17
10	I24	9	COM(+)
8	I25	7	COM(+)
6	I26	5	COM(+)
4	I27	3	–V
2	COM	1	–V



- Terminals on CN1 and CN2 are *not* connected together internally.
- COM(+) terminals are connected together internally.
- COM and COM(+) terminals are *not* connected together internally.
- –V terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 through 3-17.

Input Modules

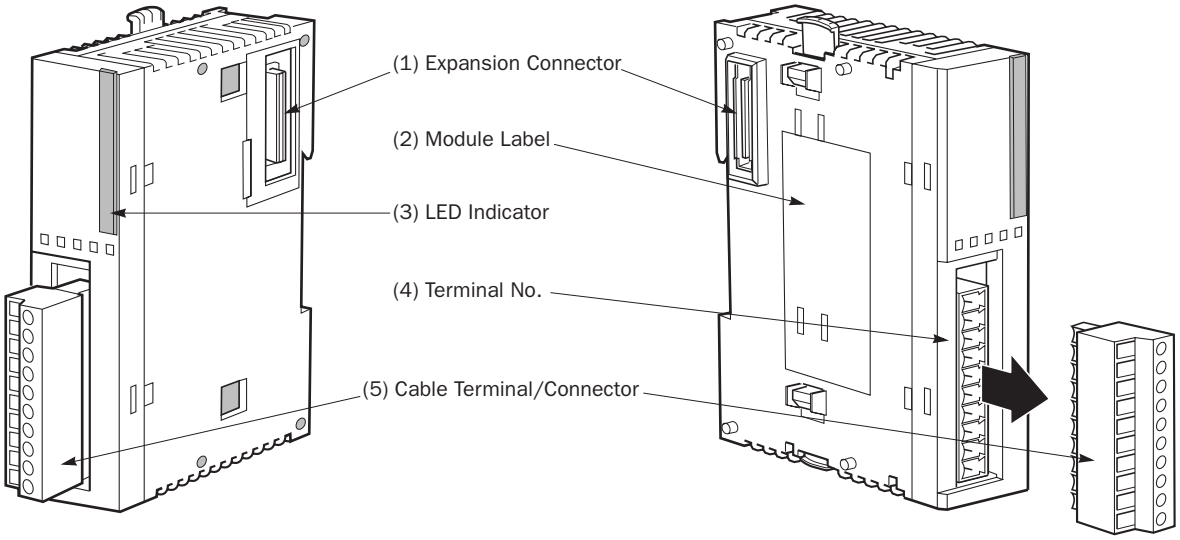
Digital input modules are available in 8-, 16-, and 32-point DC input modules and an 8-point AC input module with a screw terminal block or plug-in connector for input wiring. All DC input modules accept both sink and source DC input signals.

The input modules can be connected to the all-in-one 24-I/O type CPU module and all slim type CPU modules to expand input terminals. The all-in-one 10- and 16-I/O type CPU modules cannot connect input modules.

Input Module Type Numbers

Module Name	8-point DC Input	16-point DC Input	32-point DC Input	8-point AC Input
Screw Terminal	FC4A-N08B1	FC4A-N16B1	—	FC4A-N08A11
Connector	—	FC4A-N16B3	FC4A-N32B3	—

Parts Description



The above figures illustrate the 8-point DC input module.

(1) Expansion Connector	Connects to the CPU and other I/O modules. (The all-in-one 10- and 16-I/O type CPU modules cannot be connected.)
(2) Module Label	Indicates the input module Type No. and specifications.
(3) LED Indicator	Turns on when a corresponding input is on.
(4) Terminal No.	Indicates terminal numbers.
(5) Cable Terminal/Connector	Five different terminal/connector styles are available for wiring.

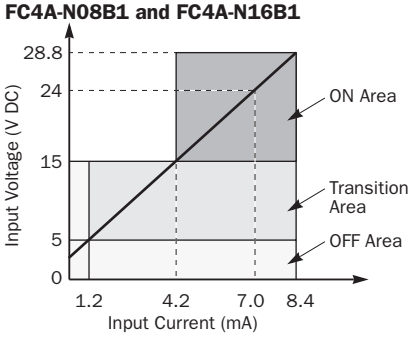
DC Input Module Specifications

Type No.		FC4A-N08B1	FC4A-N16B1	FC4A-N16B3	FC4A-N32B3
Input Points and Common Lines		8 points in 1 common line	16 points in 1 common line	16 points in 1 common line	32 points in 2 common lines
Terminal Arrangement		See Input Module Terminal Arrangement on pages 2-26 through 2-28.			
Rated Input Voltage		24V DC sink/source input signal			
Input Voltage Range		20.4 to 28.8V DC			
Rated Input Current		7 mA/point (24V DC)		5 mA/point (24V DC)	
Input Impedance		3.4 kΩ		4.4 kΩ	
Turn ON Time (24V DC)		4 ms			
Turn OFF Time (24V DC)		4 ms			
Isolation		Between input terminals: Not isolated Internal circuit: Photocoupler isolated			
External Load for I/O Interconnection		Not needed			
Signal Determination Method		Static			
Effect of Improper Input Connection		Both sinking and sourcing input signals can be connected. If any input exceeding the rated value is applied, permanent damage may be caused.			
Cable Length		3m (9.84 ft.) in compliance with electromagnetic immunity			
Connector on Mother Board		MC1.5/10-G-3.81BK (Phoenix Contact)		FL20A2MA (Oki Electric Cable)	
Connector Insertion/Removal Durability		100 times minimum			
Internal Current Draw	All Inputs ON	25 mA (5V DC) 0 mA (24V DC)	40 mA (5V DC) 0 mA (24V DC)	35 mA (5V DC) 0 mA (24V DC)	65 mA (5V DC) 0 mA (24V DC)
	All Inputs OFF	5 mA (5V DC) 0 mA (24V DC)	5 mA (5V DC) 0 mA (24V DC)	5 mA (5V DC) 0 mA (24V DC)	10 mA (5V DC) 0 mA (24V DC)
Weight		85g	100g	65g	100g

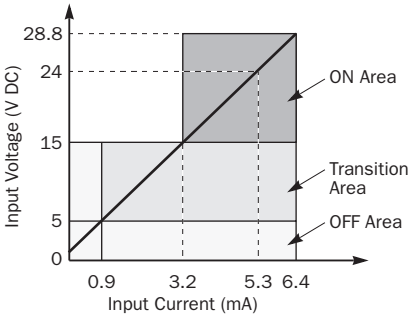
Input Operating Range

The input operating range of the Type 1 (IEC 61131-2) input module is shown below:

FC4A-N08B1 and FC4A-N16B1

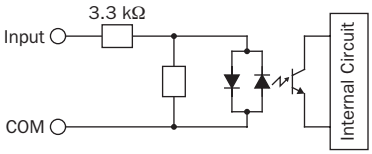


FC4A-N16B3 and FC4A-N32B3

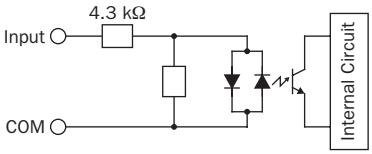


Input Internal Circuit

FC4A-N08B1 and FC4A-N16B1



FC4A-N16B3 and FC4A-N32B3

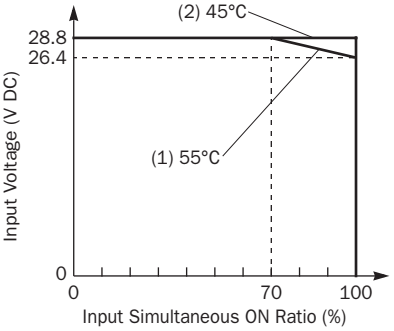


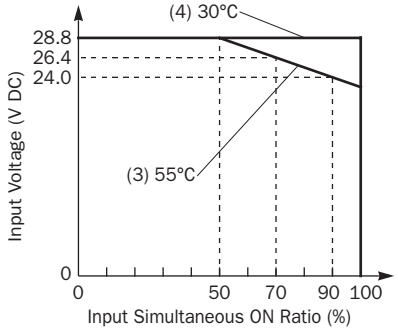
Input Usage Limits

When using the FC4A-N16B1 at 55°C in the normal mounting direction, limit the inputs which turn on simultaneously along line (1). At 45°C, all inputs can be turned on simultaneously at 28.8V DC as indicated with line (2).

When using the FC4A-N16B3 or -N32B3 at 55°C, limit the inputs which turn on simultaneously on each connector along line (3). At 30°C, all inputs can be turned on simultaneously at 28.8V DC as indicated with line (4).

When using the FC4A-N08B1, all inputs can be turned on simultaneously at 55°C, input voltage 28.8V DC.





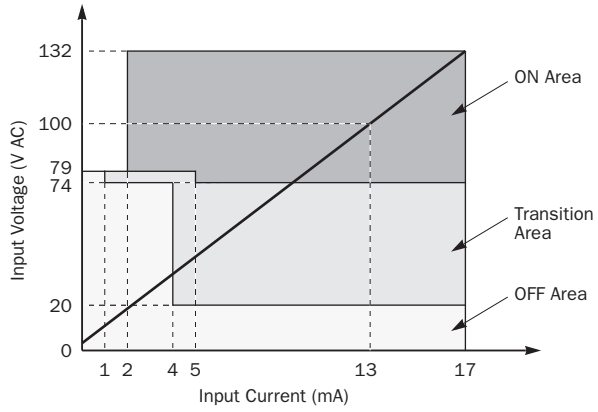
AC Input Module Specifications

Type No.		FC4A-N08A11
Input Points and Common Lines		8 points in 2 common lines
Terminal Arrangement		See Input Module Terminal Arrangement on page 2-29.
Rated Input Voltage		100 to 120V AC (50/60 Hz)
Input Voltage Range		85 to 132V AC
Rated Input Current		17 mA/point (120V AC, 60 Hz)
Input Type		AC input, Type 2 (IEC 61131-2)
Input Impedance		0.8 kΩ (60 Hz)
Turn ON Time		25 ms
Turn OFF Time		30 ms
Isolation		Between input terminals in the same common: Not isolated Between input terminals in different commons: Isolated Between input terminals and internal circuits: Photocoupler isolated
External Load for I/O Interconnection		Not needed
Signal Determination Method		Static
Effect of Improper Input Connection		If any input exceeding the rated value is applied, permanent damage may be caused.
Connector on Mother Board		MC1.5/11-G-3.81BK (Phoenix Contact)
Connector Insertion/Removal Durability		100 times minimum
Internal Current Draw	All Inputs ON	60 mA (5V DC) 0 mA (24V DC)
	All Inputs OFF	30 mA (5V DC) 0 mA (24V DC)
Weight		80g

Input Operating Range

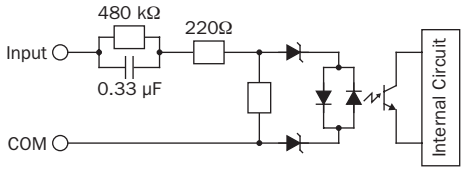
The input operating range of the Type 1, 2, 3 (IEC 61131-2) input module is shown below:

FC4A-N08A11



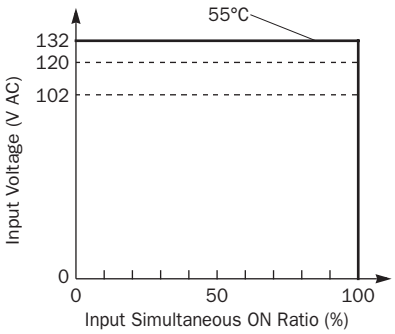
Input Internal Circuit

FC4A-N08A11



Input Usage Limits

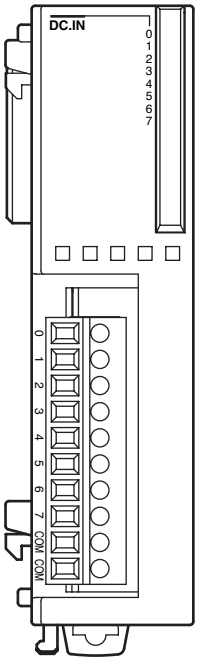
When using the FC4A-N08A11, all inputs can be turned on simultaneously at 55°C, input voltage 132V AC.



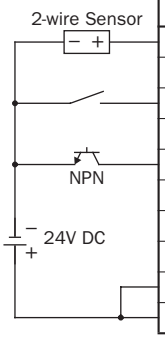
DC Input Module Terminal Arrangement and Wiring Diagrams

FC4A-N08B1 (8-point DC Input Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT10P (supplied with the input module)**

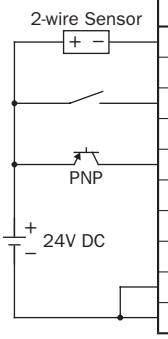


Source Input Wiring



Terminal No.	Input
0	I0
1	I1
2	I2
3	I3
4	I4
5	I5
6	I6
7	I7
COM	COM
COM	COM

Sink Input Wiring

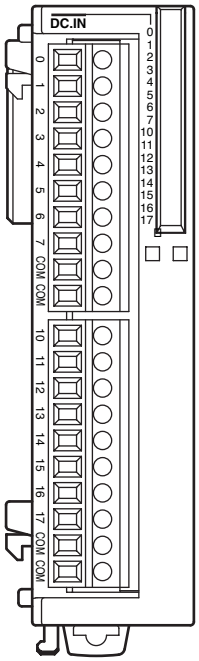


Terminal No.	Input
0	I0
1	I1
2	I2
3	I3
4	I4
5	I5
6	I6
7	I7
COM	COM
COM	COM

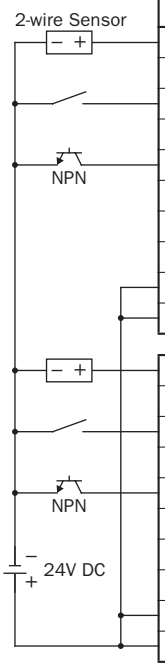
- Two COM terminals are connected together internally.
- For input wiring precautions, see page 3-13.

FC4A-N16B1 (16-point DC Input Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT10P (supplied with the input module)**

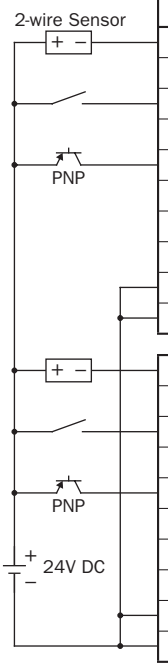


Source Input Wiring



Terminal No.	Input
0	I0
1	I1
2	I2
3	I3
4	I4
5	I5
6	I6
7	I7
COM	COM
COM	COM
10	I10
11	I11
12	I12
13	I13
14	I14
15	I15
16	I16
17	I17
COM	COM
COM	COM

Sink Input Wiring

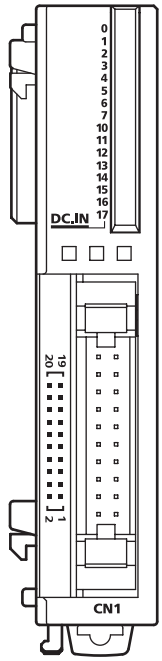


Terminal No.	Input
0	I0
1	I1
2	I2
3	I3
4	I4
5	I5
6	I6
7	I7
COM	COM
COM	COM
10	I10
11	I11
12	I12
13	I13
14	I14
15	I15
16	I16
17	I17
COM	COM
COM	COM

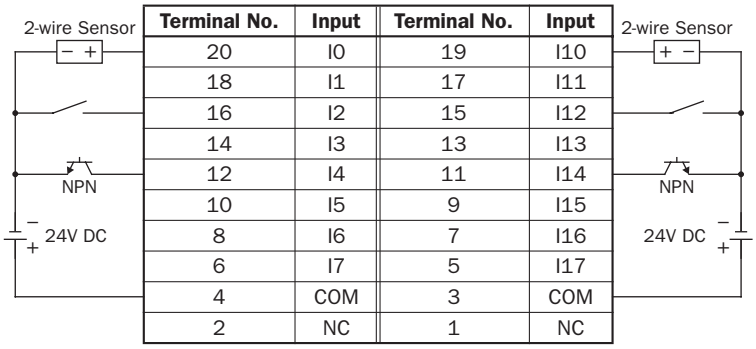
- Four COM terminals are connected together internally.
- For input wiring precautions, see page 3-13.

FC4A-N16B3 (16-point DC Input Module) — Connector Type

Applicable Connector: FC4A-PMC20P (*not* supplied with the input module)

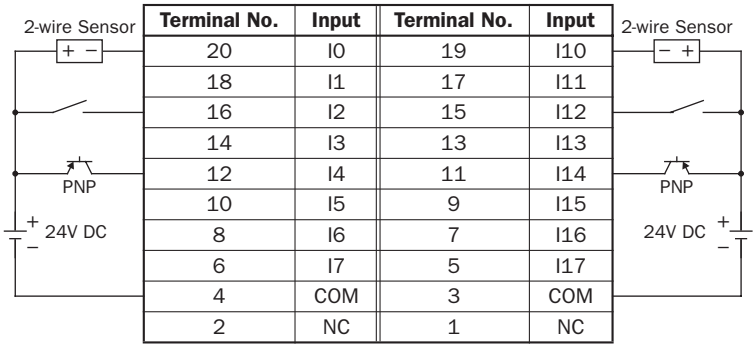


Source Input Wiring



Terminal No.	Input	Terminal No.	Input
20	I0	19	I10
18	I1	17	I11
16	I2	15	I12
14	I3	13	I13
12	I4	11	I14
10	I5	9	I15
8	I6	7	I16
6	I7	5	I17
4	COM	3	COM
2	NC	1	NC

Sink Input Wiring

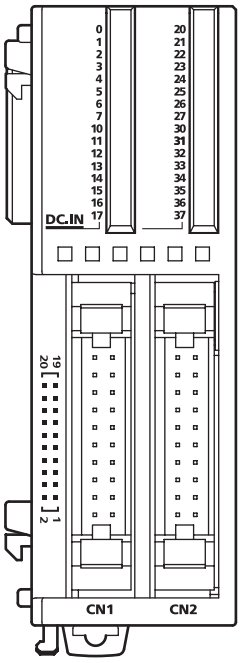


Terminal No.	Input	Terminal No.	Input
20	I0	19	I10
18	I1	17	I11
16	I2	15	I12
14	I3	13	I13
12	I4	11	I14
10	I5	9	I15
8	I6	7	I16
6	I7	5	I17
4	COM	3	COM
2	NC	1	NC

- Two COM terminals are connected together internally.
- For input wiring precautions, see page 3-13.

FC4A-N32B3 (32-point DC Input Module) — Connector Type

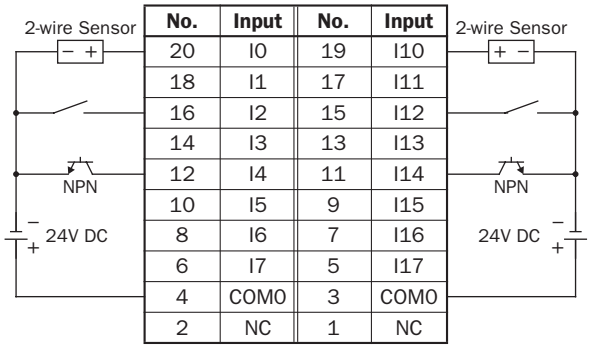
Applicable Connector: FC4A-PMC20P (*not* supplied with the input module)



- COM0 terminals are connected together internally.
- COM1 terminals are connected together internally.
- COM0 and COM1 terminals are *not* connected together internally.
- For input wiring precautions, see page 3-13.

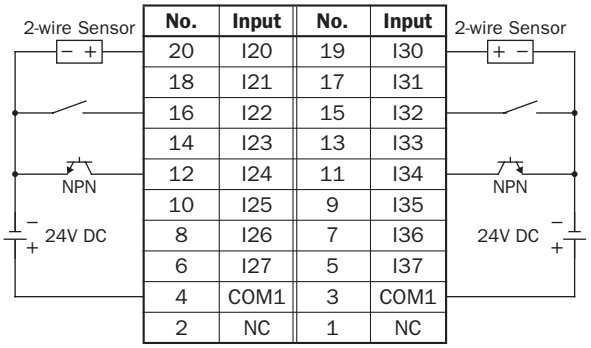
Source Input Wiring

CN1



No.	Input	No.	Input
20	I0	19	I10
18	I1	17	I11
16	I2	15	I12
14	I3	13	I13
12	I4	11	I14
10	I5	9	I15
8	I6	7	I16
6	I7	5	I17
4	COM0	3	COM0
2	NC	1	NC

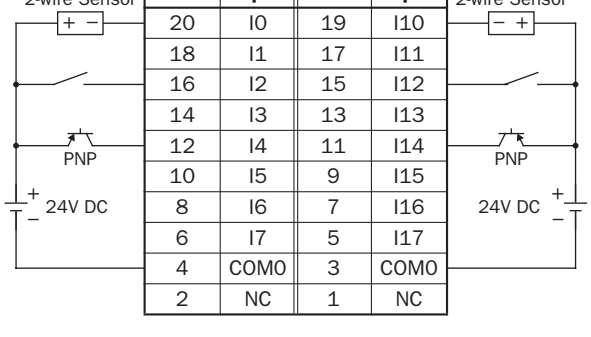
CN2



No.	Input	No.	Input
20	I20	19	I30
18	I21	17	I31
16	I22	15	I32
14	I23	13	I33
12	I24	11	I34
10	I25	9	I35
8	I26	7	I36
6	I27	5	I37
4	COM1	3	COM1
2	NC	1	NC

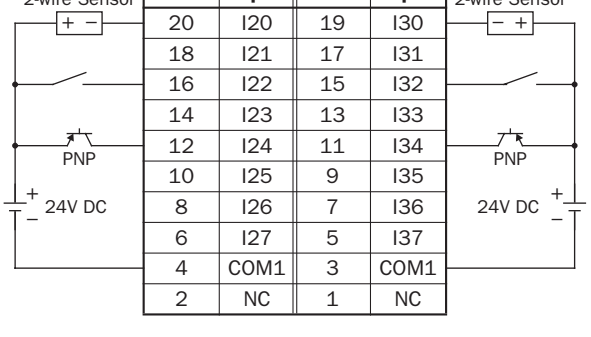
Sink Input Wiring

CN1



No.	Input	No.	Input
20	I0	19	I10
18	I1	17	I11
16	I2	15	I12
14	I3	13	I13
12	I4	11	I14
10	I5	9	I15
8	I6	7	I16
6	I7	5	I17
4	COM0	3	COM0
2	NC	1	NC

CN2

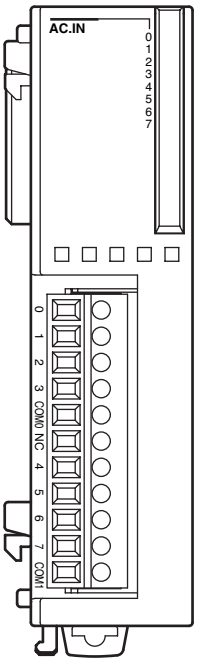


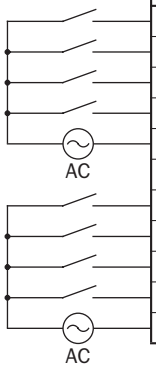
No.	Input	No.	Input
20	I20	19	I30
18	I21	17	I31
16	I22	15	I32
14	I23	13	I33
12	I24	11	I34
10	I25	9	I35
8	I26	7	I36
6	I27	5	I37
4	COM1	3	COM1
2	NC	1	NC

AC Input Module Terminal Arrangement and Wiring Diagrams

FC4A-N08A11 (8-point AC Input Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT11P (supplied with the input module)**





Terminal No.	Output
0	I0
1	I1
2	I2
3	I3
COM0	COM0
NC	NC
4	I4
5	I5
6	I6
7	I7
COM1	COM1

- Two COM terminals are *not* connected together internally.
- For input wiring precautions, see page 3-13.
- Do not connect an external load to the input terminals.

Output Modules

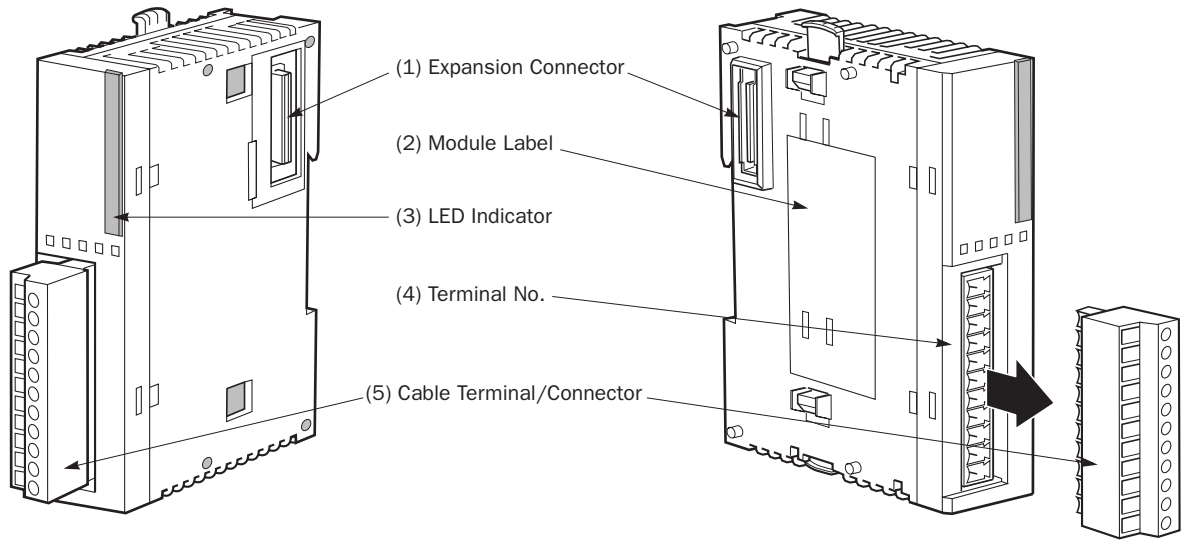
Digital output modules are available in 8- and 16-point relay output modules, 8-, 16- and 32-point transistor sink output modules, and 8-, 16- and 32-point transistor source output modules with a screw terminal block or plug-in connector for output wiring.

The output modules can be connected to the all-in-one 24-I/O type CPU module and all slim type CPU modules to expand output terminals. The all-in-one 10- and 16-I/O type CPU modules cannot connect output modules.

Output Module Type Numbers

Module Name	Terminal	Type No.
8-point Relay Output	Removable Terminal Block	FC4A-R081
16-point Relay Output		FC4A-R161
8-point Transistor Sink Output		FC4A-T08K1
8-point Transistor Source Output		FC4A-T08S1
16-point Transistor Sink Output	MIL Connector	FC4A-T16K3
16-point Transistor Source Output		FC4A-T16S3
32-point Transistor Sink Output		FC4A-T32K3
32-point Transistor Source Output		FC4A-T32S3

Parts Description



The above figures illustrate the 8-point relay output module.

(1) Expansion Connector	Connects to the CPU and other I/O modules. (The all-in-one 10- and 16-I/O type CPU modules cannot be connected.)
(2) Module Label	Indicates the output module Type No. and specifications.
(3) LED Indicator	Turns on when a corresponding output is on.
(4) Terminal No.	Indicates terminal numbers.
(5) Cable Terminal/Connector	Five different terminal/connector styles are available for wiring.

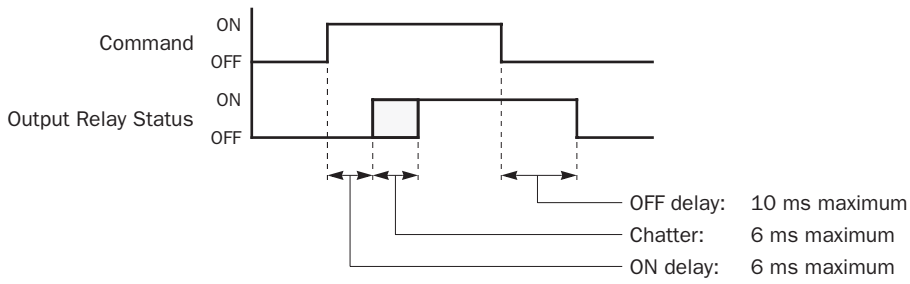
Relay Output Module Specifications

Type No.		FC4A-R081	FC4A-R161
Output Points and Common Lines		8 NO contacts in 2 common lines	16 NO contacts in 2 common lines
Terminal Arrangement		See Relay Output Module Terminal Arrangement on page 2-32.	
Maximum Load Current		2A per point	
		7A per common line	8A per common line
Minimum Switching Load		0.1 mA/0.1V DC (reference value)	
Initial Contact Resistance		30 mΩ maximum	
Electrical Life		100,000 operations minimum (rated load 1,800 operations/hour)	
Mechanical Life		20,000,000 operations minimum (no load 18,000 operations/hour)	
Rated Load (resistive/inductive)		240V AC/2A, 30V DC/2A	
Dielectric Strength		Between output and ⏚ or ⏥ terminals: 1,500V AC, 1 minute Between output terminal and internal circuit: 1,500V AC, 1 minute Between output terminals (COMs): 1,500V AC, 1 minute	
Connector on Mother Board		MC1.5/11-G-3.81BK (Phoenix Contact)	MC1.5/10-G-3.81BK (Phoenix Contact)
Connector Insertion/Removal Durability		100 times minimum	100 times minimum
Internal Current Draw	**All Outputs ON**	30 mA (5V DC) 40 mA (24V DC)	45 mA (5V DC) 75 mA (24V DC)
	All Outputs OFF	5 mA (5V DC) 0 mA (24V DC)	5 mA (5V DC) 0 mA (24V DC)
Weight		110g	145g
Contact Protection Circuit for Relay Output		See page 3-15.	

Note: When relay output modules are connected to the all-in-one 24-I/O type CPU module or any slim type CPU module, the maximum number of relay outputs that can be turned on simultaneously, including the outputs on the CPU module, are shown below.

CPU Module Type	All-in-One 24-I/O CPU Module	Slim Type CPU Module
Maximum Relay Outputs Turning On Simultaneously	33	54

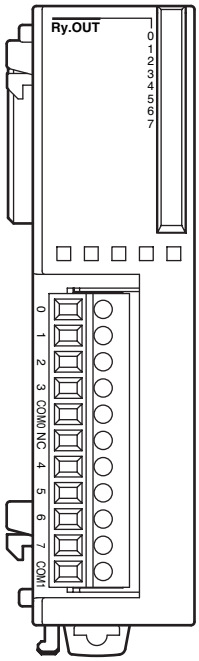
Output Delay

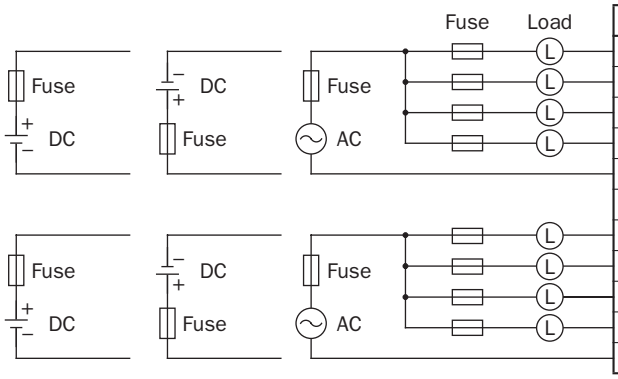


Relay Output Module Terminal Arrangement and Wiring Diagrams

FC4A-R081 (8-point Relay Output Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT11P (supplied with the output module)**



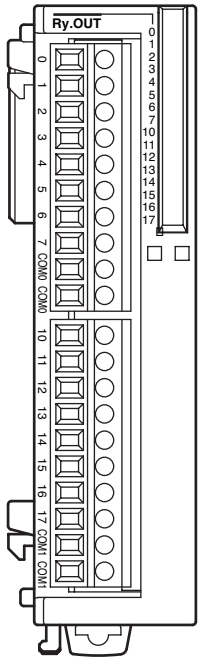


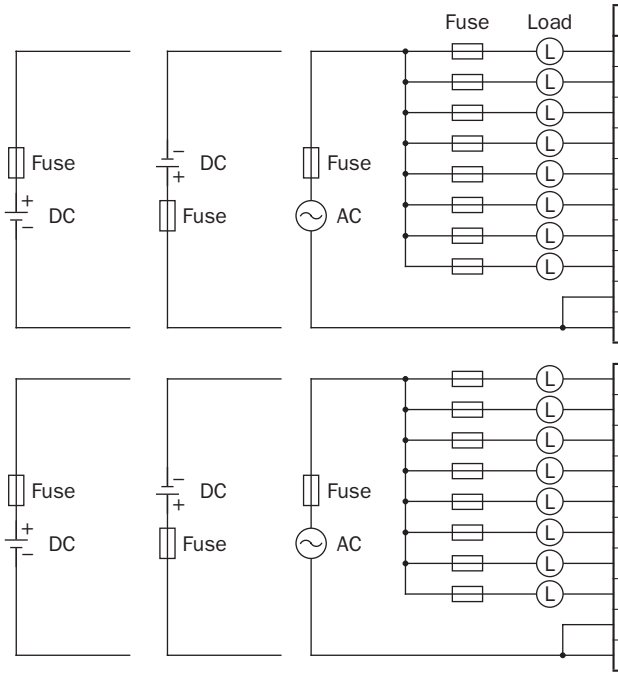
Terminal No.	Output
0	Q0
1	Q1
2	Q2
3	Q3
COM0	COM0
NC	NC
4	Q4
5	Q5
6	Q6
7	Q7
COM1	COM1

- COM0 and COM1 terminals are *not* connected together internally.
- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

FC4A-R161 (16-point Relay Output Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT10P (supplied with the output module)**





Terminal No.	Output
0	Q0
1	Q1
2	Q2
3	Q3
4	Q4
5	Q5
6	Q6
7	Q7
COM0	COM0
COM0	COM0
10	Q10
11	Q11
12	Q12
13	Q13
14	Q14
15	Q15
16	Q16
17	Q17
COM1	COM1
COM1	COM1

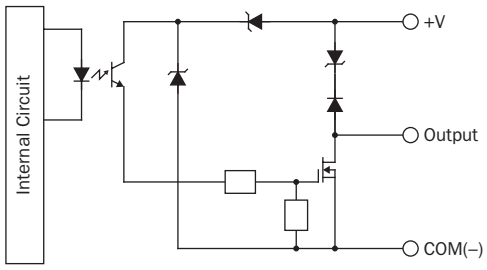
- COM0 terminals are connected together internally.
- COM1 terminals are connected together internally.
- COM0 and COM1 terminals are *not* connected together internally.
- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

Transistor Sink Output Module Specifications

Type No.		FC4A-T08K1	FC4A-T16K3	FC4A-T32K3
Output Type		Transistor sink output		
Output Points and Common Lines		8 points in 1 common line	16 points in 1 common line	32 points in 2 common lines
Terminal Arrangement		See Transistor Sink Output Module Terminal Arrangement on pages 2-34 and 2-35.		
Rated Load Voltage		24V DC		
Operating Load Voltage Range		20.4 to 28.8V DC		
Rated Load Current		0.3A per output point	0.1A per output point	
Maximum Load Current (at 28.8V DC)		0.3A per output point 3A per common line	0.1A per output point 1A per common line	
Voltage Drop (ON Voltage)		1V maximum (voltage between COM and output terminals when output is on)		
Inrush Current		1A maximum		
Leakage Current		0.1 mA maximum		
Clamping Voltage		39V±1V		
Maximum Lamp Load		8W		
Inductive Load		L/R = 10 ms (28.8V DC, 1 Hz)		
External Current Draw		100 mA maximum, 24V DC (power voltage at the +V terminal)		
Isolation		Between output terminal and internal circuit: Photocoupler isolated Between output terminals: Not isolated		
Connector on Mother Board		MC1.5/10-G-3.81BK (Phoenix Contact)	FL20A2MA (Oki Electric Cable)	
Connector Insertion/Removal Durability		100 times minimum		
Internal Current Draw	**All Outputs ON**	10 mA (5V DC) 20 mA (24V DC)	10 mA (5V DC) 40 mA (24V DC)	20 mA (5V DC) 70 mA (24V DC)
	All Outputs OFF	5 mA (5V DC) 0 mA (24V DC)	5 mA (5V DC) 0 mA (24V DC)	10 mA (5V DC) 0 mA (24V DC)
Output Delay		Turn ON time: 300 µs maximum Turn OFF time: 300 µs maximum		
Weight (approx.)		85g	70g	105g

Output Internal Circuit

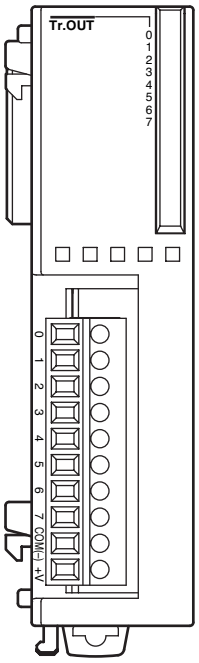
Sink Output

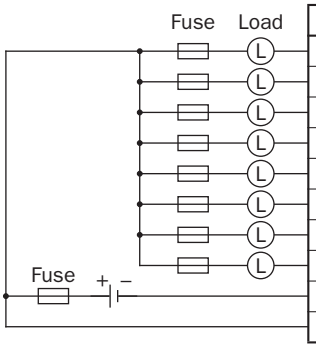


Transistor Sink Output Module Terminal Arrangement and Wiring Diagrams

FC4A-T08K1 (8-point Transistor Sink Output Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT10P (supplied with the output module)



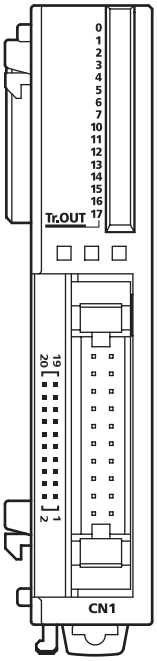


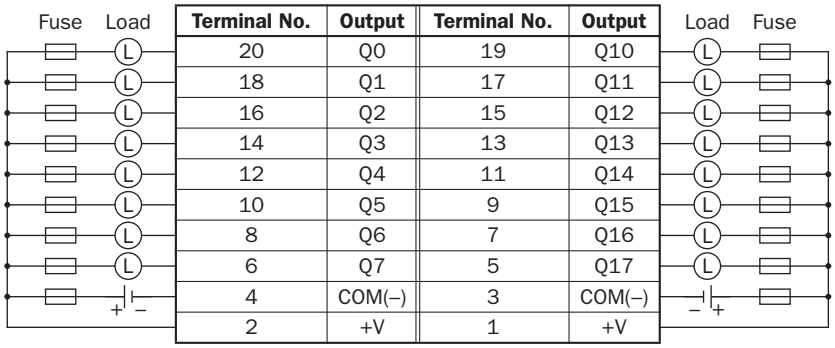
Terminal No.	Output
0	Q0
1	Q1
2	Q2
3	Q3
4	Q4
5	Q5
6	Q6
7	Q7
COM(–)	COM(–)
+V	+V

- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

FC4A-T16K3 (16-point Transistor Sink Output Module) — Connector Type

Applicable Connector: FC4A-PMC20P (*not* supplied with the output module)



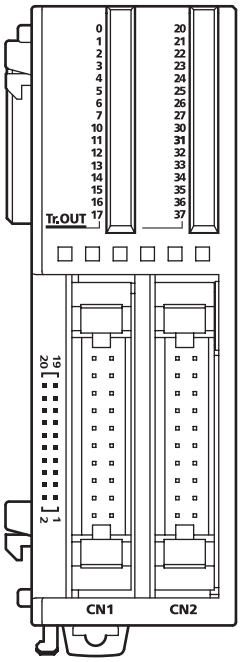


Terminal No.	Output	Terminal No.	Output
20	Q0	19	Q10
18	Q1	17	Q11
16	Q2	15	Q12
14	Q3	13	Q13
12	Q4	11	Q14
10	Q5	9	Q15
8	Q6	7	Q16
6	Q7	5	Q17
4	COM(–)	3	COM(–)
2	+V	1	+V

- COM(–) terminals are connected together internally.
- +V terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

FC4A-T32K3 (32-point Transistor Sink Output Module) — Connector Type

Applicable Connector: FC4A-PMC20P (*not* supplied with the output module)



CN1

Fuse	Load	Terminal No.	Output	Terminal No.	Output	Load	Fuse
	L	20	Q0	19	Q10	L	
	L	18	Q1	17	Q11	L	
	L	16	Q2	15	Q12	L	
	L	14	Q3	13	Q13	L	
	L	12	Q4	11	Q14	L	
	L	10	Q5	9	Q15	L	
	L	8	Q6	7	Q16	L	
	L	6	Q7	5	Q17	L	
+	–	4	COM0(–)	3	COM0(–)	–	+
		2	+V0	1	+V0		

CN2

Fuse	Load	Terminal No.	Output	Terminal No.	Output	Load	Fuse
	L	20	Q20	19	Q30	L	
	L	18	Q21	17	Q31	L	
	L	16	Q22	15	Q32	L	
	L	14	Q23	13	Q33	L	
	L	12	Q24	11	Q34	L	
	L	10	Q25	9	Q35	L	
	L	8	Q26	7	Q36	L	
	L	6	Q27	5	Q37	L	
+	–	4	COM1(–)	3	COM1(–)	–	+
		2	+V1	1	+V1		

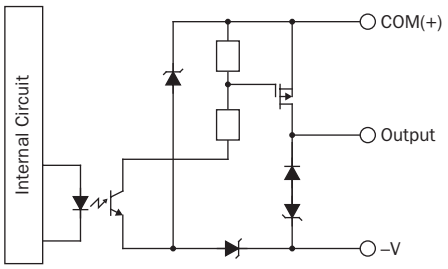
- Terminals on CN1 and CN2 are *not* connected together internally.
- COM0(–) terminals are connected together internally.
- COM1(–) terminals are connected together internally.
- +V0 terminals are connected together internally.
- +V1 terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

Transistor Source Output Module Specifications

Type No.		FC4A-T08S1	FC4A-T16S3	FC4A-T32S3
Output Type		Transistor source output		
Output Points and Common Lines		8 points in 1 common line	16 points in 1 common line	32 points in 2 common lines
Terminal Arrangement		See Transistor Source Output Module Terminal Arrangement on pages 2-37 and 2-38.		
Rated Load Voltage		24V DC		
Operating Load Voltage Range		20.4 to 28.8V DC		
Rated Load Current		0.3A per output point	0.1A per output point	
Maximum Load Current (at 28.8V DC)		0.3A per output point 3A per common line	0.1A per output point 1A per common line	
Voltage Drop (ON Voltage)		1V maximum (voltage between COM and output terminals when output is on)		
Inrush Current		1A maximum		
Leakage Current		0.1 mA maximum		
Clamping Voltage		39V±1V		
Maximum Lamp Load		8W		
Inductive Load		L/R = 10 ms (28.8V DC, 1 Hz)		
External Current Draw		100 mA maximum, 24V DC (power voltage at the –V terminal)		
Isolation		Between output terminal and internal circuit: Photocoupler isolated Between output terminals: Not isolated		
Connector on Mother Board		MC1.5/10-G-3.81BK (Phoenix Contact)	FL20A2MA (Oki Electric Cable)	
Connector Insertion/Removal Durability		100 times minimum		
Internal Current Draw	All Outputs ON	10 mA (5V DC) 20 mA (24V DC)	10 mA (5V DC) 40 mA (24V DC)	20 mA (5V DC) 70 mA (24V DC)
	All Outputs OFF	5 mA (5V DC) 0 mA (24V DC)	5 mA (5V DC) 0 mA (24V DC)	10 mA (5V DC) 0 mA (24V DC)
Output Delay		Turn ON time: 300 µs maximum Turn OFF time: 300 µs maximum		
Weight (approx.)		85g	70g	105g

Output Internal Circuit

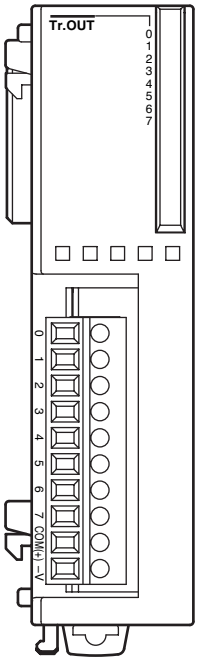
Source Output

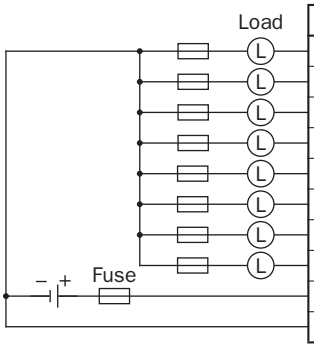


Transistor Source Output Module Terminal Arrangement and Wiring Diagrams

FC4A-T08S1 (8-point Transistor Source Output Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT10P (supplied with the output module)



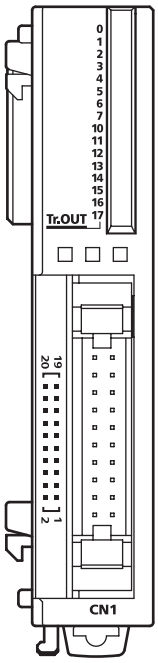


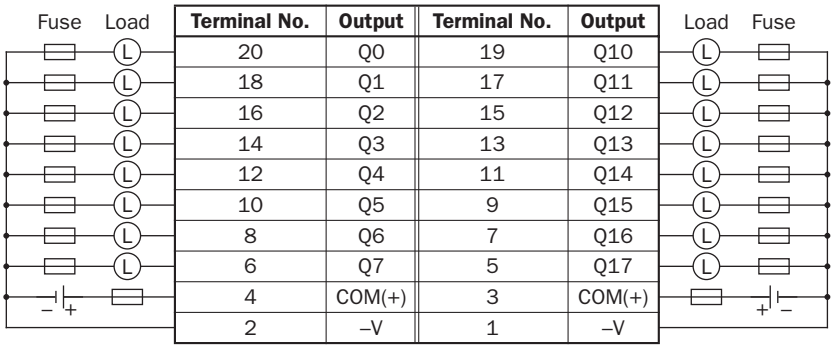
Terminal No.	Output
0	Q0
1	Q1
2	Q2
3	Q3
4	Q4
5	Q5
6	Q6
7	Q7
COM(+)	COM(+)
–V	–V

- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

FC4A-T16S3 (16-point Transistor Source Output Module) — Connector Type

Applicable Connector: FC4A-PMC20P (*not* supplied with the output module)



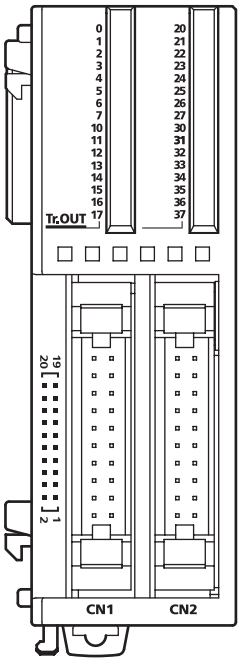


Terminal No.	Output	Terminal No.	Output
20	Q0	19	Q10
18	Q1	17	Q11
16	Q2	15	Q12
14	Q3	13	Q13
12	Q4	11	Q14
10	Q5	9	Q15
8	Q6	7	Q16
6	Q7	5	Q17
4	COM(+)	3	COM(+)
2	–V	1	–V

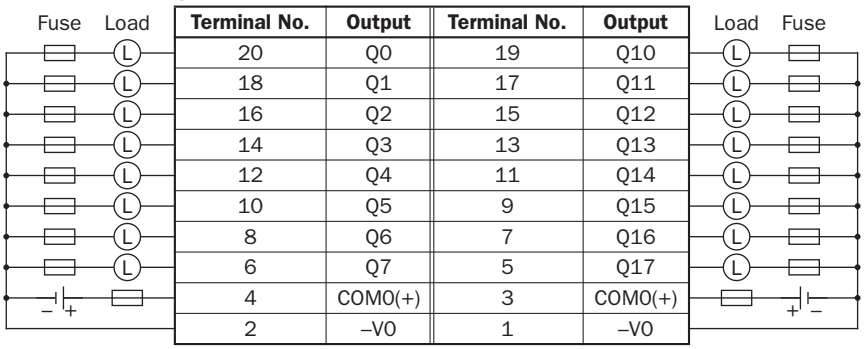
- COM(+) terminals are connected together internally.
- –V terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

FC4A-T32S3 (32-point Transistor Source Output Module) — Connector Type

Applicable Connector: FC4A-PMC20P (*not* supplied with the output module)

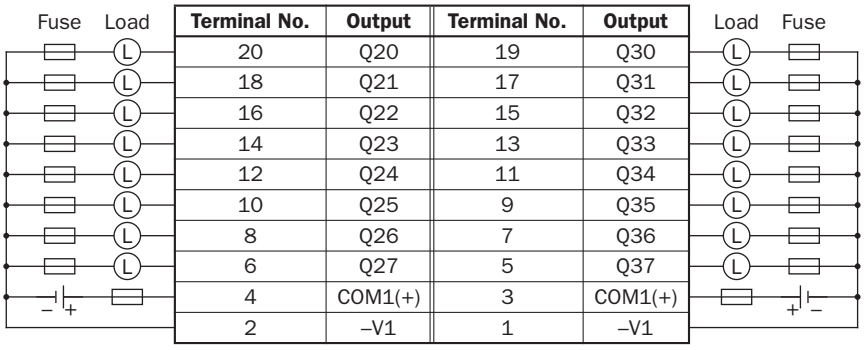


CN1



Terminal No.	Output	Terminal No.	Output
20	Q0	19	Q10
18	Q1	17	Q11
16	Q2	15	Q12
14	Q3	13	Q13
12	Q4	11	Q14
10	Q5	9	Q15
8	Q6	7	Q16
6	Q7	5	Q17
4	COM0(+)	3	COM0(+)
2	–V0	1	–V0

CN2



Terminal No.	Output	Terminal No.	Output
20	Q20	19	Q30
18	Q21	17	Q31
16	Q22	15	Q32
14	Q23	13	Q33
12	Q24	11	Q34
10	Q25	9	Q35
8	Q26	7	Q36
6	Q27	5	Q37
4	COM1(+)	3	COM1(+)
2	–V1	1	–V1

- Terminals on CN1 and CN2 are *not* connected together internally.
- COM0(+) terminals are connected together internally.
- COM1(+) terminals are connected together internally.
- –V0 terminals are connected together internally.
- –V1 terminals are connected together internally.
- Connect a fuse appropriate for the load.
- For output wiring precautions, see page 3-14.

Mixed I/O Modules

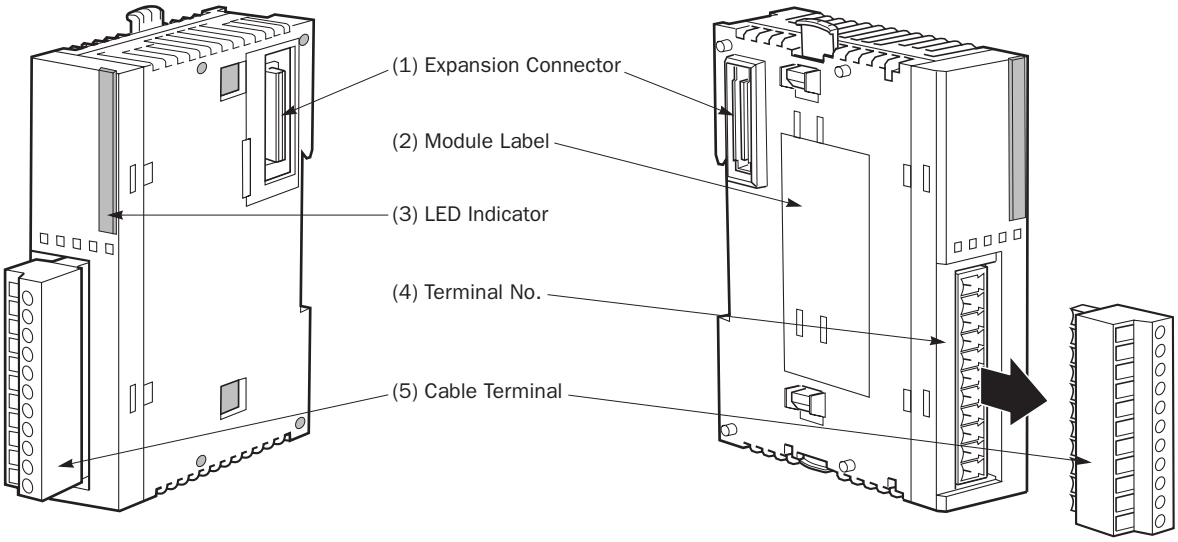
The 4-in/4-out mixed I/O module has 4-point DC sink/source inputs and 4-point relay outputs, with a screw terminal block for I/O wiring. The 16-in/8-out mixed I/O module has 16-point DC sink/source inputs and 8-point relay outputs, with a wire-clamp terminal block for I/O wiring.

The mixed I/O modules can be connected to the all-in-one 24-I/O type CPU module and all slim type CPU modules to expand input and output terminals. The all-in-one 10- and 16-I/O type CPU modules cannot connect mixed I/O modules.

Mixed I/O Module Type Numbers

Module Name	Terminal	Type No.
4-in/4-out Mixed I/O Module	Removable Terminal Block	FC4A-M08BR1
16-in/8-out Mixed I/O Module	Non-removable Wire-clamp Terminal Block	FC4A-M24BR2

Parts Description



The above figures illustrate the 4-in/4-out mixed I/O module.

(1) Expansion Connector	Connects to the CPU and other I/O modules. (The all-in-one 10- and 16-I/O type CPU modules cannot be connected.)
(2) Module Label	Indicates the mixed I/O module Type No. and specifications.
(3) LED Indicator	Turns on when a corresponding input or output is on.
(4) Terminal No.	Indicates terminal numbers.
(5) Cable Terminal	Two different terminal styles are available for wiring.

Mixed I/O Module Specifications

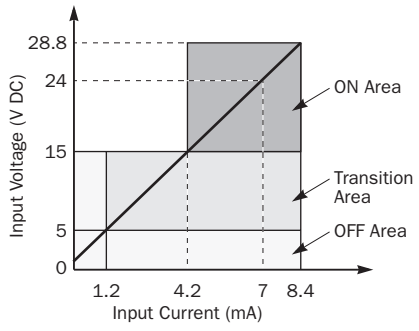
Type No.		FC4A-M08BR1	FC4A-M24BR2
I/O Points		4 inputs in 1 common line 4 outputs in 1 common line	16 inputs in 1 common line 8 outputs in 2 common lines
Terminal Arrangement		See Mixed I/O Module Terminal Arrangement on pages 2-41 and 2-42.	
Connector on Mother Board		MC1.5/11-G-3.81BK (Phoenix Contact)	Input: F6018-17P (Fujicon) Output: F6018-11P (Fujicon)
Connector Insertion/Removal Durability		100 times minimum	Not removable
Internal Current Draw	All I/Os ON	25 mA (5V DC) 20 mA (24V DC)	65 mA (5V DC) 45 mA (24V DC)
	All I/Os OFF	5 mA (5V DC) 0 mA (24V DC)	10 mA (5V DC) 0 mA (24V DC)
Weight		95g	140g

DC Input Specifications (Mixed I/O Module)

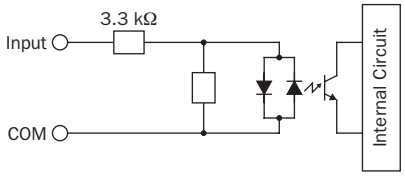
Input Points and Common Line	4 points in 1 common line	16 points in 1 common line
Rated Input Voltage	24V DC sink/source input signal	
Input Voltage Range	20.4 to 28.8V DC	
Rated Input Current	7 mA/point (24V DC)	
Input Impedance	3.4 kΩ	
Turn ON Time	4 ms (24V DC)	
Turn OFF Time	4 ms (24V DC)	
Isolation	Between input terminals: Not isolated Internal circuit: Photocoupler isolated	
External Load for I/O Interconnection	Not needed	
Signal Determination Method	Static	
Effect of Improper Input Connection	Both sinking and sourcing input signals can be connected. If any input exceeding the rated value is applied, permanent damage may be caused.	
Cable Length	3m (9.84 ft.) in compliance with electromagnetic immunity	

Input Operating Range

The input operating range of the Type 1 (IEC 61131-2) input module is shown below:

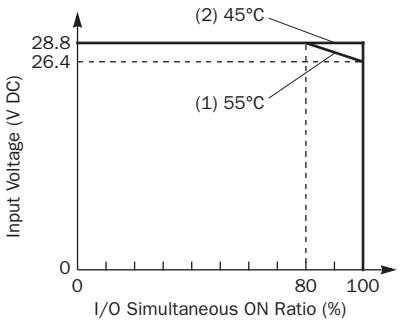


Input Internal Circuit



I/O Usage Limits

When using the FC4A-M24BR2 at an ambient temperature of 55°C in the normal mounting direction, limit the inputs and outputs, respectively, which turn on simultaneously along line (1).



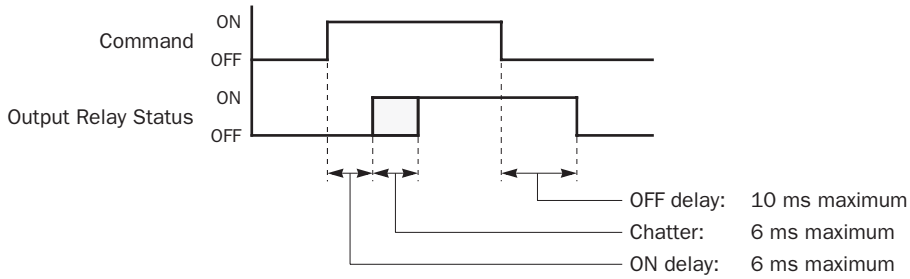
When using at 45°C, all I/Os can be turned on simultaneously at input voltage 28.8V DC as indicated with line (2).

When using the FC4A-M08BR1, all I/Os can be turned on simultaneously at 55°C, input voltage 28.8V DC.

Relay Output Specifications (Mixed I/O Module)

Type No.	FC4A-M08BR1	FC4A-M24BR2
Output Points and Common Lines	4 NO contacts in 1 common line	8 NO contacts in 2 common lines
Maximum Load Current	2A per point 7A per common line	
Minimum Switching Load	0.1 mA/0.1V DC (reference value)	
Initial Contact Resistance	30 mΩ maximum	
Electrical Life	100,000 operations minimum (rated load 1,800 operations/hour)	
Mechanical Life	20,000,000 operations minimum (no load 18,000 operations/hour)	
Rated Load (resistive/inductive)	240V AC/2A, 30V DC/2A	
Dielectric Strength	Between output and ⏚ or ⏚ terminals: 1,500V AC, 1 minute Between output terminal and internal circuit: 1,500V AC, 1 minute Between output terminals (COMs): 1,500V AC, 1 minute	
Contact Protection Circuit for Relay Output	See page 3-15.	

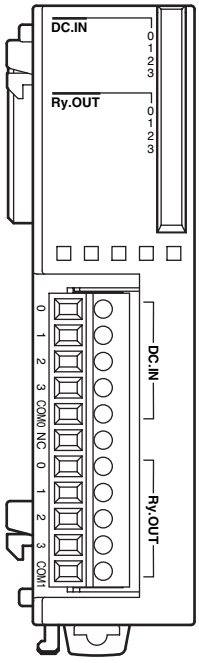
Output Delay

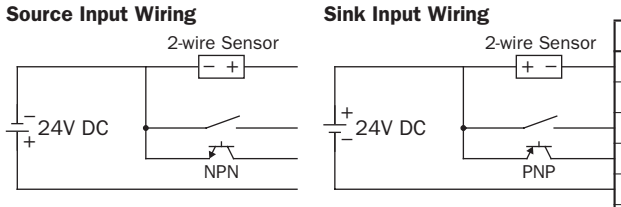


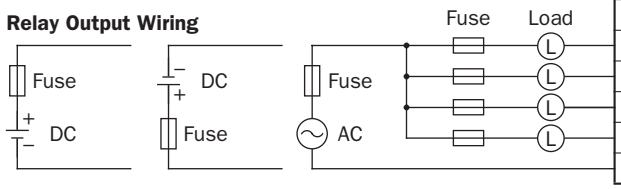
Mixed I/O Module Terminal Arrangement and Wiring Diagrams

FC4A-M08BR1 (Mixed I/O Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT11P (supplied with the mixed I/O module)



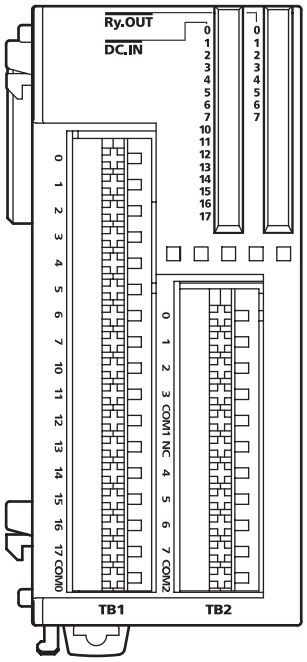




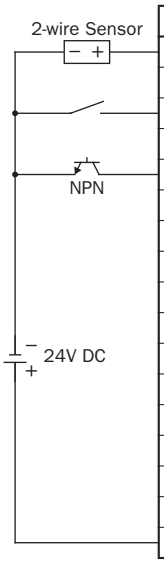
Terminal No.	I/O
0	I0
1	I1
2	I2
3	I3
COM0	COM0
NC	NC
0	Q0
1	Q1
2	Q2
3	Q3
COM1	COM1

- COM0 and COM1 terminals are *not* connected together internally.
- For wiring precautions, see pages 3-13 and 3-14.

FC4A-M24BR2 (Mixed I/O Module) — Wire-clamp Terminal Type

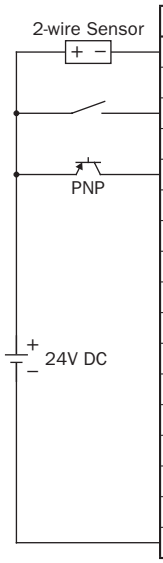


Source Input Wiring



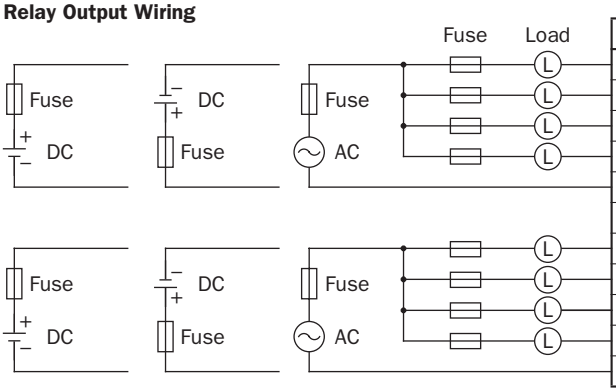
Terminal No.	Input
1	I0
2	I1
3	I2
4	I3
5	I4
6	I5
7	I6
8	I7
9	I10
10	I11
11	I12
12	I13
13	I14
14	I15
15	I16
16	I17
17	COM0

Sink Input Wiring



Terminal No.	Input
1	I0
2	I1
3	I2
4	I3
5	I4
6	I5
7	I6
8	I7
9	I10
10	I11
11	I12
12	I13
13	I14
14	I15
15	I16
16	I17
17	COM0

Relay Output Wiring



Terminal No.	Output
1	Q0
2	Q1
3	Q2
4	Q3
5	COM1
6	NC
7	Q4
8	Q5
9	Q6
10	Q7
11	COM2

- COM0, COM1, and COM2 terminals are *not* connected together internally.
- Connect a fuse appropriate for the load.
- For wiring precautions, see pages 3-13 and 3-14.

Analog I/O Modules

Analog I/O modules are available in 3-I/O types, 2-, 4-, and 8-input types, and 1- and 2-output types. The input channel can accept voltage and current signals, thermocouple and resistance thermometer signals, or thermistor signals. The output channel generates voltage and current signals.

Analog I/O Module Type Numbers

Name	I/O Signal	I/O Points	Category	Type No.
Analog I/O Module	Voltage (0 to 10V DC) Current (4 to 20mA)	2 inputs	END Refresh Type	FC4A-L03A1
	Voltage (0 to 10V DC) Current (4 to 20mA)	1 output		
	Thermocouple (K, J, T) Resistance thermometer (Pt100)	2 inputs		FC4A-L03AP1
	Voltage (0 to 10V DC) Current (4 to 20mA)	1 output		
Analog Input Module	Voltage (0 to 10V DC) Current (4 to 20mA)	2 inputs		FC4A-J2A1
	Voltage (0 to 10V DC) Current (4 to 20mA) Thermocouple (K, J, T) Resistance thermometer (Pt100, Pt1000, Ni100, Ni1000)	4 inputs	Ladder Refresh Type	FC4A-J4CN1
	Voltage (0 to 10V DC) Current (4 to 20mA)	8 inputs		FC4A-J8C1
	Thermistor (NTC, PTC)	8 inputs		FC4A-J8AT1
Analog Output Module	Voltage (0 to 10V DC) Current (4 to 20mA)	1 output	END Refresh Type	FC4A-K1A1
	Voltage (–10 to +10V DC) Current (4 to 20mA)	2 outputs	Ladder Refresh Type	FC4A-K2C1

END Refresh Type and Ladder Refresh Type

Depending on the internal circuit design for data refreshing, analog I/O modules are categorized into two types.

Analog I/O Module Category		END Refresh Type	Ladder Refresh Type
While CPU is running	Parameter Refreshing	At the end processing in the first scan	When executing ANST macro
	Analog I/O Data Refreshing	At the end processing	In the step after ANST macro (always refreshed whether input to ANST is on or off)
While CPU is stopped	Analog Output Data Refreshing	When M8025 (maintain outputs while CPU stopped) is on, output data is refreshed. When off, output is turned off.	Maintains output status when the CPU is stopped. Output data can be changed using STPA instruction while the CPU is stopped. See page 24-21.
Data Register Allocation		By default	Optionally designated in ANST macro

END Refresh Type

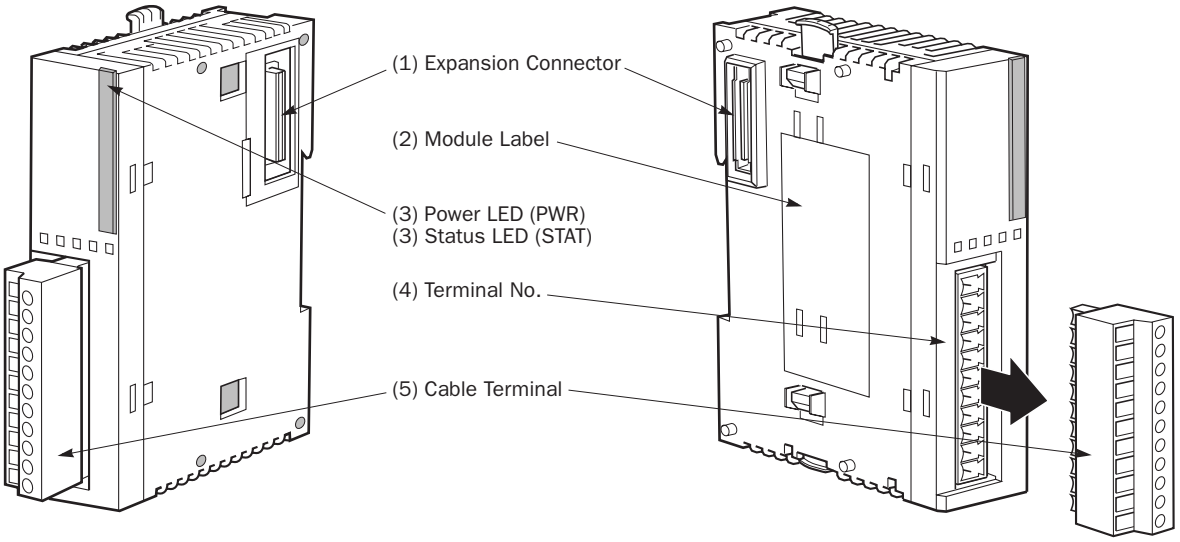
Each END refresh type analog I/O module is allocated 20 data registers to store analog I/O data and parameters for controlling analog I/O operation. These data registers are updated at every end processing while the CPU module is running. WindLDR has ANST macro to program the analog I/O modules.

The CPU module checks the analog I/O configuration only once at the end processing in the first scan. If you have changed the parameter while the CPU is running, stop and restart the CPU to enable the new parameter.

Ladder Refresh Type

Each ladder refresh type analog I/O module can be allocated any data registers to store analog I/O data and parameters for controlling analog I/O operation. The data registers are programmed in the ANST macro. Analog I/O data are updated at the ladder step following the ANST macro. Analog I/O parameters are updated when the ANST macro is executed, so analog I/O parameters can be changed while the CPU is running.

Parts Description

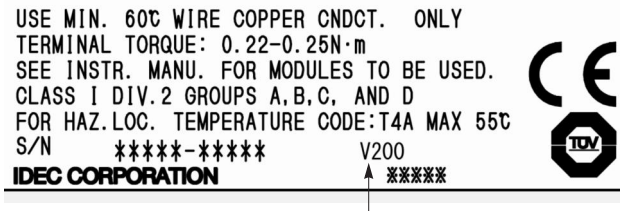


The terminal style depends on the model of analog I/O modules.

(1) Expansion Connector Connects to the CPU and other I/O modules.
(The all-in-one 10- and 16-I/O type CPU modules cannot be connected.)

(2) Module Label Indicates the analog I/O module Type No. and specifications.

Four analog I/O modules FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, and FC4A-K1A1 of version 200 or higher have the version number indicated on the module label attached to the side of the module. Confirm the version number because some specifications differ depending on the version number. Analog I/O modules earlier than version 200 do not have a version number indicated on the module label.



(3) Power LED (PWR) END refresh type FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, FC4A-K1A1:
Turns on when power is supplied to the analog I/O module.

(3) Status LED (STAT) Ladder refresh type FC4A-J4CN1, FC4A-J8C1, FC4A-J8AT1, FC4A-K2C1:
Indicates the operating status of the analog I/O module.

Status LED	Analog Input Operating Status
OFF	Analog I/O module is stopped
ON	Normal operation
Flash	Initializing Changing configuration Hardware initialization error External power supply error

(4) Terminal No. Indicates terminal numbers.

(5) Cable Terminal All analog I/O modules have a removable terminal block.

Analog I/O Module Specifications

General Specifications (END Refresh Type)

Type No.	FC4A-L03A1	FC4A-L03AP1	FC4A-J2A1	FC4A-K1A1
Rated Power Voltage	24V DC			
Allowable Voltage Range	20.4 to 28.8V DC			
Terminal Arrangement	See Analog I/O Module Terminal Arrangement on pages 2-52 to 2-55.			
Connector on Mother Board	MC1.5/11-G-3.81BK (Phoenix Contact)			
Connector Insertion/Removal Durability	100 times minimum			
Internal Current Draw	50 mA (5V DC) 0 mA (24V DC)	50 mA (5V DC) 0 mA (24V DC)	50 mA (5V DC) 0 mA (24V DC)	50 mA (5V DC) 0 mA (24V DC)
External Current Draw (Note 1)	50 (45) mA (Note 2) (24V DC)	50 (40) mA (Note 2) (24V DC)	40 (35) mA (Note 2) (24V DC)	40 mA (24V DC)
Weight (Approx.)	100g (85g) (Note 2)			

Note 1: The external current draw is the value when all analog inputs are used and the analog output value is at 100%.

Note 2: Values in ( ) represent analog I/O modules earlier than version 200. For analog I/O module version, see page 2-44.

General Specifications (Ladder Refresh Type)

Type No.	FC4A-J4CN1	FC4A-J8C1	FC4A-J8AT1	FC4A-K2C1
Rated Power Voltage	24V DC			
Allowable Voltage Range	20.4 to 28.8V DC			
Terminal Arrangement	See Analog I/O Module Terminal Arrangement on pages 2-52 to 2-55.			
Connector on Mother Board	MC1.5/10-G-3.81BK (Phoenix Contact)			
Connector Insertion/Removal Durability	100 times minimum			
Internal Current Draw	50 mA (5V DC) 0 mA (24V DC)	40 mA (5V DC) 0 mA (24V DC)	45 mA (5V DC) 0 mA (24V DC)	60 mA (5V DC) 0 mA (24V DC)
External Current Draw (Note)	55 mA (24V DC)	50 mA (24V DC)	55 mA (24V DC)	85 mA (24V DC)
Weight	140g	140g	125g	110g

Note: The external current draw is the value when all analog inputs are used and the analog output value is at 100%.

Analog Input Specifications (END Refresh Type)

Type No.		FC4A-L03A1 / FC4A-J2A1		FC4A-L03AP1	
Analog Input Signal Type		**Voltage Input**	**Current Input**	**Thermocouple**	**Resistance Thermometer**
Input Range		0 to 10V DC	4 to 20 mA DC	Type K (0 to 1300°C) Type J (0 to 1200°C) Type T (0 to 400°C)	Pt 100 3-wire type (–100 to 500°C)
Input Impedance		1 MΩ minimum	250Ω	1 MΩ minimum	1 MΩ minimum
Allowable Conductor Resistance (per wire)		—	—	—	200Ω maximum
Input Detection Current		—	—	—	1.0 mA maximum
AD Conversion	**Sample Duration Time**	10 (20) ms (Note 1)		10 (20) ms (Note 1)	20 ms
	Sample Repetition Time	20 ms		20 ms	40 (20) ms (Note 1)
	Total Input System Transfer Time (Note 2)	60 (105) ms + 1 scan time (Note 1)		60 (200) ms + 1 scan time (Note 1)	80 (200) ms + 1 scan time (Note 1)
	Type of Input	Single-ended input	Differential input		
	Operating Mode	Self-scan			
	Conversion Method	∑Δ type ADC			
Input Error	**Maximum Error at 25°C**	±0.2% of full scale		±0.2% of full scale + cold junction compensation error (±4°C maximum)	±0.2% of full scale
	Temperature Coefficient	±0.006% of full scale/°C			
	Repeatability after Stabilization Time	±0.5% of full scale			
	Non-lineality	±0.2% of full scale			
	Maximum Error	±1% of full scale			
Data	**Digital Resolution**	4096 increments (12 bits) 13,000 increments maximum (14 bits) (Note 3)			
	Input Value of LSB	2.5 mV	4 µA	K: 0.100°C/0.180°F (0.325°C) J: 0.100°C/0.180°F (0.300°C) T: 0.100°C/0.180°F (0.100°C) (Note 3)	0.100°C/0.180°F (0.150°C) (Note 3)
	Data Type in Application Program	Default: 0 to 4095 Optional: –32768 to 32767 (selectable for each channel) (Note 4)			
	Monotonicity	Yes			
	Input Data Out of Range	Detectable (Note 5)			

Type No.		FC4A-L03A1 / FC4A-J2A1		FC4A-L03AP1	
Analog Input Signal Type		Voltage Input	Current Input	Thermocouple	Resistance Thermometer
Noise Resistance	Maximum Temporary Deviation during Electrical Noise Tests	±1% maximum (when 1 kV is directly applied to the power supply line and a 1 kV clamp voltage is applied to I/O lines)			
		(±3% maximum) (Note 1) (when a 500V clamp voltage is applied to the power supply and I/O lines)			(Not assured) (Note 1)
	Input Filter	No			
	Recommended Cable for Noise Immunity	Twisted pair shielded cable		—	
	Crosstalk	2 LSB maximum			
Isolation		Between input and power circuit: Isolated Between input and internal circuit: Photocoupler-isolated			
Effect of Improper Input Connection		No damage			
Maximum Permanent Allowed Overload (No Damage)		13V DC	40 mA DC	—	
Selection of Analog Input Signal Type		Using programming software			
Calibration or Verification to Maintain Rated Accuracy		Not possible			

Note 1: Values in ( ) represent analog I/O modules earlier than version 200. For analog I/O module version, see page 2-44.

Note 2: Total input system transfer time = Sample repetition time + Internal processing time

Note 3: Minimum values represent analog input data in Celsius and Fahrenheit. Values in ( ) represent analog I/O modules earlier than version 200.

Note 4: The data processed in the analog I/O module can be linear-converted to a value between –32768 and 32767. The optional range designation, and analog I/O data minimum and maximum values can be selected using data registers allocated to analog I/O modules. See page 24-12.

Note 5: When an error is detected, a corresponding error code is stored to a data register allocated to analog I/O operating status. See page 24-6.

Analog Input Specifications (Ladder Refresh Type)

Type No.		FC4A-J4CN1 / FC4A-J8C1		FC4A-J4CN1	
Analog Input Signal Type		Voltage Input	Current Input	Thermocouple	Resistance Thermometer
Input Range		0 to 10V DC	4 to 20 mA DC	Type K: 0 to 1300°C Type J: 0 to 1200°C Type T: 0 to 400°C	Pt100, Pt1000: 3-wire type (–100 to 500°C) Ni100, Ni1000: 3-wire type (–60 to 180°C)
Input Impedance		1 MΩ	FC4A-J4CN1: 7Ω FC4A-J8C1: 100Ω	1 MΩ	—
Input Detection Current		—	—	—	0.1 mA
AD Conversion	Sample Duration Time	2 ms maximum			
	Sample Repetition Time	FC4A-J4CN1: 10 ms maximum FC4A-J8C1: 2 ms maximum		30 ms maximum	10 ms maximum
	Total Input System Transfer Time (Note 1)	FC4A-J4CN1: 50 ms × channels + 1 scan time FC4A-J8C1: 8 ms × channels + 1 scan time		85 ms × channels + 1 scan time	50 ms × channels + 1 scan time
	Type of Input	Single-ended input			
	Operating Mode	Self-scan			
	Conversion Method	FC4A-J4CN1: ∑Δ type ADC FC4A-J8C1: Successive approximation register method			
Input Error	Maximum Error at 25°C	±0.2% of full scale		±0.2% of full scale + cold junction compensation error (±3°C maximum)	Pt100, Ni100: ±0.4% of full scale Pt1000, Ni1000: ±0.2% of full scale
	Cold Junction Compensation Error	—	—	±3.0°C maximum	—
	Temperature Coefficient	±0.005% of full scale/°C			
	Repeatability after Stabilization Time	±0.5% of full scale			
	Non-lineality	±0.04% of full scale			
	Maximum Error	±1% of full scale			

Type No.		FC4A-J4CN1 / FC4A-J8C1		FC4A-J4CN1	
Analog Input Signal Type		Voltage Input	Current Input	Thermocouple	Resistance Thermometer
Data	Digital Resolution	50000 increments (16 bits)		K: Approx. 24000 increments (15 bits) J: Approx. 33000 increments (15 bits) T: Approx. 10000 increments (14 bits)	Pt100: Approx. 6400 increments (13 bits) Pt1000: Approx. 64000 increments (16 bits) Ni100: Approx. 4700 increments (13 bits) Ni1000: Approx. 47000 increments (16 bits)
	Input Value of LSB	0.2 mV	0.32 µA	K: 0.058°C J: 0.038°C T: 0.042°C	Pt100: 0.086°C Pt1000: 0.0086°C Ni100: 0.037°C Ni1000: 0.0037°C
	Data Type in Application Program	Default: 0 to 50000		Default: 0 to 50000	Pt100, Ni100: 0 to 6000 Pt1000, Ni1000: 0 to 60000
		Optional: –32768 to 32767 (selectable for each channel) (Note 2)			
		—		Temperature: Celsius, Fahrenheit	
	Monotonicity	Yes			
	Input Data Out of Range	Detectable (Note 3)			
Noise Resistance	Maximum Temporary Deviation during Electrical Noise Tests	±3% maximum (when a 500V clamp voltage is applied to the power supply and I/O lines)			Not assured
	Input Filter	Software			
	Recommended Cable for Noise Immunity	Twisted pair cable		—	
	Crosstalk	2 LSB maximum			
Isolation		Between input and power circuit: Isolated Between input and internal circuit: Photocoupler-isolated			
Effect of Improper Input Connection		No damage			
Maximum Permanent Allowed Overload (No Damage)		11V DC	22 mA DC	—	
Selection of Analog Input Signal Type		Using programming software			
Calibration or Verification to Maintain Rated Accuracy		Not possible			

Note 1: Total input system transfer time = Sample repetition time + Internal processing time
The total input system transfer time increases in proportion to the number of channels used.

Note 2: The data processed in the analog I/O module can be linear-converted to a value between –32768 and 32767. The optional range designation, and analog I/O data minimum and maximum values can be selected using data registers allocated to analog I/O modules. See page 24-12.

Note 3: When an error is detected, a corresponding error code is stored to a data register allocated to analog I/O operating status. See page 24-6.

Analog Input Specifications (Ladder Refresh Type)

Type No.		FC4A-J8AT1	
Analog Input Signal Type		NTC Thermistor	PTC Thermistor
Input Range		–50 to 150°C	
Applicable Thermistor		100 kΩ maximum	
Input Detection Current		0.1 mA	
AD Conversion	Sample Duration Time	2 ms maximum	
	Sample Repetition Time	2 ms maximum	
	Total Input System Transfer Time (Note 1)	10 ms × channels + 1 scan time (Note 1)	
	Type of Input	Single-ended input	
	Operating Mode	Self-scan	
	Conversion Method	Successive approximation register method	
Input Error	Maximum Error at 25°C	±0.2% of full scale	
	Temperature Coefficient	±0.005% of full scale/°C	
	Repeatability after Stabilization Time	±0.5% of full scale	
	Non-lineality	No	
	Maximum Error	±1% of full scale	
Data	Digital Resolution	Approx. 4000 increments (12 bits)	
	Input Value of LSB	0.05°C	
	Data Type in Application Program	Default: 0 to 4000 Optional: –32768 to 32767 (selectable for each channel) (Note 2) Temperature: Celsius, Fahrenheit (NTC only) Resistance: 0 to 10000	
	Monotonicity	Yes	
	Input Data Out of Range	Detectable (Note 3)	
Noise Resistance	Maximum Temporary Deviation during Electrical Noise Tests	±3% maximum (when a 500V clamp voltage is applied to the power supply and I/O lines)	
	Input Filter	Software	
	Recommended Cable for Noise Immunity	—	
	Crosstalk	2 LSB maximum	
Isolation		Between input and power circuit: Isolated Between input and internal circuit: Photocoupler-isolated	
Effect of Improper Input Connection		No damage	
Selection of Analog Input Signal Type		Using programming software	
Calibration or Verification to Maintain Rated Accuracy		Not possible	

Note 1: Total input system transfer time = Sample repetition time + Internal processing time
The total input system transfer time increases in proportion to the number of channels used.

Note 2: The data processed in the analog I/O module can be linear-converted to a value between –32768 and 32767. The optional range designation, and analog I/O data minimum and maximum values can be selected using data registers allocated to analog I/O modules. See page 24-12.

Note 3: When an error is detected, a corresponding error code is stored to a data register allocated to analog I/O operating status. See page 24-6.

Analog Output Specifications

Category			END Refresh Type			Ladder Refresh
Type No.			FC4A-L03A1	FC4A-L03AP1	FC4A-K1A1	FC4A-K2C1
Output Range		Voltage	0 to 10V DC			–10 to +10V DC
		Current	4 to 20 mA DC			
Load	Load Impedance		1 (2) kΩ minimum (voltage), 300Ω maximum (current) (Note 1)			
	Applicable Load Type		Resistive load			
DA Conversion	Settling Time		10 (50) ms (Note 1)	10 (130) ms (Note 1)	10 (50) ms (Note 1)	1 ms/ch
	Total Output System Transfer Time		Settling time + 1 scan time			1 ms × channels + 1 scan time
Output Error	Maximum Error at 25°C		±0.2% of full scale			
	Temperature Coefficient		±0.015% of full scale/°C			±0.005% of full scale/°C
	Repeatability after Stabilization Time		±0.5% of full scale			
	Output Voltage Drop		±1% of full scale			
	Non-lineality		±0.2% of full scale			
	Output Ripple		1 LSB maximum			±0.1% of full scale
	Overshoot		0%			
	Total Error		±1% of full scale			
Data	Digital Resolution		4096 increments (12 bits)			50000 increments (16 bits)
	Output Value of LSB	Voltage	2.5 mV			0.4 mV
		Current	4 µA			0.32 µA
	Data Type in Application Program		Default: 0 to 4095 (voltage, current)			–25000 to 25000 (voltage)
						0 to 50000 (current)
			Optional: –32768 to 32767 (selectable for each channel) (Note 2)			
	Monotonicity		Yes			
	Current Loop Open		Not detectable			
Noise Resistance	Maximum Temporary Deviation during Electrical Noise Tests (Note 3)		±1% (±3%) maximum (Note 1)			±3% maximum
	Recommended Cable for Noise Immunity		Twisted pair shielded cable			Twisted pair cable
	Crosstalk		No crosstalk because of 1 channel output			2 LSB maximum
Isolation			Between input and power circuit: Isolated Between input and internal circuit: Photocoupler-isolated			
Effect of Improper Output Connection			No damage			
Selection of Analog Output Signal Type			Using programming software			
Calibration or Verification to Maintain Rated Accuracy			Not possible			

Note 1: Values in ( ) represent analog I/O modules earlier than version 200. For analog I/O module version, see page 2-44.

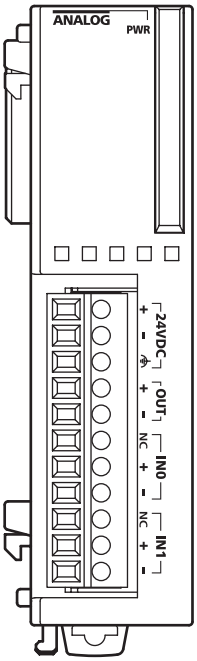
Note 2: The data processed in the analog I/O module can be linear-converted to a value between –32768 and 32767. The optional range designation, and analog I/O data minimum and maximum values can be selected using data registers allocated to analog I/O modules. See page 24-12.

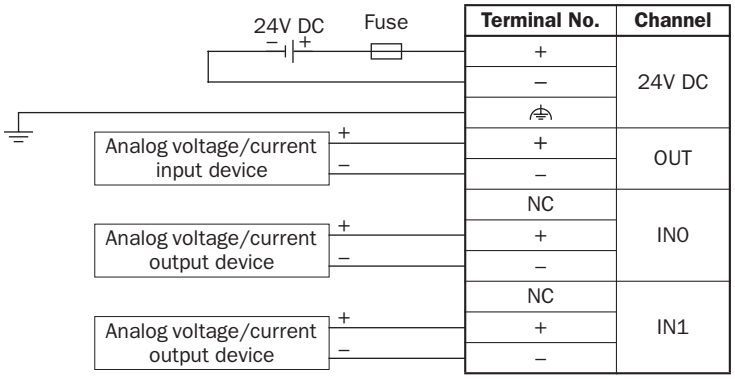
Note 3: For analog I/O modules of version 200 or higher, the value represents when 1 kV is directly applied to the power supply line and a 1 kV clamp voltage is applied to I/O lines. For analog I/O modules earlier than version 200, the value represents when a 500V clamp voltage is applied to the power supply and I/O lines.

Analog I/O Module Terminal Arrangement and Wiring Diagrams

FC4A-L03A1 (Analog I/O Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT11P (supplied with the analog I/O module)**



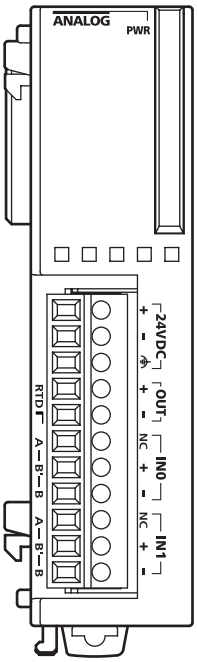


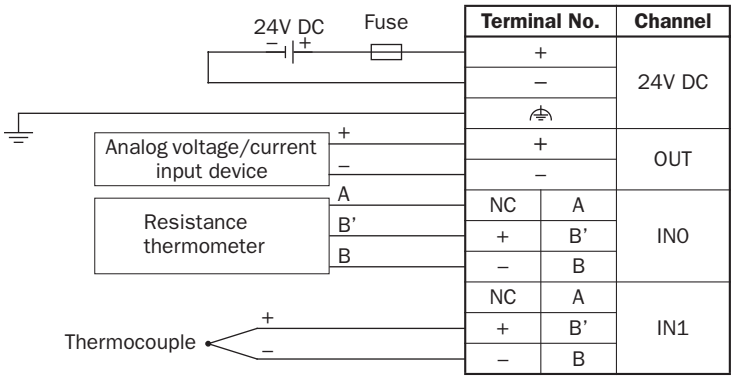
Terminal No.	Channel
+	24V DC
–	
⏚	
+	OUT
–	
NC	IN0
+	
–	
NC	IN1
+	
–	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not connect any wiring to unused terminals.
- Before turn on the power, make sure that wiring to the analog I/O module is correct. If wiring is incorrect, the analog I/O module may be damaged.

FC4A-L03AP1 (Analog I/O Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT11P (supplied with the analog I/O module)**



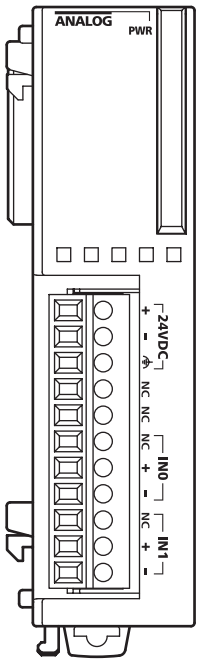


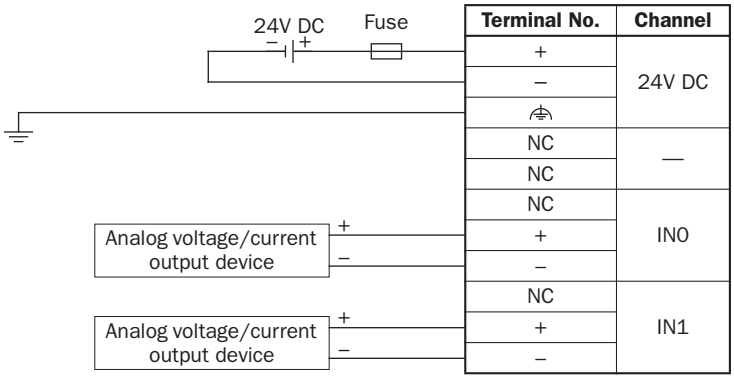
Terminal No.		Channel
+		24V DC
–		
⏚		
+		OUT
–		
NC	A	IN0
+	B'	
–	B	
NC	A	IN1
+	B'	
–	B	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- When connecting a resistance thermometer, connect the three wires to RTD (resistance temperature detector) terminals A, B', and B of input channel IN0 or IN1.
- When connecting a thermocouple, connect the two wires to terminals + and – of input channels IN0 or IN1.
- Do not connect any wiring to unused terminals.
- Do not connect the thermocouple to a hazardous voltage (60V DC or 42.4V peak or higher).

FC4A-J2A1 (Analog Input Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT11P (supplied with the analog input module)



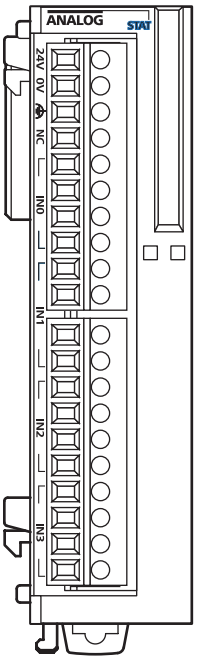


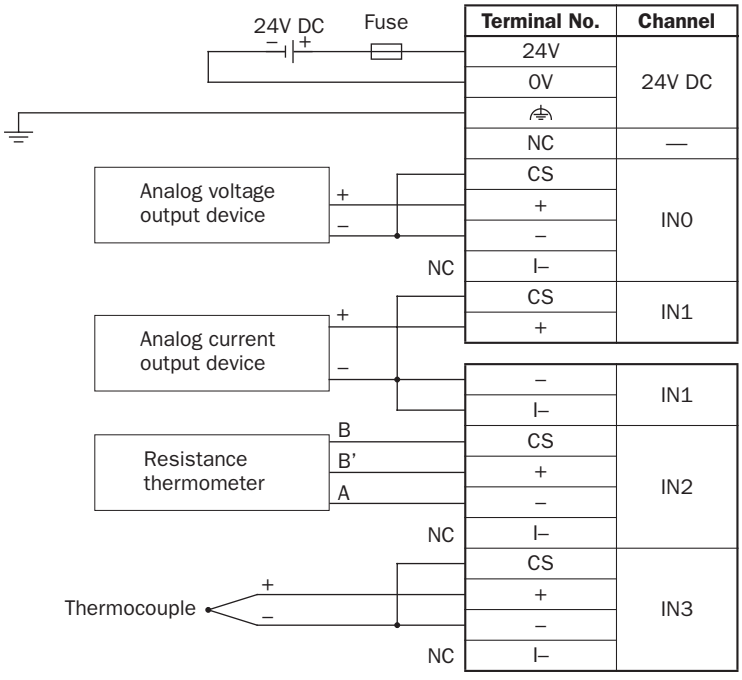
Terminal No.	Channel
+	24V DC
–	
⏚	
NC	—
NC	
NC	IN0
+	
–	
NC	IN1
+	
–	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not connect any wiring to unused terminals.

FC4A-J4CN1 (Analog Input Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT10P (supplied with the analog input module)



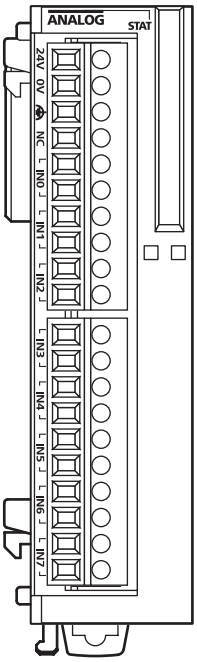


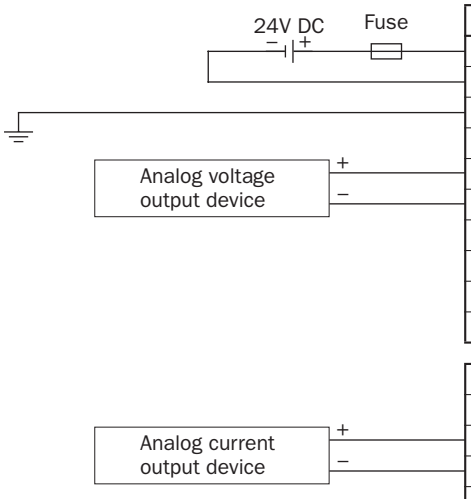
Terminal No.	Channel
24V	24V DC
0V	
⏚	
NC	—
CS	IN0
+	
–	
I–	
CS	IN1
+	
–	IN1
I–	
B	IN2
B'	
A	
–	
I–	
CS	IN3
+	
–	
I–	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- When connecting a resistance thermometer, connect three wires B, B', and A to the CS (current sense), +, and – terminals of input channels IN0 through IN3, respectively.
- When connecting a thermocouple, connect the + wire to the + terminal and the – wire to the CS and – terminals.
- Do not connect the thermocouple to a hazardous voltage (60V DC or 42.4V peak or higher).
- Do not connect any wiring to unused terminals.
- – terminals of input channels IN0 through IN3 are interconnected.

FC4A-J8C1 (Analog Input Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT10P (supplied with the analog input module)**



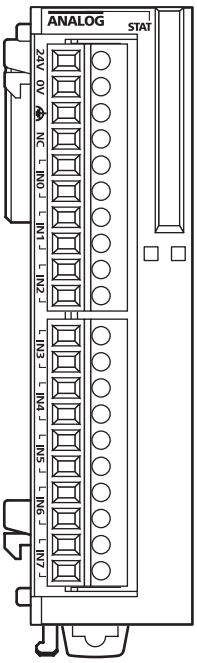


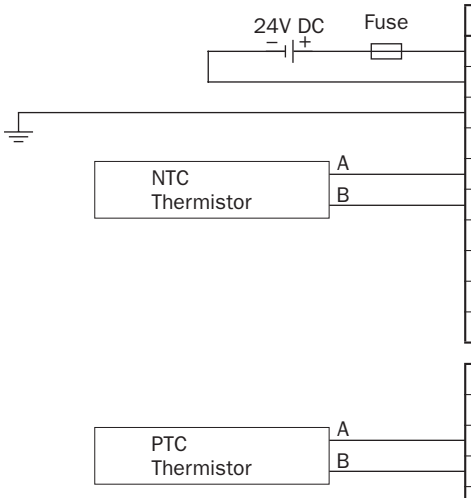
Terminal No.	Channel
24V	24V DC
0V	
⏚	
NC	—
+	IN0
–	
+	IN1
–	
+	IN2
–	
+	IN3
–	
+	IN4
–	
+	IN5
–	
+	IN6
–	
+	IN7
–	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not connect any wiring to unused terminals.
- – terminals of input channels IN0 through IN7 are interconnected.

FC4A-J8AT1 (Analog Input Module) — Screw Terminal Type

Applicable Terminal Block: **FC4A-PMT10P (supplied with the analog input module)**



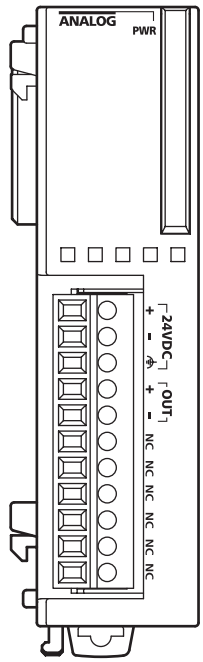


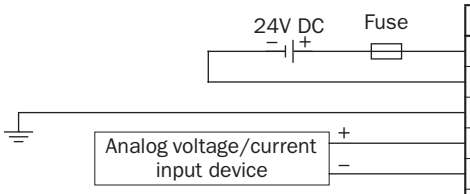
Terminal No.	Channel
24V	24V DC
0V	
⏚	
NC	—
A	IN0
B	
A	IN1
B	
A	IN2
B	
A	IN3
B	
A	IN4
B	
A	IN5
B	
A	IN6
B	
A	IN7
B	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not connect any wiring to unused terminals.

FC4A-K1A1 (Analog Output Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT11P (supplied with the analog output module)



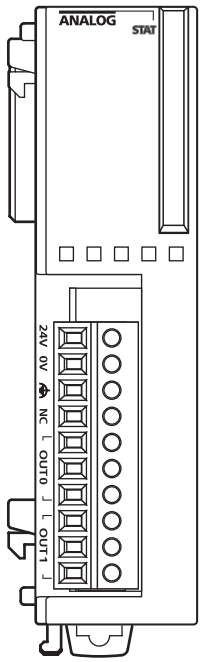


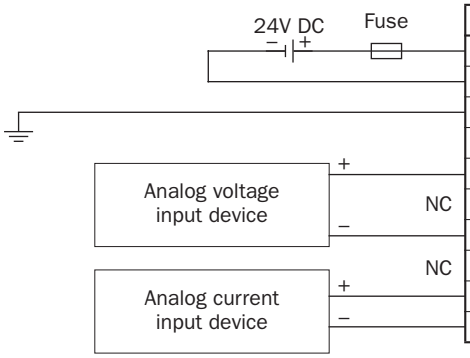
Terminal No.	Channel
+	24V DC
–	
⏚	
+	OUT
–	
NC	—
NC	
NC	
NC	—
NC	
NC	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not connect any wiring to unused terminals.

FC4A-K2C1 (Analog Output Module) — Screw Terminal Type

Applicable Terminal Block: FC4A-PMT10P (supplied with the analog output module)





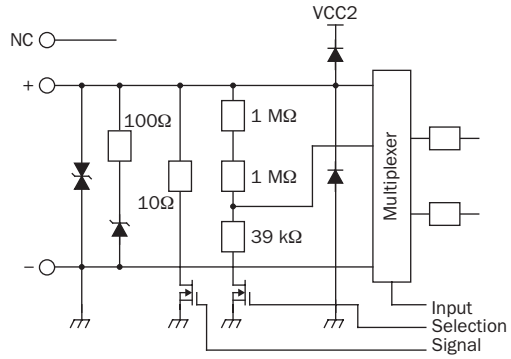
Terminal No.	Channel
24V	24V DC
0V	
⏚	
NC	—
V+	OUT0
I+	
–	
V+	OUT1
I+	
–	

- Connect a fuse appropriate for the applied voltage and current draw, at the position shown in the diagram. This is required when equipment containing the MicroSmart is destined for Europe.
- Do not connect any wiring to unused terminals.
- – terminals of output channels OUT0 and OUT1 are interconnected.

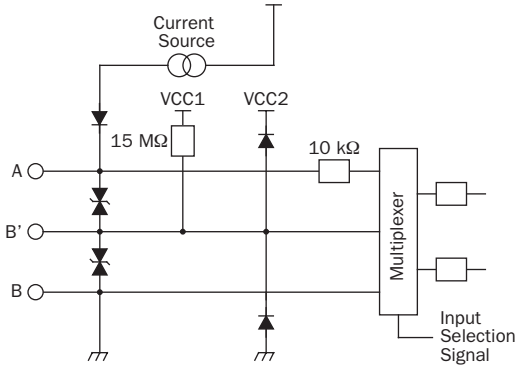
Type of Protection

Input Circuits

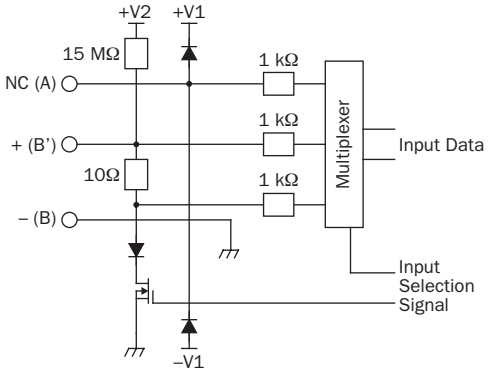
FC4A-L03A1, FC4A-J2A1 (Ver. 200 or higher)



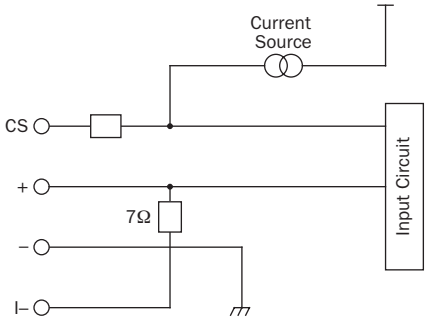
FC4A-L03AP1 (Ver. 200 or higher)



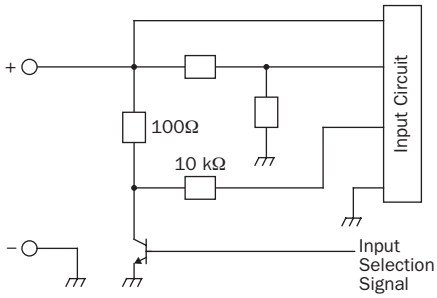
FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1



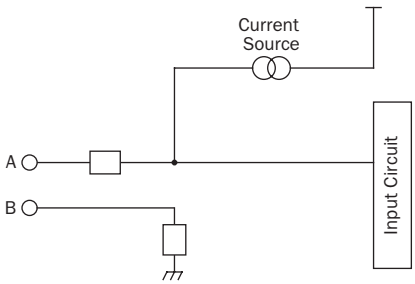
FC4A-J4CN1



FC4A-J8C1

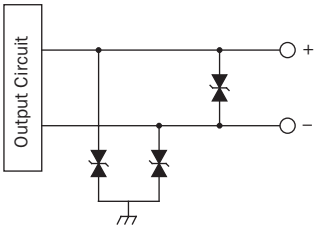


FC4A-J8AT1

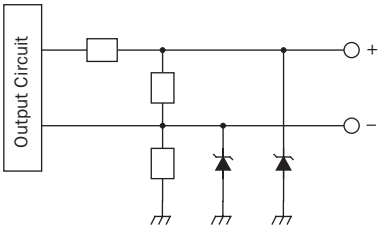


Output Circuits

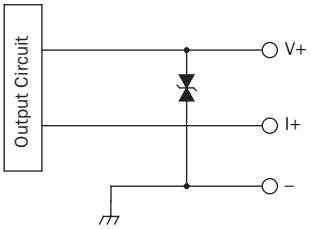
FC4A-L03A1, FC4A-L03AP1, FC4A-K1A1 (Ver. 200 or higher)



FC4A-L03A1, FC4A-L03AP1, FC4A-K1A1



FC4A-K2C1



Power Supply for Analog I/O Modules

When supplying power to the analog I/O modules, take the following considerations.

• Power Supply for FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, and FC4A-K1A1

Use separate power supplies for the MicroSmart CPU module and FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, and FC4A-K1A1. Power up the analog I/O modules at least 1 second earlier than the CPU module. This is recommended to ensure correct operation of the analog I/O control.

Note: When re-powering up the analog I/O modules FC4A-L03A1, -L03AP1, and -J2A1, a time interval is needed before turning on these modules. If a single power supply is used for the MicroSmart CPU module and the analog I/O modules, turn on the analog I/O modules at least 5 seconds (at 25°C) after turning off these modules. If separate power supplies are used for the MicroSmart CPU module and the analog I/O modules, turn on the analog I/O modules at least 30 seconds (at 25°C) after turning off the analog I/O modules whether the CPU module is powered up or not.

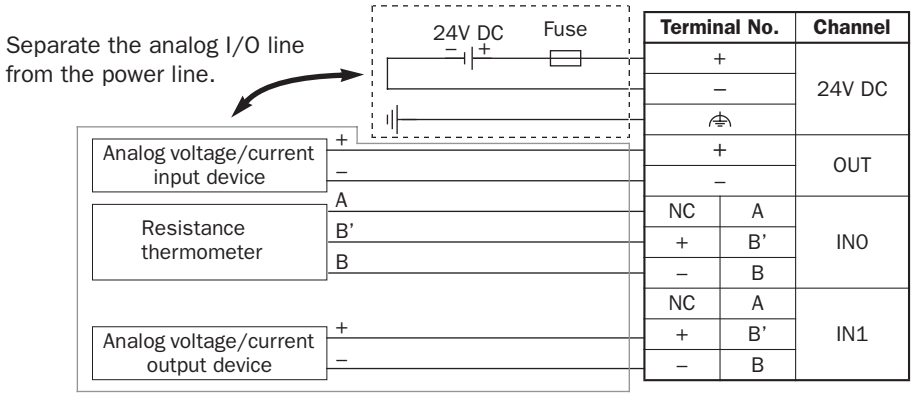
• Power Supply for FC4A-J4CN1, FC4A-J8C1, FC4A-J8AT1, and FC4A-K2C1

Use the same power supply for the MicroSmart CPU module and FC4A-J4CN1, FC4A-J8C1, FC4A-J8AT1, and FC4A-K2C1 to suppress the influence of noises.

After the CPU module has started to run, ladder refresh type analog input modules perform initialization for a maximum of 5 seconds. During this period, the analog input data have an indefinite value. Design the user program to make sure that the analog input data are read to the CPU module after the analog input operating status has changed to 0 (normal operation). For the analog input operating status, see page 26-13.

Wiring Analog I/O Lines

Separate the analog I/O lines, particularly resistance thermometer inputs, from motor lines as much as possible to suppress the influence of noises.



Terminal No.		Channel
+		24V DC
–		
⏚		
+		OUT
–		
NC	A	IN0
+	B'	
–	B	
NC	A	IN1
+	B'	
–	B	

AS-Interface Master Module

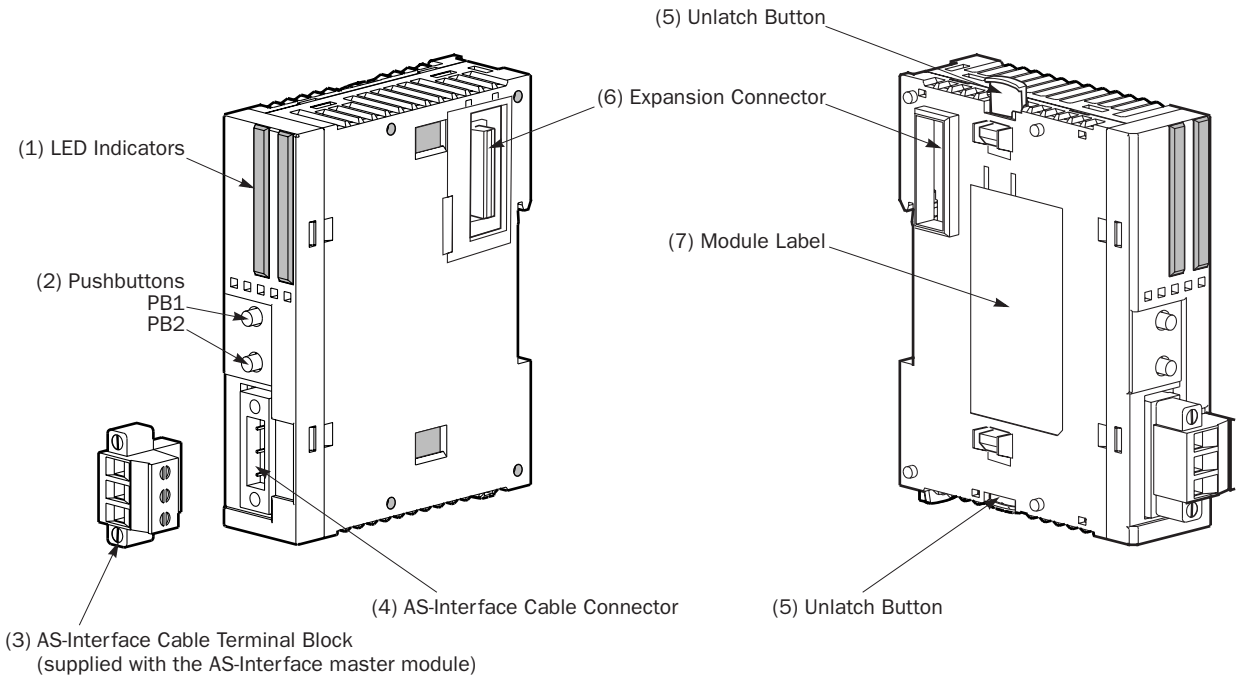
The AS-Interface master module can be used with FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3 CPU modules to communicate digital data with slaves, such as sensor, actuator, and remote I/O data.

One AS-Interface master module can be used with one CPU module. The AS-Interface master module can connect a maximum of 62 digital I/O slaves. A maximum of seven analog I/O slaves can also be connected to the AS-Interface master module (compliant with AS-Interface ver. 2.1 and analog slave profile 7.3).

AS-Interface Master Module Type Number

Module Name	Type No.
AS-Interface Master Module	FC4A-AS62M

Parts Description



(1) LED Indicators
Status LEDs: Indicate the AS-Interface bus status.
I/O LEDs: Indicate the I/O status of the slave specified by the address LEDs.
Address LEDs: Indicate slave addresses.

(2) Pushbuttons
Used to select slave addresses, change modes, and store configuration.

(3) AS-Interface Cable Terminal Block
Connects the AS-Interface cable.
One terminal block is supplied with the AS-Interface master module.
When ordering separately, specify Type No. FC4A-PMT3P and quantity (package quantity: 2).

(4) AS-Interface Cable Connector
Installs the AS-Interface cable terminal block.

(5) Unlatch Button
Used to unlatch the AS-Interface master module from the CPU or I/O module.

(6) Expansion Connector
Connects to the CPU and other I/O modules.

(7) Module Label
Indicates the AS-Interface master module Type No. and specifications.

General Specifications (AS-Interface Module)

Operating Temperature	0 to 55°C (operating ambient temperature, no freezing)
Storage Temperature	–25 to +70°C (no freezing)
Relative Humidity	Level RH1, 30 to 95% (non-condensing)
Pollution Degree	2 (IEC 60664)
Degree of Protection	IP20
Corrosion Immunity	Free from corrosive gases
Altitude	Operation: 0 to 2,000m (0 to 6,565 feet) Transport: 0 to 3,000m (0 to 9,840 feet)
Vibration Resistance	When mounted on a DIN rail: 10 to 57 Hz amplitude 0.075 mm, 57 to 150 Hz acceleration 9.8 m/s^2 2 hours per axis on each of three mutually perpendicular axes When mounted on a panel surface: 2 to 25 Hz amplitude 1.6 mm, 25 to 100 Hz acceleration 39.2 m/s^2 90 minutes per axis on each of three mutually perpendicular axes
Shock Resistance	147 m/s^2, 11 ms duration, 3 shocks per axis, on three mutually perpendicular axes (IEC 61131)
External Power Supply	AS-Interface power supply, 29.5 to 31.6V DC
AS-Interface Current Draw	65 mA (normal operation) 110 mA maximum
Effect of Improper Input Connection	No damage
Connector on Mother Board	MSTB2.5/3-GF-5.08BK (Phoenix Contact)
Connector Insertion/Removal Durability	100 times minimum
Internal Current Draw	80 mA (5V DC) 0 mA (24V DC)
AS-Interface Master Module Power Consumption	540 mW
Weight	85g

Communication Specifications (AS-Interface Module)

Maximum Bus Cycle	When 1 through 19 slaves are connected: 3 ms When 20 through 62 slaves are connected: 0.156 × (1 + N) ms where N is the number of active slaves 5 ms maximum when 31 standard or A/B slaves are connected 10 ms maximum when 62 A/B slaves are connected	
Maximum Slaves	Standard slaves: 31 A/B slaves: 62 When using a mix of standard slaves and A/B slaves together, the standard slaves can only use addresses 1(A) through 31(A). Also, when a standard slave takes a certain address, the B address of the same number cannot be used for A/B slaves.	
Maximum I/O Points	Standard slaves: 248 total (124 inputs + 124 outputs) A/B slaves: 434 total (248 inputs + 186 outputs)	
Maximum Cable Length	AS-Interface cable 2-wire flat cable	When using no repeater or extender: 100m When using a total of 2 repeaters or extenders: 300m
	Single wires	200 mm
Rated Bus Voltage	30V DC	

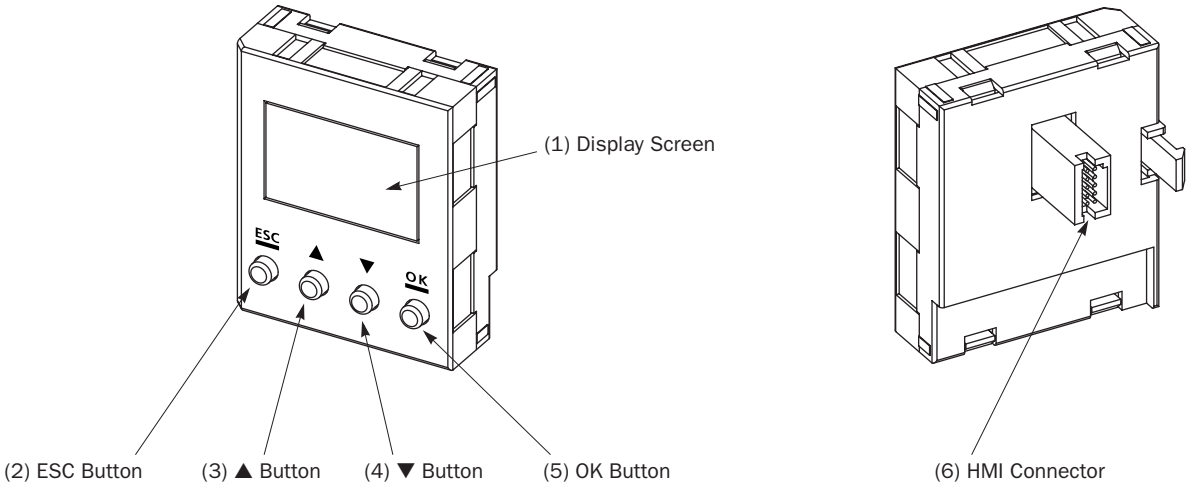
HMI Module

The optional HMI module can mount on any all-in-one type CPU module, and also on the HMI base module mounted next to any slim type CPU module. The HMI module makes it possible to manipulate the RAM data in the CPU module without using the Online menu options in WindLDR. For details about operating the HMI module, see page 5-32. For installing and removing the HMI module, see pages 3-3 and 3-4.

HMI Module Type Number

Module Name	Type No.
HMI Module	FC4A-PH1

Parts Description



(1) Display Screen	The liquid crystal display shows menus, operands, and data.
(2) ESC Button	Cancels the current operation, and returns to the immediately preceding operation.
(3) ▲ Button	Scrolls up the menu, or increments the selected operand number or value.
(4) ▼ Button	Scrolls down the menu, or decrements the selected operand number or value.
(5) OK Button	Goes into each control screen, or enters the current operation.
(6) HMI Connector	Connects to the all-in-one CPU module or HMI base module.

HMI Module Specifications

Type No.	FC4A-PH1
Power Voltage	5V DC (supplied from the CPU module)
Internal Current Draw	200 mA DC
Weight	20g

⚠ Caution

- Turn off the power to the MicroSmart before installing or removing the HMI module to prevent electrical shocks and damage to the HMI module.
- Do not touch the connector pins with hand, otherwise contact characteristics of the connector may be impaired.

HMI Base Module

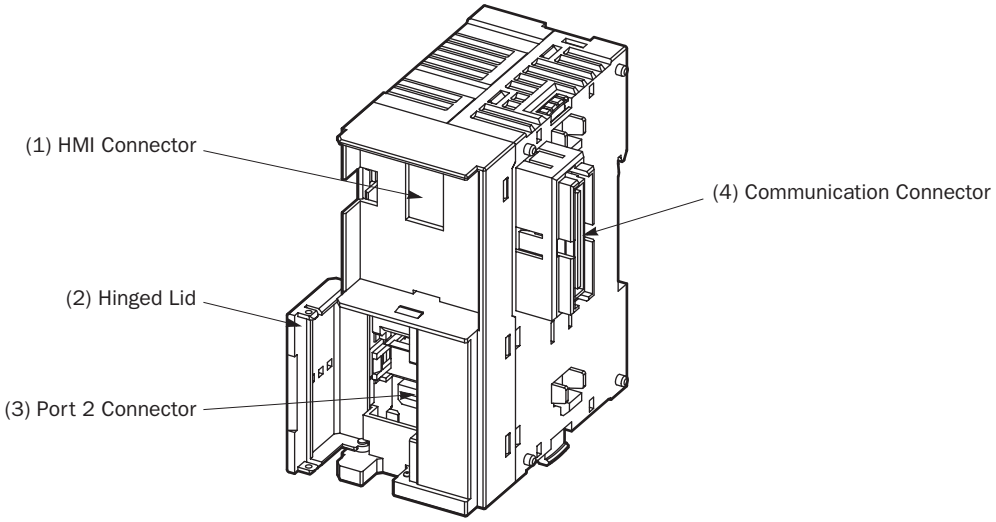
The HMI base module is used to install the HMI module when using the slim type CPU module. The HMI base module also has a port 2 connector to attach an optional RS232C or RS485 communication adapter.

When using the all-in-one type CPU module, the HMI base module is not needed to install the HMI module.

HMI Base Module Type Number

Module Name	Type No.
HMI Base Module	FC4A-HPH1

Parts Description



(1) HMI Connector	For installing the HMI module.
(2) Hinged Lid	Open the lid to gain access to the port 2 connector.
(3) Port 2 Connector	For installing an optional RS232C or RS485 communication adapter.
(4) Communication Connector	Connects to the slim type CPU module.

Communication Adapters and Communication Modules

All MicroSmart CPU modules have communication port 1 for RS232C communication. In addition, all-in-one 16- and 24-I/O type CPU modules have a port 2 connector. An optional communication adapter can be installed on the port 2 connector for RS232C or RS485 communication. The 10-I/O type CPU module does not have a port 2 connector.

A communication module can be attached to any slim type CPU module to use port 2 for additional RS232C or RS485 communication. When the HMI base module is attached to a slim type CPU module, a communication adapter can be installed to the port 2 connector on the HMI base module.

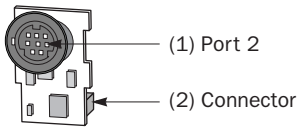
When using the RS232C communication adapter or communication module for port 2, maintenance communication, user communication, and modem communication are made possible. With the RS485 communication adapter or communication module installed, maintenance communication, data link communication, and user communication (upgraded CPU modules of slim 20-I/O relay output types and 40-I/O types only) can be used on port 2.

Communication Adapter and Communication Module Type Numbers

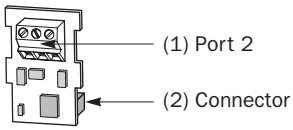
Name	Termination	Type No.
RS232C Communication Adapter	Mini DIN connector	FC4A-PC1
RS485 Communication Adapter	Mini DIN connector	FC4A-PC2
	Screw Terminal Block	FC4A-PC3
RS232C Communication Module	Mini DIN connector	FC4A-HPC1
RS485 Communication Module	Mini DIN connector	FC4A-HPC2
	Screw Terminal Block	FC4A-HPC3

Parts Description

RS232C Communication Adapter (Mini DIN)
RS485 Communication Adapter (Mini DIN)

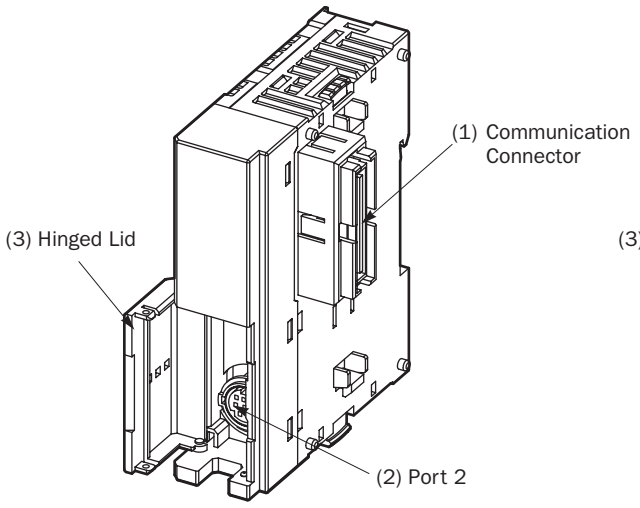


RS485 Communication Adapter (Screw Terminal)

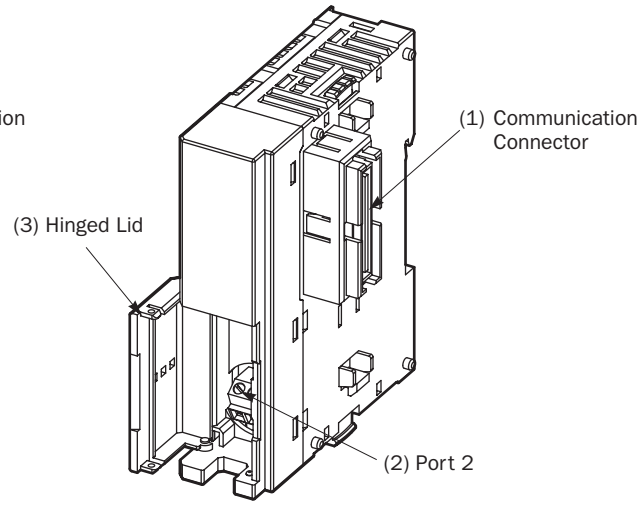


(1) Port 2 RS232C or RS485 communication port 2.
(2) Connector Connects to the port 2 connector on the all-in-one type CPU module or HMI base module.

RS232C Communication Module (Mini DIN)
RS485 Communication Module (Mini DIN)



RS485 Communication Module (Screw Terminal)



(1) Communication Connector Connects to the slim type CPU module.
(2) Port 2 RS232C or RS485 communication port 2.
(3) Hinged Lid Open the lid to gain access to port 2.

Communication Adapter and Communication Module Specifications

Type No.	FC4A-PC1 FC4A-HPC1	FC4A-PC2 FC4A-HPC2	FC4A-PC3 FC4A-HPC3
Standards	EIA RS232C	EIA RS485	EIA RS485
Maximum Baud Rate	19,200 bps	19,200 bps	Computer link: 19,200 bps User com.: 19,200 bps Data link: 38,400 bps
Maintenance Communication (Computer Link)	Possible	Possible	Possible
User Communication	Possible	Not possible	Possible (Note 1)
Modem Communication	Possible	Not possible	Not possible
Data Link Communication	Not possible	Not possible	Possible
Quantity of Slave Stations	—	—	31
Maximum Cable Length	Special cable	Special cable	200m (Note 2)
Isolation between Internal Circuit and Communication Port	Not isolated	Not isolated	Not isolated

Note 1: RS485 user communication is available on upgraded CPU modules only, see page 17-1.

Note 2: Recommended cable for RS485: Twisted-pair shielded cable with a minimum core wire of 0.3 mm^2. Conductor resistance 85 Ω/km maximum, shield resistance 20 Ω/km maximum.

The proper tightening torque of the terminal screws on the RS485 communication adapter and RS485 communication module is 0.22 to 0.25 N·m. For tightening the screws, use screwdriver SZS 0,4 x 2,5 (Phoenix Contact).

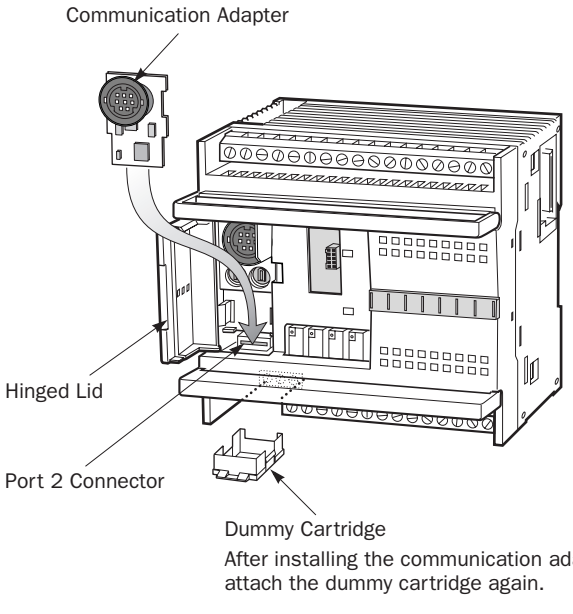
Installing the Communication Adapter and Communication Module

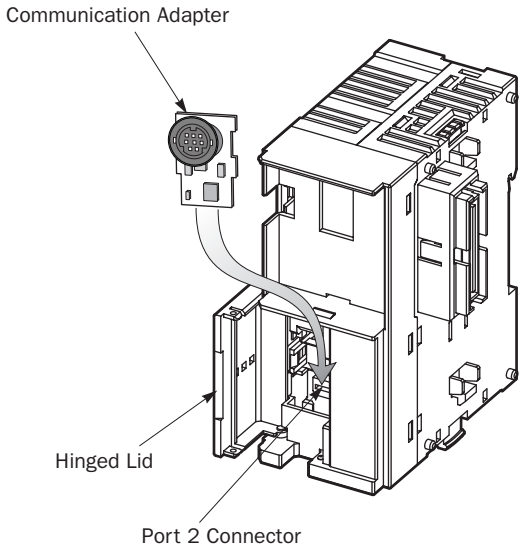
⚠ Caution

- Before installing the communication adapter or communication module, turn off the power to the MicroSmart CPU module. Otherwise, the communication adapter or CPU module may be damaged, or the MicroSmart may not operate correctly.

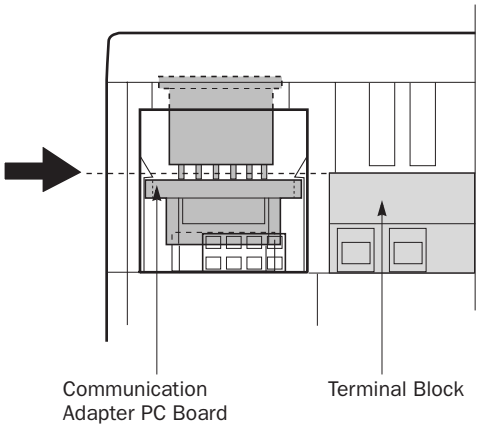
Communication Adapter

To install the communication adapter on the all-in-one type CPU module, open the hinged lid and remove the dummy cartridge. Push the communication adapter into the port 2 connector from the front until it bottoms and is secured by the latches. Similarly, when installing the communication adapter on the HMI base module, open the hinged lid, and push the communication adapter into the port 2 connector from the front until it bottoms and is secured by the latches.





After installing the communication adapter on an all-in-one type CPU module, view the communication adapter through the dummy cartridge opening, and check to see that the PC board of the communication adapter is in a lower level than the top of the terminal block.

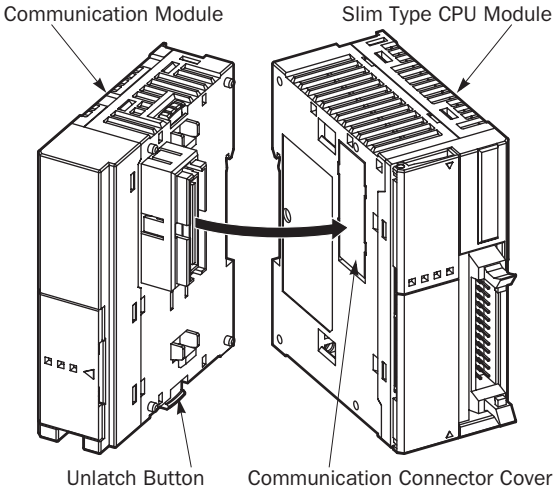


Communication Module

When installing a communication module on the slim type CPU module, remove the communication connector cover from the slim type CPU module. See page 3-6.

Place the communication module and CPU module side by side. Put the communication connectors together for easy alignment.

With the communication connectors aligned correctly and the blue unlatch button in the down position, press the communication module and CPU module together until the latches click to attach the modules together firmly. If the unlatch button is in the up position, push down the button to engage the latches.



Removing the Communication Adapter and Communication Module

⚠ Caution

- Before removing the communication adapter or communication module, turn off the power to the MicroSmart CPU module. Otherwise, the communication adapter or CPU module may be damaged, or the MicroSmart may not operate correctly.

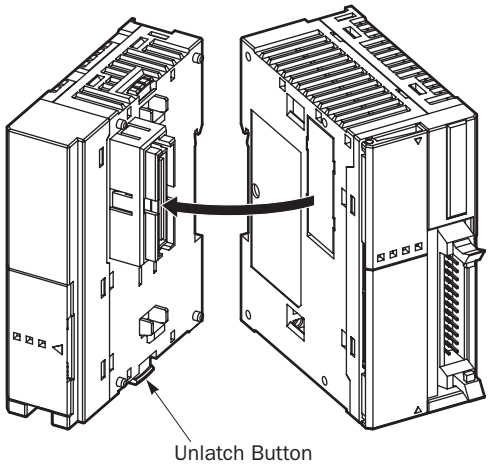
Communication Adapter

To remove the communication adapter from the all-in-one type CPU module, first remove the dummy cartridge. While pushing up the communication adapter PC board with a finger through the dummy cartridge opening, disengage the latches from the communication adapter using a flat screwdriver. Pull out the communication adapter from the port 2 connector. When removing the communication adapter from the HMI module, take similar steps.

Communication Module

If the modules are mounted on a DIN rail, first remove the modules from the DIN rail as described on page 3-7.

Push up the blue unlatch button to disengage the latches, and pull the modules apart as shown on the right.



Memory Cartridge

A user program can be stored on an optional memory cartridge installed on a MicroSmart CPU module from a computer running WindLDR, and the memory cartridge can be installed on another MicroSmart CPU module of the same type. Using a memory cartridge, the CPU module can exchange user programs without using a computer.

This feature is available on all models of CPU modules.

Memory Cartridge Type Number

Module Name	Type No.	Remarks
32KB Memory Cartridge	FC4A-PM32	
64KB Memory Cartridge	FC4A-PM64	The 64KB memory cartridge can be used on slim type CPU modules FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3 with system program ver. 201 or higher only. The program capacity expands up to 64,500 bytes (10,750 steps). To edit user programs over 32 KB (5200 steps), use WindLDR ver. 4.2 or higher.

User Program Execution Priority

Depending whether a memory cartridge is installed on the MicroSmart CPU module or not, a user program stored on the memory cartridge or on the CPU module EEPROM is executed, respectively.

Memory Cartridge	User Program Execution Priority
Installed on the CPU Module	The user program stored on the memory cartridge is executed. When the memory cartridge does not store a user program, the user program on the CPU module EEPROM is executed. When a memory cartridge is installed on the CPU module, the user program can be downloaded from the memory cartridge to the CPU module by designating in WindLDR Function Area Settings. To perform user program download from the memory cartridge, use CPU system program ver. 210 or higher and WindLDR ver 5.31 or higher.
Not installed on the CPU Module	The user program stored on the EEPROM in the CPU module is executed.

Memory Cartridge Specifications

Type No.	FC4A-PM32	FC4A-PM64
Memory Type	EEPROM	
Accessible Memory Capacity	32 KB	64 KB
Hardware for Storing Data	CPU module	
Software for Storing Data	WindLDR	WindLDR ver. 4.20 or higher
Quantity of Stored Programs	One user program can be stored on one memory cartridge.	

Note: The optional clock cartridge (FC4A-PT1) and the memory cartridge cannot be used together on the all-in-one type CPU module. The clock cartridge and the memory cartridge can be used together on the slim type CPU module.

User Program Compatibility

The CPU module can execute only user programs created for the same CPU module type. When installing a memory cartridge, make sure that the user program stored on the memory cartridge matches the CPU module type. If the user program is not for the same CPU module type, a user program syntax error occurs and the CPU module cannot run the user program.

⚠ Caution

- **Compatibility of User Program with CPU Modules**
  When a memory cartridge contains a user program for higher functionality, do not install the memory cartridge into CPU modules with lower functionality, otherwise the user program is not executed correctly. Make sure that the user program in the memory cartridge is compatible with the CPU module.

Downloading and Uploading User Program to and from Memory Cartridge

When a memory cartridge is installed on the CPU module, a user program is downloaded to and uploaded from the memory cartridge using WindLDR on a computer. When a memory cartridge is not installed on the CPU module, a user program is downloaded to and uploaded from the CPU module. For the procedures to download a user program from WindLDR on a computer, see page 4-8.

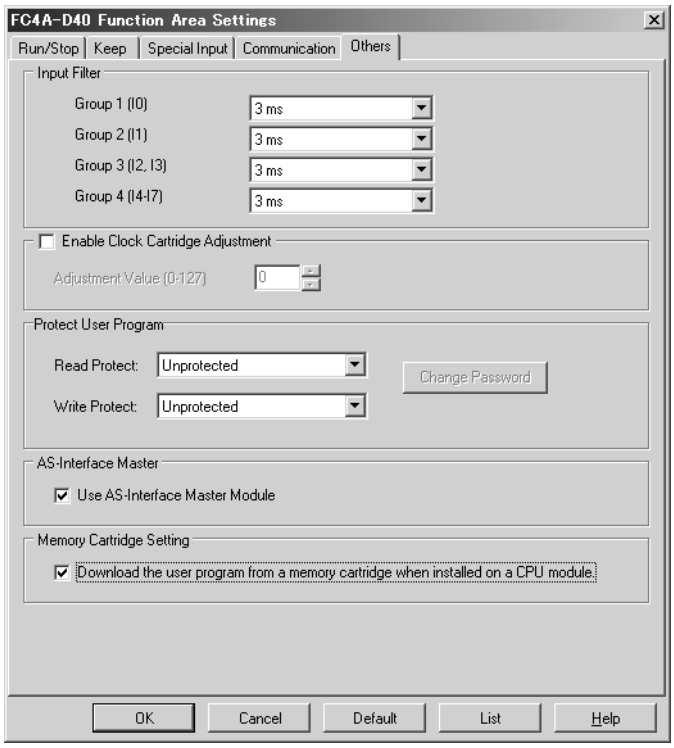
With a memory cartridge installed on a CPU module, if the user program stored on the memory cartridge does not match the CPU module type, downloading is possible, but uploading is not possible. To upload a user program, make sure that the existing user program stored on the memory cartridge matches the CPU module type. Downloading is always possible to new blank memory cartridges installed on any type of CPU modules.

Downloading User Program from Memory Cartridge to the CPU Module

To designate user program download from the memory cartridge, use a CPU module with system program ver. 210 or higher and WindLDR ver 5.31 or higher. Install a memory cartridge on the CPU module connected to a computer, and power up the CPU module.

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Others** tab.



3. Under **Memory Cartridge Setting**, click the check box to the left of **Download the user program from a memory cartridge when installed on a CPU module**.

   Checked: The user program is downloaded from the memory cartridge to the CPU module.
   Unchecked: The user program is not downloaded from the memory cartridge to the CPU module.

4. Click the **OK** button.
5. Download the user program to the memory cartridge to complete the designation in the memory cartridge.
6. Shut down the CPU module and remove the memory cartridge. Install the memory cartridge on another CPU module. Power up the CPU module, then the user program is downloaded from the memory cartridge to the CPU module.

If the user program in the CPU module is write-protected or read/write-protected, the user program can be downloaded only when the password in the memory cartridge matches the password in the CPU module. For user program protection password, see page 5-25.

Installing and Removing the Memory Cartridge

⚠ Caution

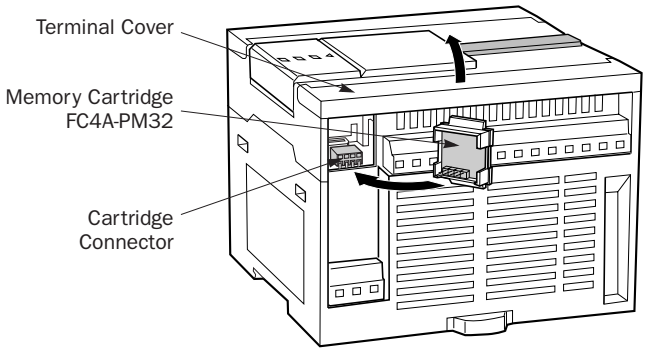
- Before installing or removing the memory cartridge, turn off the power to the MicroSmart CPU module. Otherwise, the memory cartridge or CPU module may be damaged, or the MicroSmart may not operate correctly.
- Do not touch the connector pins with hand, otherwise electrostatic discharge may damage the internal elements.

All-in-One Type CPU Module

The cartridge connector is normally closed with a dummy cartridge. To install the memory cartridge, open the terminal cover and remove the dummy cartridge from the CPU module. Make sure of correct orientation of the memory cartridge. Insert the memory cartridge into the cartridge connector until it bottoms. Do not insert the memory cartridge diagonally, otherwise the terminal pins will be deformed.

After installing the memory cartridge, close the terminal cover.

To remove the memory cartridge, hold both edges of the memory cartridge and pull it out.

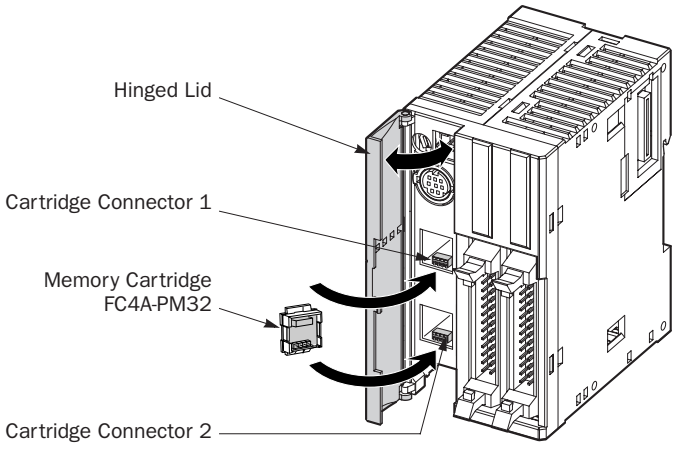


Slim Type CPU Module

Cartridge connectors 1 and 2 are normally closed with a dummy cartridge. To install the memory cartridge, open the hinged lid and remove the dummy cartridge from the CPU module. Make sure of correct orientation of the memory cartridge, and insert the memory cartridge into cartridge connector 1 or 2 until it bottoms. After installing the memory cartridge, close the hinged lid.

Only one memory cartridge can be installed to either cartridge connector 1 or 2 on the slim type CPU module. A memory cartridge and a clock cartridge can be installed at the same time.

To remove the memory cartridge, hold both edges of the memory cartridge and pull it out.



Clock Cartridge

With the optional clock cartridge installed on any type of MicroSmart CPU modules, the MicroSmart can be used for time-scheduled control such as illumination and air conditioners. For setting the calendar/clock, see page 15-5.

Clock Cartridge Type Number

Module Name	Type No.
Clock Cartridge	FC4A-PT1

Clock Cartridge Specifications

Accuracy	±30 sec/month (typical) at 25°C
Backup Duration	Approx. 30 days (typical) at 25°C after backup battery fully charged
Battery	Lithium secondary battery
Charging Time	Approx. 10 hours for charging from 0% to 90% of full charge
Battery Life	Approx. 100 recharge cycles after discharging down to 10% of full charge
Replaceability	Not possible to replace battery

The optional memory cartridge (FC4A-PM32) and the clock cartridge cannot be used together on the all-in-one type CPU module. The memory cartridge and the clock cartridge can be used together on the slim type CPU module.

Installing and Removing the Clock Cartridge

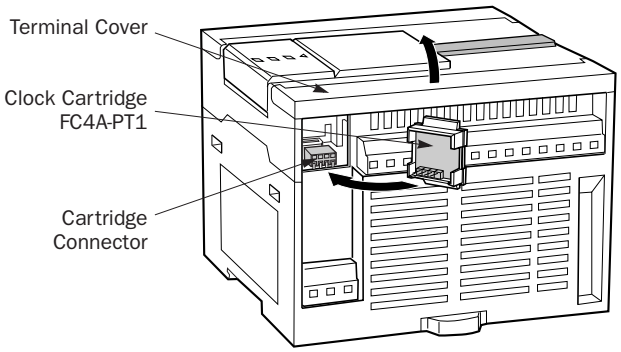
⚠ Caution

- Before installing or removing the clock cartridge, turn off the power to the MicroSmart CPU module. Otherwise, the clock cartridge or CPU module may be damaged, or the MicroSmart may not operate correctly.
- Do not touch the connector pins with hand, otherwise electrostatic discharge may damage the internal elements.

All-in-One Type CPU Module

The cartridge connector is normally closed with a dummy cartridge. To install the clock cartridge, open the terminal cover and remove the dummy cartridge from the CPU module. Make sure of correct orientation of the clock cartridge. Insert the clock cartridge into the cartridge connector until it bottoms. Do not insert the clock cartridge diagonally, otherwise the terminal pins will be deformed. After installing the clock cartridge, close the terminal cover.

To remove the clock cartridge, hold both edges of the clock cartridge and pull it out.

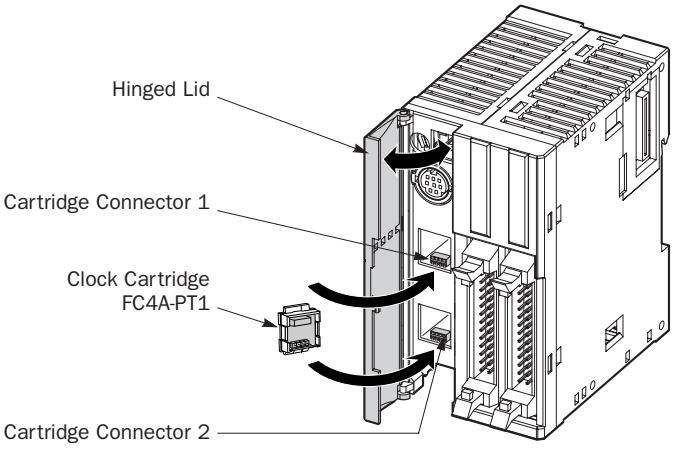


Slim Type CPU Module

To install the clock cartridge, open the hinged lid and remove the dummy cartridge from the CPU module. Make sure of correct orientation of the clock cartridge, and insert the clock cartridge into cartridge connector 1 or 2 until it bottoms. After installing the clock cartridge, close the hinged lid.

Only one clock cartridge can be installed to either cartridge connector 1 or 2 on the slim type CPU module. A clock cartridge and a memory cartridge can be installed at the same time.

To remove the clock cartridge, hold both edges of the clock cartridge and pull it out.

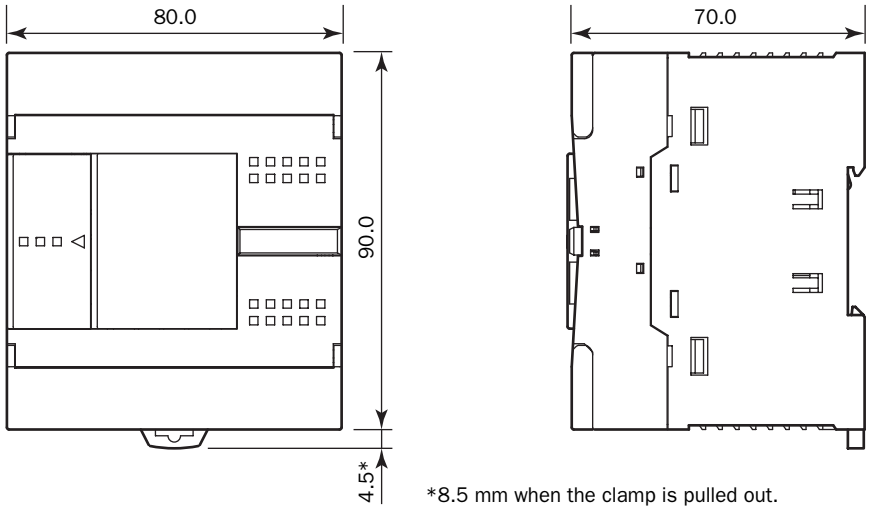


Dimensions

All MicroSmart modules have the same profile for consistent mounting on a DIN rail.

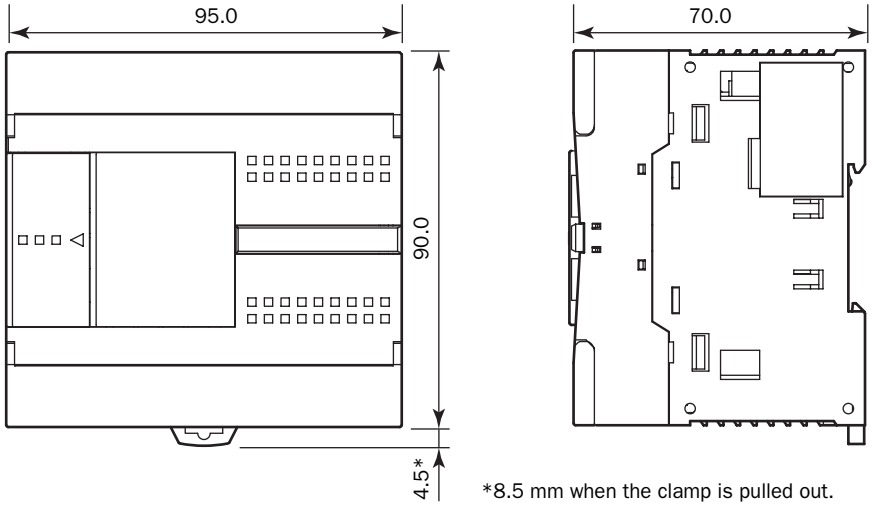
CPU Modules

FC4A-C10R2, FC4A-C10R2C, FC4A-C16R2, FC4A-C16R2C



*8.5 mm when the clamp is pulled out.

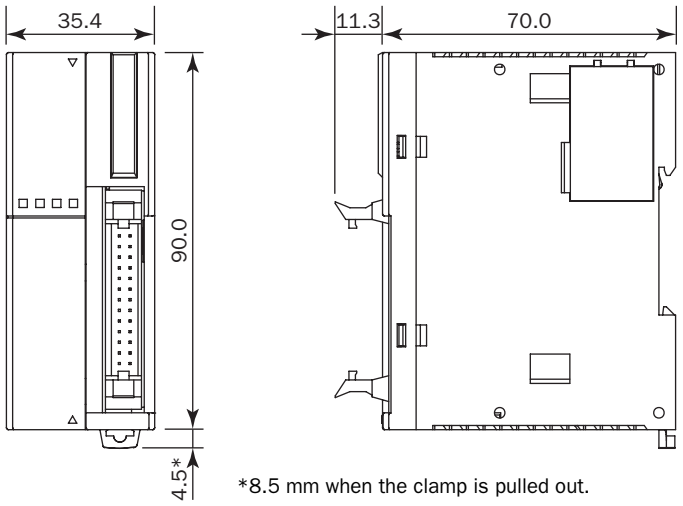
FC4A-C24R2, FC4A-C24R2C



*8.5 mm when the clamp is pulled out.

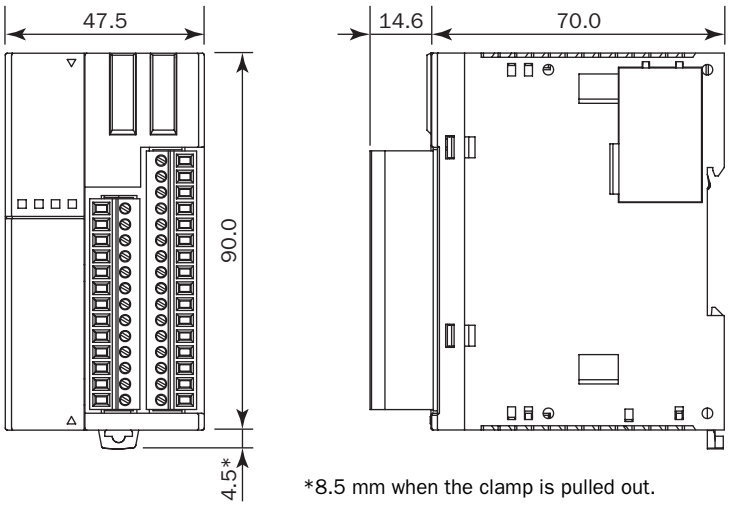
All dimensions in mm.

FC4A-D20K3, FC4A-D20S3



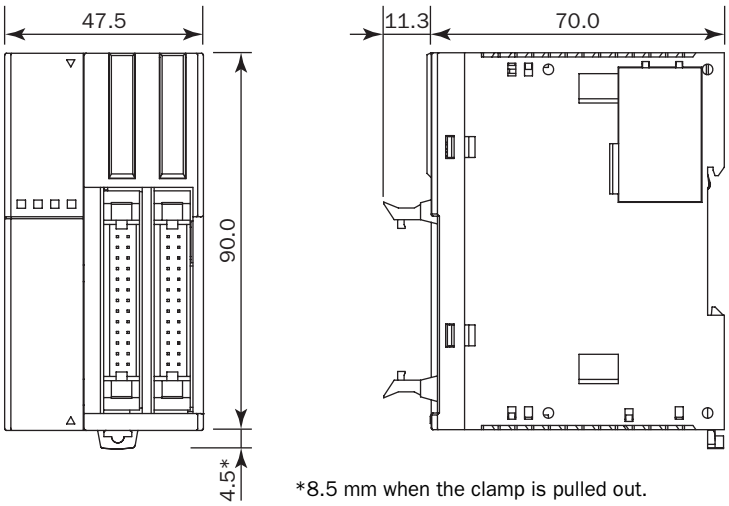
*8.5 mm when the clamp is pulled out.

FC4A-D20RK1, FC4A-D20RS1



*8.5 mm when the clamp is pulled out.

FC4A-D40K3, FC4A-D40S3

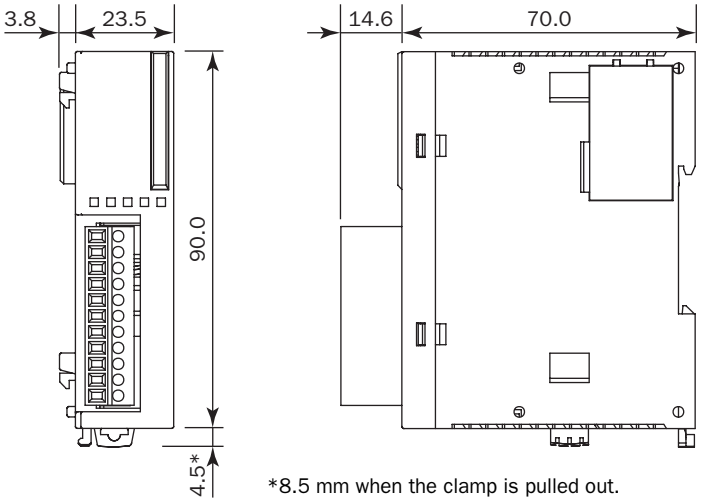


*8.5 mm when the clamp is pulled out.

All dimensions in mm.

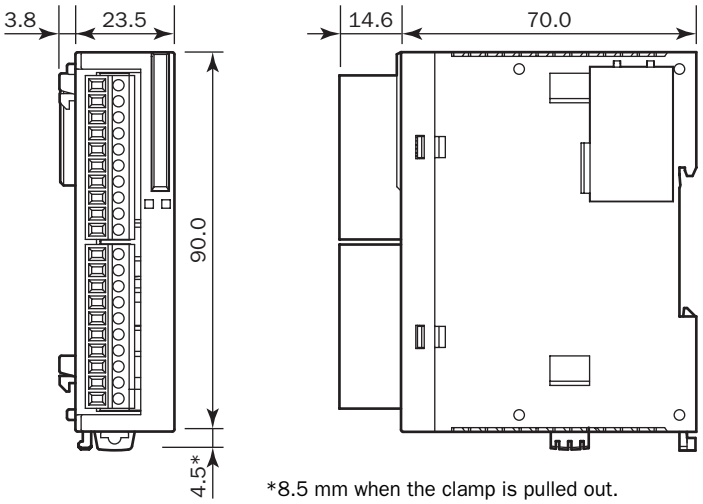
I/O Modules

FC4A-N08B1, FC4A-N08A11, FC4A-R081, FC4A-T08K1, FC4A-T08S1, FC4A-M08BR1, FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, FC4A-K1A1



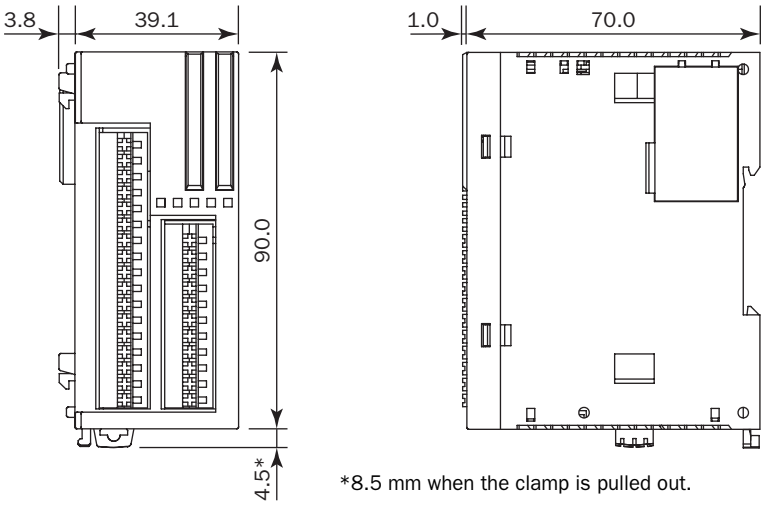
*8.5 mm when the clamp is pulled out.

FC4A-N16B1, FC4A-R161



*8.5 mm when the clamp is pulled out.

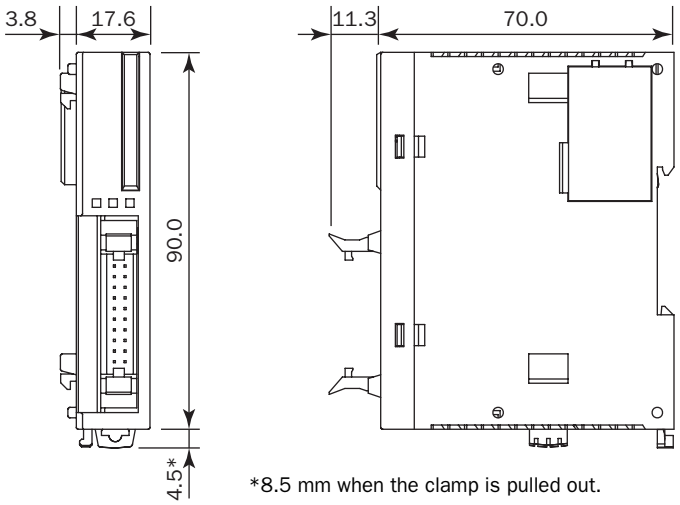
FC4A-M24BR2



*8.5 mm when the clamp is pulled out.

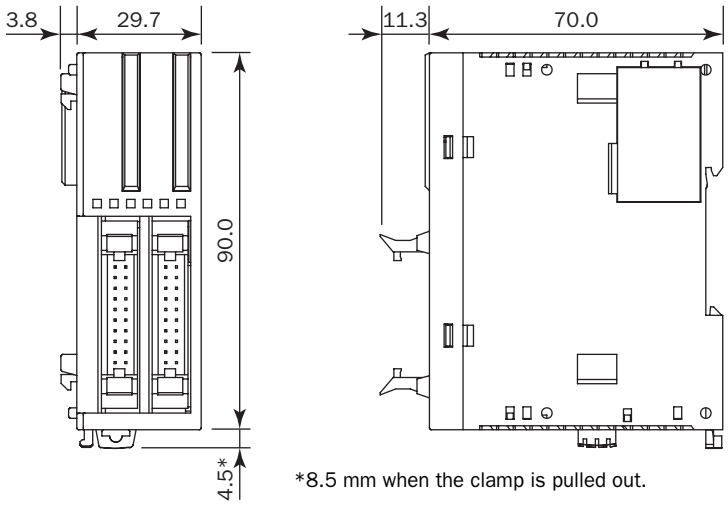
All dimensions in mm.

FC4A-N16B3, FC4A-T16K3, FC4A-T16S3



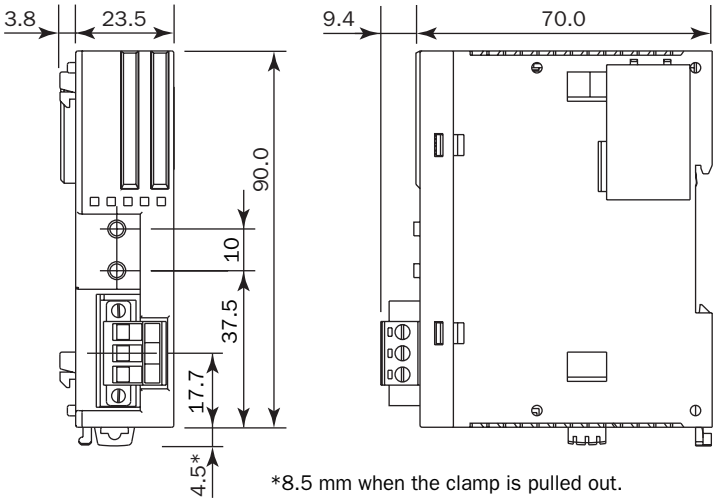
*8.5 mm when the clamp is pulled out.

FC4A-N32B3, FC4A-T32K3, FC4A-T32S3



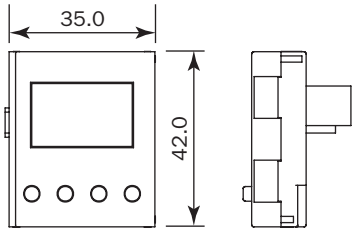
*8.5 mm when the clamp is pulled out.

AS-Interface Module FC4A-AS62M

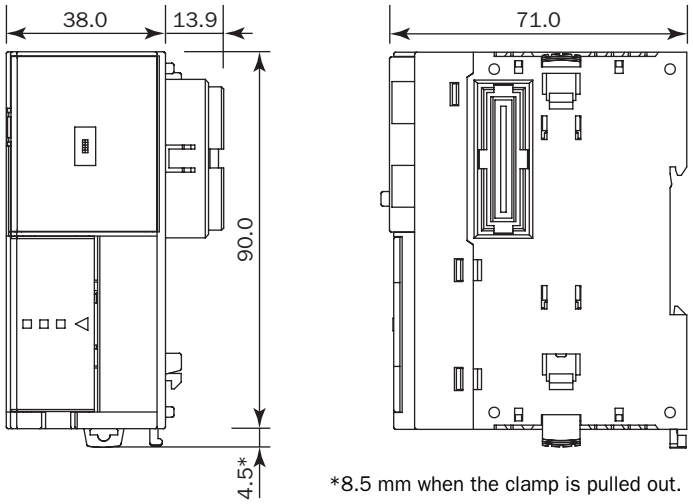


*8.5 mm when the clamp is pulled out.

HMI Module FC4A-PH1



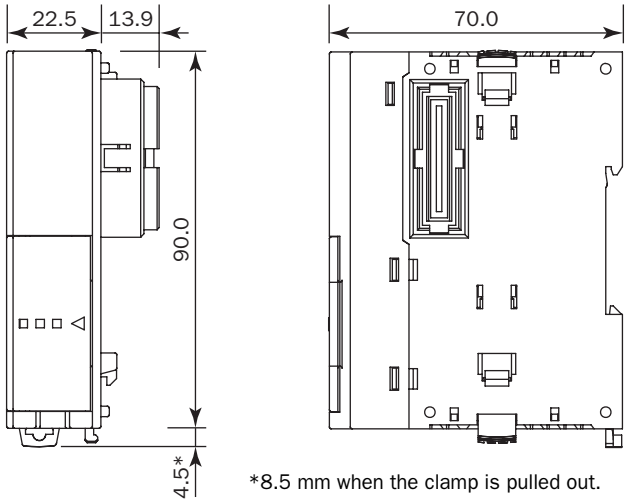
HMI Base Module FC4A-HPH1



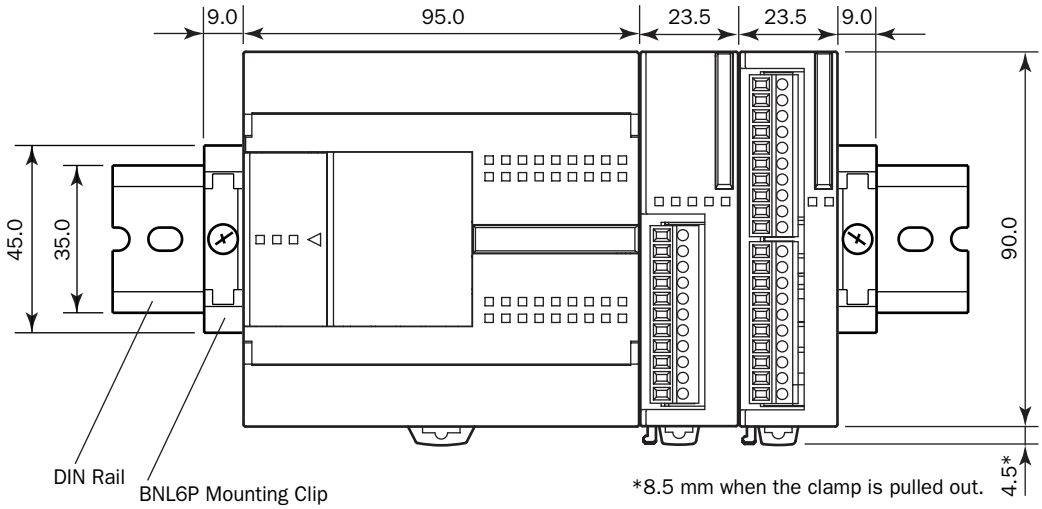
*8.5 mm when the clamp is pulled out.

All dimensions in mm.

Communication Modules FC4A-HPC1, FC4A-HPC2, FC4A-HPC3



Example: The following figure illustrates a system setup consisting of the all-in-one 24-I/O type CPU module, an 8-point relay output module, and a 16-point DC input module mounted on a 35-mm-wide DIN rail using BNL6P mounting clips.



All dimensions in mm.

3: INSTALLATION AND WIRING

Introduction

This chapter describes the methods and precautions for installing and wiring MicroSmart modules.

Before starting installation and wiring, be sure to read "Safety Precautions" in the beginning of this manual and understand precautions described under Warning and Caution.

⚠ Warning

- Turn off the power to the MicroSmart before starting installation, removal, wiring, maintenance, and inspection of the MicroSmart. Failure to turn power off may cause electrical shocks or fire hazard.
- Emergency stop and interlocking circuits must be configured outside the MicroSmart. If such a circuit is configured inside the MicroSmart, failure of the MicroSmart may cause disorder of the control system, damage, or accidents.
- Special expertise is required to install, wire, program, and operate the MicroSmart. People without such expertise must not use the MicroSmart.

⚠ Caution

- Prevent metal fragments and pieces of wire from dropping inside the MicroSmart housing. Put a cover on the MicroSmart modules during installation and wiring. Ingress of such fragments and chips may cause fire hazard, damage, or malfunction.
- Do not touch the connector pins with hand, otherwise electrostatic discharge may damage the internal elements.

Installation Location

The MicroSmart must be installed correctly for optimum performance.

The MicroSmart is designed for installation in a cabinet. Do not install the MicroSmart outside a cabinet.

The environment for using the MicroSmart is "Pollution degree 2." Use the MicroSmart in environments of pollution degree 2 (according to IEC 60664-1).

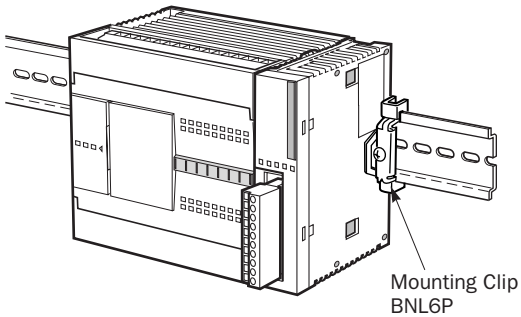
Make sure that the operating temperature does not drop below 0°C or exceed 55°C. If the temperature does exceed 55°C, use a fan or cooler.

Mount the MicroSmart on a vertical plane as shown at right.

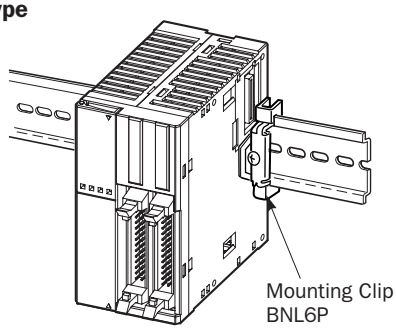
To eliminate excessive temperature build-up, provide ample ventilation. Do not install the MicroSmart near, and especially above, any device which generates considerable heat, such as a heater, transformer, or large-capacity resistor. The relative humidity should be above 30% and below 95%.

The MicroSmart should not be exposed to excessive dust, dirt, salt, direct sunlight, vibrations, or shocks. Do not use the MicroSmart in an area where corrosive chemicals or flammable gases are present. The modules should not be exposed to chemical, oil, or water splashes.

All-in-One Type



Slim Type



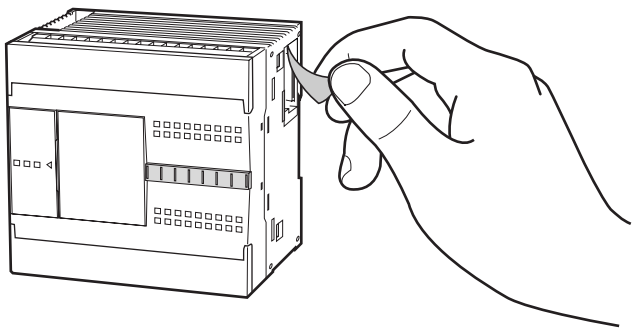
Assembling Modules

Caution

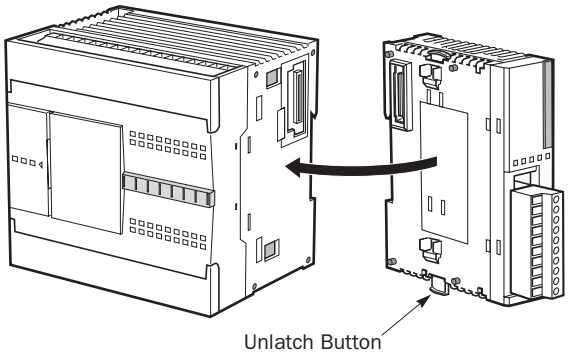
- Assemble MicroSmart modules together before mounting the modules onto a DIN rail. Attempt to assemble modules on a DIN rail may cause damage to the modules.
- Turn off the power to the MicroSmart before assembling the modules. Failure to turn power off may cause electrical shocks.

The following example demonstrates the procedure for assembling the all-in-one 24-I/O type CPU module and an I/O module together. When assembling slim type CPU modules, take the same procedure.

1. When assembling an input or output module, remove the expansion connector seal from the 24-I/O type CPU module.



2. Place the CPU module and I/O module side by side. Put the expansion connectors together for easy alignment.
3. With the expansion connectors aligned correctly and the blue unlatch button in the down position, press the CPU module and I/O module together until the latches click to attach the modules together firmly. If the unlatch button is in the up position, push down the button to engage the latches.

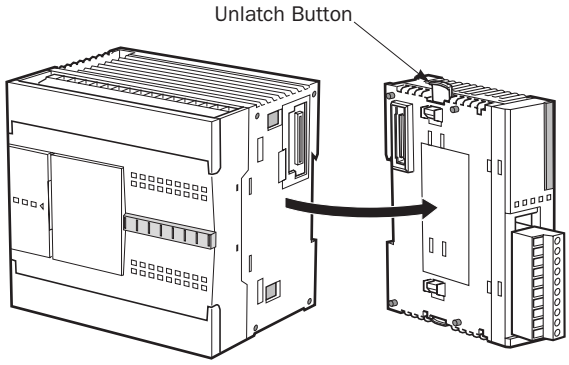


Disassembling Modules

Caution

- Remove the MicroSmart modules from the DIN rail before disassembling the modules. Attempt to disassemble modules on a DIN rail may cause damage to the modules.
- Turn off the power to the MicroSmart before disassembling the modules. Failure to turn power off may cause electrical shocks.

1. If the modules are mounted on a DIN rail, first remove the modules from the DIN rail as described on page 3-7.
2. Push up the blue unlatch button to disengage the latches, and pull the modules apart as shown. When disassembling slim type CPU modules, take the same procedure.



Installing the HMI Module

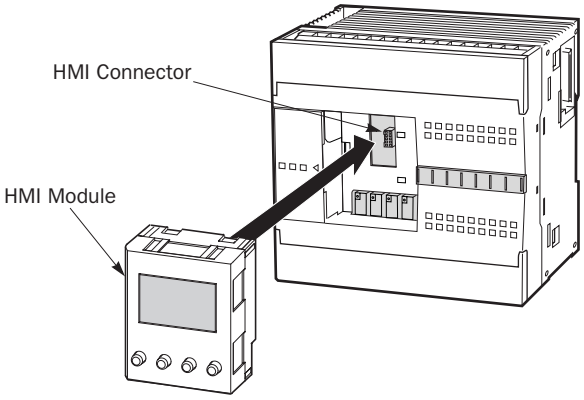
⚠ Caution

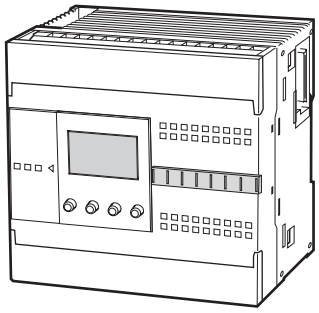
- Turn off the power to the MicroSmart before installing or removing the HMI module to prevent electrical shocks.
- Do not touch the connector pins with hand, otherwise electrostatic discharge may damage the internal elements.

The optional HMI module (FC4A-PH1) can mount on any all-in-one type CPU module, and also on the HMI base module mounted next to any slim type CPU module. For specifications of the HMI module, see page 2-60. For details about operating the HMI module, see page 5-32.

All-in-One Type

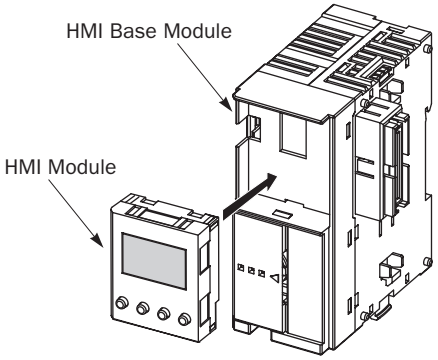
1. Remove the HMI connector cover from the CPU module. Locate the HMI connector inside the CPU module.
2. Push the HMI module into the HMI module connector in the CPU module until the latch clicks.



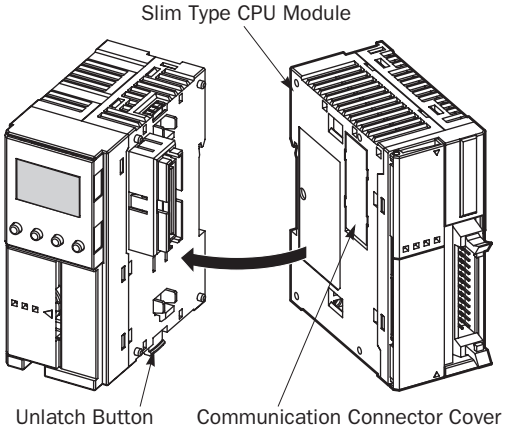


Slim Type

1. When using the HMI module with the slim type CPU module, prepare the optional HMI base module (FC4A-HPH1). See page 2-61.
2. Locate the HMI connector inside the HMI base module. Push the HMI module into the HMI connector in the HMI base module until the latch clicks.



3. Remove the communication connector cover from the slim type CPU module. See page 3-6.
4. Place the HMI base module and CPU module side by side. With the communication connectors aligned correctly and the blue unlatch button in the down position, press the HMI base module and CPU module together until the latches click to attach the modules together firmly. If the unlatch button is in the up position, push down the button to engage the latches.



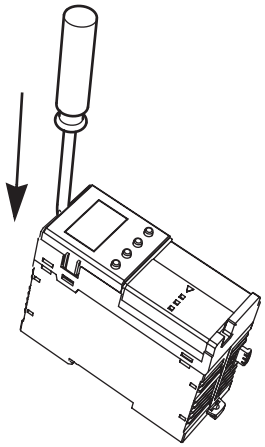
Removing the HMI Module

Caution

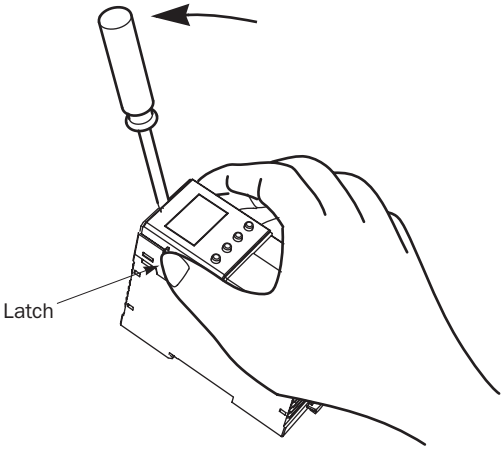
- Turn off the power to the MicroSmart before installing or removing the HMI module to prevent electrical shocks.
- Do not touch the connector pins with hand, otherwise electrostatic discharge may damage the internal elements.

This section describes the procedures for removing the HMI module from the optional HMI base module mounted next to any slim type CPU module.

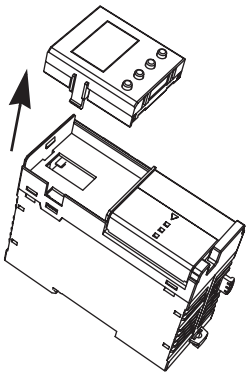
1. Insert a thin flat screwdriver (ø3.0 mm maximum) between the gap on top of the HMI module until the tip of the screwdriver bottoms.



2. While turning the screwdriver in the direction as shown, disengage the latch on the HMI module and pull out the HMI module.



3. Remove the HMI module from the HMI base module.



Removing the Terminal Blocks

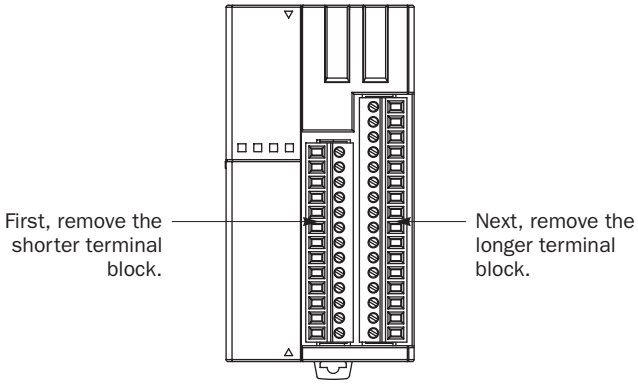
⚠ Caution

- Turn off the power to the MicroSmart before installing or removing the terminal blocks to prevent electrical shocks.
- Use the correct procedures to remove the terminal blocks, otherwise the terminal blocks may be damaged.

This section describes the procedures for removing the terminal blocks from slim type CPU modules FC4A-D20RK1 and FC4A-D20RS1.

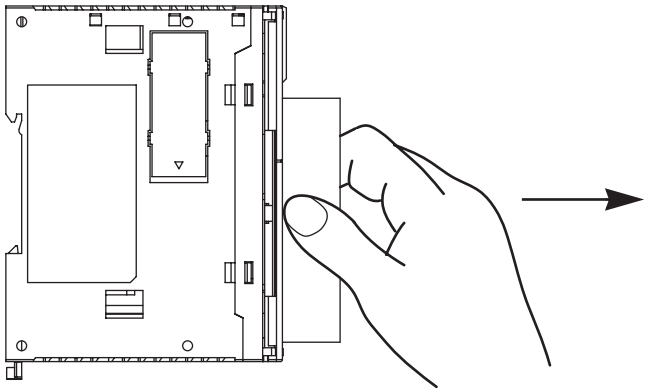
1. Before removing the terminal blocks, disconnect all wires from the terminal blocks.

Remove the shorter terminal block on the left first, then remove the longer one on the right.

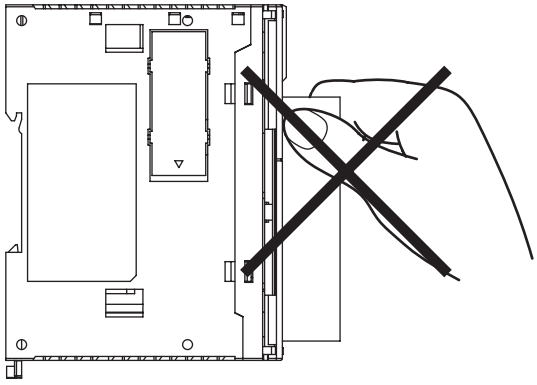


FC4A-D20RK1 and FC4A-D20RS1

2. When removing the longer terminal block, hold the center of the terminal block, and pull it out straight.



3. Do not pull one end of the longer terminal block, otherwise the terminal block may be damaged.

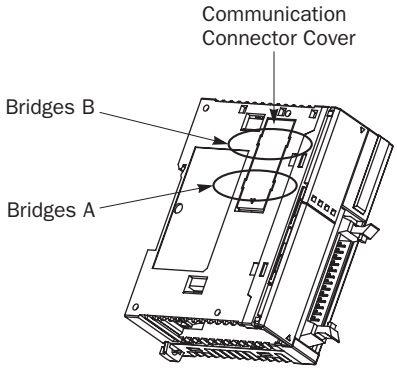


Removing the Communication Connector Cover

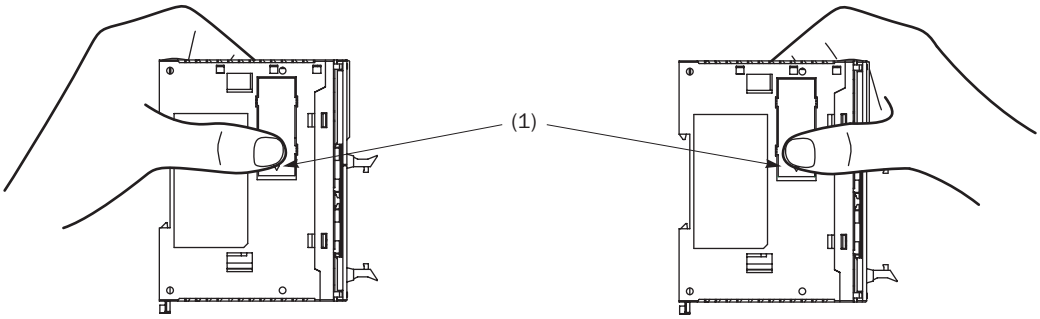
Caution

- When using a thin screwdriver to pull out the communication connector cover, insert the screwdriver carefully and do not damage the electronic parts inside the CPU module.
- When first pushing in the communication connector cover to break, take care not to injure your finger.

Before mounting a communication module or HMI base module next to the slim type CPU module, the communication connector cover must be removed from the CPU module. Break the communication connector cover on the slim type CPU module as described below.



1. Carefully push in the communication connector cover at position (1) to break bridges A as shown in either figure below.



2. The other end (2) of the communication connector cover will come out as shown at left below. Push in this end.
3. Then, the opposite end (3) will come out. If the end does not come out, insert a thin screwdriver into the gap and pull out the end (3).

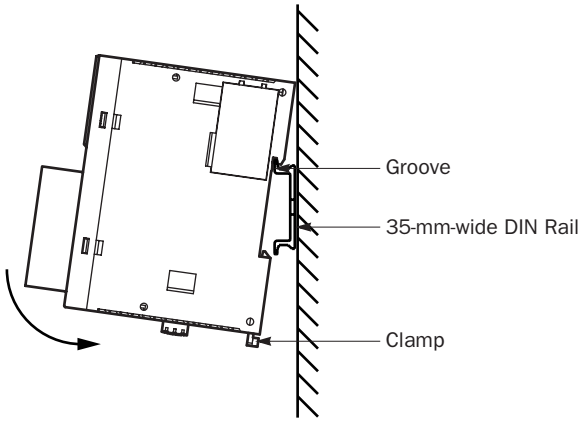
   Hold the communication connector cover at (3), and pull off the communication connector cover to break bridges B.

Mounting on DIN Rail

⚠ Caution

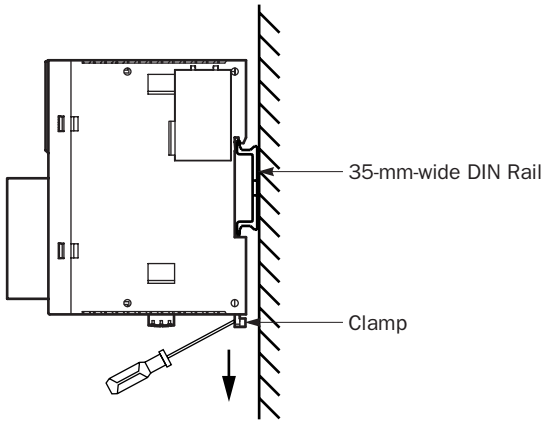
- Install the MicroSmart modules according to instructions described in this user's manual. Improper installation will result in falling, failure, or malfunction of the MicroSmart.
- Mount the MicroSmart modules on a 35-mm-wide DIN rail or a panel surface. Applicable DIN rail: IDEC's BAA1000PN10 or BAP1000PN10 (1000mm/39.4" long)

1. Fasten the DIN rail to a panel using screws firmly.
2. Pull out the clamp from each MicroSmart module, and put the groove of the module on the DIN rail. Press the modules towards the DIN rail and push in the clamps as shown on the right.
3. Use BNL6 end clips on both sides of the MicroSmart modules to prevent moving sideways.



Removing from DIN Rail

1. Insert a flat screwdriver into the slot in the clamp.
2. Pull out the clamps from the modules.
3. Turn the MicroSmart modules bottom out.

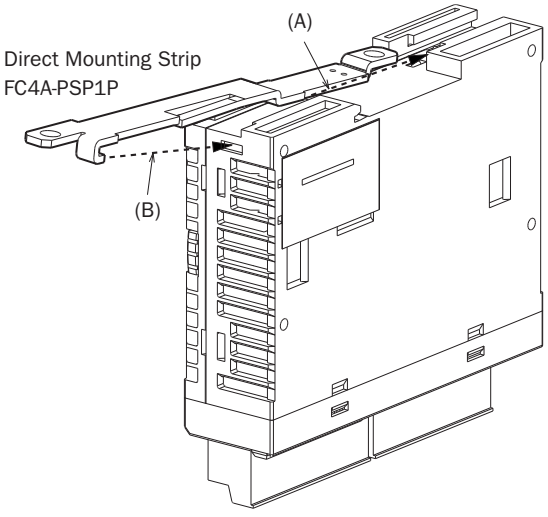


Direct Mounting on Panel Surface

MicroSmart modules can also be mounted on a panel surface inside a console. When mounting a slim type CPU module, digital I/O module, analog I/O module, HMI base module, or communication module, use optional direct mounting strip FC4A-PSP1P as described below.

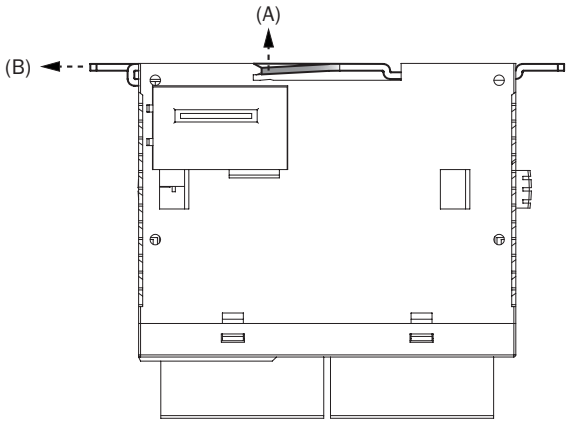
Installing the Direct Mounting Strip

1. Remove the clamp from the module by pushing the clamp inward.
2. Insert the direct mounting strip into the slot where the clamp has been removed (A). Further insert the direct mounting strip until the hook enters into the recess in the module (B).



Removing the Direct Mounting Strip

1. Insert a flat screwdriver under the latch of the direct mounting strip to release the latch (A).
2. Pull out the direct mounting strip (B).

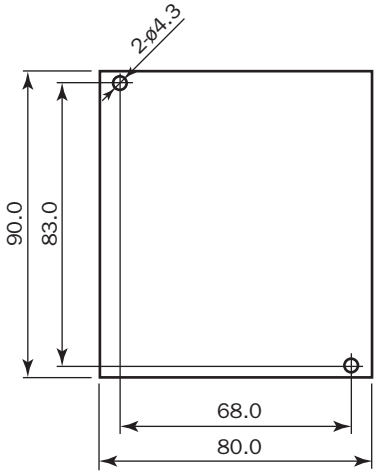


Mounting Hole Layout for Direct Mounting on Panel Surface

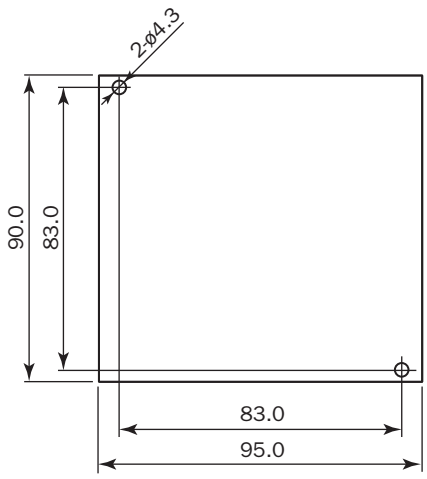
Make mounting holes of ø4.3 mm as shown below and use M4 screws (6 or 8 mm long) to mount the MicroSmart modules on the panel surface.

• CPU Modules

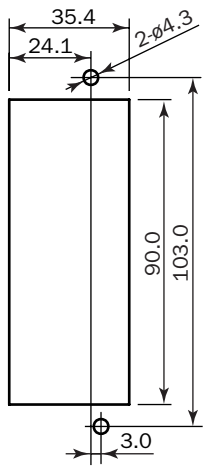
FC4A-C10R2, FC4A-C10R2C, FC4A-C16R2, FC4A-C16R2C



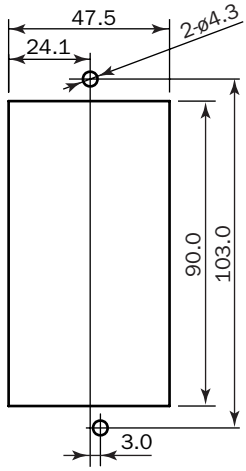
FC4A-C24R2, FC4A-C24R2C



FC4A-D20K3, FC4A-D20S3



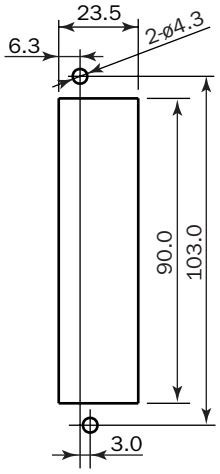
FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, FC4A-D40S3



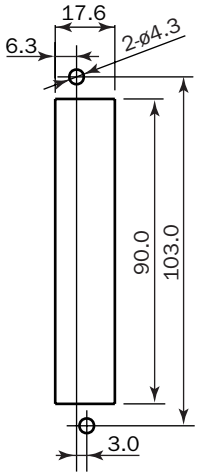
All dimensions in mm.

• I/O Modules

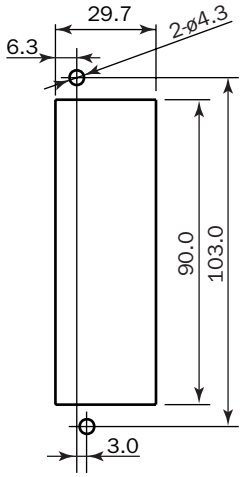
FC4A-N08B1, FC4A-N16B1, FC4A-N08A11, FC4A-R081, FC4A-R161, FC4A-T08K1, FC4A-T08S1, FC4A-M08BR1, FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, FC4A-K1A1



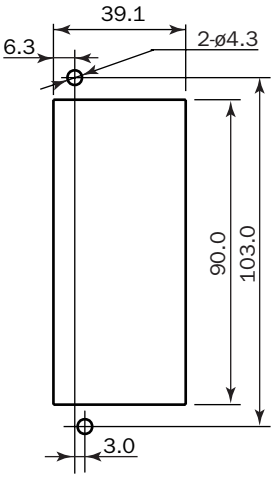
FC4A-N16B3, FC4A-T16K3, FC4A-T16S3



FC4A-N32B3, FC4A-T32K3, FC4A-T32S3

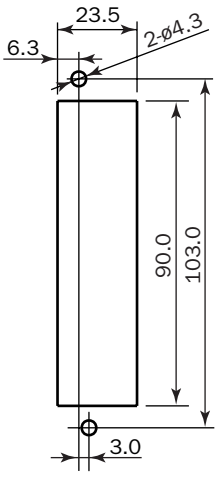


FC4A-M24BR2



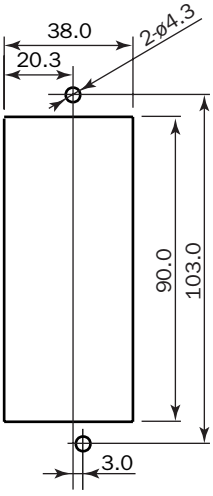
• AS-Interface Module

FC4A-AS62M



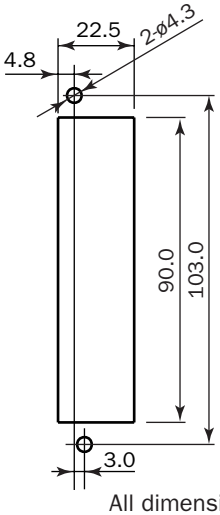
• HMI Base Module

FC4A-HPH1



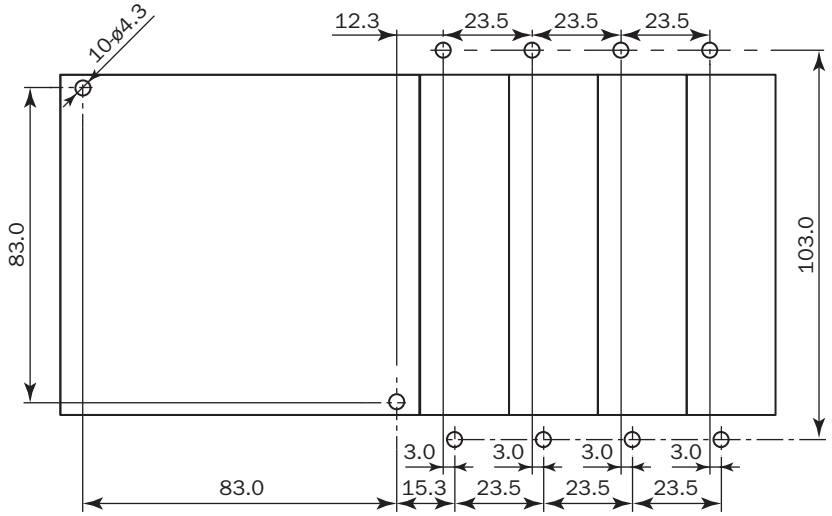
• Communication Modules

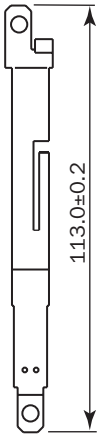
FC4A-HPC1, FC4A-HPC2, FC4A-HPC3



All dimensions in mm.

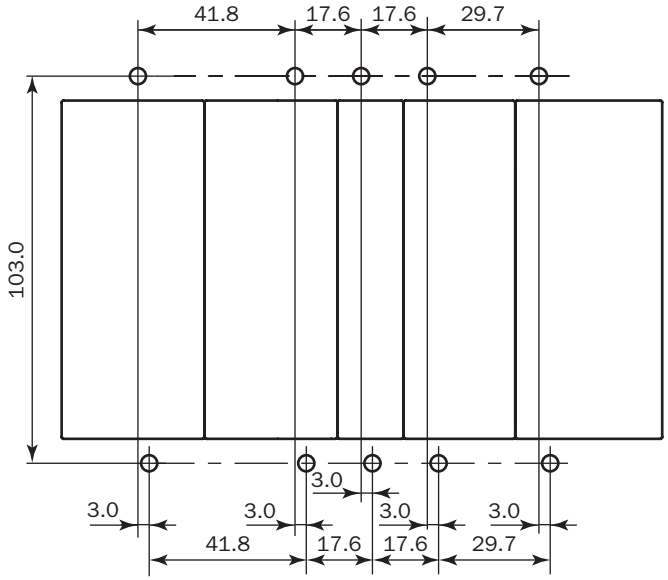
Example 1: Mounting hole layout for FC4A-C24R2 and 23.5-mm-wide I/O modules





Direct Mounting Strip
FC4A-PSP1P

Example 2: Mounting hole layout for, from left, FC4A-HPH1, FC4A-D20K3, FC4A-N16B3, FC4A-N32B3, and FC4A-M24R2 modules



All dimensions in mm.

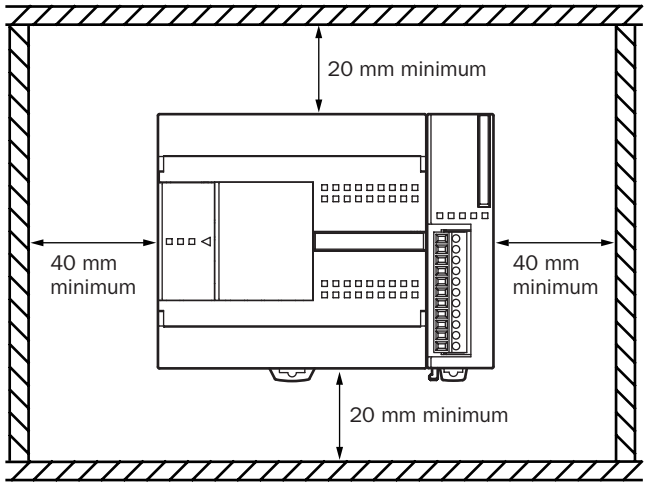
Installation in Control Panel

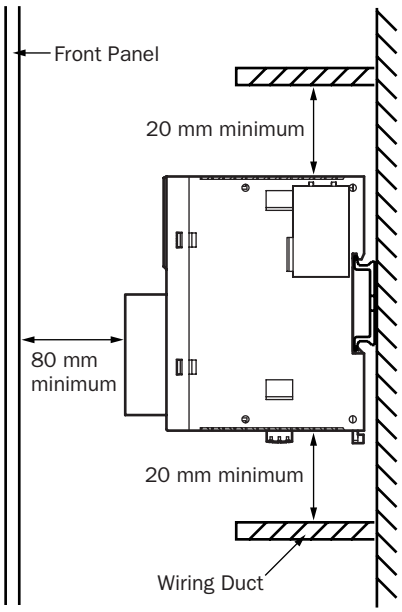
The MicroSmart modules are designed for installation in a cabinet. Do not install the MicroSmart modules outside a cabinet.

The environment for using the MicroSmart is "Pollution degree 2." Use the MicroSmart in environments of pollution degree 2 (according to IEC 60664-1).

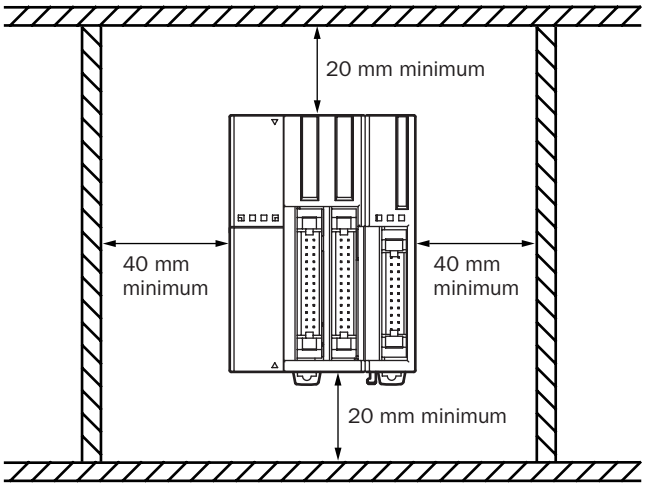
When installing the MicroSmart modules in a control panel, take the convenience of operation and maintenance, and resistance against environments into consideration.

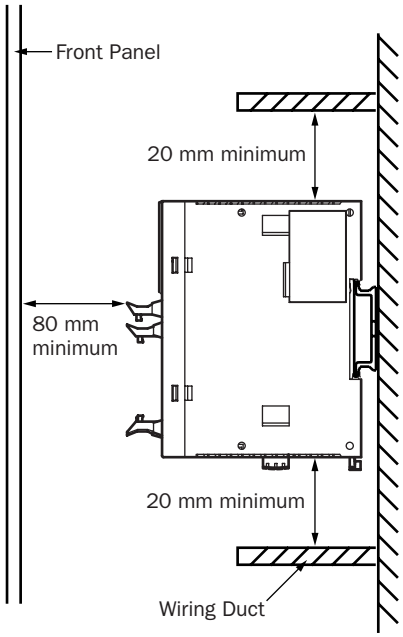
All-in-One Type CPU Module





Slim Type CPU Module



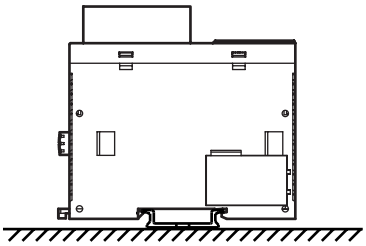


Mounting Direction

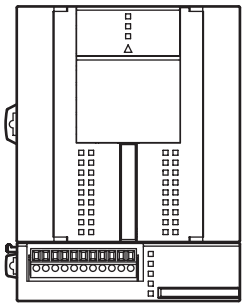
Mount the MicroSmart modules horizontally on a vertical plane as shown on the preceding page. Keep a sufficient spacing around the MicroSmart modules to ensure proper ventilation and keep the ambient temperature between 0°C and 55°C.

All-in-One Type CPU Module

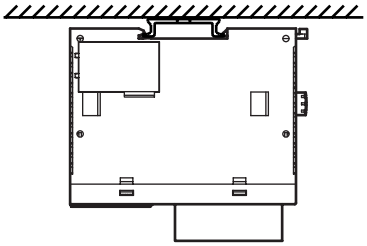
When the ambient temperature is 35°C or below, the all-in-one type CPU modules can also be mounted upright on a horizontal plane as shown at left below. When the ambient temperature is 40°C or below, the all-in-one type CPU modules can also be mounted sideways on a vertical plane as shown in the middle below.



Allowable Mounting Direction at 35°C or below



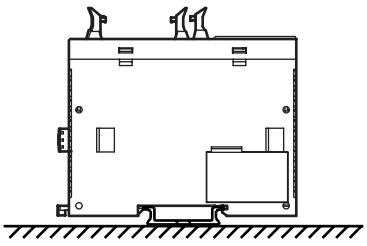
Allowable Mounting Direction at 40°C or below



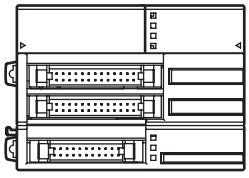
Incorrect Mounting Direction

Slim Type CPU Module

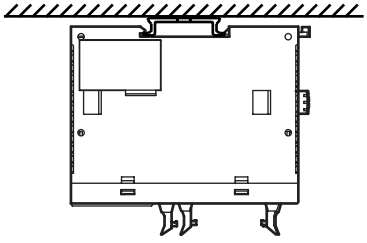
Always mount the slim type CPU modules horizontally on a vertical plane as shown on the preceding page. Any other mounting directions are not allowed.



Incorrect Mounting Direction



Incorrect Mounting Direction



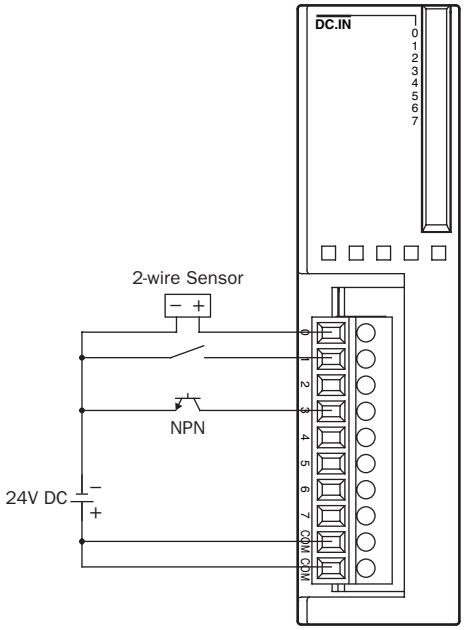
Incorrect Mounting Direction

Input Wiring

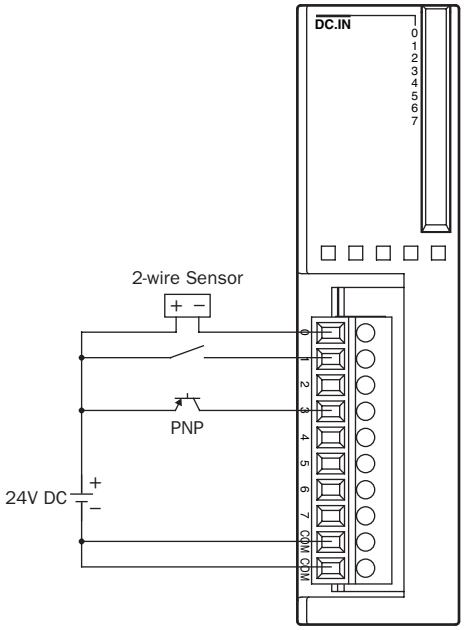
Caution

- Separate the input wiring from the output line, power line, and motor line.
- Use proper wires for input wiring.
  All-in-one type CPU modules: UL1015 AWG22 or UL1007 AWG18
  Slim type CPU and I/O modules: UL1015 AWG22

DC Source Input



DC Sink Input

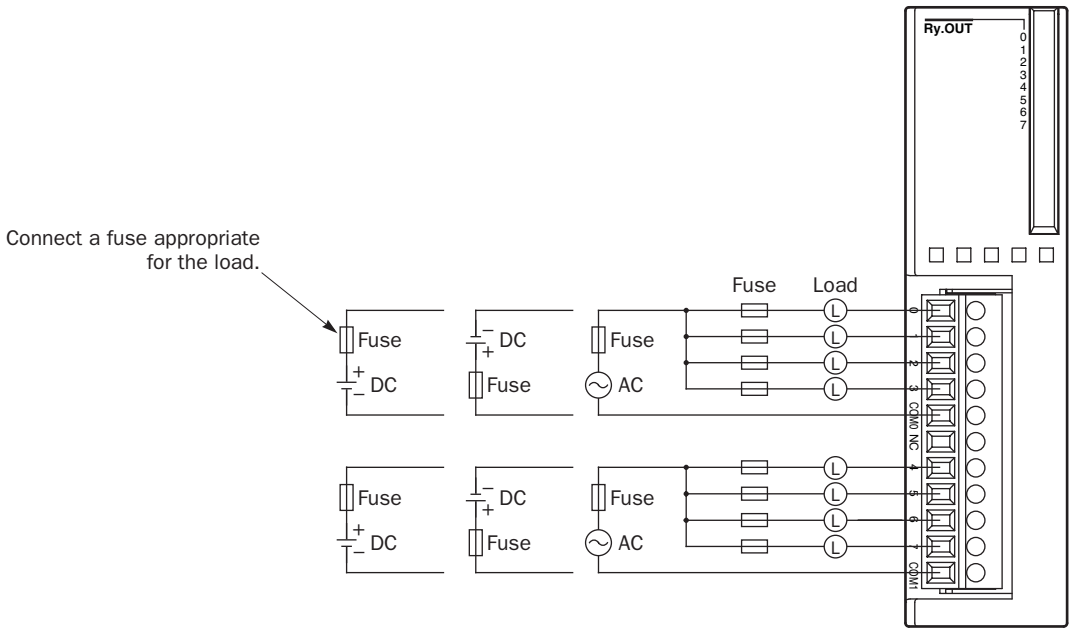


Output Wiring

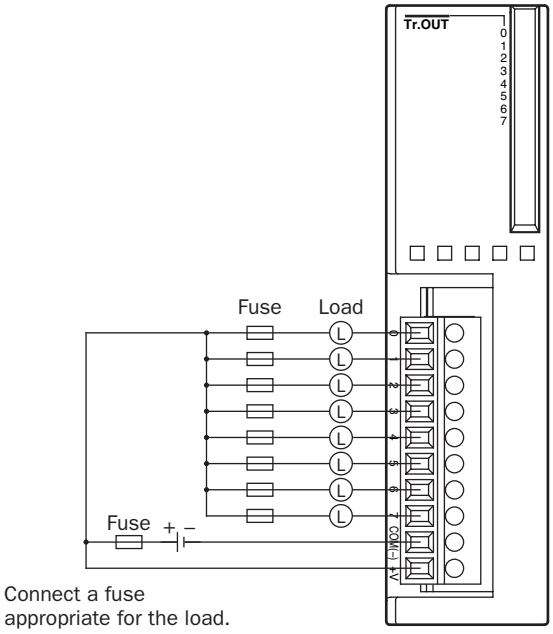
⚠ Caution

- If output relays or transistors in the MicroSmart CPU or output modules should fail, outputs may remain on or off. For output signals which may cause heavy accidents, provide a monitor circuit outside the MicroSmart.
- Connect a fuse to the output module, selecting a fuse appropriate for the load.
- Use proper wires for output wiring.
  All-in-one type CPU modules: UL1015 AWG22 or UL1007 AWG18
  Slim type CPU and I/O modules: UL1015 AWG22
- When equipment containing the MicroSmart is intended for use in European countries, insert an IEC 60127-approved fuse to each output of every module for protection against overload or short-circuit. This is required when equipment containing the MicroSmart is destined for Europe.

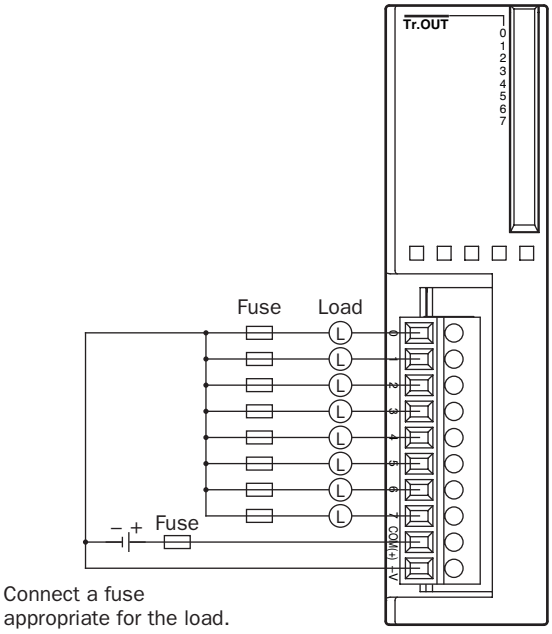
Relay Output



Transistor Sink Output



Transistor Source Output

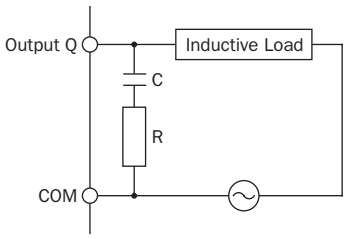


Contact Protection Circuit for Relay and Transistor Outputs

Depending on the load, a protection circuit may be needed for the relay output of the MicroSmart modules. Choose a protection circuit from A through D shown below according to the power supply and connect the protection circuit to the outside of the CPU or relay output module.

For protection of the transistor output of the MicroSmart modules, connect protection circuit C shown below to the transistor output circuit.

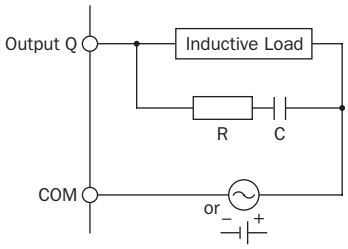
Protection Circuit A



This protection circuit can be used when the load impedance is smaller than the RC impedance in an AC load power circuit.

R: Resistor of approximately the same resistance value as the load
C: 0.1 to 1 μF

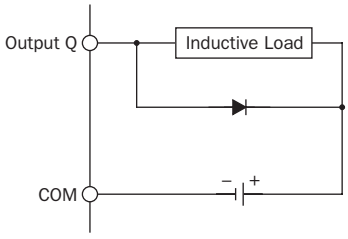
Protection Circuit B



This protection circuit can be used for both AC and DC load power circuits.

R: Resistor of approximately the same resistance value as the load
C: 0.1 to 1 μF

Protection Circuit C

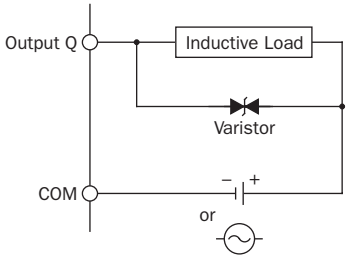


This protection circuit can be used for DC load power circuits.

Use a diode with the following ratings.

Reverse withstand voltage: Power voltage of the load circuit × 10
Forward current: More than the load current

Protection Circuit D



This protection circuit can be used for both AC and DC load power circuits.

Power Supply

All-in-One Type CPU Module (AC and DC Power)

⚠ Caution

- Use a power supply of the rated value. Use of a wrong power supply may cause fire hazard.
- The allowable power voltage range is 85 to 264V AC for the AC power type CPU module and 16.0 to 31.2V DC for the DC power type CPU module. Do not use the MicroSmart CPU module on any other voltage.
- On the AC power type CPU module, if the power voltage turns on or off very slowly between 15 and 50V AC, the CPU module may run and stop repeatedly between these voltages. If failure or disorder of the control system, damage, or accidents may be caused, provide a measure for prevention using a voltage monitoring circuit outside the MicroSmart.
- Use an IEC 60127-approved fuse on the power line outside the MicroSmart. This is required when equipment containing the MicroSmart is destined for Europe.

Power Supply Voltage

The allowable power voltage range for the all-in-one type MicroSmart CPU module is 85 to 264V AC for the AC power type and 16.0 to 31.2V DC for the DC power type.

Power failure detection voltage depends on the quantity of used input and output points. Basically, power failure is detected when the power voltage drops below 85V AC or 16.0V DC, stopping operation to prevent malfunction.

A momentary power interruption for 20 ms or less is not recognized as a power failure at the rated voltage of 100 to 240V AC or 24V DC.

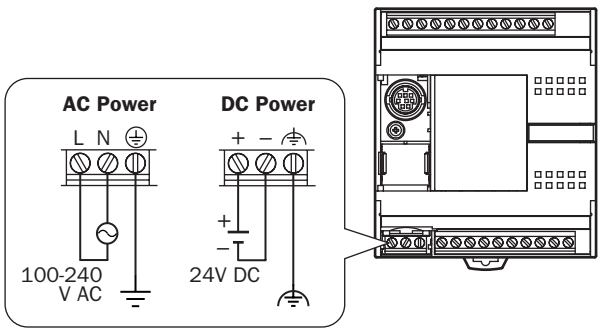
Inrush Current at Power-up

When the all-in-one AC or DC power type CPU module is powered up, an inrush current of a maximum of 35A (10- and 16-I/O type CPU modules) or 40A (24-I/O type CPU module) flows.

Power Supply Wiring

Use a stranded wire of UL1015 AWG22 or UL1007 AWG18 for power supply wiring. Make the power supply wiring as short as possible.

Run the power supply wiring as far away as possible from motor lines.



Grounding

To prevent electrical shocks, connect the ⏚ or ⏚ terminal to a proper ground using a wire of UL1007 AWG16. The grounding also prevents malfunctioning due to noise.

Do not connect the grounding wire in common with the grounding wire of motor equipment.

Slim Type CPU Module (DC Power)

⚠ Caution

- Use a power supply of the rated value. Use of a wrong power supply may cause fire hazard.
- The allowable power voltage range for the slim type MicroSmart CPU module is 20.4 to 26.4V DC. Do not use the MicroSmart on any other voltage.
- If the power voltage turns on or off very slowly between 10 to 15V DC, the MicroSmart may run and stop repeatedly between these voltages. If failure or disorder of the control system, damage, or accidents may be caused, provide a measure for prevention using a voltage monitoring circuit outside the MicroSmart.
- Use an IEC 60127-approved fuse on the power line outside the MicroSmart. This is required when equipment containing the MicroSmart is destined for Europe.

Power Supply Voltage

The allowable power voltage range for the slim type MicroSmart CPU module is 20.4 to 26.4V DC.

Power failure detection voltage depends on the quantity of used input and output points. Basically, power failure is detected when the power voltage drops below 20.4V DC, stopping operation to prevent malfunction.

A momentary power interruption for 10 ms or less is not recognized as a power failure at the rated voltage of 24V DC.

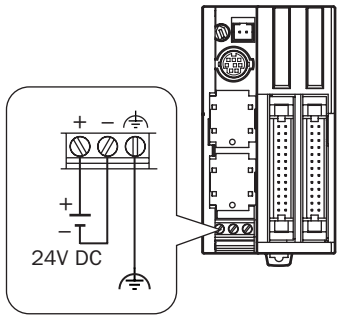
Inrush Current at Power-up

When the slim type CPU module is powered up, an inrush current of a maximum of 50A flows.

Power Supply Wiring

Use a stranded wire of UL1015 AWG22 or UL1007 AWG18 for power supply wiring. Make the power supply wiring as short as possible.

Run the power supply wiring as far away as possible from motor lines.



Grounding

To prevent electrical shocks, connect the ⏚ terminal to a proper ground using a wire of UL1015 AWG22 or UL1007 AWG18. The grounding also prevents malfunctioning due to noise.

Do not connect the grounding wire in common with the grounding wire of motor equipment.

Terminal Connection

⚠ Caution

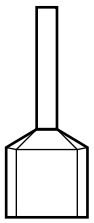
- Make sure that the operating conditions and environments are within the specification values.
- Be sure to connect the grounding wire to a proper ground, otherwise electrical shocks may be caused.
- Do not touch live terminals, otherwise electrical shocks may be caused.
- Do not touch terminals immediately after power is turned off, otherwise electrical shocks may be caused.
- When using ferrules, insert a wire to the bottom of the ferrule and crimp the ferrule.
- When connecting a stranded wire or multiple solid wires to a screw terminal block, use a ferrule. Otherwise the wire may slip off the screw terminal block.

Ferrules, Crimping Tool, and Screwdriver for Phoenix Terminal Blocks

The screw terminal block can be wired with or without using ferrules on the end of cable. Applicable ferrules for the Phoenix terminal blocks and crimping tool for the ferrules are listed below. The screwdriver is used for tightening the screw terminals on the MicroSmart modules. These ferrules, crimping tool, and screwdriver are made by Phoenix Contact and are available from Phoenix Contact.

Type numbers of the ferrules, crimping tool, and screwdriver listed below are the type numbers of Phoenix Contact. When ordering these products from Phoenix Contact, specify the Order No. and quantity listed below.

Ferrule Order No.



Quantity of Cables	Cable Size	Phoenix Type	Order No.	Pcs./Pkt.
For 1-cable connection	UL1007 AWG16	AI 1,5-8 BK	32 00 04 3	100
	UL1007 AWG18	AI 1-8 RD	32 00 03 0	100
	UL1015 AWG22	AI 0,5-8 WH	32 00 01 4	100
For 2-cable connection	UL1007 AWG18	AI-TWIN 2 x 0,75-8 GY	32 00 80 7	100
	UL1015 AWG22	AI-TWIN 2 x 0,5-8 WH	32 00 93 3	100

Crimping Tool and Screwdriver Order No.

Tool Name		Phoenix Type	Order No.	Pcs./Pkt.
Crimping Tool		CRIMPFOX ZA 3	12 01 88 2	1
Screwdriver	For CPU modules	SZS 0,6 x 3,5	12 05 05 3	10
	For I/O modules and communication adapter	SZS 0,4 x 2,5	12 05 03 7	10

Screw Terminal Tightening Torque	CPU modules	0.5 N·m
	I/O modules Communication adapter	0.22 to 0.25 N·m

4: OPERATION BASICS

Introduction

This chapter describes general information about setting up the basic MicroSmart system for programming, starting and stopping MicroSmart operation, and introduces simple operating procedures from creating a user program using WindLDR on a computer to monitoring the MicroSmart operation.

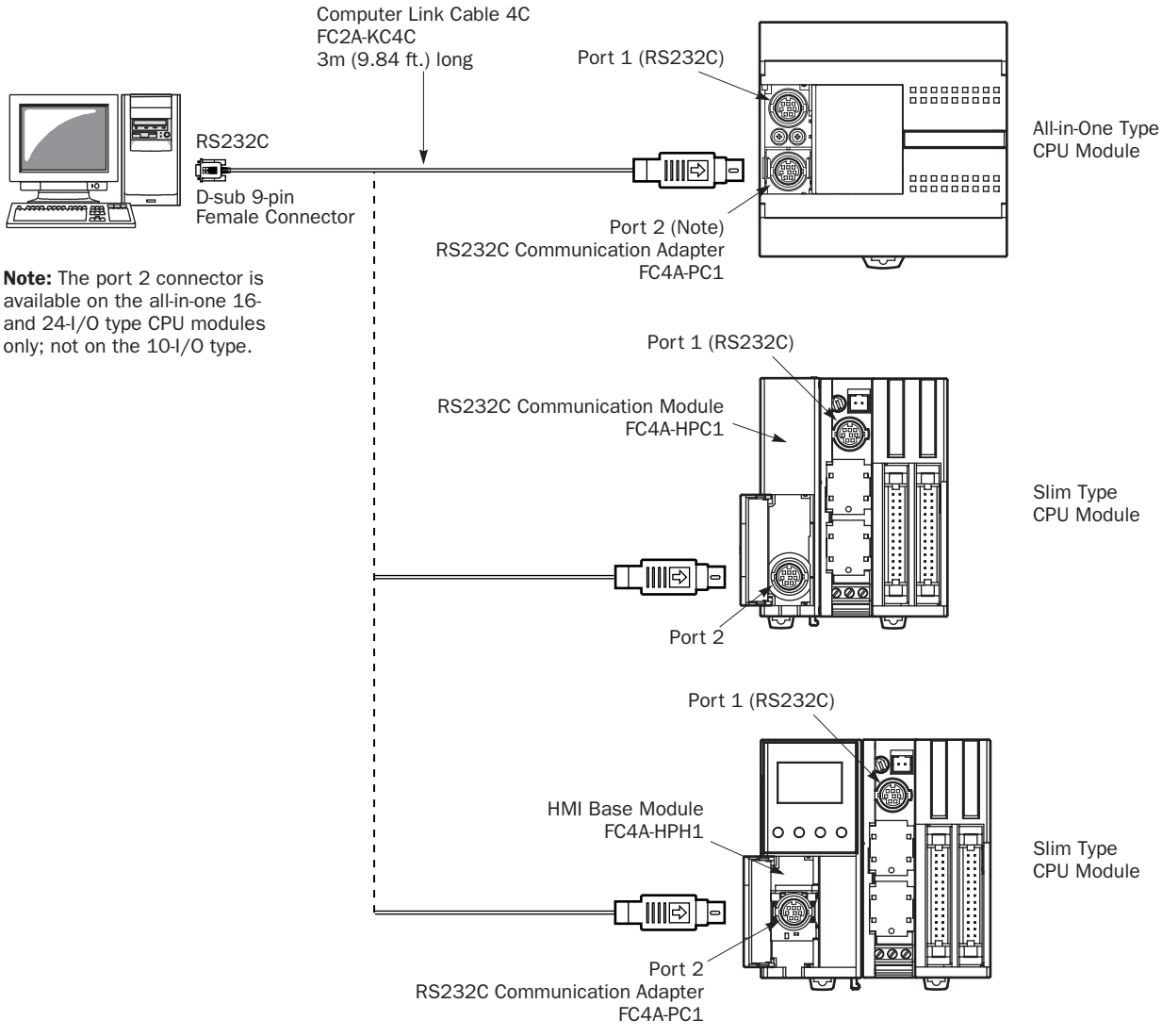
Connecting MicroSmart to PC (1:1 Computer Link System)

The MicroSmart can be connected to a Windows PC in two ways.

Computer Link through Port 1 or Port 2 (RS232C)

When connecting a Windows computer to the RS232C port 1 or port 2 on the MicroSmart CPU module, enable the maintenance protocol for the RS232C port using the Function Area Settings in WindLDR. See page 26-2.

To set up a 1:1 computer link system, connect a computer to the CPU module using the computer link cable 4C (FC2A-KC4C). The computer link cable 4C can be connected to port 1 directly. When connecting the cable to port 2 on the all-in-one 16- or 24-I/O type CPU module, install an optional RS232C communication adapter (FC4A-PC1) to the port 2 connector. When connecting to port 2 on the slim type CPU module, an optional RS232C communication module (FC4A-HPC1) is needed. The RS232C communication adapter can also be installed on the HMI base module (FC4A-HPH1).

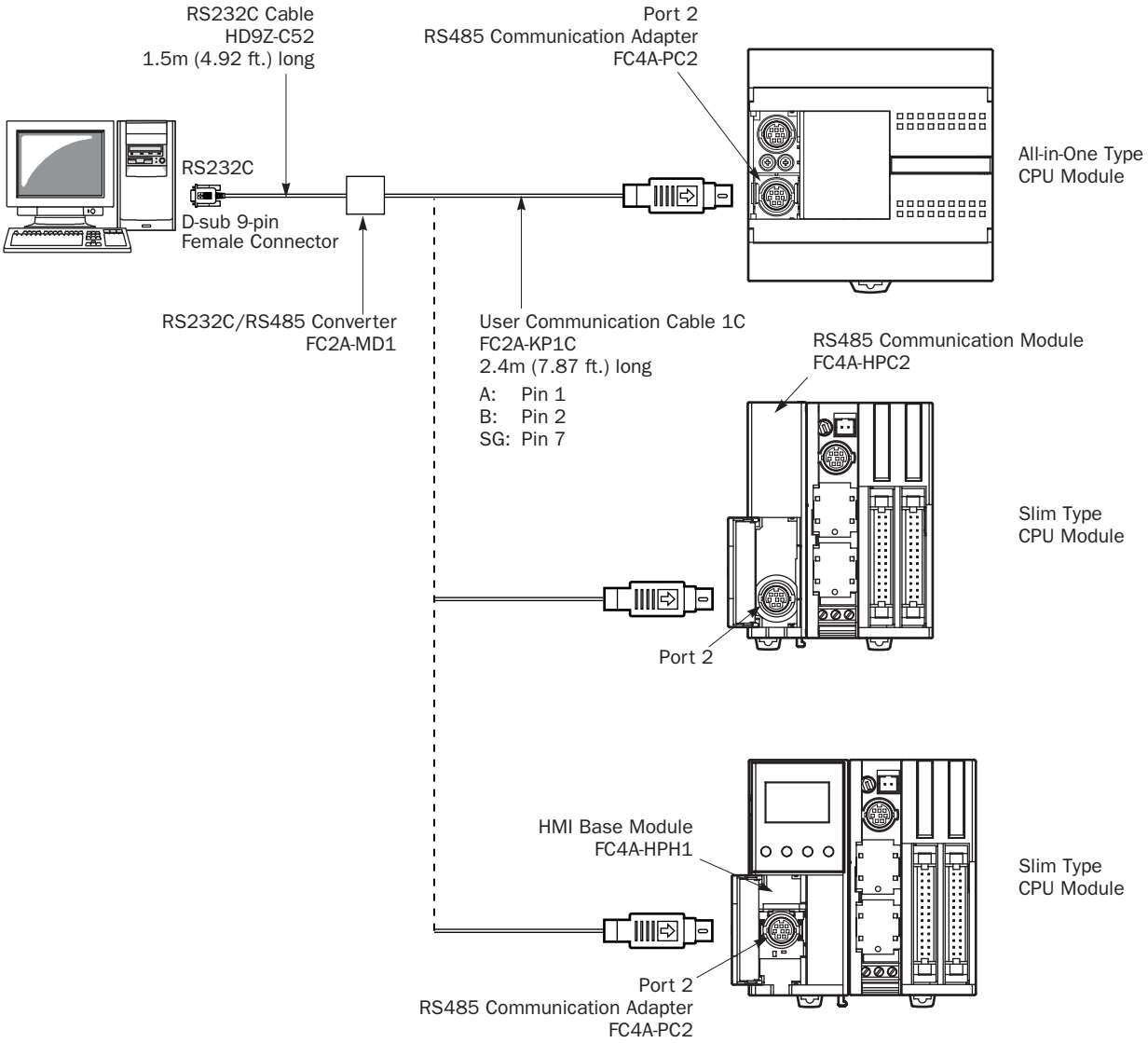


Computer Link through Port 2 (RS485)

When connecting a Windows computer to port 2 on the all-in-one 16- or 24-I/O type CPU module or slim type CPU module, enable the maintenance protocol for port 2 using the Function Area Settings in WindLDR. See page 26-2.

To set up a 1:1 computer link system using the all-in-one 16- or 24-I/O type CPU module, install an optional RS485 communication adapter (FC4A-PC2) to the port 2 connector. Connect a computer to the RS232C/RS485 converter (FC2A-MD1) using the RS232C cable (HD9Z-C52). Connect the RS232C/RS485 converter to the CPU module using the user communication cable 1C (FC2A-KP1C). The RS232C/RS485 converter is powered by an 24V DC source or an AC adapter with 9V DC output. For details about the RS232C/RS485 converter, see page 26-4.

To set up a 1:1 computer link system using the slim type CPU module, an optional RS485 communication module (FC4A-HPC2) is needed. The RS485 communication adapter can also be installed on the HMI base module (FC4A-HPH1).



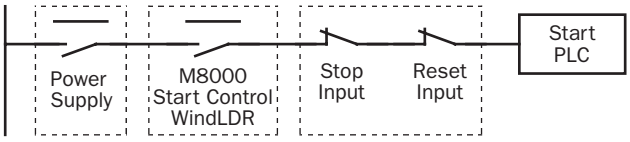
Start/Stop Operation

This section describes operations to start and stop the MicroSmart and to use the stop and reset inputs.

Caution • Make sure of safety before starting and stopping the MicroSmart. Incorrect operation on the MicroSmart may cause machine damage or accidents.

Start/Stop Schematic

The start/stop circuit of the MicroSmart consists of three blocks; power supply, M8000 (start control special internal relay), and stop/reset inputs. Each block can be used to start and stop the MicroSmart while the other two blocks are set to run the MicroSmart.



Start/Stop Operation Using WindLDR

The MicroSmart can be started and stopped using WindLDR run on a Windows PC connected to the MicroSmart CPU module. When the **PLC Start** button is pressed in the dialog box shown below, start control special internal relay M8000 is turned on to start the MicroSmart. When the **PLC Stop** button is pressed, M8000 is turned off to stop the MicroSmart.

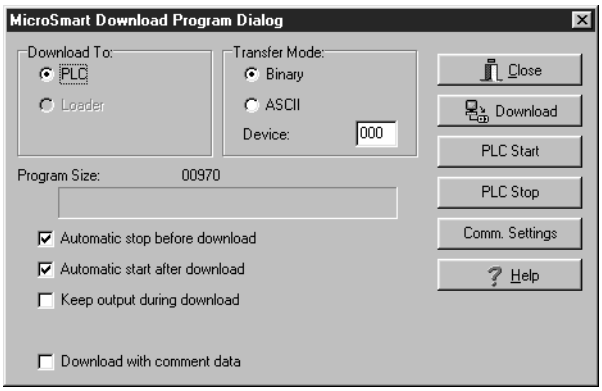
1. Connect the PC to the MicroSmart, start WindLDR, and power up the MicroSmart. See page 4-1.

2. Check that a stop input is not designated using **Configure > Function Area Settings > Run/Stop**. See page 5-2.

Note: When a stop input is designated, the MicroSmart cannot be started or stopped by turning start control special internal relay M8000 on or off.

3. Select **Online** from the WindLDR menu bar, then select **Download Program**. Or, click the download icon .

The Download Program dialog box appears.



4. Click the **PLC Start** button to start operation, then the start control special internal relay M8000 is turned on.

5. Click the **PLC Stop** button to stop operation, then the start control special internal relay M8000 is turned off.

The PLC operation can also be started and stopped while WindLDR is in the monitor mode. To access the **Start** or **Stop** button, select **Online > Monitor** and select **Online > PLC Status > Run/Stop Status**.

Note: Special internal relay M8000 is a keep type internal relay and stores the status when power is turned off. M8000 retains its previous status when power is turned on again. However, when the backup battery is dead, M8000 loses the stored status, and can be turned on or off as programmed when the MicroSmart is powered up. The selection is made in **Configure** > **Function Area Settings** > **Run/Stop** > **Run/Stop Selection at Memory Backup Error**. See page 5-3.

The backup duration is approximately 30 days (typical) at 25°C after the backup battery is fully charged.

Start/Stop Operation Using the Power Supply

The MicroSmart can be started and stopped by turning power on and off.

1. Power up the MicroSmart to start operation. See page 4-1.
2. If the MicroSmart does not start, check that start control special internal relay M8000 is on using WindLDR. If M8000 is off, turn it on. See page 4-3.
3. Turn power on and off to start and stop operation.

Note: If M8000 is off, the MicroSmart does not start operation when power is turned on. To start operation, turn power on, and turn M8000 on by clicking the **Start** button in WindLDR.

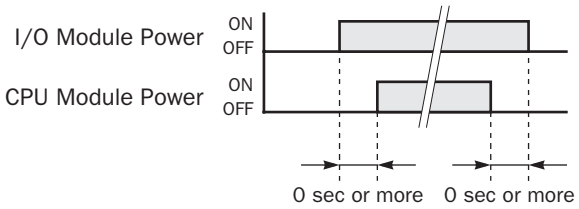
The response time of the MicroSmart at powerup depends on such factors as the contents of the user program, data link usage, and system setup. The table below shows an approximate time delay before starting operation after powerup.

Response time when no data link is used:

Program Size	After powerup, the CPU starts operation in
4,800 bytes (800 steps)	Approx. 0.5 second
15,000 bytes (2,500 steps)	Approx. 1.2 seconds
27,000 bytes (4,500 steps)	Approx. 2 seconds
64,500 bytes (10,750 steps)	Approx. 5 seconds

Order of Powerup and Powerdown

To ensure I/O data transfer, power up the I/O modules first, followed by the CPU module, or power up the CPU and I/O modules at the same time. When shutting down the system, power down the CPU first, followed by I/O modules, or power down the CPU and I/O modules at the same time.



Start/Stop Operation Using Stop Input and Reset Input

Any input terminal available on the CPU module can be designated as a stop or reset input using the Function Area Settings. The procedure for selecting stop and reset inputs is described on page 5-2.

Note: When using a stop and/or reset input to start and stop operation, make sure that start control special internal relay M8000 is on. If M8000 is off, the CPU does not start operation when the stop or reset input is turned off. M8000 is not turned on or off when the stop and/or reset input is turned on or off.

When a stop or reset input is turned on during program operation, the CPU stops operation, the RUN LED is turned off, and all outputs are turned off.

The reset input has priority over the stop input.

System Statuses at Stop, Reset, and Restart

The system statuses during running, stop, reset, and restart after stopping are listed below:

Mode	Output	Internal Relay, Shift Register, Counter, Data Register, Expansion Data Register		Timer Current Value
		Keep Type	Clear Type	
Run	Operating	Operating	Operating	Operating
Stop (Stop input ON)	OFF	Unchanged	Unchanged	Unchanged
Reset (Reset input ON)	OFF	OFF/Reset to zero	OFF/Reset to zero	Reset to zero
Restart	Unchanged	Unchanged	OFF/Reset to zero	Reset to preset

Note: Expansion data registers and AS-Interface operands are available on slim type CPU modules FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3. All expansion data registers are keep types. AS-Interface operands (M1300-M1977 and D1700-D1999) remain unchanged when the reset input is turned on.

Simple Operation

This section describes how to edit a simple program using WindLDR on a computer, transfer the program from the computer to the MicroSmart, run the program, and monitor the operation on the WindLDR screen.

Connect the MicroSmart to the computer as described on page 4-1.

Sample User Program

Create a simple program using WindLDR. The sample program performs the following operation:

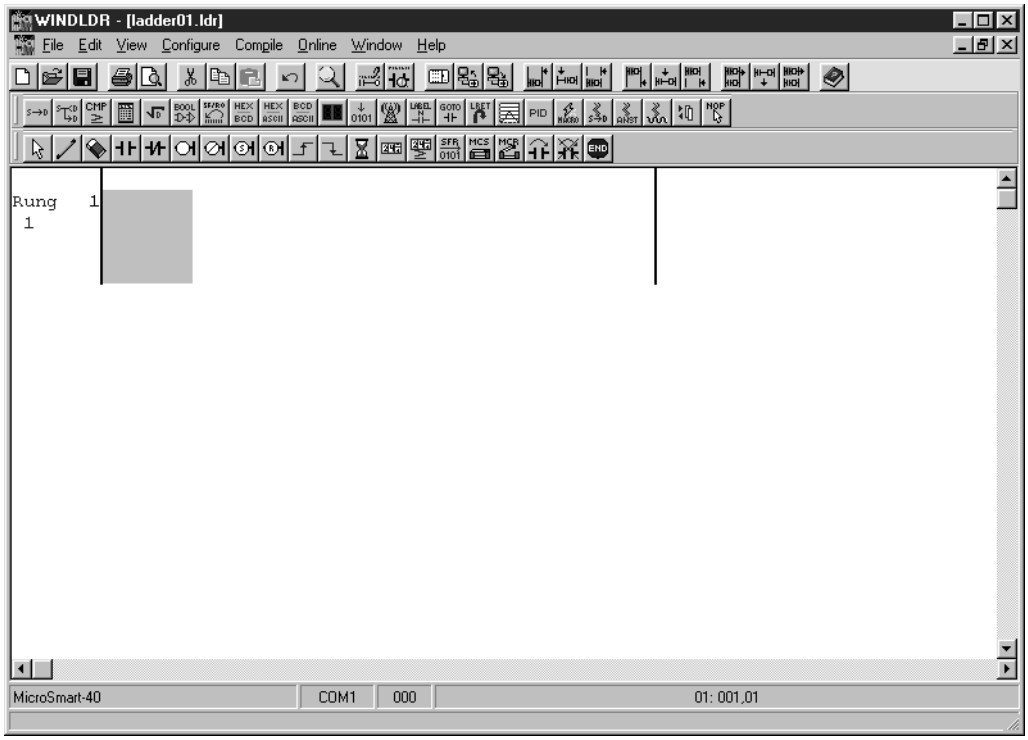
When only input I0 is turned on, output Q0 is turned on.
When only input I1 is turned on, output Q1 is turned on.
When both inputs I0 and I1 are turned on, output Q2 flashes in 1-sec increments.

Rung No.	Input I0	Input I1	Output Operation
1	ON	OFF	Output Q0 is turned ON.
2	OFF	ON	Output Q1 is turned ON.
3	ON	ON	Output Q2 flashes in 1-sec increments.

Start WindLDR

From the Start menu of Windows, select **Programs > WindLDR > WindLDR**.

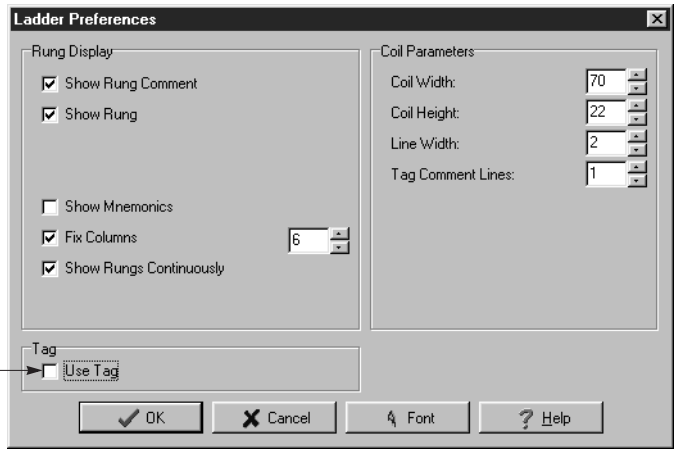
WindLDR starts and a blank ladder editing screen appears with menus and tool bars shown on top of the screen.



Disable Tag Function

The following example describes a simple procedure without using the tag function.

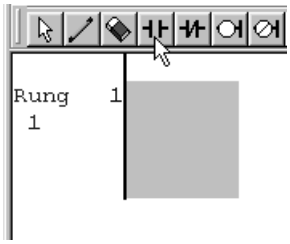
From the WindLDR menu bar, select **Configure > Ladder Preference**. The Ladder Preference dialog box appears, then uncheck the check box under Tag to disuse the tag function. Click **OK** to close the dialog box.



Uncheck the Use Tag check box.

Edit User Program Rung by Rung

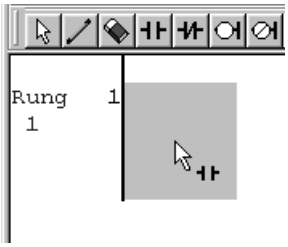
Start the user program with the LOD instruction by inserting a NO contact of input I0.



1. Click the **Normally Open** contact icon .

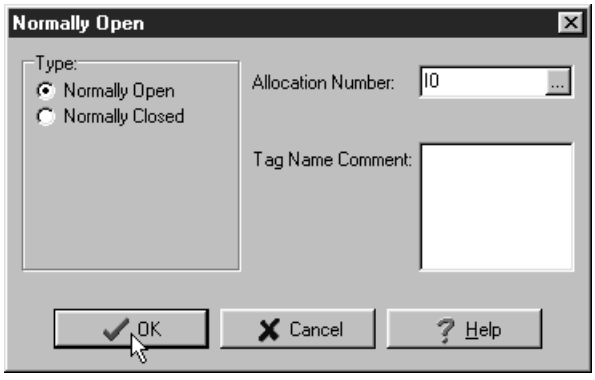
2. Move the mouse pointer to the first column of the first line where you want to insert a NO contact, and click the left mouse button.

   Note: Another method to insert a NO (or NC) contact is to move the mouse pointer where you want to insert the contact, and type A (or B).



The Normally Open dialog box appears.

3. Enter **I0** in the Allocation Number field, and click **OK**.



A NO contact of input I0 is programmed in the first column of the first ladder line.

Next, program the ANDN instruction by inserting a NC contact of input I1.

4. Click the **Normally Closed** contact icon .

5. Move the mouse pointer to the second column of the first ladder line where you want to insert a NC contact, and click the left mouse button.

The Normally Closed dialog box appears.

6. Enter **I1** in the Allocation Number field, and click **OK**.

A NC contact of input I1 is programmed in the second column of the first ladder line.

At the end of the first ladder line, program the OUT instruction by inserting a NO coil of output Q0.

7. Click the **Output** coil icon .

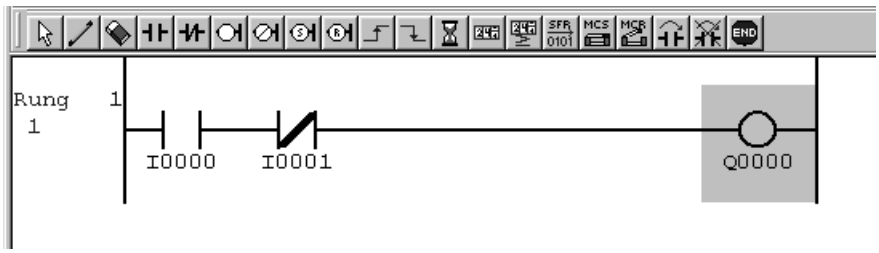
8. Move the mouse pointer to the third column of the first ladder line where you want to insert an output coil, and click the left mouse button.

Note: Another method to insert an instruction (either basic or advanced) is to type the instruction symbol, OUT, where you want to insert the instruction.

The Output dialog box appears.

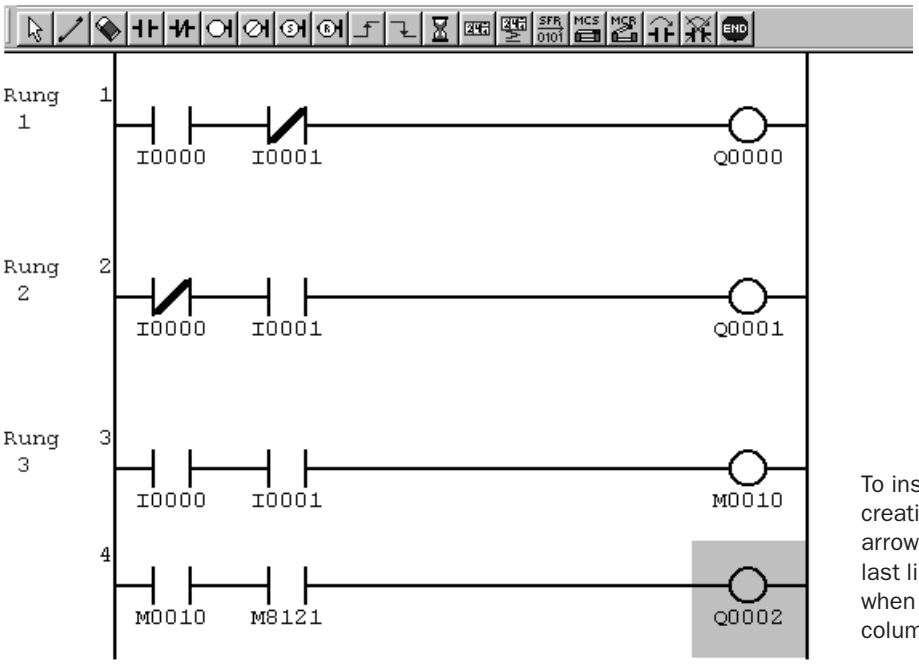
9. Enter **Q0** in the Allocation Number field, and click **OK**.

A NO output coil of output Q0 is programmed in the right-most column of the first ladder line. This completes programming for rung 1.



Continue programming for rungs 2 and 3 by repeating similar procedures.

A new rung is inserted by pressing the **Enter** key while the cursor is on the preceding rung. A new rung can also be inserted by selecting **Edit > Append > Rung**. When completed, the ladder program looks like below.



To insert a new ladder line without creating a new rung, press the down arrow key when the cursor is on the last line or press the right arrow key when the cursor is at the right-most column of the last line.

Now, save the file with a new name.

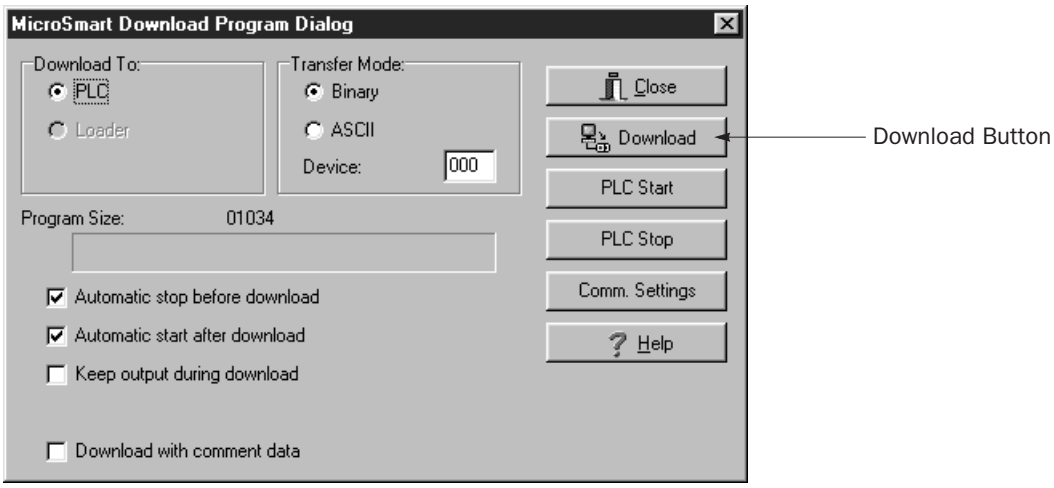
10. From the menu bar, select **File > Save As** and type **TEST01.LDR** in the File Name field. Change the Folder or Drive as necessary.

Click **OK**, and the file is saved in the selected folder and drive.

Download Program

You can download the user program from WindLDR running on a computer to the MicroSmart.

From the WindLDR menu bar, select **Online > Download Program**. The Download Program Dialog appears, then click the **Download** button. The user program is downloaded to the MicroSmart.



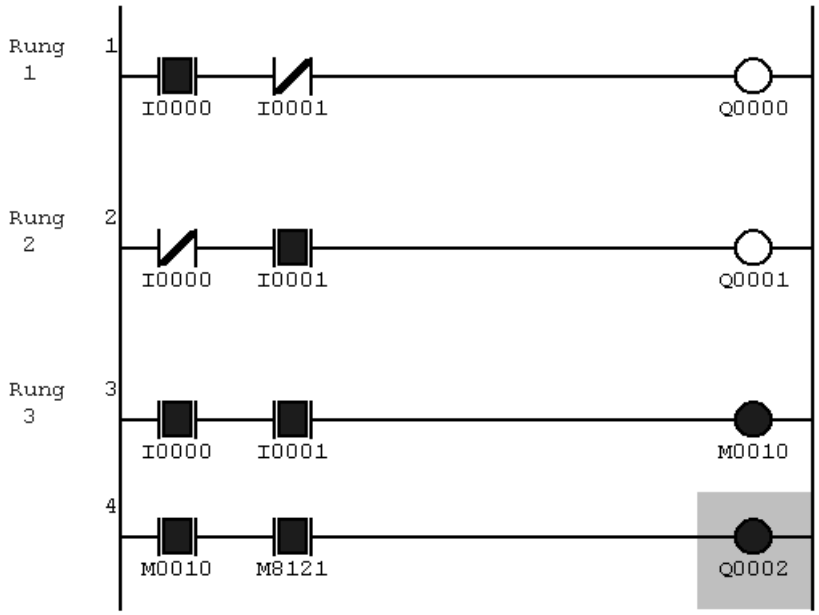
Note: When downloading a user program, all values and selections in the Function Area Settings are also downloaded to the MicroSmart. For Function Area Settings, see pages 5-1 through 5-25.

Monitor Operation

Another powerful function of WindLDR is to monitor the PLC operation on the computer. The input and output statuses of the sample program can be monitored in the ladder diagram.

From the WindLDR menu bar, select **Online > Monitor**.

When both inputs I0 and I1 are on, the ladder diagram on the monitor screen looks as follows:



Rung 1: When both inputs I0 and I1 are on, output Q0 is turned off.

Rung 2: When both inputs I0 and I1 are on, output Q1 is turned off.

Rung 3: When both input I0 and I1 are on, internal relay M10 is turned on.

M8121 is the 1-sec clock special internal relay.

While M10 is on, output Q2 flashes in 1-sec increments.

Quit WindLDR

When you have completed monitoring, you can quit WindLDR either directly from the monitor screen or from the editing screen. In both cases, from the menu bar select **File > Exit WindLDR**.

5: SPECIAL FUNCTIONS

Introduction

The MicroSmart features special functions such as stop/reset inputs, run/stop selection at memory backup error, keep designation for internal relays, shift registers, counters, and data registers. These functions are programmed using the Function Area Settings menu. Also included in the Function Area Settings are high-speed counter, catch input, interrupt input, communication protocol selection for port 1 and port 2, input filter, and user program read/write protection.

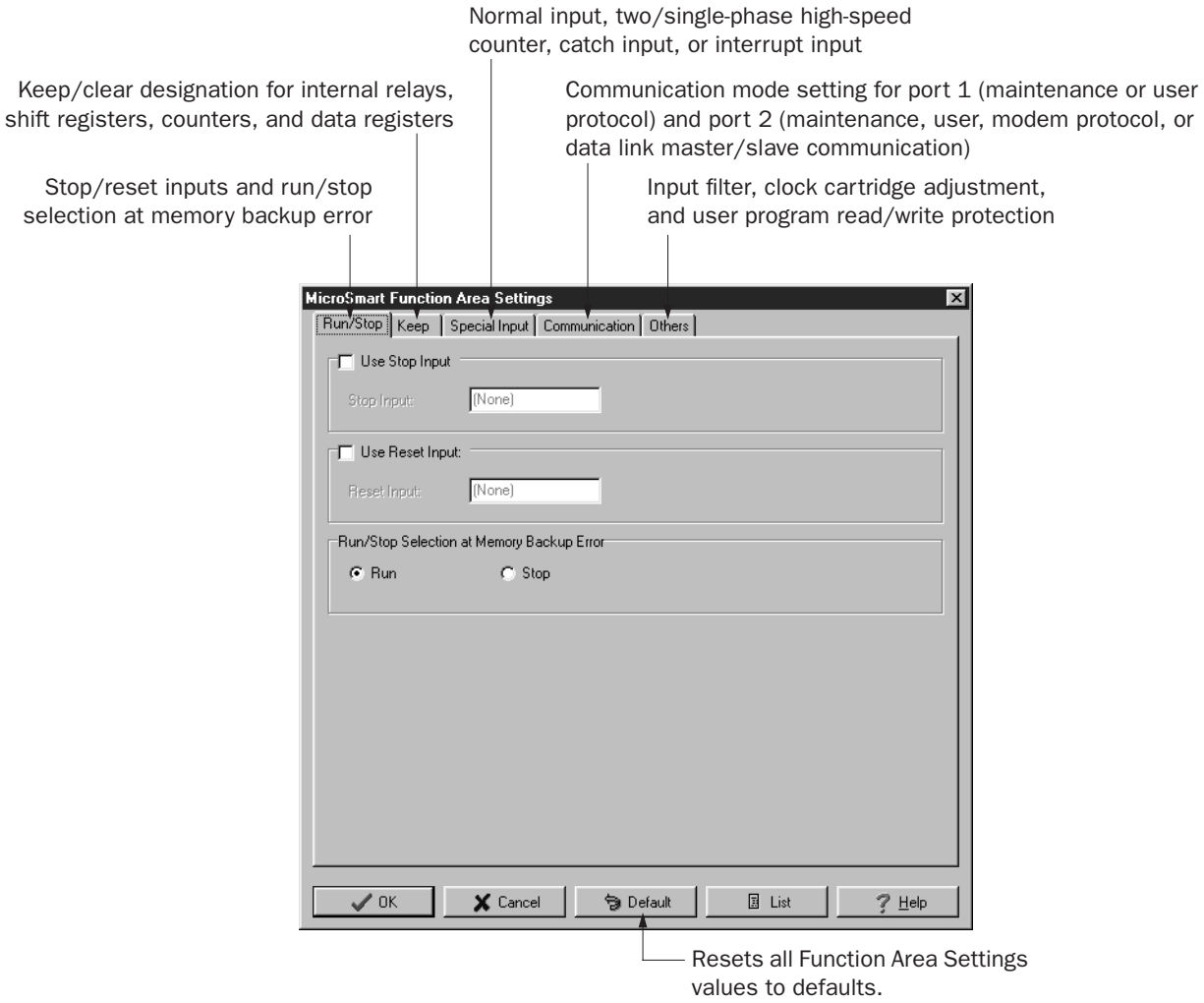
This chapter describes these special functions. Clock function, analog potentiometer function, memory cartridge, and constant scan features are also described in this chapter.

The Function Area Settings for communication functions are detailed in chapters 17 and 25 through 27.

> **⚠ Caution** • Since all Function Area Settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Function Area Settings

Various special functions are programmed in the Function Area Settings. To call the Function Area Settings dialog box, start WindLDR on a Windows PC. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.



Detailed information is described on the following pages.

Stop Input and Reset Input

As described on page 4-3, the MicroSmart can be started and stopped using a stop input or reset input, which can be designated from the Function Area Settings menu. When the designated stop or reset input is turned on, the MicroSmart stops operation. For the system statuses in the stop and reset modes, see page 4-4.

Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

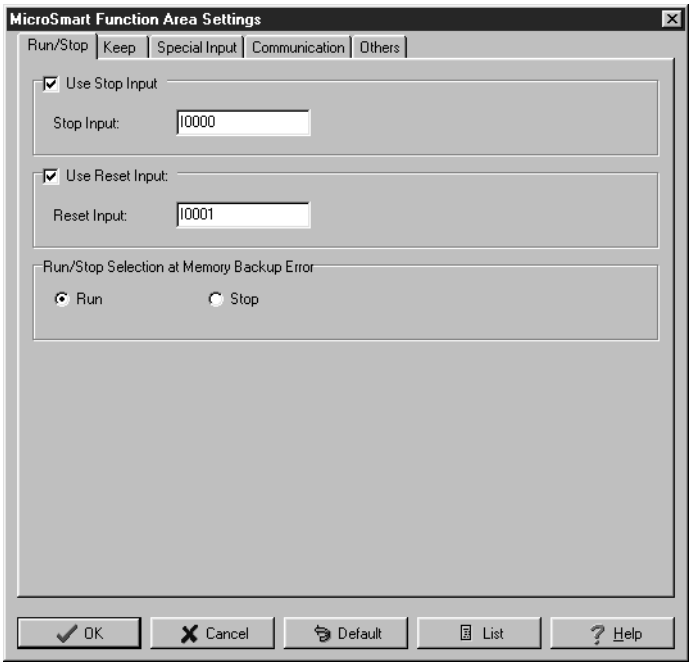
Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Run/Stop** tab.

Stop Input: Click the check box on the left of Use Stop Input and type a desired input number available on the CPU module in the Stop Input field.

Reset Input: Click the check box on the left of Use Reset Input and type a desired reset number available on the CPU module in the Reset Input field.

This example designates input I0 as a stop input and input I1 as a reset input.



Default: No stop and reset inputs are designated.

3. Click the **OK** button.

Run/Stop Selection at Memory Backup Error

Start control special internal relay M8000 maintains its status when the CPU is powered down. After the CPU has been off for a period longer than the battery backup duration, the data designated to be maintained during power failure is broken. The Run/Stop Selection at Memory Backup Error dialog box is used to select whether to start or stop the CPU when attempting to restart operation after the "keep" data in the CPU RAM has been lost.

When a built-in lithium battery is fully charged, data of internal relays, shift registers, counters, and data registers stored in the RAM are maintained for approximately 30 days.

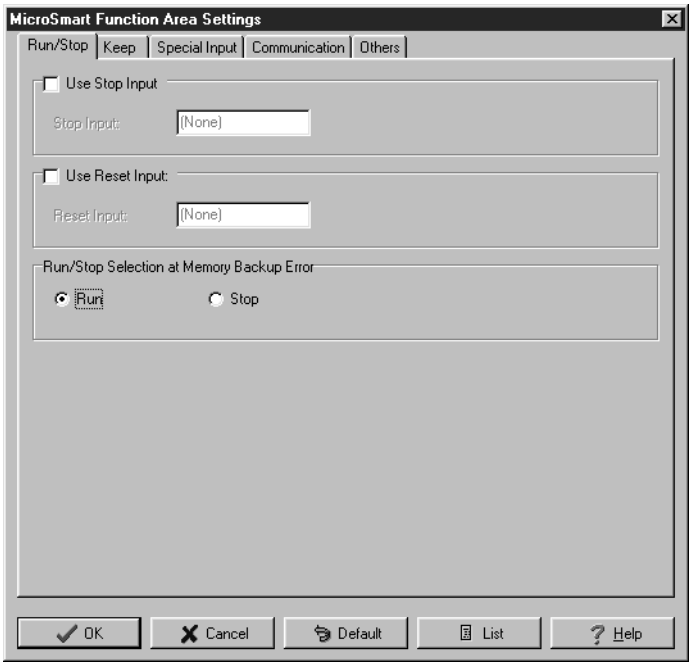
Since this setting relates to the user program, the user program must be downloaded to the MicroSmart after changing this setting.

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Run/Stop** tab.

Run (Default): Click the button on the left to start the CPU at memory backup error.

Stop: Click the button on the right to stop the CPU when attempting to start at memory backup error. When the CPU does not start because of the Stop selection, the CPU can not be started alone, then the CPU can still be started by sending a start command from WindLDR to turn on start control special internal relay M8000. For start/stop operation, see page 4-3.



3. Click the **OK** button.

Keep Designation for Internal Relays, Shift Registers, Counters, and Data Registers

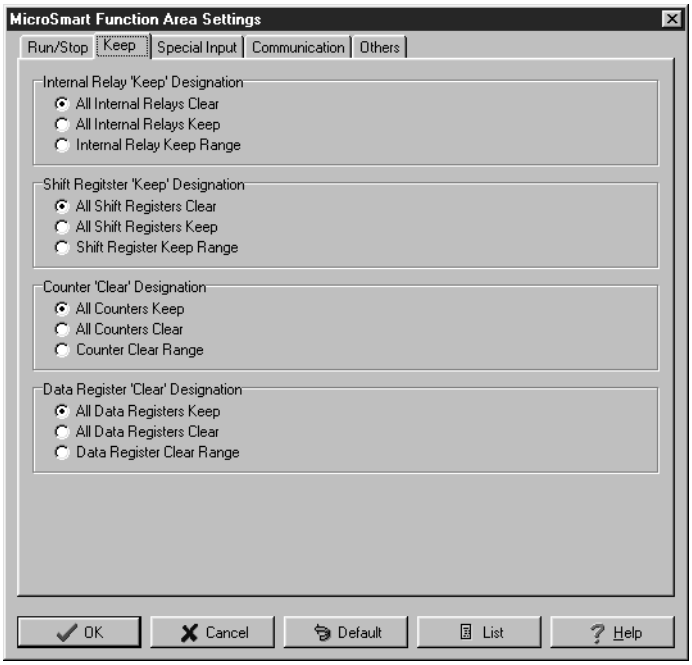
The statuses of internal relays and shift register bits are usually cleared at startup. It is also possible to designate all or a block of consecutive internal relays or shift register bits as “keep” types. Counter current values and data register values are usually maintained at powerup. It is also possible to designate all or a block of consecutive counters and data registers as “clear” types.

When the CPU is stopped, these statuses and values are maintained. When the CPU is reset by turning on a designated reset input, these statues and values are cleared despite the settings in the Keep dialog box shown below. The keep/clear settings in this dialog box have effect when restarting the CPU.

Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Keep** tab. The Keep page appears.



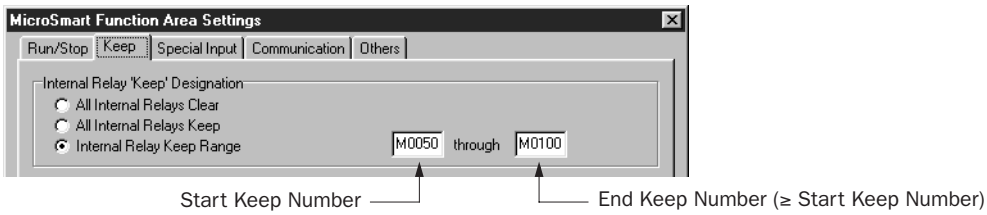
Internal Relay 'Keep' Designation

All Internal Relays Clear: All internal relay statuses are cleared at startup (default).

All Internal Relays Keep: All internal relay statuses are maintained at startup.

Internal Relay Keep Range: A designated area of internal relays are maintained at startup. Enter the start "keep" number in the left field and the end "keep" number in the right field. The start "keep" number must be smaller than or equal to the end "keep" number.

Valid internal relay numbers are M0 through M317 (FC4A-C10R2 and FC4A-C10R2C CPU modules) or M0 through M1277 (other CPU modules). Special internal relays and AS-Interface internal relays cannot be designated.



When a range of M50 through M100 is designated as shown in the example above, M50 through M100 are keep types, M0 through M47 and M101 through M1277 are clear types.

Shift Register 'Keep' Designation

All Shift Registers Clear: All shift register bit statuses are cleared at startup (default).

All Shift Registers Keep: All shift register bit statuses are maintained at startup.

Shift Register Keep Range: A designated area of shift register bits are maintained at startup. Enter the start "keep" number in the left field and the end "keep" number in the right field. The start "keep" number must be smaller than or equal to the end "keep" number.

Valid shift register bit numbers are R0 through R63 (FC4A-C10R2 and FC4A-C10R2C CPU modules) or R0 through R127 (other CPU modules).

When a range of R17 through R32 is designated, R17 through R32 are keep types, R0 through R16 and R33 through R127 are clear types.

Counter 'Clear' Designation

All Counters Keep: All counter current values are maintained at startup (default).

All Counters Clear: All counter current values are cleared at startup.

Counter Clear Range: A designated area of counter current values are cleared at startup. Enter the start "clear" number in the left field and the end "clear" number in the right field. The start "clear" number must be smaller than or equal to the end "clear" number.

Valid counter numbers are C0 through C31 (FC4A-C10R2 and FC4A-C10R2C CPU modules) or C0 through C99 (other CPU modules).

When a range of C0 through C10 is designated, C0 through C10 are clear types, and C11 through C99 are keep types.

Data Register 'Clear' Designation

All Data Registers Keep: All data register values are maintained at startup (default).

All Data Registers Clear: All data register values are cleared at startup.

Data Register Clear Range: A designated area of data register values are cleared at startup. Enter the start "clear" number in the left field and the end "clear" number in the right field. The start "clear" number must be smaller than or equal to the end "clear" number.

Valid data register numbers are D0 through D399 (FC4A-C10R2 and FC4A-C10R2C CPU modules) or D0 through D1299 (others). Special data registers, expansion data registers, and AS-Interface data registers cannot be designated. All expansion data registers are keep types.

When a range of D100 through D1299 is designated, D0 through D99 are keep types, and D100 through D1299 are clear types.

High-speed Counter

This section describes the high-speed counter function to count many pulse inputs within one scan. Using the built-in 16-bit high-speed counter, the MicroSmart counts up to 65535 high-speed pulses from a rotary encoder or proximity switch without regard to the scan time, compares the current value with a preset value, and turns on the output when the current value reaches the preset value. This function can be used for simple motor control or to measure lengths of objects.

The all-in-one type CPU modules and slim type CPU modules have different high-speed counter configurations.

High-speed Counters on All-in-One Type CPU Modules

All-in-one type CPU modules have four high-speed counters; HSC1 through HSC4. HSC1 can be used as a two-phase or single-phase high-speed counter. HSC2 through HSC4 are single-phase high-speed counters. All high-speed counter functions are selected using the Function Area Settings in WindLDR.

High-speed Counter Operation Modes and Input Terminals (All-in-One Type CPU Modules)

High-speed Counter No.	HSC1			HSC2	HSC3	HSC4
Input Terminal	I0	I1	I2	I3	I4	I5
Two-phase High-speed Counter	Phase A	Phase B	Reset Input (Phase Z)	—	—	—
Single-phase High-speed Counter	—	Pulse Input	Reset Input	Pulse Input	Pulse Input	Pulse Input

For wiring high-speed counter input signals, use twisted-pair shielded cables.

Two-phase High-speed Counter HSC1 (All-in-One Type CPU Modules)

Two-phase high-speed counter HSC1 operates in the rotary encoder mode, and counts up or down input pulses to input terminals I0 (phase A) and I1 (phase B). When the current value overflows 65535 or underflows 0, a designated comparison output turns on. Any output terminal available on the CPU module can be designated as a comparison output. When input I2 (reset input) is turned on, the current value is reset to a predetermined reset value, and the two-phase high-speed counter counts subsequent input pulses starting at the reset value.

Two special data registers and six special internal relays are assigned to control and monitor the two-phase high-speed counter operation. The current value is stored in data register D8045 (current value) and is updated every scan. The value stored in D8046 (reset value) is used as a reset value. When a high-speed counter reset input (I2 or M8032) is turned on, the current value in D8045 is reset to the value stored in D8046.

The two-phase high-speed counter is enabled while gate input special internal relay M8031 is on and is disabled while M8031 is off. When current value overflow or underflow occurs while counting up or down, special internal relay M8131 or M8132 turns on in the next scan, respectively. At this point, the D8045 current value is reset to the D8046 reset value for the subsequent counting cycle. When comparison output reset special internal relay M8030 is turned on, the designated comparison output is turned off. When reset input I2 is turned on to reset the current value, reset status special internal relay M8130 turns on in the next scan. When reset input special internal relay M8032 is turned on, M8130 does not turn on. See page 5-13.

Note: When using input I2 as a phase Z input, set 0 to reset value special data register D8046.

Special Internal Relays for Two-phase High-speed Counter (All-in-One Type CPU Modules)

Description	High-speed Counter No.				ON	Read/Write
	HSC1	HSC2	HSC3	HSC4		
Comparison Output Reset	M8030	—	—	—	Turns off comparison output	R/W
Gate Input	M8031	—	—	—	Enables counting	R/W
Reset Input	M8032	—	—	—	Resets the current value	R/W
Reset Status	M8130	—	—	—	Current value reset by I2	Read only
Current Value Overflow	M8131	—	—	—	Overflow occurred	Read only
Current Value Underflow	M8132	—	—	—	Underflow occurred	Read only

Note: Special internal relays M8130 through M8132 go on for only one scan.

Special Data Registers for Two-phase High-speed Counter (All-in-One Type CPU Modules)

Description	High-speed Counter No.				Updated	Read/Write
	HSC1	HSC2	HSC3	HSC4		
High-speed Counter Current Value	D8045	—	—	—	Every scan	Read only
High-speed Counter Reset Value	D8046	—	—	—	—	R/W

Two-phase High-speed Counter Specifications (All-in-One Type CPU Modules)

Maximum Counting Frequency	20 kHz
Counting Range	0 to 65535 (16 bits)
Operation Mode	Rotary encoder (phases A, B, Z)
Gate Control	Enable/disable counting
Current Value Reset	Current value is reset to a given value when the current value overflows 65535 or underflows 0, or when reset input I2 or reset input special internal relay M8032 is turned on.
Control/Status Relays	Special internal relays are provided to control and monitor the high-speed counter operation.
Comparison Output	Any output number available on the CPU module can be designated as a comparison output which turns on when the current value reaches the preset value. Output numbers on expansion output or mixed I/O modules cannot be designated as a comparison output.

Single-phase High-speed Counters HSC1 through HSC4 (All-in-One Type CPU Modules)

HSC1 can also be used as a single-phase high-speed counter as well as HSC2 through HSC4. The four single-phase high-speed counters count input pulses to the input terminal allocated to each high-speed counter. When the preset value is reached, a designated comparison output turns on, and the current value is reset to 0 to count subsequent input pulses.

Two special data registers and four special internal relays are assigned to control and monitor the single-phase high-speed counter operation. The current value is stored in a special data register (current value) and is updated every scan. The value stored in another special data register (preset value) is used as a preset value. When a reset input special internal relay is turned on, the current value is reset to 0.

The single-phase high-speed counter is enabled while a gate input special internal relay is on and is disabled while the gate input is off. When the current value reaches the preset value, a special internal relay (comparison ON status) turns on in the next scan. At this point, the current value is reset to 0, and the value stored in a preset value special data register takes effect for the subsequent counting cycle. When a comparison output reset special internal relay is turned on, the designated comparison output is turned off.

In addition, only the single-phase high-speed counter HSC1 has reset input I2 and reset status special internal relay M8130. When reset input I2 is turned on to reset the current value to 0, reset status special internal relay M8130 turns on in the next scan. When reset input special internal relay M8032 is turned on, M8130 does not turn on. See page 5-14.

Special Internal Relays for Single-phase High-speed Counters (All-in-One Type CPU Modules)

Description	High-speed Counter No.				ON	Read/Write
	HSC1	HSC2	HSC3	HSC4		
Comparison Output Reset	M8030	M8034	M8040	M8044	Turns off comparison output	R/W
Gate Input	M8031	M8035	M8041	M8045	Enables counting	R/W
Reset Input	M8032	M8036	M8042	M8046	Resets the current value	R/W
Reset Status	M8130	—	—	—	Current value reset by I2	Read only
Comparison ON Status	M8131	M8133	M8134	M8136	Preset value reached	Read only

Note: Special internal relays M8130, M8131, M8133, M8134, and M8136 go on for only one scan.

Special Data Registers for Single-phase High-speed Counters (All-in-One Type CPU Modules)

Description	High-speed Counter No.				Updated	Read/Write
	HSC1	HSC2	HSC3	HSC4		
High-speed Counter Current Value	D8045	D8047	D8049	D8051	Every scan	Read only
High-speed Counter Preset Value	D8046	D8048	D8050	D8052	—	R/W

Single-phase High-speed Counter Specifications (All-in-One Type CPU Modules)

Maximum Counting Frequency	HSC1: 20 kHz HSC2 through HSC4: 5 kHz
Counting Range	0 to 65535 (16 bits)
Operation Mode	Adding counter
Gate Control	Enable/disable counting
Current Value Reset	Current value is reset to 0 when the current value reaches the preset value or when reset input I2 (HSC1 only) or a reset input special internal relay is turned on.
Status Relays	Special internal relays for indicating high-speed counter statuses.
Comparison Output	Any output number available on the CPU module can be designated as a comparison output which turns on when the current value reaches the preset value. Output numbers on expansion output or mixed I/O modules cannot be designated as a comparison output.

High-speed Counters on Slim Type CPU Modules

Slim type CPU modules have four high-speed counters; HSC1 through HSC4. HSC1 and HSC4 can be used as a two-phase or single-phase high-speed counter. HSC2 and HSC3 are single-phase high-speed counters. All high-speed counter functions are selected using the Function Area Settings in WindLDR.

High-speed Counter Operation Modes and Input Terminals (Slim Type CPU Modules)

High-speed Counter No.	HSC1			HSC2	HSC3	HSC4		
Input Terminal	I0	I1	I2	I3	I4	I5	I6	I7
Two-phase High-speed Counter	Phase A	Phase B	Reset Input (Phase Z)	—	—	Reset Input (Phase Z)	Phase A	Phase B
Single-phase High-speed Counter	—	Pulse Input	Reset Input	Pulse Input	Pulse Input	Reset Input	—	Pulse Input

For wiring high-speed counter input signals, use twisted-pair shielded cables.

Two-phase High-speed Counters HSC1 and HSC4 (Slim Type CPU Modules)

Two-phase high-speed counter HSC1 or HSC4 operates in the rotary encoder mode, and counts up or down input pulses to input terminals I0 or I6 (phase A) and I1 or I7 (phase B), respectively. When the current value overflows 65535 or underflows 0, a designated comparison output turns on. Any output terminal available on the CPU module can be designated as a comparison output. When input I2 or I5 (reset input) is turned on, the current value is reset to a predetermined reset value, and the two-phase high-speed counter counts subsequent input pulses starting at the reset value.

Two special data registers and six special internal relays are assigned to control and monitor each two-phase high-speed counter operation. The current value is stored in data register D8045 or D8051(current value) and is updated every scan. The value stored in D8046 or D8052 (reset value) is used as a reset value. When a high-speed counter reset input (I2/I5 or M8032/M8046) is turned on, the current value in D8045 or D8051 is reset to the value stored in D8046 or D8052.

The two-phase high-speed counter is enabled while gate input special internal relay M8031 or M8045 is on and is disabled while M8031 or M8045 is off. When current value overflow or underflow occurs while counting up or down, special internal relay M8131/M8136 or M8132/M8137 turns on in the next scan, respectively. At this point, the D8045 or D8051 current value is reset to the D8046 or D8052 reset value for the subsequent counting cycle. When comparison output reset special internal relay M8030 or M8044 is turned on, the designated comparison output is turned off. When reset input I2 or I5 is turned on to reset the current value, reset status special internal relay M8130 or M8135 turns on in the next scan. When reset input special internal relay M8032 or M8046 is turned on, M8130 or M8135 does not turn on. See page 5-13.

Note: When using input I2 or I5 as a phase Z input, set 0 to reset value special data register D8046 or D8052, respectively.

Special Internal Relays for Two-phase High-speed Counter (Slim Type CPU Modules)

Description	High-speed Counter No.				ON	Read/Write
	HSC1	HSC2	HSC3	HSC4		
Comparison Output Reset	M8030	—	—	M8044	Turns off comparison output	R/W
Gate Input	M8031	—	—	M8045	Enables counting	R/W
Reset Input	M8032	—	—	M8046	Resets the current value	R/W
Reset Status	M8130	—	—	M8135	Current value reset by I2 or I5	Read only
Current Value Overflow	M8131	—	—	M8136	Overflow occurred	Read only
Current Value Underflow	M8132	—	—	M8137	Underflow occurred	Read only

Note: Special internal relays M8130 through M8132 and M8135 through M8137 go on for only one scan.

Special Data Registers for Two-phase High-speed Counter (Slim Type CPU Modules)

Description	High-speed Counter No.				Updated	Read/Write
	HSC1	HSC2	HSC3	HSC4		
High-speed Counter Current Value	D8045	—	—	D8051	Every scan	Read only
High-speed Counter Reset Value	D8046	—	—	D8052	—	R/W

Two-phase High-speed Counter Specifications (Slim Type CPU Modules)

Maximum Counting Frequency	20 kHz
Counting Range	0 to 65535 (16 bits)
Operation Mode	Rotary encoder (phases A, B, Z)
Gate Control	Enable/disable counting
Current Value Reset	Current value is reset to a given value when the current value overflows 65535 or underflows 0, or when reset input I2/I5 or reset input special internal relay M8032/M8046 is turned on.
Control/Status Relays	Special internal relays are provided to control and monitor the high-speed counter operation.
Comparison Output	Any output number available on the CPU module can be designated as a comparison output which turns on when the current value reaches the preset value. Output numbers on expansion output or mixed I/O modules cannot be designated as a comparison output.

Single-phase High-speed Counters HSC1 through HSC4 (Slim Type CPU Modules)

HSC1 and HSC4 can also be used as a single-phase high-speed counter as well as HSC2 and HSC3. The four single-phase high-speed counters count input pulses to the input terminal allocated to each high-speed counter. When the preset value is reached, a designated comparison output turns on, and the current value is reset to 0 to count subsequent input pulses.

Two special data registers and four special internal relays are assigned to control and monitor the single-phase high-speed counter operation. The current value is stored in a special data register (current value) and is updated every scan. The value stored in another special data register (preset value) is used as a preset value. When a reset input special internal relay is turned on, the current value is reset to 0.

The single-phase high-speed counter is enabled while a gate input special internal relay is on and is disabled while the gate input is off. When the current value reaches the preset value, a special internal relay (comparison ON status) turns on in the next scan. At this point, the current value is reset to 0, and the value stored in a preset value special data register takes effect for the subsequent counting cycle. When a comparison output reset special internal relay is turned on, the designated comparison output is turned off.

In addition, only the single-phase high-speed counter HSC1 or HSC4 has reset input I2 or I5 and reset status special internal relay M8130 or M8135. When reset input I2 or I5 is turned on to reset the current value to 0, reset status special internal relay M8130 or M8135 turns on in the next scan. When reset input special internal relay M8032 or M8046 is turned on, M8130 or M8135 does not turn on. See page 5-14.

Special Internal Relays for Single-phase High-speed Counters (Slim Type CPU Modules)

Description	High-speed Counter No.				ON	Read/Write
	HSC1	HSC2	HSC3	HSC4		
Comparison Output Reset	M8030	M8034	M8040	M8044	Turns off comparison output	R/W
Gate Input	M8031	M8035	M8041	M8045	Enables counting	R/W
Reset Input	M8032	M8036	M8042	M8046	Resets the current value	R/W
Reset Status	M8130	—	—	M8135	Current value reset by I2 or I5	Read only
Comparison ON Status	M8131	M8133	M8134	M8136	Preset value reached	Read only

Note: Special internal relays M8130, M8131, M8133, M8134, M8135, and M8136 go on for only one scan.

Special Data Registers for Single-phase High-speed Counters (Slim Type CPU Modules)

Description	High-speed Counter No.				Updated	Read/Write
	HSC1	HSC2	HSC3	HSC4		
High-speed Counter Current Value	D8045	D8047	D8049	D8051	Every scan	Read only
High-speed Counter Preset Value	D8046	D8048	D8050	D8052	—	R/W

Single-phase High-speed Counter Specifications (Slim Type CPU Modules)

Maximum Counting Frequency	HSC1 and HSC4: 20 kHz HSC2 and HSC3: 5 kHz
Counting Range	0 to 65535 (16 bits)
Operation Mode	Adding counter
Gate Control	Enable/disable counting
Current Value Reset	Current value is reset to 0 when the current value reaches the preset value, when reset input I2 (HSC1) or I5 (HSC4), or when a reset input special internal relay is turned on.
Status Relays	Special internal relays for indicating high-speed counter statuses.
Comparison Output	Any output number available on the CPU module can be designated as a comparison output which turns on when the current value reaches the preset value. Output numbers on expansion output or mixed I/O modules cannot be designated as a comparison output.

Clearing High-speed Counter Current Value

The high-speed counter current value is reset to the reset value (two-phase high-speed counter) or to zero (single-phase high-speed counters) in five ways:

- when the CPU is powered up,
- when a user program is downloaded to the CPU,
- when reset input I2 (HSC1) or I5 (HSC4 on slim type CPU only) is turned on,
- when current value overflow or underflow occurs (two-phase) or when the preset value is reached (single-phase), or
- when the reset input (*not* the high-speed counter reset input) designated in the Function Area Settings is turned on.

Precautions for Downloading High-speed Counter Program

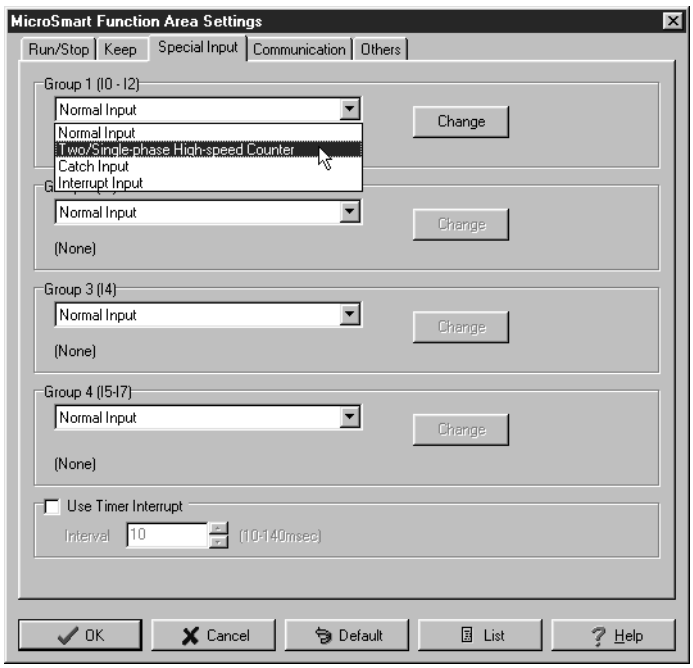
When downloading a user program containing a high-speed counter, turn off the gate input before downloading the user program.

If a user program containing a high-speed counter is downloaded while the gate input is on, the high-speed counter is disabled. Then, to enable counting, stop and restart the MicroSmart. Or, turn off the gate input, and 3 scans later turn on the gate input again. For ladder programs to delay the gate input 3 scans, see pages 5-16 and 5-17.

Programming WindLDR (All-in-One Type CPU Modules)

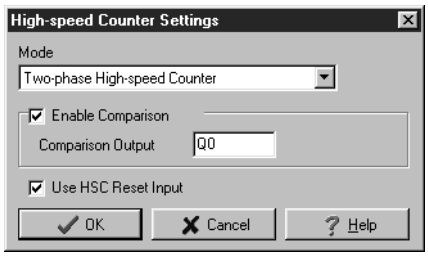
1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.

2. Select the **Special Input** tab.



3. When using high-speed counter HSC1, select **Two/Single-phase High-speed Counter** in the Group 1 pull-down list box.

   When using high-speed counters HSC2 through HSC4, select **Single-phase High-speed Counter** in the Groups 2 through 4 pull-down list boxes.



The High-speed Counter Settings dialog box appears.

Mode

Select **Two-phase High-speed Counter** or **Single-phase High-speed Counter** for HSC1. Only Single-phase High-speed Counter is available for HSC2 through HSC4.

Enable Comparison

Click the check box to enable the high-speed counter comparison output, and specify an output number available on the CPU module in the **Comparison Output** field. When current value overflow or underflow occurs (two-phase high-speed counter) or when the preset value is reached (single-phase high-speed counter), the specified comparison output is turned on and remains on until a comparison output reset special internal relay (M8030, M8034, M8040, or M8044) is turned on.

Use HSC Reset Input

Click the check box to enable high-speed counter reset input I2 for HSC1 only. When input I2 is turned on, the current value in D8045 is reset depending on the high-speed counter mode.

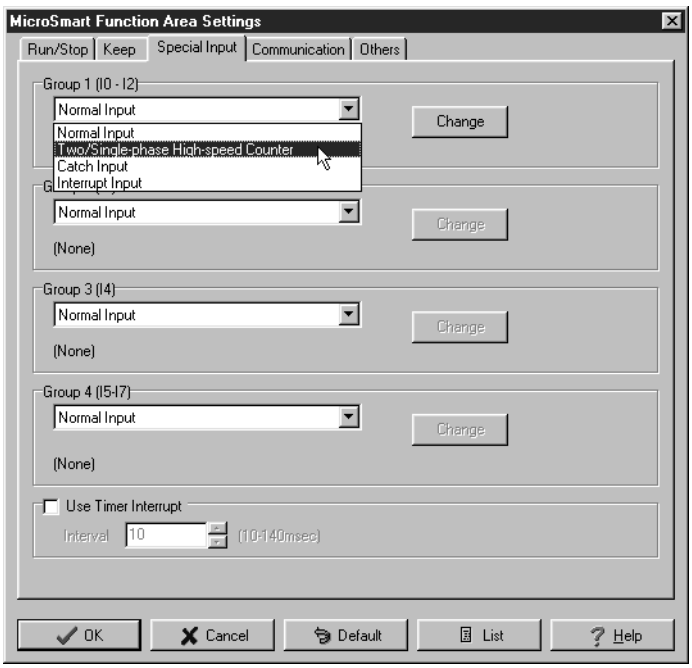
Two-phase	The current value is reset to the value stored in D8046 (high-speed counter reset value). The two-phase high-speed counter counts subsequent input pulses starting at the reset value.
Single-phase	The current value is reset to 0. The value stored in D8046 (high-speed counter preset value) at this point takes effect for the subsequent counting cycle.

Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Programming WindLDR (Slim Type CPU Modules)

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.

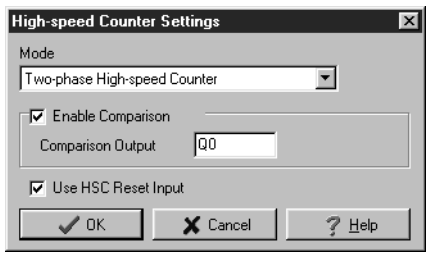
2. Select the **Special Input** tab.



3. When using high-speed counter HSC1 or HSC4, select **Two/Single-phase High-speed Counter** in the Group 1 or 4 pull-down list box.

   When using high-speed counters HSC2 or HSC3, select **Single-phase High-speed Counter** in the Group 2 or 3 pull-down list box.

The High-speed Counter Settings dialog box appears.



Mode

Select **Two-phase High-speed Counter** or **Single-phase High-speed Counter** for HSC1 or HSC4. Only Single-phase High-speed Counter is available for HSC2 and HSC3.

Enable Comparison

Click the check box to enable the high-speed counter comparison output, and specify an output number available on the CPU module in the **Comparison Output** field. When current value overflow or underflow occurs (two-phase high-speed counter) or when the preset value is reached (single-phase high-speed counter), the specified comparison output is turned on and remains on until a comparison output reset special internal relay (M8030, M8034, M8040, or M8044) is turned on.

Use HSC Reset Input

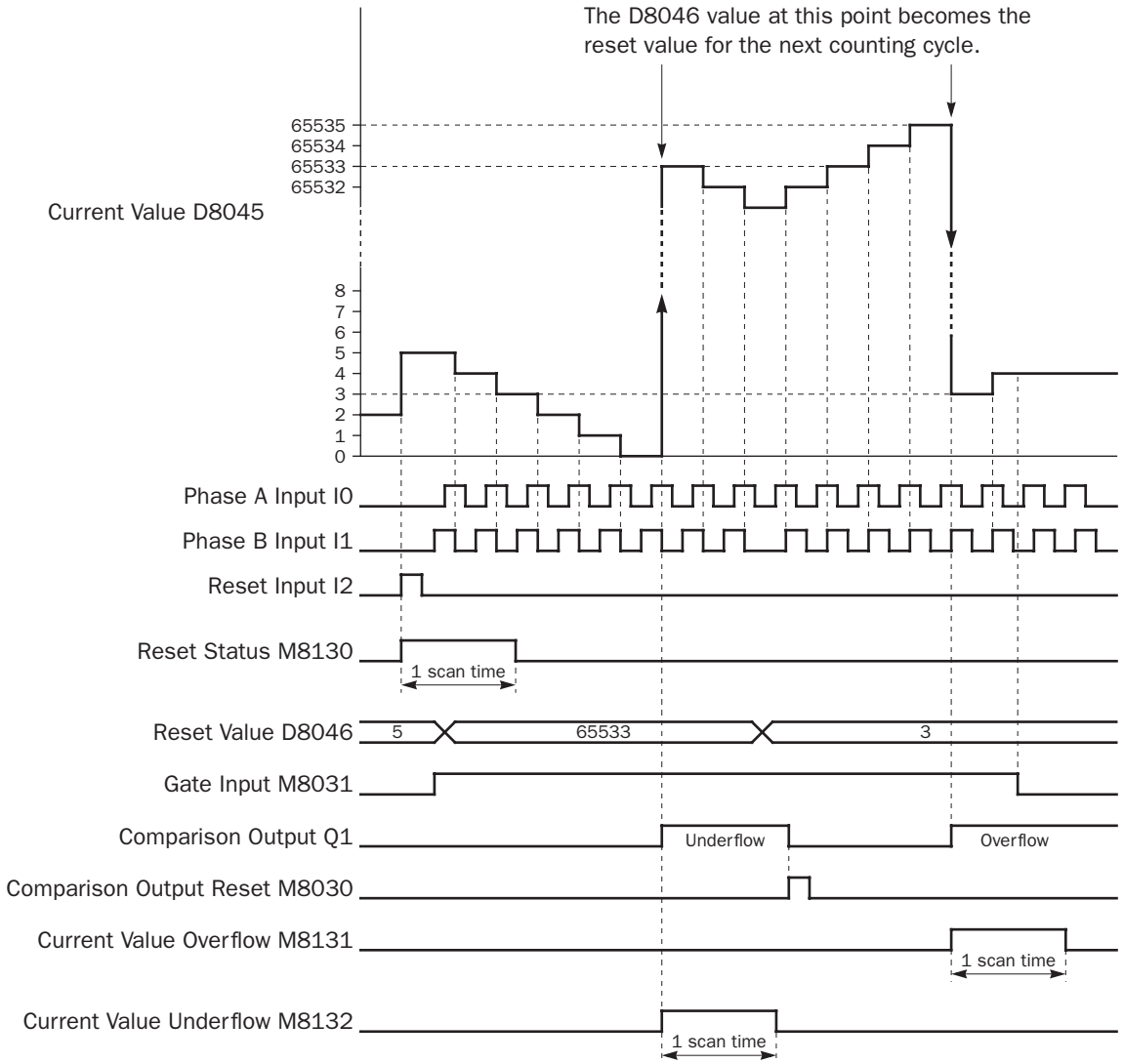
Click the check box to enable high-speed counter reset input I2 for HSC1 or I5 for HSC4 only. When input I2 or I5 is turned on, the current value in D8045 or D8051 is reset depending on the high-speed counter mode.

Two-phase	The current value is reset to the value stored in D8046 or D8052 (high-speed counter reset value). The two-phase high-speed counter counts subsequent input pulses starting at the reset value.
Single-phase	The current value is reset to 0. The value stored in D8046 or D8052 (high-speed counter preset value) at this point takes effect for the subsequent counting cycle.

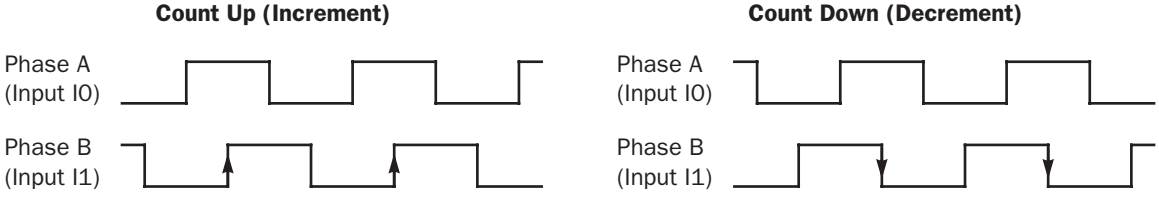
Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Two-phase High-speed Counter Timing Chart

Example: Reset input I2 is used. Q1 is designated as a comparison output.

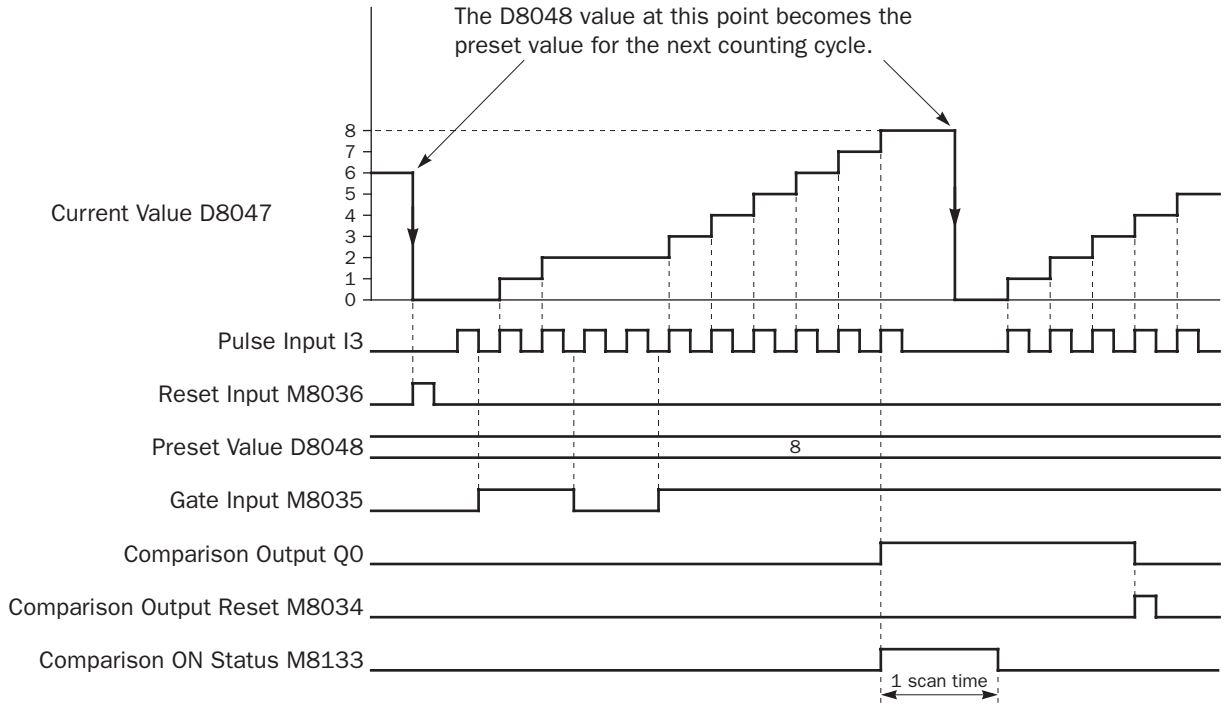


- When reset input I2 is turned on, the D8046 reset value is set to the D8045 current value, then reset status M8130 turns on for one scan. If reset input M8032 is turned on, reset status M8130 does not turn on.
- While gate input M8031 is on, the two-phase high-speed counter counts up or down depending on the phase difference between phase A (input I0) and phase B (input I1).



Single-phase High-speed Counter Timing Chart

Example: Single–phase high-speed counter HSC2
Preset value is 8. Q0 is designated as a comparison output.



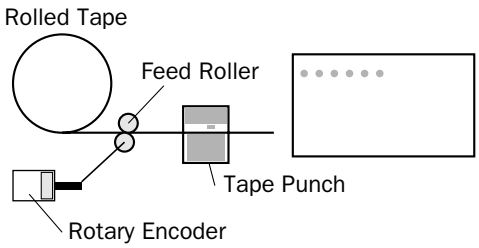
- When reset input M8036 is turned on, the D8047 current value is cleared to 0, then the D8048 preset value takes effect for the next counting cycle.
- While gate input M8035 is on, single-phase high-speed counter HSC2 counts pulse inputs to input I3.
- The D8047 current value is updated every scan.
- When the D8047 current value reaches the D8048 preset value, comparison ON status M8133 goes on for one scan. At the same time, comparison output Q0 turns on and remains on until comparison output reset M8034 is turned on.
- When the D8047 current value reaches the D8048 preset value, the D8048 preset value at that point takes effect for the next counting cycle.

Example: Two-phase High-speed Counter for Counting Input Pulses from Rotary Encoder

This example demonstrates a program for two-phase high-speed counter HSC1 to punch holes in a paper tape at regular intervals.

Description of Operation

A rotary encoder is linked to the tape feed roller directly, and the output pulses from the rotary encoder are counted by the two-phase high-speed counter in the MicroSmart CPU module. When the high-speed counter counts 2,700 pulses, the comparison output is turned on. When the comparison output is turned on, the high-speed counter continues another cycle of counting. The comparison output remains on for 0.5 second to punch holes in the tape, and is turned off before the high-speed counter counts 2,700 pulses again.

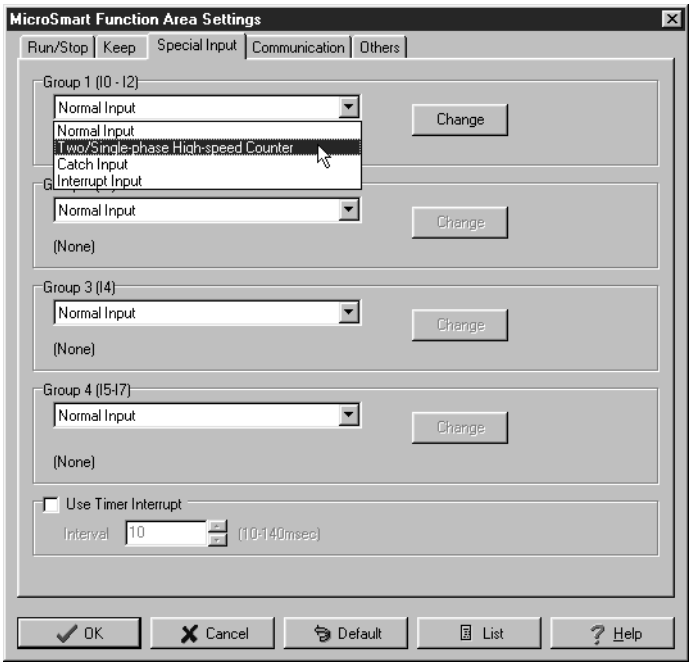


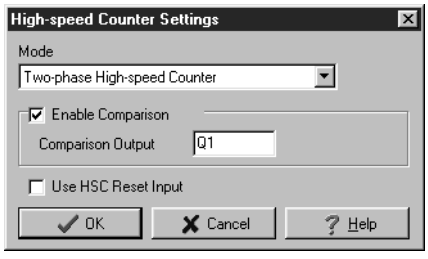
Program Parameters

Group 1 (I0 - I2)	Two/Single-phase High-speed Counter
High-speed Counter Settings	Two-phase High-speed Counter
Enable Comparison	Yes
Comparison Output	Q1
Use HSC Reset Input (I2)	No
HSC Reset Value (D8046)	To cause current value overflow every 2700 pulses, store 62836 to D8046 (65535 – 2700 + 1 = 62836)
Timer Preset Value	0.5 sec (needed for punching) programmed in TIM instruction

Note: This example does not use the phase Z signal (input I2).

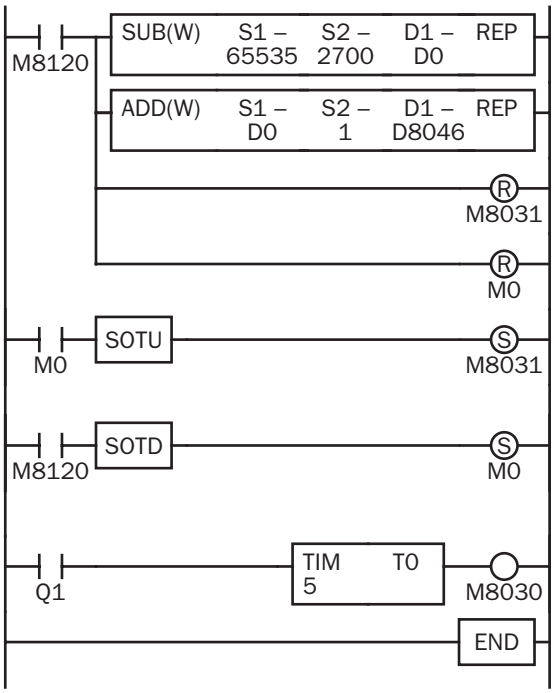
Programming WindLDR





Ladder Diagram

When the MicroSmart starts operation, reset value 62836 is stored to reset value special internal relay D8046. Gate input special internal relay M8031 is turned on at the end of the third scan to start the high-speed counter to count input pulses.



M8120 is the initialize pulse special internal relay.

1st scan

SUB and ADD instructions are used to store a reset value of 62836 (65535 – 2700 + 1) to D8046 (reset value).

M8031 (gate input) is turned off.

M0 is turned off.

3rd scan

At the rising edge of M0, M8031 (gate input) is turned on. After the END processing of the third scan, HSC1 starts counting.

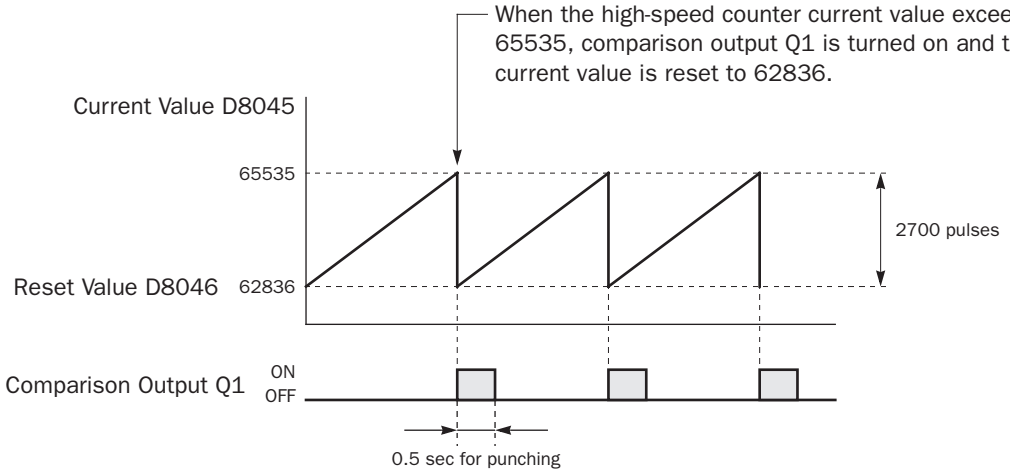
2nd scan

At the falling edge of M8120 (initialize pulse), M0 is turned on. HSC1 is initialized in the END processing of the second scan.

When HSC1 overflows 65535, output Q1 (comparison output) is turned on to start timer T0. HSC1 starts to repeat counting.

When the timer times out 0.5 sec, M8030 (comparison output reset) is turned on to turn off output Q1.

Timing Chart



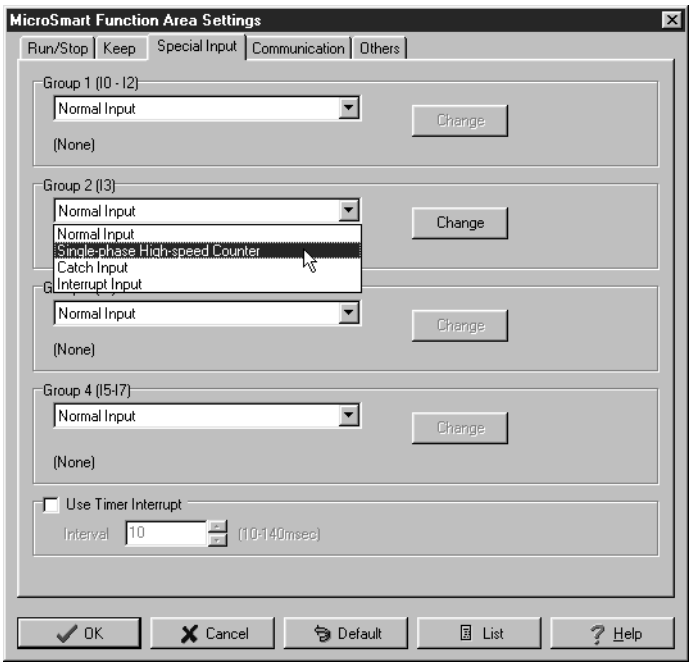
Example: Single-phase High-speed Counter

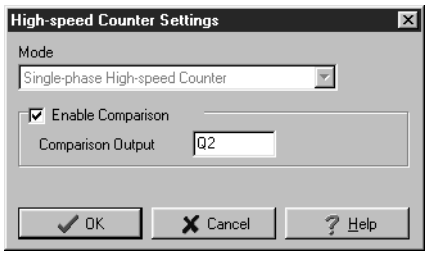
This example demonstrates a program for single-phase high-speed counter HSC2 to count input pulses and turn on output Q2 every 1000 pulses.

Program Parameters

Group 2 (I3)	Single-phase High-speed Counter
Enable Comparison	Yes
Comparison Output	Q2
HSC Preset Value (D8048)	1000

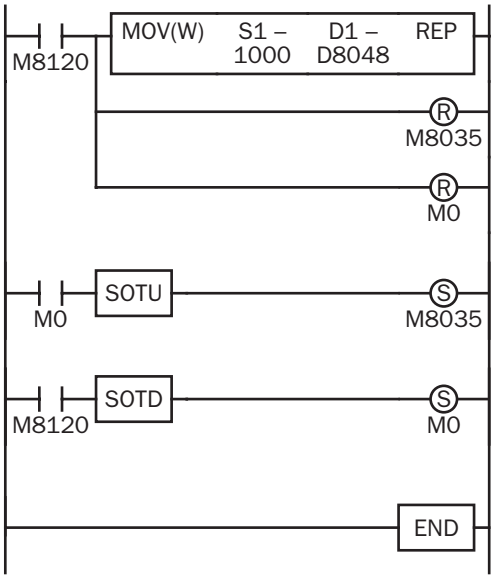
Programming WindLDR





Ladder Diagram

When the MicroSmart starts operation, preset value 1000 is stored to preset value special internal relay D8048. Gate input special internal relay M8035 is turned on at the end of the third scan to start the high-speed counter to count input pulses.



M8120 is the initialize pulse special internal relay.

1st scan

MOV instruction stores a reset value of 1000 to D8048 (preset value).

M8035 (gate input) is turned off.

M0 is turned off.

3rd scan

At the rising edge of M0, M8035 (gate input) is turned on. After the END processing of the third scan, HSC2 starts counting.

2nd scan

At the falling edge of M8120 (initialize pulse), M0 is turned on. HSC2 is initialized in the END processing of the second scan.

When HSC2 current value reaches 1000, output Q2 (comparison output) is turned on, and HSC2 starts to repeat counting from zero.

Catch Input

The catch input function is used to receive short pulses from sensor outputs regardless of the scan time. Input pulses shorter than one scan time can be received. Four inputs I2 through I5 can be designated to catch a rising or falling edge of short input pulses, and the catch input statuses are stored to special internal relays M8154 through M8157, respectively. The Function Area Settings dialog box is used to designate inputs I2 through I5 as a catch input.

Normal input signals to input terminals are read when the END instruction is executed at the end of a scan.

Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Catch Input Specifications

Minimum Turn ON Pulse Width	40 µs
Minimum Turn OFF Pulse Width	150 µs

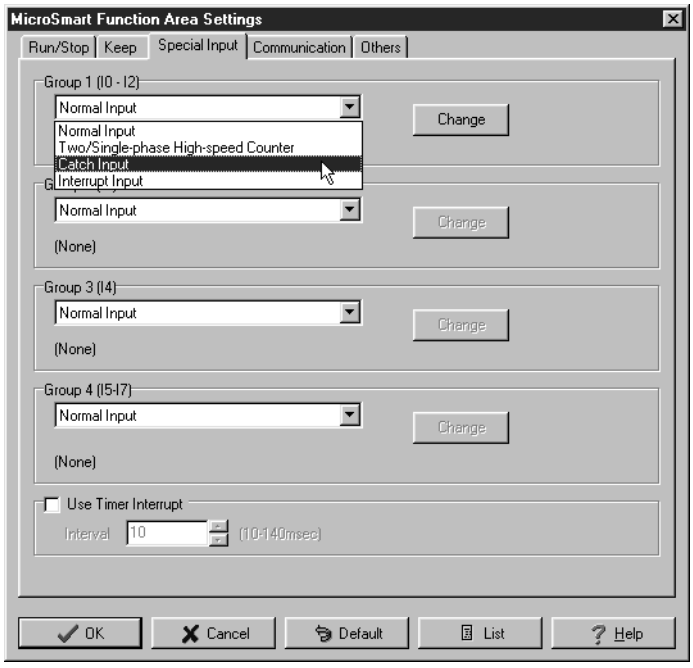
Note: Input filter settings have no effect on the catch inputs. For the input filter function, see page 5-24.

Catch Input Terminals and Special Internal Relays for Catch Inputs

Group	Catch Input No.	Special Internal Relay for Catch Input
Group 1	I2	M8154
Group 2	I3	M8155
Group 3	I4	M8156
Group 4	I5	M8157

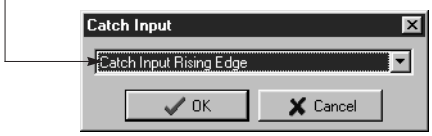
Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Special Input** tab.



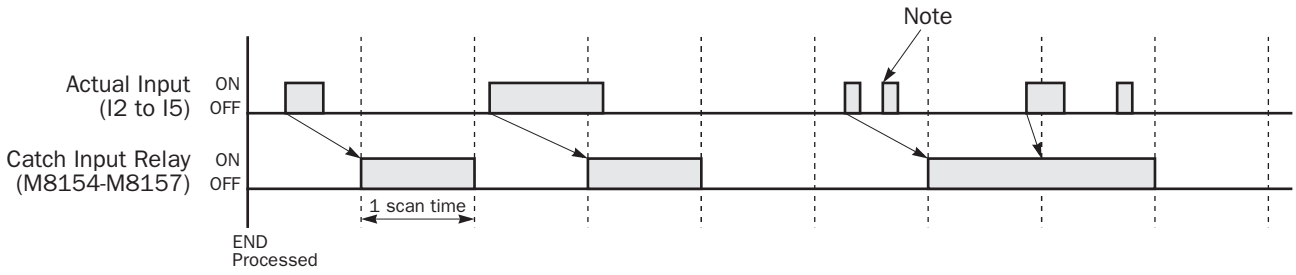
Catch Input Rising/Falling Edge Selection

Catch Input Rising Edge
Catch Input Falling Edge

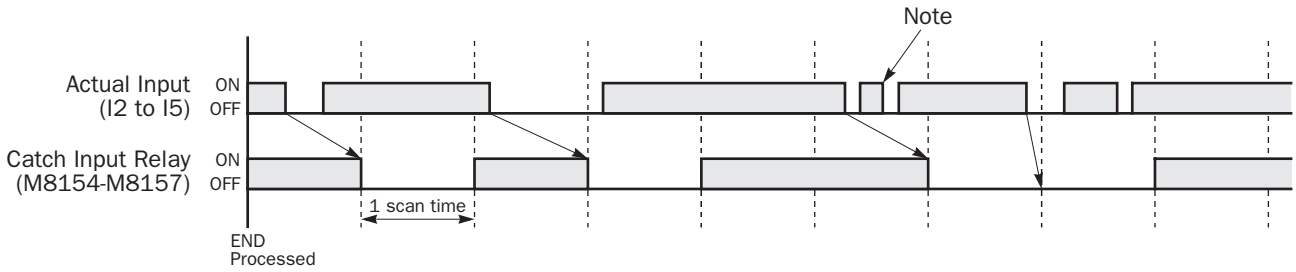


3. Select **Catch Input** in the Groups 1 through 4 pull-down list boxes. The Catch Input dialog box appears.
4. Select **Catch Input Rising Edge** or **Catch Input Falling Edge** in the pull-down list.

Catching Rising Edge of Input Pulse



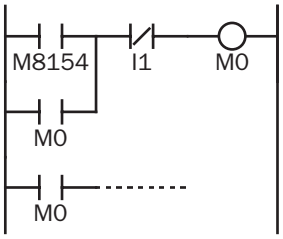
Catching Falling Edge of Input Pulse



Note: When two or more pulses enter within one scan, subsequent pulses are ignored.

Example: Maintaining Catch Input

When a catch input is received, the catch input relay assigned to a catch input is turned on for only one scan. This example demonstrates a program to maintain a catch input status for more than one scan.



Input I2 is designated as a catch input using the Function Area Settings.

When input I2 is turned on, special internal relay M8154 is turned on, and M0 is maintained in the self-holding circuit.

When NC input I1 is turned off, the self-holding circuit is unlatched, and M0 is turned off.

M0 is used as an input condition for the subsequent program instructions.

Interrupt Input

All MicroSmart CPU modules have an interrupt input function. When a quick response to an external input is required, such as positioning control, the interrupt input can call a subroutine to execute an interrupt program.

Four inputs I2 through I5 can be designated to execute interrupt at a rising and/or falling edge of input pulses. When an interrupt is initiated by inputs I2 through I5, program execution immediately jumps to a predetermined label number stored in special data registers D8032 through D8035, respectively. The Function Area Settings dialog box is used to designate inputs I2 through I5 as an interrupt input, normal input, high-speed counter input, or catch input.

Normal input signals to input terminals are read when the END instruction is executed at the end of a scan.

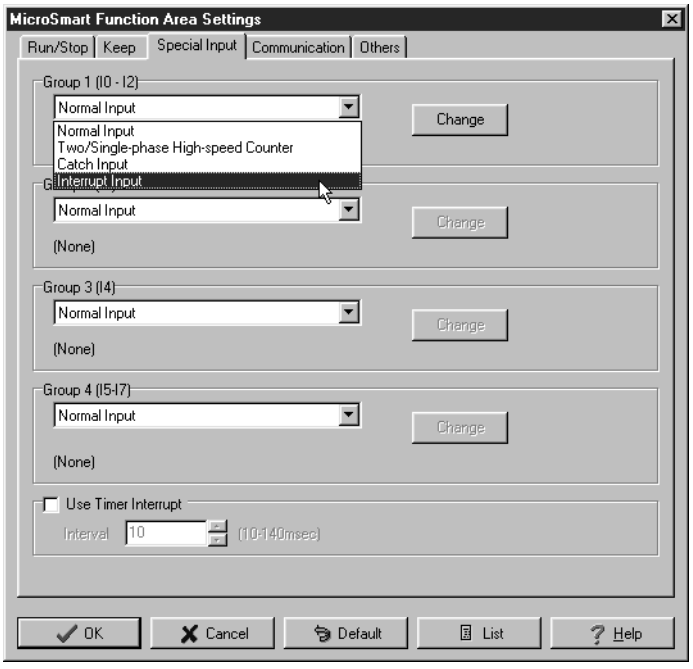
Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Interrupt Input Terminals, Special Data Registers, and Special Internal Relays for Interrupt Inputs

Group	Interrupt Input No.	Interrupt Input Jump Destination Label No.	Interrupt Input Status
Group 1	I2	D8032	M8140
Group 2	I3	D8033	M8141
Group 3	I4	D8034	M8142
Group 4	I5	D8035	M8143

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Special Input** tab.



Interrupt Input Rising/Falling Edge Selection

Interrupt at Rising Edge
: Interrupt occurs when the interrupt input turns on.

Interrupt at Falling Edge
: Interrupt occurs when the interrupt input turns off.

Interrupt at Both Edges
: Interrupt occurs when the interrupt input turns on or off.

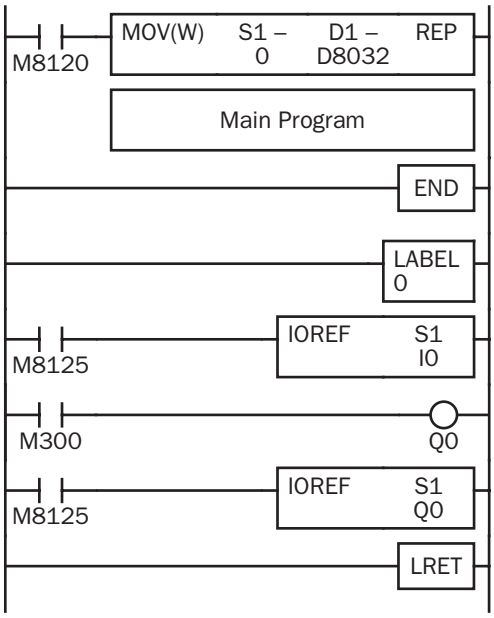
3. Select **Interrupt Input** in the Groups 1 through 4 pull-down list boxes. the Interrupt Input dialog box appears.
4. Select an interrupt edge in the pull-down list for each group.

Disable and Enable Interrupts

The interrupt inputs I2 through I5 and timer interrupt are normally enabled while the CPU is running, and can also be individually disabled using the DI instruction or enabled using the EI instruction. When interrupt inputs I2 through I5 are enabled, special internal relay M8140 through M8143 are turned on, respectively. See page 18-7.

Example: Interrupt Input

The following example demonstrates a program of using the interrupt input function, with input I2 designated as an interrupt input. When the interrupt input is turned on, the input I0 status is immediately transferred to output Q0 using the IOREF (I/O refresh) instruction before the END instruction is executed. For the IOREF instruction, see page 18-5.



M8120 is the initialize pulse special internal relay.

D8032 stores 0 to designate jump destination label 0 for interrupt input I2.

The interrupt program is separated from the main program by the END instruction.

When input I2 is on, program execution jumps to label 0.

M8125 is the in-operation output special internal relay.

IOREF immediately reads input I0 status to internal relay M300.

M300 turns on or off the output Q0 internal memory.

Another IOREF immediately writes the output Q0 internal memory status to actual output Q0.

Program execution returns to the main program.

Insert LRET at the end of the subroutine to return to the main program.

Notes for Using Interrupt Inputs and Timer Interrupt:

- When using an interrupt input or timer interrupt, separate the interrupt program from the main program using the END instruction at the end of the main program.
- When an interrupt program calls another subroutine, a maximum of 3 subroutine calls can be nested. If more than 3 calls are nested, a user program execution error occurs, turning on special internal relay M8004 and the ERR LED.
- When using an interrupt input or timer interrupt, include the label number of the interrupt program to be executed when an interrupt occurs. The label numbers stored in data registers D8032 through D8036 specify the interrupt programs for interrupt inputs I2 through I5 and timer interrupt, respectively.
- When more than one interrupt input or timer interrupt is turned on at the same time, interrupt program execution is given priority to inputs I2, I3, I4, timer interrupt, and I5, in that order. If an interrupt is initiated while another interrupt program is executed, the subsequent interrupt program is executed after the prior interrupt is completed. Multiple interrupt programs cannot be executed simultaneously.
- When a communication function is used, such as data link, the interrupt program size must be limited to the executable time listed in the table below.

Communication Function	Baud Rate (bps)	Executable Time of Interrupt Program (µs)
Not Used	—	670 maximum
Used	1200, 2400, 4800, 9600	670 maximum
Used	19200	170 maximum

- If the interrupt program is longer than the value listed above, the entire system performance is affected. The timer and filter functions may not operate correctly, and communication error may be caused in the data link or communication with display units. Make sure that the interrupt program execution time is within the values listed above, referring to the execution times on page A-1. When using high-speed counters, the interrupt program size has to be much smaller.
- When using the data link and interrupt inputs, select 19200 bps for the baud rate of the data link communication.
- Make sure that the execution time of the interrupt program is shorter than interrupt intervals sufficiently.
- Interrupt programs cannot use the following instructions: SOTU, SOTD, TML, TIM, TMH, TMS, CNT, CDP, CUD, SFR, SFRN, ROOT, WKTIM, WKTBL, DISP, DGRD, TXD1/2, RXD1/2, DI, EI, XYFS, CVXTY, CVYTX, PULS1/2, PWM1/2, RAMP, ZRN1/2, PID, DTML, DTIM, DTMH, DTMS, and TTIM.
- The overhead from interrupt occurrence to interrupt program execution is approximately 60 µs. When using the high-speed counter, the overhead may be extended.

Timer Interrupt

In addition to the interrupt input as described in the preceding section, slim type CPU modules FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K1, and FC4A-D40S1 have a timer interrupt function. When a repetitive operation is required, the timer interrupt can be used to call a subroutine repeatedly at predetermined intervals of 10 through 140 ms.

The Function Area Settings dialog box is used to enable the timer interrupt and to specify the interval, from 10 to 140 ms, to execute the timer interrupt. When the timer interrupt is enabled, the program execution jumps to the jump destination label number stored in special data register D8036 repeatedly while the CPU is running. When the interrupt program is completed, the program execution returns to the main program at the address where the interrupt occurred.

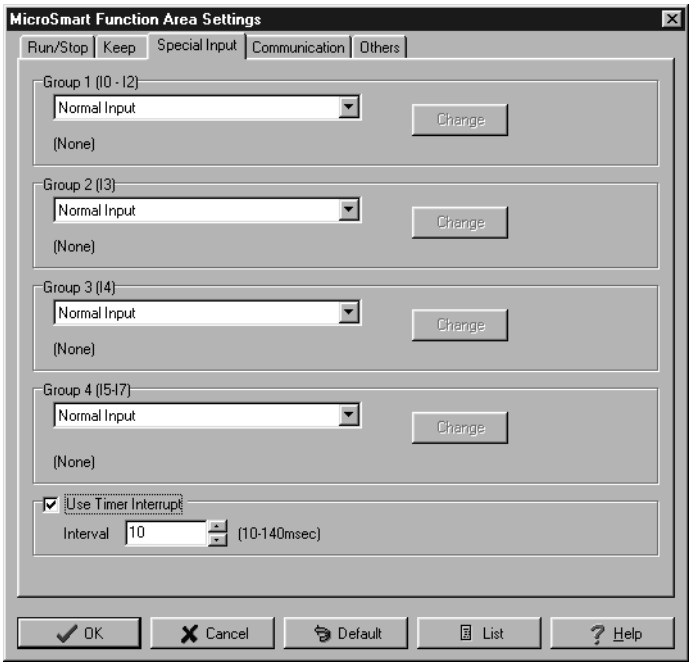
Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Special Data Register and Special Internal Relay for Timer Interrupt

Interrupt	Special Data Register for Timer Interrupt Jump Destination Label No.	Special Internal Relay for Timer Interrupt Status
Timer Interrupt	D8036	M8144

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Special Input** tab.



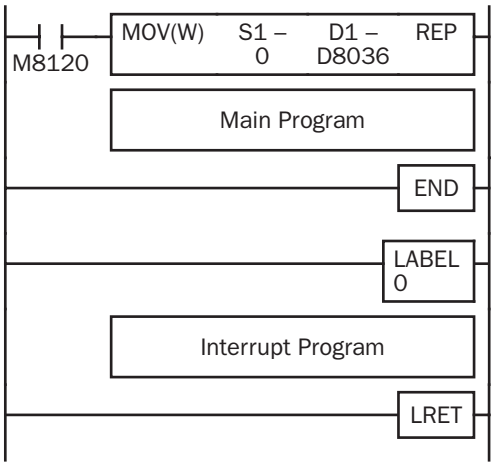
3. Click the check box on the left of Timer Interrupt to use the timer interrupt function.
4. Select an interval to execute the timer interrupt, from 10 to 140 ms.

Disable and Enable Interrupts

The timer interrupt and interrupt inputs I2 through I5 are normally enabled while the CPU is running, and can also be individually disabled using the DI instruction or enabled using the EI instruction. When timer interrupt is enabled, M8144 is turned on. When disabled, M8144 is turned off. See page 18-7.

Example: Timer Interrupt

The following example demonstrates a program of using the timer interrupt function. The Function Area Settings must also be completed to use the timer interrupt function as described on the preceding page.



M8120 is the initialize pulse special internal relay.

D8036 stores 0 to designate jump destination label 0 for timer interrupt.

The interrupt program is separated from the main program by the END instruction.

While the CPU is running, program execution jumps to label 0 repeatedly at intervals selected in the Function Area Settings.

Each time the interrupt program is completed, program execution returns to the main program at the address where timer interrupt occurred.

Insert LRET at the end of the subroutine to return to the main program.

Notes for Using Timer Interrupt and Interrupt Inputs:

- When using a timer interrupt or interrupt input, separate the interrupt program from the main program using the END instruction at the end of the main program.
- When an interrupt program calls another subroutine, a maximum of 3 subroutine calls can be nested. If more than 3 calls are nested, a user program execution error occurs, turning on special internal relay M8004 and the ERR LED.
- When using a timer interrupt or interrupt input, include the label number of the interrupt program to be executed when an interrupt occurs. The label numbers stored in data registers D8032 through D8036 specify the interrupt programs for interrupt inputs I2 through I5 and timer interrupt, respectively.
- When more than one interrupt input or timer interrupt is turned on at the same time, interrupt program execution is given priority to inputs I2, I3, I4, timer interrupt, and I5, in that order. If an interrupt is initiated while another interrupt program is executed, the subsequent interrupt program is executed after the prior interrupt is completed. Multiple interrupt programs cannot be executed simultaneously.
- When a communication function is used, such as data link, the interrupt program size must be limited to the executable time listed in the table below.

Communication Function	Baud Rate (bps)	Executable Time of Interrupt Program (µs)
Not Used	—	670 maximum
Used	1200, 2400, 4800, 9600	670 maximum
Used	19200	170 maximum

- If the interrupt program is longer than the value listed above, the entire system performance is affected. The timer and filter functions may not operate correctly, and communication error may be caused in the data link or communication with display units. Make sure that the interrupt program execution time is within the values listed above, referring to the execution times on page A-1. When using high-speed counters, the interrupt program size has to be much smaller.
- When using the data link and interrupt inputs, select 19200 bps for the baud rate of the data link communication.
- Make sure that the execution time of the interrupt program is shorter than interrupt intervals sufficiently.
- Interrupt programs cannot use the following instructions: SOTU, SOTD, TML, TIM, TMH, TMS, CNT, CDP, CUD, SFR, SFRN, ROOT, WKTIM, WKTBL, DISP, DGRD, TXD1/2, RXD1/2, DI, EI, XYFS, CVXTY, CVYTX, PULS1/2, PWM1/2, RAMP, ZRN1/2, PID, DTML, DTIM, DTMH, DTMS, and TTIM.
- If the interrupt program execution time exceeds 670 µs when using the timer interrupt, a user program execution error occurs, turning on special internal relay M8004 and the ERR LED.
- The overhead from interrupt occurrence to interrupt program execution is approximately 60 µs. When using the high-speed counter, the overhead may be extended.

Input Filter

The input filter function is used to reject input noises. The catch input function described in the preceding section is used to read short input pulses to special internal relays. On the contrary, the input filter rejects short input pulses when the MicroSmart is used with input signals containing noises.

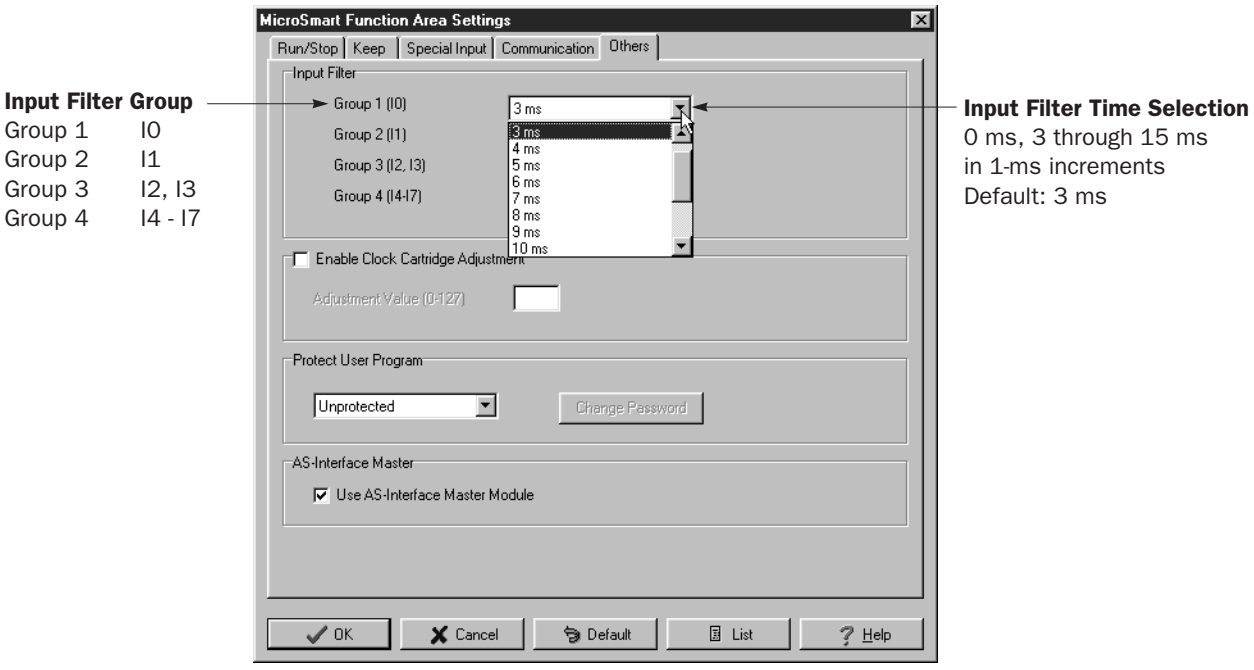
Different input filter values can be selected for inputs I0 through I7 in four groups using the Function Area Settings. Selectable input filter values to pass input signals are 0 ms, and 3 through 15 ms in 1-ms increments. Default value is 3 ms for all inputs I0 through I7. Inputs I10 and above on all-in-one and 20-I/O slim type CPU modules are provided with a fixed filter of 3 ms. Inputs I10 and above on 40-I/O slim type CPU modules and all expansion input modules have a fixed filter of 4 ms. The input filter rejects inputs shorter than the selected input filter value minus 2 ms.

Normal inputs require a pulse width of the filter value plus one scan time to receive input signals. When using the input filter function, select **Normal Input** on the Special Input page in the Function Area Settings.

Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Others** tab.



3. Select an input filter value for each group of inputs.

Input Filter Values and Input Operation

Depending on the selected values, the input filter has three response areas to reject or pass input signals.

Reject area:	Input signals do not pass the filter (selected filter value minus 2 ms).
Indefinite area:	Input signals may be rejected or passed.
Pass area:	Input signals pass the filter (selected filter value).

Example: Input Filter 8 ms

To reject input pulses of 6 ms or less, select input filter value of 8 ms. Then input pulses of 8 ms plus one scan time are accepted correctly at the END processing.

	6 ms	8 ms + 1 scan	
Input	Rejected	Indefinite	Accepted

User Program Protection

The user program in the MicroSmart CPU module can be protected from reading, writing, or both using the Function Area Settings in WindLDR. The read/write protection can be temporarily disabled using a predetermined password.

Upgraded CPU modules with system program version 210 or higher have an option for read protection without a password, making it possible to inhibit reading completely.

⚠ Warning

- Before proceeding with the following steps, make sure to note the password, which is needed to disable the user program protection. If the user program in the MicroSmart CPU module is write- or read/write-protected, the user program cannot be changed without the password.

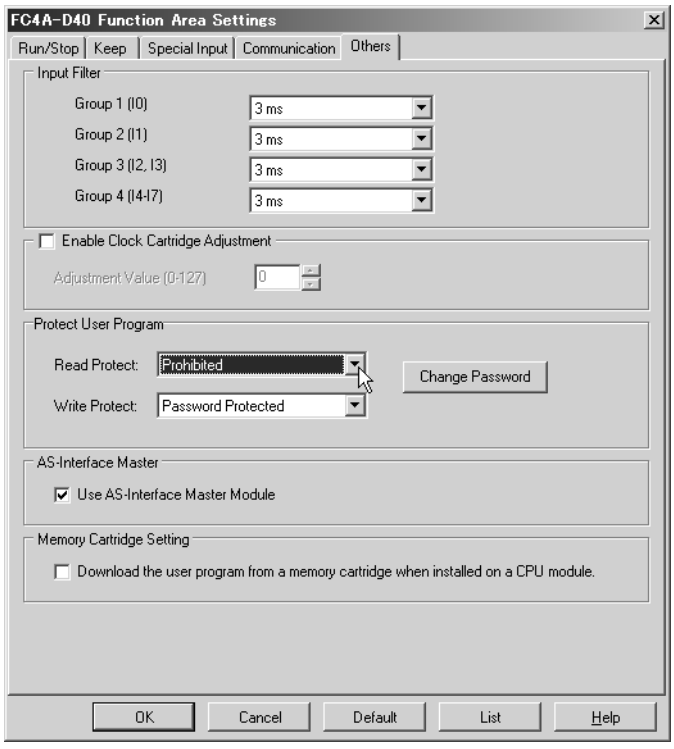
⚠ Caution

- If the user program is read-protected without using a password, the read protection cannot be temporarily disabled using the password, thus the user program cannot be read out by any means. To disable the read protection, download another user program without user program protection.

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Settings dialog box appears.

2. Select the **Others** tab.



3. Under **Protect User Program**, select required protect modes for **Read Protect** and **Write Protect** in the pull-down list.

Unprotected:	The user program in the CPU module can be read and written without a password.
Password Protected:	Prevents unauthorized copying or inadvertent replacement of the user program. The protection can be temporarily disabled using a predetermined password.
Prohibited:	Prevents copying of the user program completely. This option is available for read protection only and can not be temporarily disabled using a password. To select this option, use a CPU module with system program ver. 210 or higher and WindLDR ver 5.31 or higher

4. When a password protect mode is selected, the Password Setting dialog box appears.

   Enter a password of 1 through 8 ASCII characters from the key board in the **Password** field, and enter the same password in the **Confirm Password** field. Click the **OK** button to return to the Others tab page.

5. Click the **OK** button and download the user program to the MicroSmart after changing any of these settings.

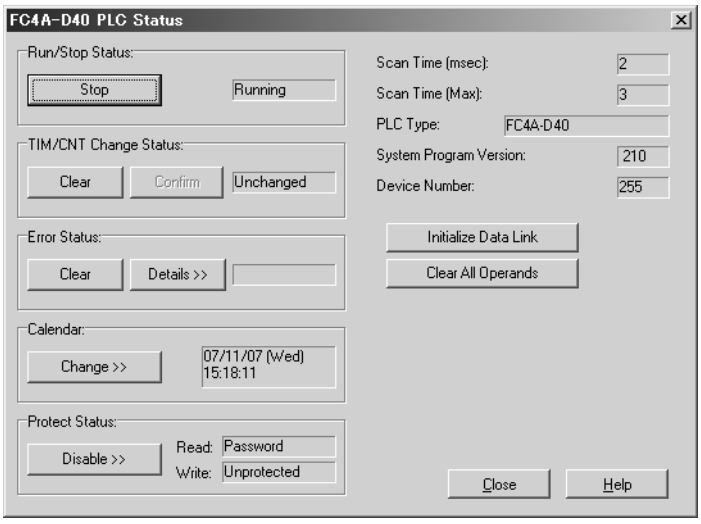
Disabling Protection

When the user program is password-protected against read and/or write, the protection can be temporarily disabled using WindLDR.

If the user program is read-prohibited, the read protection cannot be disabled, thus the user program cannot be read out by any means. To disable the read protection, download another user program without user program protection.

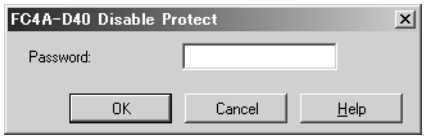
1. From the WindLDR menu bar, select **Online > Monitor**. The monitor mode is enabled.

2. From the WindLDR menu bar, select **Online > PLC Status**. The PLC Status dialog box appears.



3. Under the **Protect Status** in the PLC Status dialog box, click the **Disable** button. The Disable Protect dialog box appears.

4. Enter the password, and click the **OK** button.



The user program protection is disabled temporarily, and reading is allowed once.

Enabling Protection

When the CPU module is powered up again, the protection designated in the user program takes effect again.

For read or read/write protect, once the user program is uploaded, the protection automatically takes effect again. For write protect, the protection designated in the newly downloaded user program takes effect.

To change the protection permanently, change the protection settings and download the user program.

Constant Scan Time

The scan time may vary whether basic and advanced instructions are executed or not depending on input conditions to these instructions. The scan time can be made constant by entering a required scan time preset value into special data register D8022 reserved for constant scan time. When performing accurate repetitive control, make the scan time constant using this function. The constant scan time preset value can be between 1 and 1,000 ms.

The scan time error is ±1 ms of the preset value normally. When the data link or other communication functions are used, the scan time error may be increased to several milliseconds.

When the actual scan time is longer than the scan time preset value, the scan time cannot be reduced to the constant value.

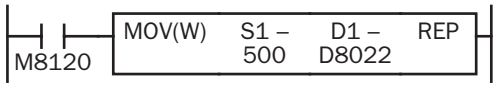
Special Data Registers for Scan Time

In addition to D8022, three more special data registers are reserved to indicate current, maximum, and minimum scan time values.

D8022	Constant Scan Time Preset Value (1 to 1,000 ms)
D8023	Scan Time Current Value (ms)
D8024	Scan Time Maximum Value (ms)
D8025	Scan Time Minimum Value (ms)

Example: Constant Scan Time

This example sets the scan time to a constant value of 500 ms.



M8120 is the initialize pulse special internal relay.

When the CPU starts operation, the MOV (move) instruction stores 500 to special data register D8022.

The scan time is set to a constant value of 500 ms.

Partial Program Download

Normally, the CPU module has to be stopped before downloading a user program. The all-in-one 16- and 24-I/O type CPU modules and all slim type CPU modules have run-time program download capabilities to download a user program containing small changes while the CPU is running in either 1:1 or 1:N computer link system. This function is particularly useful to make small modifications to the user program and to confirm the changes while the CPU is running. The all-in-one 10-I/O type CPU module does not have this functionality.

Before performing the partial program download during operation, a user program has to be downloaded to the CPU module using the ordinary program download. Add or delete a part of the same user program, or make small changes to the same user program using WindLDR, and download the modified user program while the CPU is running to confirm the changes on-line.

Another method of using this feature is: upload the user program from the CPU module to WindLDR, make changes, and download the modified user program using the partial program download while the CPU is running.

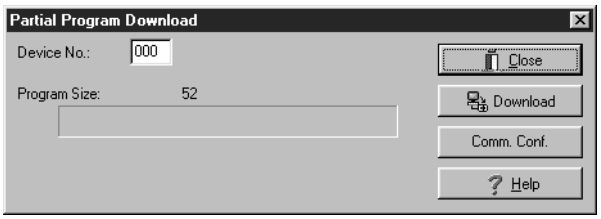
In either case, do not convert the ladder diagram to mnemonic codes to generate a code file (**Compile > Convert Ladder**) before using the partial program download. WindLDR attaches a unique code to every code file when a ladder diagram is converted to mnemonic codes. When the partial program download is attempted, WindLDR compares the unique codes of the user programs in the CPU module and currently opened on WindLDR. Only when WindLDR verifies that the unique codes are identical, the partial program download is enabled.

⚠ Caution

- The partial program download may cause unexpected operation of the MicroSmart. Before starting the partial program download, make sure of safety after understanding the function correctly.
- If a user program syntax error or user program writing error occurs during the partial program download, the CPU module is stopped and all outputs are turned off, which may cause hazards depending on the application.

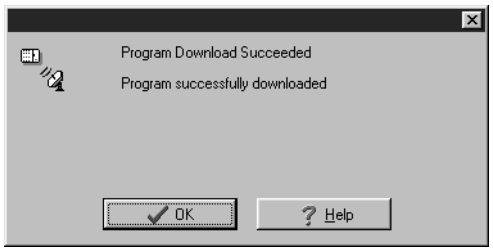
Programming WindLDR

1. Make changes to the user program using WindLDR. From the WindLDR menu bar, select **Online > Partial Program Download** while the CPU module is running. The Partial Program Download dialog box appears.



2. When using the MicroSmart in a 1:N computer link system, enter the device number of the CPU module in the Device No. field. When using in a 1:1 computer link system, leave the default value in the Device No. field.

3. Click the **Download** button to start the partial program download.



4. When the partial program download is completed successfully, the above dialog box is displayed. Click the **OK** button to return to the WindLDR editing screen.

Using Partial Program Download

The partial program download function can download a maximum of 600 bytes (100 steps) of user program. When the modified rungs of the user program exceed 600 bytes, the partial program download cannot be used. Make sure that modification is within 600 bytes.

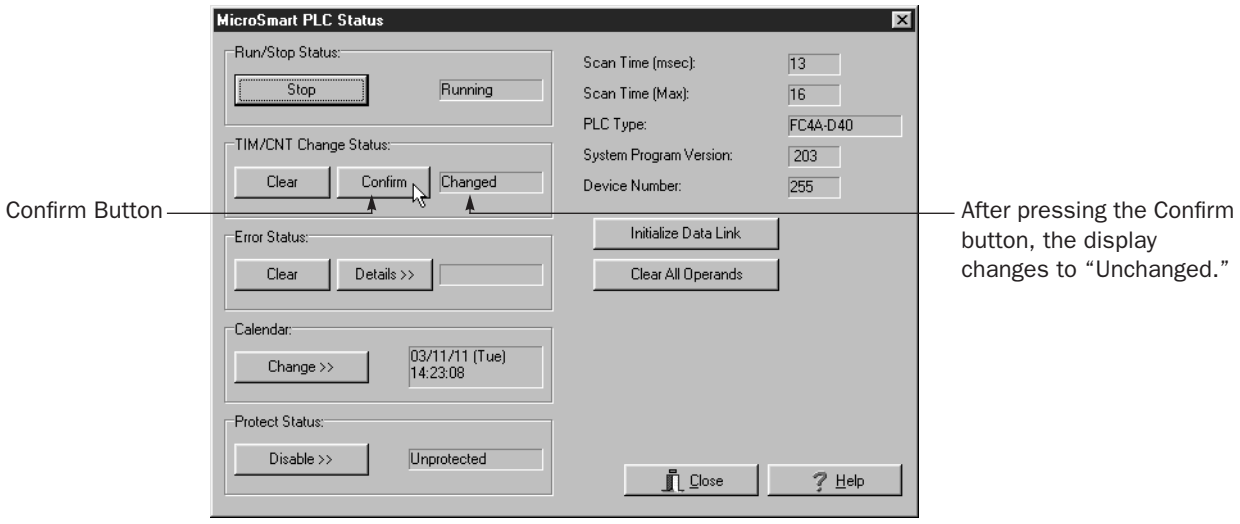
When modifying two or more rungs of a user program, make sure that the difference between the first address and last address of the modifications is within 600 bytes (100 steps).

While the partial program download is in progress, the scan time extends for several scans by approximately 200 ms per scan.

While the partial program download is in progress, the statuses of outputs, internal relays, shift registers, timers, counters, and data registers are unchanged.

When timer or counter preset values in the CPU RAM have been changed by using WindLDR (**Online > Point Write**), the new preset values are cleared if the downloaded user program includes changes of the timer or counter instructions, and the preset values of the downloaded user program take effect. Similarly, when a timer or counter is designated as a destination operand of an advanced instruction and the timer/counter preset value has been changed by the advanced instruction, the new preset value is also cleared. However, if the downloaded user program does not include changes of timer or counter instructions, the new preset values remain in effect.

If you do not want to clear the new preset values during the partial program download, you can import the new preset values to the user program. Access the PLC Status dialog box from the Online menu in the monitoring mode. Then click the **Confirm** button in the TIM/CNT Change Status field. (The displayed status will switch from **Changed** to **Unchanged**.) Upload the user program, which has new preset values in place of the original preset values. Make changes to the uploaded user program, then perform the partial program download. Note that the **Confirm** button has effect on both timer and counter preset values.



While the partial program download is in progress, interrupt inputs, timer interrupt, and catch inputs are disabled temporarily until the downloaded user program is loaded to the user program area (RAM) in the CPU module.

When changes are made to user communication instructions, pulse instructions, input filters, catch inputs, interrupt inputs, timer interrupt, expansion data registers, high-speed counters, or Function Area Settings, download the entire user program. If the partial program download includes these changes, the user program cannot run correctly.

When you want to delete a rung and perform the partial program download, use the disable rung command (**Right Mouse > Disable Rung**) on WindLDR instead. When a rung is deleted, the partial program download may not be performed because each rung contains information to enable partial program download.

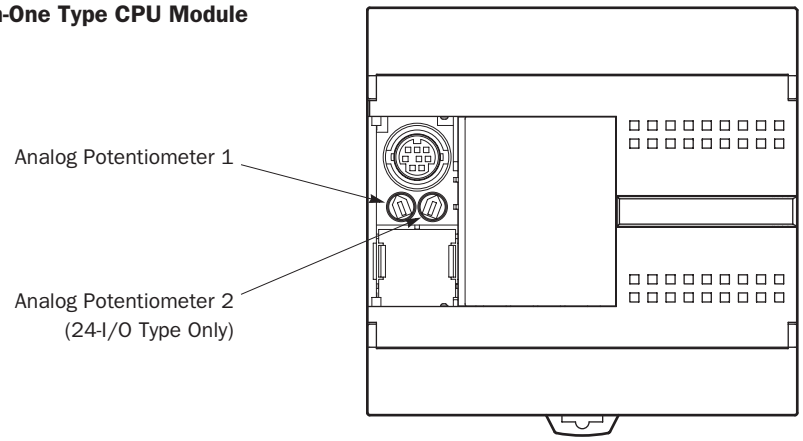
When a user program has been downloaded with comment data to the CPU module, the partial program download cannot be performed. Make sure that the CPU module contains a user program downloaded without comment data to enable partial program download.

Analog Potentiometers

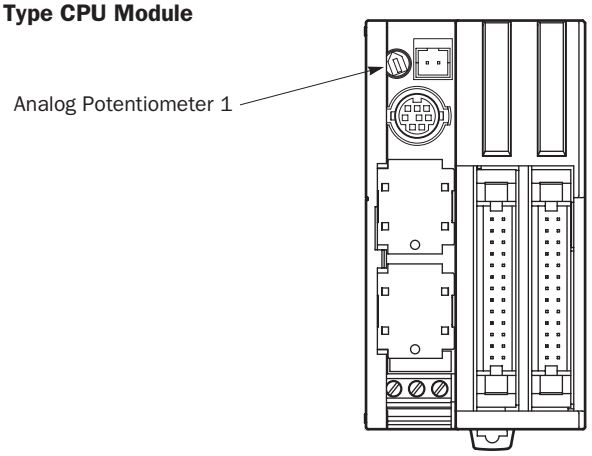
The all-in-one 10- and 16-I/O type CPU modules and every slim type CPU module have one analog potentiometer. Only the 24-I/O type CPU module has two analog potentiometers. The values (0 through 255) set with analog potentiometers 1 and 2 are stored to data registers D8057 and D8058, respectively, and updated in every scan.

The analog potentiometer can be used to change the preset value for a timer or counter.

All-in-One Type CPU Module



Slim Type CPU Module

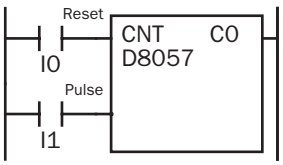


Special Data Registers for Analog Potentiometers

CPU Module	Analog Potentiometer 1	Analog Potentiometer 2
FC4A-C24R2 and FC4A-C24R2C	D8057	D8058
Other CPU Modules	D8057	—

Example: Changing Counter Preset Value Using Analog Potentiometer

This example demonstrates a program to change a counter preset value using analog potentiometer 1.



Analog potentiometer 1 value is stored to data register D8057, which is used as a preset value for counter C0.

The preset value is changed between 0 and 255 using the potentiometer.

Analog Voltage Input

Every slim type CPU module has an analog voltage input connector. When an analog voltage of 0 through 10V DC is applied to the analog voltage input connector, the signal is converted to a digital value of 0 through 255 and stored to special data register D8058. The data is updated in every scan.

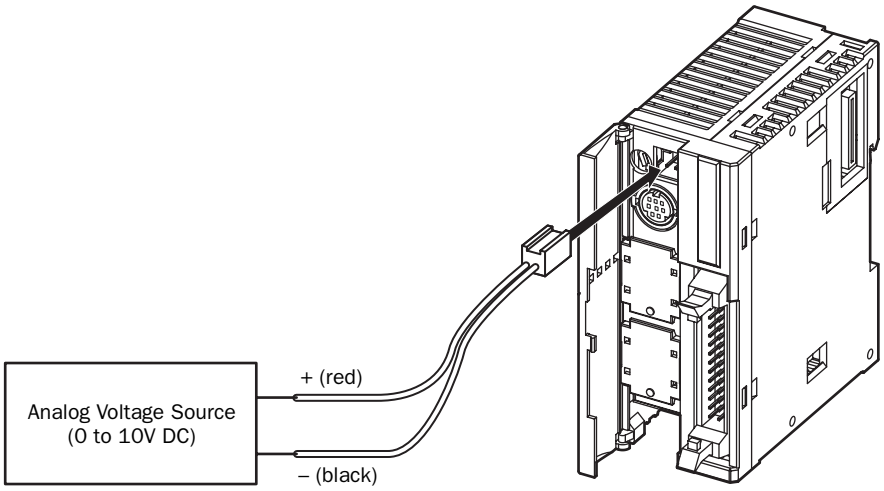
Special Data Register for Analog Voltage Input

CPU Module	Analog Voltage Input Data
Slim Type CPU Modules	D8058

To connect an external analog source, use the attached cable.

The cable is also available optionally.

Cable Name	Type No.
Analog Voltage Input Cable (1m/3.28 ft. long)	FC4A-PMAC2P (package quantity 2)



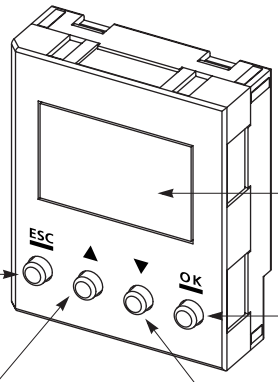
HMI Module

This section describes the functions and operation of the optional HMI module (FC4A-PH1). The HMI module can be installed on any all-in-one type CPU module, and also on the HMI base module mounted next to any slim type CPU module. The HMI module makes it possible to manipulate the RAM data in the CPU module without using the Online menu options in WindLDR. For details about the specifications of the HMI module, see page 2-60.

HMI module functions include:

- Displaying timer/counter current values and changing timer/counter preset values
- Displaying and changing data register values
- Setting and resetting bit operand statuses, such as inputs, outputs, internal relays, and shift register bits
- Displaying and clearing error data
- Starting and stopping the PLC
- Displaying and changing calendar/clock data (only when using the clock cartridge)
- Confirming changed timer/counter preset values

Parts Description



Display Screen
The liquid crystal display shows menus, operands, and data.

ESC Button
Cancels the current operation, and returns to the immediately preceding operation.

OK Button
Goes into each control screen, or enters the current operation.

▲ Button
Scrolls up the menu, or increments the selected operand number or value.

▼ Button
Scrolls down the menu, or decrements the selected operand number or value.

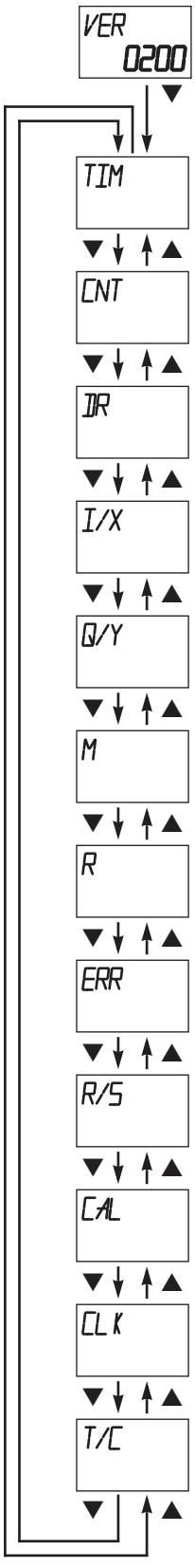
⚠ Caution

- Power up the MicroSmart CPU module after installing the HMI module. If the HMI module is installed or removed while the MicroSmart is powered up, the HMI module may fail to operate correctly.
- If an invalid operand or a value over 65535 is entered, the display screen flashes to signal an error. When an error screen displays, press the **ESC** button and repeat the correct key operation.

Key Operation for Scrolling Menus after Power-up

The chart below shows the sequence of scrolling menus using the ▼ and ▲ buttons on the HMI module after power-up.

While a menu screen is shown, press the **OK** button to enter into each control screen where operand numbers and values are selected. For details of each operation, see the following pages.



Initial Screen
Indicates the PLC system program version or the same menu as when the PLC was powered down, depending on the value stored in special data register D8068 (see the next page).

Press the ▼ button to switch to the timer menu.

Timer Menu
Displays a timer current value, and changes the timer preset value.

Counter Menu
Displays a counter current value, and changes the counter preset value.

Data Register Menu
Displays a data register value, and changes the data register value.

Input Menu
Displays an input status, and sets or resets the input.

Output Menu
Displays an output status, and sets or resets the output.

Internal Relay Menu
Displays an internal relay status, and sets or resets the internal relay.

Shift Register Menu
Displays a shift register bit status, and sets or resets the shift register bit.

Error Menu
Displays general error codes, and clears the general error codes.

Run/Stop Menu
Displays the run/stop status of the PLC, and starts or stops the PLC.

Calendar Menu
Displays and changes the calendar data.

Clock Menu
Displays and changes the clock data.

Timer/Counter Changed Preset Value Confirm Menu
Confirms changed timer/counter preset values.
(The changed preset values in the MicroSmart CPU module RAM are written to EEPROM.)

Selection of HMI Module Initial Screen

Special data register D8068 is available on upgraded CPU modules with system program version shown in the table below. For the procedure to confirm the system program version of the CPU module, see page 29-1.

CPU Module	All-in-One Type			Slim Type	
	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
System Program Version	203 or higher	202 or higher	202 or higher	202 or higher	201 or higher

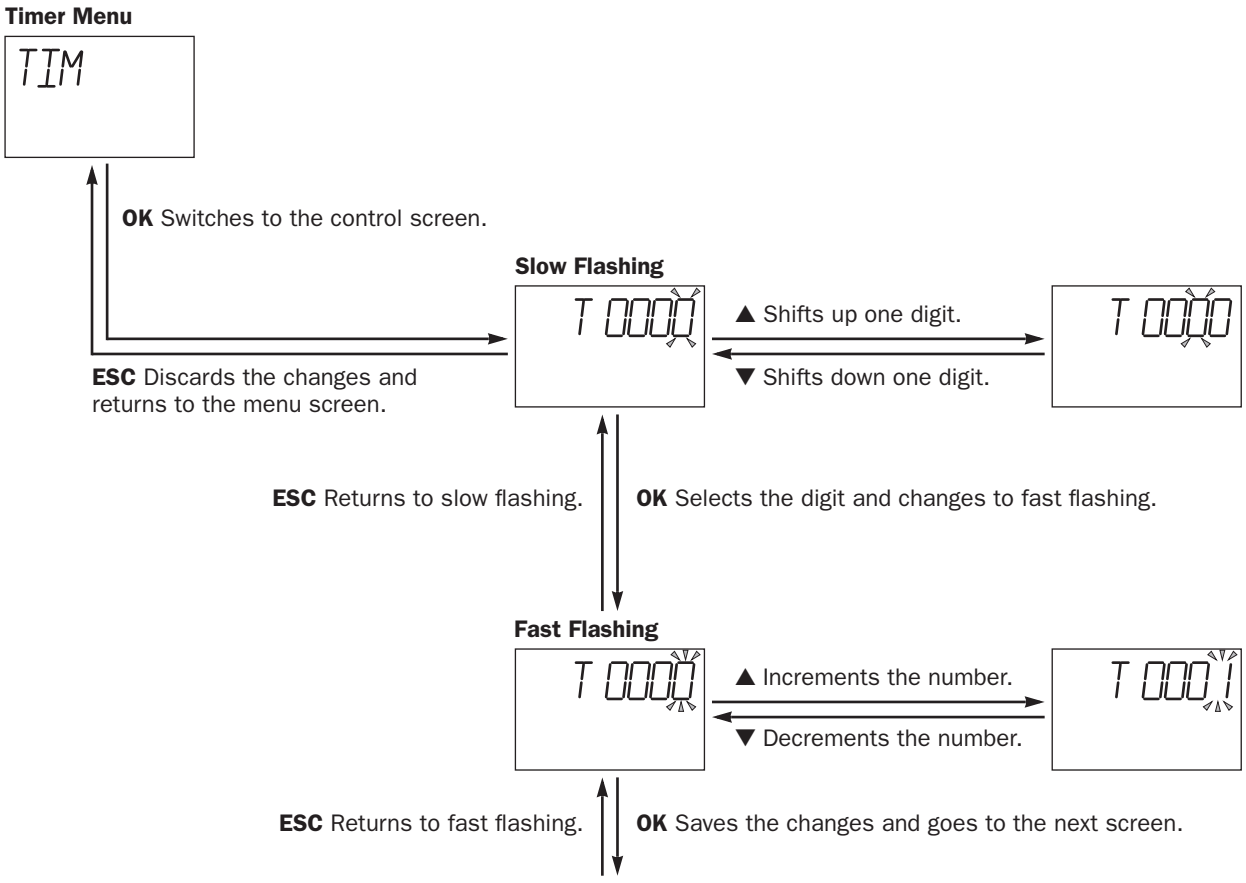
D8068 can be used to select the initial screen display of the HMI module when the CPU module is powered up.

Data Register	Value	Description
D8068	0, 2 through 65535	Mode 1: Indicates the PLC program version each time the PLC is powered up.
	1	Mode 2: Indicates the same menu as when the PLC was shut down.

When a keep data error occurs, mode 1 is enabled regardless of the value stored in data register D8068.

Key Operation for Selecting Operand Number

When the **OK** button is pressed while a menu screen is shown, the screen switches to the control screen of the menu. For example, while the timer menu is on the display, pressing the **OK** button switches the screen to the timer control screen, where operand numbers and values are selected. For operation examples, see the following pages.



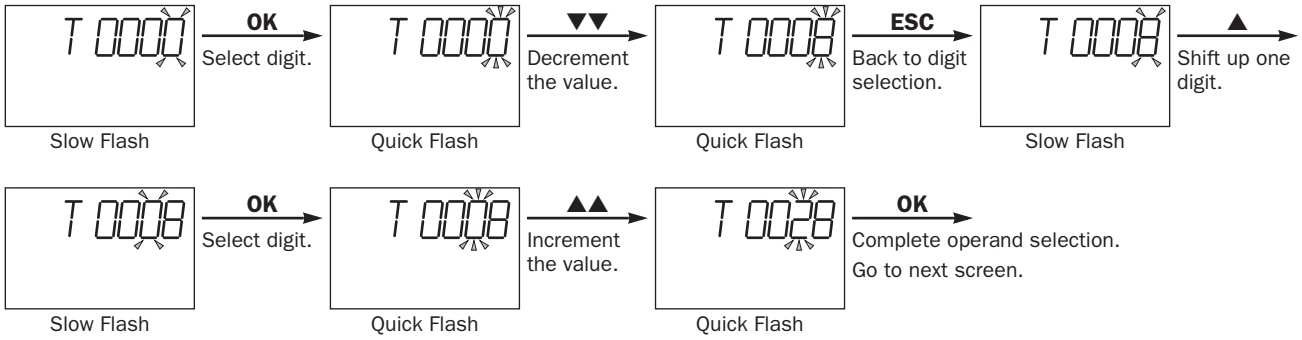
Displaying Timer/Counter Current Values and Changing Timer/Counter Preset Values

This section describes the procedure for displaying a timer current value and for changing the timer preset value for an example. The same procedure applies to counter current values and preset values.

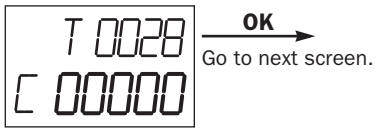
Example: Change timer T28 preset value 820 to 900

1. Select the Timer menu.

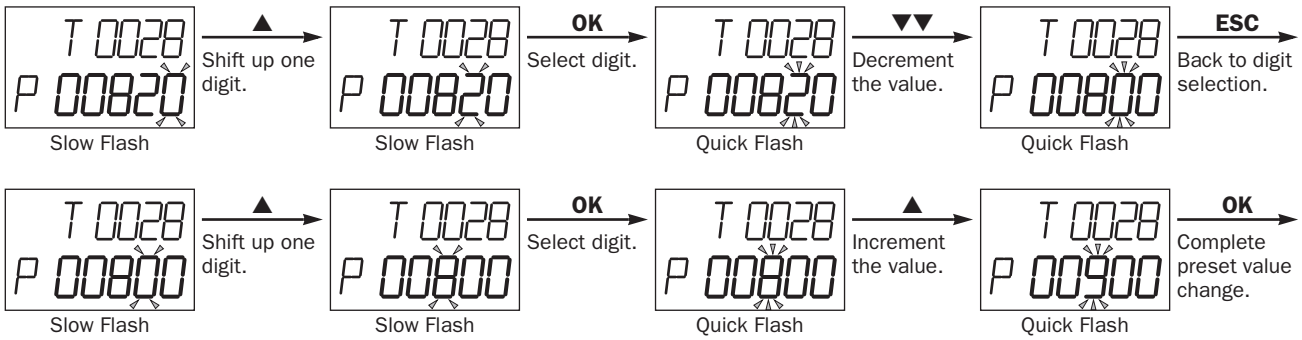
2. Select the operand number.



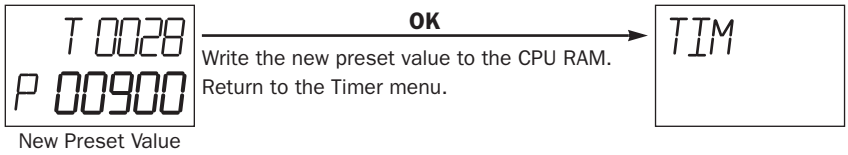
3. The current value of the selected timer number is displayed.



4. The preset value of the selected timer number is displayed. Change the preset value to 900 as described below.



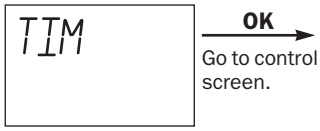
5. The changed preset value is displayed without flashing. Write the new preset value to the CPU module RAM.



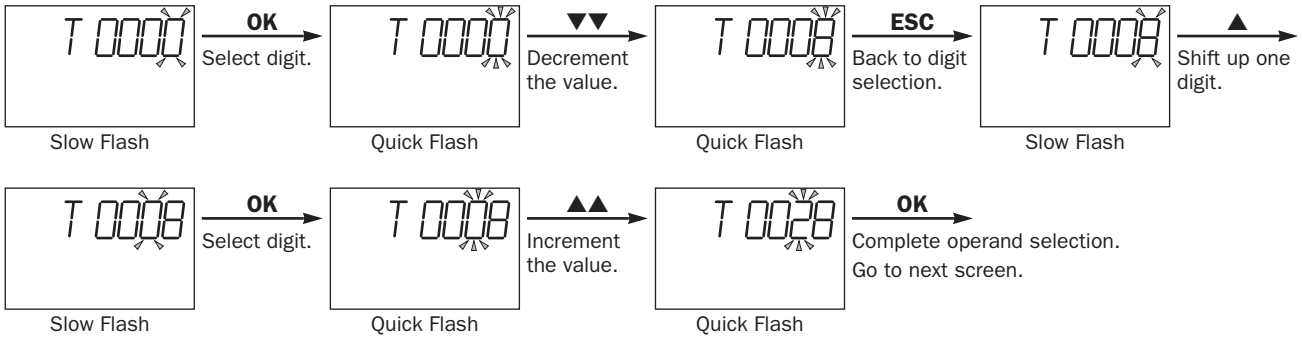
Note: The changed timer/counter preset values are stored in the MicroSmart CPU module RAM and backed up for 30 days by a lithium backup battery. If required, the changed preset values can be written from the MicroSmart CPU module RAM to the EEPROM using the Timer/Counter Changed Preset Value Confirm menu described on page 5-36. For the data movement in the CPU module, see page 7-13.

Example: When timer T28 preset value is designated using a data register

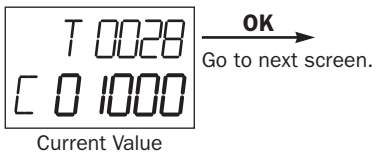
1. Select the Timer menu.



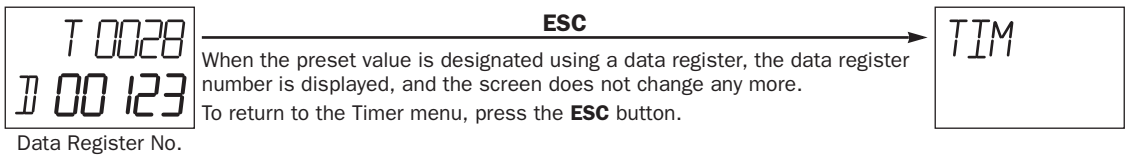
2. Select the operand number.



3. The current value of the selected timer number is displayed.



4. The data register number designated as a preset value is displayed.

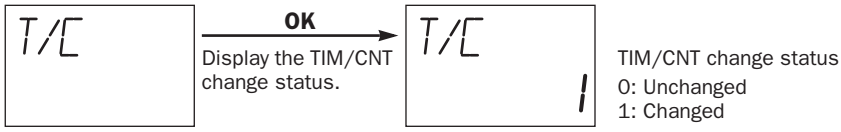


Confirming Changed Timer/Counter Preset Values

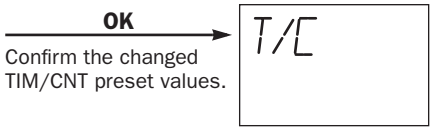
This section describes the procedure for writing changed timer/counter preset values from the MicroSmart CPU module RAM to the EEPROM. This operation writes the changed preset values of both timers and counters at once.

The changed timer/counter preset values are stored in the MicroSmart CPU module RAM and backed up for 30 days by a lithium backup battery. If required, the changed preset values can be written to the MicroSmart CPU module EEPROM as described below. For the data movement in the CPU module, see page 7-13.

1. Select the Timer/Counter Changed Preset Value Confirm menu.



2. Confirm the changed timer/counter preset values, and write the changes from the RAM to the EEPROM.



The Timer/Counter Changed Preset Value Confirm menu is restored.

To abort confirming the changed timer/counter preset values, press the **ESC** button instead of the **OK** button; the Timer/Counter Changed Preset Value Confirm menu is restored.

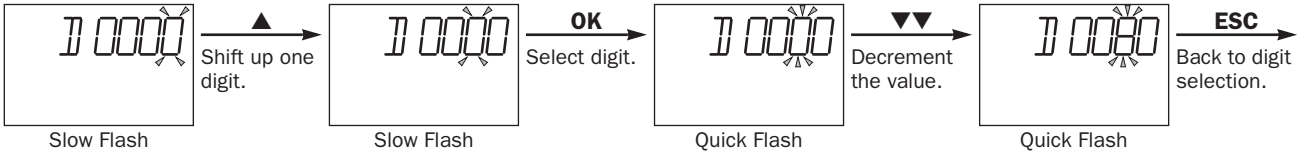
Displaying and Changing Data Register Values

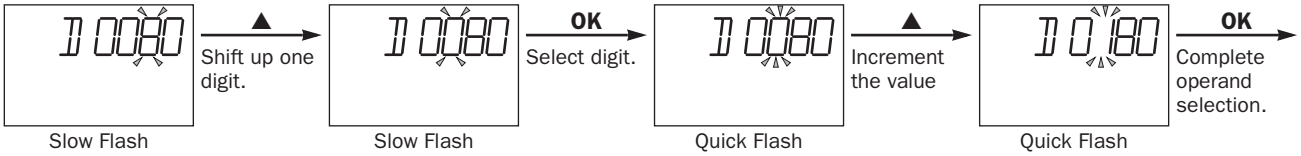
This section describes the procedure for displaying and changing the data register value.

Example: Change data register D180 value to 1300

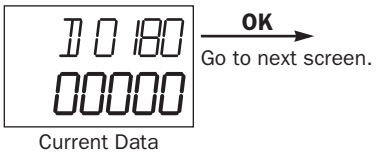
1. Select the Data Register menu.

2. Select the operand number.

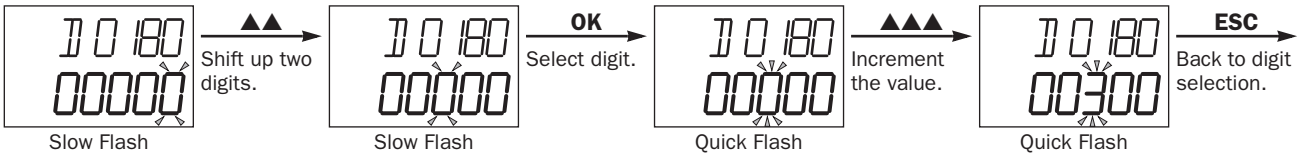


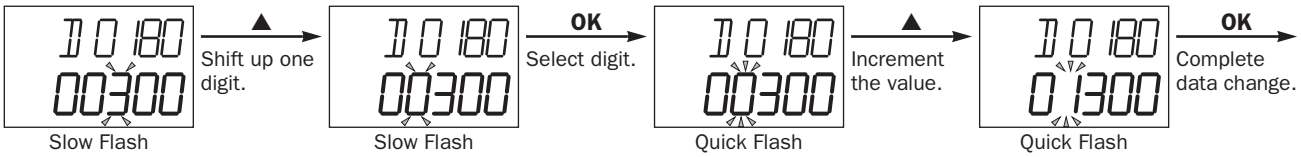


3. The data of the selected data register number is displayed.

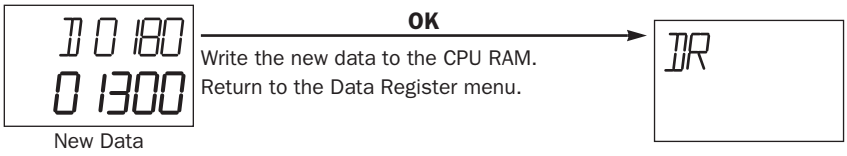


4. Change the data to 1300 as described below.





5. The changed data is displayed without flashing. Save the changes.



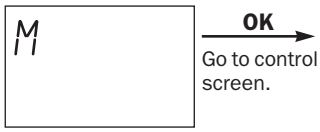
Setting and Resetting Bit Operand Status

Bit operand statuses, such as inputs, outputs, internal relays, and shift register bits, can be displayed, and set or reset using the MHI module.

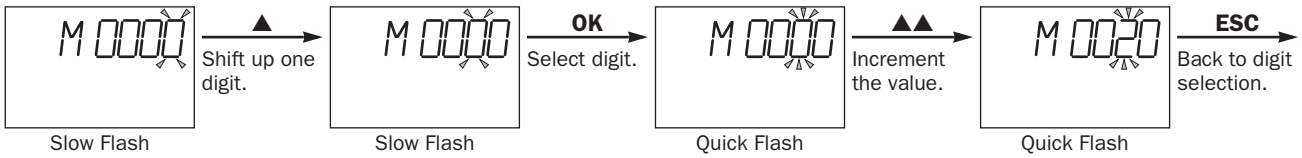
This section describes the procedure for displaying an internal relay status and for setting the internal relay for an example. The same procedure applies to inputs, outputs, and shift register bits.

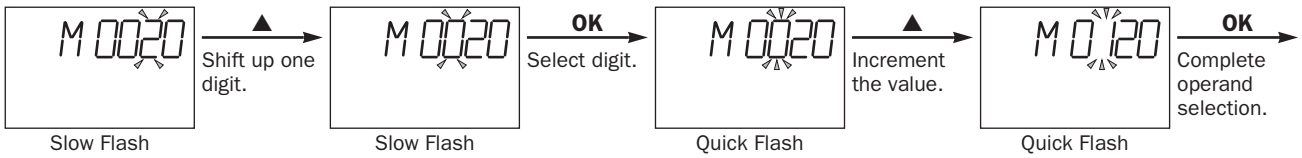
Example: Set internal relay M120

1. Select the Internal Relay menu.

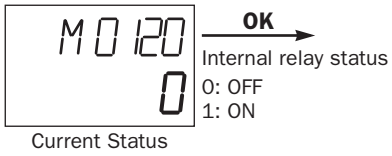


2. Select the operand number.

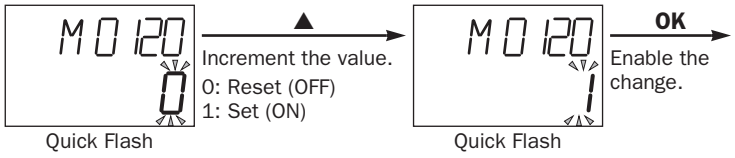




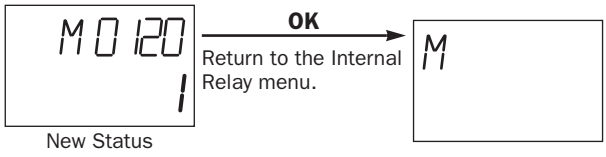
3. The status of the selected internal relay number is displayed.



4. Select 1 (set) or 0 (reset) using the ▲ or ▼ button.



5. The changed status is displayed without flashing.

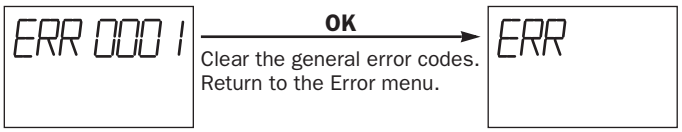


Displaying and Clearing Error Data

This section describes the procedure for displaying general error codes and for clearing the general error codes.

1. Select the Error menu.

2. General error codes are displayed. Clear the general error codes.



To abort clearing the general error codes, press the **ESC** button instead of the **OK** button; the Error menu is restored.

For details about general error codes, see page 29-3.

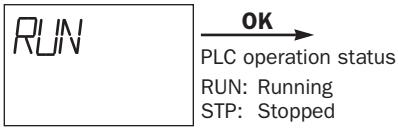
Starting and Stopping the PLC

This section describes the procedure for starting and stopping the PLC operation using the HMI module.

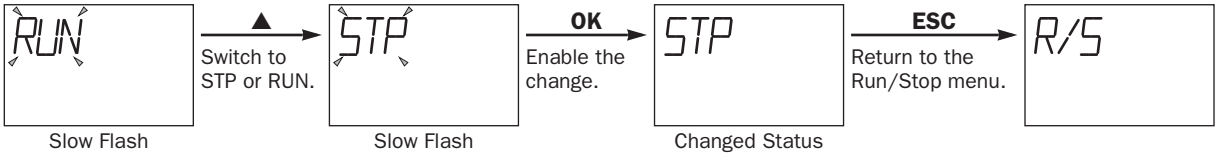
Note: The procedure described below turns on or off start control special internal relay M8000 to start or stop the PLC operation. When a stop input is designated, the PLC cannot be started or stopped by turning start control special internal relay M8000 on or off; the procedure described below does not work. See page 4-3.

1. Select the Run/Stop menu.

2. The PLC operation status is displayed.



3. Select RUN or STP to start or stop the PLC operation, respectively, using the ▲ or ▼ button.



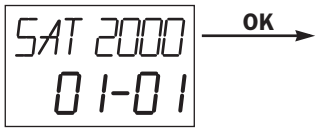
Displaying and Changing Calendar Data (only when using the clock cartridge)

When an optional clock cartridge (FC4A-PT1) is installed in the MicroSmart CPU module, the calendar data of the clock cartridge can be displayed and changed using the HMI module as described in this section.

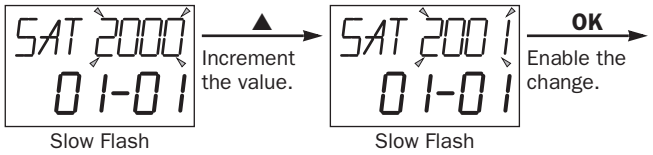
Example: Change calendar data from Saturday, 01/01/2000 to Wednesday, 04/04/2001

1. Select the Calendar menu.

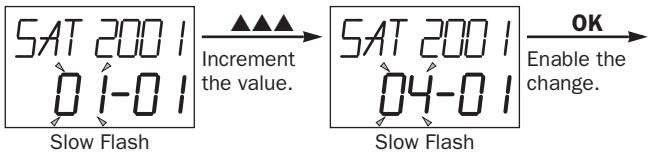
2. The calendar data is displayed.



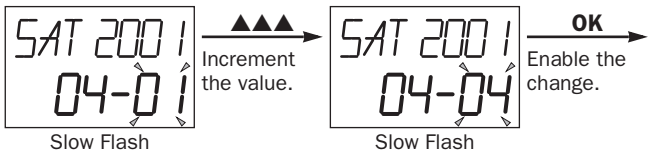
3. Change the year data using the ▲ or ▼ button.



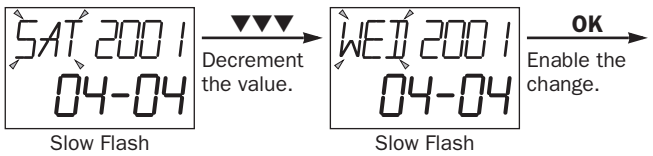
4. Change the month data using the ▲ or ▼ button.



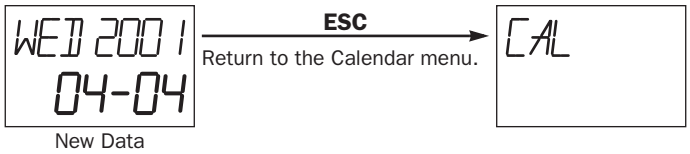
5. Change the day data using the ▲ or ▼ button.



6. Change the day of week data using the ▲ or ▼ button.



7. The new calendar data is displayed without flashing.



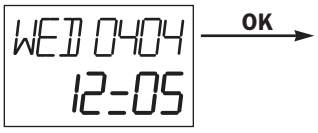
Displaying and Changing Clock Data (only when using the clock cartridge)

When an optional clock cartridge (FC4A-PT1) is installed in the MicroSmart CPU module, the clock data of the clock cartridge can be displayed and changed using the HMI module as described in this section.

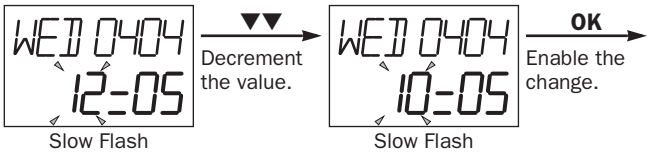
Example: Change clock data from 12:05 to 10:10

1. Select the Clock menu.

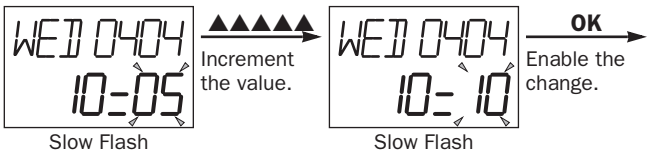
2. The clock data is displayed.



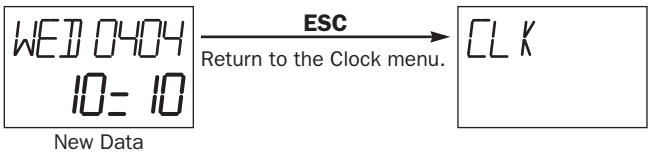
3. Change the hour data using the ▲ or ▼ button.



4. Change the minute data using the ▲ or ▼ button.



5. The new clock data is displayed without flashing.



Expansion Data Registers

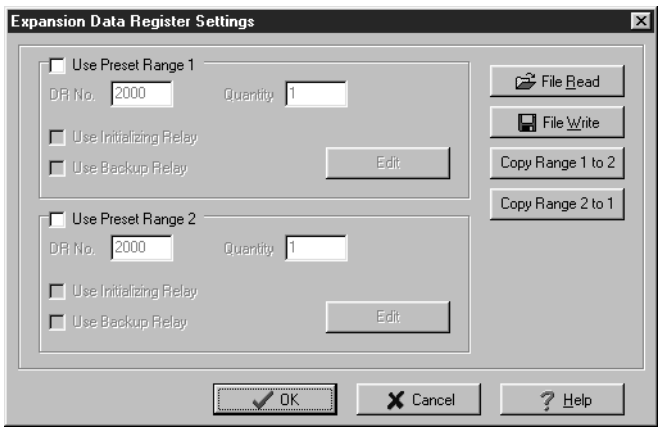
Slim type CPU modules FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3 have expansion data registers D2000 through D7999. These expansion data registers are normally used as ordinary data registers to store numerical data while the CPU module is executing a user program. In addition, numerical data can be set to designated ranges of expansion data registers using the expansion data register editor on WindLDR. When the user program is downloaded from WindLDR to the CPU module, the preset values of the expansion data registers are also downloaded to the EEPROM in the CPU module. Each time the CPU is powered up, the preset values of the expansion data registers stored in the EEPROM are loaded to the RAM and the user program in the RAM is executed.

Since the data in the EEPROM is non-volatile, the preset values of the expansion data registers are maintained semi-permanently and restored in the RAM each time the CPU is powered up. This feature is useful when particular numerical data must not be lost. Furthermore, data register values can be easily entered in the form of either numbers or character strings using the expansion data register editor on WindLDR.

Programming WindLDR

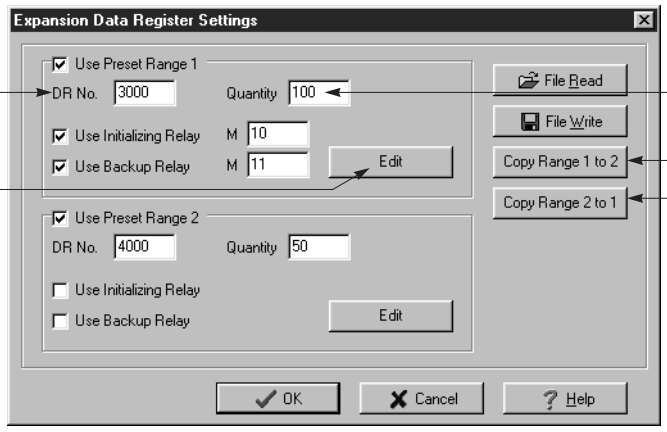
1. From the WindLDR menu bar, select **Configure > Expansion Data Register Settings**.

   The Expansion Data Register Settings dialog box appears.



2. Click the check box to use the preset range 1 or 2.

Among expansion data registers D2000 through D7999, two ranges can be specified for preset data registers.



First Data Register No.

Edit Button
Go to the expansion data register editor screen.

Quantity of Data Registers

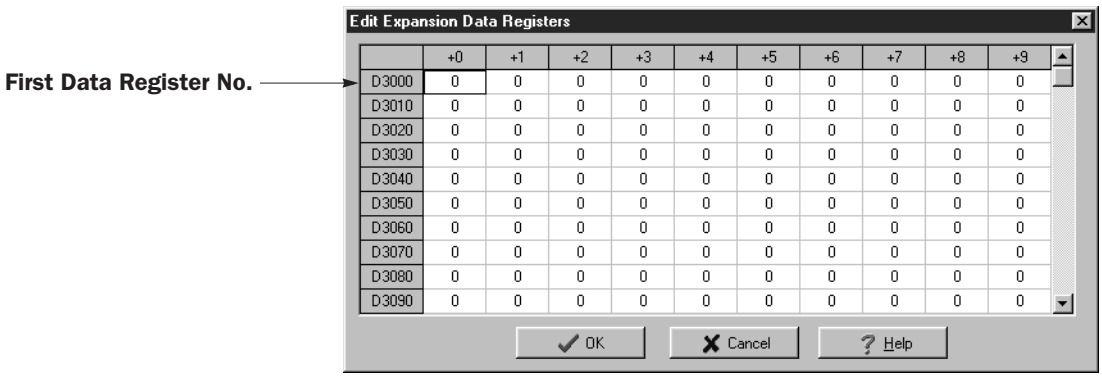
Copy Range
Copy the data between User Preset Ranges 1 and 2.

Use Preset Range 1 or 2: Click the check box, and type the first data register number in the **DR No.** box and the quantity of data registers to store preset values in the **Quantity** box.

Use Initializing Relay: Click the check box and specify an internal relay number to use as an initializing relay. When the initializing relay is turned on while the CPU is powered up, the preset values of the expansion data registers in the EEPROM are loaded to the RAM.

Use Backup Relay: Click the check box and specify an internal relay number to use as a backup relay. When the backup relay is turned on while the CPU is powered up, the values of the preset expansion data registers in the RAM overwrite the preset values in the EEPROM.

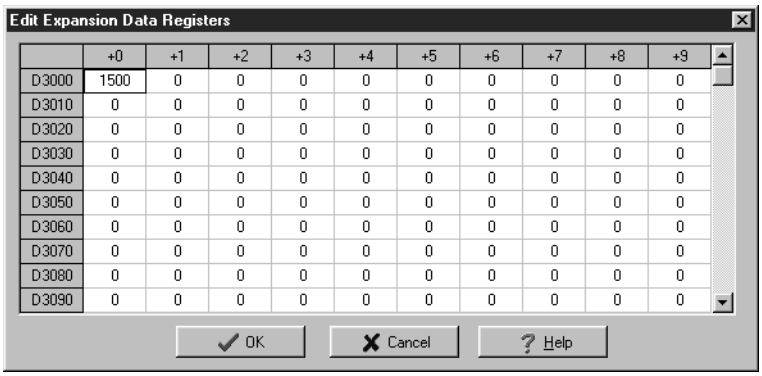
3. Click the **Edit** button. The Edit Expansion Data Registers screen appears.



The specified quantity of data registers are reserved to store preset values in the Edit Expansion Data Registers screen. You can enter numerical values to these data registers individually, in the form of character strings, or fill the same value to consecutive data registers.

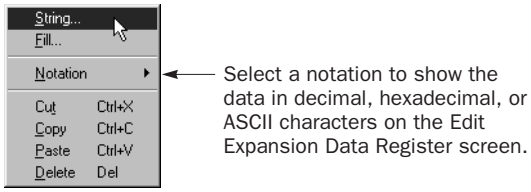
Enter Individual Values

Click the data register number in the Edit Expansion Data Registers screen where you want to enter a numerical value, and type a value 0 through 65535. When finished, click **OK** to return to the Expansion Data Register Settings dialog box.



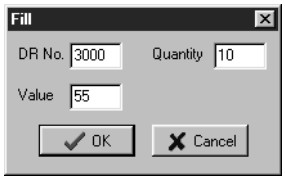
Enter Character String

Click the right mouse button at the data register number in the Edit Expansion Data Registers screen where you want to enter a character string. A pop-up menu appears. Select **String** in the pop-up menu, then the String dialog box appears. Type required characters, and click **OK**. The entered characters are converted in pairs into ASCII decimal values and stored to data registers, starting with the selected data register number.



Fill Same Value

Click the right mouse button at the data register number in the Edit Expansion Data Registers screen where you want to enter numerical values. A pop-up menu appears. Select **Fill** in the pop-up menu, then the Fill dialog box appears. Type the first data register number, the quantity of data registers, and the value. When finished, click **OK**. The value is entered to consecutive data registers.



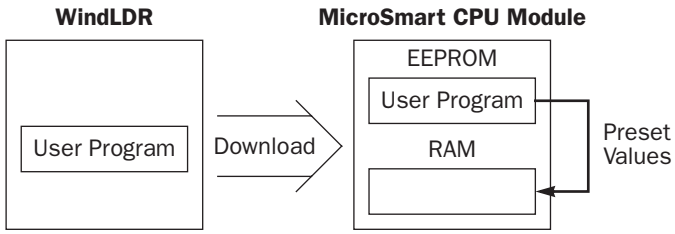
4. After editing the preset values of expansion data registers, download the user program to the CPU module since these settings relate to the user program.

Data Movement of Preset Data Registers

Like preset values for timers and counters (page 7-13), the preset data of expansion data registers can be changed in the RAM, the changed data can be cleared, and also stored to the EEPROM. The data movement is described below.

At Power-up and User Program Download

When the user program is downloaded to the CPU module, the data of preset data registers are also downloaded to the EEPROM. Each time the CPU is powered up, the data of preset data registers are loaded to the RAM. If the data of the expansion data registers have been changed as a result of advanced instructions or through communication, the changed data is cleared and initialized with the data of the preset data registers when the CPU is powered up again.

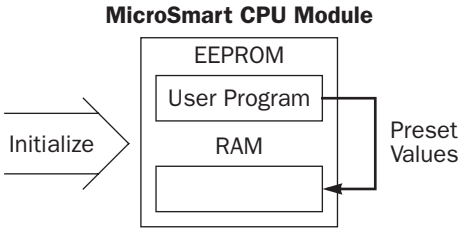


Since expansion data registers D2000 through D7999 are all "keep" types, the data in ordinary data registers are retained when the CPU is powered down.

Initializing Relay

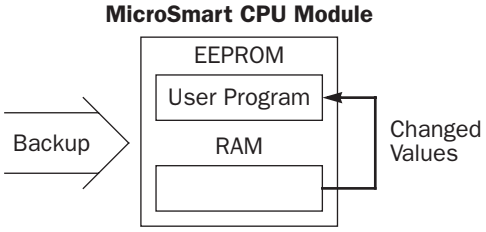
When the internal relay designated as an initializing relay is turned on, the data of preset data registers are loaded to the RAM as is the case when the CPU is powered up.

When the initialization is complete, the initializing relay is turned off automatically. When a user program is used to turn on the initializing relay, use a SOTU or SOTD to make sure that the initializing relay turns on for one scan only. When an initializing relay is not designated, the initialization cannot be performed.



Backup Relay

When the internal relay designated as a backup relay is turned on, the data of preset data registers are written from the RAM to the EEPROM as is the case with confirming changed timer/counter preset values. When the CPU is powered up again, the new data is loaded from the EEPROM to the RAM. When the user program is uploaded to WindLDR, the new data is also uploaded to the expansion data registers.



When the backup is complete, the backup relay is turned off automatically. When a user program is used to turn on the backup relay, use a SOTU or SOTD to make sure that the backup relay turns on for one scan only. When a backup relay is not designated, the backup cannot be performed.

Special Internal Relays for Expansion Data Registers

While data write from the RAM to expansion data register preset range 1 or 2 in the EEPROM is in progress, special internal relay M8026 or M8027 turns on, respectively. When data write is complete, the special internal relay turns off.

Notes for Using Expansion Data Registers:

- All expansion data registers are "keep" types and cannot be designated as "clear" types using the Function Area Settings.
- When expansion data registers are designated as source or destination operands of advanced instructions, the execution time takes slightly longer compared with ordinary data registers D0 through D1299.
- When a user program RAM sum check error has occurred, the data of preset expansion data registers are loaded to the RAM as is the case when the CPU is powered up.
- When the initializing relay is turned on, the scan time is extended until the data load from the EEPROM is completed by approximately 7 ms for every 1000 words of data read from the EEPROM. The data size can be calculated from the following formula:

  Data size (words) = 8.5 + Quantity of preset data registers

- When the backup relay is turned on, the scan time is extended until the data write to the EEPROM is completed for several scans by approximately 200 ms in every scan.
- Writing to the EEPROM can be repeated a maximum of 100,000 times. Keep writing to the EEPROM to a minimum.

6: ALLOCATION NUMBERS

Introduction

This chapter describes allocation numbers available for the MicroSmart to program basic and advanced instructions. Special internal relays and special data registers are also described.

The MicroSmart is programmed using operands such as inputs, outputs, internal relays, timers, counters, shift registers, and data registers.

Inputs (I) are relays to receive input signals through the input terminals.
Outputs (Q) are relays to send the processed results of the user program to the output terminals.
Internal relays (M) are relays used in the CPU and cannot be outputted to the output terminals.
Special internal relays (M) are internal relays dedicated to specific functions.
Timers (T) are relays used in the user program, available in 1-sec, 100-ms, 10-ms, and 1-ms timers.
Counters (C) are relays used in the user program, available in adding counters and reversible counters.
Shift registers (R) are registers to shift the data bits according to pulse inputs.
Data registers (D) are registers used to store numerical data. Some of the data registers are dedicated to special functions.

Operand Allocation Numbers

Available I/O numbers depend on the type of the MicroSmart CPU module and the combination of I/O modules. I/O modules can be used with only the 24-I/O type CPU module among all-in-one type CPU modules. All slim type CPU modules can be used with I/O modules to expand the I/O points. For details of I/O, internal relay, and special internal relay numbers, see page 6-3.

All-in-One Type CPU Modules

Operand	FC4A-C10R2 FC4A-C10R2C		FC4A-C16R2 FC4A-C16R2C		FC4A-C24R2 FC4A-C24R2C	
	Allocation No.	Points	Allocation No.	Points	Allocation No.	Points
Input (I)	I0 - I5	6	I0 - I7 I10	9	I0 - I7 I10 - I15	14
Expansion Input (I)	—	—	—	—	I30 - I107	64 (78 total)
Output (Q)	Q0 - Q3	4	Q0 - Q6	7	Q0 - Q7 Q10 - Q11	10
Expansion Output (Q)	—	—	—	—	Q30 - Q107	64 (74 total)
Internal Relay (M)	M0 - M317	256	M0 - M1277	1024	M0 - M1277	1024
Special Internal Relay (M)	M8000 - M8157	128	M8000 - M8157	128	M8000 - M8157	128
Shift Register (R)	R0 - R63	64	R0 - R127	128	R0 - R127	128
Timer (T)	T0 - T31	32	T0 - T99	100	T0 - T99	100
Counter (C)	C0 - C31	32	C0 - C99	100	C0 - C99	100
Data Register (D)	D0 - D399	400	D0 - D1299	1300	D0 - D1299	1300
Special Data Register (D)	D8000 - D8099	100	D8000 - D8199	200	D8000 - D8199	200

Notes:

- The least significant digit of input, output, internal relay, and special internal relay operand number is an octal number (0 through 7). Upper digits are decimal numbers.
- The allocation numbers of expansion inputs and outputs start with I30 and Q30, respectively.
- Note that input and output allocation numbers are not continuous between the CPU module and expansion I/O modules.
- The 24-I/O type CPU module (FC4A-C24R2 and FC4A-C24R2C) can add a maximum of 64 I/O points, and use a maximum of 88 points of inputs and outputs in total.

Slim Type CPU Modules

Operand	FC4A-D20K3 FC4A-D20S3		FC4A-D20RK1 FC4A-D20RS1		FC4A-D40K3 FC4A-D40S3	
	Allocation No.	Points	Allocation No.	Points	Allocation No.	Points
Input (I)	I0 - I7 I10 - I13	12	I0 - I7 I10 - I13	12	I0 - I7 I10 - I17 I20 - I27	24
Expansion Input (I)	I30 - I187	128 (140 total)	I30 - I307	224 (236 total)	I30 - I307	224 (248 total)
Output (Q)	Q0 - Q7	8	Q0 - Q7	8	Q0 - Q7 Q10 - Q17	16
Expansion Output (Q)	Q30 - Q187	128 (136 total)	Q30 - Q307	224 (232 total)	Q30 - Q307	224 (240 total)
Internal Relay (M)	M0 - M1277	1024	M0 - M1277	1024	M0 - M1277	1024
AS-Interface Internal Relay (M)	—	—	M1300 - M1997	560	M1300 - M1997	560
Special Internal Relay (M)	M8000 - M8157	128	M8000 - M8157	128	M8000 - M8157	128
Shift Register (R)	R0 - R127	128	R0 - R127	128	R0 - R127	128
Timer (T)	T0 - T99	100	T0 - T99	100	T0 - T99	100
Counter (C)	C0 - C99	100	C0 - C99	100	C0 - C99	100
Data Register (D)	D0 - D1299	1300	D0 - D1299	1300	D0 - D1299	1300
AS-Interface Data Register (D)	—	—	D1700 - D1999	300	D1700 - D1999	300
Expansion Data Register (D)	—	—	D2000 - D7999	6000	D2000 - D7999	6000
Special Data Register (D)	D8000 - D8199	200	D8000 - D8199	200	D8000 - D8199	200

Notes:

- The least significant digit of input, output, internal relay, and special internal relay operand number is an octal number (0 through 7). Upper digits are decimal numbers.
- The allocation numbers of expansion inputs and outputs start with I30 and Q30, respectively.
- Note that input and output allocation numbers are not continuous between the CPU module and expansion I/O modules.
- A maximum of 7 expansion I/O modules can be mounted on all slim type CPU modules. The maximum I/O points depend on the CPU module type as described below.
- The 20-I/O type CPU module (FC4A-D20K3 and FC4A-D20S3) can add a maximum of 128 I/O points, and use a maximum of 148 points of inputs and outputs in total.
- The 20-I/O relay output type CPU module (FC4A-D20RK1 and FC4A-D20RS1) can add a maximum of 224 I/O points, and use a maximum of 244 points of inputs and outputs in total.
- The 40-I/O type CPU module (FC4A-D40K3 and FC4A-D40S3) can add a maximum of 224 I/O points, and use a maximum of 264 points of inputs and outputs in total.
- Four models of the slim type CPU modules (FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3) with system program ver. 201 and higher can use the AS-Interface master module, and have additional internal relays and data registers for AS-Interface communication. Use WindLDR ver. 4.20 or higher to program the AS-Interface operands. For details about AS-Interface communication, see a separate user's manual for the AS-Interface master module.
- When the AS-Interface master module is not connected, these AS-Interface operands can be used for basic and advanced instructions like ordinary internal relays and data registers. Note that these operands cannot be designated for keep or clear operands in the Function Area Settings dialog box in WindLDR. In addition, the clear operand data command of the maintenance communication protocol and the designated reset input do not work on these AS-Interface operands. The statuses of these AS-Interface operands are maintained at power-up or when a reset input is turned on, but are cleared when a keep data error occurs.

I/O, Internal Relay, and Special Internal Relay Operand Allocation Numbers

Operand	Allocation Numbers				CPU Module
Input (I)	I0-I5				FC4A-C10R2/C
	I0-I7	I10			FC4A-C16R2/C
	I0-I7 I30-I37 I70-I77	I10-I15 I40-I47 I80-I87	I50-I57 I90-I97	I60-I67 I100-I107	FC4A-C24R2/C
	I0-I7 I30-I37 I70-I77 I110-I117 I150-I157	I10-I13 I40-I47 I80-I87 I120-I127 I160-I167	I50-I57 I90-I97 I130-I137 I170-I177	I60-I67 I100-I107 I140-I147 I180-I187	FC4A-D20K3 FC4A-D20S3
	I0-I7 I30-I37 I70-I77 I110-I117 I150-I157 I190-I197 I230-I237 I270-I277	I10-I13 I40-I47 I80-I87 I120-I127 I160-I167 I200-I207 I240-I247 I280-I287	I50-I57 I90-I97 I130-I137 I170-I177 I210-I217 I250-I257 I290-I297	I60-I67 I100-I107 I140-I147 I180-I187 I220-I227 I260-I267 I300-I307	FC4A-D20RK1 FC4A-D20RS1
	I0-I7 I30-I37 I70-I77 I110-I117 I150-I157 I190-I197 I230-I237 I270-I277	I10-I17 I40-I47 I80-I87 I120-I127 I160-I167 I200-I207 I240-I247 I280-I287	I20-I27 I50-I57 I90-I97 I130-I137 I170-I177 I210-I217 I250-I257 I290-I297	I60-I67 I100-I107 I140-I147 I180-I187 I220-I227 I260-I267 I300-I307	FC4A-D40K3 FC4A-D40S3
Output (Q)	Q0-Q3				FC4A-C10R2/C
	Q0-Q6				FC4A-C16R2/C
	Q0-Q7 Q30-Q37 Q70-Q77	Q10-Q11 Q40-Q47 Q80-Q87	Q50-Q57 Q90-Q97	Q60-Q67 Q100-Q107	FC4A-C24R2/C
	Q0-Q7 Q30-Q37 Q70-Q77 Q110-Q117 Q150-Q157	Q40-Q47 Q80-Q87 Q120-Q127 Q160-Q167	Q50-Q57 Q90-Q97 Q130-Q137 Q170-Q177	Q60-Q67 Q100-Q107 Q140-Q147 Q180-Q187	FC4A-D20K3 FC4A-D20S3
	Q0-Q7 Q30-Q37 Q70-Q77 Q110-Q117 Q150-Q157 Q190-Q197 Q230-Q237 Q270-Q277	Q40-Q47 Q80-Q87 Q120-Q127 Q160-Q167 Q200-Q207 Q240-Q247 Q280-Q287	Q50-Q57 Q90-Q97 Q130-Q137 Q170-Q177 Q210-Q217 Q250-Q257 Q290-Q297	Q60-Q67 Q100-Q107 Q140-Q147 Q180-Q187 Q220-Q227 Q260-Q267 Q300-Q307	FC4A-D20RK1 FC4A-D20RS1
	Q0-Q7 Q30-Q37 Q70-Q77 Q110-Q117 Q150-Q157 Q190-Q197 Q230-Q237 Q270-Q277	Q10-Q17 Q40-Q47 Q80-Q87 Q120-Q127 Q160-Q167 Q200-Q207 Q240-Q247 Q280-Q287	Q50-Q57 Q90-Q97 Q130-Q137 Q170-Q177 Q210-Q217 Q250-Q257 Q290-Q297	Q60-Q67 Q100-Q107 Q140-Q147 Q180-Q187 Q220-Q227 Q260-Q267 Q300-Q307	FC4A-D40K3 FC4A-D40S3

Operand	Allocation Numbers				CPU Module
Internal Relay (M)	M0-M7	M10-M17	M20-M27	M30-M37	All types
	M40-M47	M50-M57	M60-M67	M70-M77	
	M80-M87	M90-M97	M100-M107	M110-M117	
	M120-M127	M130-M137	M140-M147	M150-M157	
	M160-M167	M170-M177	M180-M187	M190-M197	
	M200-M207	M210-M217	M220-M227	M230-M237	
	M240-M247	M250-M257	M260-M267	M270-M277	
	M280-M287	M290-M297	M300-M307	M310-M317	
	M320-M327	M330-M337	M340-M347	M350-M357	All types except FC4A-C10R2/C
	M360-M367	M370-M377	M380-M387	M390-M397	
	M400-M407	M410-M417	M420-M427	M430-M437	
	M440-M447	M450-M457	M460-M467	M470-M477	
	M480-M487	M490-M497	M500-M507	M510-M517	
	M520-M527	M530-M537	M540-M547	M550-M557	
	M560-M567	M570-M577	M580-M587	M590-M597	
	M600-M607	M610-M617	M620-M627	M630-M637	
	M640-M647	M650-M657	M660-M667	M670-M677	
	M680-M687	M690-M697	M700-M707	M710-M717	
	M720-M727	M730-M737	M740-M747	M750-M757	
	M760-M767	M770-M777	M780-M787	M790-M797	
	M800-M807	M810-M817	M820-M827	M830-M837	
	M840-M847	M850-M857	M860-M867	M870-M877	
	M880-M887	M890-M897	M900-M907	M910-M917	
	M920-M927	M930-M937	M940-M947	M950-M957	
	M960-M967	M970-M977	M980-M987	M990-M997	
	M1000-M1007	M1010-M1017	M1020-M1027	M1030-M1037	
	M1040-M1047	M1050-M1057	M1060-M1067	M1070-M1077	
	M1080-M1087	M1090-M1097	M1100-M1107	M1110-M1117	
	M1120-M1127	M1130-M1137	M1140-M1147	M1150-M1157	
	M1160-M1167	M1170-M1177	M1180-M1187	M1190-M1197	
	M1200-M1207	M1210-M1217	M1220-M1227	M1230-M1237	
	M1240-M1247	M1250-M1257	M1260-M1267	M1270-M1277	
Special Internal Relay (M) M8080-M8157 for read only	M8000-M8007	M8010-M8017	M8020-M8027	M8030-M8037	All types
	M8040-M8047	M8050-M8057	M8060-M8067	M8070-M8077	
	M8080-M8087	M8090-M8097	M8100-M8107	M8110-M8117	
	M8120-M8127	M8130-M8137	M8140-M8147	M8150-M8157	

Operand Allocation Numbers for END Refresh Type Analog I/O Modules

Analog I/O Module Number	Analog Input Channel 0	Analog Input Channel 1	Analog Output	Reserved
1	D760-D765	D766-D771	D772-D777	D778, D779
2	D780-D785	D786-D791	D792-D797	D798, D799
3	D800-D805	D806-D811	D812-D817	D818, D819
4	D820-D825	D826-D831	D832-D837	D838, D839
5	D840-D845	D846-D851	D852-D857	D858, D859
6	D860-D865	D866-D871	D872-D877	D878, D879
7	D880-D885	D886-D891	D892-D897	D898, D899

Note: Each analog I/O module uses 20 data registers. When analog modules are not connected, the corresponding data registers can be used as ordinary data registers.

Operand Allocation Numbers for AS-Interface Master Module

MicroSmart CPU Module		AS-Interface Master Module EEPROM
Operand	**Allocation No.**	**AS-Interface Object**
AS-Interface Internal Relays	M1300-M1617	Digital input (IDI)
	M1620-M1937	Digital output (ODI)
	M1940-M1997	Status information
AS-Interface Data Registers	D1700-D1731	Analog input
	D1732-D1763	Analog output
	D1764-D1767	List of active slaves (LAS)
	D1768-D1771	List of detected slaves (LDS)
	D1772-D1775	List of peripheral fault slaves (LPF)
	D1776-D1779	List of projected slaves (LPS)
	D1780-D1811	Configuration data image A (CDI)
	D1812-D1843	Configuration data image B (CDI)
	D1844-D1875	Permanent configuration data A (PCD)
	D1876-D1907	Permanent configuration data B (PCD)
	D1908-D1923	Parameter image (PI)
	D1924-D1939	Permanent parameter (PP)
	D1940	Slave 0 ID1 code
	D1941-D1945	For ASI command description
	D1946-D1999	— Reserved —

Note: AS-Interface master module uses internal relays and data registers shown above. When AS-Interface master module is not connected, these internal relays and data registers can be used as ordinary internal relays and data registers.

Operand Allocation Numbers for Data Link Master Station

Slave Station Number	Allocation Number		
	Transmit Data to Slave Station	Receive Data from Slave Station	Data Link Communication Error
Slave Station 1	D900-D905	D906-D911	D8069
Slave Station 2	D912-D917	D918-D923	D8070
Slave Station 3	D924-D929	D930-D935	D8071
Slave Station 4	D936-D941	D942-D947	D8072
Slave Station 5	D948-D953	D954-D959	D8073
Slave Station 6	D960-D965	D966-D971	D8074
Slave Station 7	D972-D977	D978-D983	D8075
Slave Station 8	D984-D989	D990-D995	D8076
Slave Station 9	D996-D1001	D1002-D1007	D8077
Slave Station 10	D1008-D1013	D1014-D1019	D8078
Slave Station 11	D1020-D1025	D1026-D1031	D8079
Slave Station 12	D1032-D1037	D1038-D1043	D8080
Slave Station 13	D1044-D1049	D1050-D1055	D8081
Slave Station 14	D1056-D1061	D1062-D1067	D8082
Slave Station 15	D1068-D1073	D1074-D1079	D8083
Slave Station 16	D1080-D1085	D1086-D1091	D8084
Slave Station 17	D1092-D1097	D1098-D1103	D8085
Slave Station 18	D1104-D1109	D1110-D1115	D8086
Slave Station 19	D1116-D1121	D1122-D1127	D8087
Slave Station 20	D1128-D1133	D1134-D1139	D8088
Slave Station 21	D1140-D1145	D1146-D1151	D8089
Slave Station 22	D1152-D1157	D1158-D1163	D8090
Slave Station 23	D1164-D1169	D1170-D1175	D8091
Slave Station 24	D1176-D1181	D1182-D1187	D8092
Slave Station 25	D1188-D1193	D1194-D1199	D8093
Slave Station 26	D1200-D1205	D1206-D1211	D8094
Slave Station 27	D1212-D1217	D1218-D1223	D8095
Slave Station 28	D1224-D1229	D1230-D1235	D8096
Slave Station 29	D1236-D1241	D1242-D1247	D8097
Slave Station 30	D1248-D1253	D1254-D1259	D8098
Slave Station 31	D1260-D1265	D1266-D1271	D8099

Note: When any slave stations are not connected, master station data registers which are assigned to the vacant slave stations can be used as ordinary data registers.

Operand Allocation Numbers for Data Link Slave Station

Data	Allocation Number		
	Transmit Data to Master Station	Receive Data from Master Station	Data Link Communication Error
Slave Station Data	D900-D905	D906-D911	D8069

Note: Slave station data registers D912 through D1271 and D8070 through D8099 can be used as ordinary data registers.

Special Internal Relays

Special internal relays M8000 through M8077 are read/write internal relays used for controlling the CPU operation and communication. Special internal relays M8080 through M8157 are read-only internal relays primarily used for indicating the CPU statuses. All special internal relays cannot be used as destinations of advanced instructions.

Internal relays M300 through M315 are used to read input operand statuses of the IOREF (I/O refresh) instruction.

⚠ Caution • Do not change the status of reserved special internal relays, otherwise the MicroSmart may not operate correctly.

Special Internal Relay Allocation Numbers (Read/Write)

Allocation Number	Description	CPU Stopped	Power OFF
M8000	Start Control	Maintained	Maintained
M8001	1-sec Clock Reset	Cleared	Cleared
M8002	All Outputs OFF	Cleared	Cleared
M8003	Carry (Cy) or Borrow (Bw)	Cleared	Cleared
M8004	User Program Execution Error	Cleared	Cleared
M8005	Data Link Communication Error	Maintained	Cleared
M8006	Data Link Communication Prohibit Flag (Master Station)	Maintained	Maintained
M8007	Data Link Communication Initialize Flag (Master Station) Data Link Communication Stop Flag (Slave Station)	Cleared	Cleared
M8010	Status LED	Operating	Cleared
M8011	HMI Write Prohibit Flag	Maintained	Cleared
M8012	HMI Operation Prohibit Flag	Maintained	Cleared
M8013	Calendar/Clock Data Write/Adjust Error Flag	Operating	Cleared
M8014	Calendar/Clock Data Read Error Flag	Operating	Cleared
M8015	Calendar/Clock Data Read Prohibit Flag	Maintained	Cleared
M8016	Calendar Data Write Flag	Operating	Cleared
M8017	Clock Data Write Flag	Operating	Cleared
M8020	Calendar/Clock Data Write Flag	Operating	Cleared
M8021	Clock Data Adjust Flag	Operating	Cleared
M8022	User Communication Receive Instruction Cancel Flag (Port 1)	Cleared	Cleared
M8023	User Communication Receive Instruction Cancel Flag (Port 2)	Cleared	Cleared
M8024	BMOV/WSFT Executing Flag	Maintained	Maintained
M8025	Maintain Outputs While CPU Stopped	Maintained	Cleared
M8026	Expansion Data Register Data Writing Flag (Preset Range 1)	Operating	Maintained
M8027	Expansion Data Register Data Writing Flag (Preset Range 2)	Operating	Maintained
M8030	High-speed Counter 1 (I0-I2) Comparison Output Reset	Cleared	Cleared
M8031	High-speed Counter 1 (I0-I2) Gate Input	Maintained	Cleared
M8032	High-speed Counter 1 (I0-I2) Reset Input	Maintained	Cleared
M8033	— Reserved —	—	—
M8034	High-speed Counter 2 (I3) Comparison Output Reset	Cleared	Cleared
M8035	High-speed Counter 2 (I3) Gate Input	Maintained	Cleared
M8036	High-speed Counter 2 (I3) Reset Input	Maintained	Cleared
M8037	— Reserved —	—	—
M8040	High-speed Counter 3 (I4) Comparison Output Reset	Cleared	Cleared
M8041	High-speed Counter 3 (I4) Gate Input	Maintained	Cleared
M8042	High-speed Counter 3 (I4) Reset Input	Maintained	Cleared
M8043	— Reserved —	—	—
M8044	High-speed Counter 4 (I5-I7) Comparison Output Reset	Cleared	Cleared

Allocation Number	Description	CPU Stopped	Power OFF
M8045	High-speed Counter 4 (I5-I7) Gate Input	Maintained	Cleared
M8046	High-speed Counter 4 (I5-I7) Reset Input	Maintained	Cleared
M8047	— Reserved —	—	—
M8050	Modem Mode (Originate): Initialization String Start	Maintained	Maintained
M8051	Modem Mode (Originate): ATZ Start	Maintained	Maintained
M8052	Modem Mode (Originate): Dialing Start	Maintained	Maintained
M8053	Modem Mode (Disconnect): Disconnect Line Start	Maintained	Maintained
M8054	Modem Mode (General Command): AT Command Start	Maintained	Maintained
M8055	Modem Mode (Answer): Initialization String Start	Maintained	Maintained
M8056	Modem Mode (Answer): ATZ Start	Maintained	Maintained
M8057	Modem Mode AT Command Execution	Maintained	Cleared
M8060	Modem Mode (Originate): Initialization String Completion	Maintained	Cleared
M8061	Modem Mode (Originate): ATZ Completion	Maintained	Cleared
M8062	Modem Mode (Originate): Dialing Completion	Maintained	Cleared
M8063	Modem Mode (Disconnect): Disconnect Line Completion	Maintained	Cleared
M8064	Modem Mode (General Command): AT Command Completion	Maintained	Cleared
M8065	Modem Mode (Answer): Initialization String Completion	Maintained	Cleared
M8066	Modem Mode (Answer): ATZ Completion	Maintained	Cleared
M8067	Modem Mode Operational State	Maintained	Cleared
M8070	Modem Mode (Originate): Initialization String Failure	Maintained	Cleared
M8071	Modem Mode (Originate): ATZ Failure	Maintained	Cleared
M8072	Modem Mode (Originate): Dialing Failure	Maintained	Cleared
M8073	Modem Mode (Disconnect): Disconnect Line Failure	Maintained	Cleared
M8074	Modem Mode (General Command): AT Command Failure	Maintained	Cleared
M8075	Modem Mode (Answer): Initialization String Failure	Maintained	Cleared
M8076	Modem Mode (Answer): ATZ Failure	Maintained	Cleared
M8077	Modem Mode Line Connection Status	Maintained	Cleared

Special Internal Relay Allocation Numbers (Read Only)

Allocation Number	Description	CPU Stopped	Power OFF
M8080	Data Link Slave Station 1 Communication Completion Relay (Master Station) Data Link Communication Completion Relay (Slave Station)	Operating	Cleared
M8081	Data Link Slave Station 2 Communication Completion Relay	Operating	Cleared
M8082	Data Link Slave Station 3 Communication Completion Relay	Operating	Cleared
M8083	Data Link Slave Station 4 Communication Completion Relay	Operating	Cleared
M8084	Data Link Slave Station 5 Communication Completion Relay	Operating	Cleared
M8085	Data Link Slave Station 6 Communication Completion Relay	Operating	Cleared
M8086	Data Link Slave Station 7 Communication Completion Relay	Operating	Cleared
M8087	Data Link Slave Station 8 Communication Completion Relay	Operating	Cleared
M8090	Data Link Slave Station 9 Communication Completion Relay	Operating	Cleared
M8091	Data Link Slave Station 10 Communication Completion Relay	Operating	Cleared
M8092	Data Link Slave Station 11 Communication Completion Relay	Operating	Cleared
M8093	Data Link Slave Station 12 Communication Completion Relay	Operating	Cleared
M8094	Data Link Slave Station 13 Communication Completion Relay	Operating	Cleared
M8095	Data Link Slave Station 14 Communication Completion Relay	Operating	Cleared
M8096	Data Link Slave Station 15 Communication Completion Relay	Operating	Cleared
M8097	Data Link Slave Station 16 Communication Completion Relay	Operating	Cleared

Allocation Number	Description	CPU Stopped	Power OFF
M8100	Data Link Slave Station 17 Communication Completion Relay	Operating	Cleared
M8101	Data Link Slave Station 18 Communication Completion Relay	Operating	Cleared
M8102	Data Link Slave Station 19 Communication Completion Relay	Operating	Cleared
M8103	Data Link Slave Station 20 Communication Completion Relay	Operating	Cleared
M8104	Data Link Slave Station 21 Communication Completion Relay	Operating	Cleared
M8105	Data Link Slave Station 22 Communication Completion Relay	Operating	Cleared
M8106	Data Link Slave Station 23 Communication Completion Relay	Operating	Cleared
M8107	Data Link Slave Station 24 Communication Completion Relay	Operating	Cleared
M8110	Data Link Slave Station 25 Communication Completion Relay	Operating	Cleared
M8111	Data Link Slave Station 26 Communication Completion Relay	Operating	Cleared
M8112	Data Link Slave Station 27 Communication Completion Relay	Operating	Cleared
M8113	Data Link Slave Station 28 Communication Completion Relay	Operating	Cleared
M8114	Data Link Slave Station 29 Communication Completion Relay	Operating	Cleared
M8115	Data Link Slave Station 30 Communication Completion Relay	Operating	Cleared
M8116	Data Link Slave Station 31 Communication Completion Relay	Operating	Cleared
M8117	Data Link All Slave Station Communication Completion Relay	Operating	Cleared
M8120	Initialize Pulse	Cleared	Cleared
M8121	1-sec Clock	Operating	Cleared
M8122	100-ms Clock	Operating	Cleared
M8123	10-ms Clock	Operating	Cleared
M8124	Timer/Counter Preset Value Changed	Maintained	Maintained
M8125	In-operation Output	Cleared	Cleared
M8126-M8127	— Reserved —	—	—
M8130	High-speed Counter 1 (I0-I2) Reset Status	Maintained	Cleared
M8131	High-speed Counter 1 (I0-I2) Current Value Overflow (two-phase) High-speed Counter 1 (I0-I2) Comparison ON Status (single-phase)	Maintained	Cleared
M8132	High-speed Counter 1 (I0-I2) Current Value Underflow	Maintained	Cleared
M8133	High-speed Counter 2 (I3) Comparison ON Status	Maintained	Cleared
M8134	High-speed Counter 3 (I4) Comparison ON Status	Maintained	Cleared
M8135	High-speed Counter 4 (I5-I7) Reset Status	Maintained	Cleared
M8136	High-speed Counter 4 (I5-I7) Current Value Overflow (two-phase) High-speed Counter 4 (I5-I7) Comparison ON Status (single-phase)	Maintained	Cleared
M8137	High-speed Counter 4 (I5-I7) Current Value Underflow	Maintained	Cleared
M8140	Interrupt Input I2 Status	Cleared	Cleared
M8141	Interrupt Input I3 Status	Cleared	Cleared
M8142	Interrupt Input I4 Status	Cleared	Cleared
M8143	Interrupt Input I5 Status	Cleared	Cleared
M8144	Timer Interrupt Status	Cleared	Cleared
M8145-M8147	— Reserved —	—	—
M8150	Comparison Result Greater Than	Maintained	Cleared
M8151	Comparison Result Less Than	Maintained	Cleared
M8152	Comparison Result Equal To	Maintained	Cleared
M8153	— Reserved —	—	—
M8154	Catch Input I2 ON/OFF Status	Maintained	Cleared
M8155	Catch Input I3 ON/OFF Status	Maintained	Cleared
M8156	Catch Input I4 ON/OFF Status	Maintained	Cleared
M8157	Catch Input I5 ON/OFF Status	Maintained	Cleared

M8000 Start Control

M8000 is used to control the operation of the CPU. The CPU stops operation when M8000 is turned off while the CPU is running. M8000 can be turned on or off using the WindLDR Online menu. When a stop or reset input is designated, M8000 must remain on to control the CPU operation using the stop or reset input. For the start and stop operation, see page 4-3.

M8000 maintains its status when the CPU is powered down. When the data to be maintained during power failure is broken after the CPU has been off for a period longer than the battery backup duration, the CPU restarts operation or not as selected in **Function Area Settings > Run/Stop > Run/Stop Selection at Memory Backup Error**. See page 5-3.

M8001 1-sec Clock Reset

While M8001 is on, M8121 (1-sec clock) is turned off.

M8002 All Outputs OFF

When M8002 is turned on, all outputs (Q0 through Q307) go off until M8002 is turned off. Self-maintained circuits using outputs also go off and are not restored when M8002 is turned off.

M8003 Carry (Cy) and Borrow (Bw)

When a carry or borrow results from executing an addition or subtraction instruction, M8003 turns on. M8003 is also used for the bit shift and rotate instructions. See pages 11-2 and 13-1.

M8004 User Program Execution Error

When an error occurs while executing a user program, M8004 turns on. The cause of the user program execution error can be checked using **Online > Monitor > PLC Status > Error Status > Details**. See page 29-6.

M8005 Data Link Communication Error

When an error occurs during communication in the data link system, M8005 turns on. The M8005 status is maintained when the error is cleared and remains on until M8005 is reset using WindLDR or until the CPU is turned off. The cause of the data link communication error can be checked using **Online > Monitor > PLC Status > Error Status > Details**. See page 25-4.

M8006 Data Link Communication Prohibit Flag (Master Station)

When M8006 at the master station is turned on in the data link system, data link communication is stopped. The M8006 status is maintained when the CPU is turned off and remains on until M8006 is reset using WindLDR.

M8007 Data Link Communication Initialize Flag (Master Station)
Data Link Communication Stop Flag (Slave Station)

M8007 has a different function at the master or slave station of the data link communication system.

Master station: Data link communication initialize flag

When M8007 at the master station is turned on during operation, the link configuration is checked to initialize the data link system. When a slave station is powered up after the master station, turn M8007 on to initialize the data link system. After a data link setup is changed, M8007 must also be turned on to ensure correct communication.

Slave station: Data link communication stop flag

When a slave station does not receive communication data from the master station for 10 sec or more in the data link system, M8007 turns on. When the slave station receives correct communication data, M8007 turns off.

M8010 Status LED

When M8010 is turned on or off, the STAT LED on the CPU module turns on or off, respectively.

M8011 HMI Write Prohibit Flag

When M8011 is turned on, the HMI module is disabled from writing data to prevent unauthorized modifications, such as direct set/reset, changing timer/counter preset values, and entering data into data registers.

M8012 HMI Operation Prohibit Flag

When M8012 is turned on, the HMI module is disabled from all operations, reducing the scan time. To turn off M8012, power down and up the CPU, or use the Point Write on WindLDR.

M8013 Calendar/Clock Data Write/Adjust Error Flag

When an error occurs while calendar/clock data is written or clock data is adjusted, M8013 turns on. If calendar/clock data is written or clock data is adjusted successfully, M8013 turns off.

M8014 Calendar/Clock Data Read Error Flag

When an error occurs while calendar/clock data is read, M8014 turns on. If calendar/clock data is read successfully, M8014 turns off.

M8015 Calendar/Clock Data Read Prohibit Flag

When a clock cartridge is installed, the calendar/clock data is continuously read to the special data registers D8008 through D8014 for calendar/clock current data whether the CPU is running or stopped. When M8015 is turned on while the CPU is running, calendar/clock data reading is prohibited to reduce the scan time.

M8016 Calendar Data Write Flag

When M8016 is turned on, data in data registers D8015 through D8018 (calendar new data) are set to the clock cartridge installed on the CPU module. See page 15-6.

M8017 Clock Data Write Flag

When M8017 is turned on, data in data registers D8019 through D8021 (clock new data) are set to the clock cartridge installed on the CPU module. See page 15-6.

M8020 Calendar/Clock Data Write Flag

When M8020 is turned on, data in data registers D8015 through D8021 (calendar/clock new data) are set to the clock cartridge installed on the CPU module. See page 15-6.

M8021 Clock Data Adjust Flag

When M8021 is turned on, the clock is adjusted with respect to seconds. If *seconds* are between 0 and 29 for current time, adjustment for *seconds* will be set to 0 and minutes remain the same. If *seconds* are between 30 and 59 for current time, adjustment for *seconds* will be set to 0 and *minutes* are incremented one. See page 15-6.

M8022 User Communication Receive Instruction Cancel Flag (Port 1)

When M8022 is turned on, all RXD1 instructions ready for receiving user communication through port 1 are disabled.

M8023 User Communication Receive Instruction Cancel Flag (Port 2)

When M8023 is turned on, all RXD2 instructions ready for receiving user communication through port 2 are disabled.

M8024 BMOV/WSFT Executing Flag

While the BMOV or WSFT is executed, M8024 turns on. When completed, M8024 turns off. If the CPU is powered down while executing BMOV or WSFT, M8024 remains on when the CPU is powered up again.

M8025 Maintain Outputs While CPU Stopped

Outputs are normally turned off when the CPU is stopped. M8025 is used to maintain the output statuses when the CPU is stopped. When the CPU is stopped with M8025 turned on, the output ON/OFF statuses are maintained. When the CPU restarts, M8025 is turned off automatically.

M8026 Expansion Data Register Data Writing Flag (Preset Range 1)
M8027 Expansion Data Register Data Writing Flag (Preset Range 2)

While data write from the CPU RAM to expansion data register preset range 1 or 2 in the EEPROM is in progress, M8026 or M8027 turns on, respectively. When data write is complete, the special internal relay turns off.

M8030, M8034, M8040, M8044 High-speed Counter Comparison Output Reset

When M8030, M8034, M8040, or M8044 is turned on, the comparison output of high-speed counter 1, 2, 3, or 4 is turned off, respectively. See page 5-6.

M8031, M8035, M8041, M8045 High-speed Counter Gate Input

While M8031, M8035, M8041, or M8045 is on, counting is enabled for high-speed counter 1, 2, 3, or 4, respectively. See page 5-6.

M8032, M8036, M8042, M8046 High-speed Counter Reset Input

When M8032 or M8046 is turned on while two-phase high-speed counter 1 or 4 is enabled, the current value in D8045 or D8051 is reset to the value stored in D8046 or D8052 (high-speed counter reset value) and the two-phase high-speed counter counts subsequent input pulses starting at the reset value.

When M8032, M8036, M8042, or M8046 is turned on while single-phase high-speed counter 1, 2, 3, or 4 is enabled, the current value in D8045, D8047, D8049, or D8051 is reset to 0, respectively.

M8050-M8077 Special Internal Relays for Modem Mode

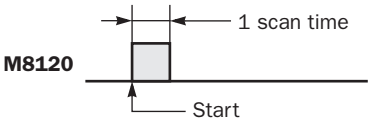
See page 27-2.

M8080-M8117 Special Internal Relays for Data Link Communication

See page 25-6.

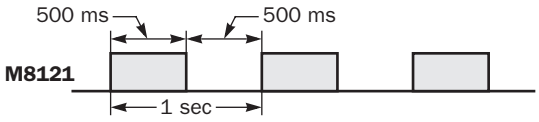
M8120 Initialize Pulse

When the CPU starts operation, M8120 turns on for a period of one scan.



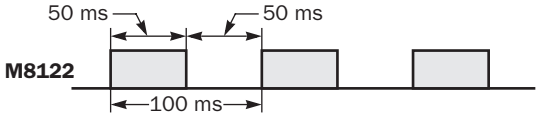
M8121 1-sec Clock

While M8001 (1-sec clock reset) is off, M8121 generates clock pulses in 1-sec increments, with a duty ratio of 1:1 (500 ms on and 500 ms off).



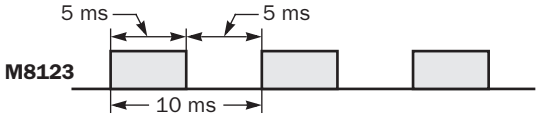
M8122 100-ms Clock

M8122 always generates clock pulses in 100-ms increments, whether M8001 is on or off, with a duty ratio of 1:1 (50 ms on and 50 ms off).



M8123 10-ms Clock

M8123 always generates clock pulses in 10-ms increments, whether M8001 is on or off, with a duty ratio of 1:1 (5 ms on and 5 ms off).



M8124 Timer/Counter Preset Value Changed

When timer or counter preset values are changed in the CPU module RAM, M8124 turns on. When a user program is downloaded to the CPU from WindLDR or when the changed timer/counter preset value is cleared, M8124 turns off.

Timer or counter preset and current values can be changed using WindLDR without transferring the entire program to the CPU again (see pages 7-8 and 7-10). When a timer or counter is designated as a destination of an advanced instruction, the timer/counter preset value is also changed.

M8125 In-operation Output

M8125 remains on while the CPU is running.

M8130 High-speed Counter 1 (I0-I2) Reset Status (ON for 1 scan)

When reset input I2 is turned on while high-speed counter 1 is enabled either in two-phase or single-phase high-speed counter mode, M8130 turns on for one scan to indicate that the high-speed counter 1 current value is reset. When reset input special internal relay M8032 is turned on, M8130 does not turn on.

M8131 High-speed Counter 1 (I0-I2) Current Value Overflow (two-phase high-speed counter) (ON for 1 scan) High-speed Counter 1 (I0-I2) Comparison ON Status (single-phase high-speed counter) (ON for 1 scan)

When the current value of high-speed counter 1 exceeds 65535 while two-phase high-speed counter is enabled, M8131 turns on for one scan.

When the current value of high-speed counter 1 reaches the preset value while single-phase high-speed counter is enabled, M8131 turns on for one scan.

M8132 High-speed Counter 1 (I0-I2) Current Value Underflow (ON for 1 scan)

When the current value of high-speed counter 1 drops blow 0 while two-phase high-speed counter is enabled, M8132 turns on for one scan.

M8133 High-speed Counter 2 (I3) Comparison ON Status (ON for 1 scan)

When the current value of high-speed counter 2 reaches the preset value, M8133 turns on for one scan.

M8134 High-speed Counter 3 (I4) Comparison ON Status (ON for 1 scan)

When the current value of high-speed counter 3 reaches the preset value, M8134 turns on for one scan.

M8135 High-speed Counter 4 (I5-I7) Reset Status (ON for 1 scan)

When reset input I5 is turned on while high-speed counter 4 is enabled either in two-phase or single-phase high-speed counter mode, M8135 turns on for one scan to indicate that the high-speed counter 4 current value is reset. When reset input special internal relay M8046 is turned on, M8135 does not turn on.

M8136 High-speed Counter 4 (I5-I7) Current Value Overflow (two-phase high-speed counter) (ON for 1 scan)
High-speed Counter 4 (I5-I7) Comparison ON Status (single-phase high-speed counter) (ON for 1 scan)

When the current value of high-speed counter 4 exceeds 65535 while two-phase high-speed counter is enabled, M8136 turns on for one scan.

When the current value of high-speed counter 4 reaches the preset value while single-phase high-speed counter is enabled, M8136 turns on for one scan.

M8137 High-speed Counter 4 (I5-I7) Current Value Underflow (ON for 1 scan)

When the current value of high-speed counter 4 drops blow 0 while two-phase high-speed counter is enabled, M8137 turns on for one scan.

M8140, M8141, M8142, M8143 Interrupt Input Status

When interrupt inputs I2 through I5 are enabled, M8140 through M8143 are turned on, respectively. When disabled, these internal relays are turned off.

M8144 Timer Interrupt Status

When timer interrupt is enabled, M8144 is turned on. When disabled, M8144 is turned off.

M8150 Comparison Result Greater Than

When the CMP= instruction is used, M8150 is turned on when the value of operand designated by S1 is greater than that of operand designated by S2 (S1 > S2). See page 10-2.

When the ICMP>= instruction is used, M8150 is turned on when the value of operand designated by S2 is greater than that of operand designated by S1 (S2 < S1). See page 10-4.

M8151 Comparison Result Equal To

When the CMP= instruction is used, M8151 is turned on when the value of operand designated by S1 is equal to that of operand designated by S2 (S1 = S2). See page 10-2.

When the ICMP>= instruction is used, M8151 is turned on when the value of operand designated by S3 is greater than that of operand designated by S2 (S3 > S2). See page 10-4.

M8152 Comparison Result Less Than

When the CMP= instruction is used, M8152 is turned on when the value of operand designated by S1 is less than that of operand designated by S2 (S1 < S2). See page 10-2.

When the ICMP>= instruction is used, M8152 is turned on when the value of operand designated by S2 is less than that of operand designated by S1 and greater than that of operand designated by S3 (S1 > S2 > S3). See page 10-4.

M8154, M8155, M8156, M8157 Catch Input ON/OFF Status

When a rising or falling input edge is detected during a scan, the input statuses of catch inputs I2 through I5 at the moment are set to M8154 through M8157, respectively, without regard to the scan status. Only one edge is detected in one scan. For the catch input function, see page 5-18.

Special Data Registers

Caution • Do not change the data of reserved special data registers, otherwise the MicroSmart may not operate correctly.

Special Data Register Allocation Numbers

Allocation Number	Description	Updated	See Page
D8000	System Setup ID (Quantity of Inputs)	When I/O initialized	6-16
D8001	System Setup ID (Quantity of Outputs)	When I/O initialized	6-16
D8002	CPU Module Type Information	Power-up	6-17
D8003	Memory Cartridge Information	Power-up	6-17
D8004	— Reserved —	—	—
D8005	General Error Code	When error occurred	29-3
D8006	User Program Execution Error Code	When error occurred	29-6
D8007	— Reserved —	—	—
D8008	Year (Current Data) Read only	Every 500 ms	15-5
D8009	Month (Current Data) Read only	Every 500 ms	15-5
D8010	Day (Current Data) Read only	Every 500 ms	15-5
D8011	Day of Week (Current Data) Read only	Every 500 ms	15-5
D8012	Hour (Current Data) Read only	Every 500 ms	15-5
D8013	Minute (Current Data) Read only	Every 500 ms	15-5
D8014	Second (Current Data) Read only	Every 500 ms	15-5
D8015	Year (New Data) Write only		15-5
D8016	Month (New Data) Write only		15-5
D8017	Day (New Data) Write only		15-5
D8018	Day of Week (New Data) Write only		15-5
D8019	Hour (New Data) Write only		15-5
D8020	Minute (New Data) Write only		15-5
D8021	Second (New Data) Write only		15-5
D8022	Constant Scan Time Preset Value (1 to 1000 ms)		5-27
D8023	Scan Time Current Value (ms)	Every scan	5-27
D8024	Scan Time Maximum Value (ms)	At occurrence	5-27
D8025	Scan Time Minimum Value (ms)	At occurrence	5-27
D8026	Communication Mode Information	Every scan	6-17
D8027	Port 1 Communication Device Number (0 through 31)	Every scan	26-2
D8028	Port 2 Communication Device Number (0 through 31)	Every scan	26-2
D8029	System Program Version	Power-up	6-17
D8030	Communication Adapter Information	Power-up	6-17
D8031	Optional Cartridge Information	Power-up	6-17
D8032	Interrupt Input Jump Destination Label No. (I2)		5-20
D8033	Interrupt Input Jump Destination Label No. (I3)		5-20
D8034	Interrupt Input Jump Destination Label No. (I4)		5-20
D8035	Interrupt Input Jump Destination Label No. (I5)		5-20
D8036	Timer Interrupt Jump Destination Label No.	—	5-22
D8037	Quantity of Expansion I/O Modules	When I/O initialized	6-17
D8038-D8044	— Reserved —	—	—

Special Data Registers for High-speed Counters

Allocation Number	Description	Updated	See Page
D8045	High-speed Counter 1 (I0-I2) Current Value	Every scan	5-7, 5-8
D8046	High-speed Counter 1 (I0-I2) Reset Value (two-phase) High-speed Counter 1 (I0-I2) Preset Value (single-phase)		5-7, 5-8
D8047	High-speed Counter 2 (I3) Current Value	Every scan	5-8
D8048	High-speed Counter 2 (I3) Preset Value		5-8
D8049	High-speed Counter 3 (I4) Current Value	Every scan	5-8
D8050	High-speed Counter 3 (I4) Preset Value		5-8
D8051	High-speed Counter 4 (I5-I7) Current Value	Every scan	5-8
D8052	High-speed Counter 4 (I5-I7) Reset Value (two-phase) High-speed Counter 4 (I5-I7) Preset Value (single-phase)		5-8
D8053-D8054	— Reserved —	—	—

Special Data Register for Pulse Outputs (upgraded CPU modules only)

D8055	Current Pulse Frequency of PULS1 or RAMP (Q0)	Every scan	20-4, 20-17
D8056	Current Pulse Frequency of PULS2 or RAMP (Q1)	Every scan	20-4, 20-17

Special Data Registers for Analog Potentiometers

D8057	Analog Potentiometer 1 Value (All CPU modules)	Every scan	5-30
D8058	Analog Potentiometer 2 Value (All-in-one 24-I/O type CPU) Analog Voltage Input (Slim type CPU modules)	Every scan	5-30, 5-31
D8059-D8067	— Reserved —	—	—

Special Data Register for HMI Module

D8068	HMI Module Initial Screen Selection		5-34

Special Data Registers for Data Link Master/Slave Stations

D8069	Slave Station 1 Slave Station	Communication Error (at Master Station) Communication Error (at Slave Station)	When error occurred	25-4
D8070	Slave Station 2	Communication Error (at Master Station)	When error occurred	25-4
D8071	Slave Station 3	Communication Error (at Master Station)	When error occurred	25-4
D8072	Slave Station 4	Communication Error (at Master Station)	When error occurred	25-4
D8073	Slave Station 5	Communication Error (at Master Station)	When error occurred	25-4
D8074	Slave Station 6	Communication Error (at Master Station)	When error occurred	25-4
D8075	Slave Station 7	Communication Error (at Master Station)	When error occurred	25-4
D8076	Slave Station 8	Communication Error (at Master Station)	When error occurred	25-4
D8077	Slave Station 9	Communication Error (at Master Station)	When error occurred	25-4
D8078	Slave Station 10	Communication Error (at Master Station)	When error occurred	25-4
D8079	Slave Station 11	Communication Error (at Master Station)	When error occurred	25-4
D8080	Slave Station 12	Communication Error (at Master Station)	When error occurred	25-4
D8081	Slave Station 13	Communication Error (at Master Station)	When error occurred	25-4
D8082	Slave Station 14	Communication Error (at Master Station)	When error occurred	25-4
D8083	Slave Station 15	Communication Error (at Master Station)	When error occurred	25-4
D8084	Slave Station 16	Communication Error (at Master Station)	When error occurred	25-4

Allocation Number	Description	Updated	See Page
D8085	Slave Station 17 Communication Error (at Master Station)	When error occurred	25-4
D8086	Slave Station 18 Communication Error (at Master Station)	When error occurred	25-4
D8087	Slave Station 19 Communication Error (at Master Station)	When error occurred	25-4
D8088	Slave Station 20 Communication Error (at Master Station)	When error occurred	25-4
D8089	Slave Station 21 Communication Error (at Master Station)	When error occurred	25-4
D8090	Slave Station 22 Communication Error (at Master Station)	When error occurred	25-4
D8091	Slave Station 23 Communication Error (at Master Station)	When error occurred	25-4
D8092	Slave Station 24 Communication Error (at Master Station)	When error occurred	25-4
D8093	Slave Station 25 Communication Error (at Master Station)	When error occurred	25-4
D8094	Slave Station 26 Communication Error (at Master Station)	When error occurred	25-4
D8095	Slave Station 27 Communication Error (at Master Station)	When error occurred	25-4
D8096	Slave Station 28 Communication Error (at Master Station)	When error occurred	25-4
D8097	Slave Station 29 Communication Error (at Master Station)	When error occurred	25-4
D8098	Slave Station 30 Communication Error (at Master Station)	When error occurred	25-4
D8099	Slave Station 31 Communication Error (at Master Station)	When error occurred	25-4

Special Data Registers for Port 2

D8100-D8102	— Reserved —	—	—
D8103	On-line Mode Protocol Selection	When sending/receiving data	27-3
D8104	RS232C Control Signal Status	Every scan	17-29
D8105	RS232C DSR Input Control Signal Option	When sending/receiving data	17-30
D8106	RS232C DTR Output Control Signal Option	When sending/receiving data	17-30
D8107-D8108	— Reserved —	—	—
D8109	Retry Cycles	At retry	27-3
D8110	Retry Interval	Every scan during retry	27-3
D8111	Modem Mode Status	At status transition	27-3
D8112-D8114	— Reserved —	—	—
D8115-D8129	AT Command Result Code	When returning result code	27-3
D8130-D8144	AT Command String	When sending AT command	27-3
D8145-D8169	Initialization String	When sending init. string	27-3
D8170-D8199	Telephone Number	When dialing	27-3

Note: D8100 through D8199 are not available on the all-in-one 10-I/O type CPU module; all other CPU modules have D8100 through D8199.

D8000 System Setup ID (Quantity of Inputs)

The total of input points provided on the CPU module and connected expansion input modules is stored to D8000. When a mixed I/O module (4 inputs and 4 outputs) is connected, 8 input points are added to the total.

D8001 System Setup ID (Quantity of Outputs)

The total of output points provided on the CPU module and connected expansion output modules is stored to D8001. When a mixed I/O module (4 inputs and 4 outputs) is connected, 8 output points are added to the total.

D8002 CPU Module Type Information

Information about the CPU module type is stored to D8002.

0: FC4A-C10R2 or FC4A-C10R2C
1: FC4A-C16R2 or FC4A-C16R2C
2: FC4A-D20K3 or FC4A-D20S3
3: FC4A-C24R2 or FC4A-C24R2C
4: FC4A-D40K3 or FC4A-D40S3
6: FC4A-D20RK1 or FC4A-D20RS1

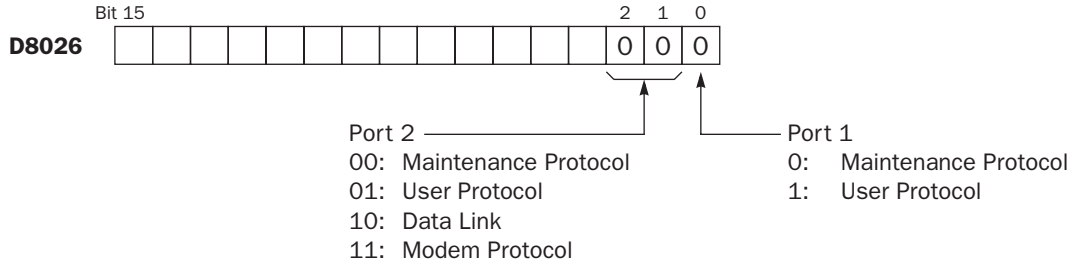
D8003 Memory Cartridge Information

When an optional memory cartridge is installed on the CPU module cartridge connector, information about the user program stored on the memory cartridge is stored to D8003.

0: FC4A-C10R2 or FC4A-C10R2C
1: FC4A-C16R2 or FC4A-C16R2C
2: FC4A-D20K3 or FC4A-D20S3
3: FC4A-C24R2 or FC4A-C24R2C
4: FC4A-D40K3 or FC4A-D40S3
6: FC4A-D20RK1 or FC4A-D20RS1
255: The memory cartridge does not store any user program.

D8026 Communication Mode Information

Communication mode information of port 1 and port 2 is stored to D8026.



D8029 System Program Version

The PLC system program version number is stored to D8029. This value is indicated in the PLC status dialog box called from the WindLDR menu bar. Select **Online > Monitor**, then select **Online > PLC Status**. See page 29-1.

D8030 Communication Adapter Information

Information about the communication adapter installed on the port 2 connector (except for all-in-one 10-I/O type CPU module) is stored to D8030.

0: RS232C communication adapter is installed
1: RS485 communication adapter is installed or no communication adapter is installed

D8031 Optional Cartridge Information

Information about the optional cartridge installed on the CPU module is stored to D8031.

0: No optional cartridge is installed
1: Clock cartridge is installed
2: Memory cartridge is installed
3: Clock cartridge and memory cartridge are installed

D8037 Quantity of Expansion I/O Modules

The quantity of expansion I/O modules connected to the all-in-one 24-I/O type CPU module or any slim type CPU module is stored to D8037.

Expansion I/O Module Operands

Expansion I/O modules are available in digital I/O modules and analog I/O modules.

Among the all-in-one type CPU modules, only the 24-I/O type CPU modules (FC4A-C24R2 and FC4A-C24R2C) can connect a maximum of four expansion I/O modules including analog I/O modules.

All slim type CPU modules can connect a maximum of seven expansion I/O modules including analog I/O modules.

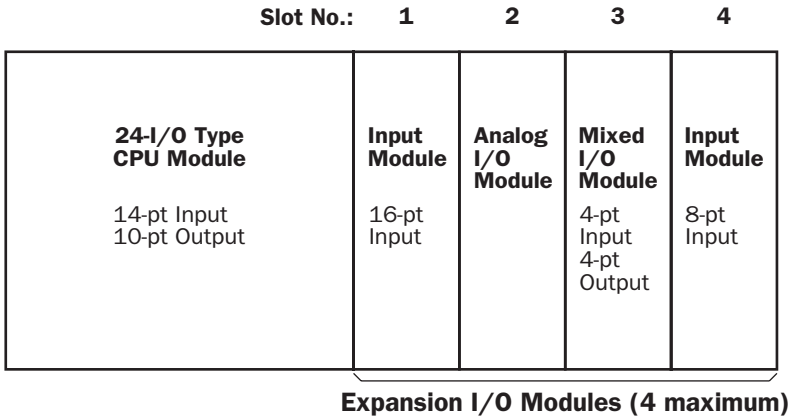
I/O Expansion for All-in-One Type CPU Modules

A maximum of four input, output, mixed I/O, or analog I/O modules can be mounted with the 24-I/O type CPU module, so that the I/O points can be expanded to a maximum of 78 inputs or 74 outputs. The total of inputs and outputs can be a maximum of 88 points. Input and output numbers are automatically allocated to each digital I/O module, starting with I30 and Q30, in the order of increasing distance from the CPU module. Expansion I/O modules cannot be mounted with the 10- and 16-I/O type CPU modules (FC4A-C10R2, FC4A-C10R2C, FC4A-C16R2, and FC4A-C16R2C).

I/O Allocation Numbers (All-in-One Type CPU Modules)

Operand	FC4A-C10R2 FC4A-C10R2C		FC4A-C16R2 FC4A-C16R2C		FC4A-C24R2 FC4A-C24R2C	
	Allocation No.	Points	Allocation No.	Points	Allocation No.	Points
Input (I)	I0 - I5	6	I0 - I7 I10	9	I0 - I7 I10 - I15	14
Expansion Input (I)	—	—	—	—	I30 - I107	64 (78 total)
Output (Q)	Q0 - Q3	4	Q0 - Q6	7	Q0 - Q7 Q10 - Q11	10
Expansion Output (Q)	—	—	—	—	Q30 - Q107	64 (74 total)

Example:



The system setup shown above will have I/O operand numbers allocated for each module as follows:

Slot No.	Module	I/O Operand Numbers
	24-I/O Type CPU Module	I0 to I7, I10 to I15, Q0 to Q7, Q10 and Q11
1	**16-pt Input Module**	I30 to I37, I40 to I47
2	**Analog I/O Module**	See page 24-8.
3	**4/4-pt Mixed I/O Module**	I50 to I53, Q30 to Q33
4	**8-pt Input Module**	I60 to I67

The I/O numbers of the CPU module start with I0 and Q0. The I/O numbers of the expansion I/O modules start with I30 and Q30. The mixed I/O module has 4 inputs and 4 outputs. When an I/O module is mounted next to a mixed I/O module, note that the allocation numbers skip four points as shown above.

Input and output modules may be grouped together for easy identification of I/O numbers. When the I/O modules are relocated, the I/O numbers are renumbered automatically.

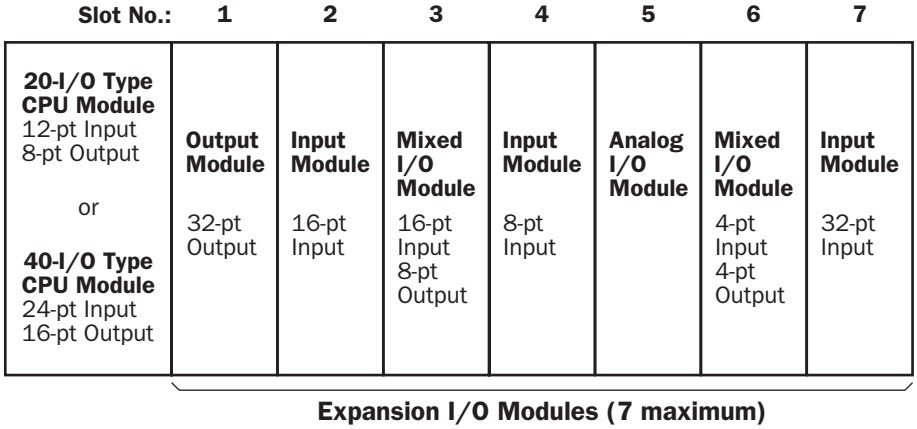
I/O Expansion for Slim Type CPU Modules

All slim type CPU modules can connect a maximum of seven expansion I/O modules including analog I/O modules. The expandable I/O points and the maximum total I/O points vary with the type of CPU module as listed below.

Allocation Numbers (Slim Type CPU Modules)

Operand	FC4A-D20K3 FC4A-D20S3		FC4A-D20RK1 FC4A-D20RS1		FC4A-D40K3 FC4A-D40S3	
	Allocation No.	Points	Allocation No.	Points	Allocation No.	Points
Input (I)	I0 - I7 I10 - I13	12	I0 - I7 I10 - I13	12	I0 - I7 I10 - I17 I20 - I27	24
Expansion Input (I)	I30 - I187	128 (140 total)	I30 - I307	224 (236 total)	I30 - I307	224 (248 total)
Output (Q)	Q0 - Q7	8	Q0 - Q7	8	Q0 - Q7 Q10 - Q17	16
Expansion Output (Q)	Q30 - Q187	128 (136 total)	Q30 - Q307	224 (232 total)	Q30 - Q307	224 (240 total)
Maximum Total I/O Points	148		244		264	

Example:



The system setup shown above will have I/O operand numbers allocated for each module as follows:

Slot No.	Module	I/O Operand Numbers
	40-I/O Type CPU Module	I0 to I7, I10 to I17, I20 to I27, Q0 to Q7, Q10 to Q27
1	**32-pt Output Module**	Q30 to Q37, Q40 to Q47, Q50 to Q57, Q60 to Q67
2	**16-pt Input Module**	I30 to I37, I40 to I47
3	**16/8-pt Mixed I/O Module**	I50 to I57, I60 to I67, Q70 to Q77
4	**8-pt Input Module**	I70 to I77
5	**Analog I/O Module**	See page 24-8.
6	**4/4-pt Mixed I/O Module**	I80 to I83, Q80 to Q83
7	**32-pt Input Module**	I90 to I97, I100 to I107, I110 to I117, I120 to I127

The I/O numbers of the CPU module start with I0 and Q0. The I/O numbers of the expansion I/O modules start with I30 and Q30. When an I/O module is mounted next to a 4/4-point mixed I/O module, note that the allocation numbers skip four points as shown above.

Input and output modules may be grouped together for easy identification of I/O numbers. When the I/O modules are relocated, the I/O numbers are renumbered automatically.

7: BASIC INSTRUCTIONS

Introduction

This chapter describes programming of the basic instructions, available operands, and sample programs.

All basic instructions are available on all MicroSmart CPU modules.

Basic Instruction List

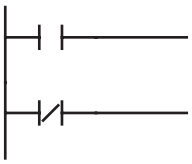
Symbol	Name	Function	Qty of Bytes	See Page
AND	And	Series connection of NO contact	4	7-4
AND LOD	And Load	Series connection of circuit blocks	5	7-5
ANDN	And Not	Series connection of NC contact	4	7-4
BPP	Bit Pop	Restores the result of bit logical operation which was saved temporarily	2	7-6
BPS	Bit Push	Saves the result of bit logical operation temporarily	5	7-6
BRD	Bit Read	Reads the result of bit logical operation which was saved temporarily	3	7-6
CC=	Counter Comparison (=)	Equal to comparison of counter current value	7	7-14
CC≥	Counter Comparison (≥)	Greater than or equal to comparison of counter current value	7	7-14
CDP	Dual Pulse Reversible Counter	Dual pulse reversible counter (0 to 65535)	4	7-10
CNT	Adding Counter	Adding counter (0 to 65535)	4	7-10
CUD	Up/Down Selection Reversible Counter	Up/down selection reversible counter (0 to 65535)	4	7-10
DC=	Data Register Comparison (=)	Equal to comparison of data register value	8	7-16
DC≥	Data Register Comparison (≥)	Greater than or equal to comparison of data register value	8	7-16
END	End	Ends a program	2	7-26
JEND	Jump End	Ends a jump instruction	4	7-25
JMP	Jump	Jumps a designated program area	4	7-25
LOD	Load	Stores intermediate results and reads contact status	6	7-2
LODN	Load Not	Stores intermediate results and reads inverted contact status	6	7-2
MCR	Master Control Reset	Ends a master control	4	7-23
MCS	Master Control Set	Starts a master control	4	7-23
OR	Or	Parallel connection of NO contact	4	7-4
OR LOD	Or Load	Parallel connection of circuit blocks	5	7-5
ORN	Or Not	Parallel connection of NC contact	4	7-4
OUT	Output	Outputs the result of bit logical operation	6	7-2
OUTN	Output Not	Outputs the inverted result of bit logical operation	6	7-2
RST	Reset	Resets output, internal relay, or shift register bit	6	7-3
SET	Set	Sets output, internal relay, or shift register bit	6	7-3
SFR	Shift Register	Forward shift register	6	7-18
SFRN	Shift Register Not	Reverse shift register	6	7-18
SOTD	Single Output Down	Falling-edge differentiation output	5	7-22
SOTU	Single Output Up	Rising-edge differentiation output	5	7-22
TIM	100-ms Timer	Subtracting 100-ms timer (0 to 6553.5 sec)	4	7-7
TMH	10-ms Timer	Subtracting 10-ms timer (0 to 655.35 sec)	4	7-7
TML	1-sec Timer	Subtracting 1-sec timer (0 to 65535 sec)	4	7-7
TMS	1-ms Timer	Subtracting 1-ms timer (0 to 65.535 sec)	4	7-7

LOD (Load) and LODN (Load Not)

The LOD instruction starts the logical operation with a NO (normally open) contact. The LODN instruction starts the logical operation with a NC (normally closed) contact.

A total of eight LOD and/or LODN instructions can be programmed consecutively.

Ladder Diagram



Valid Operands

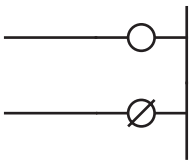
Instruction	I	Q	M	T	C	R
LOD LODN	0-307	0-307	0-1277 8000-8157	0-99	0-99	0-127

The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.

OUT (Output) and OUTN (Output Not)

The OUT instruction outputs the result of bit logical operation to the specified operand. The OUTN instruction outputs the inverted result of bit logical operation to the specified operand.

Ladder Diagram



Valid Operands

Instruction	I	Q	M	T	C	R
OUT OUTN	—	0-307	0-1277 8000-8077	—	—	—

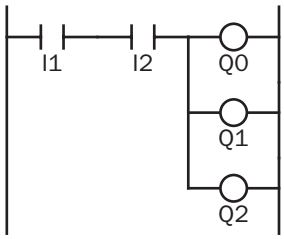
The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.

Caution • For restrictions on ladder programming of OUT and OUTN instructions, see page 29-22.

Multiple OUT and OUTN

There is no limit to the number of OUT and OUTN instructions that can be programmed into one rung.

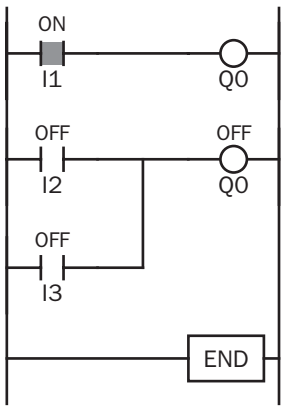
Ladder Diagram



Programming multiple outputs of the same output number is not recommended. However, when doing so, it is good practice to separate the outputs with the JMP/JEND set of instructions, or the MCS/MCR set of instructions. These instructions are detailed later in this chapter.

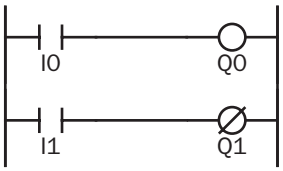
When the same output number is programmed more than once within one scan, the output nearest to the END instruction is given priority for outputting. In the example on the right, output Q0 is off.

Ladder Diagram



Examples: LOD (Load), OUT (Output), and NOT

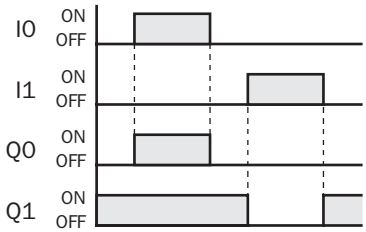
Ladder Diagram



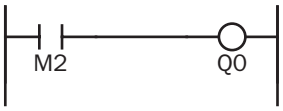
Program List

Instruction	Data
LOD	I0
OUT	Q0
LOD	I1
OUTN	Q1

Timing Chart



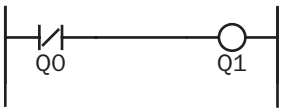
Ladder Diagram



Program List

Instruction	Data
LOD	M2
OUT	Q0

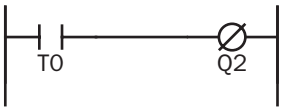
Ladder Diagram



Program List

Instruction	Data
LODN	Q0
OUT	Q1

Ladder Diagram



Program List

Instruction	Data
LOD	T0
OUTN	Q2

Ladder Diagram

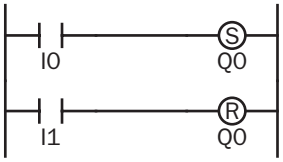
Program List

Instruction	Data
LODN	C1
OUT	Q10

SET and RST (Reset)

The SET and RST (reset) instructions are used to set (on) or reset (off) outputs, internal relays, and shift register bits. The same output can be set and reset many times within a program. SET and RST instructions operate in every scan while the input is on.

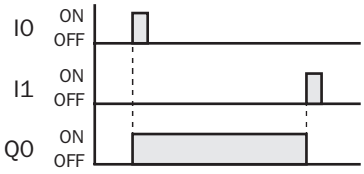
Ladder Diagram



Program List

Instruction	Data
LOD	I0
SET	Q0
LOD	I1
RST	Q0

Timing Chart



Valid Operands

Instruction	I	Q	M	T	C	R
SET RST	—	0-307	0-1277 8000-8077	—	—	0-127

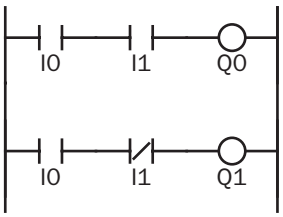
The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.

Caution • For restrictions on ladder programming of SET and RST instructions, see page 29-22.

AND and ANDN (And Not)

The AND instruction is used for programming a NO contact in series. The ANDN instruction is used for programming a NC contact in series. The AND or ANDN instruction is entered after the first set of contacts.

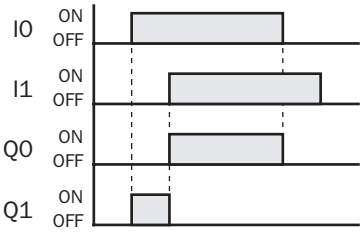
Ladder Diagram



Program List

Instruction	Data
LOD	I0
AND	I1
OUT	Q0
LOD	I0
ANDN	I1
OUT	Q1

Timing Chart



When both inputs I0 and I1 are on, output Q0 is on. When either input I0 or I1 is off, output Q0 is off.

When input I0 is on and input I1 is off, output Q1 is on. When either input I0 is off or input I1 is on, output Q1 is off.

Valid Operands

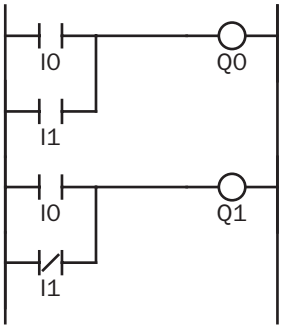
Instruction	I	Q	M	T	C	R
AND ANDN	0-307	0-307	0-1277 8000-8157	0-99	0-99	0-127

The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.

OR and ORN (Or Not)

The OR instruction is used for programming a NO contact in parallel. The ORN instruction is used for programming a NC contact in parallel. The OR or ORN instruction is entered after the first set of contacts.

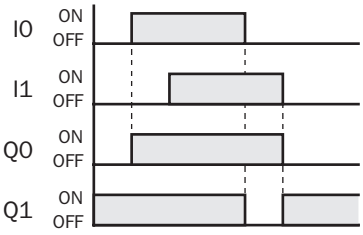
Ladder Diagram



Program List

Instruction	Data
LOD	I0
OR	I1
OUT	Q0
LOD	I0
ORN	I1
OUT	Q1

Timing Chart



When either input I0 or I1 is on, output Q0 is on. When both inputs I0 and I1 are off, output Q0 is off.

When either input I0 is on or input I1 is off, output Q1 is on. When input I0 is off and input I1 is on, output Q1 is off.

Valid Operands

Instruction	I	Q	M	T	C	R
OR ORN	0-307	0-307	0-1277 8000-8157	0-99	0-99	0-127

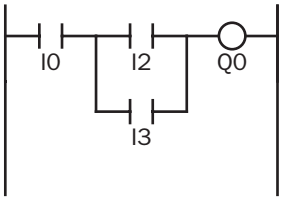
The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.

AND LOD (Load)

The AND LOD instruction is used to connect, in series, two or more circuits starting with the LOD instruction. The AND LOD instruction is the equivalent of a "node" on a ladder diagram.

When using WindLDR, the user need not program the AND LOD instruction. The circuit in the ladder diagram shown below is converted into AND LOD when the ladder diagram is compiled.

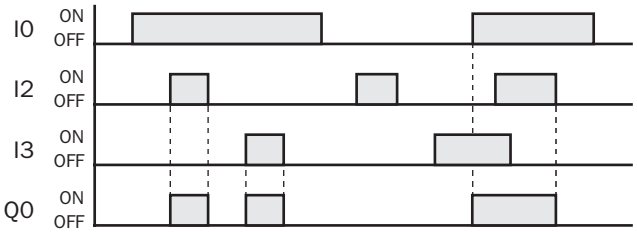
Ladder Diagram



Program List

Instruction	Data
LOD	I0
LOD	I2
OR	I3
ANDLOD	
OUT	Q0

Timing Chart



When input I0 is on and either input I2 or I3 is on, output Q0 is on.

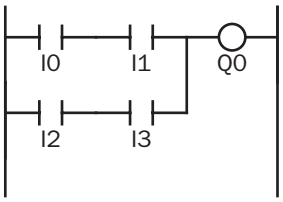
When input I0 is off or both inputs I2 and I3 are off, output Q0 is off.

OR LOD (Load)

The OR LOD instruction is used to connect, in parallel, two or more circuits starting with the LOD instruction. The OR LOD instruction is the equivalent of a "node" on a ladder diagram.

When using WindLDR, the user need not program the OR LOD instruction. The circuit in the ladder diagram shown below is converted into OR LOD when the ladder diagram is compiled.

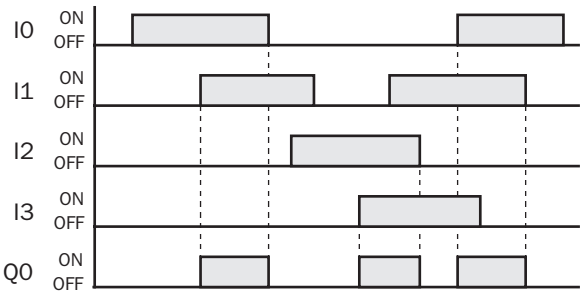
Ladder Diagram



Program List

Instruction	Data
LOD	I0
AND	I1
LOD	I2
AND	I3
ORLOD	
OUT	Q0

Timing Chart



When both inputs I0 and I1 are on or both inputs I2 and I3 are on, output Q0 is on.

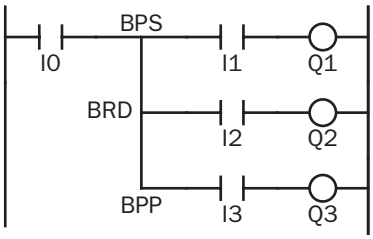
When either input I0 or I1 is off and either input I2 or I3 is off, output Q0 is off.

BPS (Bit Push), BRD (Bit Read), and BPP (Bit Pop)

The BPS (bit push) instruction is used to save the result of bit logical operation temporarily.
The BRD (bit read) instruction is used to read the result of bit logical operation which was saved temporarily.
The BPP (bit pop) instruction is used to restore the result of bit logical operation which was saved temporarily.

When using WindLDR, the user need not program the BPS, BRD, and BPP instructions. The circuit in the ladder diagram shown below is converted into BPS, BRD, and BPP when the ladder diagram is compiled.

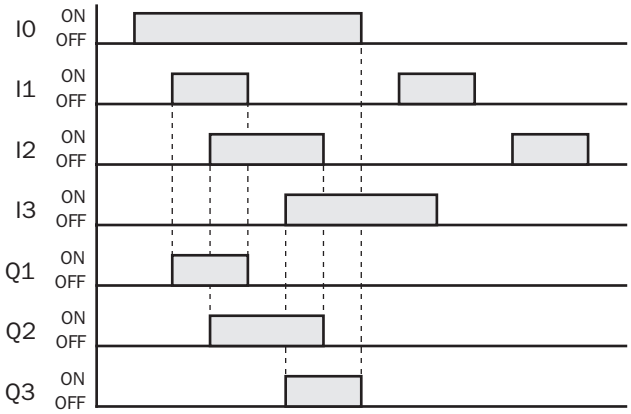
Ladder Diagram



Program List

Instruction	Data
LOD	I0
BPS	
AND	I1
OUT	Q1
BRD	
AND	I2
OUT	Q2
BPP	
AND	I3
OUT	Q3

Timing Chart



When both inputs I0 and I1 are on, output Q1 is turned on.

When both inputs I0 and I2 are on, output Q2 is turned on.

When both inputs I0 and I3 are on, output Q3 is turned on.

TML, TIM, TMH, and TMS (Timer)

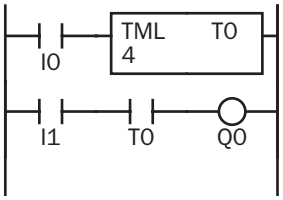
Four types of timedown timers are available; 1-sec timer TML, 100-ms timer TIM, 10-ms timer TMH, and 1-ms timer TMS. A total of 32 timers (all-in-one 10-I/O type CPU module) or 100 timers (other CPU modules) can be programmed in a user program. Each timer must be allocated to a unique number T0 through T31 or T99.

Timer	Allocation Number	Range	Increments	Preset Value
TML (1-sec timer)	T0 to T99	0 to 65535 sec	1 sec	Constant: 0 to 65535 Data registers: D0 to D1299 D2000 to D7999
TIM (100-ms timer)	T0 to T99	0 to 6553.5 sec	100 ms	
TMH (10-ms timer)	T0 to T99	0 to 655.35 sec	10 ms	
TMS (1-ms timer)	T0 to T99	0 to 65.535 sec	1 ms	

The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.
The preset value can be 0 through 65535 and designated using a decimal constant or data register.

TML (1-sec Timer)

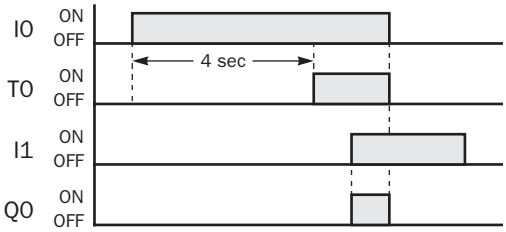
Ladder Diagram (TML)



Program List

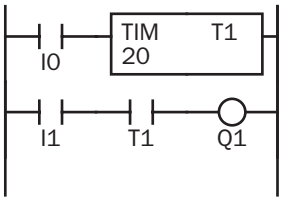
Instruction	Data
LOD	I0
TML	T0
	4
LOD	I1
AND	T0
OUT	Q0

Timing Chart



TIM (100-ms Timer)

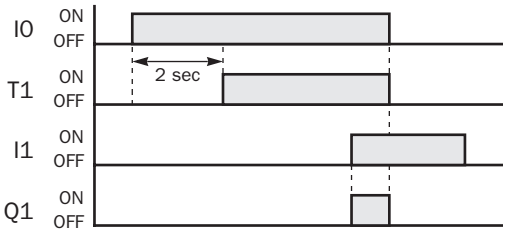
Ladder Diagram (TIM)



Program List

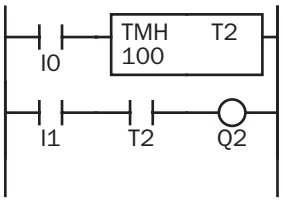
Instruction	Data
LOD	I0
TIM	T1
	20
LOD	I1
AND	T1
OUT	Q1

Timing Chart



TMH (10-ms Timer)

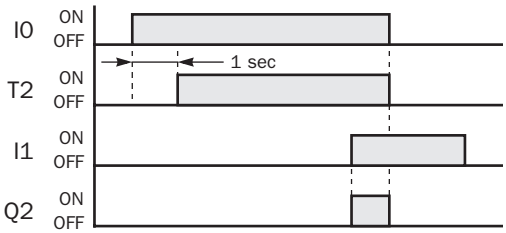
Ladder Diagram (TMH)



Program List

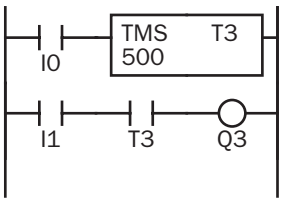
Instruction	Data
LOD	I0
TMH	T2
	100
LOD	I1
AND	T2
OUT	Q2

Timing Chart



TMS (1-ms Timer)

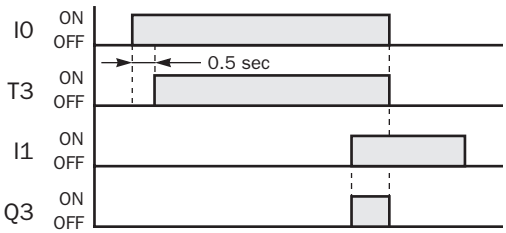
Ladder Diagram (TMS)



Program List

Instruction	Data
LOD	I0
TMS	T3
	500
LOD	I1
AND	T3
OUT	Q3

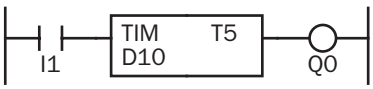
Timing Chart



Timer Circuit

The preset value 0 through 65535 can be designated using a data register D0 through D1299 or D2000 through D7999; then the data of the data register becomes the preset value. Directly after the TML, TIM, TMH, or TMS instruction, the OUT, OUTN, SET, RST, TML, TIM, TMH, or TMS instruction can be programmed.

Ladder Diagram



Program List

Instruction	Data
LOD	I1
TIM	T5
	D10
OUT	Q0

Caution

- For restrictions on ladder programming of timer instructions, see page 29-22.

- Timedown from the preset value is initiated when the operation result directly before the timer input is on.
- The timer output turns on when the current value (timed value) reaches 0.
- The current value returns to the preset value when the timer input is off.
- Timer preset and current values can be changed using WindLDR without downloading the entire program to the CPU again. From the WindLDR menu bar, select **Online > Monitor**, then select **Online > Point Write**. To change a timer preset value, specify the timer number with a capital T and a new preset value. If the timer preset value is changed during timedown, the timer remains unchanged for that cycle. The change will be reflected in the next time cycle. To change a timer current value, specify the timer number with a small t and a new current value while the timer is in operation. The change takes effect immediately.
- If the timer preset value is changed to 0, then the timer stops operation, and the timer output is turned on immediately.
- If the current value is changed during timedown, the change becomes effective immediately.
- For the data movement when changing, confirming, and clearing preset values, see page 7-13. Preset values can also be changed and changed preset values can be confirmed using the HMI module. See pages 5-35 and 5-36.

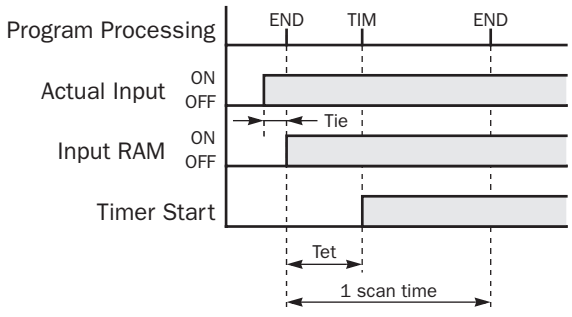
Timer Accuracy

Timer accuracy due to software configuration depends on three factors: timer input error, timer counting error, and timeout output error. These errors are not constant but vary with the user program and other causes.

Timer Input Error

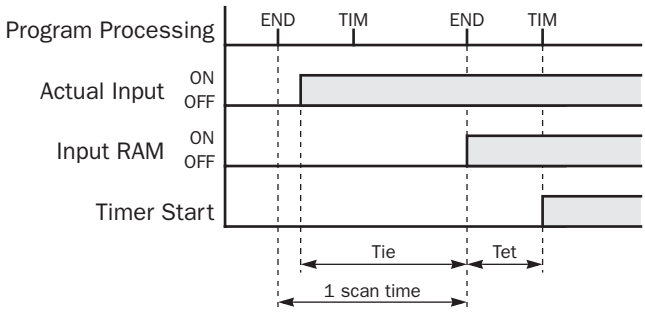
The input status is read at the END processing and stored to the input RAM. So, an error occurs depending on the timing when the timer input turns on in a scan cycle. The same error occurs on the normal input and the catch input. The timer input error shown below does not include input delay caused by the hardware.

Minimum Error



When the input turns on immediately before the END processing, Tie is almost 0. Then the timer input error is only Tet (behind error) and is at its minimum.

Maximum Error



When the input turns on immediately after the END processing, Tie is almost equal to one scan time. Then the timer input error is Tie + Tet = one scan time + Tet (behind error) and is at its maximum.

Tie: Time from input turning on to the END processing
Tet: Time from the END processing to the timer instruction execution

Timer Accuracy, continued

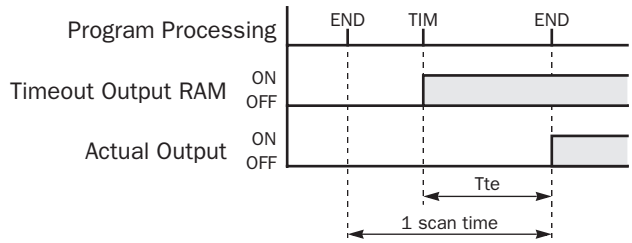
Timer Counting Error

Every timer instruction operation is individually based on asynchronous 16-bit reference timers. Therefore, an error occurs depending on the status of the asynchronous 16-bit timer when the timer instruction is executed.

Error		TML (1-sec timer)	TIM (100-ms timer)	TMH (10-ms timer)	TMS (1-ms timer)
Maximum	Advance error	1000 ms	100 ms	10 ms	1 ms
	Behind error	1 scan time	1 scan time	1 scan time	1 scan time

Timeout Output Error

The output RAM status is set to the actual output when the END instruction is processed. So, an error occurs depending on the timing when the timeout output turns on in a scan cycle. The timeout output error shown below does not include output delay caused by the hardware.



Timeout output error is equal to Tte (behind error) and can be between 0 and one scan time.

0 < Tte < 1 scan time

Tte: Time from the timer instruction execution to the END processing

Maximum and Minimum of Errors

Error		Timer Input Error	Timer Counting Error	Timeout Output Error	Total Error
Minimum	Advance error	0 (Note)	0	0 (Note)	0
	Behind error	Tet	0	Tte	0
Maximum	Advance error	0 (Note)	Increment	0 (Note)	Increment – (Tet + Tte)
	Behind error	1 scan time + Tet	1 scan time	Tte	2 scan times + (Tet + Tte)

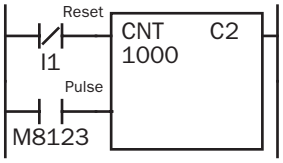
Notes: Advance error does not occur at the timer input and timeout output.
Tet + Tte = 1 scan time
Increment is 1 sec (TML), 100 ms (TIM), 10 ms (TMH), or 1 ms (TMS).
The maximum advance error is: Increment – 1 scan time
The maximum behind error is: 3 scan times

The timer input error and timeout output error shown above do not include the input response time (behind error) and output response time (behind error) caused by hardware.

Power Failure Memory Protection

Timers TML, TIM, TMH, and TMS do not have power failure protection. A timer with this protection can be devised using a counter instruction and special internal relay M8121 (1-sec clock), M8122 (100-ms clock), or M8123 (10-ms clock).

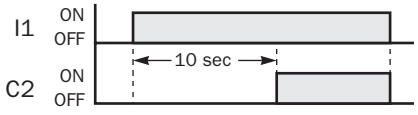
Ladder Diagram
(10-sec Timer)



Program List

Instruction	Data
LODN	I1
LOD	M8123
CNT	C2
	1000

Timing Chart



Note: Designate counter C2 used in this program as a keep type counter. See page 5-4.

CNT, CDP, and CUD (Counter)

Three types of counters are available; adding (up) counter CNT, dual-pulse reversible counter CDP, and up/down selection reversible counter CUD. A total of 32 counters (all-in-one 10-I/O type CPU module) or 100 counters (other CPU modules) can be programmed in a user program. Each counter must be allocated to a unique number C0 through C31 or C99.

Counter	Allocation Number	Preset Value
CNT (adding counter)	C0 to C99	Constant: 0 to 65535 Data registers: D0 to D1299 D2000 to D7999
CDP (dual-pulse reversible counter)	C0 to C99	
CUD (up/down selection reversible counter)	C0 to C99	

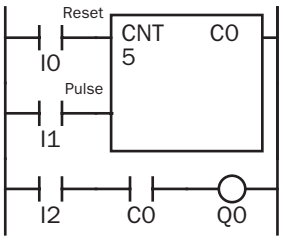
The valid operand range depends on the CPU module type. For details, see pages 6-1 and 6-2.
The preset value can be 0 through 65535 and designated using a decimal constant or data register.

CNT (Adding Counter)

When counter instructions are programmed, two addresses are required. The circuit for an adding (UP) counter must be programmed in the following order: reset input, pulse input, the CNT instruction, and a counter number C0 through C99, followed by a counter preset value from 0 to 65535.

The preset value can be designated using a decimal constant or a data register. When a data register is used, the data of the data register becomes the preset value.

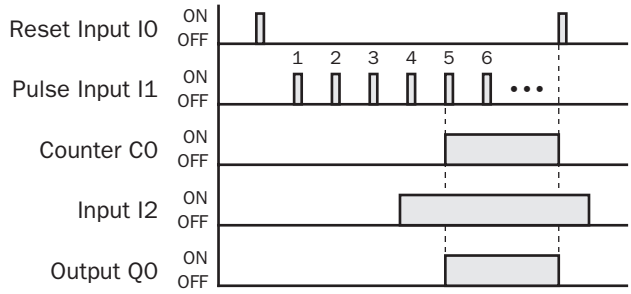
Ladder Diagram



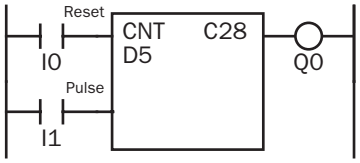
Program List

Instruction	Data
LOD	I0
LOD	I1
CNT	C0
	5
LOD	I2
AND	C0
OUT	Q0

Timing Chart



- The preset value 0 through 65535 can be designated using a data register D0 through D1299 or D2000 through D7999; then the data of the data register becomes the preset value. Directly after the CNT instruction, the OUT, OUTN, SET, RST, TML, TIM, TMH, or TMS instruction can be programmed.



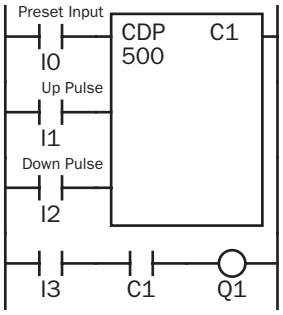
- The same counter number cannot be programmed more than once.
- While the reset input is off, the counter counts the leading edges of pulse inputs and compares them with the preset value.
- When the current value reaches the preset value, the counter turns output on. The output stays on until the reset input is turned on.
- When the reset input changes from off to on, the current value is reset.
- When the reset input is on, all pulse inputs are ignored.
- The reset input must be turned off before counting may begin.
- When power is off, the counter's current value is held, and can also be designated as "clear" type counters using Function Area Settings (see page 5-4).
- Counter preset and current values can be changed using WindLDR without downloading the entire program to the CPU again. From the WindLDR menu bar, select **Online > Monitor**, then select **Online > Point Write**. To change a counter preset value, specify the counter number with a capital C and a new preset value. To change a counter current value, specify the counter number with a small c and a new current value while the counter reset input is off.
- When the preset or current value is changed during counter operation, the change becomes effective immediately.
- For the data movement when changing, confirming, and clearing preset values, see page 7-13. Preset values can also be changed and changed preset values can be confirmed using the HMI module. See pages 5-35 and 5-36.

CDP (Dual-Pulse Reversible Counter)

The dual-pulse reversible counter CDP has up and down pulse inputs, so that three inputs are required. The circuit for a dual-pulse reversible counter must be programmed in the following order: preset input, up-pulse input, down-pulse input, the CDP instruction, and a counter number C0 through C99, followed by a counter preset value from 0 to 65535.

The preset value can be designated using a decimal constant or a data register. When a data register is used, the data of the data register becomes the preset value.

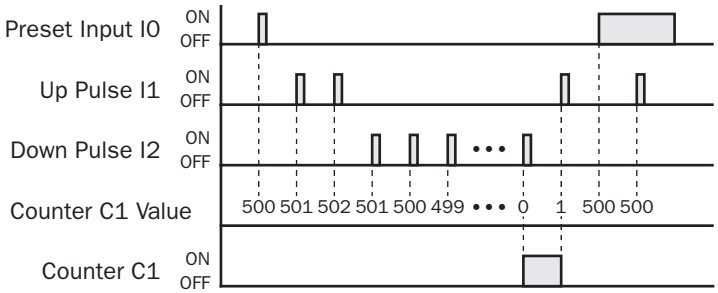
Ladder Diagram



Program List

Instruction	Data
LOD	I0
LOD	I1
LOD	I2
CDP	C1
	500
LOD	I3
AND	C1
OUT	Q1

Timing Chart



Caution

- For restrictions on ladder programming of counter instructions, see page 29-22.

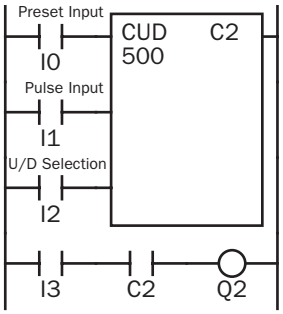
- The same counter number cannot be programmed more than once.
- The preset input must be turned on initially so that the current value returns to the preset value.
- The preset input must be turned off before counting may begin.
- When the up pulse and down pulses are on simultaneously, no pulse is counted.
- The counter output is on only when the current value is 0.
- After the current value reaches 0 (counting down), it changes to 65535 on the next count down.
- After the current value reaches 65535 (counting up), it changes to 0 on the next count up.
- When power is off, the counter's current value is held, and can also be designated as "clear" type counters using the Function Area Settings (see page 5-4).
- Counter preset and current values can be changed using WindLDR without downloading the entire program to the CPU again. From the WindLDR menu bar, select **Online > Monitor**, then select **Online > Point Write**. To change a counter preset value, specify the counter number with a capital C and a new preset value. To change a counter current value, specify the counter number with a small c and a new current value while the counter preset input is off.
- When the preset or current value is changed during counter operation, the change becomes effective immediately.
- For the data movement when changing, confirming, and clearing preset values, see page 7-13. Preset values can also be changed and changed preset values can be confirmed using the HMI module. See pages 5-35 and 5-36.

CUD (Up/Down Selection Reversible Counter)

The up/down selection reversible counter CUD has a selection input to switch the up/down gate, so that three inputs are required. The circuit for an up/down selection reversible counter must be programmed in the following order: preset input, pulse input, up/down selection input, the CUD instruction, and a counter number C0 through C99, followed by a counter preset value from 0 to 65535.

The preset value can be designated using a decimal constant or a data register. When a data register is used, the data of the data register becomes the preset value.

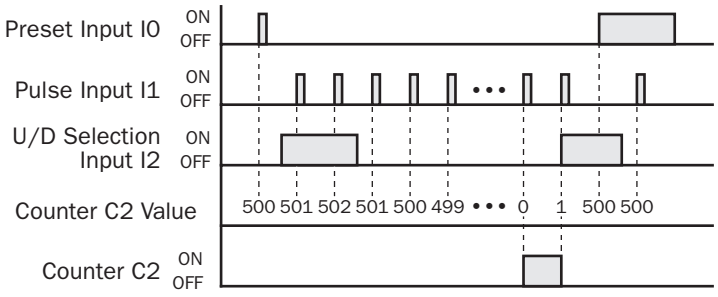
Ladder Diagram



Program List

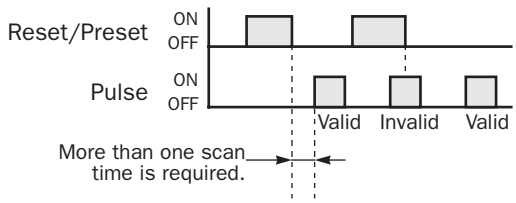
Instruction	Data
LOD	I0
LOD	I1
LOD	I2
CUD	C2
	500
LOD	I3
AND	C2
OUT	Q2

Timing Chart



Valid Pulse Inputs

The reset or preset input has priority over the pulse input. One scan after the reset or preset input has changed from on to off, the counter starts counting the pulse inputs as they change from off to on.



Caution

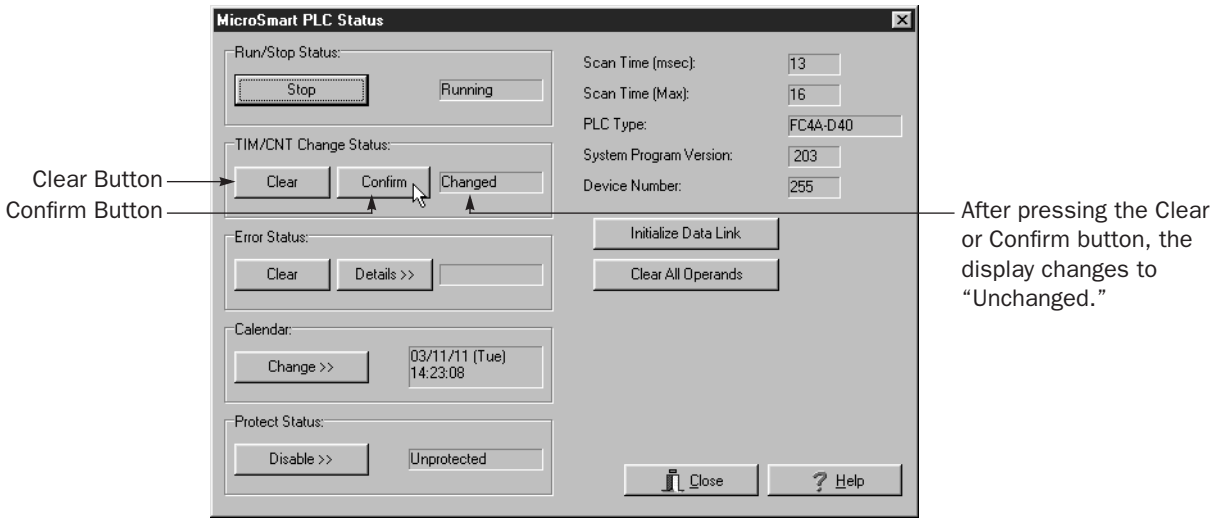
- For restrictions on ladder programming of counter instructions, see page 29-22.

- The same counter number cannot be programmed more than once.
- The preset input must be turned on initially so that the current value returns to the preset value.
- The preset input must be turned off before counting may begin.
- The up mode is selected when the up/down selection input is on.
- The down mode is selected when the up/down selection input is off.
- The counter output is on only when the current value is 0.
- After the current value reaches 0 (counting down), it changes to 65535 on the next count down.
- After the current value reaches 65535 (counting up), it changes to 0 on the next count up.
- When power is off, the counter's current value is held, and can also be designated as "clear" type counters using the Function Area Settings (see page 5-4).
- Counter preset and current values can be changed using WindLDR without downloading the entire program to the CPU again. From the WindLDR menu bar, select **Online > Monitor**, then select **Online > Point Write**. To change a counter preset value, specify the counter number with a capital C and a new preset value. To change a counter current value, specify the counter number with a small c and a new current value while the counter preset input is off.
- When the preset or current value is changed during counter operation, the change becomes effective immediately.
- For the data movement when changing, confirming, and clearing preset values, see page 7-13. Preset values can also be changed and changed preset values can be confirmed using the HMI module. See pages 5-35 and 5-36.

Changing, Confirming, and Clearing Preset Values for Timers and Counters

Preset values for timers and counters can be changed using the Point Write command on WindLDR for transferring a new value to the MicroSmart CPU module RAM as described on preceding pages. After changing the preset values temporarily, the changes can be written to the user program in the MicroSmart CPU module EEPROM or cleared from the RAM.

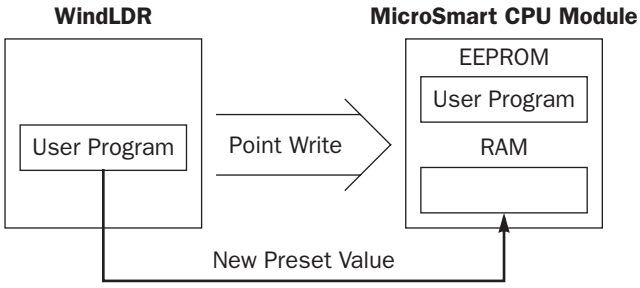
Access the PLC Status dialog box from the Online menu in the monitoring mode.



Data movement when changing a timer/counter preset value

When changing a timer/counter preset value using Point Write on WindLDR, the new preset value is written to the MicroSmart CPU module RAM. The user program and preset values in the EEPROM are not changed.

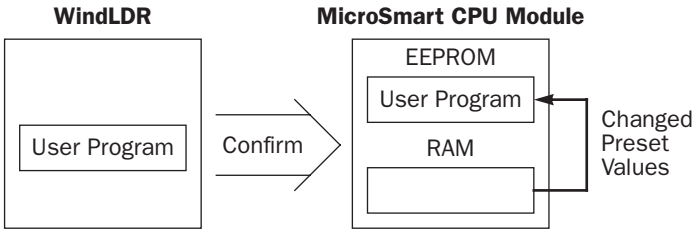
Note: The HMI module can also be used to change preset values and confirm changed preset values. See pages 5-35 and 5-36.



Data movement when confirming changed preset values

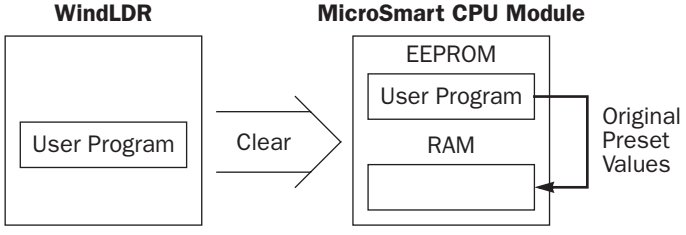
When the Confirm button is pressed before pressing the Clear button, the changed timer/counter preset values in the MicroSmart CPU module RAM are written to the EEPROM.

When uploading the user program after confirming, the user program with changed preset values is uploaded from the MicroSmart CPU module EEPROM to WindLDR.



Data movement when clearing changed preset values to restore original values

Changing preset values for timers and counters in the MicroSmart CPU module RAM does not automatically update preset values in the user memory, EEPROM. This is useful for restoring original preset values. When the Clear button is pressed before pressing the Confirm button, the changed timer/counter preset values are cleared from the RAM and the original preset values are loaded from the EEPROM to the RAM.



CC= and CC≥ (Counter Comparison)

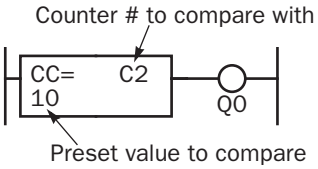
The CC= instruction is an equivalent comparison instruction for counter current values. This instruction will constantly compare current values to the value that has been programmed in. When the counter value equals the given value, the desired output will be initiated.

The CC≥ instruction is an equal to or greater than comparison instruction for counter current values. This instruction will constantly compare current values to the value that has been programmed in. When the counter value is equal to or greater than the given value, the desired output will be initiated.

When a counter comparison instruction is programmed, two addresses are required. The circuit for a counter comparison instruction must be programmed in the following order: the CC= or CC≥ instruction; a counter number C0 through C31 (all-in-one 10-I/O type CPU module) or C99 (other CPU modules); followed by a preset value to compare from 0 to 65535.

The preset value can be designated using a decimal constant or a data register D0 through D399 (all-in-one 10-I/O type CPU module) or D1299 (other CPU modules), or D2000 through D7999 (slim type CPU modules). When a data register is used, the data of the data register becomes the preset value.

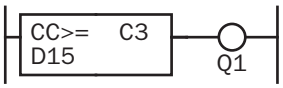
Ladder Diagram (CC=)



Program List

Instruction	Data
CC=	C2
	10
OUT	Q0

Ladder Diagram (CC≥)

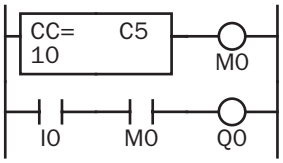


Program List

Instruction	Data
CC>=	C3
	D15
OUT	Q1

- The CC= and CC≥ instructions can be used repeatedly for different preset values.
- The comparison instructions only compare the current value. The status of the counter does not affect this function.
- The comparison instructions also serve as an implicit LOD instruction, and must be programmed at the beginning of a ladder line.
- The comparison instructions can be used with internal relays, which are ANDed or ORed at a separate program address.
- Like the LOD instruction, the comparison instructions can be followed by the AND and OR instructions.

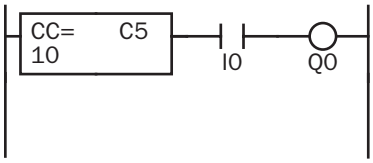
Ladder Diagram



Program List

Instruction	Data
CC=	C5
	10
OUT	M0
LOD	I0
AND	M0
OUT	Q0

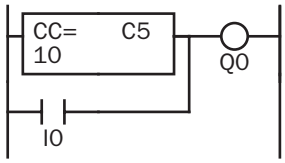
Ladder Diagram



Program List

Instruction	Data
CC=	C5
	10
AND	I0
OUT	Q0

Ladder Diagram

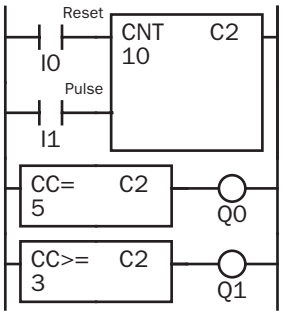


Program List

Instruction	Data
CC=	C5
	10
OR	I0
OUT	Q0

Examples: CC= and CC≥ (Counter Comparison)

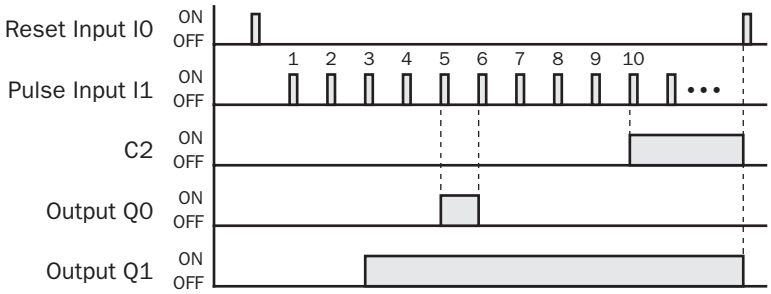
Ladder Diagram 1



Program List

Instruction	Data
LOD	I0
LOD	I1
CNT	C2
	10
CC=	C2
	5
OUT	Q0
CC≥	C2
	3
OUT	Q1

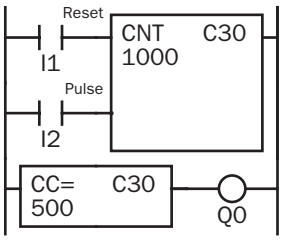
Timing Chart



Output Q0 is on when counter C2 current value is 5.

Output Q1 is turned on when counter C2 current value reaches 3 and remains on until counter C2 is reset.

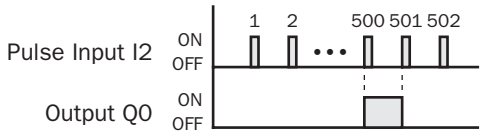
Ladder Diagram 2



Program List

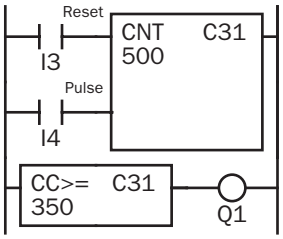
Instruction	Data
LOD	I1
LOD	I2
CNT	C30
	1000
CC=	C30
	500
OUT	Q0

Timing Chart



Output Q0 is on when counter C30 current value is 500.

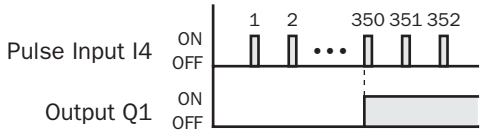
Ladder Diagram 3



Program List

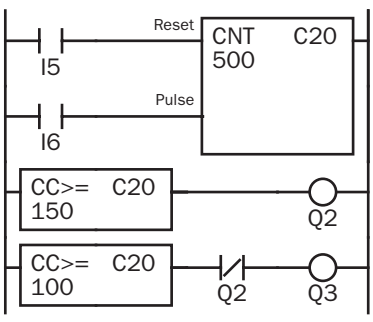
Instruction	Data
LOD	I3
LOD	I4
CNT	C31
	500
CC>=	C31
	350
OUT	Q1

Timing Chart



Output Q1 is turned on when counter C31 current value reaches 350 and remains on until counter C31 is reset.

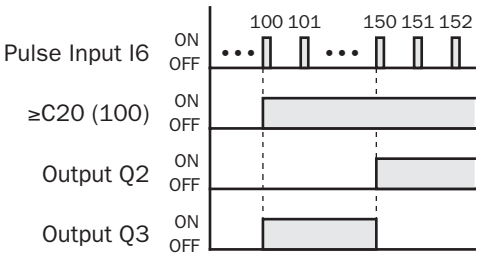
Ladder Diagram 4



Program List

Instruction	Data
LOD	I5
LOD	I6
CNT	C20
	500
CC>=	C20
	150
OUT	Q2
CC>=	C20
	100
ANDN	Q2
OUT	Q3

Timing Chart



Output Q3 is on when counter C20 current value is between 100 and 149.

DC= and DC≥ (Data Register Comparison)

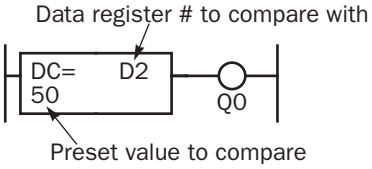
The DC= instruction is an equivalent comparison instruction for data register values. This instruction will constantly compare data register values to the value that has been programmed in. When the data register value equals the given value, the desired output will be initiated.

The DC≥ instruction is an equal to or greater than comparison instruction for data register values. This instruction will constantly compare data register values to the value that has been programmed in. When the data register value is equal to or greater than the given value, the desired output will be initiated.

When a data register comparison instruction is programmed, two addresses are required. The circuit for a data register comparison instruction must be programmed in the following order: the DC= or DC≥ instruction, a data register number D0 through D399 (all-in-one 10-I/O type CPU module) or D1299 (other CPU modules), or D2000 through D7999 (slim type CPU modules); followed by a preset value to compare from 0 to 65535.

The preset value can be designated using a decimal constant or a data register D0 through D399 (all-in-one 10-I/O type CPU module) or D1299 (other CPU modules), or D2000 through D7999 (slim type CPU modules). When a data register is used, the data of the data register becomes the preset value.

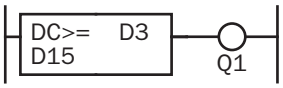
Ladder Diagram (DC=)



Program List

Instruction	Data
DC=	D2
	50
OUT	Q0

Ladder Diagram (DC≥)

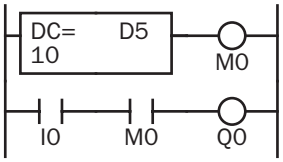


Program List

Instruction	Data
DC>=	D3
	D15
OUT	Q1

- The DC= and DC≥ instructions can be used repeatedly for different preset values.
- The comparison instructions also serve as an implicit LOD instruction, and must be programmed at the beginning of a ladder line.
- The comparison instructions can be used with internal relays, which are ANDed or ORed at a separate program address.
- Like the LOD instruction, the comparison instructions can be followed by the AND and OR instructions.

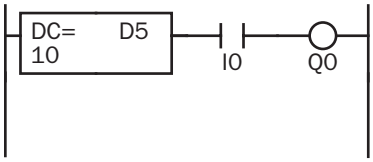
Ladder Diagram



Program List

Instruction	Data
DC=	D5
	10
OUT	M0
LOD	I0
AND	M0
OUT	Q0

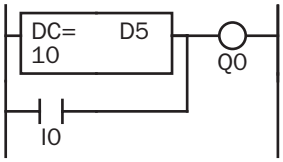
Ladder Diagram



Program List

Instruction	Data
DC=	D5
	10
AND	I0
OUT	Q0

Ladder Diagram

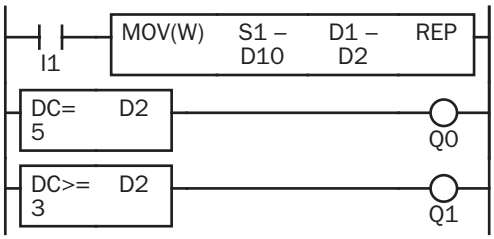


Program List

Instruction	Data
DC=	D5
	10
OR	I0
OUT	Q0

Examples: DC= and DC≥ (Data Register Comparison)

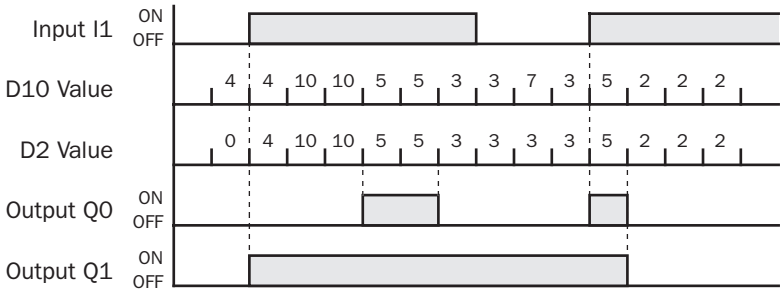
Ladder Diagram 1



Program List

Instruction	Data
LOD	I1
MOV(W)	
	D10 –
	D2 –
DC=	D2
	5
OUT	Q0
DC≥	D2
	3
OUT	Q1

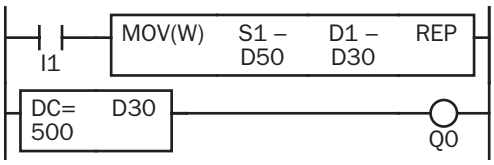
Timing Chart



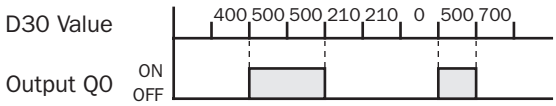
Output Q0 is on when data register D2 value is 5.

Output Q1 is on when data register D2 value is 3 or more.

Ladder Diagram 2

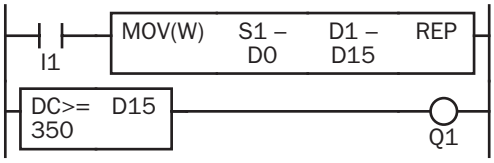


Timing Chart

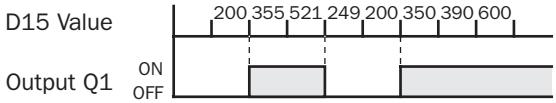


Output Q0 is on when data register D30 value is 500.

Ladder Diagram 3

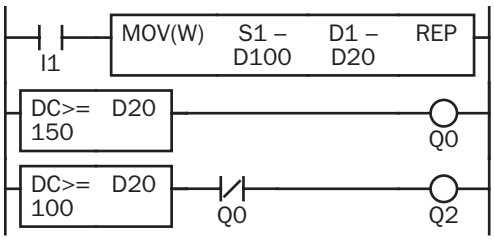


Timing Chart

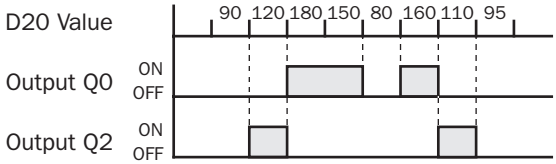


Output Q1 is on when data register D15 value is 350 or more.

Ladder Diagram 4



Timing Chart



Output Q2 is on while data register D20 value is between 149 and 100.

SFR and SFRN (Forward and Reverse Shift Register)

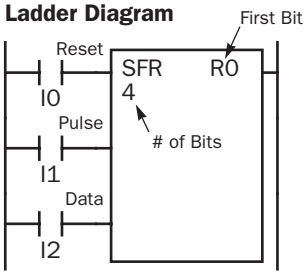
The shift register consists of a total of 64 bits (all-in-one 10-I/O type CPU module) or 128 bits (other CPU modules) which are allocated to R0 through R63 or R127, respectively. Any number of available bits can be selected to form a train of bits which store on or off status. The on/off data of constituent bits is shifted in the forward direction (forward shift register) or in the reverse direction (reverse shift register) when a pulse input is turned on.

Forward Shift Register (SFR)

When SFR instructions are programmed, two addresses are always required. The SFR instruction is entered, followed by a shift register number selected from appropriate operand numbers. The shift register number corresponds to the first, or head bit. The number of bits is the second required address after the SFR instruction.

The SFR instruction requires three inputs. The forward shift register circuit must be programmed in the following order: reset input, pulse input, data input, and the SFR instruction, followed by the first bit and the number of bits.

Ladder Diagram

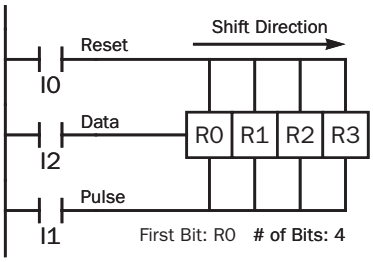


CPU Type	All-in-One 10-I/O	Others
First Bit	R0 to R63	R0 to R127
# of Bits	1 to 64	1 to 128

Program List

Instruction	Data
LOD	I0
LOD	I1
LOD	I2
SFR	R0
	4

Structural Diagram



Reset Input

The reset input will cause the value of each bit of the shift register to return to zero. Initialize pulse special internal relay, M8120, may be used to initialize the shift register at start-up.

Pulse Input

The pulse input triggers the data to shift. The shift is in the forward direction for a forward shift register and in reverse for a reverse shift register. A data shift will occur upon the leading edge of a pulse; that is, when the pulse *turns on*. If the pulse has been on and stays on, no data shift will occur.

Data Input

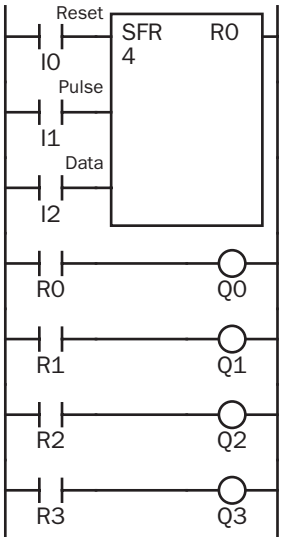
The data input is the information which is shifted into the first bit when a forward data shift occurs, or into the last bit when a reverse data shift occurs.

Note: When power is turned off, the statuses of all shift register bits are normally cleared. It is also possible to maintain the statuses of shift register bits by using the Function Area Settings as required. See page 5-4.

Caution • For restrictions on ladder programming of shift register instructions, see page 29-22.

Forward Shift Register (SFR), continued

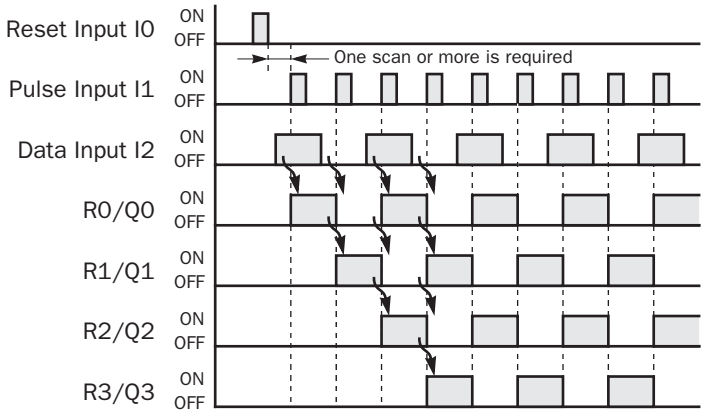
Ladder Diagram



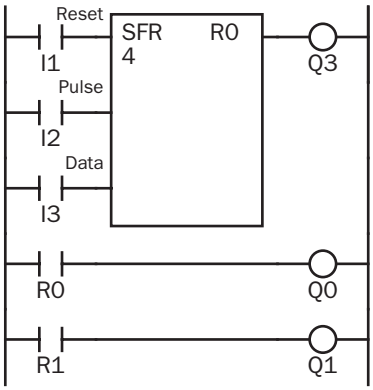
Program List

Instruction	Data
LOD	I0
LOD	I1
LOD	I2
SFR	R0
	4
LOD	R0
OUT	Q0
LOD	R1
OUT	Q1
LOD	R2
OUT	Q2
LOD	R3
OUT	Q3

Timing Chart



Ladder Diagram

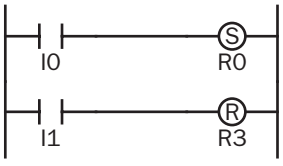


Program List

Instruction	Data
LOD	I1
LOD	I2
LOD	I3
SFR	R0
	4
OUT	Q3
LOD	R0
OUT	Q0
LOD	R1
OUT	Q1

- The last bit status output can be programmed directly after the SFR instruction. In this example, the status of bit R3 is read to output Q3.
- Each bit can be loaded using the LOD R# instruction.

Setting and Resetting Shift Register Bits



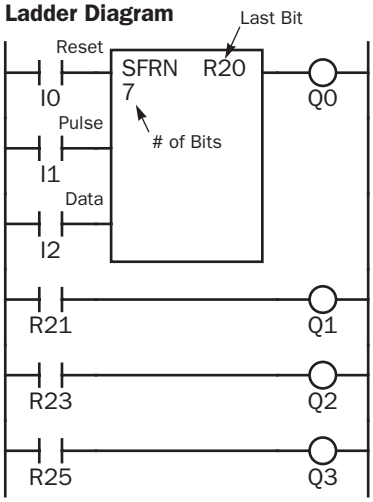
- Any shift register bit can be turned on using the SET instruction.
- Any shift register bit can be turned off using the RST instruction.
- The SET or RST instruction is actuated by any input condition.

Reverse Shift Register (SFRN)

For reverse shifting, use the SFRN instruction. When SFRN instructions are programmed, two addresses are always required. The SFRN instructions are entered, followed by a shift register number selected from appropriate operand numbers. The shift register number corresponds to the lowest bit number in a string. The number of bits is the second required address after the SFRN instructions.

The SFRN instruction requires three inputs. The reverse shift register circuit must be programmed in the following order: reset input, pulse input, data input, and the SFRN instruction, followed by the last bit and the number of bits.

Ladder Diagram



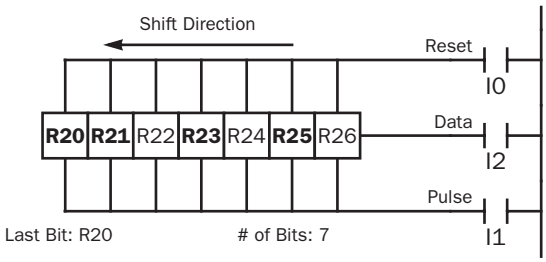
CPU Type	All-in-One 10-I/O	Others
Last Bit	R0 to R63	R0 to R127
# of Bits	1 to 64	1 to 128

Program List

Instruction	Data
LOD	I0
LOD	I1
LOD	I2
SFRN	R20
	7
OUT	Q0
LOD	R21
OUT	Q1
LOD	R23
OUT	Q2
LOD	R25
OUT	Q3

- The last bit status output can be programmed directly after the SFRN instruction. In this example, the status of bit R20 is read to output Q0.
- Each bit can be loaded using the LOD R# instructions.
- For details of reset, pulse, and data inputs, see page 7-18.

Structural Diagram



Note: Output is initiated only for those bits highlighted in bold print.

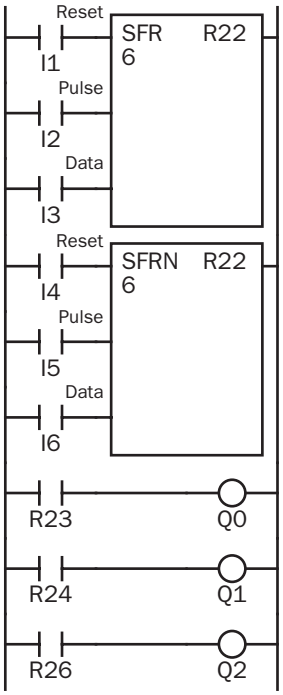
Note: When power is turned off, the statuses of all shift register bits are normally cleared. It is also possible to maintain the statuses of shift register bits by using the Function Area Settings as required. See page 5-4.

Caution • For restrictions on ladder programming of shift register instructions, see page 29-22.

Bidirectional Shift Register

A bidirectional shift register can be created by first programming the SFR instruction as detailed in the Forward Shift Register section on page 7-18. Next, the SFRN instruction is programed as detailed in the Reverse Shift Register section on page 7-20.

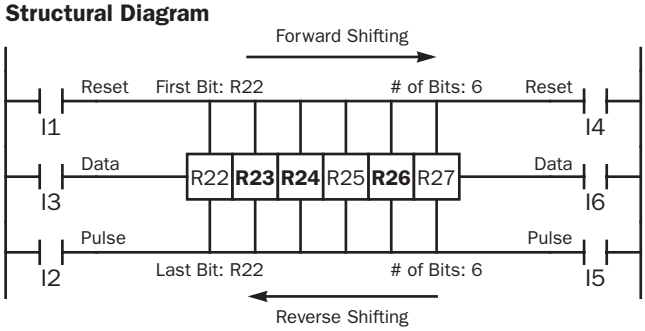
Ladder Diagram



Program List

Instruction	Data
LOD	I1
LOD	I2
LOD	I3
SFR	R22
	6
LOD	I4
LOD	I5
LOD	I6
SFRN	R22
	6
LOD	R23
OUT	Q0
LOD	R24
OUT	Q1
LOD	R26
OUT	Q2

Structural Diagram



Note: Output is initiated only for those bits highlighted in bold print.

SOTU and SOTD (Single Output Up and Down)

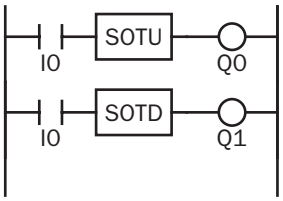
The SOTU instruction "looks for" the transition of a given input from off to on. The SOTD instruction looks for the transition of a given input from on to off. When this transition occurs, the desired output will turn on for the length of one scan. The SOTU or SOTD instruction converts an input signal to a "one-shot" pulse signal.

A total of 512 (all-in-one 10-I/O type CPU module) or 3072 (other CPU modules) SOTU and SOTD instructions can be used in a user program.

If operation is started while the given input is already on, the SOTU output will not turn on. The transition from off to on is what triggers the SOTU instruction.

When a relay of the CPU or relay output module is defined as the SOTU or SOTD output, it may not operate if the scan time is not compatible with relay requirements.

Ladder Diagram



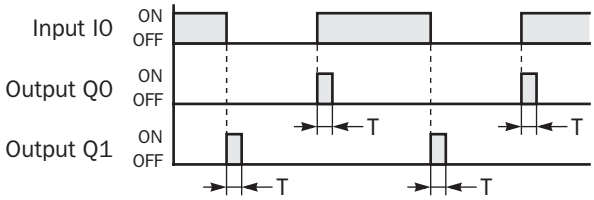
Program List

Instruction	Data
LOD	I0
SOTU	
OUT	Q0
LOD	I0
SOTD	
OUT	Q1

Caution

- For restrictions on ladder programming of SOTU and SOTD instructions, see page 29-22.

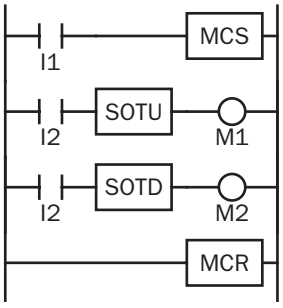
Timing Chart



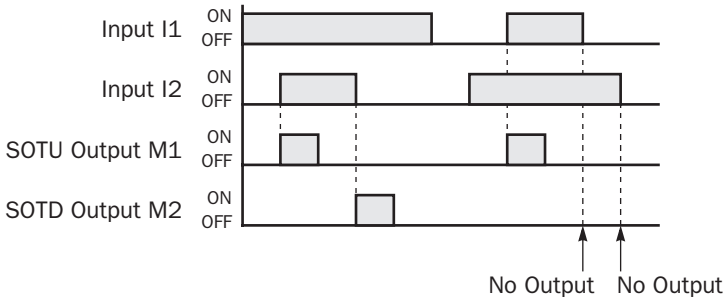
Note: "T" equals one scan time (one-shot pulse).

There is a special case when the SOTU and SOTD instructions are used between the MCS and MCR instructions (which are detailed on page 7-23). If input I2 to the SOTU instruction turns on while input I1 to the MCS instruction is on, then the SOTU output turns on. If input I2 to the SOTD instruction turns off while input I1 is on, then the SOTD output turns on. If input I1 turns on while input I2 is on, then the SOTU output turns on. However, if input I1 turns off while input I2 is on, then the SOTD output does not turn on as shown below.

Ladder Diagram



Timing Chart



MCS and MCR (Master Control Set and Reset)

The MCS (master control set) instruction is usually used in combination with the MCR (master control reset) instruction. The MCS instruction can also be used with the END instruction, instead of the MCR instruction.

When the input preceding the MCS instruction is off, the MCS is executed so that all inputs to the portion between the MCS and the MCR are forced off. When the input preceding the MCS instruction is on, the MCS is not executed so that the program following it is executed according to the actual input statuses.

When the input condition to the MCS instruction is off and the MCS is executed, other instructions between the MCS and MCR are executed as follows:

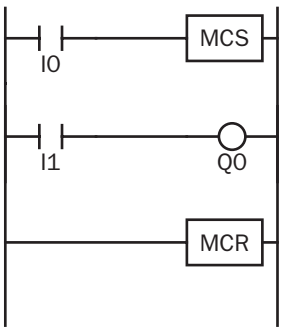
Instruction	Status
SOTU	Rising edges (ON pulses) are not detected.
SOTD	Falling edges (OFF pulses) are not detected.
OUT	All are turned off.
OUTN	All are turned on.
SET and RST	All are held in current status.
TML, TIM, TMH, and TMS	Current values are reset to zero. Timeout statuses are turned off.
CNT, CDP, and CUD	Current values are held. Pulse inputs are turned off. Countout statuses are turned off.
SFR and SFRN	Shift register bit statuses are held. Pulse inputs are turned off. The output from the last bit is turned off.

Input conditions cannot be set for the MCR instruction.

More than one MCS instruction can be used with one MCR instruction.

Corresponding MCS/MCR instructions cannot be nested within another pair of corresponding MCS/MCR instructions.

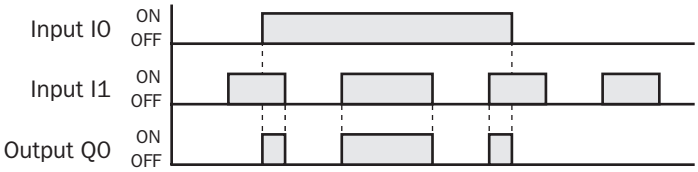
Ladder Diagram



Program List

Instruction	Data
LOD	I0
MCS	
LOD	I1
OUT	Q0
MCR	

Timing Chart



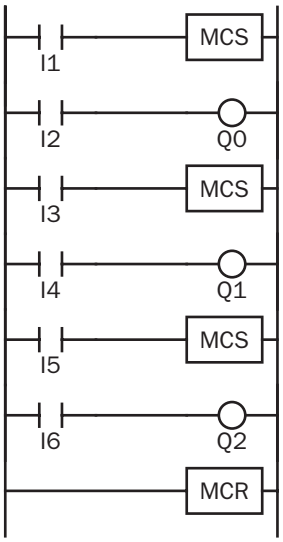
When input I0 is off, MCS is executed so that the subsequent input is forced off.

When input I0 is on, MCS is not executed so that the following program is executed according to the actual input statuses.

MCS and MCR (Master Control Set and Reset), continued

Multiple Usage of MCS instructions

Ladder Diagram



Program List

Instruction	Data
LOD	I1
MCS	
LOD	I2
OUT	Q0
LOD	I3
MCS	
LOD	I4
OUT	Q1
LOD	I5
MCS	
LOD	I6
OUT	Q2
MCR	

This master control circuit will give priority to I1, I3, and I5, in that order.

When input I1 is off, the first MCS is executed so that subsequent inputs I2 through I6 are forced off.

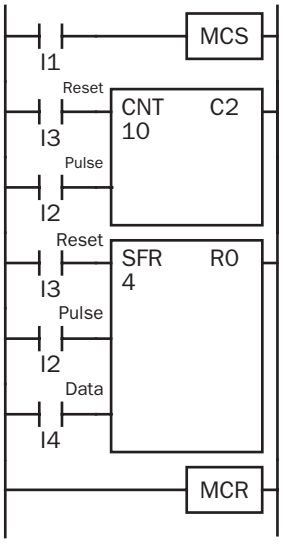
When input I1 is on, the first MCS is not executed so that the following program is executed according to the actual input statuses of I2 through I6.

When I1 is on and I3 is off, the second MCS is executed so that subsequent inputs I4 through I6 are forced off.

When both I1 and I3 are on, the first and second MCSs are not executed so that the following program is executed according to the actual input statuses of I4 through I6.

Counter and Shift Register in Master Control Circuit

Ladder Diagram

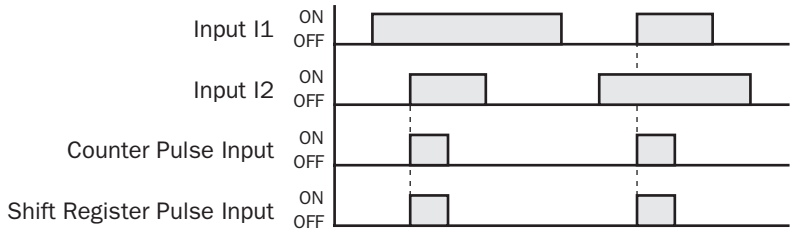


When input I1 is on, the MCS is not executed so that the counter and shift register are executed according to actual statuses of subsequent inputs I2 through I4.

When input I1 is off, the MCS is executed so that subsequent inputs I2 through I4 are forced off.

When input I1 is turned on while input I2 is on, the counter and shift register pulse inputs are turned on as shown below.

Timing Chart



JMP (Jump) and JEND (Jump End)

The JMP (jump) instruction is usually used in combination with the JEND (jump end) instruction. At the end of a program, the JMP instruction can also be used with the END instruction, instead of the JEND instruction.

These instructions are used to proceed through the portion of the program between the JMP and the JEND *without* processing. This is similar to the MCS/MCR instructions, except that the portion of the program between the MCS and MCR instruction *is* executed.

When the operation result immediately before the JMP instruction is on, the JMP is valid and the program is *not* executed. When the operation result immediately before the JMP instruction is off, the JMP is invalid and the program is executed.

When the input condition to the JMP instruction is on and the JMP is executed, other instructions between the JMP and JEND are executed as follows:

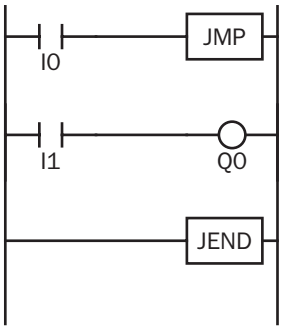
Instruction	Status
SOTU	Rising edges (ON pulses) are not detected.
SOTD	Falling edges (OFF pulses) are not detected.
OUT and OUTN	All are held in current status.
SET and RST	All are held in current status.
TML, TIM, TMH, and TMS	Current values are held. Timeout statuses are held.
CNT, CDP, and CUD	Current values are held. Pulse inputs are turned off. Countout statuses are held.
SFR and SFRN	Shift register bit statuses are held. Pulse inputs are turned off. The output from the last bit is held.

Input conditions cannot be set for the JEND instruction.

More than one JMP instruction can be used with one JEND instruction.

Corresponding JMP/JEND instructions cannot be nested within another pair of corresponding JMP/JEND instructions.

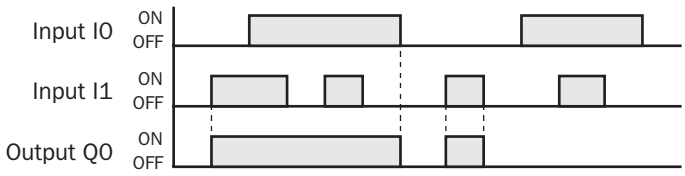
Ladder Diagram



Program List

Instruction	Data
LOD	I0
JMP	
LOD	I1
OUT	Q0
JEND	

Timing Chart

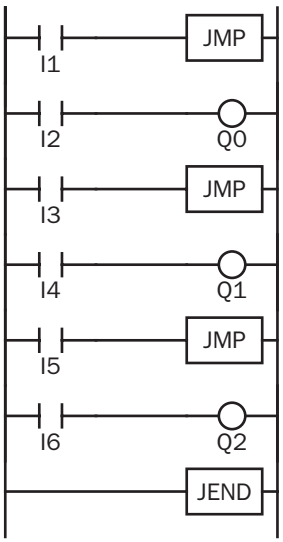


When input I0 is on, JMP is executed so that the subsequent output status is held.

When input I0 is off, JMP is not executed so that the following program is executed according to the actual input statuses.

JMP (Jump) and JEND (Jump End), continued

Ladder Diagram



Program List

Instruction	Data
LOD	I1
JMP	
LOD	I2
OUT	Q0
LOD	I3
JMP	
LOD	I4
OUT	Q1
LOD	I5
JMP	
LOD	I6
OUT	Q2
JEND	

This jump circuit will give priority to I1, I3, and I5, in that order.

When input I1 is on, the first JMP is executed so that subsequent output statuses of Q0 through Q2 are held.

When input I1 is off, the first JMP is not executed so that the following program is executed according to the actual input statuses of I2 through I6.

When I1 is off and I3 is on, the second JMP is executed so that subsequent output statuses of Q1 and Q2 are held.

When both I1 and I3 are off, the first and second JMPs are not executed so that the following program is executed according to the actual input statuses of I4 through I6.

END

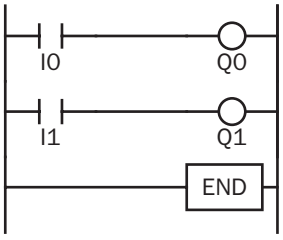
The END instruction is always required at the end of a program; however, it is not necessary to program the END instruction after the last programmed instruction. The END instruction already exists at every unused address. (When an address is used for programming, the END instruction is removed.)

A *scan* is the execution of all instructions from address zero to the END instruction. The time required for this execution is referred to as one *scan time*. The scan time varies with respect to program length, which corresponds to the address where the END instruction is found.

During the scan time, program instructions are processed sequentially. This is why the output instruction closest to the END instruction has priority over a previous instruction for the same output. No output is initiated until all logic within a scan is processed.

Output occurs simultaneously, and this is the first part of the END instruction execution. The second part of the END instruction execution is to monitor all inputs, also done simultaneously. Then program instructions are ready to be processed sequentially once again.

Ladder Diagram



Program List

Instruction	Data
LOD	I0
OUT	Q0
LOD	I1
OUT	Q1
END	

8: ADVANCED INSTRUCTIONS

Introduction

This chapter describes general rules of using advanced instructions, terms, data types, and formats used for advanced instructions.

Advanced Instruction List

Group	Symbol	Name	Data Type		Qty of Bytes	See Page
			W	I		
NOP	NOP	No Operation			2	8-7
Move	MOV	Move	X	X	16	9-1
	MOVN	Move Not	X	X	16	9-4
	IMOV	Indirect Move	X		24 to 28	9-5
	IMOVN	Indirect Move Not	X		24 to 28	9-6
	BMOV	Block Move	X		18	9-7
	IBMV	Indirect Bit Move			24	9-8
	IBMVN	Indirect Bit Move Not			24	9-10
Data Comparison	CMP=	Compare Equal To	X	X	20	10-1
	CMP<>	Compare Unequal To	X	X	20	10-1
	CMP<	Compare Less Than	X	X	20	10-1
	CMP>	Compare Greater Than	X	X	20	10-1
	CMP<=	Compare Less Than or Equal To	X	X	20	10-1
	CMP>=	Compare Greater Than or Equal To	X	X	20	10-1
	ICMP>=	Interval Compare Greater Than or Equal To	X	X	22	10-4
Binary Arithmetic	ADD	Addition	X	X	20	11-1
	SUB	Subtraction	X	X	20	11-1
	MUL	Multiplication	X	X	20	11-1
	DIV	Division	X	X	20	11-1
	ROOT	Root	X		14	11-7
Boolean Computation	ANDW	AND Word	X		20	12-1
	ORW	OR Word	X		20	12-1
	XORW	Exclusive OR Word	X		20	12-1
Shift and Rotate	SFTL	Shift Left	X		12	13-1
	SFTR	Shift Right	X		12	13-3
	BCDLS	BCD Left Shift	X		14	13-4
	WSFT	Word Shift	X		18	13-5
	ROTL	Rotate Left	X		12	13-6
	ROTR	Rotate Right	X		12	13-7

Group	Symbol	Name	Data Type		Qty of Bytes	See Page
			W	I		
Data Conversion	HTOB	Hex to BCD	X		14	14-1
	BTOH	BCD to Hex	X		14	14-2
	HTOA	Hex to ASCII	X		18	14-3
	ATOH	ASCII to Hex	X		18	14-5
	BTOA	BCD to ASCII	X		18	14-7
	ATOB	ASCII to BCD	X		18	14-9
	ENCO	Encode	X		16	14-11
	DECO	Decode	X		16	14-12
	BCNT	Bit Count	X		18	14-13
	ALT	Alternate Output	X		10	14-14
Week Programmer	WKTIM	Week Timer	X		24	15-1
	WKTBL	Week Table	X		13 to 89	15-2
Interface	DISP	Display	X		16	16-1
	DGRD	Digital Read	X		20	16-3
User Communication	TXD1	Transmit 1	X		21 to 819	17-6
	TXD2	Transmit 2	X		21 to 819	17-6
	RXD1	Receive 1	X		21 to 819	17-15
	RXD2	Receive 2	X		21 to 819	17-15
Program Branching	LABEL	Label	X		8	18-1
	LJMP	Label Jump	X		10	18-1
	LCAL	Label Call	X		10	18-3
	LRET	Label Return	X		6	18-3
	IOREF	I/O Refresh	X		16	18-5
	DI	Disable Interrupt	X		8	18-7
	EI	Enable Interrupt	X		8	18-7
Coordinate Conversion	XYFS	XY Format Set	X	X	24 to 124	19-1
	CVXTY	Convert X to Y	X	X	18	19-2
	CVYTX	Convert Y to X	X	X	18	19-3
Pulse	PULS1	Pulse Output 1	X		12	20-1
	PULS2	Pulse Output 2	X		12	20-1
	PWM1	Pulse Width Modulation 1	X		24	20-7
	PWM2	Pulse Width Modulation 2	X		24	20-7
	RAMP	Ramp Pulse Output	X		14	20-13
	ZRN1	Zero Return 1	X		18	20-24
	ZRN2	Zero Return 2	X		18	20-24
PID Instruction	PID	PID Control	X		26	21-2
Dual / Teaching Timer	DTML	1-sec Dual Timer	X		22	22-1
	DTIM	100-ms Dual Timer	X		22	22-1
	DTMH	10-ms Dual Timer	X		22	22-1
	DTMS	1-ms Dual Timer	X		22	22-1
	TTIM	Teaching Timer	X		10	22-3
Intelligent Module Access	RUNA	Run Access	X	X	20	23-2
	STPA	Stop Access	X	X	20	23-4

Advanced Instruction Applicable CPU Modules

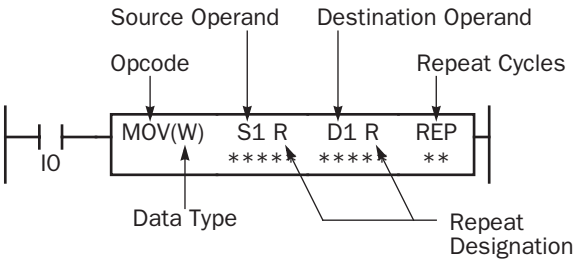
Applicable advanced instructions depend on the type of CPU modules as listed in the table below.

Group	Symbol	All-in-One Type CPU Modules			Slim Type CPU Modules	
		FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
NOP	NOP	X	X	X	X	X
Move	MOV	X	X	X	X	X
	MOVN	X	X	X	X	X
	IMOV	X	X	X	X	X
	IMOVN	X	X	X	X	X
	BMOV					X
	IBMV					X
	IBMVN					X
Data Comparison	CMP=	X	X	X	X	X
	CMP<>	X	X	X	X	X
	CMP<	X	X	X	X	X
	CMP>	X	X	X	X	X
	CMP<=	X	X	X	X	X
	CMP>=	X	X	X	X	X
	ICMP>=					X
Binary Arithmetic	ADD	X	X	X	X	X
	SUB	X	X	X	X	X
	MUL	X	X	X	X	X
	DIV	X	X	X	X	X
	ROOT	X	X	X	X	X
Boolean Computation	ANDW	X	X	X	X	X
	ORW	X	X	X	X	X
	XORW	X	X	X	X	X
Shift and Rotate	SFTL	X	X	X	X	X
	SFTR	X	X	X	X	X
	BCDLS					X
	WSFT					X
	ROTL	X	X	X	X	X
	ROTR	X	X	X	X	X
Data Conversion	HTOB	X	X	X	X	X
	BTOH	X	X	X	X	X
	HTOA	X	X	X	X	X
	ATOH	X	X	X	X	X
	BTOA	X	X	X	X	X
	ATOB	X	X	X	X	X
	ENCO					X
	DECO					X
	BCNT					X
	ALT					X

Group	Symbol	All-in-One Type CPU Modules			Slim Type CPU Modules	
		FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
Week Programmer	WKTIM	X	X	X	X	X
	WKTBL	X	X	X	X	X
Interface	DISP			X	X	X
	DGRD			X	X	X
User Communication	TXD1	X	X	X	X	X
	TXD2		X	X	X	X
	RXD1	X	X	X	X	X
	RXD2		X	X	X	X
Program Branching	LABEL	X	X	X	X	X
	LJMP	X	X	X	X	X
	LCAL	X	X	X	X	X
	LRET	X	X	X	X	X
	IOREF	X	X	X	X	X
	DI					X
	EI					X
Coordinate Conversion	XYFS			X	X	X
	CVXTY			X	X	X
	CVYTX			X	X	X
Pulse	PULS1				X	X
	PULS2				X	X
	PWM1				X	X
	PWM2				X	X
	RAMP				X	X
	ZRN1					X
	ZRN2					X
PID Instruction	PID			X	X	X
Dual / Teaching Timer	DTML					X
	DTIM					X
	DTMH					X
	DTMS					X
	TTIM					X
Intelligent Module Access	RUNA			▲	▲	X
	STPA			▲	▲	X

Advanced instructions marked with ▲ can be used on the CPU modules with system program version 204 or higher.

Structure of an Advanced Instruction



Repeat Designation
Specifies whether repeat is used for the operand or not.

Repeat Cycles
Specifies the quantity of repeat cycles: 1 through 99.

Opcode
The opcode is a symbol to identify the advanced instruction.

Data Type
Specifies the word (W) or integer (I) data type.

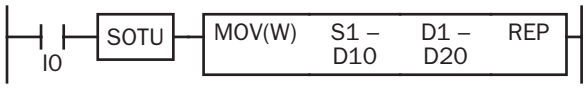
Source Operand
The source operand specifies the 16-bit data to be processed by the advanced instruction. Some advanced instructions require two source operands.

Destination Operand
The destination operand specifies the 16-bit data to store the result of the advanced instruction. Some advanced instructions require two destination operands.

Input Condition for Advanced Instructions

Almost all advanced instructions must be preceded by a contact, except NOP (no operation), LABEL (label), LRET (label return), and STPA (stop access) instructions. The input condition can be programmed using a bit operand such as input, output, internal relay, or shift register. Timer and counter can also be used as an input condition to turn on the contact when the timer times out or the counter counts out.

While the input condition is on, the advanced instruction is executed in each scan. To execute the advanced instruction only at the rising or falling edge of the input, use the SOTU or SOTD instruction.



While the input condition is off, the advanced instruction is not executed and operand statuses are held.

Source and Destination Operands

The source and destination operands specify 16-bit data. When a bit operand such as input, output, internal relay, or shift register is designated as a source or destination operand, 16 points starting with the designated number are processed as source or destination data. When a word operand such as timer or counter is designated as a source operand, the current value is read as source data. When a timer or counter is designated as a destination operand, the result of the advanced instruction is set to the preset value for the timer or counter. When a data register is designated as a source or destination operand, the data is read from or written to the designated data register.

Using Timer or Counter as Source Operand

Since all timer instructions — TML (1-sec timer), TIM (100-ms timer), TMH (10-ms timer), and TMS (1-ms timer) — subtract from the preset value, the current value is decremented from the preset value and indicates the remaining time. As described above, when a timer is designated as a source operand of an advanced instruction, the current value, or the remaining time, of the timer is read as source data. Adding counters CNT start counting at 0, and the current value is incremented up to the preset value. Reversible counters CDP and CUD start counting at the preset value and the current value is incremented or decremented from the preset value. When any counter is designated as a source operand of an advanced instruction, the current value is read as source data.

Using Timer or Counter as Destination Operand

As described above, when a timer or counter is designated as a destination operand of an advanced instruction, the result of the advanced instruction is set to the preset value of the timer or counter. Timer and counter preset values can be 0 through 65535.

When a timer or counter preset value is designated using a data register, the timer or counter cannot be designated as a destination of an advanced instruction. When executing such an advanced instruction, a user program execution error will result. For details of user program execution error, see page 29-6.

Note: When a user program execution error occurs, the result is not set to the destination.

Data Types for Advanced Instructions

When using the move, data comparison, and binary arithmetic instructions, data types can be selected from word (W) or integer (I). For other advanced instructions, the data is processed in units of 16-bit word; except the coordinate conversion instructions use the integer data type.

Data Type	Symbol	Bits	Quantity of Data Registers Used	Range of Decimal Values
Word (Unsigned 16 bits)	**W**	16 bits	1	0 to 65,535
Integer (Signed 15 bits)	**I**	16 bits	1	–32,768 to 32,767

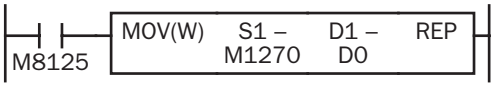
Decimal Values and Hexadecimal Storage

The following table shows hexadecimal equivalents which are stored in the CPU, as a result of addition and subtraction of the decimal values shown:

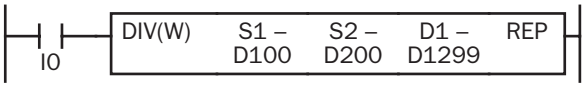
Data Type	Result of Addition	Hexadecimal Storage	Result of Subtraction	Hexadecimal Storage
Word	0	0000	65535	FFFF
	65535	FFFF	0	0000
	131071	(CY) FFFF	–1	(BW) FFFF
			–65535	(BW) 0001
			–65536	(BW) 0000
Integer	65534	(CY) 7FFE	65534	(BW) 7FFE
	32768	(CY) 0000	32768	(BW) 0000
	32767	7FFF	32767	7FFF
	0	0000	0	0000
	–1	FFFF	–1	FFFF
	–32767	8001	–32767	8001
	–32768	8000	–32768	8000
	–32769	(CY) FFFF	–32769	(BW) FFFF
	–65535	(CY) 8001	–65535	(BW) 8001

Discontinuity of Operand Areas

Each operand area is discrete and does not continue, for example, from input to output or from output to internal relay. In addition, special internal relays M8000 through M8157 are in a separate area from internal relays M0 through M1277. Data registers D0 through D1299, expansion data registers D2000 through D7999, and special data registers D8000 through D8199 are in separate areas and do not continue with each other.

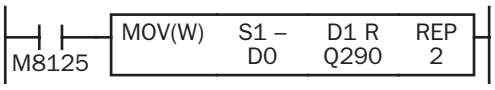


The internal relay ends at M1277. Since the MOV (move) instruction reads 16 internal relays, the last internal relay exceeds the valid range, resulting in a user program syntax error.



This program results in a user program syntax error. The destination of the DIV (division) instruction requires two data registers D1299 and D1300. Since D1300 exceeds the valid range, a user program syntax error occurs.

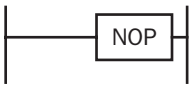
Advanced instructions execute operation only on the available operands in the valid area. If a user program syntax error is found during programming, WindLDR rejects the program instruction and shows an error message.



The MOV (move) instruction sets data of data register D0 to 16 outputs Q290 through Q307 in the first repeat cycle. The destination of the second cycle is the next 16 outputs Q310 through Q327, which are invalid, resulting in a user program syntax error.

For details about repeat operations of each advanced instruction, see the following chapters.

NOP (No Operation)



No operation is executed by the NOP instruction.

The NOP instruction may serve as a place holder. Another use would be to add a delay to the CPU scan time, in order to simulate communication with a machine or application, for debugging purposes.

The NOP instruction does not require an input and operand.

Details of all other advanced instructions are described in the following chapters.

9: MOVE INSTRUCTIONS

Introduction

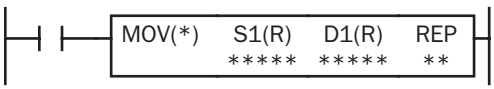
Data can be moved using the MOV (move), MOVN (move not), IMOV (indirect move), or IMOVN (indirect move not) instruction. The moved data is 16-bit data, and the repeat operation can also be used to increase the quantity of data moved. In the MOV or MOVN instruction, the source and destination operand are designated by S1 and D1 directly. In the IMOV or IMOVN instruction, the source and destination operand are determined by the offset values designated by S2 and D2 added to source operand S1 and destination operand D1.

The BMOV (block move) instruction is useful to move consecutive blocks of timer, counter, and data register values.

The IBMV (indirect bit move) and IBMVN (indirect bit move not) instructions move one bit of data from a source operand to a destination operand. Both operands are determined by adding an offset to the operand. When using the repeat operation, data of consecutive bits can be moved.

Since the move instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

MOV (Move)



S1 → D1
When input is on, 16- bit data from operand designated by S1 is moved to operand designated by D1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	First operand number to move	X	X	X	X	X	X	X	X	1-99
D1 (Destination 1)	First operand number to move to	—	X	▲	X	X	X	X	—	1-99

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1. Source operand can be both internal relays M0 through M1277 and special internal relays M8000 through M8157.

When T (timer) or C (counter) is used as S1, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Valid Data Types

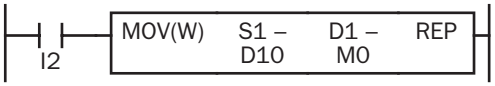
W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

Examples: MOV

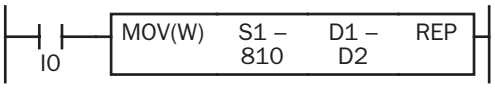
The following examples are described using the word data type. Data move operation for the integer data type is the same for the word data type.



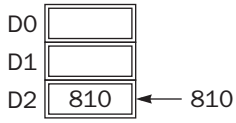
D10 → M0
When input I2 is on, the data in data register D10 designated by source operand S1 is moved to 16 internal relays starting with M0 designated by destination operand D1.

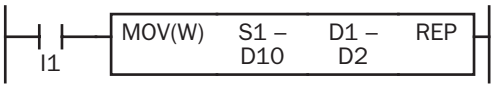
D10 [12345] ⟶ M0 through M7, M10 through M17

The data in the source data register is converted into 16-bit binary data, and the ON/OFF statuses of the 16 bits are moved to internal relays M0 through M7 and M10 through M17. M0 is the LSB (least significant bit). M17 is the MSB (most significant bit).

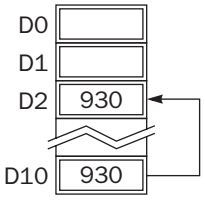


810 → D2
When input I0 is on, constant 810 designated by source operand S1 is moved to data register D2 designated by destination operand D1.





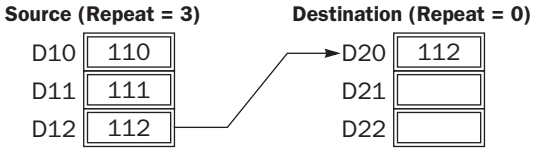
D10 → D2
When input I1 is on, the data in data register D10 designated by source operand S1 is moved to data register D2 designated by destination operand D1.



Repeat Operation in the Move Instructions

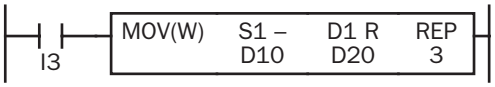
Repeat Source Operand

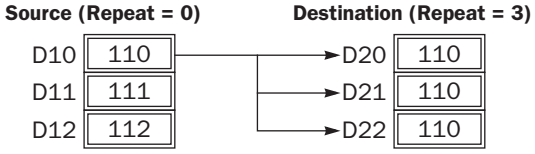
When the S1 (source) is designated with repeat, operands as many as the repeat cycles starting with the operand designated by S1 are moved to the destination. As a result, only the last of the source operands is moved to the destination.



Repeat Destination Operand

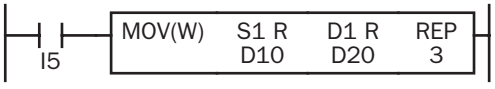
When the D1 (destination) is designated to repeat, the source operand designated by S1 is moved to all destination operands as many as the repeat cycles starting with the destination designated by D1.

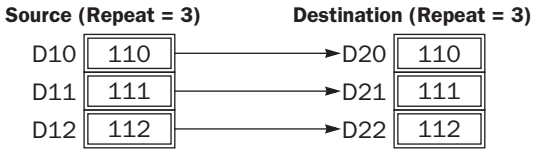




Repeat Source and Destination Operands

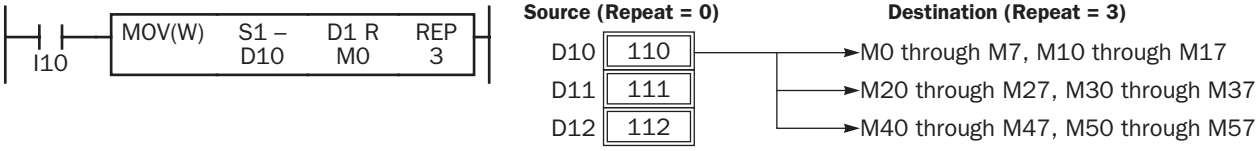
When both S1 (source) and D1 (destination) are designated to repeat, operands as many as the repeat cycles starting with the operand designated by S1 are moved to the same quantity of operands starting with the operand designated by D1.





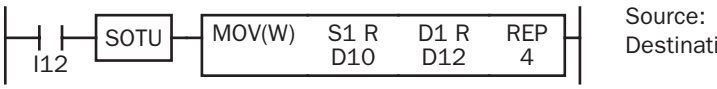
Repeat Bit Operands

The MOV (move) instruction moves 16-bit data. When a bit operand such as input, output, internal relay, or shift register is designated as the source or destination operand, 16 bits starting with the one designated by S1 or D1 are the target data. If a repeat operation is designated for a bit operand, the target data increases in 16-bit increments.

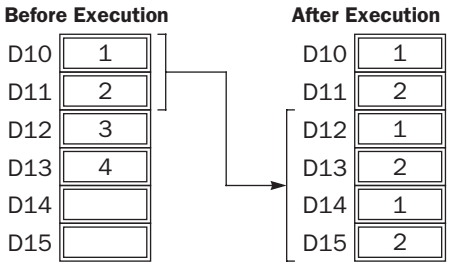


Overlapped Operands by Repeat

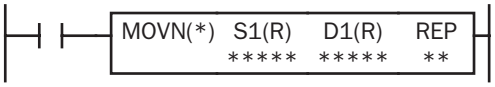
If the repeat operation is designated for both the source and destination and if a portion of the source and destination areas overlap each other, then the source data in the overlapped area is also changed.



Source: D10 through D13 (Repeat = 4)
Destination: D12 through D15 (Repeat = 4)



MOVN (Move Not)



S1 NOT → D1
When input is on, 16-bit data from operand designated by S1 is inverted bit by bit and moved to operand designated by D1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	First operand number to move	X	X	X	X	X	X	X	X	1-99
D1 (Destination 1)	First operand number to move to	—	X	▲	X	X	X	X	—	1-99

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

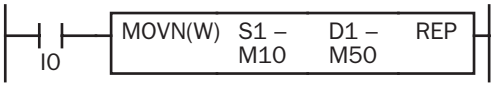
Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

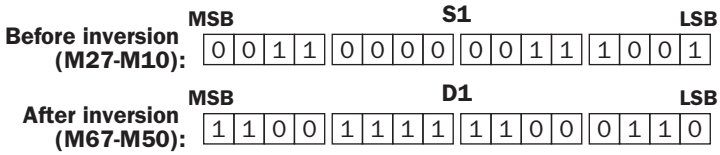
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

Examples: MOVN

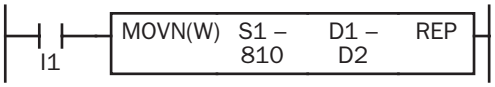


M10 NOT → M50
When input I0 is on, the 16 internal relays starting with M10 designated by source operand S1 are inverted bit by bit and moved to 16 internal relays starting with M50 designated by destination operand D1.

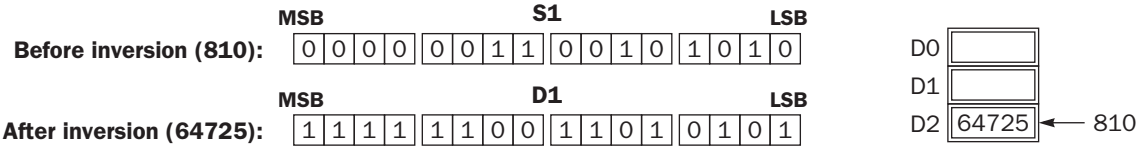
M10 through M17, M20 through M27 NOT ⟶ M50 through M57, M60 through M67

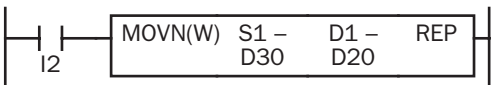


The ON/OFF statuses of the 16 internal relays M10 through M17 and M20 through M27 are inverted and moved to 16 internal relays M50 through M57 and M60 through M67. M50 is the LSB (least significant bit), and M67 is the MSB (most significant bit).

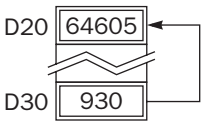


810 NOT → D2
When input I1 is on, decimal constant 810 designated by source operand S1 is converted into 16-bit binary data, and the ON/OFF statuses of the 16 bits are inverted and moved to data register D2 designated by destination operand D1.

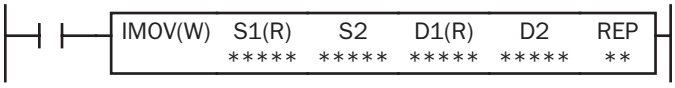




D30 NOT → D20
When input I2 is on, the data in data register D30 designated by S1 is inverted bit by bit and moved to data register D20 designated by D1.



IMOV (Indirect Move)



S1 + S2 → D1 + D2

When input is on, the values contained in operands designated by S1 and S2 are added to determine the source of data. The 16-bit data so determined is moved to destination, which is determined by the sum of values contained in operands designated by D1 and D2.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Base address to move from	X	X	X	X	X	X	X	—	1-99
S2 (Source 2)	Offset for S1	X	X	X	X	X	X	X	—	—
D1 (Destination 1)	Base address to move to	—	X	▲	X	X	X	X	—	1-99
D2 (Destination 2)	Offset for D1	X	X	X	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1, S2, or D2, the operand data is the timer/counter current value. When T (timer) or C (counter) is used as D1, the operand data is the timer/counter preset value which can be 0 through 65535.

Either source operand S2 or destination operand D2 does not have to be designated. If S2 or D2 is not designated, the source or destination operand is determined by S1 or D1 without offset.

Make sure that the source data determined by S1 + S2 and the destination data determined by D1 + D2 are within the valid operand range. If the derived source or destination operand is out of the valid operand range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

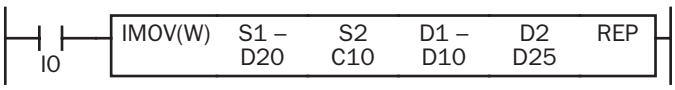
Valid Data Types

W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

Example: IMOV



D20 + C10 → D10 + D25

Source operand S1 and destination operand D1 determine the type of operand. Source operand S2 and destination operand D2 are the offset values to determine the source and destination operands.

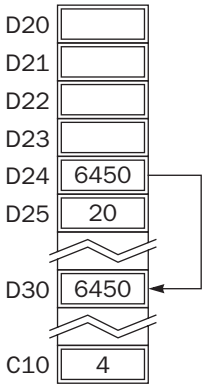
If the current value of counter C10 designated by source operand S2 is 4, the source data is determined by adding the offset to data register D20 designated by source operand S1:

D(20 + 4) = D24

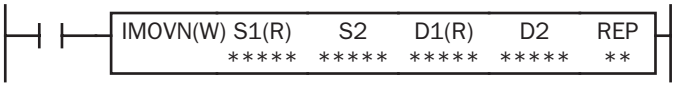
If data register D25 contains a value of 20, the destination is determined by adding the offset to data register D10 designated by destination operand D1:

D(10 + 20) = D30

As a result, when input I0 is on, the data in data register D24 is moved to data register D30.



IMOVN (Indirect Move Not)



S1 + S2 NOT → D1 + D2

When input is on, the values contained in operands designated by S1 and S2 are added to determine the source of data. The 16-bit data so determined is inverted and moved to destination, which is determined by the sum of values contained in operands designated by D1 and D2.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Base address to move from	X	X	X	X	X	X	X	—	1-99
S2 (Source 2)	Offset for S1	X	X	X	X	X	X	X	—	—
D1 (Destination 1)	Base address to move to	—	X	▲	X	X	X	X	—	1-99
D2 (Destination 2)	Offset for D1	X	X	X	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1, S2, or D2, the operand data is the timer/counter current value. When T (timer) or C (counter) is used as D1, the operand data is the timer/counter preset value which can be 0 through 65535.

Either source operand S2 or destination operand D2 does not have to be designated. If S2 or D2 is not designated, the source or destination operand is determined by S1 or D1 without offset.

Make sure that the source data determined by S1 + S2 and the destination data determined by D1 + D2 are within the valid operand range. If the derived source or destination operand is out of the valid operand range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

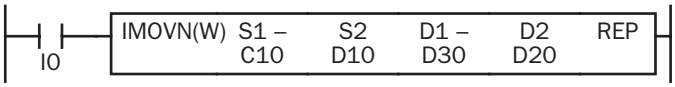
Valid Data Types

W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

Example: IMOVN



C10 + D10 NOT → D30 + D20

Source operand S1 and destination operand D1 determine the type of operand. Source operand S2 and destination operand D2 are the offset values to determine the source and destination operands.

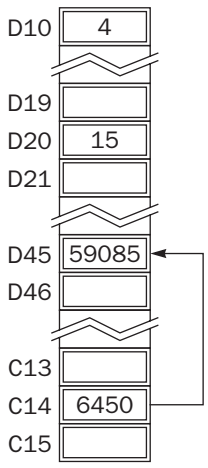
If the data of data register D10 designated by source operand S2 is 4, then the source data is determined by adding the offset to counter C10 designated by source operand S1:

C(10 + 4) = C14

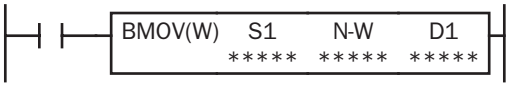
If data register D20 designated by destination operand D2 contains a value of 15, then the destination is determined by adding the offset to data register D30 designated by destination operand D1:

D(30 + 15) = D45

As a result, when input I0 is on, the current value of counter C14 is inverted and moved to data register D45.

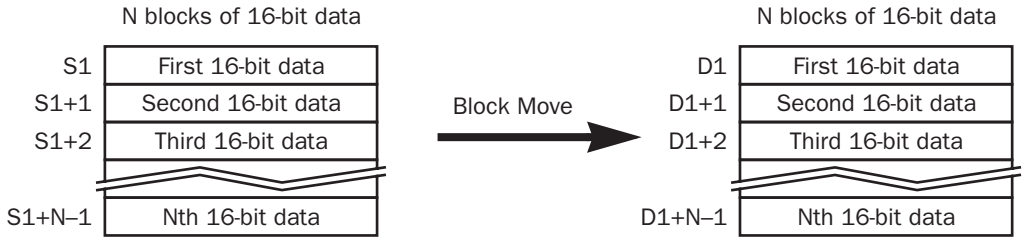


BMOV (Block Move)



S1, S1+1, S1+2, ... , S1+N–1 → D1, D1+1, D1+2, ... , D1+N–1

When input is on, N blocks of 16-bit word data starting with operand designated by S1 are moved to N blocks of destinations, starting with operand designated by D1. N-W specifies the quantity of blocks to move.



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	First operand number to move	X	X	X	X	X	X	X	—	—
N-W (N words)	Quantity of blocks to move	X	X	X	X	X	X	X	X	—
D1 (Destination 1)	First operand number to move to	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1 or N-W, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Make sure that the last source data determined by S1+N–1 and the last destination data determined by D1+N–1 are within the valid operand range. If the derived source or destination operand is out of the valid operand range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Valid Data Types

W (word)	I (integer)
X	—

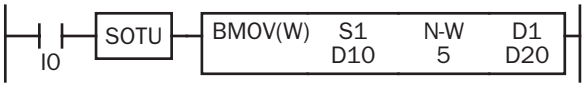
When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source, N-W, or destination, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source, N-W, or destination, 1 point is used.

Special Internal Relay M8024: BMOV/WSFT Executing Flag

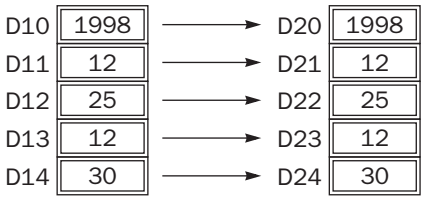
While the BMOV or WSFT is executed, M8024 turns on. When completed, M8024 turns off. If the CPU is powered down while executing BMOV or WSFT, M8024 remains on when the CPU is powered up again.

Example: BMOV

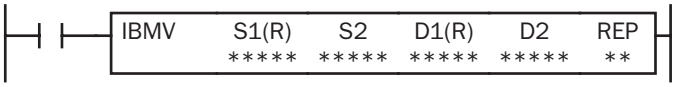


D10 through D14 → D20 through D24

When input I0 is turned on, data of 5 data registers starting with D10 designated by source operand S1 is moved to 5 data registers starting with D20 designated by destination operand D1.



IBMV (Indirect Bit Move)



S1 + S2 → D1 + D2

When input is on, the values contained in operands designated by S1 and S2 are added to determine the source of data. The 1-bit data so determined is moved to destination, which is determined by the sum of values contained in operands designated by D1 and D2.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Base address to move from	X	X	X	X	—	—	X	0 or 1	1-99
S2 (Source 2)	Offset for S1	X	X	X	X	X	X	X	0-65535	—
D1 (Destination 1)	Base address to move to	—	X	▲	X	—	—	X	—	1-99
D2 (Destination 2)	Offset for D1	X	X	X	X	X	X	X	0-65535	—

For the valid operand number range, see pages 6-1 and 6-2.

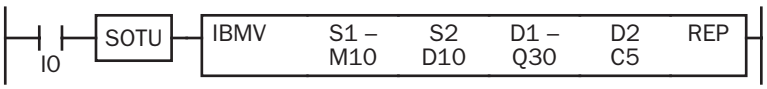
▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2 or D2, the timer/counter current value is read out.

Make sure that the last source data determined by S1+S2 and the last destination data determined by D1+D2 are within the valid operand range. If the derived source or destination operand is out of the valid operand range, a user program execution error will result, turning on special internal relay M8004 and ERR LED on the CPU module.

Either source operand S2 or destination operand D2 does not have to be designated. If S2 or D2 is not designated, the source or destination operand is determined by S1 or D1 without offset.

Examples: IBMV

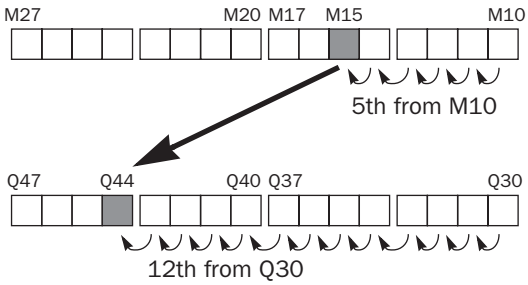


M10 + D10 → Q30 + C5

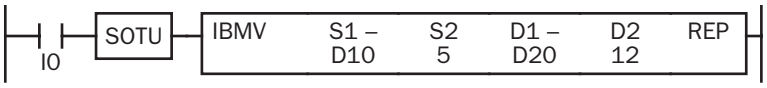
Source operand S1 and destination operand D1 determine the type of operand. Source operand S2 and destination operand D2 are the offset values to determine the source and destination operands.

If the value of data register D10 designated by source operand S2 is 5, the source data is determined by adding the offset to internal relay M10 designated by source operand S1.

If the current value of counter C5 designated by destination operand D2 is 12, the destination is determined by adding the offset to output Q30 designated by destination operand D1.



As a result, when input I0 is on, the ON/OFF status of internal relay M15 is moved to output Q44.

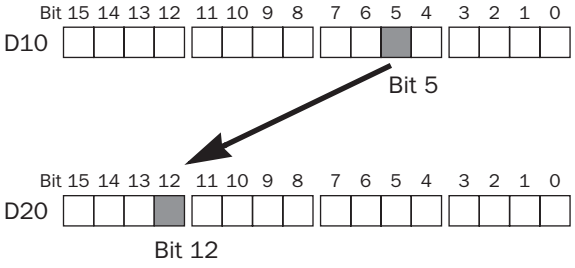


D10 + 5 → D20 + 12

Since source operand S1 is a data register and the value of source operand S2 is 5, the source data is bit 5 of data register D10 designated by source operand S1.

Since destination operand D1 is a data register and the value of source operand D2 is 12, the destination is bit 12 of data register D20 designated by destination operand D1.

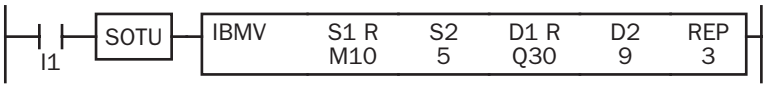
As a result, when input I0 is on, the ON/OFF status of data register D10 bit 5 is moved to data register D20 bit 12.



Repeat Operation in the Indirect Bit Move Instructions

Repeat Bit Operands (Source and Destination)

If a repeat operation is designated for bit operands such as input, output, internal relay, or shift register, bit operands as many as the repeat cycles are moved.

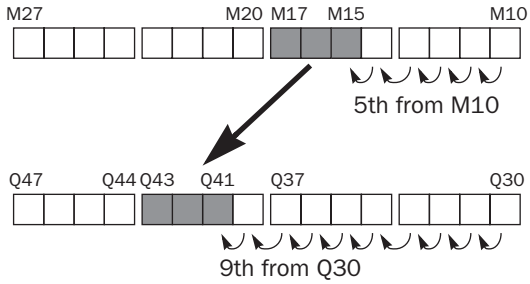


M10 + 5 → Q30 + 9
Repeat = 3

Since source operand S1 is internal relay M10 and the value of source operand S2 is 5, the source data is 3 internal relays starting with M15.

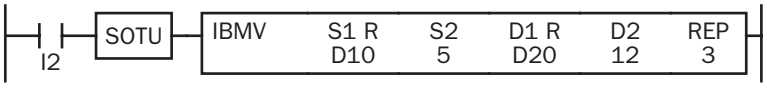
Since destination operand D1 is output Q30 and the value of destination operand D2 is 9, the destination is 3 outputs starting with Q41.

As a result, when input I1 is on, the ON/OFF statuses of internal relays M15 through M17 are moved to outputs Q41 through Q43.



Repeat Word Operands (Source and Destination)

If a repeat operation is designated for word operands such as data register, bit statuses as many as the repeat cycles in the designated data register are moved.

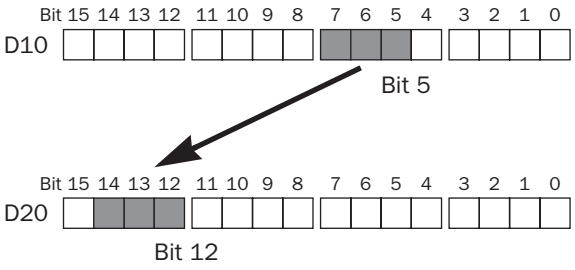


D10 + 5 → D20 + 12
Repeat = 3

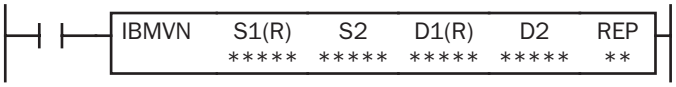
Since source operand S1 is data register D10 and the value of source operand S2 is 5, the source data is 3 bits starting with bit 5 of data register D10.

Since destination operand D1 is data register D20 and the value of destination operand D2 is 12, the destination is 3 bits starting with bit 12 of data register D20.

As a result, when input I2 is on, the ON/OFF statuses of data register D10 bits 5 through 7 are moved to data register D20 bits 12 through 14.



IBMVN (Indirect Bit Move Not)



S1 + S2 NOT → D1 + D2

When input is on, the values contained in operands designated by S1 and S2 are added to determine the source of data. The 1-bit data so determined is inverted and moved to destination, which is determined by the sum of values contained in operands designated by D1 and D2.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Base address to move from	X	X	X	X	—	—	X	0 or 1	1-99
S2 (Source 2)	Offset for S1	X	X	X	X	X	X	X	0-65535	—
D1 (Destination 1)	Base address to move to	—	X	▲	X	—	—	X	—	1-99
D2 (Destination 2)	Offset for D1	X	X	X	X	X	X	X	0-65535	—

For the valid operand number range, see pages 6-1 and 6-2.

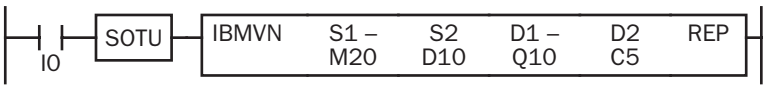
▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2 or D2, the timer/counter current value is read out.

Make sure that the last source data determined by S1+S2 and the last destination data determined by D1+D2 are within the valid operand range. If the derived source or destination operand is out of the valid operand range, a user program execution error will result, turning on special internal relay M8004 and ERR LED on the CPU module.

Either source operand S2 or destination operand D2 does not have to be designated. If S2 or D2 is not designated, the source or destination operand is determined by S1 or D1 without offset.

Examples: IBMVN

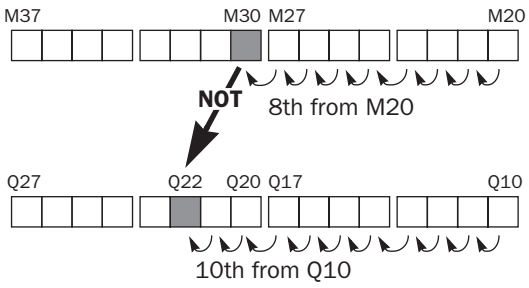


M20 + D10 NOT → Q10 + C5

Source operand S1 and destination operand D1 determine the type of operand. Source operand S2 and destination operand D2 are the offset values to determine the source and destination operands.

If the value of data register D10 designated by source operand S2 is 8, the source data is determined by adding the offset to internal relay M20 designated by source operand S1.

If the current value of counter C5 designated by destination operand D2 is 10, the destination is determined by adding the offset to output Q10 designated by destination operand D1.



As a result, when input I0 is on, the ON/OFF status of internal relay M30 is inverted and moved to output Q22.

10: DATA COMPARISON INSTRUCTIONS

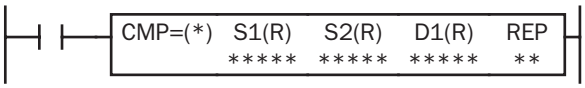
Introduction

Data can be compared using data comparison instructions, such as equal to, unequal to, less than, greater than, less than or equal to, and greater than or equal to. When the comparison result is true, an output or internal relay is turned on. The repeat operation can also be used to compare more than one set of data.

Three values can also be compared using the ICMP>= instruction.

Since the data comparison instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

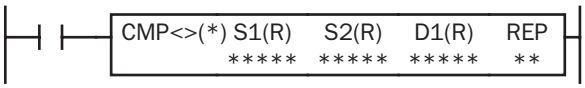
CMP= (Compare Equal To)



S1 = S2 → D1 on

When input is on, 16-bit data designated by source operands S1 and S2 are compared. When S1 data is equal to S2 data, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

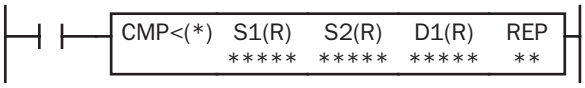
CMP<> (Compare Unequal To)



S1 ≠ S2 → D1 on

When input is on, 16-bit data designated by source operands S1 and S2 are compared. When S1 data is not equal to S2 data, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

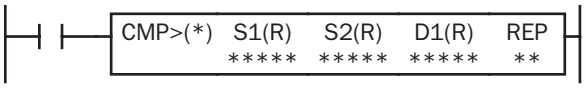
CMP< (Compare Less Than)



S1 < S2 → D1 on

When input is on, 16-bit data designated by source operands S1 and S2 are compared. When S1 data is less than S2 data, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

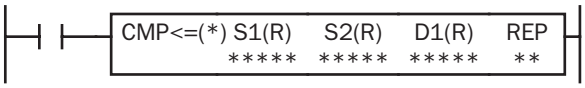
CMP> (Compare Greater Than)



S1 > S2 → D1 on

When input is on, 16-bit data designated by source operands S1 and S2 are compared. When S1 data is greater than S2 data, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

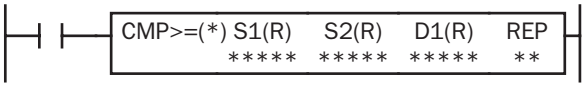
CMP<= (Compare Less Than or Equal To)



S1 ≤ S2 → D1 on

When input is on, 16-bit data designated by source operands S1 and S2 are compared. When S1 data is less than or equal to S2 data, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

CMP>= (Compare Greater Than or Equal To)



S1 ≥ S2 → D1 on

When input is on, 16-bit data designated by source operands S1 and S2 are compared. When S1 data is greater than or equal to S2 data, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data to compare	X	X	X	X	X	X	X	X	1-99
S2 (Source 2)	Data to compare	X	X	X	X	X	X	X	X	1-99
D1 (Destination 1)	Comparison output	—	X	▲	—	—	—	—	—	1-99

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1 or S2, the timer/counter current value is read out.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

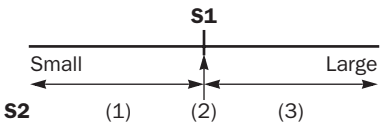
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

The destination uses only 1 point of output or internal relay. When repeat is designated for the destination, outputs or internal relays as many as the repeat cycles are used.

Special Internal Relays M8150, M8151, and M8152 in CMP=

Slim type CPU modules FC4A-D20RK1, -D20RS1, -D40K3, and -D40S3 have three special internal relays to indicate the comparison result of the CMP= instruction. Depending on the result, one of the three special internal relays turns on.

When S1 > S2, M8150 (greater than) turns on.
When S1 = S2, M8151 (equal to) turns on.
When S1 < S2, M8152 (less than) turns on.



S2 Value	M8150	M8151	M8152	D1 Status
(1) S1 > S2	**ON**	OFF	OFF	OFF
(2) S1 = S2	OFF	**ON**	OFF	**ON**
(3) S1 < S2	OFF	OFF	**ON**	OFF

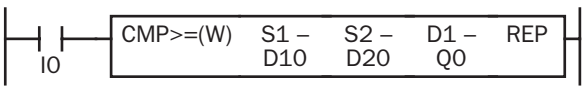
When repeat is designated, the comparison result of the last repeat cycle turns on one of the three special internal relays.

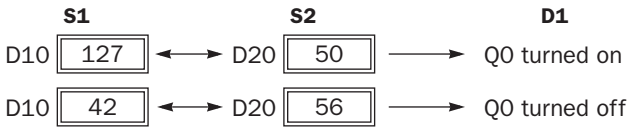
When more than one CMP= or ICMP>= instruction is used, M8150, M8151, or M8152 indicates the result of the instruction that was executed last.

Examples: CMP>=

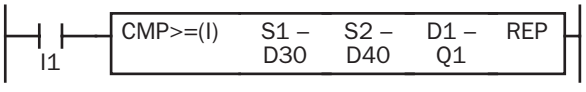
The following examples are described using the CMP≥ instruction. Data comparison operation for all other data comparison instructions is the same for the CMP≥ instruction.

- **Data Type: Word**





- **Data Type: Integer**

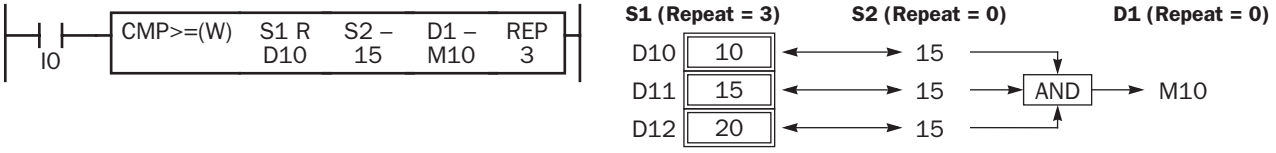


Repeat Operation in the Data Comparison Instructions

The following examples are described using the CMP≥ instruction of the word data type. Repeat operation for all other data comparison instructions and the integer data type is the same for the following examples.

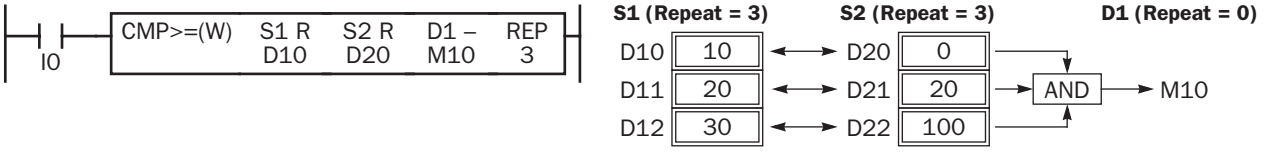
Repeat One Source Operand

When only S1 (source) is designated to repeat, source operands (as many as the repeat cycles, starting with the operand designated by S1) are compared with the operand designated by S2. The comparison results are ANDed and set to the destination operand designated by D1.



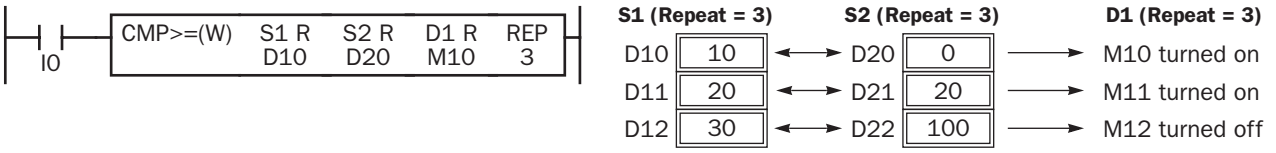
Repeat Two Source Operands

When S1 (source) and S2 (source) are designated to repeat, source operands (as many as the repeat cycles, starting with the operands designated by S1 and S2) are compared with each other. The comparison results are ANDed and set to the destination operand designated by D1.



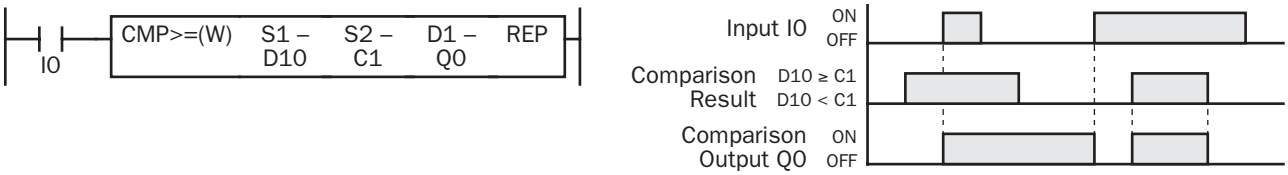
Repeat Source and Destination Operands

When S1, S2 (source), and D1 (destination) are designated to repeat, source operands (as many as the repeat cycles, starting with the operands designated by S1 and S2) are compared with each other. The comparison results are set to destination operands (as many as the repeat cycles, starting with the operand designated by D1).

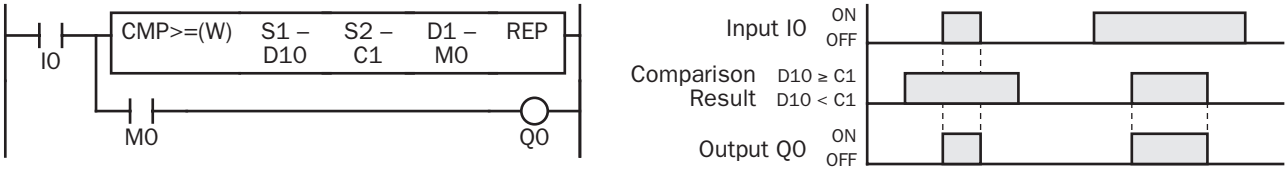


Comparison Output Status

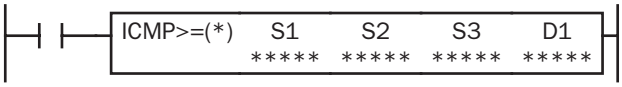
The comparison output is usually maintained while the input to the data comparison instruction is off. If the comparison output is on, the on status is maintained when the input is turned off as demonstrated by this program.



This program turns the output off when the input is off.



ICMP>= (Interval Compare Greater Than or Equal To)



S1 ≥ S2 ≥ S3 → D1 on

When input is on, the 16-bit data designated by S1, S2, and S3 are compared. When the condition is met, destination operand D1 is turned on. When the condition is not met, D1 is turned off.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data to compare	X	X	X	X	X	X	X	X	—
S2 (Source 2)	Data to compare	X	X	X	X	X	X	X	X	—
S3 (Source 3)	Data to compare	X	X	X	X	X	X	X	X	—
D1 (Destination 1)	Comparison output	—	X	▲	—	—	—	—	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1, S2, or S3, the timer/counter current value is read out.

When the data of S1 is smaller than that of S3 (S1 < S3), a user program execution error will result, turning on special internal relay M8004 and ERR LED on the CPU module.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source, 1 point is used.

The destination uses only one output or internal relay regardless of the selected data type.

Special Internal Relays M8150, M8151, and M8152 in ICMP>=

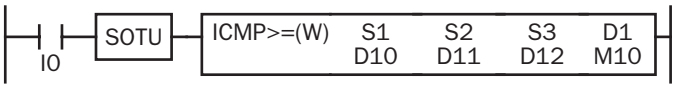
Three special internal relays are provided to indicate the comparison result of the ICMP>= instruction. Depending on the result, one of the three special internal relays turns on. S1 must always be greater than or equal to S3 (S1 ≥ S3).

When S2 > S1, M8150 turns on.
When S2 < S3, M8151 turns on.
When S1 > S2 > S3, M8152 turns on.

S2 Value	M8150	M8151	M8152	D1 Status
(1) S2 < S3	OFF	**ON**	OFF	OFF
(2) S2 = S3	OFF	OFF	OFF	**ON**
(3) S3 < S2 < S1	OFF	OFF	**ON**	**ON**
(4) S2 = S1	OFF	OFF	OFF	**ON**
(5) S2 > S1	**ON**	OFF	OFF	OFF

When more than one ICMP>= or CMP= instruction is used, M8150, M8151, or M8152 indicates the result of the instruction that was executed last.

Example: ICMP>=



D10 ≥ D11 ≥ D12 → M10 goes on

When input I0 is turned on, data of data registers D10, D11, and D12 designated by source operands S1, S2, and S3 are compared. When the condition is met, internal relay M10 designated by destination operand D1 is turned on. When the condition is not met, M10 is turned off.

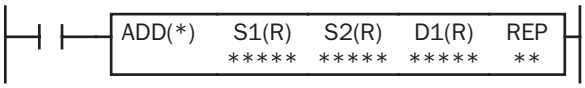
11: BINARY ARITHMETIC INSTRUCTIONS

Introduction

The binary arithmetic instructions make it possible for the user to program computations using addition, subtraction, multiplication, and division. For addition and subtraction operands, internal relay M8003 is used to carry or to borrow.

The ROOT instruction can be used to calculate the square root of the value stored in a data register.

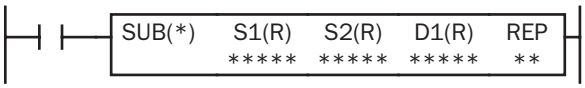
ADD (Addition)



S1 + S2 → D1, CY

When input is on, 16-bit data designated by source operands S1 and S2 are added. The result is set to destination operand D1 and carry (M8003).

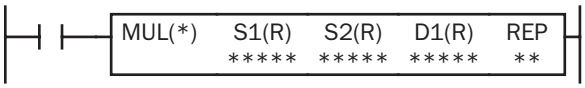
SUB (Subtraction)



S1 – S2 → D1, BW

When input is on, 16-bit data designated by source operand S2 is subtracted from 16-bit data designated by source operand S1. The result is set to destination operand D1 and borrow (M8003).

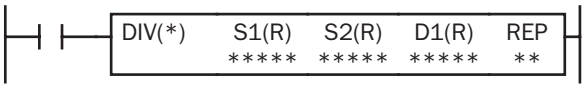
MUL (Multiplication)



S1 × S2 → D1·D1+1

When input is on, 16-bit data designated by source operand S1 is multiplied by 16-bit data designated by source operand S2. The result is set to 32-bit data designated by destination operand D1.

DIV (Division)



S1 ÷ S2 → D1 (quotient), D1+1 (remainder)

When input is on, 16-bit data designated by source operand S1 is divided by 16-bit data designated by source operand S2. The quotient is set to 16-bit destination operand D1, and the remainder is set to the next 16-bit data.

When S2 is 0 (dividing by 0), the ERR LED and special internal relay M8004 (user program execution error) are turned on.

A user program execution error also occurs in the following division operation.

Data type I: –32768 ÷ (–1)

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for calculation	X	X	X	X	X	X	X	X	1-99
S2 (Source 2)	Data for calculation	X	X	X	X	X	X	X	X	1-99
D1 (Destination 1)	Destination to store results	—	X	▲	X	X	X	X	—	1-99

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1 or S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Since the binary arithmetic instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

Using Carry or Borrow Signals

When the D1 (destination) data is out of the valid data range as a result of addition, a carry occurs, and special internal relay M8003 is turned on. When the D1 (destination) data is out of the valid data range as a result of subtraction, a borrow occurs, and special internal relay M8003 is turned on.

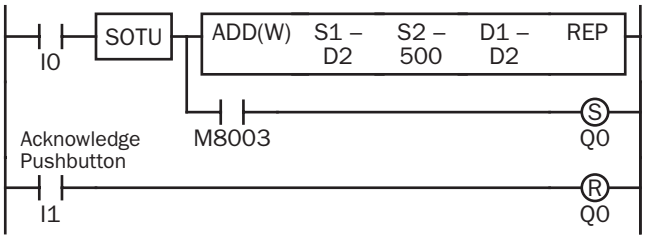
Data Type	Carry occurs when D1 is	Borrow occurs when D1 is
W (word)	over 65,535	below 0
I (integer)	below –32,768 or over 32,767	below –32,768 or over 32,767

There are three ways to program the carrying process (see examples below). If a carry never goes on, the program does not have to include internal relay M8003 to process carrying. If a carry goes on unexpectedly, an output can be programmed to be set as a warning indicator. If a carry goes on, the number of times a carry occurs can be added to be used as one word data in a specified register.

Examples: ADD

• Data Type: Word

This example demonstrates the use of a carry signal from special internal relay M8003 to set an alarm signal.



D2 + 500 → D2

When a carry occurs, output Q0 is set as a warning indicator.

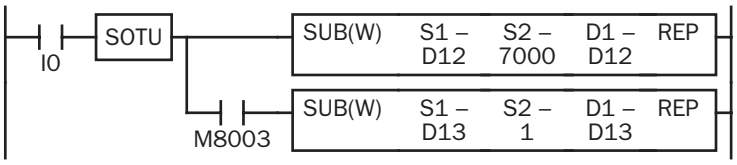
When the acknowledge pushbutton (input I1) is pressed, the warning indicator is reset.

• Data Type: Integer

Example: SUB

- **Data Type: Word**

The following example demonstrates the use of special internal relay M8003 to process a borrow.



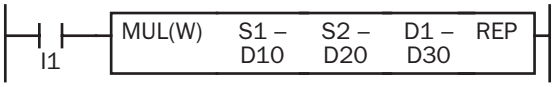
D12 – 7000 → D12

To process borrowing so that the number of times a borrow occurs is subtracted from D13.

When a borrow occurs, D13 is decremented by one.

Examples: MUL

- **Data Type: Word**

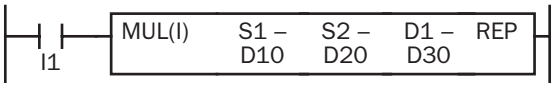


D10 500 (01F4h) × D20 600 (0258h) → D30·D31 300000 (000493E0h)

When input I1 is on, data of D10 is multiplied by data of D20, and the result is set to D30 and D31.

D30 4 (0004h)
D31 37856 (93E0h)

- **Data Type: Integer**



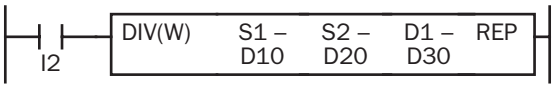
D10 –500 (FE0Ch) × D20 600 (0258h) → D30·D31 –300000 (FFFB6C20h)

D30 65531 (FFFBh)
D31 27680 (6C20h)

Note: Since the destination uses two word operands in the multiplication operation, data register D399 (10-I/O type CPU module) or D1299 (other CPU modules) cannot be used as destination operand D1. When using a bit operand such as internal relay for destination, 32 internal relays are required; so internal relay M281 (10-I/O type CPU module) or M1241 (other CPU modules) or a larger number cannot be used as destination operand D1.

Examples: DIV

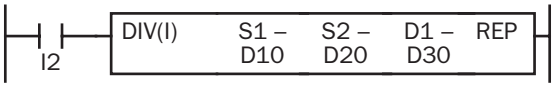
- **Data Type: Word**



D10 50 ÷ D20 7 → D30 7 (Quotient) D31 1 (Remainder)

When input I2 is on, data of D10 is divided by data of D20. The quotient is set to D30, and the remainder is set to D31.

- **Data Type: Integer**



D10 50 ÷ D20 –7 → D30 –7 (Quotient) D31 1 (Remainder)

Note: Since the destination uses two word operands in the division operation, data register D399 (10-I/O type CPU module) or D1299 (other CPU modules) cannot be used as destination operand D1. When using a bit operand such as internal relay for destination, 32 internal relays are required; so M281 (10-I/O type CPU module) or M1241 (other CPU modules) or a larger number cannot be used as destination operand D1.

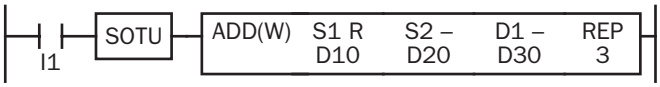
Repeat Operation in the ADD and SUB Instructions

Source operands S1 and S2 and destination operand D1 can be designated to repeat individually or in combination. When destination operand D1 is not designated to repeat, the final result is set to destination operand D1. When repeat is designated, consecutive operands as many as the repeat cycles starting with the designated operand are used.

Since the repeat operation works similarly on the ADD (addition) and SUB (subtraction) instructions of the word and integer data types, the following examples are described using the ADD instruction of the word data type.

Repeat One Source Operand

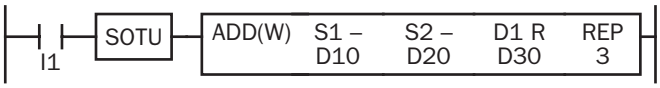
When only S1 (source) is designated to repeat, the final result is set to destination operand D1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 0)
D10 10	+	D20 25	→	D30 (35)
D11 15	+	D20 25	→	D30 (40)
D12 20	+	D20 25	→	D30 45

Repeat Destination Operand Only

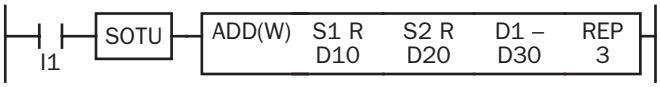
When only D1 (destination) is designated to repeat, the same result is set to 3 operands starting with D1.



S1 (Repeat = 0)		S2 (Repeat = 0)		D1 (Repeat = 3)
D10 10	+	D20 25	→	D30 35
D10 10	+	D20 25	→	D31 35
D10 10	+	D20 25	→	D32 35

Repeat Two Source Operands

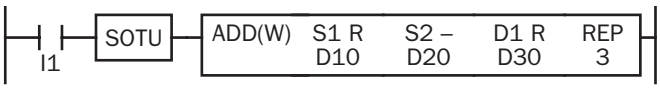
When S1 and S2 (source) are designated to repeat, the final result is set to destination operand D1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 0)
D10 10	+	D20 25	→	D30 (35)
D11 15	+	D21 35	→	D30 (50)
D12 20	+	D22 45	→	D30 65

Repeat Source and Destination Operands

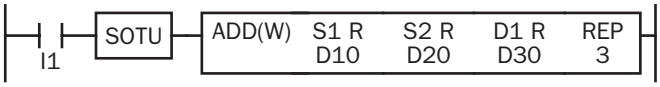
When S1 (source) and D1 (destination) are designated to repeat, different results are set to 3 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 3)
D10 10	+	D20 25	→	D30 35
D11 15	+	D20 25	→	D31 40
D12 20	+	D20 25	→	D32 45

Repeat All Source and Destination Operands

When all operands are designated to repeat, different results are set to 3 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 3)
D10 10	+	D20 25	→	D30 35
D11 15	+	D21 35	→	D31 50
D12 20	+	D22 45	→	D32 65

Note: Special internal relay M8003 (carry/borrow) is turned on when a carry or borrow occurs in the last repeat operation. When a user program execution error occurs in any repeat operation, special internal relay M8004 (user program execution error) and the ERR LED are turned on and maintained while operation for other instructions is continued.

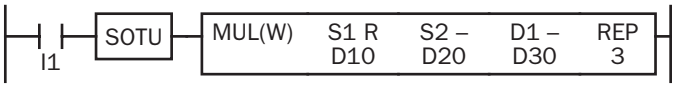
Repeat Operation in the MUL Instruction

Since the MUL (multiplication) instruction uses two destination operands, the result is stored to destination operands as described below. Source operands S1 and S2 and destination operand D1 can be designated to repeat individually or in combination. When destination operand D1 is not designated to repeat, the final result is set to destination operand D1 and D1+1. When repeat is designated, consecutive operands as many as the repeat cycles starting with the designated operand are used.

Since the repeat operation works similarly on the word and integer data types, the following examples are described using the word data type.

Repeat One Source Operand

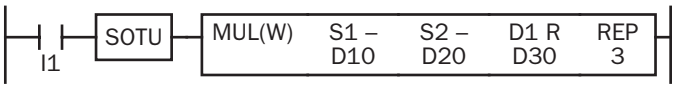
When only S1 (source) is designated to repeat, the final result is set to destination operands D1 and D1+1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 0)	
D10	×	D20	→	(D30)	(D31)
D11	×	D20	→	(D30)	(D31)
D12	×	D20	→	D30	D31

Repeat Destination Operand Only

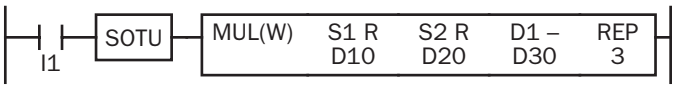
When only D1 (destination) is designated to repeat, the same result is set to 6 operands starting with D1.



S1 (Repeat = 0)		S2 (Repeat = 0)		D1 (Repeat = 3)	
D10	×	D20	→	D30	D31
D10	×	D20	→	D32	D33
D10	×	D20	→	D34	D35

Repeat Two Source Operands

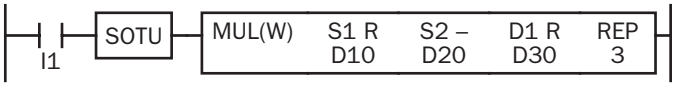
When S1 and S2 (source) are designated to repeat, the final result is set to destination operands D1 and D1+1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 0)	
D10	×	D20	→	(D30)	(D31)
D11	×	D21	→	(D30)	(D31)
D12	×	D22	→	D30	D31

Repeat Source and Destination Operands

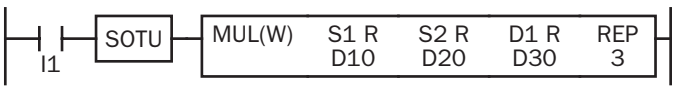
When S1 (source) and D1 (destination) are designated to repeat, different results are set to 6 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 3)	
D10	×	D20	→	D30	D31
D11	×	D20	→	D32	D33
D12	×	D20	→	D34	D35

Repeat All Source and Destination Operands

When all operands are designated to repeat, different results are set to 6 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 3)	
D10	×	D20	→	D30	D31
D11	×	D21	→	D32	D33
D12	×	D22	→	D34	D35

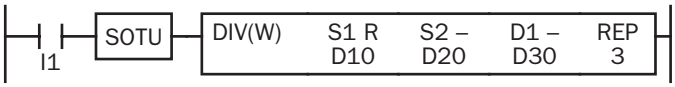
Repeat Operation in the DIV Instruction

Since the DIV (division) instruction uses two destination operands, the quotient and remainder are stored as described below. Source operands S1 and S2 and destination operand D1 can be designated to repeat individually or in combination. When destination operand D1 is not designated to repeat, the final result is set to destination operand D1 (quotient) and D1+1 (remainder). When repeat is designated, consecutive operands as many as the repeat cycles starting with the designated operand are used.

Since the repeat operation works similarly on the word and integer data types, the following examples are described using the word data type.

Repeat One Source Operand

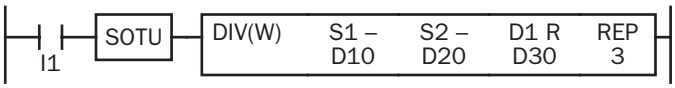
When only S1 (source) is designated to repeat, the final result is set to destination operands D1 and D1+1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 0)	
D10	÷	D20	→	(D30)	(D31)
D11	÷	D20	→	(D30)	(D31)
D12	÷	D20	→	D30	D31
				Quotient	Remainder

Repeat Destination Operand Only

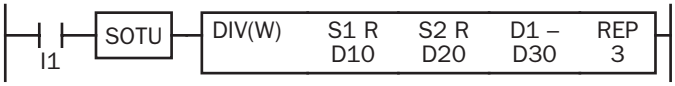
When only D1 (destination) is designated to repeat, the same result is set to 6 operands starting with D1.



S1 (Repeat = 0)		S2 (Repeat = 0)		D1 (Repeat = 3)	
D10	÷	D20	→	D30	D33
D10	÷	D20	→	D31	D34
D10	÷	D20	→	D32	D35
				Quotient	Remainder

Repeat Two Source Operands

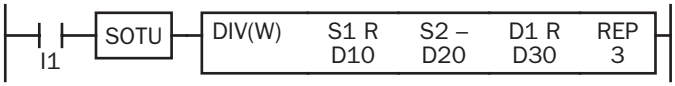
When S1 and S2 (source) are designated to repeat, the final result is set to destination operands D1 and D1+1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 0)	
D10	÷	D20	→	(D30)	(D31)
D11	÷	D21	→	(D30)	(D31)
D12	÷	D22	→	D30	D31
				Quotient	Remainder

Repeat Source and Destination Operands

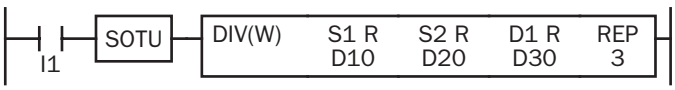
When S1 (source) and D1 (destination) are designated to repeat, different results are set to 6 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 3)	
D10	÷	D20	→	D30	D33
D11	÷	D20	→	D31	D34
D12	÷	D20	→	D32	D35
				Quotient	Remainder

Repeat All Source and Destination Operands

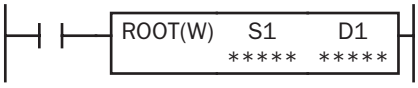
When all operands are designated to repeat, different results are set to 6 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 3)	
D10	÷	D20	→	D30	D33
D11	÷	D21	→	D31	D34
D12	÷	D22	→	D32	D35
				Quotient	Remainder

Note: When a user program execution error occurs in any repeat operation, special internal relay M8004 (user program execution error) and the ERR LED are turned on and maintained while operation for other instructions is continued.

ROOT (Root)



$\sqrt{S1} \rightarrow D1$

When input is on, the square root of operand designated by S1 is extracted and is stored to the destination designated by D1.

Valid values are 0 to 65535. The square root is calculated to two decimals, omitting the figures below the second place of decimals.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Binary data	—	—	—	—	—	—	X	X	—
D1 (Destination 1)	Destination to store results	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

Since the ROOT instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

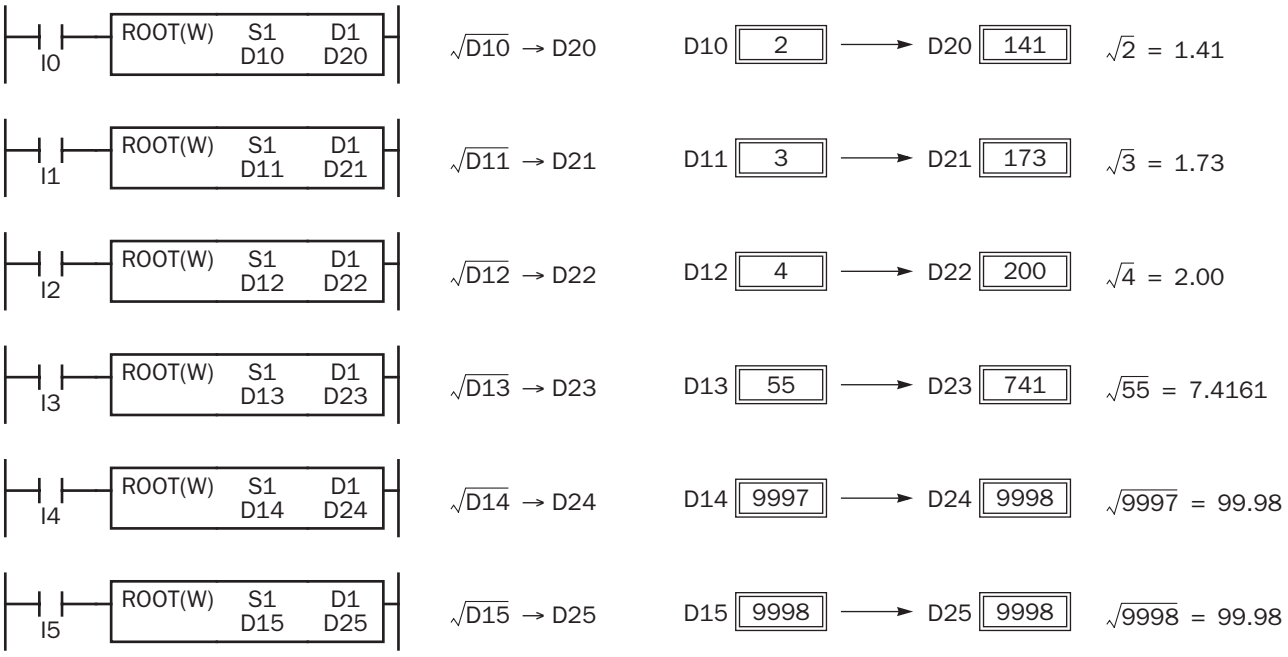
The ROOT instruction cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Valid Data Types

W (word)	I (integer)
X	—

When a word operand such as D (data register) is designated as the source or destination, 1 point (word data type) is used.

Examples: ROOT

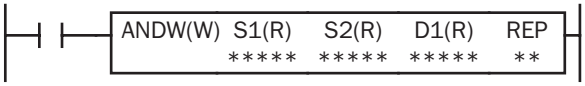


12: BOOLEAN COMPUTATION INSTRUCTIONS

Introduction

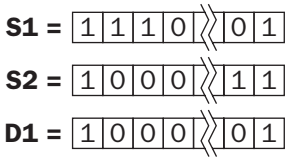
Boolean computations use the AND, OR, and exclusive OR statements as carried out by the ANDW, ORW, and XORW instructions in the word data type, respectively.

ANDW (AND Word)



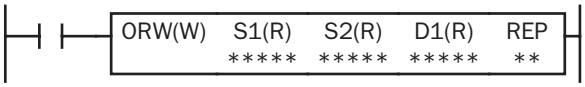
S1 • S2 → D1

When input is on, 16-bit data designated by source operands S1 and S2 are ANDed, bit by bit. The result is set to destination operand D1.



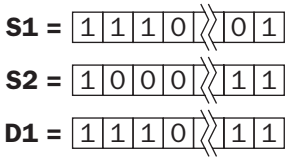
S1	S2	D1
0	0	0
0	1	0
1	0	0
1	1	1

ORW (OR Word)



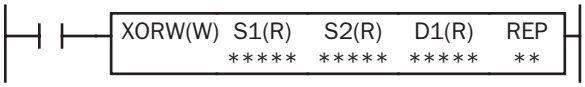
S1 + S2 → D1

When input is on, 16-bit data designated by source operands S1 and S2 are ORed, bit by bit. The result is set to destination operand D1.



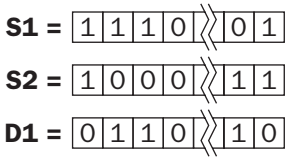
S1	S2	D1
0	0	0
0	1	1
1	0	1
1	1	1

XORW (Exclusive OR Word)



S1 ⊕ S2 → D1

When input is on, 16-bit data designated by source operands S1 and S2 are exclusive ORed, bit by bit. The result is set to destination operand D1.



S1	S2	D1
0	0	0
0	1	1
1	0	1
1	1	0

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for computation	X	X	X	X	X	X	X	X	1-99
S2 (Source 2)	Data for computation	X	X	X	X	X	X	X	X	1-99
D1 (Destination 1)	Destination to store results	—	X	▲	X	X	X	X	—	1-99

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1 or S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Since the Boolean computation instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Valid Data Types

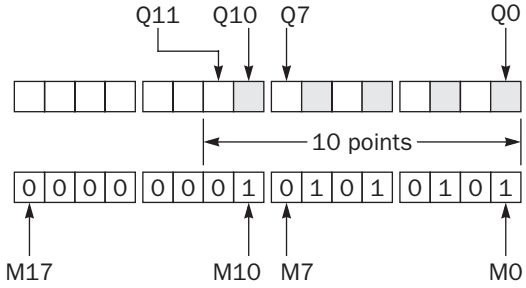
W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used. When repeat is designated for a bit operand, the quantity of operand bits increases in 16-point increments.

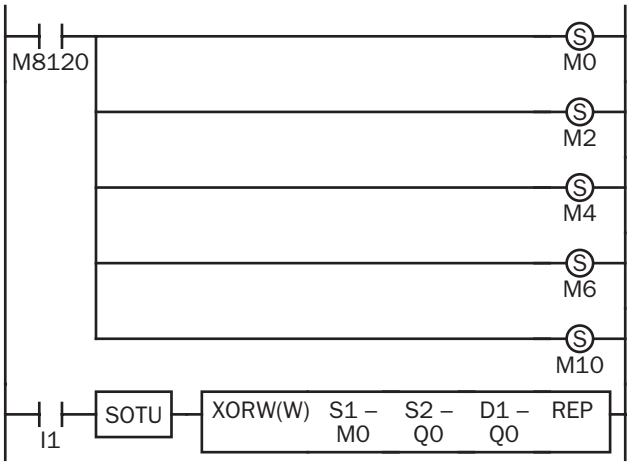
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used. When repeat is designated for a word operand, the quantity of operand words increases in 1-point increments.

Example: XORW

To convert optional output status among a series of 10 output points, use the XORW instruction in combination with 10 internal relay points.



This program will invert the status of the shaded outputs at the left from on to off, and those not shaded from off to on.



Ten outputs Q0 through Q11 are assigned to 10 internal relays M0 through M11.

Five internal relays M0, M2, M4, M6, and M10 are set by initialize pulse special internal relay M8120.

When input I1 is turned on, the XORW instruction is executed to invert the status of outputs Q0, Q2, Q4, Q6, and Q10.

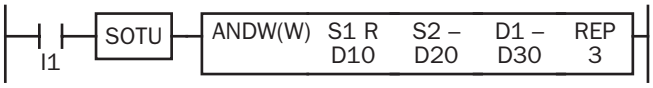
Repeat Operation in the ANDW, ORW, and XORW Instructions

Source operands S1 and S2 and destination operand D1 can be designated to repeat individually or in combination. When destination operand D1 is not designated to repeat, the final result is set to destination operand D1. When repeat is designated, consecutive operands as many as the repeat cycles starting with the designated operand are used.

Since the repeat operation works similarly on the ANDW (AND word), ORW (OR word), and XORW (exclusive OR word) instructions of the word and integer data types, the following examples are described using the ANDW instruction of the word data type.

Repeat One Source Operand

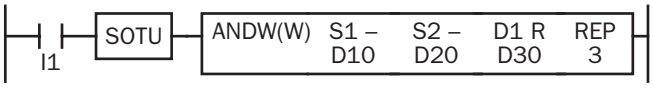
When only S1 (source) is designated to repeat, the final result is set to destination operand D1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 0)
D10	·	D20	→	(D30)
D11	·	D20	→	(D30)
D12	·	D20	→	D30

Repeat Destination Operand Only

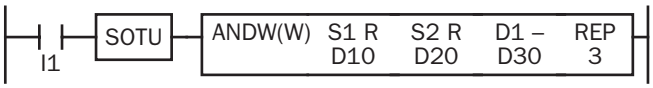
When only D1 (destination) is designated to repeat, the same result is set to 3 operands starting with D1.



S1 (Repeat = 0)		S2 (Repeat = 0)		D1 (Repeat = 3)
D10	·	D20	→	D30
D10	·	D20	→	D31
D10	·	D20	→	D32

Repeat Two Source Operands

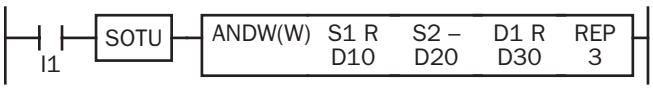
When S1 and S2 (source) are designated to repeat, the final result is set to destination operand D1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 0)
D10	·	D20	→	(D30)
D11	·	D21	→	(D30)
D12	·	D22	→	D30

Repeat Source and Destination Operands

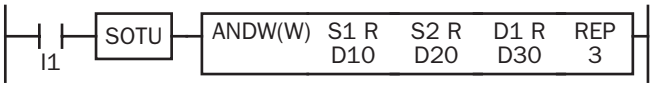
When S1 (source) and D1 (destination) are designated to repeat, different results are set to 3 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 0)		D1 (Repeat = 3)
D10	·	D20	→	D30
D11	·	D20	→	D31
D12	·	D20	→	D32

Repeat All Source and Destination Operands

When all operands are designated to repeat, different results are set to 3 operands starting with D1.



S1 (Repeat = 3)		S2 (Repeat = 3)		D1 (Repeat = 3)
D10	·	D20	→	D30
D11	·	D21	→	D31
D12	·	D22	→	D32

13: SHIFT / ROTATE INSTRUCTIONS

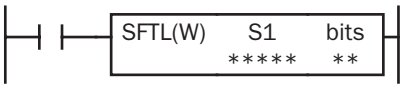
Introduction

Bit shift and rotate instructions are used to shift the 16-bit data in the designated source operand S1 to the left or right by the quantity of bits designated. The result is set to the source operand S1 and a carry (special internal relay M8003).

The BCD left shift instruction shifts the BCD digits in two consecutive data registers to the left.

The word shift instruction is used to move 16-bit data to a destination data register and shifts down the data of subsequent data registers as many as designated.

SFTL (Shift Left)

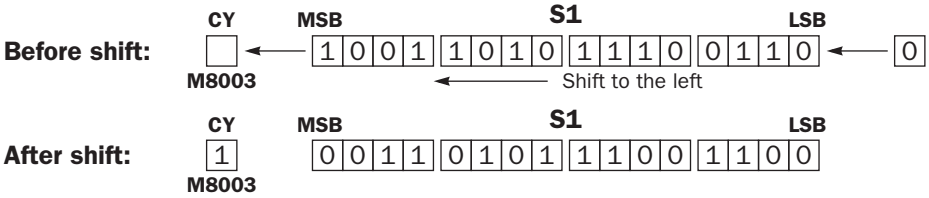


CY ← S1

When input is on, 16-bit data of the designated source operand S1 is shifted to the left by the quantity of bits designated by operand bits.

The result is set to the source operand S1, and the last bit status shifted out is set to a carry (special internal relay M8003). Zeros are set to the LSB.

When bits to shift = 1



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for bit shift	—	X	▲	X	—	—	X	—	—
bits	Quantity of bits to shift	—	—	—	—	—	—	—	1-15	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as S1. Special internal relays cannot be designated as S1.

The quantity of bits to shift can be 1 through 15.

Since the SFTL instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

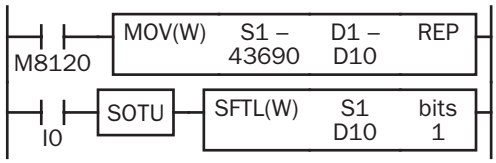
Valid Data Types

W (word)	I (integer)
X	—

When a bit operand such as Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

When a word operand such as D (data register) is designated as the source, 1 point is used.

Example: SFTL

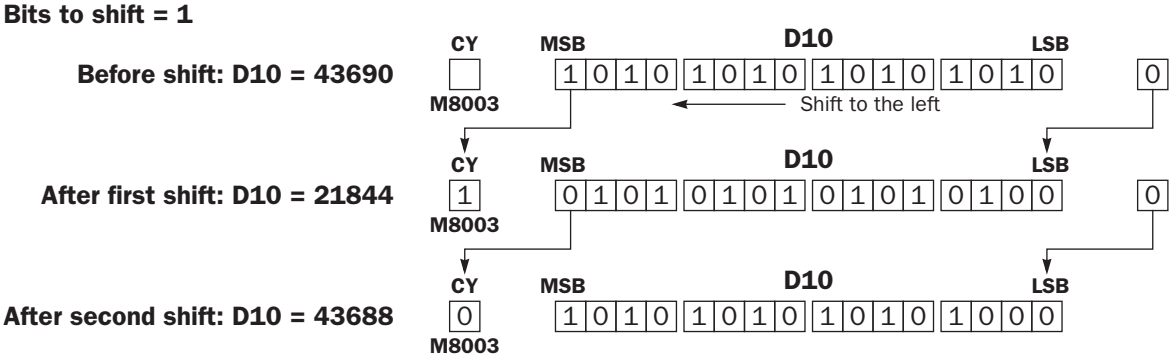


M8120 is the initialize pulse special internal relay.

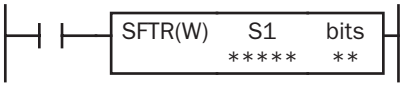
When the CPU starts operation, the MOV (move) instruction sets 43690 to data register D10.

Each time input I0 is turned on, 16-bit data of data register D10 is shifted to the left by 1 bit as designated by operand bits. The last bit status shifted out is set to a carry (special internal relay M8003). Zeros are set to the LSB.

Bits to shift = 1



SFTR (Shift Right)

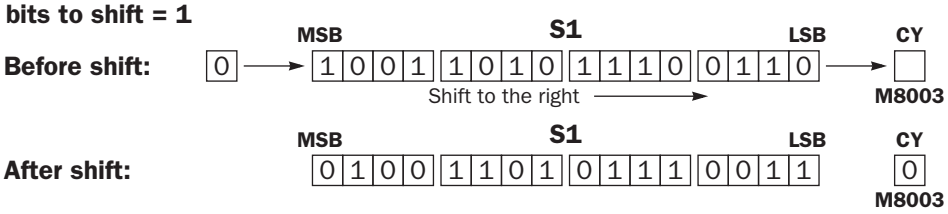


S1 → CY

When input is on, 16-bit data of the designated source operand S1 is shifted to the right by the quantity of bits designated by operand bits.

The result is set to the source operand S1, and the last bit status shifted out is set to a carry (special internal relay M8003). Zeros are set to the MSB.

When bits to shift = 1



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for bit shift	—	X	▲	X	—	—	X	—	—
bits	Quantity of bits to shift	—	—	—	—	—	—	—	1-15	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as S1. Special internal relays cannot be designated as S1.

The quantity of bits to shift can be 1 through 15.

Since the SFTR instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

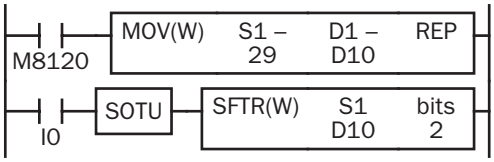
Valid Data Types

W (word)	I (integer)
X	—

When a bit operand such as Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

When a word operand such as D (data register) is designated as the source, 1 point is used.

Example: SFTR

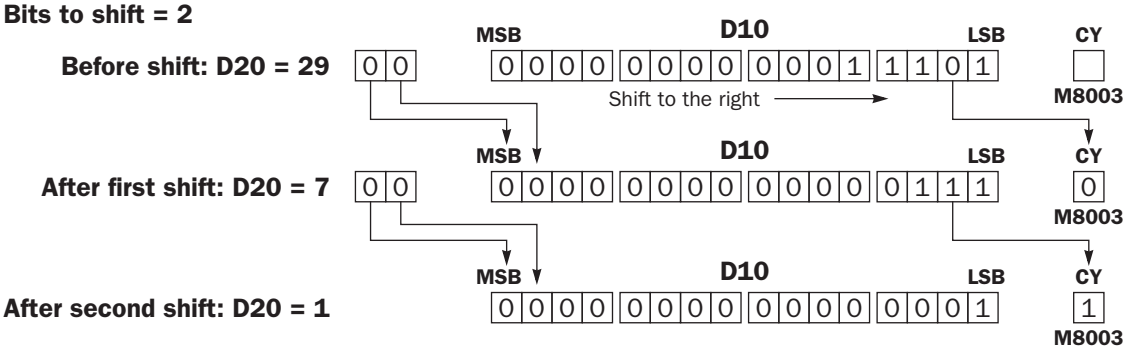


M8120 is the initialize pulse special internal relay.

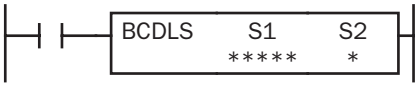
When the CPU starts operation, the MOV (move) instruction sets 29 to data register D10.

Each time input I0 is turned on, 16-bit data of data register D10 is shifted to the right by 2 bits as designated by operand bits. The last bit status shifted out is set to a carry (special internal relay M8003). Zeros are set to the MSB.

Bits to shift = 2



BCDLS (BCD Left Shift)



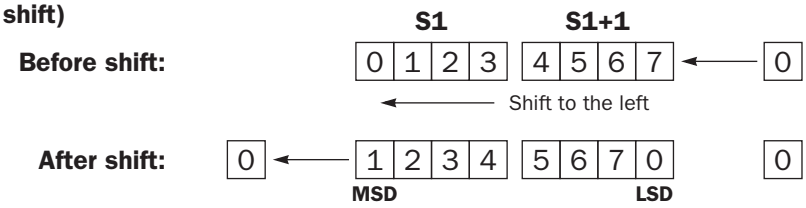
When input is on, the 32-bit binary data designated by S1 is converted into 8 BCD digits, shifted to the left by the quantity of digits designated by S2, and converted back to 32-bit binary data.

Valid values for each of S1 and S1+1 are 0 through 9999.

The quantity of digits to shift can be 1 through 7.

Zeros are set to the lowest digits as many as the digits shifted.

When S2 = 1 (digits to shift)



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for BCD shift	—	—	—	—	—	—	X	—	—
S2 (Source 2)	Quantity of digits to shift	X	X	X	X	X	X	X	1-7	—

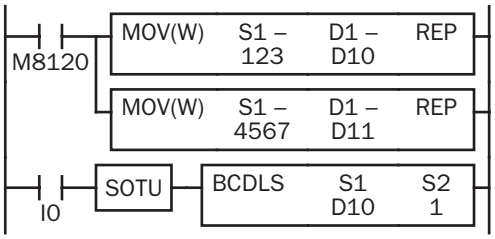
For the valid operand number range, see pages 6-1 and 6-2.

When T (timer) or C (counter) is used as S2, the timer/counter current value is read out.

The quantity of digits to shift designated as S2 can be 1 through 7.

Make sure that the source data determined by S1 and S1+1 is between 0 and 9999 for each data register. If either source data is over 9999, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module. When S2 is over 7, a user program execution error will also result.

Example: BCDLS



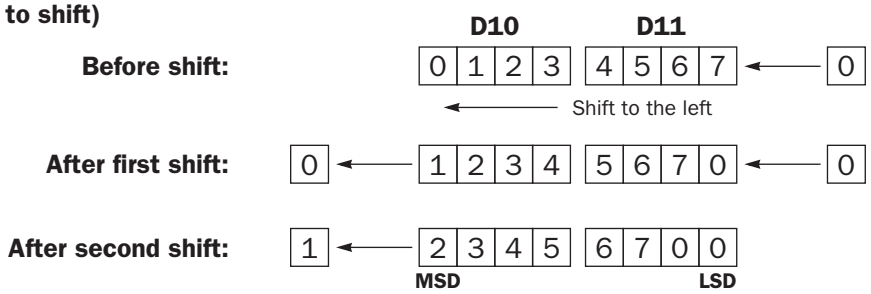
M8120 is the initialize pulse special internal relay.

When the CPU starts operation, the MOV (move) instructions set 123 and 4567 to data registers D10 and D11, respectively.

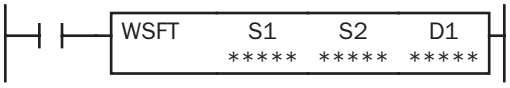
Each time input I0 is turned on, the 32-bit binary data of data registers D10 and D11 designated by S1 is converted into 8 BCD digits, shifted to the left by 1 digit as designated by operand S2, and converted back to 32-bit binary data.

Zeros are set to the lowest digits as many as the digits shifted.

When S2 = 1 (digits to shift)

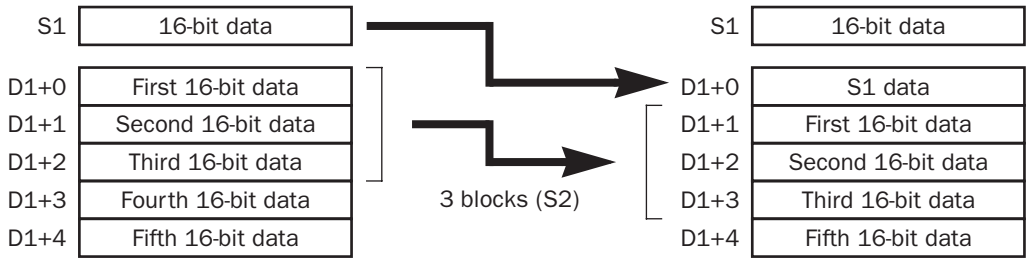


WSFT (Word Shift)



When input is on, N blocks of 16-bit word data starting with operand designated by D1 are shifted up to the next 16-bit positions. At the same time, the data designated by operand S1 is moved to operand designated by D1. S2 specifies the quantity of blocks to move.

When S2 = 3 (quantity of blocks to shift)



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Source data for word shift	X	X	X	X	X	X	X	X	—
S2 (Source 2)	Quantity of blocks to shift	X	X	X	X	X	X	X	X	—
D1 (Destination 1)	First operand number to shift	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

When T (timer) or C (counter) is used as S1 or S2, the timer/counter current value is read out.

Valid Data Types

W (word)	I (integer)
X	—

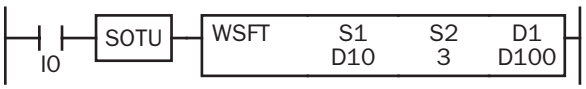
When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as source S1 or S2, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as source S1 or S2, 1 point is used.

Special Internal Relay M8024: BMOV/WSFT Executing Flag

While the BMOV or WSFT is executed, M8024 turns on. When completed, M8024 turns off. If the CPU is powered down while executing BMOV or WSFT, M8024 remains on when the CPU is powered up again.

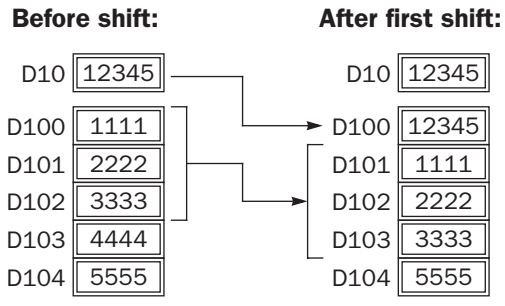
Example: WSFT



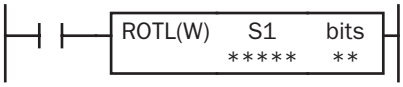
D100 through D102 → D101 through D103

D10 → D100

When input I0 is turned on, data of 3 data registers starting with D100 designated by destination operand D1 is shifted to the next data registers. Data of data register D10 designated by source operand S1 is moved to D100 designated by destination operand D1.



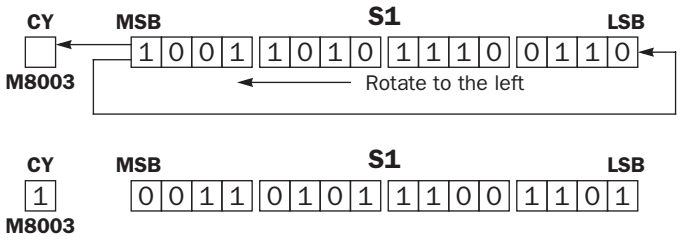
ROTL (Rotate Left)



When input is on, 16-bit data of the designated source operand S1 is rotated to the left by the quantity of bits designated by operand bits.

The result is set to the source operand S1, and the last bit status rotated out is set to a carry (special internal relay M8003).

When bits to rotate = 1



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for bit rotation	—	X	▲	X	—	—	X	—	—
bits	Quantity of bits to rotate	—	—	—	—	—	—	—	1-15	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as S1. Special internal relays cannot be designated as S1.

The quantity of bits to rotate can be 1 through 15.

Since the ROTL instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

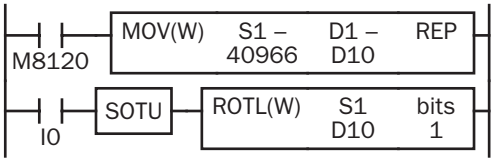
Valid Data Types

W (word)	I (integer)
X	—

When a bit operand such as Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

When a word operand such as D (data register) is designated as the source, 1 point is used.

Example: ROTL



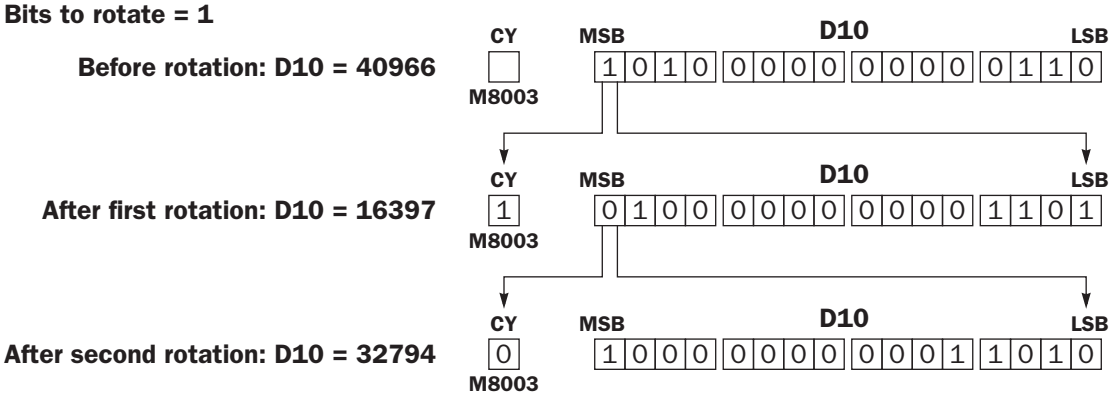
M8120 is the initialize pulse special internal relay.

When the CPU starts operation, the MOV (move) instruction sets 40966 to data register D10.

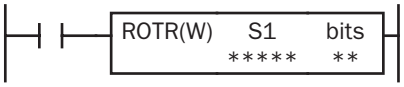
Each time input I0 is turned on, 16-bit data of data register D10 is rotated to the left by 1 bit as designated by operand bits.

The status of the MSB is set to a carry (special internal relay M8003).

Bits to rotate = 1



ROTR (Rotate Right)

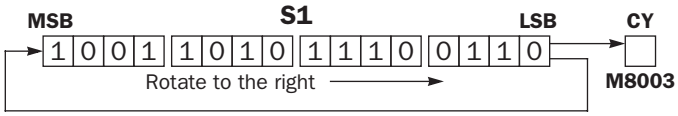


When input is on, 16-bit data of the designated source operand S1 is rotated to the right by the quantity of bits designated by operand bits.

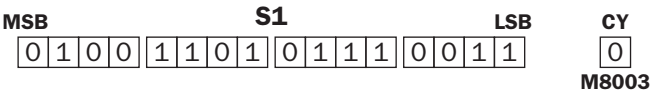
The result is set to the source operand S1, and the last bit status rotated out is set to a carry (special internal relay M8003).

When bits to rotate = 1

Before rotation:



After rotation:



Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data for bit rotation	—	X	▲	X	—	—	X	—	—
bits	Quantity of bits to rotate	—	—	—	—	—	—	—	1-15	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as S1. Special internal relays cannot be designated as S1.

The quantity of bits to rotate can be 1 through 15.

Since the ROTR instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

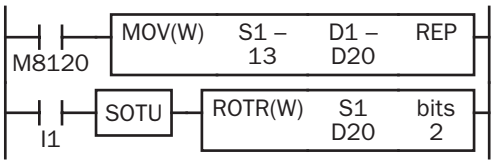
Valid Data Types

W (word)	I (integer)
X	—

When a bit operand such as Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

When a word operand such as D (data register) is designated as the source, 1 point is used.

Example: ROTR



M8120 is the initialize pulse special internal relay.

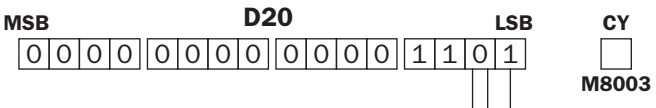
When the CPU starts operation, the MOV (move) instruction sets 13 to data register D20.

Each time input I1 is turned on, 16-bit data of data register D20 is rotated to the right by 2 bits as designated by operand bits.

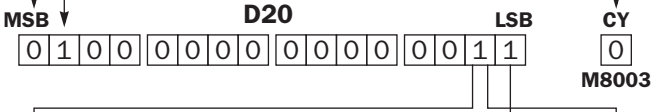
The last bit status rotated out is set to a carry (special internal relay M8003).

Bits to rotate = 2

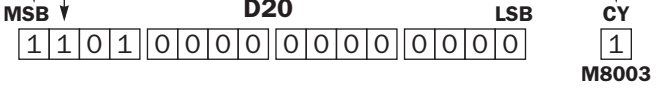
Before rotation: D20 = 13



After first rotation: D20 = 16387



After second rotation: D20 = 53248



14: DATA CONVERSION INSTRUCTIONS

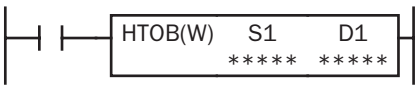
Introduction

Data conversion instructions convert data format among binary, BCD, and ASCII.

The ENCO (encode), DECO (decode), and BCNT (bit count) instructions processes bit operand data.

The ALT (alternate output) instruction turns on and off an output each time an input button is pressed.

HTOB (Hex to BCD)



S1 → D1

When input is on, the 16-bit data designated by S1 is converted into BCD and stored to the destination designated by operand D1.

Valid values for the source operand are 0 through 9999.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Binary data to convert	X	X	X	X	X	X	X	X	—
D1 (Destination 1)	Destination to store conversion results	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Valid values for the source operand are 0 through 9999 (270Fh). Make sure that the source designated by S1 is within the valid value range. If the source data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Since the HTOB instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

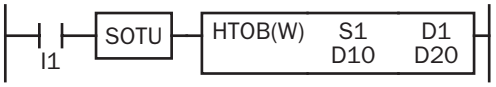
Valid Data Types

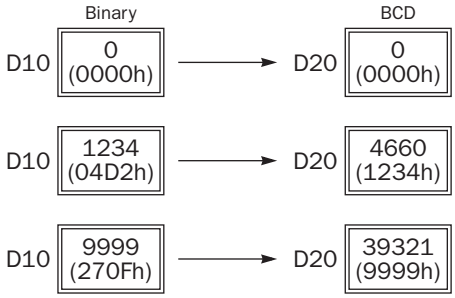
W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used.

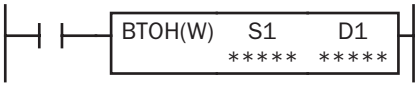
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used.

Example: HTOB





BTOH (BCD to Hex)



S1 → D1

When input is on, the BCD data designated by S1 is converted into 16-bit binary data and stored to the destination designated by operand D1.

Valid values for the source operand are 0 through 9999 (BCD).

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	BCD data to convert	X	X	X	X	X	X	X	X	—
D1 (Destination 1)	Destination to store conversion results	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S1, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Valid values for the source operand are 0 through 9999 (BCD). Make sure that each digit of the source designated by S1 is 0 through 9. If the source data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Since the BTOH instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

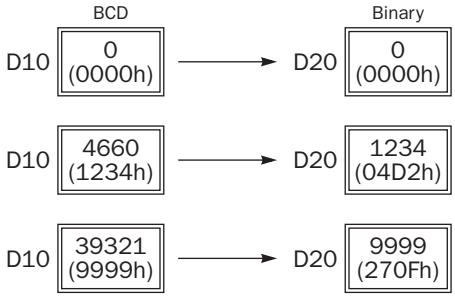
Valid Data Types

W (word)	I (integer)
X	—

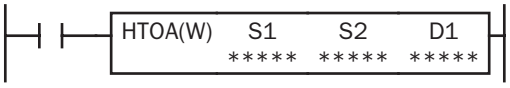
When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used.

Example: BTOH



HTOA (Hex to ASCII)



S1 → D1, D1+1, D1+2, D1+3

When input is on, the 16-bit binary data designated by S1 is read from the lowest digit as many as the quantity of digits designated by S2, converted into ASCII data, and stored to the destination starting with the operand designated by D1.

The quantity of digits to convert can be 1 through 4.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Binary data to convert	X	X	X	X	X	X	X	X	—
S2 (Source 2)	Quantity of digits to convert	X	X	X	X	X	X	X	1-4	—
D1 (Destination 1)	Destination to store conversion results	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

When T (timer) or C (counter) is used as S1 or S2, the timer/counter current value is read out.

The quantity of digits to convert can be 1 through 4. Make sure that the quantity of digits designated by S2 is within the valid range. If the S2 data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Since the HTOA instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Valid Data Types

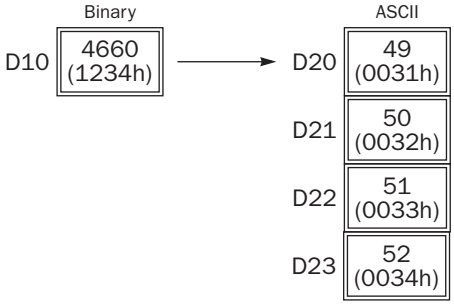
W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

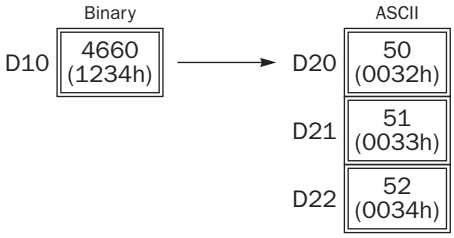
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used.

Examples: HTOA

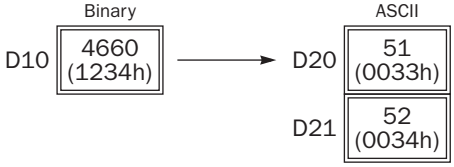
- **Quantity of Digits: 4**



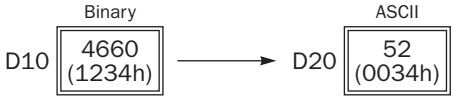
- **Quantity of Digits: 3**



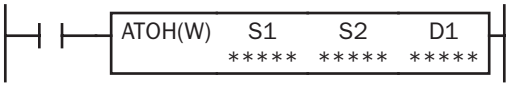
- **Quantity of Digits: 2**



- **Quantity of Digits: 1**



ATOH (ASCII to Hex)



S1, S1+1, S1+2, S1+3 → D1

When input is on, the ASCII data designated by S1 as many as the quantity of digits designated by S2 is converted into 16-bit binary data, and stored to the destination designated by operand D1.

Valid values for source data to convert are 30h to 39h and 41h to 46h.

The quantity of digits to convert can be 1 through 4.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	ASCII data to convert	—	—	—	—	—	—	X	—	—
S2 (Source 2)	Quantity of digits to convert	X	X	X	X	X	X	X	1-4	—
D1 (Destination 1)	Destination to store conversion results	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Valid values for source S1 data to convert are 30h to 39h and 41h to 46h. Make sure that the values for each source designated by S1 and the quantity of digits designated by S2 are within the valid range. If the S1 or S2 data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Since the ATOH instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Valid Data Types

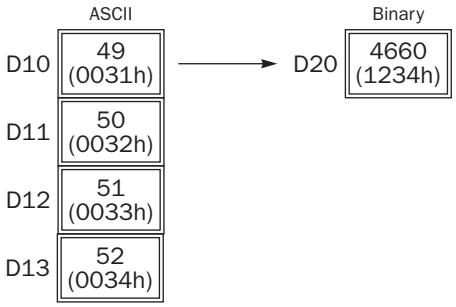
W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used.

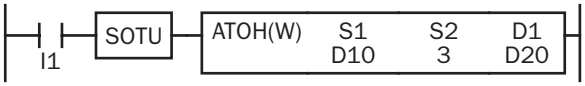
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used.

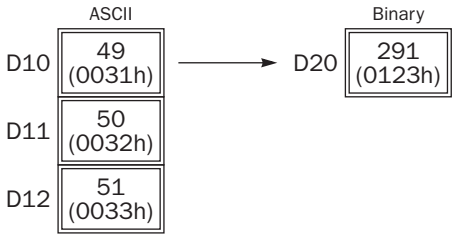
Examples: ATOH

- **Quantity of Digits: 4**

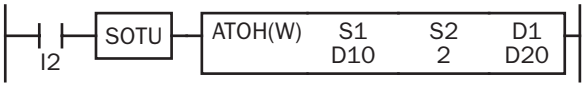


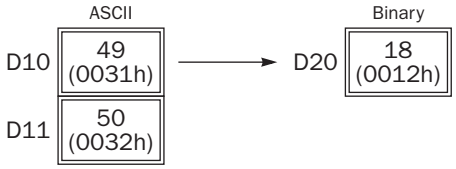
- **Quantity of Digits: 3**



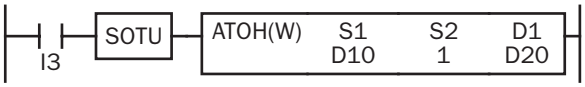


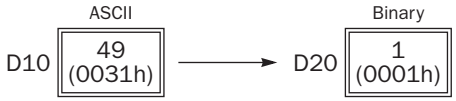
- **Quantity of Digits: 2**



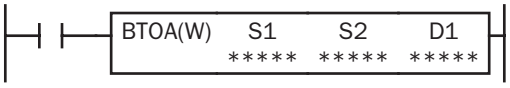


- **Quantity of Digits: 1**





BTOA (BCD to ASCII)



S1 → D1, D1+1, D1+2, D1+3, D1+4

When input is on, the 16-bit binary data designated by S1 is converted into BCD, and converted into ASCII data. The data is read from the lowest digit as many as the quantity of digits designated by S2. The result is stored to the destination starting with the operand designated by D1.

The quantity of digits to convert can be 1 through 5.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Binary data to convert	X	X	X	X	X	X	X	X	—
S2 (Source 2)	Quantity of digits to convert	X	X	X	X	X	X	X	1-5	—
D1 (Destination 1)	Destination to store conversion results	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

When T (timer) or C (counter) is used as S1 or S2, the timer/counter current value is read out.

The quantity of digits to convert can be 1 through 5. Make sure that the quantity of digits designated by S2 is within the valid range. If the S2 data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Since the BTOA instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Valid Data Types

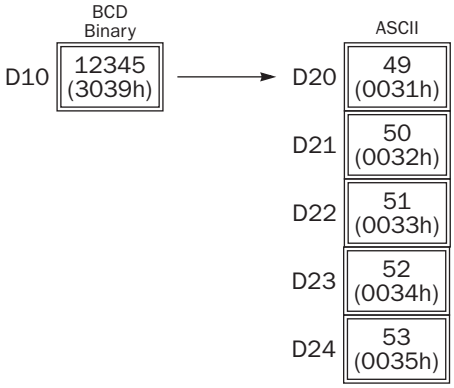
W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source, 16 points are used.

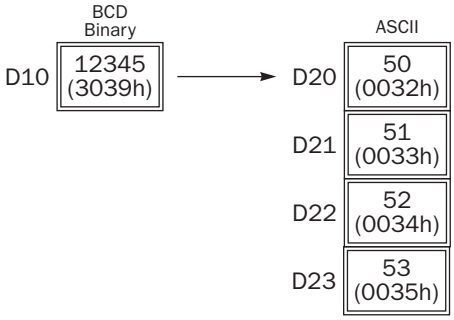
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used.

Examples: BTOA

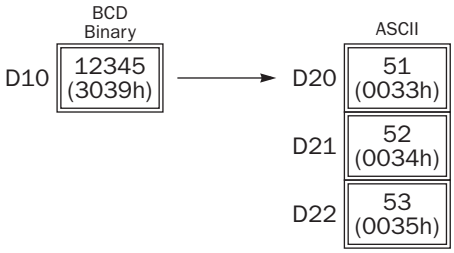
- **Quantity of Digits: 5**



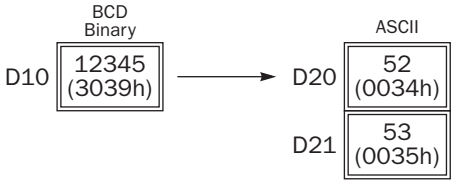
- **Quantity of Digits: 4**



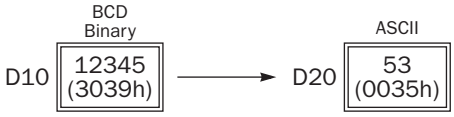
- **Quantity of Digits: 3**



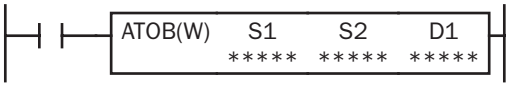
- **Quantity of Digits: 2**



- **Quantity of Digits: 1**



ATOB (ASCII to BCD)



S1, S1+1, S1+2, S1+3, S1+4 → D1

When input is on, the ASCII data designated by S1 as many as the quantity of digits designated by S2 is converted into BCD, and converted into 16-bit binary data. The result is stored to the destination designated by operand D1.

Valid values for source data to convert are 30h through 39h.

The quantity of digits to convert can be 1 through 5.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	ASCII data to convert	—	—	—	—	—	—	X	—	—
S2 (Source 2)	Quantity of digits to convert	X	X	X	X	X	X	X	1-5	—
D1 (Destination 1)	Destination to store conversion results	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Valid values for source S1 data to convert are 30h through 39h. Make sure that the values for each source designated by S1 and the quantity of digits designated by S2 are within the valid range. If the S1 or S2 data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Since the ATOB instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Valid Data Types

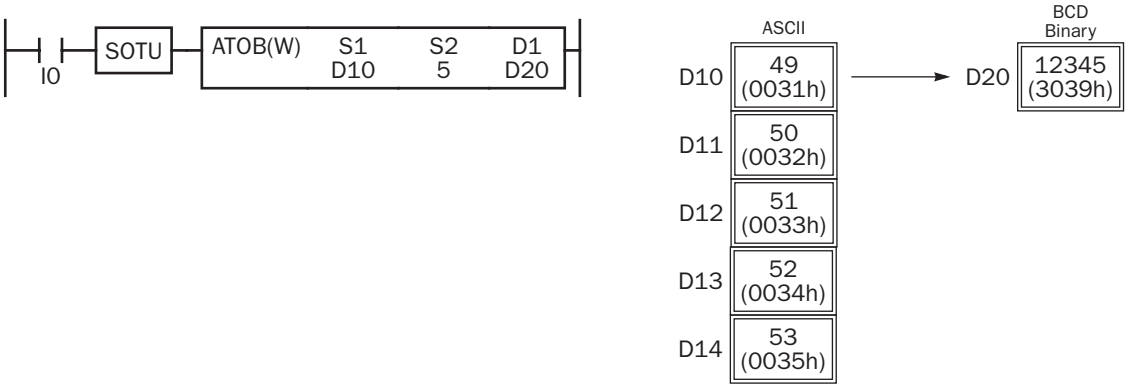
W (word)	I (integer)
X	—

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as the source or destination, 16 points are used.

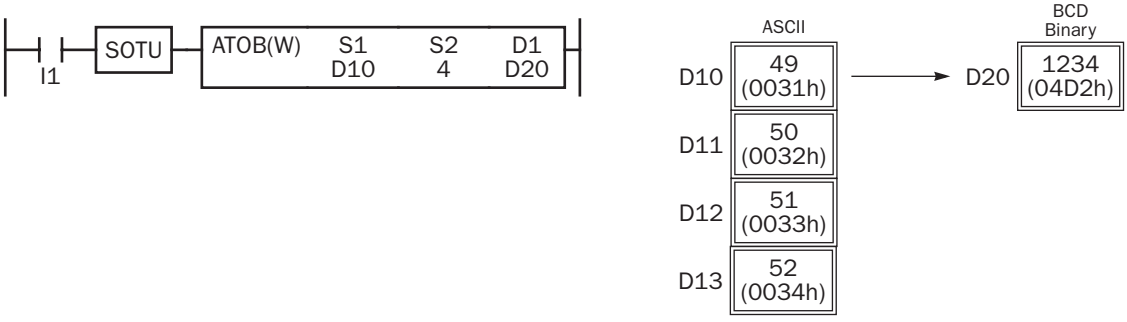
When a word operand such as T (timer), C (counter), or D (data register) is designated as the source or destination, 1 point is used.

Examples: ATOB

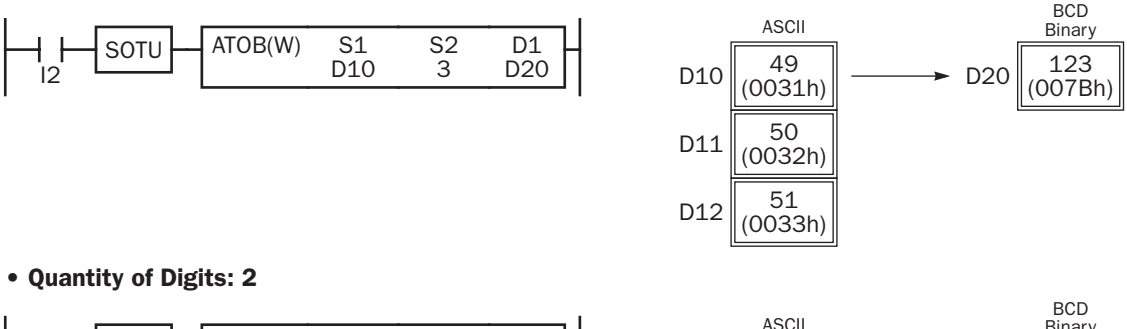
- **Quantity of Digits: 5**



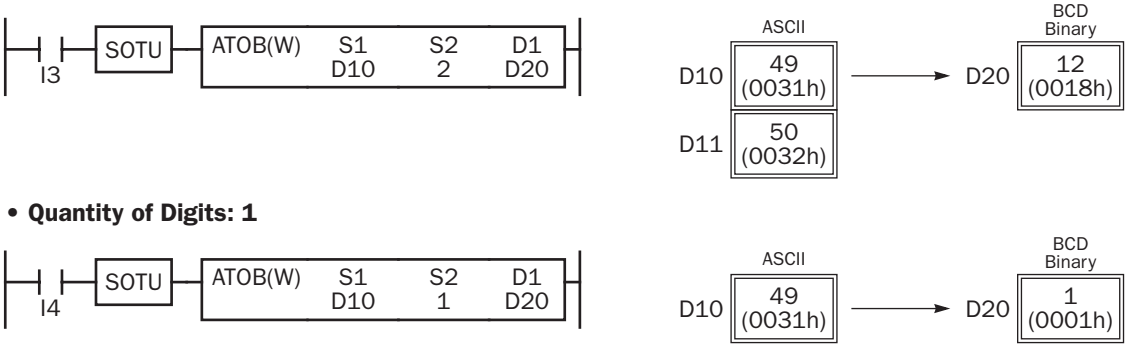
- **Quantity of Digits: 4**



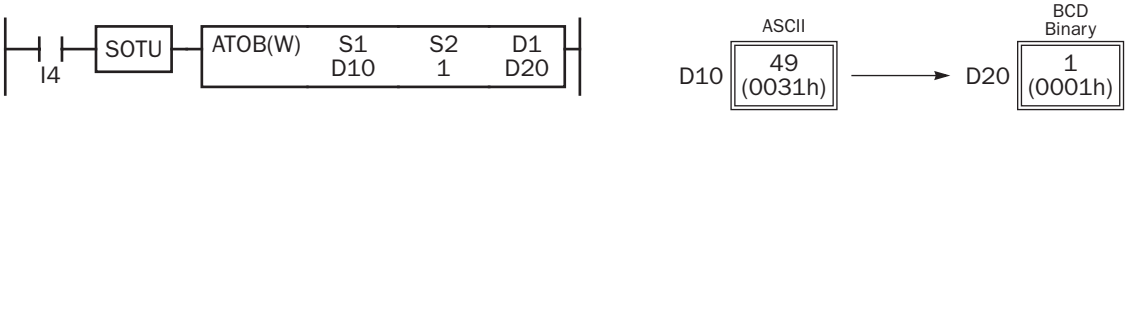
- **Quantity of Digits: 3**



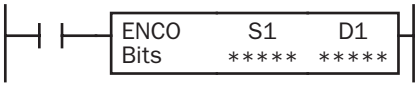
- **Quantity of Digits: 2**



- **Quantity of Digits: 1**



ENCO (Encode)



When input is on, a bit which is on is sought. The search begins at S1 until the first point which is set (on) is located. The quantity of points from S1 to the first set point (offset) is stored to the destination designated by operand D1.

If no point is on in the searched area, 65535 is stored to D1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	First bit to start search	X	X	X	X	—	—	X	—	—
D1 (Destination 1)	Destination to store search results	—	X	▲	X	—	—	X	—	—
Bits	Quantity of bits searched	—	—	—	—	—	—	—	1-256	—

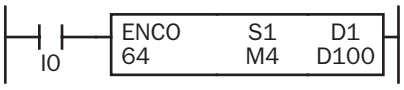
For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

Valid values for Bits to designate the quantity of bits searched are 1 through 256. Make sure that the search area designated by S1 plus Bits is within the valid value range. If the source data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

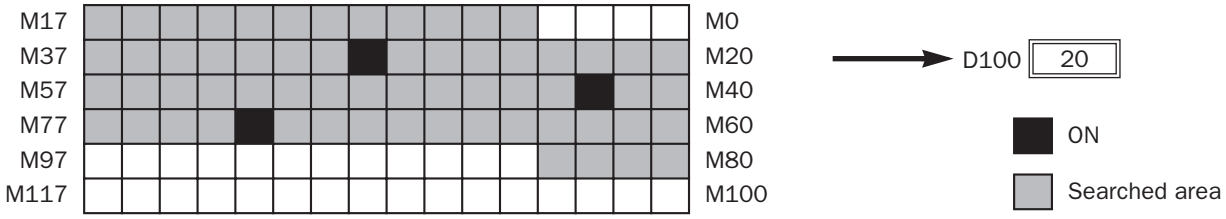
Since the ENCO instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

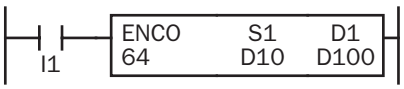
Examples: ENCO



When input I0 is on, a bit which is on is sought in 64 bits starting at internal relay M4 designated by operand S1.

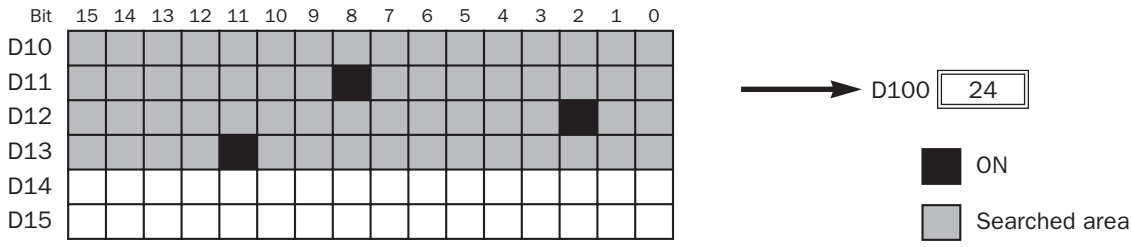
Since internal relay M30 is the first point that is on, the offset from the first search point is 20, and 20 is stored to data register D100 designated by operand D1.



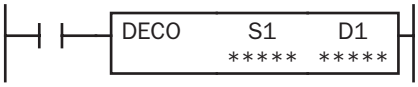


When input I1 is on, a bit which is on is sought in 64 bits starting at bit 0 of data register D10 designated by operand S1.

Since bit 8 of data register D11 is the first point that is on, the offset from the first search point is 24, and 24 is stored to data register D100 designated by operand D1.



DECO (Decode)



When input is on, the values contained in operands designated by S1 and D1 are added to determine the destination, and the bit so determined is turned on.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Offset	X	X	X	X	—	—	X	0-255	—
D1 (Destination 1)	First bit to count offset	—	X	▲	X	—	—	X	—	—

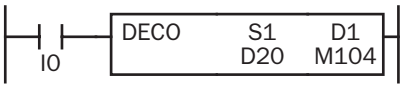
For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

Valid values for the offset designated by source operand S1 are 0 through 255. Make sure that the offset designated by S1 and the last bit of destination data determined by the sum of S1 and D1 are within the valid value range. If the offset or destination data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

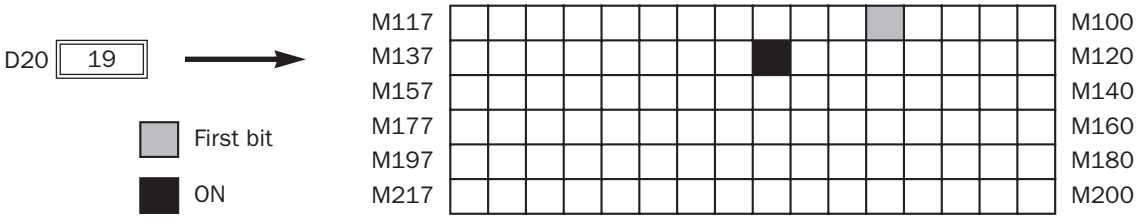
Since the DECO instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

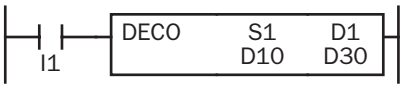
Examples: DECO



When input I0 is on, the destination bit is determined by adding the value contained in data register D20 designated by operand S1 to internal relay M104 designated by destination operand D1.

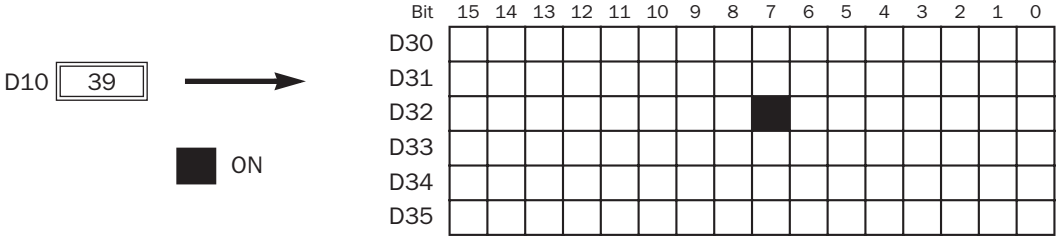
Since 19th bit from internal relay M104 is internal relay M127, the bit so determined is turned on.



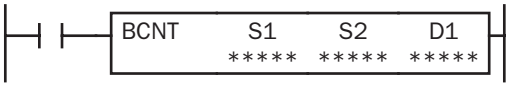


When input I1 is on, the destination bit is determined by adding the value contained in data register D10 designated by operand S1 to data register D30 designated by destination operand D1.

Since 39th bit from data register D30 bit 0 is data register D32 bit 7, the bit so determined is turned on.



BCNT (Bit Count)



When input is on, bits which are on are sought in an array of consecutive bits starting at the point designated by source operand S1. Source operand S2 designates the quantity of bits searched. The quantity of bits which are on is stored to the destination designated by operand D1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	First bit to start search	X	X	X	X	—	—	X	—	—
S2 (Source 2)	Quantity of bits searched	X	X	X	X	X	X	X	1-256	—
D1 (Destination 1)	Destination to store quantity of ON bits	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

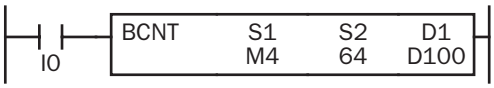
▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

Valid values for S2 to designate the quantity of bits searched are 1 through 256. Make sure that the search area designated by S1 plus S2 is within the valid value range. If the source data is out of the valid range, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

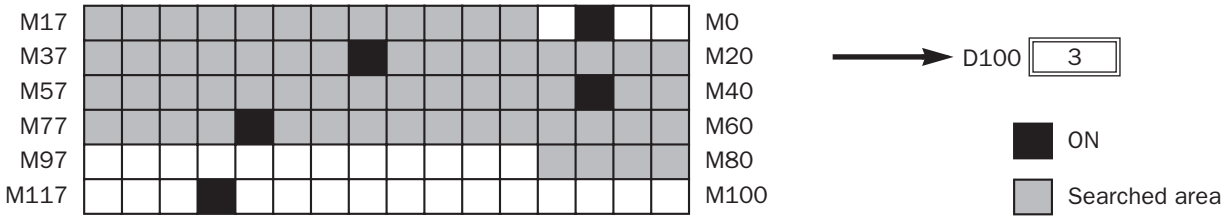
Since the BCNT instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

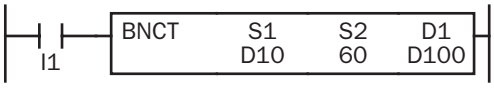
Examples: BCNT



When input is on, bits which are on are sought in an array of 64 bits starting at internal relay M4 designated by source operand S1.

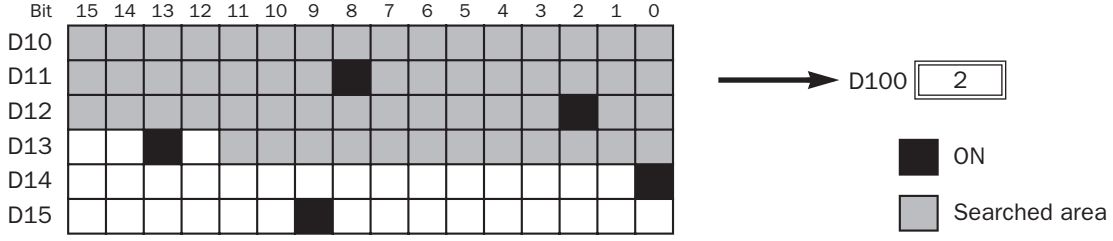
Since 3 bits are on in the searched area, the quantity is stored to data register D100 designated by destination operand D1.



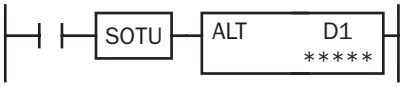


When input I0 is on, bits which are on are sought in 60 bits starting at bit 0 of data register D10 designated by operand S1.

Since 2 bits are on among the 60 bits, 2 is stored to data register D100 designated by operand D1.



ALT (Alternate Output)



When input is turned on, output, internal relay, or shift register bit designated by D1 is turned on and remains on after the input is turned off.

When input is turned on again, the designated output, internal relay, or shift register bit is turned off.

The ALT instruction must be used with a SOTU or SOTD instruction, otherwise the designated output, internal relay, or shift register bit repeats to turn on and off in each scan.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

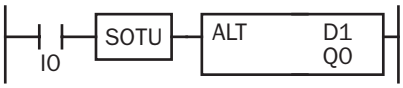
Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
D1 (Destination 1)	Bit to turn on and off	—	X	X	X	—	—	—	—	—

For the valid operand number range, see pages 6-1 and 6-2.

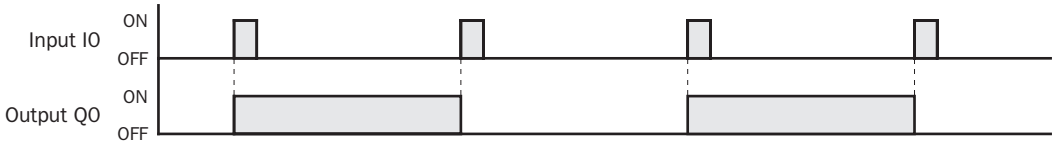
Since the ALT instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction must be used.

Example: ALT



When input I0 is turned on, output Q0 designated by operand D1 is turned on and remains after input I0 is turned off.

When input I0 is turned on again, output Q0 is turned off.



15: WEEK PROGRAMMER INSTRUCTIONS

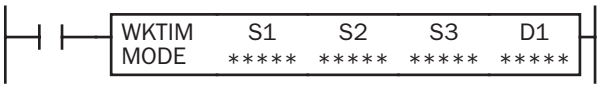
Introduction

WKTIM instructions can be used as many as required to turn on and off designated outputs and internal relays at predetermined times and days of the week.

Once the internal calendar/clock is set, the WKTIM instruction compares the predetermined time with the clock data in the clock cartridge. When the preset time is reached, internal relay or output designated as destination operand is turned on or off as scheduled. For setting the calendar/clock, see page 15-5.

For the specifications of the clock cartridge, see page 2-68.

WKTIM (Week Timer)



When input is on, the WKTIM compares the S1 and S2 preset data with the current day and time.

When the current day and time reach the presets, an output or internal relay designated by operand D1 is turned on, depending on the week table output control designated by MODE.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
MODE	Week table output control	—	—	—	—	—	—	—	0-2	—
S1 (Source 1)	Day of week comparison data	—	—	—	—	—	—	X	0-127	—
S2 (Source 2)	Hour/minute comparison data to turn on	—	—	—	—	—	—	X	0-2359	—
S3 (Source 3)	Hour/minute comparison data to turn off	—	—	—	—	—	—	X	0-2359	—
D1 (Destination 1)	Comparison ON output	—	X	▲	—	—	—	—	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

MODE — Week table output control (0 through 2)

0: Disable the week table

When the current day and time reach the presets for S1, S2, and S3, the designated output or internal relay is turned on or turned off. Set 0 for MODE when the WKTBL is not used; the WKTBL instruction is ignored even if it is programmed.

1: Additional days in the week table

When the current time reaches the hour/minute comparison data set for S2 or S3 on the special day programmed in the WKTBL, the designated output or internal relay is turned on (S2) or turned off (S3).

2: Skip days in the week table

On the special day programmed in the WKTBL, the designated output or internal relay is not turned on or off, even when the current day and time reach the presets for S1, S2, and S3.

Note: When 1 or 2 is set for MODE, program special days in the week table using the WKTBL instruction, followed by the WKTIM instruction. If the WKTBL instruction is not programmed when 1 or 2 is set for MODE in the WKTIM instruction, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module. The same error also occurs if the WKTIM instruction is executed before the WKTBL instruction.

S1 — Day of week comparison data (0 through 127)

Specify the days of week to turn on the output or internal relay designated by D1.

Day of Week	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Value	1	2	4	8	16	32	64

Designate the total of the values as operand S1 to turn on the output or internal relay.

Example: To turn on the output on Mondays through Fridays, designate 62 as S1 because 2 + 4 + 8 + 16 + 32 = 62.

S2 — Hour/minute comparison data to turn on
S3 — Hour/minute comparison data to turn off

Specify the hours and minutes to turn on (S2) or to turn off (S3) the output or internal relay designated by D1.

Hour	Minute	Disable Comparison
00 through 23	00 through 59	10000

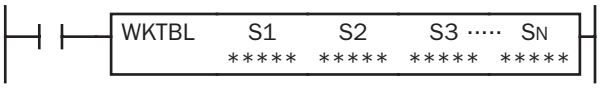
Example: To turn on the output or internal relay at 8:30 a.m. using the WKTIM instruction, designate 830 as S2. To turn off the output or internal relay at 5:05 p.m., designate 1705 as S3.

When 10000 is set to hour/minute comparison data, the comparison data is ignored. For example, if 10000 is set to the hour/minute comparison data to turn off (S3), the WKTIM instruction compares only the hour/minute comparison data to turn on (S2).

When the hour/minute comparison data to turn on (S2) is larger than the hour/minute comparison data to turn off (S3), the comparison ON output (D1) turns on at S2 on the day designated by S1, remains on across 0 a.m., and turns off at S3 on the next day. For example, if S2 is 2300, S3 is 100, and Monday is included in S1, then the output designated by D1 turns on at 23 p.m. on Monday and turns off at 1 a.m. on Tuesday.

Make sure that the values set for MODE, S1, S2, and S3 are within the valid ranges. If any data is over the valid value, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

WKTBL (Week Table)



S1, S2, S3, ... , SN → Week Table (N ≤ 20)

When input is on, N blocks of special month/day data in operands designated by S1, S2, S3, ... , SN are set to the week table.

The quantity of special days can be up to 20.

The special days stored in the week table are used to add or skip days to turn on or off the comparison outputs programmed in subsequent WKTIM instructions.

The WKTBL must precede the WKTIM instructions.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Special month/day data	—	—	—	—	—	—	X	101-1231	—

For the valid operand number range, see pages 6-1 and 6-2.

S1 through SN — Special month/day data

Specify the months and days to add or skip days to turn on or off the comparison outputs programmed in WKTIM instructions.

Month	Day
01 through 12	01 through 31

Example: To set July 4 as a special day, designate 704 as S1.

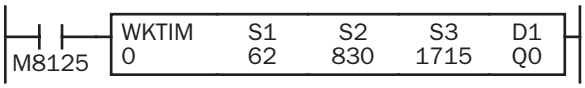
Make sure that the values set for S1 through SN are within the valid ranges. If any data is over the valid value, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Examples: WKTIM and WKTBL

• Without Special Days (MODE = 0)

This example is the basic program for week programmer application without using the WKTBL (week table) instruction. While the CPU is running, the WKTIM compares the S1, S2, and S3 preset data with the current day and time.

When the current day and time reach the presets, an output designated by operand D1 is turned on and off.



M8125 is the in-operation output special internal relay.

S1 (62) specifies Monday through Friday.

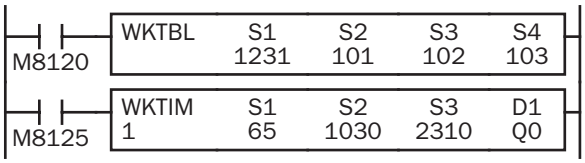
The WKTIM turns on output Q0 at 8:30 and turns off output Q0 at 17:15 on Monday through Friday.

• With Additional Days in the Week Table (MODE = 1)

When the current time reaches the hour/minute preset time on the special days programmed in the WKTBL, the designated output is turned on or turned off. In addition, the designated output is turned on and off every week as designated by operand S1 of WKTIM.

In normal execution, when the current day and time coincide with the preset day (S1) and time (S2 or S3) of the WKTIM, the designated output is turned on or off. Execution on the special days has precedence over execution on normal days.

This example demonstrates operation on special days in addition to regular weekends. The output is turned on from 10:30 a.m. to 11:10 p.m. on every Saturday and Sunday. Without regard to the day of week, the output is also turned on December 31 through January 3.



M8120 is the initialize pulse special internal relay.

WKTBL designates Dec. 31 to Jan. 3 as special days.

MODE (1) adds special days.

S1 (65) specifies Saturday and Sunday.

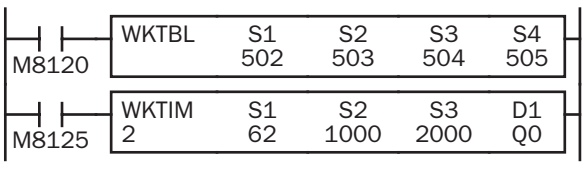
WKTIM turns on output Q0 at 10:30 and turns off at 23:10 on every Saturday, Sunday, and special days.

• With Skip Days in the Week Table (MODE = 2)

On the special days programmed in the WKTBL, the designated output is *not* turned on or off, while the designated output is turned on and off every week as designated by operand S1 of WKTIM.

In normal execution, when the current day and time coincide with the preset day (S1) and time (S2 or S3), the designated output is turned on or off. Execution on the special days has precedence over execution on normal days.

This example demonstrates operation aborted on special days. The output is turned on from 10:00 a.m. to 8:00 p.m. on every Monday through Friday, but is not turned on from May 2 through May 5.



WKTBL designates May 2 to May 5 as special days.

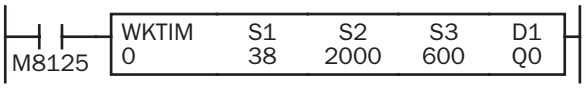
MODE (2) skips special days.

S1 (62) specifies Monday to Friday.

WKTIM turns on output Q0 at 10:00 and turns off at 20:00 on every Monday through Friday except on special days.

• Keep Output ON across 0 a.m.

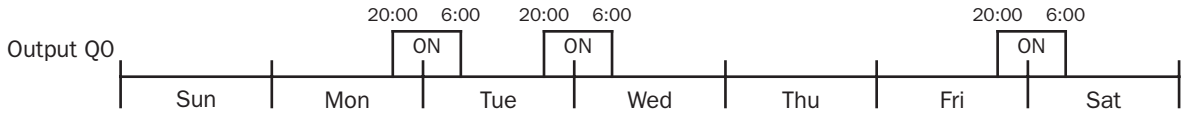
When the hour/minute comparison data to turn on (S2) is larger than the hour/minute comparison data to turn off (S3), the comparison ON output (D1) turns on at S2 on the day designated by S1, remains on across 0 a.m., and turns off at S3 on the next day. This example demonstrates a program to keep the designated output on across 0 a.m. and turn off the output on the next day.



M8125 is the in-operation output special internal relay.

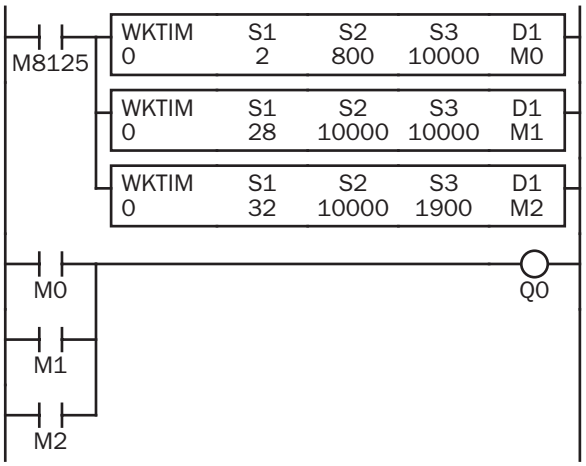
S1 (38) specifies Monday, Tuesday, and Friday.

The WKTIM turns on output Q0 at 20:00 on Monday, Tuesday, and Friday, and turns off output Q0 at 6:00 on the next day.



• Keep Output ON for Several Days

Multiple WKTIM instructions can be used to keep an output on for more than 24 hours. This example demonstrates a program to keep the designated output on from 8 a.m on every Monday to 7 p.m. on every Friday.



M8125 is the in-operation output special internal relay.

S1 (2) specifies Monday.

S1 (28) specifies Tuesday, Wednesday, and Thursday.

S1 (32) specifies Friday.

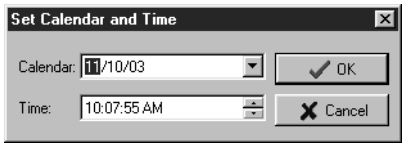
S2 (10000) and S3 (10000) disable comparison of hour and minute data.

While internal relay M0, M1, or M2 is on, output Q0 remains on.

Setting Calendar/Clock Using WindLDR

Before using the clock cartridge for the first time, the calendar/clock data in the clock cartridge must be set using WindLDR or executing a user program to transfer correct calendar/clock data from special data registers allocated to the calendar/clock. Once the calendar/clock data is stored, the data is held by the backup battery in the clock cartridge.

1. Select **Online** from the WindLDR menu bar, then select **Monitor**. The screen display changes to the monitor window.
2. From the **Online** menu, select **PLC Status**. The MicroSmart PLC Status dialog box is displayed. The current calendar/clock data is read out from the clock cartridge and displayed in the Calendar box.
3. Click the **Change** button in the Calendar box. The Set Calendar and Time dialog box comes up with the date and time values read from the computer internal clock.



4. Click the **Down Arrow** button on the right of **Calendar**, then a calendar is displayed where you can change the year, month, and date. Enter or select new values.
5. To change hours and minutes, click in the **Time** box, and type a new value or use the up/down keys. When new values are entered, click the **OK** button to transfer the new values to the clock cartridge.

Setting Calendar/Clock Using a User Program

Another way of setting the calendar/clock data is to store the values in special data registers dedicated to the calendar and clock and to turn on special internal relay M8016, M8017, or M8020. Data registers D8015 through D8021 do not hold the current values of the calendar/clock data but hold unknown values before executing a user program.

Special Data Registers for Calendar/Clock Data

Data Register No.	Data	Value	Read/Write	Updated
D8008	Year (current data)	0 to 99	Read only	500 ms or one scan time whichever is larger
D8009	Month (current data)	1 to 12		
D8010	Day (current data)	1 to 31		
D8011	Day of week (current data)	0 to 6 (Note)		
D8012	Hour (current data)	0 to 23		
D8013	Minute (current data)	0 to 59		
D8014	Second (current data)	0 to 59		
D8015	Year (new data)	0 to 99	Write only	Not updated
D8016	Month (new data)	1 to 12		
D8017	Day (new data)	1 to 31		
D8018	Day of week (new data)	0 to 6 (Note)		
D8019	Hour (new data)	0 to 23		
D8020	Minute (new data)	0 to 59		
D8021	Second (new data)	0 to 59		

Note: The day of week value is assigned for both current and new data as follows:

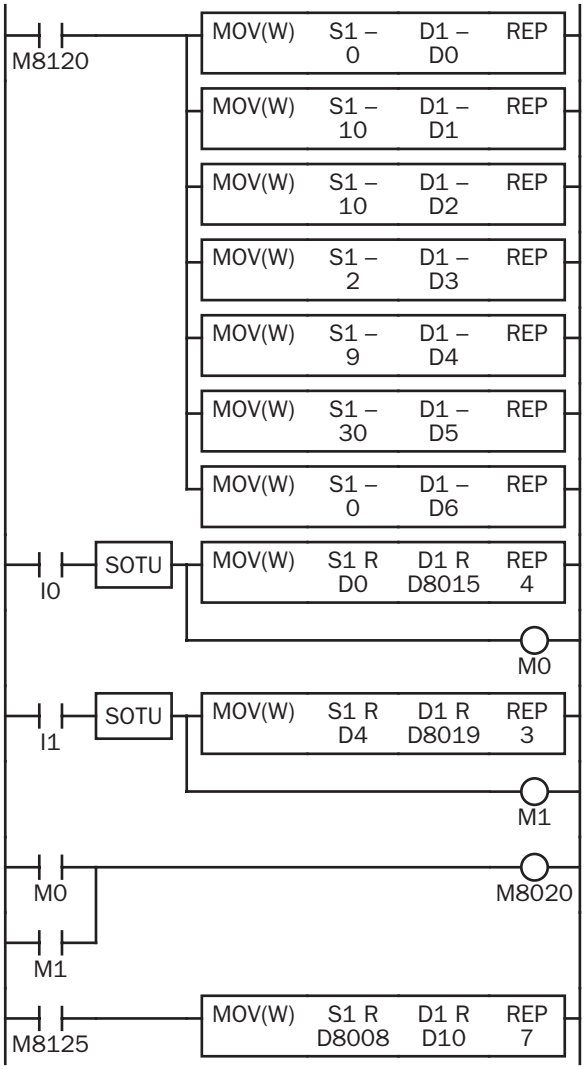
0	1	2	3	4	5	6
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Special Internal Relays for Calendar/Clock Data

M8016	Calendar Data Write Flag	When M8016 is turned on, data in data registers D8015 through D8018 (calendar new data) are set to the clock cartridge installed on the CPU module.
M8017	Clock Data Write Flag	When M8017 is turned on, data in data registers D8019 through D8021 (clock new data) are set to the clock cartridge installed on the CPU module.
M8020	Calendar/Clock Data Write Flag	When M8020 is turned on, data in data registers D8015 through D8021 (calendar/clock new data) are set to the clock cartridge installed on the CPU module.

Example: Setting Calendar/Clock Data

This example demonstrates how to set calendar/clock data using a ladder program. After storing new calendar/clock data into data registers D8015 through D8021, special internal relay M8020 (calendar/clock data write flag) must be turned on to set the new calendar/clock data to the clock cartridge.



M8120 is the initialize pulse special internal relay.

When the CPU starts, seven MOV(W) instructions store calendar/clock data to data registers D0 through D6.

When input I0 is turned on, new calendar data (year, month, day, and day of week) are moved to data registers D8015 through D8018, and internal relay M0 is turned on for 1 scan time.

When input I1 is turned on, new clock data (hour, minute, and second) are moved to data registers D8019 through D8021, and internal relay M1 is turned on for 1 scan time.

When either M0 or M1 is turned on, calendar/clock data write flag special internal relay M8020 is turned on to set the new calendar/clock data to the clock cartridge.

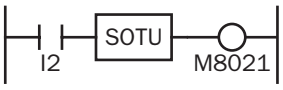
M8125 is the in-operation output special internal relay.

While the CPU is running, the MOV(W) moves current calendar/clock data to data registers D10 through D16.

Adjusting Clock Using a User Program

Special internal relay M8021 (clock data adjust flag) is provided for adjusting the clock data. When M8021 is turned on, the clock is adjusted with respect to seconds. If *seconds* are between 0 and 29 for current time, adjustment for *seconds* will be set to 0 and minutes remain the same. If *seconds* are between 30 and 59 for current time, adjustment for *seconds* will be set to 0 and *minutes* are incremented one. M8021 is useful for precise timing which starts at zero seconds.

Example: Adjusting Calendar/Clock Data to 0 Seconds

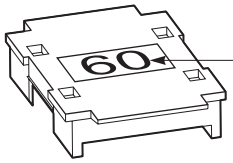


When input I2 is turned on, clock data adjust flag special internal relay M8021 is turned on and the clock is adjusted with respect to seconds.

Adjusting Clock Cartridge Accuracy

The optional clock cartridge (FC4A-PT1) has an initial monthly error of ±2 minutes at 25°C. The accuracy of the clock cartridge can be improved to ±30 seconds using Enable Clock Cartridge Adjustment in the Function Area Settings.

Before starting the clock cartridge adjustment, confirm the adjustment value indicated on the clock cartridge. This value is an adjustment parameter measured on each clock cartridge at factory before shipment.

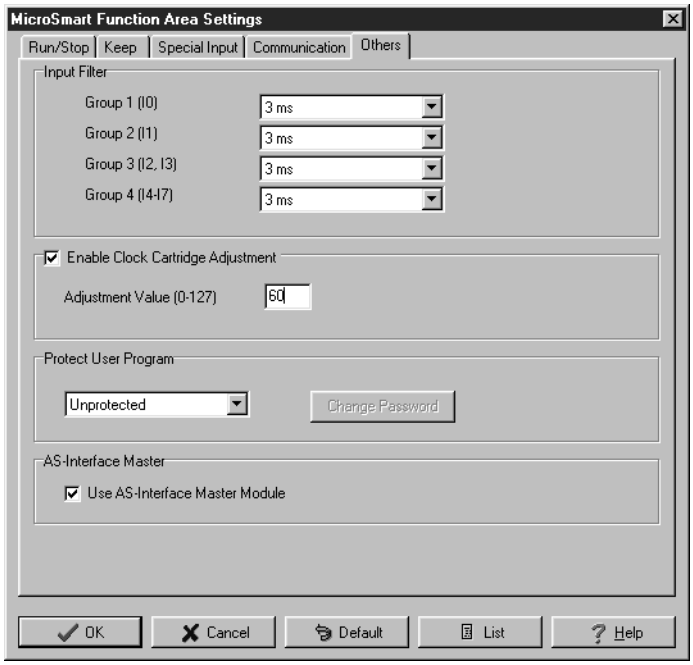


Adjustment Value

The adjustment value indicated on the clock cartridge was measured at 25°C to achieve the best accuracy. When using the clock cartridge at other temperatures, the clock cartridge accuracy may be impaired.

Programming WindLDR

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Setting dialog box appears.
2. Select the **Others** tab.



3. Click the check box to enable the clock cartridge adjustment, and type the adjustment value found on the clock cartridge in the Adjustment Value field.
4. Click the **OK** button.
5. Download the user program to the CPU module, and turn off and on the power to the CPU module.

Clock Cartridge Backup Duration

The clock cartridge data is backed up by a lithium battery in the clock cartridge and held for approximately 30 days at 25°C. If the CPU module is not powered up for a period longer than the backup duration, the clock data is initialized to the following values.

Calendar: 00/01/01
Time: 0:00:00 AM

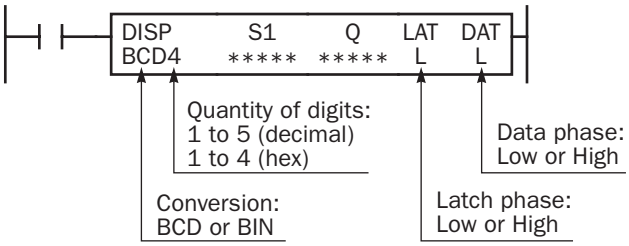
16: INTERFACE INSTRUCTIONS

Introduction

The DISP (display) instruction is used to display 1 through 5 digits of timer/counter current values and data register data on 7-segment display units.

The DGRD (digital read) instruction is used to read 1 through 5 digits of digital switch settings to a data register. This instruction is useful to change preset values for timers and counters using digital switches.

DISP (Display)



When input is on, data designated by source operand S1 is set to outputs or internal relays designated by operand Q. This instruction is used to output 7-segment data to display units.

Eight DISP instructions can be used in a user program.

Display data can be 0 through 65535 (FFFFh).

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Note: The DISP instruction requires transistor output terminals. When using all-in-one 24-I/O type CPU module FC4A-C24R2 or FC4A-C24R2C, connect a transistor output module.

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Data to display	—	—	—	—	X	X	X	—	—
Q (Output)	First output number to display data	—	X	▲	—	—	—	—	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as Q. Special internal relays cannot be designated as Q. When T (timer) or C (counter) is used as S1, the timer/counter current value is read out.

Conversion

BCD: To connect BCD (decimal) display units
BIN: To connect BIN (hexadecimal) display units

Latch Phase and Data Phase

Select the latch and data phases to match the phases of the display units in consideration of sink or source output of the output module.

Output Points

The quantity of required output points is 4 plus the quantity of digits to display. When displaying 4 digits with output Q0 designated as the first output number, 8 consecutive output points must be reserved starting with Q0 through Q7.

Display Processing Time

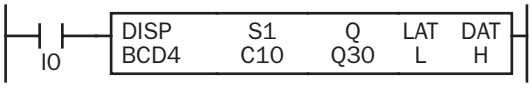
Displaying one digit of data requires 3 scan times after the input to the DISP instruction is turned on. Keep the input to the DISP instruction for the period of time shown below to process all digits of the display data.

Display Processing Time
3 scan times × Quantity of digits

When the scan time is less than 2 ms, the data cannot be displayed correctly. When the scan time is too short to ensure normal display, set a value of 3 or more (in ms) to data register D8022 (constant scan time preset value). See page 5-27.

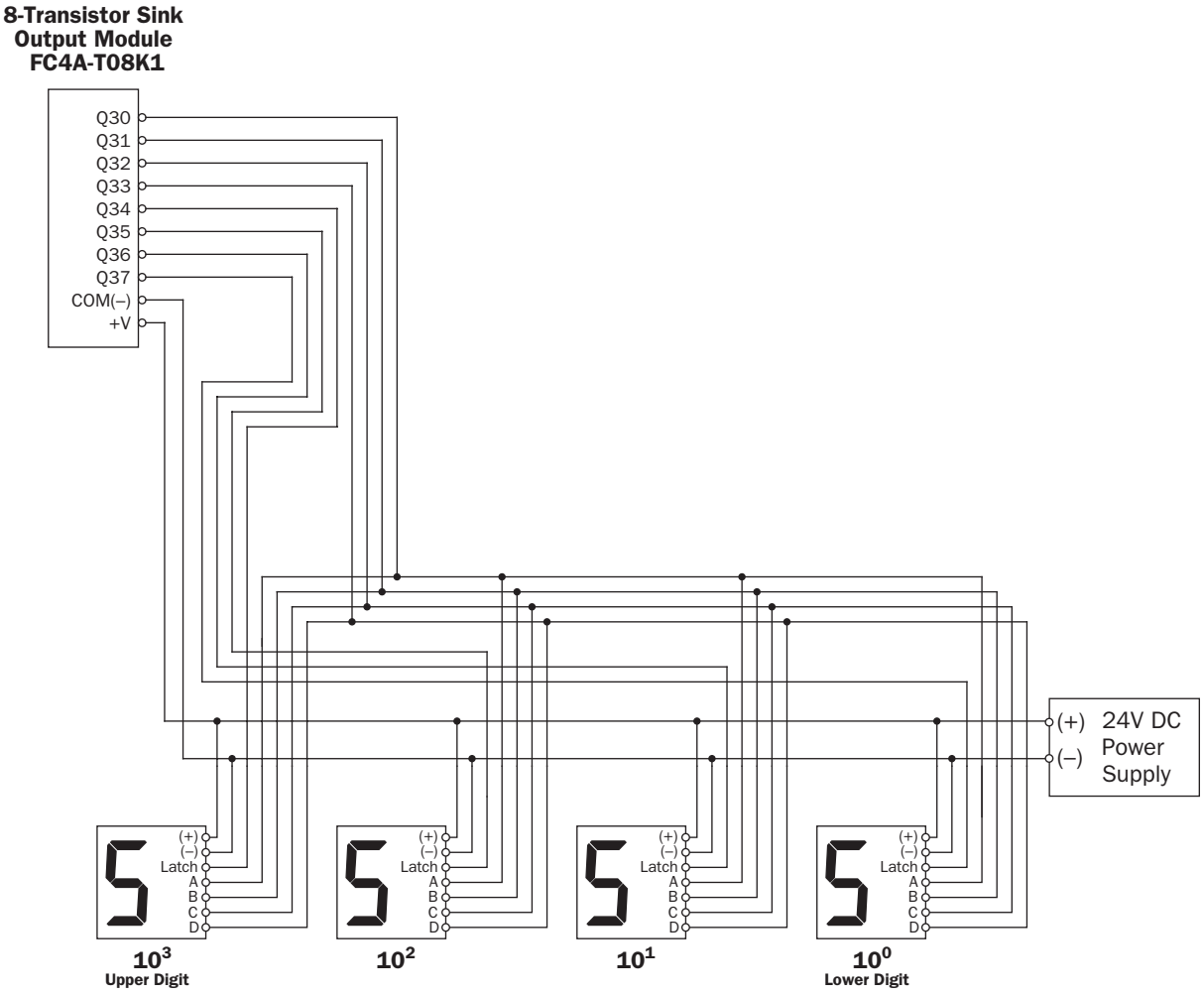
Example: DISP

The following example demonstrates a program to display the 4-digit current value of counter CNT10 on 7-segment display units (IDEC's DD3S-F31N) connected to the transistor sink output module.

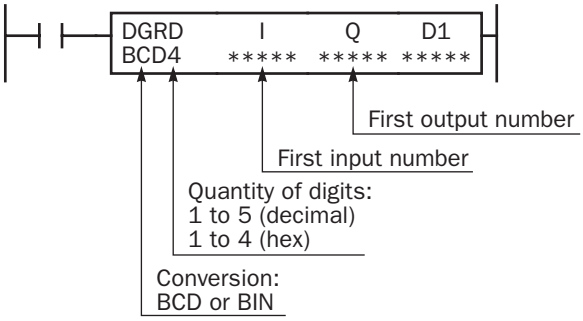


When input I0 is on, the 4-digit current value of counter C10 is displayed on 7-segment digital display units.

Output Wiring Diagram



DGRD (Digital Read)



When input is on, data designated by operands I and Q is set to a data register designated by destination operand D1.

This instruction can be used to change preset values for timer and counter instructions using digital switches. The data that can be read using this instruction is 0 through 65535 (5 digits), or FFFFh.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Note: The DGRD instruction requires transistor output terminals. When using all-in-one 24-I/O type CPU module FC4A-C24R2 or FC4A-C24R2C, connect a transistor output module.

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
I	First input number to read	X	—	—	—	—	—	—	—	—
Q	First output number for digit selection	—	X	—	—	—	—	—	—	—
D1 (Destination 1)	Destination to store results	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

The DGRD instruction can read 65535 (5 digits) at the maximum. When the read value exceeds 65535 with the quantity of digits set to 5, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

Note: The DGRD instruction can be used up to 16 times in a user program. When transferring a user program containing more than 16 DGRD instructions to the CPU, a user program syntax error occurs, turning on the ERR LED. The user program cannot be executed.

Conversion

BCD: To connect BCD (decimal) digital switches
BIN: To connect BIN (hexadecimal) digital switches

Input Points

Inputs are used to read the data from digital switches. The quantity of required input points is always 4. Four input points must be reserved starting with the input number designated by operand I. For example, when input I0 is designated as operand I, inputs I0 through I3 are used.

Output Points

Outputs are used to select the digits to read. The quantity of required output points is equal to the quantity of digits to read. When connecting the maximum of 5 digital switches, 5 output points must be reserved starting with the output number designated by operand Q. For example, when output Q0 is designated as operand Q to read 3 digits, outputs Q0 through Q2 are used.

Digital Switch Data Reading Time

Reading digital switch data requires the following time after the input to the DGRD instruction is turned on. Keep the input to the DGRD instruction for the period of time shown below to read the digital switch data. For example, when reading data from 5 digital switches to the destination operand, 14 scans are required

Digital Switch Data Reading Time
2 scan times × (Quantity of digits + 2)

Adjusting Scan Time

The DGRD instruction requires a scan time longer than the filter time plus 6 ms.

Minimum Required Scan Time
(Scan time) ≥ (Filter time) + 6 ms

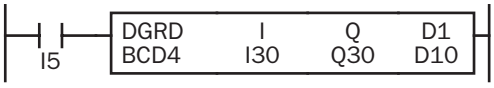
The filter time depends on the input terminal used as shown below.

Input Terminals	Filter Time
I0 through I7 on CPU Modules	Filter value selected in the Function Area Settings (default 3 ms) See Input Filter on page 5-24.
I10 through I15 on CPU Modules (except slim 40-I/O type CPU Module)	3 ms (fixed)
I10 through I27 on slim 40-I/O type CPU Module	4 ms (fixed)
Inputs on Expansion Input Modules	4 ms (fixed)

When the actual scan time is too short to execute the DGRD instruction, use the constant scan function. When the input filter time is set to 3 ms, set a value of 9 or more (in ms) to special data register D8022 (constant scan time preset value). See page 5-27. When the input filter time is changed, set a proper value to D8022 to make sure of the minimum required scan time shown above.

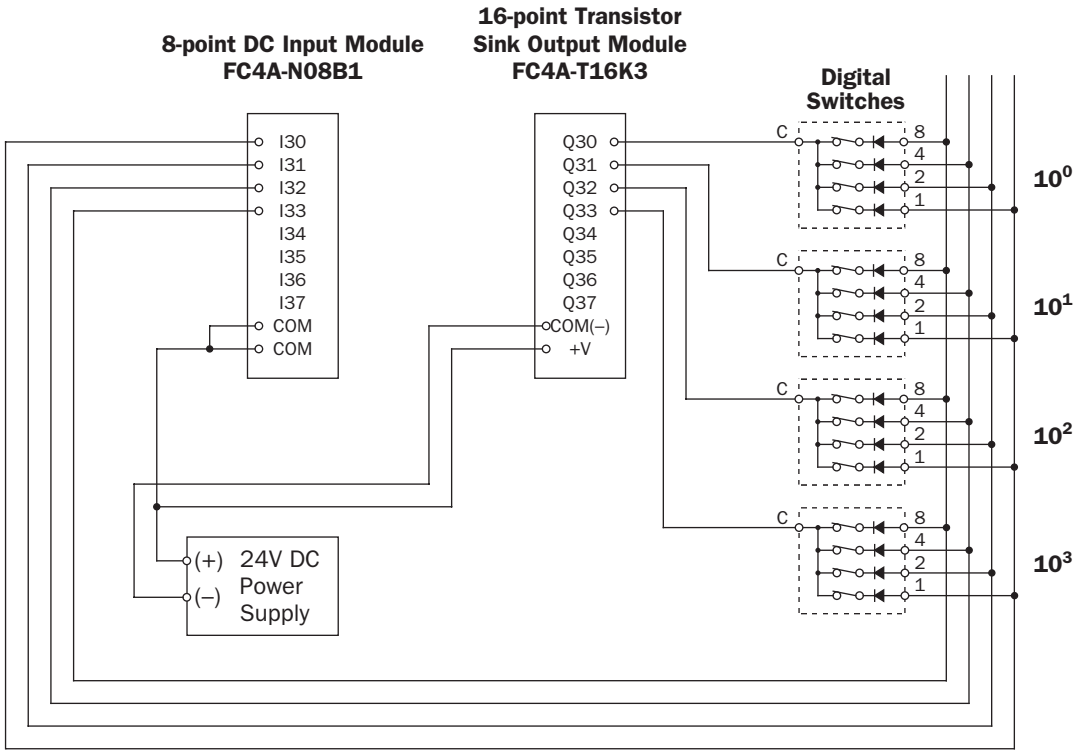
Example: DGRD

The following example demonstrates a program to read data from four digital switches (IDEC's DFBN-031D-B) to a data register in the CPU module, using a 8-point DC input module and a 16-point transistor sink output module.



When input I5 is on, the 4-digit value from BCD digital switches is read to data register D10.

I/O Wiring Diagram



17: User Communication Instructions

Introduction

This chapter describes the user communication function for communication between the MicroSmart and external devices with an RS232C port. The MicroSmart uses user communication instructions for transmitting and receiving communication to and from external devices.

Upgrade Information

Upgraded CPU modules of slim 20-I/O relay output types and 40-I/O types can also use the user communication through the RS485 port and three additional BCC calculation formulas; ADD-2Comp, Modbus ASCII, and Modbus RTU. Applicable CPU modules and system program version are shown in the table below. For the procedure to confirm the system program version of the CPU module, see page 29-1.

CPU Module	All-in-One Type			Slim Type	
	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
RS485 User Communication Compatibility	—	204 or higher	204 or higher	204 or higher	202 or higher
User Communication BCC Upgrade (ADD-2Comp, Modbus ASCII, and Modbus RTU)					

Using the RS485 user communication, the MicroSmart CPU module can communicate with a maximum of 31 RS485 devices.

The upgraded CPU modules can use three new BCC calculation formulas of ADD-2comp, Modbus ASCII, and Modbus RTU for transmit instructions TXD1 and TXD2 and receive instructions RXD1 and RXD2. Use WindLDR ver. 4.40 or higher to program the new BCC. For calculation examples, see page 17-36.

New BCC Calculation Formulas

BCC Name	Description
ADD-2comp	Add the characters in the range from the BCC calculation start position to the byte immediately before the BCC, then invert the result bit by bit, and add 1.
Modbus ASCII	Calculate the BCC using LRC (longitudinal redundancy check) for the range from the BCC calculation start position to the byte immediately before the BCC.
Modbus RTU	Calculate the BCC using CRC-16 (cyclic redundancy checksum) for the range from the BCC calculation start position to the byte immediately before the BCC. The generation polynomial is: $X^{16} + X^{15} + X^2 + 1$.

User Communication Overview

The user communication mode is used for linking the MicroSmart to an RS232C communication device such as a computer, modem, printer, or barcode reader.

The all-in-one 10-I/O type CPU module has one RS232C port. The 16- and 24-I/O type CPU modules have one RS232C port and port 2 connector as standard. By installing an optional RS232C communication adapter (FC4A-PC1) to the port 2 connector, the 16- and 24-I/O type CPU modules can communicate with two external devices simultaneously.

Every slim type CPU module has one RS232C port. An optional RS232C communication module can be attached to any slim type CPU module to use port 2 for additional RS232C communication. When an optional HMI base module is attached to a slim type CPU module, an optional RS232C communication adapter can be installed to the port 2 connector on the HMI base module.

User communication transmit and receive instructions can be programmed to match the communication protocol of the equipment to communicate with. Possibility of communication using the user communication mode can be determined referring to the user communication mode specifications described below.

User Communication Mode Specifications

Type	RS232C User Communication	RS485 User Communication
Communication Port	Port 1 and Port 2	Port 2
Connection Device Quantity	1 per port	31 maximum
Standards	EIA RS232C	EIA RS485
Baud Rate	1200, 2400, 4800, 9600, 19200 bps	
Data Bits	7 or 8 bits	
Parity	Odd, Even, None	
Stop Bits	1 or 2 bits	
Receive Timeout	10 to 2540 msec (10-msec increments) or none (Receive timeout is disabled when 2550 msec is selected.) The receive timeout has an effect when using RXD1/RXD2 instructions.	
Communication Method	Start-stop synchronization system half-duplex	
Maximum Cable Length	2.4m	200m
Maximum Transmit Data	200 bytes	
Maximum Receive Data	200 bytes	
BCC Calculation	XOR, ADD, ADD-2comp *, Modbus ASCII *, Modbus RTU *	

Note *: WindLDR 4.0 or higher is needed to use these BCC calculation formulas.

Connecting RS232C Equipment through RS232C Port 1 or 2

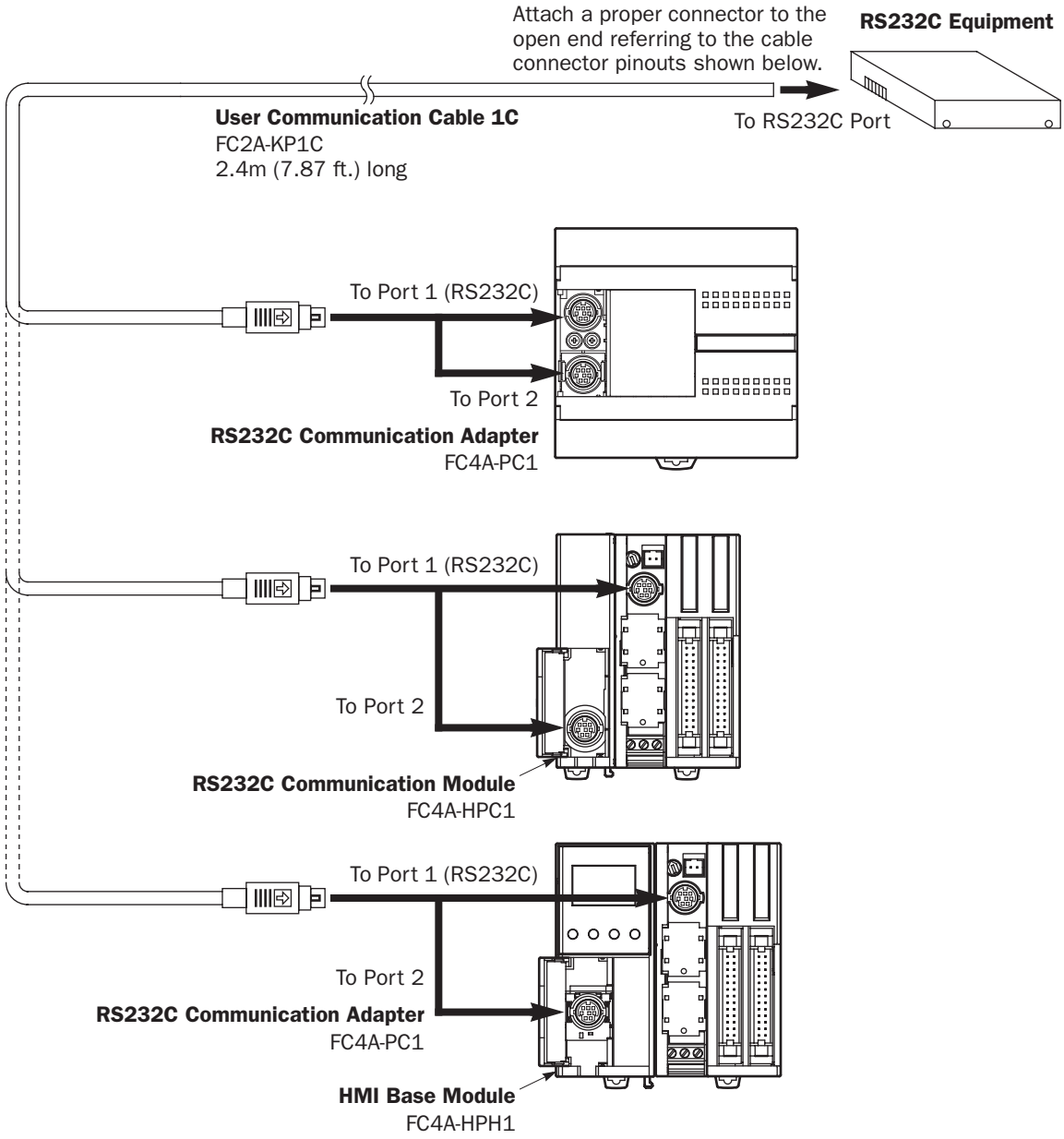
When using port 2 for RS232C communication on the all-in-one 16- or 24-I/O type CPU module, install the RS232C communication adapter (FC4A-PC1) to the port 2 connector.

When using port 2 for RS232C communication on the slim type CPU module, mount the RS232C communication module (FC4A-HPC1) next to the CPU module.

When using port 2 for RS232C communication on the slim type CPU module with the optional HMI module, install the RS232C communication adapter (FC4A-PC1) to the port 2 connector on the HMI base module.

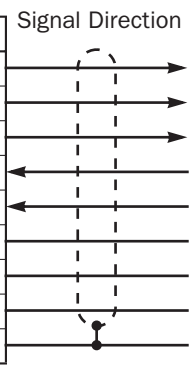
To connect an RS232C communication device to the RS232C port 1 or 2 on the MicroSmart CPU module, use the user communication cable 1C (FC2A-KP1C). One end of the user communication cable 1C is not provided with a connector, and can be terminated with a proper connector to plug in to communicate with the RS232C port. See the figure on page 17-3.

RS232C User Communication System Setup



Cable Connector Pinouts

Pin	Port 1	Port 2	AWG#		Color
1	NC (no connection)	RTS (request to send)	28	Twisted	Black
2	NC (no connection)	DTR (data terminal ready)	28		Yellow
3	TXD (transmit data)	TXD (transmit data)	28		Blue
4	RXD (receive data)	RXD (receive data)	28		Green
5	NC (no connection)	DSR (data set ready)	28		Brown
6	CMSW (communication switch)	SG (signal ground)	28		Gray
7	SG (signal ground)	SG (signal ground)	26	Twisted	Red
8	NC (no connection)	NC (no connection)	26		White
Cover	—	—	—		Shield



Note: When preparing a cable for port 1, keep pins 6 and 7 open. If pins 6 and 7 are connected together, user communication cannot be used.

Connecting RS485 Equipment through RS485 Port 2

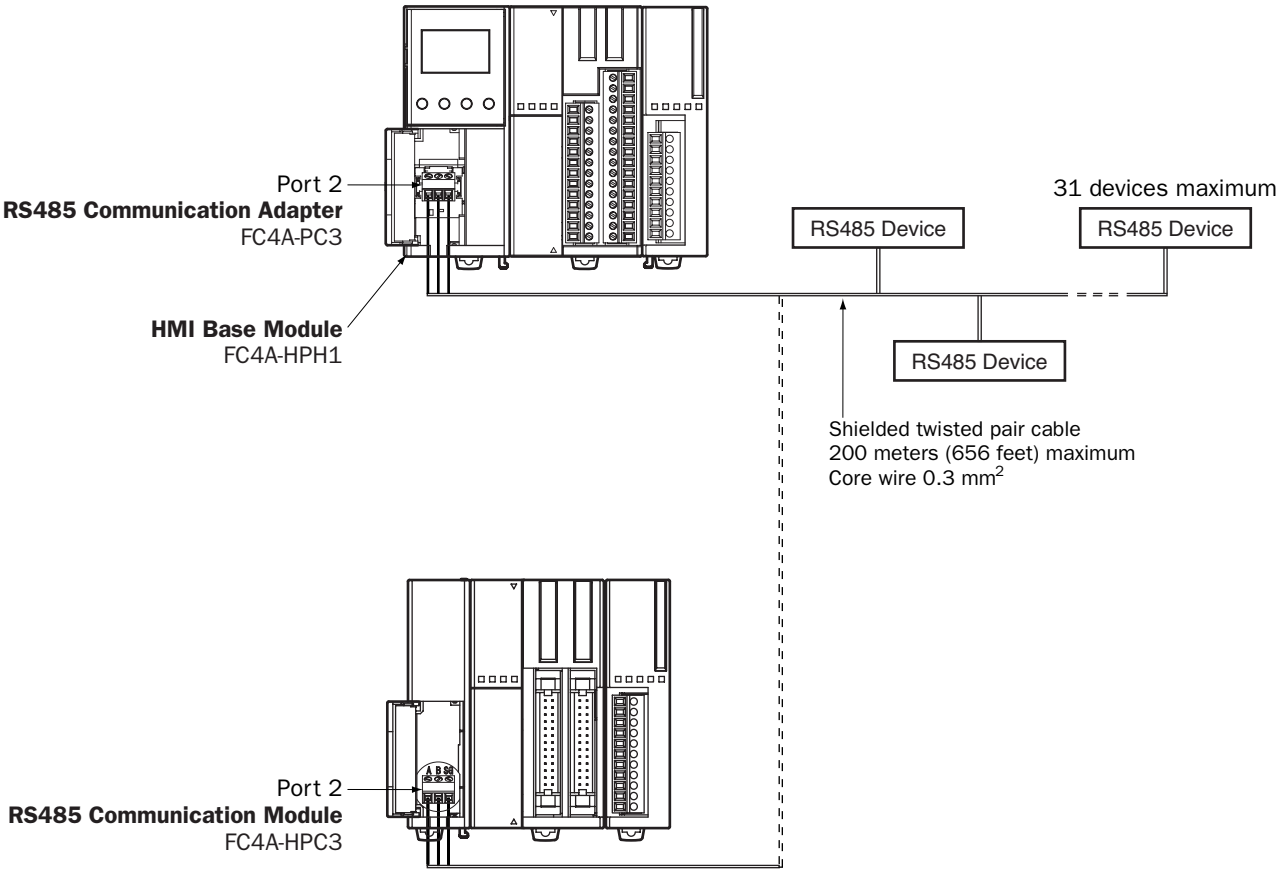
Upgraded slim type CPU modules can use the RS485 user communication function. Using the RS485 user communication, a maximum of 31 RS485 devices can be connected to the MicroSmart CPU module.

When using port 2 for RS485 communication on the slim type CPU module, mount the RS485 communication module (FC4A-HPC3) next to the CPU module.

When using port 2 for RS485 communication on the slim type CPU module with the optional HMI module, install the RS485 communication adapter (FC4A-PC3) to the port 2 connector on the HMI base module (FC4A-HPH1).

Connect RS485 device to the RS485 terminals A, B, and SG of port 2 on the MicroSmart CPU module using a shielded twisted pair cable as shown below. The total length of the cable for the RS485 user communication can be extended up to 200 meters (656 feet).

RS485 User Communication System Setup



Programming WindLDR

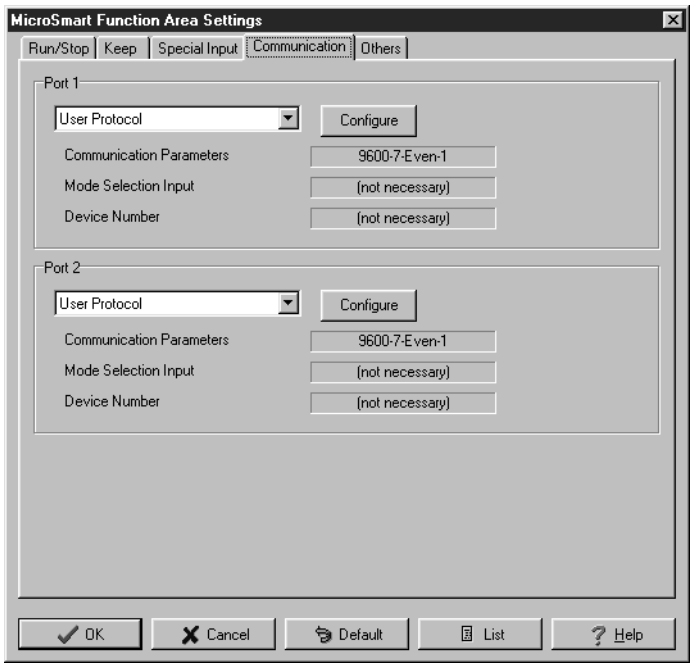
When using the user communication function to communicate with an external RS232C or RS485 device, set the communication parameters for the MicroSmart to match those of the external device.

Note: Since communication parameters in the Function Area Settings relate to the user program, the user program must be downloaded to the MicroSmart CPU module after changing any of these settings.

1. Select **Configure** from the WindLDR menu bar, then select **Function Area Settings**.

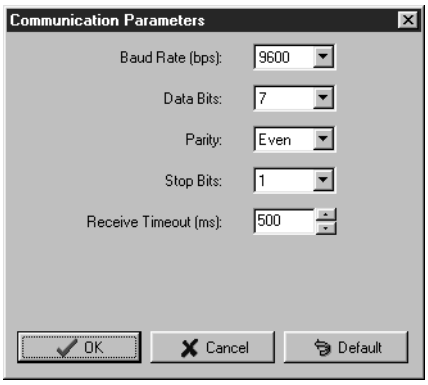
The Function Area Setting dialog box appears.

2. Click the **Communication** tab.



3. Select **User Protocol** in the Port 1 or Port 2 list box. (Click the **Configure** button when changing previous settings.)

The Communication Parameters dialog box appears.

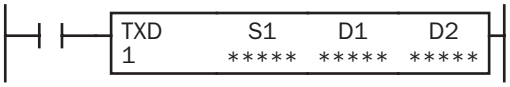


When **2550 ms** is selected in the Receive Timeout box, the receive timeout function is disabled.

4. Select communication parameters to the same values for the device to communicate with.

5. Click the **OK** button.

TXD1 (Transmit 1)

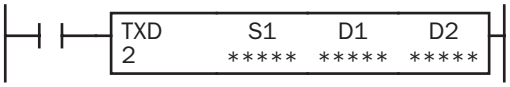


When input is on, data designated by S1 is converted into a specified format and transmitted through port 1 to a remote terminal with an RS232C port.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

TXD2 (Transmit 2)



When input is on, data designated by S1 is converted into a specified format and transmitted through port 2 to a remote terminal with an RS232C. Upgraded CPU modules can also use the RS485 port.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Transmit data	—	—	—	—	—	—	X	X	—
D1 (Destination 1)	Transmit completion output	—	X	▲	—	—	—	—	—	—
D2 (Destination 2)	Transmit status register	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

Transmit data designated by operand S1 can be a maximum of 200 bytes.

When transmission is complete, an output or internal relay, designated by operand D1, is turned on.

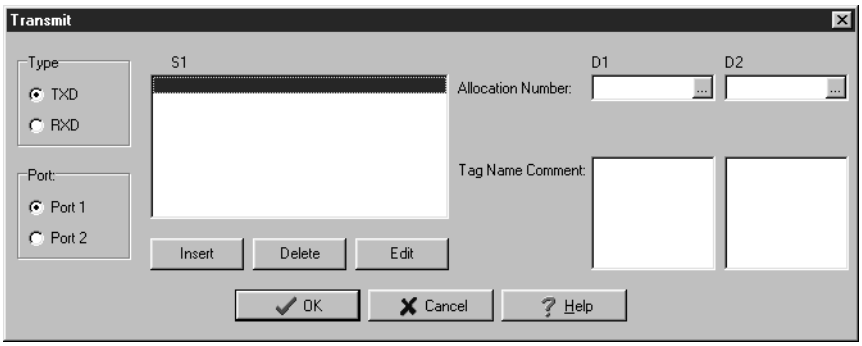
Destination 2 occupies two consecutive data registers starting with the operand designated by D2. The transmit status data register, D0 through D1298 or D2000 through D7998, stores the status of transmission and error code. The next data register stores the byte count of transmitted data. The same data registers can not be used as transmit status registers for TXD1/TXD2 instructions and receive status registers for RXD1/RXD2 instructions.

The TXD1/TXD2 instructions cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Precautions for Programming TXD Instruction

- The MicroSmart has five formatting areas each for executing TXD1 and TXD2 instructions, so five TXD1 and five TXD2 instructions can be processed at the same time. If inputs to more than five TXD1 or TXD2 instructions are turned on at the same time, an error code is set to the transmit status data register, designated by operand D2, in the excessive TXD instructions that cannot be executed.
- If the input for a TXD instruction is turned on while another TXD instruction is executed, the subsequent TXD instruction is executed 2 scan times after the preceding TXD instruction is completed.
- Since TXD instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

User Communication Transmit Instruction Dialog Box in WindLDR



Selections and Operands in Transmit Instruction Dialog Box

Type	**TXD**	Transmit instruction
	RXD	Receive instruction
Port	**Port 1**	Transmit user communication through port 1 (TXD1)
	Port 2	Transmit user communication through port 2 (TXD2)
S1	**Source 1**	Enter the data to transmit in this area. Transmit data can be constant values (character or hexadecimal), data registers, or BCC.
D1	**Destination 1**	Transmit completion output can be an output or internal relay.
D2	**Destination 2**	Transmit status register can be data register D0 through D1298 or D2000 through D7998. The next data register stores the byte count of transmitted data.

Transmit Data

Transmit data is designated by source operand S1 using constant values or data registers. BCC code can also be calculated automatically and appended to the transmit data. One TXD instruction can transmit 200 bytes of data at the maximum.

S1 (Source 1)

Transmit Data	Operand	Conversion Type	Transmit Digits (Bytes)	Repeat	BCC Calculation	Calculation Start Position
Constant	00h-7Fh (FFh)	No conversion	1	—	—	—
Data Register	D0-D1299 D2000-D7999	A: Binary to ASCII B: BCD to ASCII –: No conversion	1-4 1-5 1-2	1-99	—	—
BCC	—	A: Binary to ASCII –: No conversion	1-2	—	X: XOR A: ADD C: Add-2comp M: Modbus ASCII M: Modbus RTU	1-15

Designating Constant as S1

When a constant value is designated as source operand S1, one-byte data is transmitted without conversion. The valid transmit data value depends on the data bits selected in the Communication Parameters dialog box, which is called from **Configure > Fun Area Settings > Communication**, followed by selecting **User Protocol** in Port 1 or Port 2 list box and clicking the **Configure** button. When 7 data bits are selected as default, 00h through 7Fh is transmitted. When 8 data bits are selected, 00h through FFh is transmitted. Constant values are entered in character or hexadecimal notation into the source data.

Constant (Character)

Any character available on the computer keyboard can be entered. One character is counted as one byte.

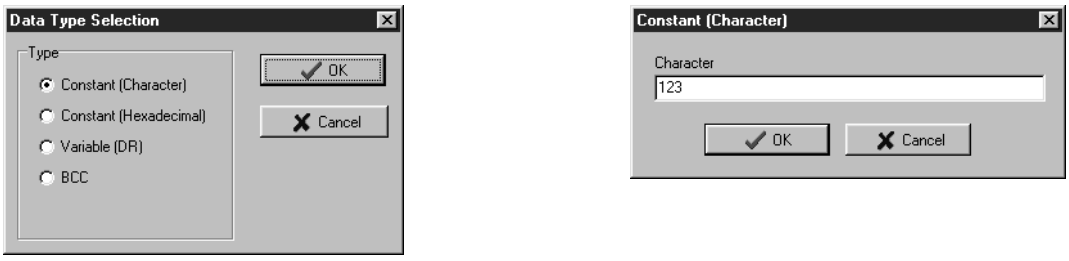
Constant (Hexadecimal)

Use this option to enter the hexadecimal code of any ASCII character. ASCII control codes NUL (00h) through US (1Fh) can also be entered using this option.

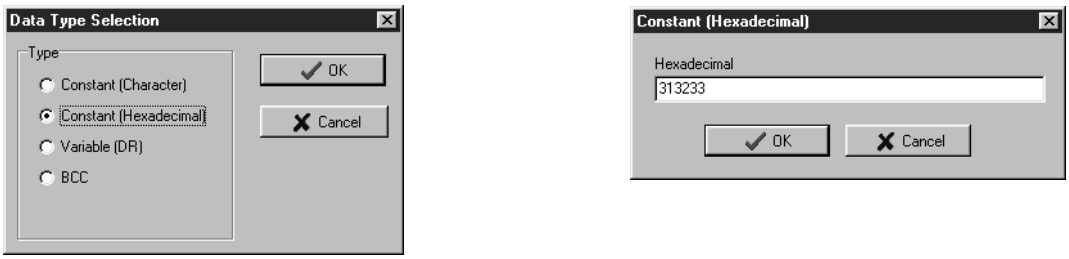
Example:

The following example shows two methods to enter 3-byte ASCII data "1" (31h), "2" (32h), "3" (33h).

(1) Constant (Character)



(2) Constant (Hexadecimal)



Designating Data Register as S1

When a data register is designated as source operand S1, conversion type and transmit digits must also be designated. The data stored in the designated data register is converted and a designated quantity of digits of the resultant data is transmitted. Conversion types are available in Binary to ASCII, BCD to ASCII, and no conversion.

When repeat is designated, data of data registers as many as the repeat cycles are transmitted, starting with the designated data register. Repeat cycles can be up to 99.

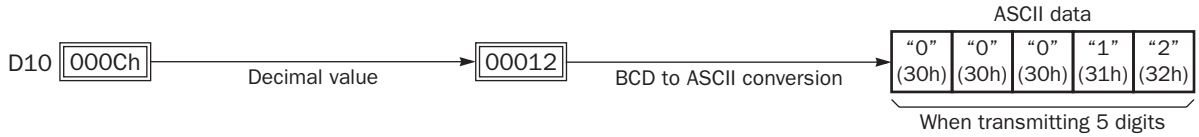
Conversion Type

The transmit data is converted according to the designated conversion type as described below:

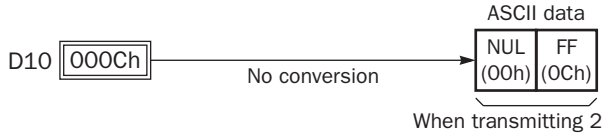
Example: D10 stores 000Ch (12)

(1) Binary to ASCII conversion

(2) BCD to ASCII conversion



(3) No conversion

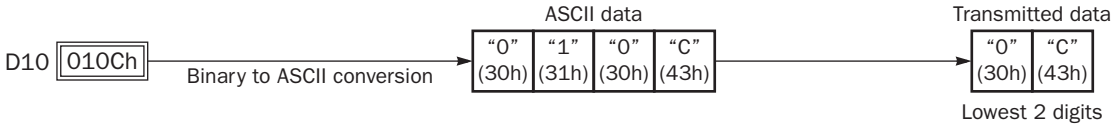


Transmit Digits (Bytes)

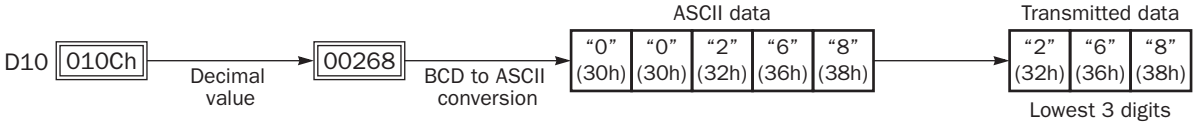
After conversion, the transmit data is taken out in specified digits. Possible digits depend on the selected conversion type.

Example: D10 stores 010Ch (268)

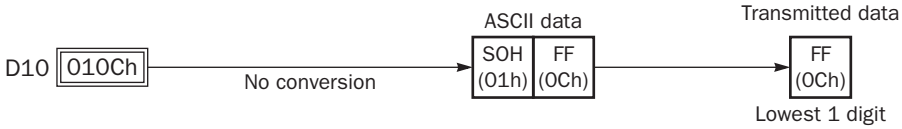
(1) Binary to ASCII conversion, Transmit digits = 2



(2) BCD to ASCII conversion, Transmit digits = 3



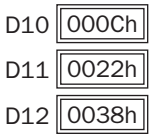
(3) No conversion, Transmit digits = 1



Repeat Cycles

When a data register is designated to repeat, consecutive data registers, as many as the repeat cycles, are used for transmit data in the same conversion type and transmit digits.

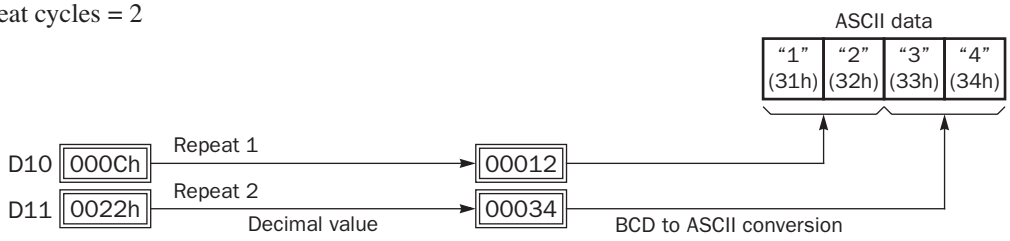
Example:



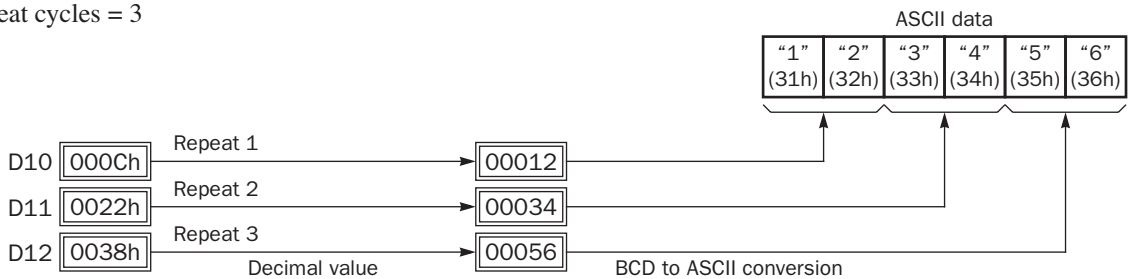
Data register No.: D10
Transmit digits: 2
Conversion type: BCD to ASCII

Data of data registers starting with D10 is converted in BCD to ASCII and is transmitted according to the designated repeat cycles.

(1) Repeat cycles = 2



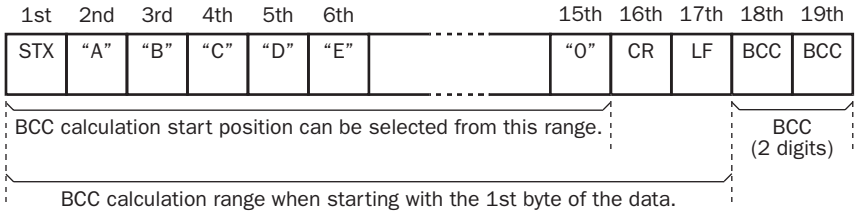
(2) Repeat cycles = 3



BCC (Block Check Character)

Block check characters can be appended to the transmit data. The start position for the BCC calculation can be selected from the first byte through the 15th byte. The BCC, calculated in either XOR or ADD, can be 1 or 2 digits.

Upgraded CPU modules can also use ADD-2comp, Modbus ASCII, and Modbus RTU to calculate the BCC.

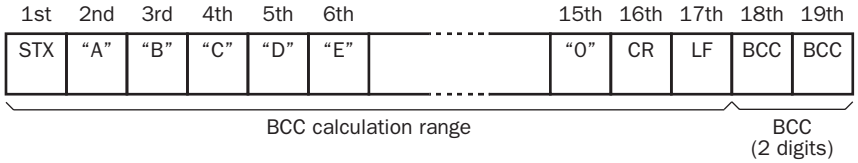


BCC Calculation Start Position

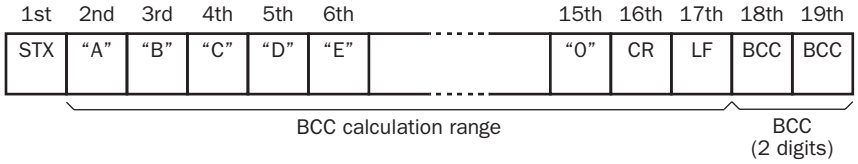
The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC is calculated for the range starting at the designated position up to the byte immediately before the BCC of the transmit data.

Example: Transmit data consists of 17 bytes plus 2 BCC digits.

(1) Calculation start position = 1



(2) Calculation start position = 2



BCC Calculation Formula

BCC calculation formula can be selected from XOR (exclusive OR) or ADD (addition) operation. ADD-2comp, Modbus ASCII, and Modbus RTU can also be selected for the upgraded CPU modules, using WindLDR ver. 4.40 or higher.

Example: Conversion results of transmit data consist of 41h, 42h, 43h, 44h, and 45h.

ASCII data

"A" (41h)	"B" (42h)	"C" (43h)	"D" (44h)	"E" (45h)

(1) BCC calculation formula = XOR

Calculation result = 41h ⊕ 42h ⊕ 43h ⊕ 44h ⊕ 45h = 41h

(2) BCC calculation formula = ADD

Calculation result = 41h + 42h + 43h + 44h + 45h = 14Fh → 4Fh (Only the last 1 or 2 digits are used as BCC.)

(3) BCC calculation formula = ADD-2comp

Calculation result = B1

(4) BCC calculation formula = Modbus ASCII

Calculation result = A8

(5) BCC calculation formula = Modbus RTU

Calculation result = 91h 50h

Conversion Type

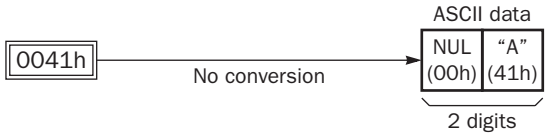
The BCC calculation result can be converted or not according to the designated conversion type as described below:

Example: BCC calculation result is 0041h.

(1) Binary to ASCII conversion

Note: On WindLDR, Modbus ASCII is defaulted to binary to ASCII conversion.

(2) No conversion



Note: On WindLDR, Modbus RTU is defaulted to no conversion.

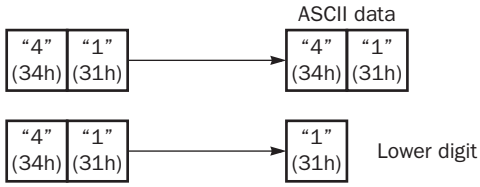
BCC Digits (Bytes)

The quantity of digits (bytes) of the BCC code can be selected from 1 or 2.

Example:

(1) BCC digits = 2

(2) BCC digits = 1



Note: On WindLDR, Modbus ASCII and Modbus RTU are defaulted to 2 digits.

Transmit Completion Output

Designate an output, Q0 through Q107, or an internal relay, M0 through M1277, as an operand for the transmit completion output. Special internal relays cannot be used.

When the start input for a TXD instruction is turned on, preparation for transmission is initiated, followed by data transmission. When a sequence of all transmission operation is complete, the designated output or internal relay is turned on.

Transmit Status

Designate a data register, D0 through D1298 or D2000 through D7998, as an operand to store the transmit status information including a transmission status code and a user communication error code.

Transmit Status Code

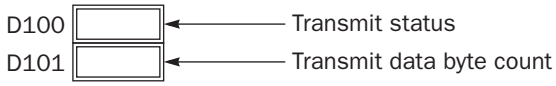
Transmit Status Code	Status	Description
16	Preparing transmission	From turning on the start input for a TXD instruction, until the transmit data is stored in the internal transmit buffer
32	Transmitting data	From enabling data transmission by an END processing, until all data transmission is completed
48	Data transmission complete	From completing all data transmission, until the END processing is completed for the TXD instruction
64	Transmit instruction complete	All transmission operation is completed and the next transmission is made possible

If the transmit status code is other than shown above, an error of transmit instruction is suspected. See User Communication Error Code on page 17-27.

Transmit Data Byte Count

The data register next to the operand designated for transmit status stores the byte count of data transmitted by the TXD instruction. When BCC is included in the transmit data, the byte count of the BCC is also included in the transmit data byte count.

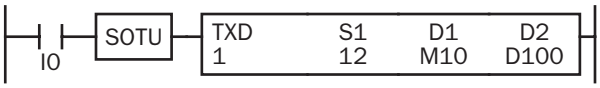
Example: Data register D100 is designated as an operand for transmit status.



Programming TXD Instruction Using WindLDR

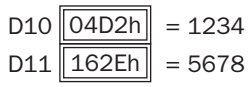
The following example demonstrates how to program a TXD instruction including a start delimiter, BCC, and end delimiter using WindLDR.

TXD sample program:

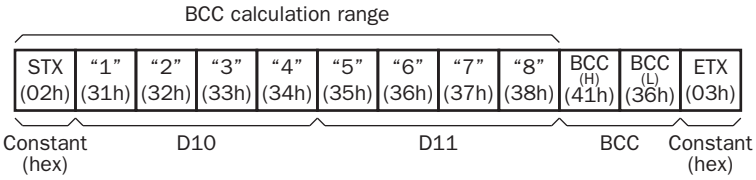


Communication port:	Port 1
Transmit completion output:	M10
Transmit status register:	D100
Transmit data byte count:	D101

Data register contents:

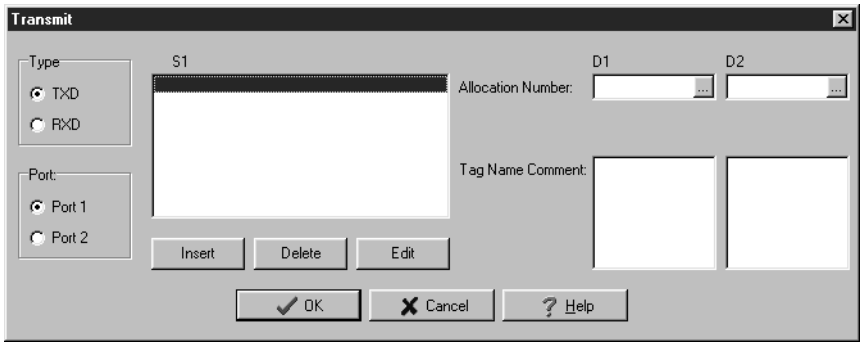


Transmit data example:



1. Start to program a TXD instruction. Move the cursor where you want to insert the TXD instruction, and type **TXD**. You can also insert the TXD instruction by clicking the User Communication icon in the menu bar and clicking where you want to insert the TXD instruction in the program edit area.

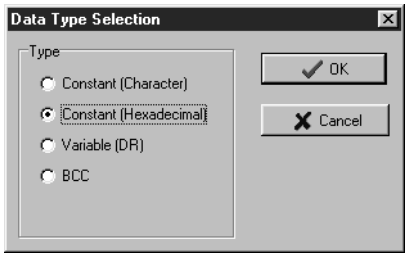
The Transmit instruction dialog box appears.

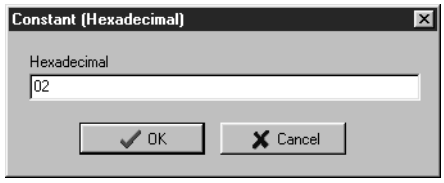


2. Check that **TXD** is selected in the Type box and click **Port 1** in the Port box. Then, click **Insert**.

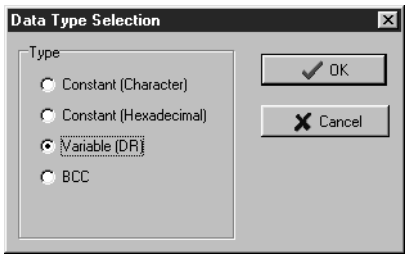
The Data Type Selection dialog box appears. You will program source operand S1 using this dialog box.

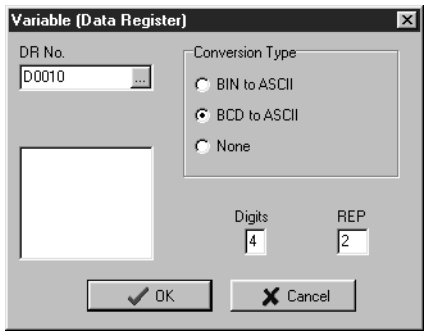
3. Click **Constant (Hexadecimal)** in the Type box and click **OK**. Next, in the Constant (Hexadecimal) dialog box, type **02** to program the start delimiter STX (02h). When finished, click **OK**.



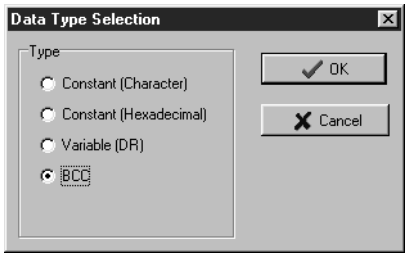


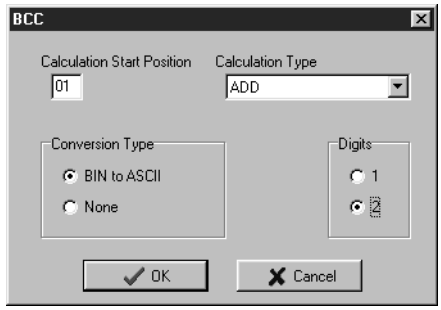
4. Since the Transmit instruction dialog box reappears, repeat the above procedure. In the Data Type Selection dialog box, click **Variable (DR)** and click **OK**. Next, in the Variable (Data Register) dialog box, type **D10** in the DR No. box and click **BCD to ASCII** to select the BCD to ASCII conversion. Enter **4** in the Digits box (4 digits) and **2** in the REP box (2 repeat cycles). When finished, click **OK**.



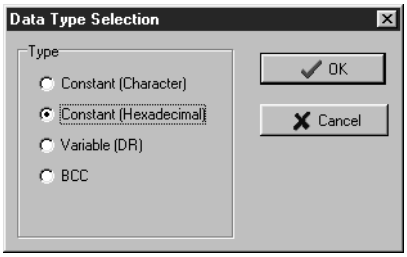


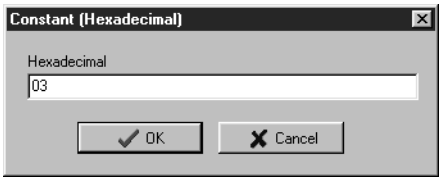
5. Again in the Data Type Selection dialog box, click **BCC** and click **OK**. Next, in the BCC dialog box, enter **1** in the Calculation Start Position box, click **ADD** for the Calculate Type, click **BIN to ASCII** for the Conversion Type, and click **2** for the Digits. When finished, click **OK**.



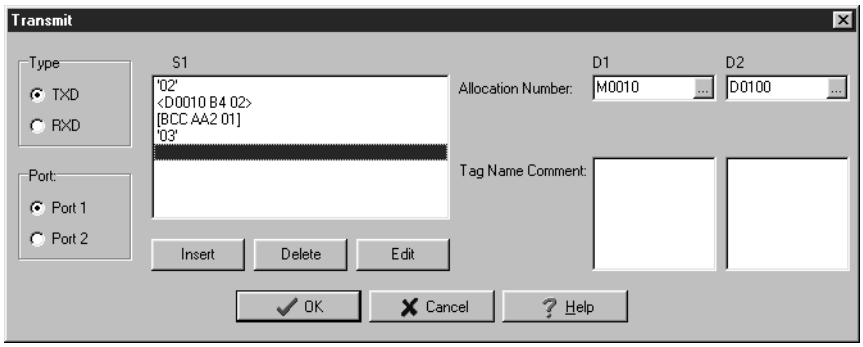


6. Once again in the Data Type Selection dialog box, click **Constant (Hexadecimal)** and click **OK**. Next, in the Constant (Hexadecimal) dialog box, type **03** to program the end delimiter ETX (03h). When finished, click **OK**.

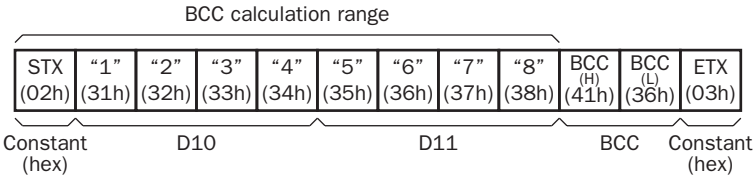




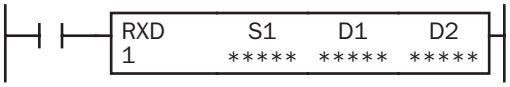
7. In the Transmit instruction dialog box, type **M10** in the destination D1 box and type **D100** in the destination D2 box. When finished, click **OK**.



Programming of the TXD1 instruction is complete and the transmit data is specified as follows:



RXD1 (Receive 1)

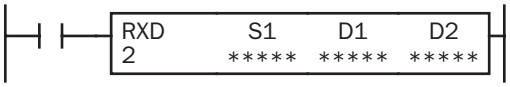


When input is on, data received through port 1 from a remote terminal is converted and stored in data registers according to the receive format designated by S1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

RXD2 (Receive 2)



When input is on, data received through port 2 from a remote terminal is converted and stored in data registers according to the receive format designated by S1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Receive format	—	—	—	—	—	—	X	X	—
D1 (Destination 1)	Receive completion output	—	X	▲	—	—	—	—	—	—
D2 (Destination 2)	Receive status	—	—	—	—	—	—	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

Receive format designated by operand S1 can be a maximum of 200 bytes.

When data receive is complete, an output or internal relay, designated by operand D1, is turned on.

Destination 2 occupies two consecutive data registers starting with the operand designated by D2. The receive status data register, D0 through D1298 or D2000 through D7998, stores the status of data receive and error code. The next data register stores the byte count of received data. The same data registers can not be used as transmit status registers for TXD1/TXD2 instructions and receive status registers for RXD1/RXD2 instructions.

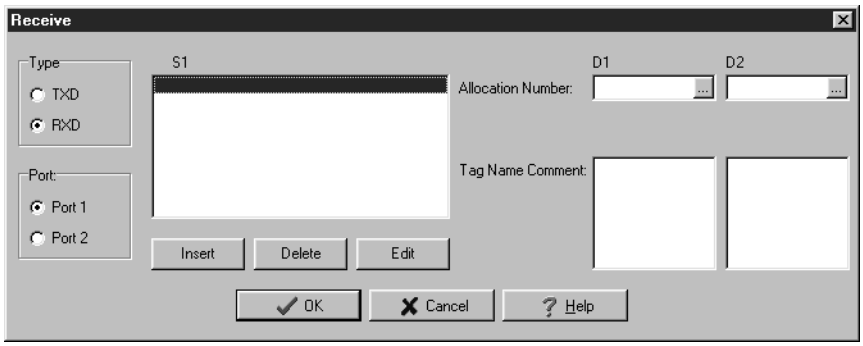
While RXD1/RXD2 instructions are ready for receiving data after a receive format is complete, turning on the user communication receive instruction cancel flag M8022 or M8023 cancels all RXD1/RXD2 instructions.

The RXD1/RXD2 instructions cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Precautions for Programming the RXD Instruction

- The MicroSmart can execute a maximum of five RXD1 and five RXD2 instructions that have a start delimiter at the same time. If a start delimiter is not programmed in RXD1/RXD2 instructions, the MicroSmart can execute only one RXD1 and one RXD2 instructions at a time. If the start input for a RXD1/RXD2 instruction is turned on while another RXD1/RXD2 instruction without a start delimiter is executed, a user communication error occurs.
- Since RXD instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.
- Once the input to the RXD instruction is turned on, the RXD is activated and ready for receiving incoming communication even after the input is turned off. When the RXD completes data receiving, the RXD is deactivated if the input to the RXD is off. Or, if the input is on, the RXD is made ready for receiving another communication. M8022/M8023 deactivate all RXD instructions waiting for incoming communication.

User Communication Receive Instruction Dialog Box in WindLDR



Selections and Operands in Receive Instruction Dialog Box

Type	**TXD**	Transmit instruction
	RXD	Receive instruction
Port	**Port 1**	Receive user communication through port 1 (RXD1)
	Port 2	Receive user communication through port 2 (RXD2)
S1	**Source 1**	Enter the receive format in this area. The receive format can include a start delimiter, data register to store incoming data, end delimiter, BCC, and skip.
D1	**Destination 1**	Receive completion output can be an output or internal relay.
D2	**Destination 2**	Receive status register can be data register D0 through D1298 or D2000 through D7998. The next data register stores the byte count of received data.

Receive Format

Receive format, designated by source operand S1, specifies data registers to store received data, data digits for storing data, data conversion type, and repeat cycles. A start delimiter and an end delimiter can be included in the receive format to discriminate valid incoming communication. When some characters in the received data are not needed, "skip" can be used to ignore a specified number of characters. BCC code can also be appended to the receive format to verify the received data. One RXD instruction can receive 200 bytes of data at the maximum.

S1 (Source 1)

Receive Format	Operand	Receive Digits (Bytes)	Conversion Type	Repeat	BCC Calculation	Calculation Start Position	Skip Bytes
Data Register	D0-D1299 D2000-D7999	1-4 1-5 1-2	A: ASCII to Binary B: ASCII to BCD –: No conversion	1-99	—	—	—
Start Delimiter	00h-7Fh (FFh)	—	No conversion	—	—	—	—
End Delimiter	00h-7Fh (FFh)	—	No conversion	—	—	—	—
BCC	—	1-2	A: Binary to ASCII –: No conversion	—	X: XOR A: ADD C: Add-2comp M: Modbus ASCII M: Modbus RTU	1-15	—
Skip	—	—	—	—	—	—	1-99

Designating Data Register as S1

When a data register is designated as source operand S1, receive digits and conversion type must also be designated. The received data is divided into a block of specified receive digits, converted in a specified conversion type, and stored to the designated data register. Conversion types are available in ASCII to Binary, ASCII to BCD, and no conversion.

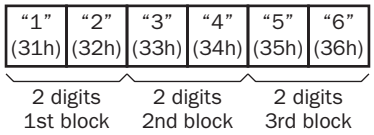
When repeat is designated, received data is divided, converted, and stored into data registers as many as the repeat cycles, starting with the designated data register. Repeat cycles can be up to 99.

Receive Digits

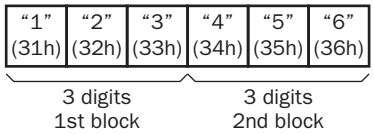
The received data is divided into a block of specified receive digits before conversion as described below:

Example: Received data of 6 bytes are divided in different receive digits. (Repeat is also designated.)

(1) Receive digits = 2



(2) Receive digits = 3

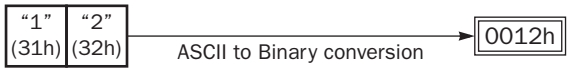


Conversion Type

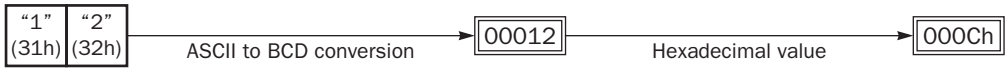
The data block of the specified receive digits is then converted according to the designated conversion type as described below:

Example: Received data has been divided into a 2-digit block.

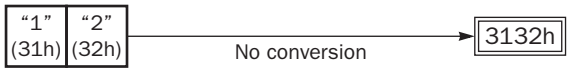
(1) ASCII to Binary conversion



(2) ASCII to BCD conversion



(3) No conversion

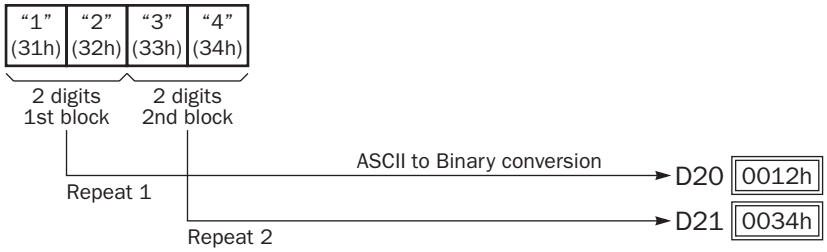


Repeat Cycles

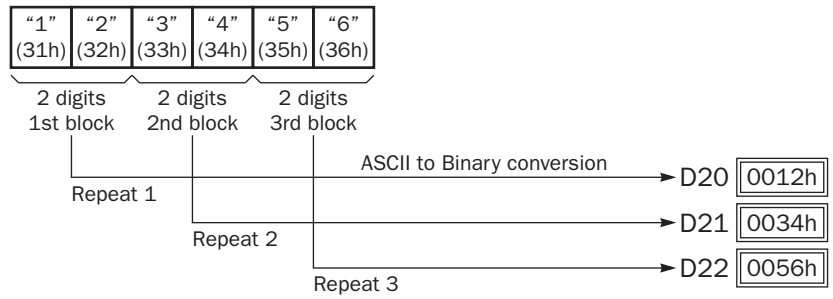
When a data register is designated to repeat, the received data is divided and converted in the same way as specified, and the converted data is stored to consecutive data registers as many as the repeat cycles.

Example: Received data of 6 bytes is divided into 2-digit blocks, converted in ASCII to Binary, and stored to data registers starting at D20.

(1) Repeat cycles = 2



(2) Repeat cycles = 3



Designating Constant as Start Delimiter

A start delimiter can be programmed at the first byte in the receive format of a RXD1/RXD2 instruction; the MicroSmart will recognize the beginning of valid communication, although a RXD1/RXD2 instruction without a start delimiter can also be executed.

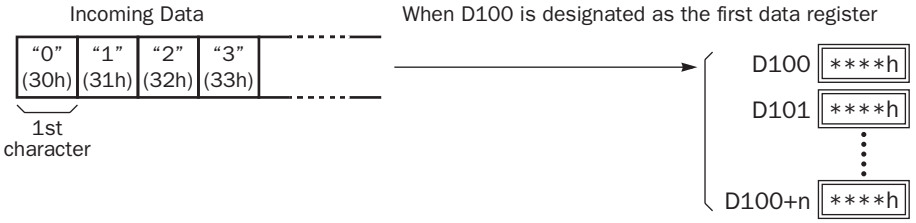
When a constant value is designated at the first byte of source operand S1, the one-byte data serves as a start delimiter to start the processing of the received data. The valid start delimiter value depends on the data bits selected in the **Communication Parameters** dialog box, which is called from **Configure > Fun Area Settings > Communication**, followed by selecting **User Protocol** in **Port 1** or **Port 2** list box and clicking the **Configure** button. When 7 data bits are selected as default, start delimiters can be 00h through 7Fh. When 8 data bits are selected, start delimiters can be 00h through FFh. Constant values are entered in character or hexadecimal notation into the source data.

A maximum of five RXD1 and five RXD2 instructions with different start delimiters can be executed at the same time. When the first byte of the incoming data matches the start delimiter of a RXD1/RXD2 instruction, the received data is processed and stored according to the receive format specified in the RXD1/RXD2 instruction. If the first byte of the incoming data does not match the start delimiter of any RXD1/RXD2 instruction that is executed, the MicroSmart discards the incoming data and waits for the next communication.

While a RXD1/RXD2 instruction without a start delimiter is executed, any incoming data is processed continuously according to the receive format. Only one RXD1 and one RXD2 instructions without a start delimiter can be executed at a time. If start inputs to two or more RXD1/RXD2 instructions without a start delimiter are turned on simultaneously, one at the smallest address is executed and the corresponding completion output is turned on.

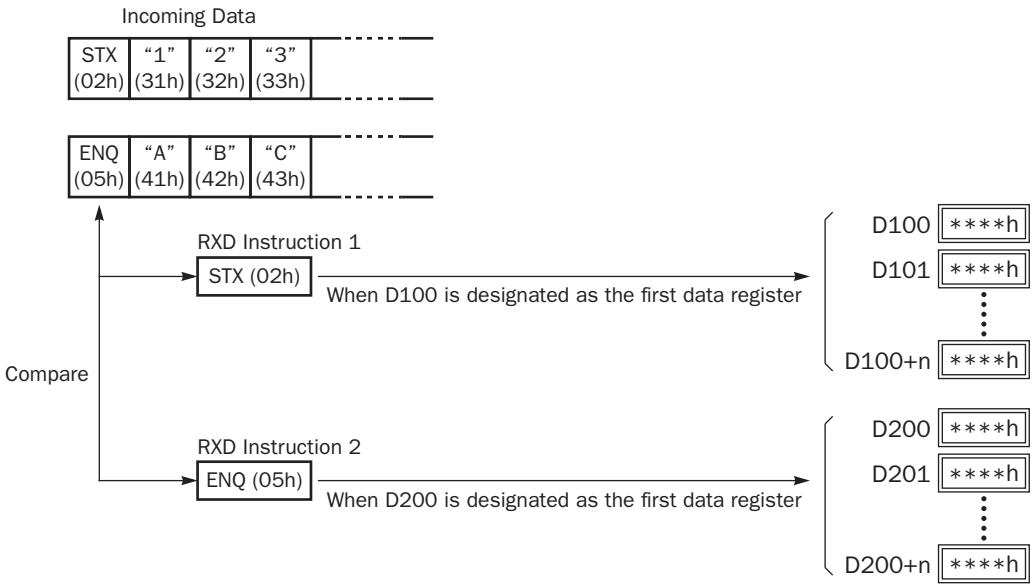
Example:

(1) When a RXD1/RXD2 instruction without a start delimiter is executed



The incoming data is divided, converted, and stored to data registers according to the receive format.

(2) When RXD1/RXD2 instructions with start delimiters STX (02h) and ENQ (05h) are executed



The incoming data is divided, converted, and stored to data registers according to the receive format. Start delimiters are not stored to data registers.

Designating Constant as End Delimiter

An end delimiter can be programmed at other than the first byte in the receive format of a RXD instruction; the MicroSmart will recognize the end of valid communication, although RXD instructions without an end delimiter can also be executed.

When a constant value is designated at other than the first byte of source operand S1, the one- or multiple-byte data serves as an end delimiter to end the processing of the received data. The valid end delimiter value depends on the data bits selected in the Communication Parameters dialog box, which is called from **Configure > Fun Area Settings > Communication**, followed by selecting **User Protocol** in Port 1 or Port 2 list box and clicking the **Configure** button. When 7 data bits are selected as default, end delimiters can be 00h through 7Fh. When 8 data bits are selected, end delimiters can be 00h through FFh. Constant values are entered in character or hexadecimal notation into the source data.

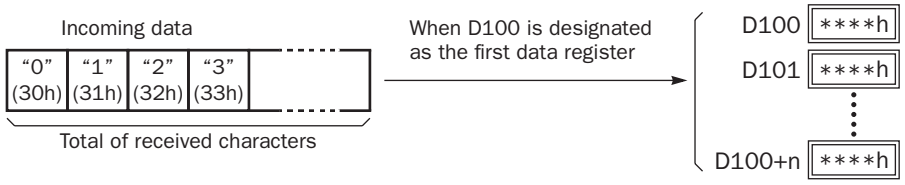
If a character in incoming data matches the end delimiter, the RXD instruction ends receiving data at this point and starts subsequent receive processing as specified. Even if a character matches the end delimiter at a position earlier than expected, the RXD instruction ends receiving data there.

If a BCC code is included in the receive format of a RXD instruction, an end delimiter can be positioned immediately before or after the BCC code. If a data register or skip is designated between the BCC and end delimiter, correct receiving is not ensured.

When a RXD instruction without an end delimiter is executed, data receiving ends when the specified bytes of data in the receive format, such as data registers and skips, have been received. In addition, data receiving also ends when the interval between incoming data characters exceeds the receive timeout value specified in the Communication Parameters dialog box whether the RXD has an end delimiter or not. The character interval timer is started when the first character of incoming communication is received and restarted each time the next character is received. When a character is not received within a predetermined period of time, timeout occurs and the RXD ends data receive operation.

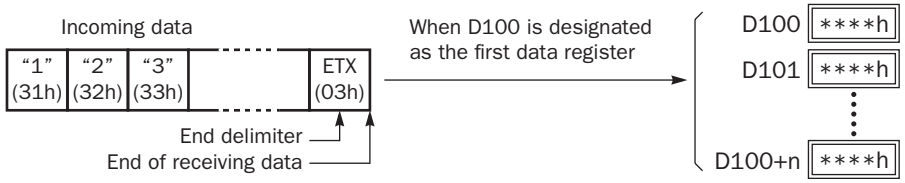
Example:

(1) When a RXD instruction without an end delimiter is executed



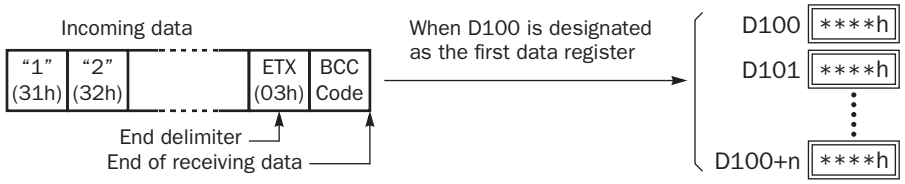
The incoming data is divided, converted, and stored to data registers according to the receive format.
Receive operation is completed when the total characters programmed in RXD are received.

(2) When a RXD instruction with end delimiter ETX (03h) and without BCC is executed



The incoming data is divided, converted, and stored to data registers according to the receive format.
The end delimiter is not stored to a data register.
Any data arriving after the end delimiter is discarded.

(3) When a RXD instruction with end delimiter ETX (03h) and one-byte BCC is executed

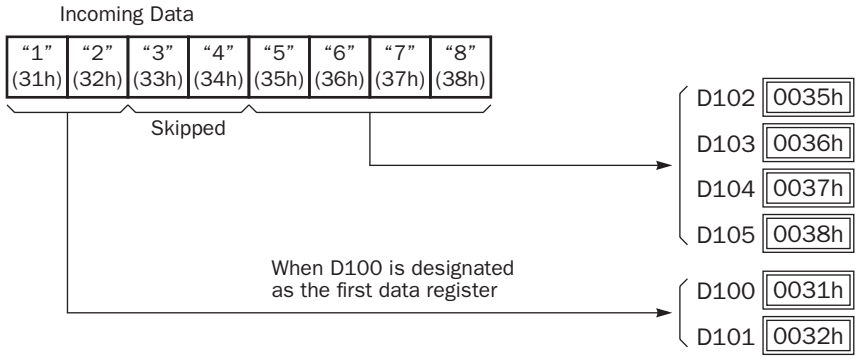


The incoming data is divided, converted, and stored to data registers according to the receive format.
The end delimiter and BCC code are not stored to data registers.
After receiving the end delimiter, the MicroSmart receives only the one-byte BCC code.

Skip

When "skip" is designated in the receive format, a specified quantity of digits in the incoming data are skipped and not stored to data registers. A maximum of 99 digits (bytes) of characters can be skipped continuously.

Example: When a RXD instruction with skip for 2 digits starting at the third byte is executed



BCC (Block Check Character)

The MicroSmart has an automatic BCC calculation function to detect a communication error in incoming data. If a BCC code is designated in the receive format of a RXD instruction, the MicroSmart calculates a BCC value for a specified starting position through the position immediately preceding the BCC and compares the calculation result with the BCC code in the received incoming data. The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC, calculated in either XOR or ADD, can be 1 or 2 digits.

Upgraded CPU modules can also use ADD-2comp, Modbus ASCII, and Modbus RTU to calculate the BCC.

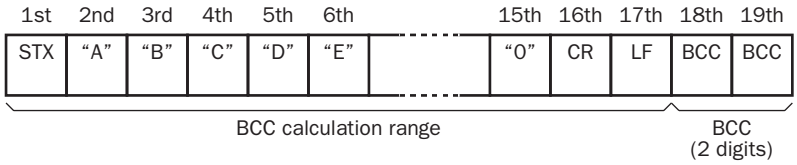
When an end delimiter is not used in the RXD instruction, the BCC code must be positioned at the end of the receive format designated in Source 1 operand. When an end delimiter is used, the BCC code must be immediately before or after the end delimiter. The MicroSmart reads a specified number of BCC digits in the incoming data according to the receive format to calculate and compare the received BCC code with the BCC calculation results.

BCC Calculation Start Position

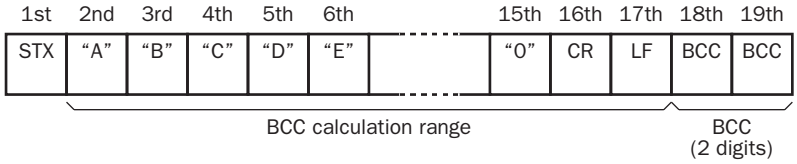
The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC is calculated for the range starting at the designated position up to the byte immediately before the BCC of the receive data.

Example: Received data consists of 17 bytes plus 2 BCC digits.

(1) Calculation start position = 1



(2) Calculation start position = 2



BCC Calculation Formula

BCC calculation formula can be selected from XOR (exclusive OR) or ADD (addition) operation. ADD-2comp, Modbus ASCII, and Modbus RTU can also be selected for the upgraded CPU modules.

Example: Incoming data consists of 41h, 42h, 43h, 44h, and 45h.

(1) BCC Calculation formula = XOR

Calculation result = 41h ⊕ 42h ⊕ 43h ⊕ 44h ⊕ 45h = 41h

(2) BCC Calculation formula = ADD

Calculation result = 41h + 42h + 43h + 44h + 45h = 14Fh → 4Fh (Only the last 1 or 2 digits are used as BCC.)

(3) BCC calculation formula = ADD-2comp

Calculation result = B1

(4) BCC calculation formula = Modbus ASCII

Calculation result = A8

(5) BCC calculation formula = Modbus RTU

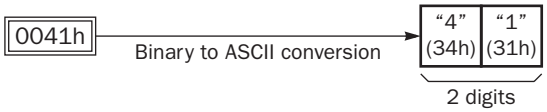
Calculation result = 91h 50h

Conversion Type

The BCC calculation result can be converted or not according to the designated conversion type as described below:

Example: BCC calculation result is 0041h.

(1) Binary to ASCII conversion



Note: On WindLDR, Modbus ASCII is defaulted to binary to ASCII conversion.

(2) No conversion

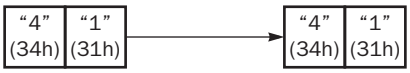
Note: On WindLDR, Modbus RTU is defaulted to no conversion.

BCC Digits (Bytes)

The quantity of digits (bytes) of the BCC code can be selected from 1 or 2.

Example:

(1) BCC digits = 2



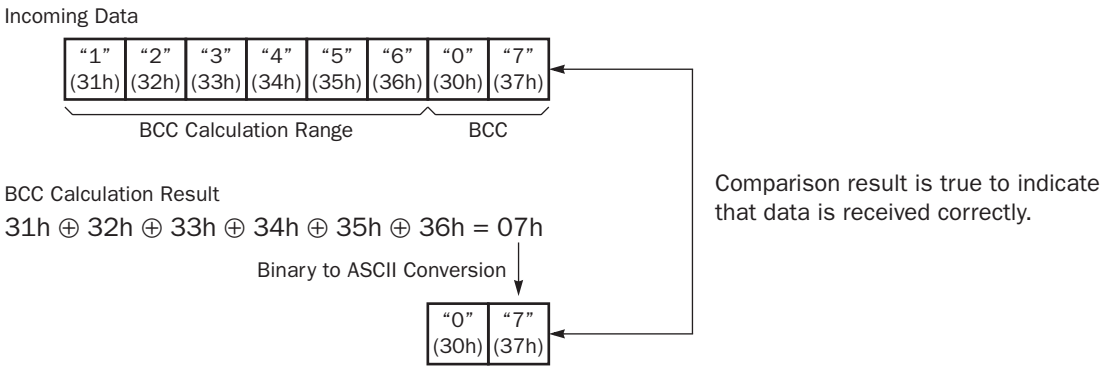
(2) BCC digits = 1

Note: On WindLDR, Modbus ASCII and Modbus RTU are defaulted to 2 digits.

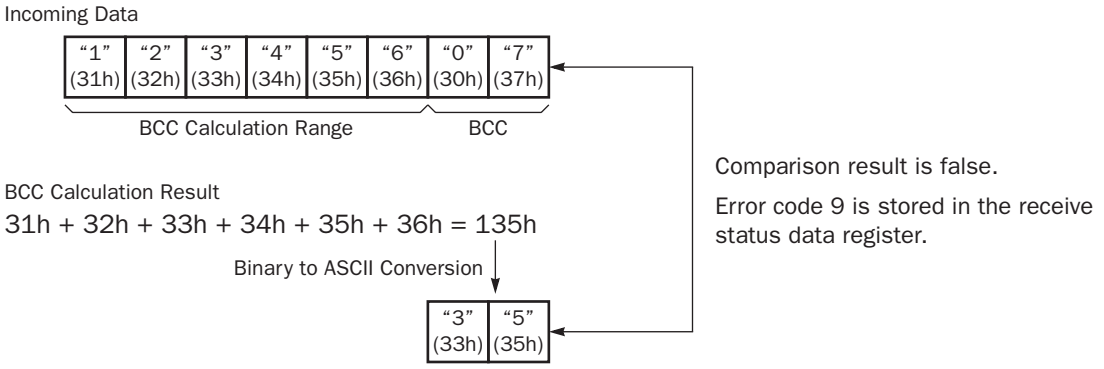
Comparing BCC Codes

The MicroSmart compares the BCC calculation result with the BCC code in the received incoming data to check for any error in the incoming communication due to external noises or other causes. If a disparity is found in the comparison, an error code is stored in the data register designated as receive status in the RXD instruction. For user communication error code, see page 17-27.

Example 1: BCC is calculated for the first byte through the sixth byte using the XOR format, converted in binary to ASCII, and compared with the BCC code appended to the seventh and eighth bytes of the incoming data.



Example 2: BCC is calculated for the first byte through the sixth byte using the ADD format, converted in binary to ASCII, and compared with the BCC code appended to the seventh and eighth bytes of the incoming data.



Receive Completion Output

Designate an output, Q0 through Q107, or internal relay, M0 through M1277, as an operand for the receive completion output.

When the start input for a RXD instruction is turned on, preparation for receiving data is initiated, followed by data conversion and storage. When a sequence of all data receive operation is complete, the designated output or internal relay is turned on.

Conditions for Completion of Receiving Data

After starting to receive data, the RXD instruction can be completed in three ways:

- When an end delimiter is received (except when a BCC exists immediately after the end delimiter).
- When receive timeout occurs.
- When a specified byte count of data has been received.

Data receiving is completed when one of the above three conditions is met. To abort a RXD instruction, use the user communication receive instruction cancel flag M8022 or M8023. See page 17-24.

Receive Status

Designate a data register, D0 through D1298 or D2000 through D7998, as an operand to store the receive status information including a receive status code and a user communication error code.

Receive Status Code

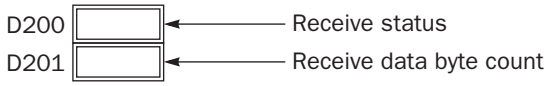
Receive Status Code	Status	Description
16	Preparing data receive	From turning on the start input for a RXD instruction to read the receive format, until the RXD instruction is enabled by an END processing
32	Receiving data	From enabling the RXD instruction by an END processing, until incoming data is received
48	Data receive complete	From receiving incoming data, until the received data is converted and stored in data registers according to the receive format
64	Receive instruction complete	All data receive operation is completed and the next data receive is made possible
128	User communication receive instruction cancel flag active	RXD instructions are cancelled by special internal relay M8022 or M8023

If the receive status code is other than shown above, an error of receive instruction is suspected. See User Communication Error Code on page 17-27.

Receive Data Byte Count

The data register next to the operand designated for receive status stores the byte count of data received by the RXD instruction. When a start delimiter, end delimiter, and BCC are included in the received data, the byte counts for these codes are also included in the receive data byte count.

Example: Data register D200 is designated as an operand for receive status.



User Communication Receive Instruction Cancel Flag M8022/M8023

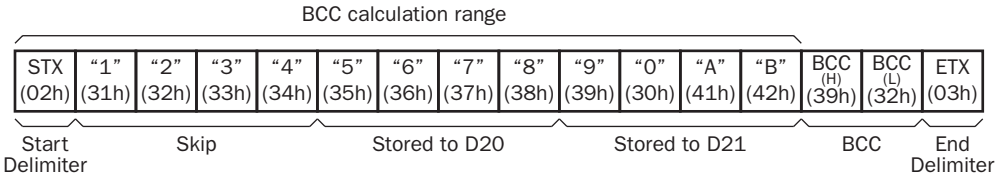
Special internal relays M8022 and M8023 are used to cancel all RXD1 and RXD2 instructions, respectively. While the MicroSmart has completed receive format and is ready for receiving incoming data, turning on M8022 or M8023 cancels all receive instructions for port 1 or port 2, respectively. This function is useful to cancel receive instructions only, without stopping the MicroSmart.

To make the cancelled RXD instructions active, turn off the flag and turn on the input to the RXD instruction again.

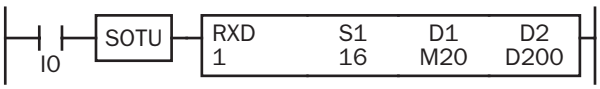
Programming RXD Instruction Using WindLDR

The following example demonstrates how to program a RXD instruction including a start delimiter, skip, BCC, and end delimiter using WindLDR. Converted data is stored to data registers D20 and D21. Internal relay M20 is used as destination D1 for the receive completion output. Data register D200 is used as destination D2 for the receive status, and data register D201 is used to store the receive data byte count.

Receive data example:



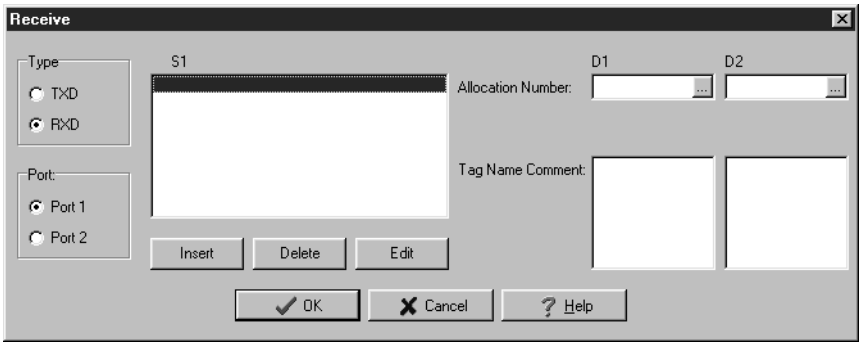
RXD sample program:



Communication port:	Port 1
Receive completion output:	M20
Receive status register:	D200
Receive data byte count:	D201

1. Start to program a RXD instruction. Move the cursor where you want to insert the RXD instruction, and type **RXD**. You can also insert the RXD instruction by clicking the User Communication icon in the menu bar and clicking where you want to insert the RXD instruction in the program edit area, then the Transmit dialog box appears. Click **RXD** to change the dialog box to the Receive dialog box.

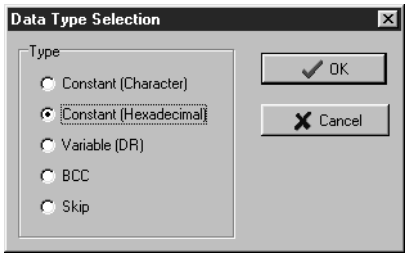
The Receive instruction dialog box appears.

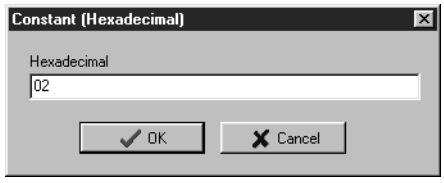


2. Check that **RXD** is selected in the Type box and click **Port 1** in the Port box. Then, click **Insert**.

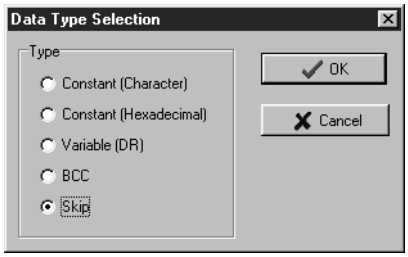
The Data Type Selection dialog box appears. You will program source operand S1 using this dialog box.

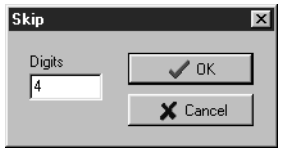
3. Click **Constant (Hexadecimal)** in the Type box and click **OK**. Next, in the Constant (Hexadecimal) dialog box, type **02** to program the start delimiter STX (02h). When finished, click **OK**.



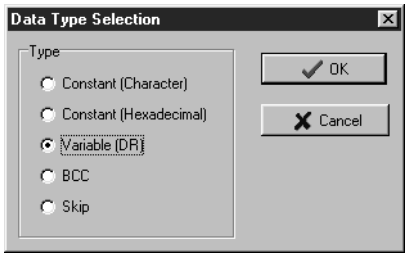


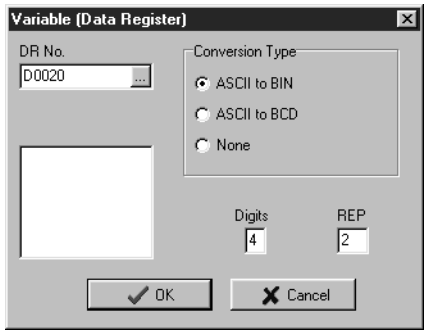
4. Since the Receive instruction dialog box reappears, repeat the above procedure. In the Data Type Selection dialog box, click **Skip** and click **OK**. Next, in the Skip dialog box, type **4** in the Digits box and click **OK**.



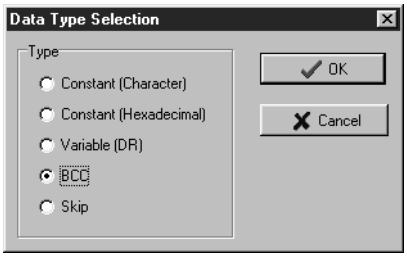


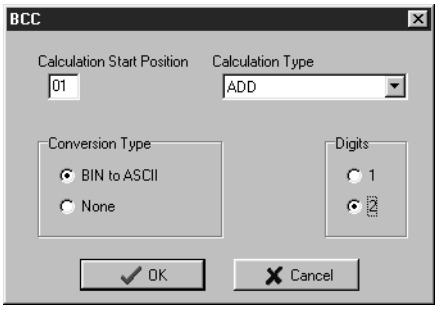
5. Again in the Data Type Selection dialog box, click **Variable (DR)** and click **OK**. Next, in the Variable (Data Register) dialog box, type **D20** in the DR No. box and click **ASCII to BIN** to select ASCII to binary conversion. Enter **4** in the Digits box (4 digits) and **2** in the REP box (2 repeat cycles). When finished, click **OK**.



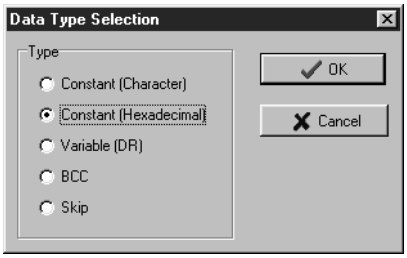


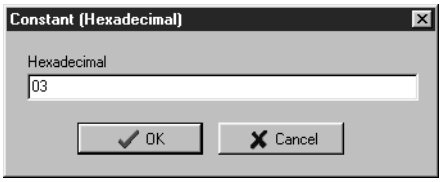
6. Again in the Data Type Selection dialog box, click **BCC** and click **OK**. Next, in the BCC dialog box, enter **1** in the Calculation Start Position box, click **ADD** for the Calculation Type, click **BIN to ASCII** for the Conversion Type, and click **2** for the Digits. When finished, click **OK**.



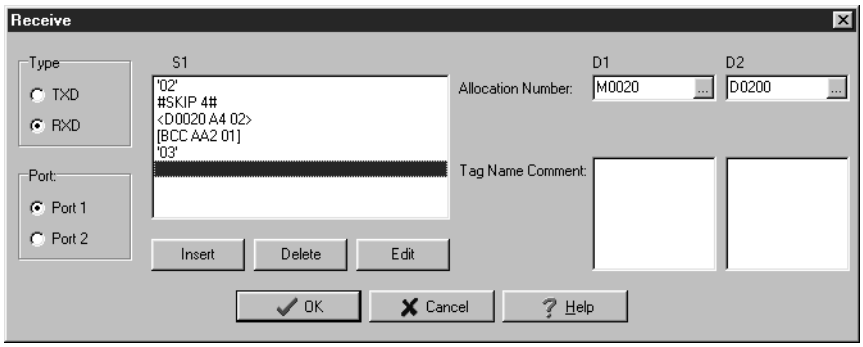


7. Once again in the Data Type Selection dialog box, click **Constant (Hexadecimal)** and click **OK**. Next, in the Constant (Hexadecimal) dialog box, type **03** to program the end delimiter ETX (03h). When finished, click **OK**.





8. In the Receive instruction dialog box, type **M20** in the destination D1 box and type **D200** in the destination D2 box. When finished, click **OK**.



Programming of the RXD1 instruction is complete and the receive data will be stored as follows:

D20 | 5678h | = 22136
D21 | 90ABh | = 37035

User Communication Error

When a user communication error occurs, a user communication error code is stored in the data register designated as a transmit status in the TXD instruction or as a receive status in the RXD instruction. When multiple errors occur, the final error code overwrites all preceding errors and is stored in the status data register.

The status data register also contains transmit/receive status code. To extract a user communication error code from the status data register, divide the value by 16. The remainder is the user communication error code. See pages 17-11 and 17-23.

To correct the error, correct the user program by referring to the error causes described below:

User Communication Error Code

User Communication Error Code	Error Cause	Transmit/Receive Completion Output
1	Start inputs to more than 5 TXD instructions are on simultaneously.	Transmit completion outputs of the first 5 TXD instructions from the top of the ladder diagram are turned on.
2	Transmission destination busy timeout	Goes on after busy timeout.
3	Start inputs to more than 5 RXD instructions with a start delimiter are on simultaneously.	Among the first 5 RXD instructions from the top of the ladder diagram, receive completion outputs of RXD instructions go on if the start delimiter matches the first byte of the received data.
4	While a RXD instruction without a start delimiter is executed, another RXD instruction with or without a start delimiter is executed.	The receive completion output of the RXD instruction at a smaller address goes on.
5	— Reserved —	—
6	— Reserved —	—
7	The first byte of received data does not match the specified start delimiter.	No effect on the receive completion output. If incoming data with a matching start delimiter is received subsequently, the receive completion output goes on.
8	When ASCII to binary or ASCII to BCD conversion is specified in the receive format, any code other than 0 to 9 and A to F is received. (These codes are regarded as 0 during conversion.)	The receive completion output goes on.
9	BCC calculated from the RXD instruction does not match the BCC appended to the received data.	The receive completion output goes on.
10	The end delimiter code specified in the RXD instruction does not match the received end delimiter code.	The receive completion output goes on.
11	Receive timeout between characters (After receiving one byte of data, the next byte is not received in the period specified for the receive timeout value.)	The receive completion output goes on.
12	Overrun error (Before the receive processing is completed, the next data is received.)	The receive completion output goes off.
13	Framing error (Detection error of start bit or stop bit)	No effect on the completion output.
14	Parity check error (Error is found in the parity check.)	No effect on the completion output.
15	TXD1/RXD1 (or TXD2/RXD2) instruction is executed while user protocol is not selected for port 1 (or port 2) in the Function Area Settings.	No effect on the completion output.

ASCII Character Code Table

Upper Bit / Lower Bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	$^{N}U_{L}$	$^{D}L_{E}$	SP	0	@	P	`	p								
Decimal	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	$^{S}O_{H}$	$^{D}C_{1}$	!	1	A	Q	a	q								
Decimal	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	$^{S}T_{X}$	$^{D}C_{2}$	”	2	B	R	b	r								
Decimal	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	$^{E}T_{X}$	$^{D}C_{3}$	#	3	C	S	c	s								
Decimal	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	$^{E}O_{T}$	$^{D}C_{4}$	$	4	D	T	d	t								
Decimal	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	$^{E}N_{Q}$	$^{N}A_{K}$	%	5	E	U	e	u								
Decimal	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	$^{A}C_{K}$	$^{S}Y_{N}$	&	6	F	V	f	v								
Decimal	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	$^{B}E_{L}$	$^{E}T_{B}$	’	7	G	W	g	w								
Decimal	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	BS	$^{C}A_{N}$	(	8	H	X	h	x								
Decimal	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9	HT	EM	)	9	I	Y	i	y								
Decimal	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A	LF	$^{S}U_{B}$	*	:	J	Z	j	z								
Decimal	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	VT	$^{E}S_{C}$	+	;	K	[	k	{								
Decimal	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C	FF	FS	,	<	L	\	l	|								
Decimal	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D	CR	GS	-	=	M	]	m	}								
Decimal	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E	SO	RS	.	>	N	^	n	~								
Decimal	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F	SI	US	/	?	O	_	o									
Decimal	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

RS232C Line Control Signals

While the MicroSmart is in the user communication mode, special data registers can be used to enable or disable DSR and DTR control signal options for port 2. Port 2 is available on the 16- and 24-I/O type CPU modules only, and an optional RS232C adapter must be installed on the port 2 connector to enable the RS232C communication. The DSR and DTR control signal options cannot be used for port 1.

The RTS signal line of port 2 remains on.

Special Data Registers for Port 2 RS232C Line Control Signals

Special data registers D8104 through D8106 are allocated for RS232C line control signals.

RS232C Port	DR No.	Data Register Function	DR Value Updated	R/W
Port 2	D8104	Control signal status	Every scan	R
	D8105	DSR input control signal option	When sending/receiving data	R/W
	D8106	DTR output control signal option	When sending/receiving data	R/W

Control Signal Status D8104

Special data register D8104 stores a value to show that DSR and DTR are on or off at port 2. The data of D8104 is updated at every END processing.

D8104 Value	DTR	DSR	Description
0	OFF	OFF	Both DSR and DTR are off
1	OFF	ON	DSR is on
2	ON	OFF	DTR is on
3	ON	ON	Both DSR and DTR are on

DSR Control Signal Status in RUN and STOP Modes

Communication Mode	D8105 Value	DSR (Input) Status	
		RUN Mode	STOP Mode
User Communication Mode	0 (default)	No effect	No effect (TXD/RXD disabled)
	1	ON: Enable TXD/RXD OFF: Disable TXD/RXD	No effect (TXD/RXD disabled)
	2	ON: Disable TXD/RXD OFF: Enable TXD/RXD	No effect (TXD/RXD disabled)
	3	ON: Enable TXD OFF: Disable TXD	No effect (TXD/RXD disabled)
	4	ON: Disable TXD OFF: Enable TXD	No effect (TXD/RXD disabled)
	5 or more	No effect	No effect (TXD/RXD disabled)
Maintenance Mode	—	No effect	No effect

DTR Control Signal Status in RUN and STOP Modes

Communication Mode	D8106 Value	DTR (Output) Status	
		RUN Mode	STOP Mode
User Communication Mode	0 (default)	ON	OFF
	1	OFF	OFF
	2	RXD enabled: ON RXD disabled: OFF	OFF
	3 or more	ON	OFF
Maintenance Mode	—	ON	ON

DSR Input Control Signal Option D8105

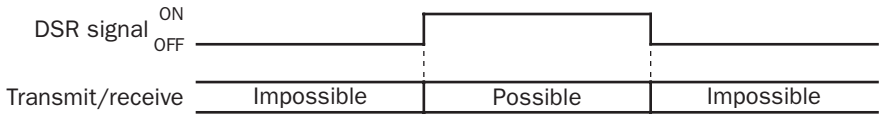
Special data register D8105 is used to control data flow between the MicroSmart RS232C port 2 and the remote terminal depending on the DSR (data set ready) signal sent from the remote terminal. The DSR signal is an input to the MicroSmart to determine the status of the remote terminal. The remote terminal informs the MicroSmart using DSR whether the remote terminal is ready for receiving data or is sending valid data.

The DSR control signal option can be used only for the user communication through the RS232C port 2.

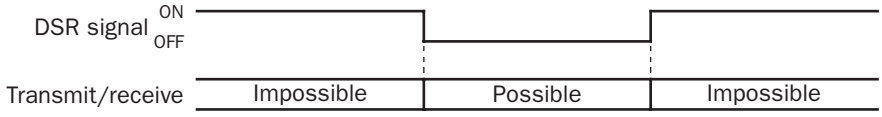
D8105 = 0 (system default):

DSR is not used for data flow control. When DSR control is not needed, set 0 to D8105.

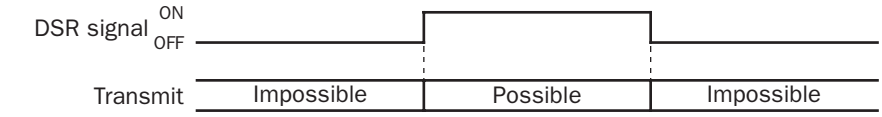
D8105 = 1: When DSR is on, the MicroSmart can transmit and receive data.



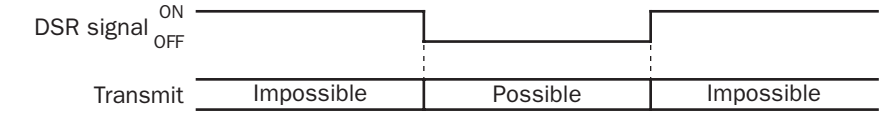
D8105 = 2: When DSR is off, the MicroSmart can transmit and receive data.



D8105 = 3: When DSR is on, the MicroSmart can transmit data. This function is usually called "Busy Control" and is used for controlling transmission to a remote terminal with a slow processing speed, such as a printer. When the remote terminal is busy, data input to the remote terminal is restricted.



D8105 = 4: When DSR is off, the MicroSmart can transmit data.



D8105 = 5 or more: Same as D8105 = 0. DSR is not used for data flow control.

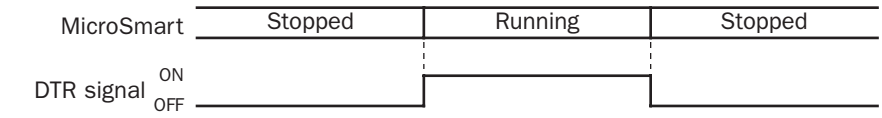
DTR Output Control Signal Option D8106

Special data register D8106 is used to control the DTR (data terminal ready) signal to indicate the MicroSmart operating status or transmitting/receiving status.

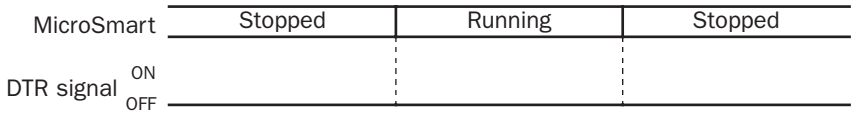
The DTR control signal option can be used only for the user communication through the RS232C port 2.

D8106 = 0 (system default):

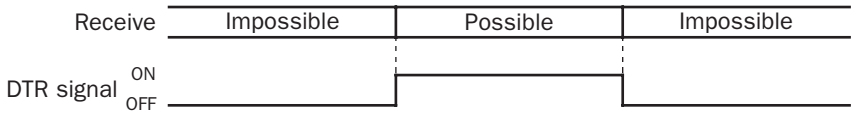
While the MicroSmart is running, DTR is on whether the MicroSmart is transmitting or receiving data. While the MicroSmart is stopped, DTR remains off. Use this option to indicate the MicroSmart operating status.



D8106 = 1: Whether the MicroSmart is running or stopped, DTR remains off.



D8106 = 2: While the MicroSmart can receive data, DTR is turned on. While the MicroSmart can not receive data, DTR remains off. Use this option when flow control of receive data is required.

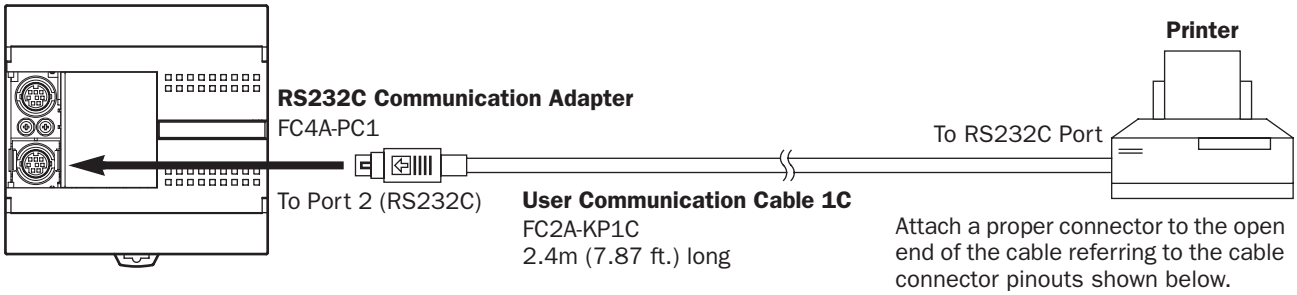


D8106 = 3 or more: Same as D8106 = 0.

Sample Program – User Communication TXD

This example demonstrates a program to send data to a printer using the user communication TXD2 (transmit) instruction, with the optional RS232C communication adapter installed on the port 2 connector of the 24-I/O type CPU module.

System Setup



Cable Connection and Pinouts

Mini DIN Connector Pinouts

Description		Color	Pin
Shield		—	**Cover**
NC	No Connection	Black	**1**
NC	No Connection	Yellow	**2**
TXD	Transmit Data	Blue	**3**
NC	No Connection	Green	**4**
DSR	Data Set Ready	Brown	**5**
NC	No Connection	Gray	**6**
SG	Signal Ground	Red	**7**
NC	No Connection	White	**8**

D-sub 9-pin Connector Pinouts

Pin	Description	
1	NC	No Connection
2	NC	No Connection
3	DATA	Receive Data
4	NC	No Connection
5	GND	Ground
6	NC	No Connection
7	NC	No Connection
8	BUSY	Busy Signal
9	NC	No Connection

The name of BUSY terminal differs depending on printers, such as DTR. The function of this terminal is to send a signal to remote equipment whether the printer is ready to print data or not. Since the operation of this signal may differ depending on printers, confirm the operation before connecting the cable.

⚠ Caution

- Do not connect any wiring to the NC (no connection) pins; otherwise, the MicroSmart and the printer may not work correctly and may be damaged.

Description of Operation

The data of counter C2 and data register D30 are printed every minute. A printout example is shown on the right.

Printout Example

```
--- PRINT TEST ---

11H 00M

CNT2...0050
D030...3854

--- PRINT TEST ---

11H 01M

CNT2...0110
D030...2124
```

Programming Special Data Register

Special data register D8105 is used to monitor the BUSY signal and to control the transmission of print data.

Special DR	Value	Description
D8105	3	While DSR is on (not busy), the CPU sends data. While DSR is off (busy), the CPU stops data transmission. If the off duration exceeds a limit (approx. 5 sec), a transmission busy timeout error will occur, and the remaining data is not sent. The transmit status data register stores an error code. See pages 17-11 and 17-27.

The MicroSmart monitors the DSR signal to prevent the receive buffer of the printer from overflowing. For the DSR signal, see page 17-30.

Setting User Communication Mode in WindLDR Function Area Settings

Since this example uses the RS232C port 2, select User Protocol for Port 2 in the Function Area Settings using WindLDR. See page 17-5.

Setting Communication Parameters

Set the communication parameters to match those of the printer. See page 17-5. For details of the communication parameters of the printer, see the user's manual for the printer. An example is shown below:

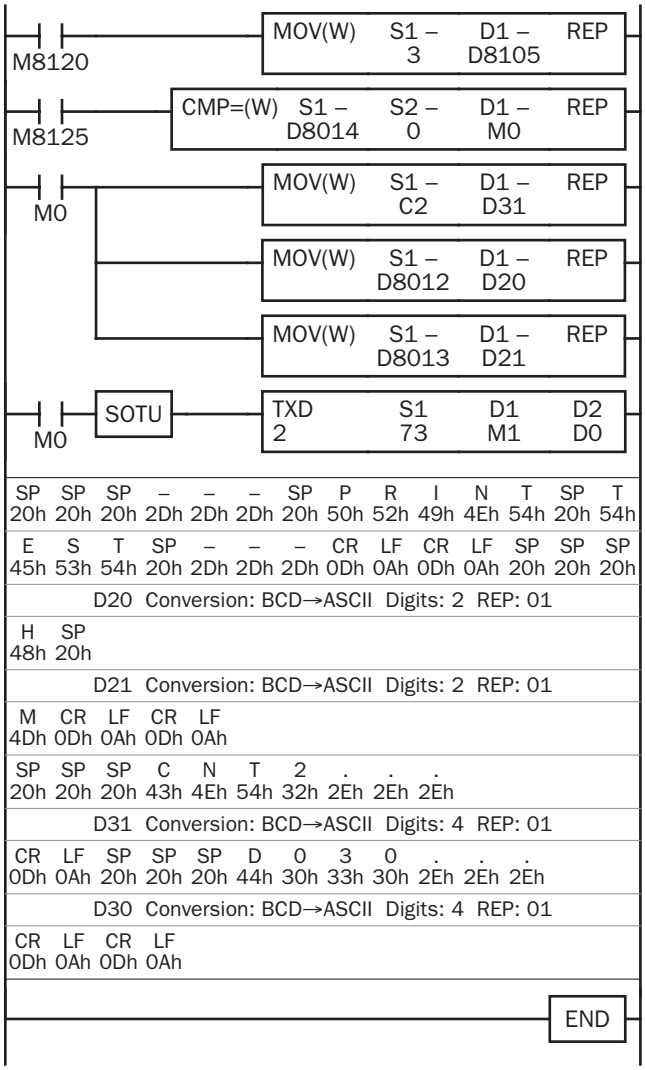
Communication Parameters:

Baud rate	9600 bps
Data bits	8
Parity check	None
Stop bits	1

Note: The receive timeout value is used for the RXD instruction in the user communication mode. Since this example uses only the TXD instruction, the receive timeout value has no effect.

Ladder Diagram

The second data stored in special data register D8014 is compared with 0 using the CMP= (compare equal to) instruction. Each time the condition is met, the TXD2 instruction is executed to send the C2 and D30 data to the printer. A counting circuit for counter C2 is omitted from this sample program.



M8120 is the initialize pulse special internal relay.

3 → D8105 to enable the DSR option for busy control.

M8125 is the in-operation output special internal relay.

CMP=(W) compares the D8014 second data with 0.

When the D8014 data equals 0 second, M0 is turned on.

Counter C2 current value is moved to D31.

D8012 hour data is moved to D20.

D8013 minute data is moved to D21.

TXD2 is executed to send 73-byte data through the RS232C port 2 to the printer.

D20 hour data is converted from BCD to ASCII, and 2 digits are sent.

D21 minute data is converted from BCD to ASCII, and 2 digits are sent.

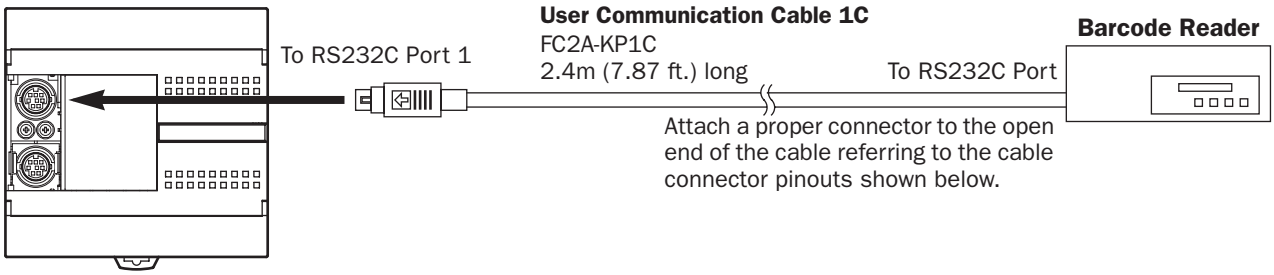
D31 counter C2 data is converted from BCD to ASCII, and 4 digits are sent.

D30 data is converted from BCD to ASCII, and 4 digits are sent.

Sample Program – User Communication RXD

This example demonstrates a program to receive data from a barcode reader with a RS232C port using the user communication RXD1 (receive) instruction.

System Setup



Mini DIN Connector Pinouts

Description		Color	Pin
Shield		—	**Cover**
NC	No Connection	Black	**1**
NC	No Connection	Yellow	**2**
TXD	Transmit Data	Blue	**3**
RXD	Receive Data	Green	**4**
NC	No Connection	Brown	**5**
NC	No Connection	Gray	**6**
SG	Signal Ground	Red	**7**
NC	No Connection	White	**8**

D-sub 25-pin Connector Pinouts

Pin	Description	
1	FG	Frame Ground
2	TXD1	Transmit Data
3	RXD1	Receive Data
7	GND	Ground

Caution

- Do not connect any wiring to the NC (no connection) pins; otherwise, the MicroSmart and the barcode reader may not work correctly and may be damaged.

Description of Operation

A barcode reader is used to scan barcodes of 8 numerical digits. The scanned data is sent to the MicroSmart through the RS232C port 1 and stored to data registers. The upper 8 digits of the data are stored to data register D20 and the lower 8 digits are stored to data register D21.

Setting User Communication Mode in WindLDR Function Area Settings

Since this example uses the RS232C port 1, select User Protocol for Port 1 in the Function Area Settings using WindLDR. See page 17-5.

Setting Communication Parameters

Set the communication parameters to match those of the barcode reader. See page 17-5. For details of the communication parameters of the barcode reader, see the user's manual for the barcode reader. An example is shown below:

Communication Parameters:

Baud rate	9600 bps
Data bits	7
Parity check	Even
Stop bits	1

Configuring Barcode Reader

The values shown below are an example of configuring a barcode reader. For actual settings, see the user's manual for the barcode reader.

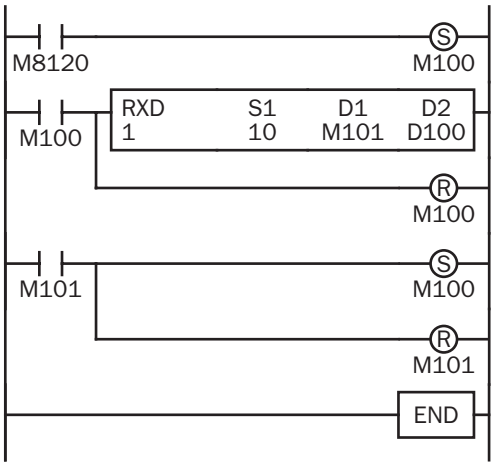
Synchronization mode	Auto			
Read mode	Single read or multiple read			
Communication parameter	Baud rate: Parity check:	9600 bps Even	Data bits: Stop bit:	7 1
Other communication settings	Header: Data echo back: Output timing: Data output filter: Sub serial:	02h No Output priority 1 No No	Terminator: BCR data output: Character suppress: Main serial input:	03h Yes No No
Comparison preset mode	Not used			

Allocation Numbers

M100	Input to start receiving barcode data
M101	Receive completion output for barcode data
M8120	Initialize pulse special internal relay
D20	Store barcode data (upper 4 digits)
D21	Store barcode data (lower 4 digits)
D100	Receive status data register for barcode data
D101	Receive data byte count data register

Ladder Diagram

When the MicroSmart starts operation, the RXD1 instruction is executed to wait for incoming data. When data receive is complete, the data is stored to data registers D20 and D21. The receive completion signal is used to execute the RXD1 instruction to wait for another incoming data.



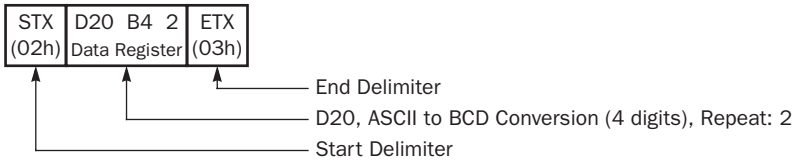
M8120 is the initialize pulse special internal relay used to set M100.

At the rising edge of M100, RXD1 is executed to be ready for receiving data.

Even after M100 is reset, RXD1 still waits for incoming data.

When data receive is complete, M101 is turned on, then M100 is set to execute RXD1 to receive the next incoming data.

RXD1 Data



New BCC Calculation Examples

The upgraded CPU modules can use three new BCC calculation formulas of ADD-2comp, Modbus ASCII, and Modbus RTU for transmit instructions TXD1 and TXD2 and receive instructions RXD1 and RXD2. Use WindLDR ver. 4.40 or higher to program the new BCC. These block check characters are calculated as described below.

ADD-2comp

1. Add the characters in the range from the BCC calculation start position to the byte immediately before the BCC.
2. Invert the result bit by bit, and add 1 (2's complement).
3. Store the result to the BCC position according to the designated conversion type (Binary to ASCII conversion or No conversion) and the designated quantity of BCC digits.

Example: Binary to ASCII conversion, 2 BCC digits

When the result of step **2** is 175h, the BCC will consist of 37h, 35h.

Modbus ASCII — Calculating the LRC (longitudinal redundancy check)

1. Convert the ASCII characters in the range from the BCC calculation start position to the byte immediately before the BCC, in units of two characters, to make 1-byte hexadecimal data. (Example: 37h, 35h → 75h)
2. Add up the results of step **1**.
3. Invert the result bit by bit, and add 1 (2's complement).
4. Convert the lowest 1-byte data to ASCII characters. (Example: 75h → 37h, 35h)
5. Store the two digits to the BCC (LRC) position.

Modbus RTU — Calculating the CRC-16 (cyclic redundancy checksum)

1. Take the exclusive OR (XOR) of FFFFh and the first 1-byte data at the BCC calculation start position.
2. Shift the result by 1 bit to the right. When a carry occurs, take the exclusive OR (XOR) of A001h, then go to step **3**. If not, directly go to step **3**.
3. Repeat step **2**, shifting 8 times.
4. Take the exclusive OR (XOR) of the result and the next 1-byte data.
5. Repeat step **2** through step **4** up to the byte immediately before the BCC.
6. Swap the higher and lower bytes of the result of step **5**, and store the resultant CRC-16 to the BCC (CRC) position. (Example: 1234h → 34h, 12h)

18: PROGRAM BRANCHING INSTRUCTIONS

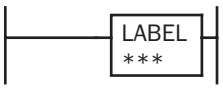
Introduction

The program branching instructions reduce execution time by making it possible to bypass portions of the program whenever certain conditions are not satisfied.

The basic program branching instructions are LABEL and LJMP, which are used to tag an address and jump to the address which has been tagged. Programming tools include "either/or" options between numerous portions of a program and the ability to call one of several subroutines which return execution to where the normal program left off.

The DI or EI instruction disables or enables interrupt inputs and timer interrupt individually.

LABEL (Label)



This is the label number, from 0 to 127, used at the program address where the execution of program instructions begins for a program branch.

An END instruction may be used to separate a tagged portion of the program from the main program. In this way, scan time is minimized by *not* executing the program branch unless input conditions are satisfied.

Note: The same label number cannot be used more than once.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
Label number	Tag for LJMP and LCAL	—	—	—	—	—	—	—	0-127	—

LJMP (Label Jump)

When input is on, jump to the address with label 0 through 127 designated by S1.

When input is off, no jump takes place, and program execution proceeds with the next instruction.

The LJMP instruction is used as an "either/or" choice between two portions of a program. Program execution does *not* return to the instruction following the LJMP instruction, after the program branch.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Label number to jump to	—	—	—	—	—	—	X	0-127	—

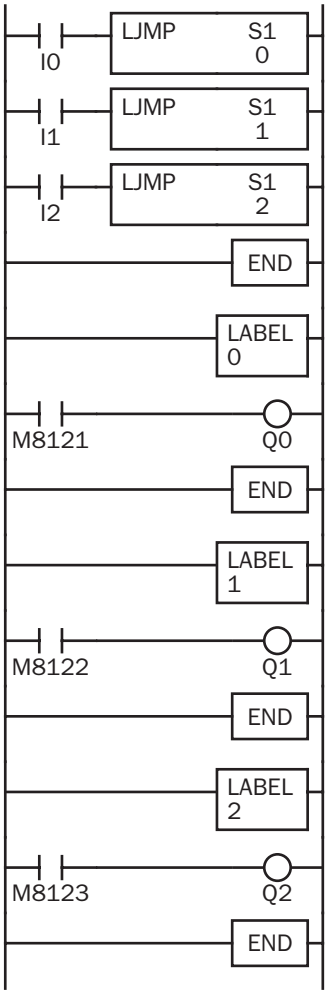
For the valid operand number range, see pages 6-1 and 6-2.

Since the LJMP instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Note: Make sure that a LABEL instruction of the label number used for a LJMP instruction is programmed. When designating S1 using other than a constant, the value for the label is a variable. When using a variable for a label, make sure that all probable LABEL numbers are included in the user program. If a matching label does not exist, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Example: LJMP and LABEL

The following example demonstrates a program to jump to three different portions of program depending on the input.



When input I0 is on, program execution jumps to label 0.

When input I1 is on, program execution jumps to label 1.

When input I2 is on, program execution jumps to label 2.

M8121 is the 1-sec clock special internal relay.

When jump occurs to label 0, output Q0 oscillates in 1-sec increments.

M8122 is the 100-ms clock special internal relay.

When jump occurs to label 1, output Q1 oscillates in 100-ms increments.

M8123 is the 10-ms clock special internal relay.

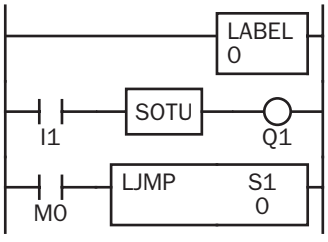
When jump occurs to label 2, output Q2 oscillates in 10-ms increments.

Using the Timer Instruction with Program Branching

When the timer start input of the TML, TIM, TMH or TMS instruction is already on, timedown begins immediately at the location jumped to, starting with the timer current value. When using a program branch, it is important to make sure that timers are initialized when desired, after the jump. If it is necessary to initialize the timer instruction (set to the preset value) after the jump, the timer's start input should be kept off for one or more scan cycles before initialization. Otherwise, the timer input on will not be recognized.

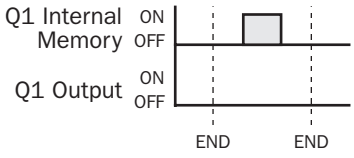
Using the SOTU/SOTD Instructions with Program Branching

Check that pulse inputs of counters and shift registers, and input of single outputs (SOTU and SOTD) are maintained during the jump, if required. Hold the input off for one or more scan cycles after the jump for the rising or falling edge transition to be recognized.



Although normally, the SOTU instruction produces a pulse for one scan, when used in a program branch the SOTU pulse will last only until the next time the same SOTU instruction is executed.

In the example on the left, the program branch will loop as long as internal relay M0 remains on. However, the SOTU produces a pulse output only during the first loop.



Since the END instruction is not executed as long as M0 remains on, output Q1 is not turned on even if input I1 is on.

LCAL (Label Call)

When input is on, the address with label 0 through 127 designated by S1 is called. When input is off, no call takes place, and program execution proceeds with the next instruction.

The LCAL instruction calls a subroutine, and returns to the main program after the branch is executed. A LRET instruction (see below) must be placed at the end of a program branch which is called, so that normal program execution resumes by returning to the instruction following the LCAL instruction.

Note: The END instruction must be used to separate the main program from any subroutines called by the LCAL instruction.

A maximum of four LCAL instructions can be nested. When more than four LCAL instructions are nested, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

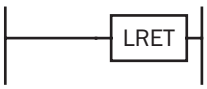
Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Label number to call	—	—	—	—	—	—	X	0-127	—

For the valid operand number range, see pages 6-1 and 6-2.

Since the LCAL instruction is executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used as required.

Note: Make sure that a LABEL instruction of the label number used for a LCAL instruction is programmed. When designating S1 using other than a constant, the value for the label is a variable. When using a variable for a label, make sure that all probable LABEL numbers are included in the user program. If a matching label does not exist, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

LRET (Label Return)



This instruction is placed at the end of a subroutine called by the LCAL instruction. When the subroutine is completed, normal program execution resumes by returning to the instruction following the LCAL instruction.

The LRET must be placed at the end of the subroutine starting with a LABEL instruction. When the LRET is programmed at other places, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

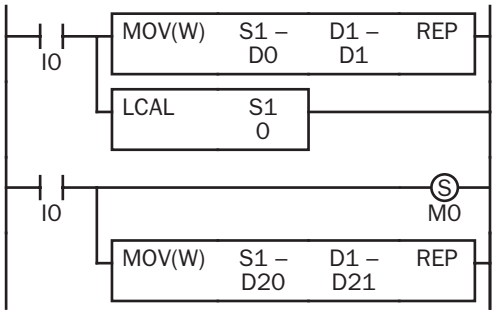
Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
—	—	—	—	—	—	—	—	—	—	—

Correct Structure for Calling Subroutine

When a LCAL instruction is executed, the remaining program instructions on the same rung may not be executed upon return, if input conditions are changed by the subroutine. After the LRET instruction of a subroutine, program execution begins with the instruction following the LCAL instruction, depending on current input condition.

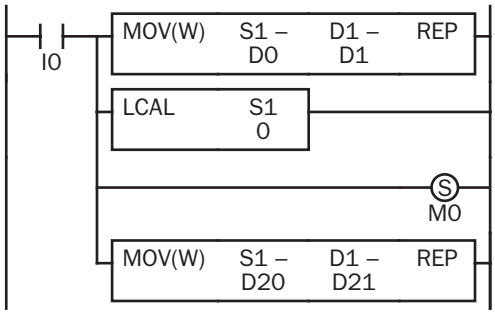
When instructions following a LCAL instruction must be executed after the subroutine is called, make sure the subroutine does not change input conditions unfavorably. In addition, include subsequent instructions in a new ladder line, separated from the LCAL instruction.

Correct



Separate the ladder line for each LCAL instruction.

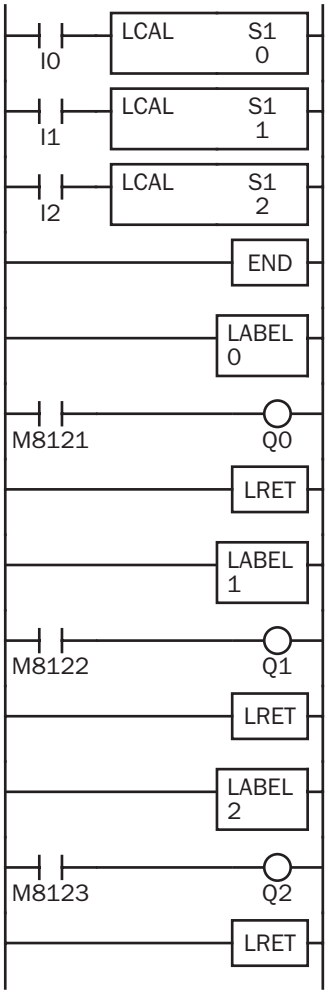
Incorrect



I0 status may be changed by the subroutine upon return.

Example: LCAL and LRET

The following example demonstrates a program to call three different portions of program depending on the input. When the subroutine is complete, program execution returns to the instruction following the LCAL instruction.



When input I0 is on, program execution jumps to label 0.

When input I1 is on, program execution jumps to label 1.

When input I2 is on, program execution jumps to label 2.

M8121 is the 1-sec clock special internal relay.

When jump occurs to label 0, output Q0 oscillates in 1-sec increments.

Program execution returns to the address of input I1.

M8122 is the 100-ms clock special internal relay.

When jump occurs to label 1, output Q1 oscillates in 100-ms increments.

Program execution returns to the address of input I2.

M8123 is the 10-ms clock special internal relay.

When jump occurs to label 2, output Q2 oscillates in 10-ms increments.

Program execution returns to the address of END.

IOREF (I/O Refresh)

When input is on, 1-bit I/O data designated by source operand S1 is refreshed immediately regardless of the scan time.

When I (input) is used as S1, the actual input status is immediately read into an internal relay starting with M300 allocated to each input available on the CPU module.

When Q (output) is used as S1, the output data in the RAM is immediately written to the actual output available on the CPU module.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
X	X	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	I/O for refresh	X	X	—	—	—	—	—	—	—

Only input or output numbers available on the CPU module can be designated as S1. Input and output numbers for expansion I/O modules cannot be designated as S1. For the valid operand number range, see pages 6-1 and 6-2.

Input Operand Numbers and Allocated Internal Relays

Input Operand	Internal Relay	Input Operand	Internal Relay	Input Operand	Internal Relay
I0	M300	I5	M305	I12 *	M312
I1	M301	I6	M306	I13 *	M313
I2	M302	I7	M307	I14 *	M314
I3	M303	I10 *	M310	I15 *	M315
I4	M304	I11 *	M311		

Note *: Slim type CPU modules FC4A-D40K3 and FC4A-D40S3 cannot use I10 through I15 as source operand S1; only I0 through I7 can be designated as source operand S1 for the FC4A-D40K3 and FC4A-D40S3.

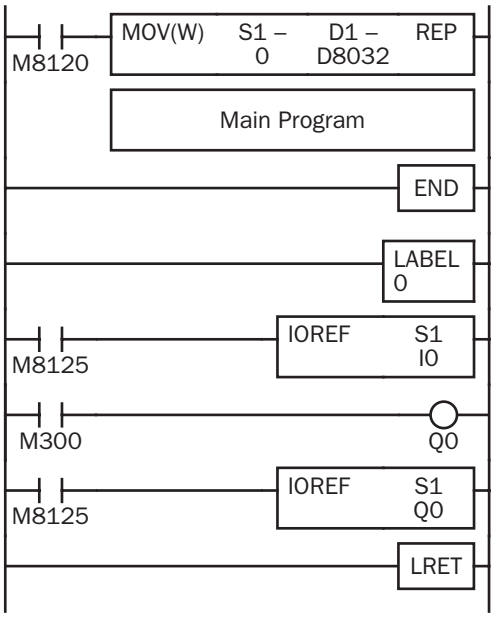
During normal execution of a user program, I/O statuses are refreshed simultaneously when the END instruction is executed at the end of a scan. When a real-time response is needed to execute an interrupt, the IOREF instruction can be used. When the input to the IOREF instruction is turned on, the status of the designated input or output is read or written immediately.

When the IOREF instruction is executed for an input, the filter does not take effect and the input status at the moment is read to a corresponding internal relay.

The actual input status of the same input number is read to the internal input memory when the END instruction is executed as in the normal scanning, then the filter value has effect as designated in the Function Area Settings. See page 5-24.

Example: IOREF

The following example demonstrates a program to transfer the input I0 status to output Q0 using the IOREF instruction. Input I2 is designated as an interrupt input. For the interrupt input function, see page 5-20.



M8120 is the initialize pulse special internal relay.

D8032 stores 0 to designate jump destination label 0 for interrupt input I2.

When input I2 is on, program execution jumps to label 0.

M8125 is the in-operation output special internal relay.

IOREF immediately reads input I0 status to internal relay M300.

M300 turns on or off the output Q0 internal memory.

Another IOREF immediately writes the output Q0 internal memory status to actual output Q0.

Program execution returns to the main program.

DI (Disable Interrupt)

When input is on, interrupt inputs and timer interrupt designated by source operand S1 are disabled.

EI (Enable Interrupt)

When input is on, interrupt inputs and timer interrupt designated by source operand S1 are enabled.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Interrupt inputs and timer interrupt	—	—	—	—	—	—	—	1-31	—

Interrupt inputs I2 through I5 and timer interrupt selected in the Function Area Settings are normally enabled when the CPU starts. When the DI instruction is executed, interrupt inputs and timer interrupt designated as source operand S1 are disabled even if the interrupt condition is met in the user program area subsequent to the DI instruction. When the EI instruction is executed, disabled interrupt inputs and timer interrupt designated as source operand S1 are enabled again in the user program area subsequent to the EI instruction. Different operands can be selected for the DI and EI instructions to disable and enable interrupt inputs selectively. For Interrupt Input and Timer Interrupt, see pages 5-20 and 5-22.

Make sure that interrupt inputs and timer interrupt designated as source operand S1 are selected in the Function Area Settings. Otherwise, when the DI or EI instruction is executed, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

The DI and EI instructions cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

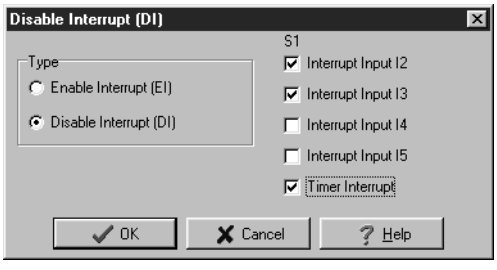
Special Internal Relays M8140-M8144: Interrupt Status

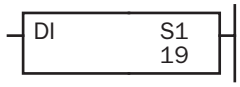
Special internal relays M8140 through M8144 are provided to indicate whether interrupt inputs and timer interrupt are enabled or disabled.

Interrupt	Interrupt Enabled	Interrupt Disabled
Interrupt Input I2	M8140 ON	M8140 OFF
Interrupt Input I3	M8141 ON	M8141 OFF
Interrupt Input I4	M8142 ON	M8142 OFF
Interrupt Input I5	M8143 ON	M8143 OFF
Timer Interrupt	M8144 ON	M8144 OFF

Programming WindLDR

In the Disable Interrupt (DI) or Enable Interrupt (EI) dialog box, click the check box on the left of Interrupt Inputs I2 through I5 or Timer Interrupt to select source operand S1. The example below selects interrupt inputs I2, I3, and timer interrupt for the DI instruction, and a 19 will be shown as source operand S1.



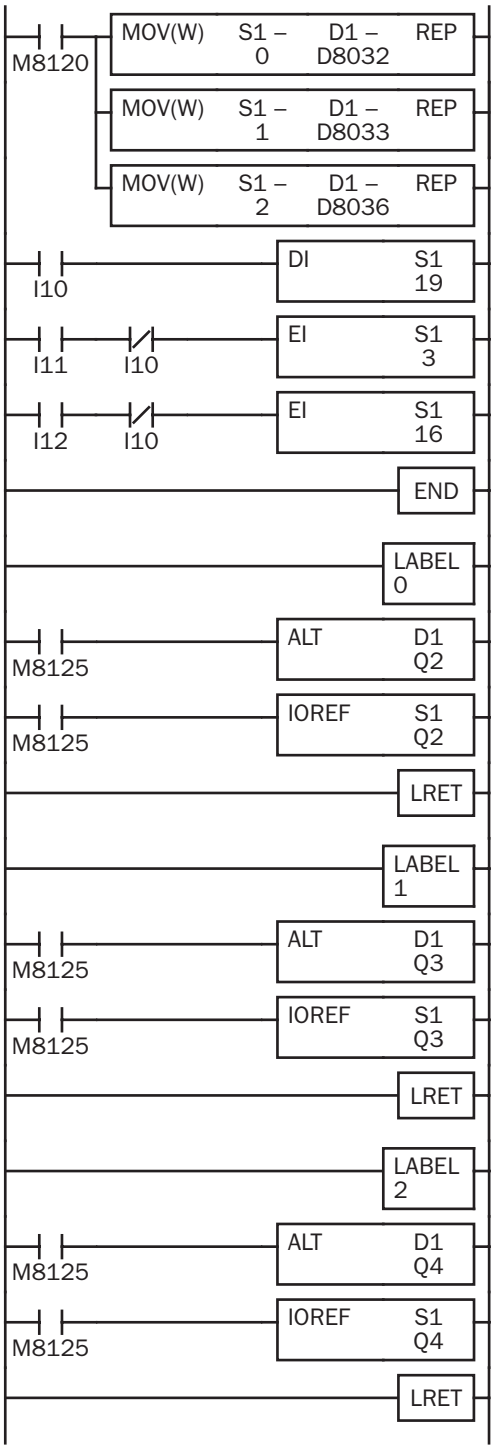


The total of selected interrupt inputs and timer interrupt is shown as source operand S1.

Interrupt	S1 Value
Interrupt Input I2	1
Interrupt Input I3	2
Interrupt Input I4	4
Interrupt Input I5	8
Timer Interrupt	16

Example: DI and EI

The following example demonstrates a program to disable and enable interrupt inputs and timer interrupt selectively. For the interrupt input and timer interrupt functions, see pages 5-20 and 5-22. In this example, inputs I2 and I3 are designated as interrupt inputs and timer interrupt is used with interrupt intervals of 100 ms.



M8120 is the initialize pulse special internal relay.

D8032 stores jump destination label number 0 for interrupt input I2.

D8033 stores jump destination label number 1 for interrupt input I3.

D8036 stores jump destination label number 2 for timer interrupt.

When input I10 is on, DI disables interrupt inputs I2, I3, and timer interrupt, then M8140, M8141, and M8144 turn off.

When input I11 is on and I10 is off, EI enables interrupt inputs I2 and I3, then M8140 and M8141 turn on.

When input I12 is on and I10 is off, EI enables timer interrupt, then M8144 turns on.

End of the main program.

When input I2 is on, program execution jumps to label 0.

M8125 is the in-operation output special internal relay.

ALT turns on or off the output Q2 internal memory.

IOREF immediately writes the output Q2 internal memory status to actual output Q2.

Program execution returns to the main program.

When input I3 is on, program execution jumps to label 1.

M8125 is the in-operation output special internal relay.

ALT turns on or off the output Q3 internal memory.

IOREF immediately writes the output Q3 internal memory status to actual output Q3.

Program execution returns to the main program.

Timer interrupt occurs every 100 ms, then program execution jumps to label 2.

M8125 is the in-operation output special internal relay.

ALT turns on or off the output Q4 internal memory.

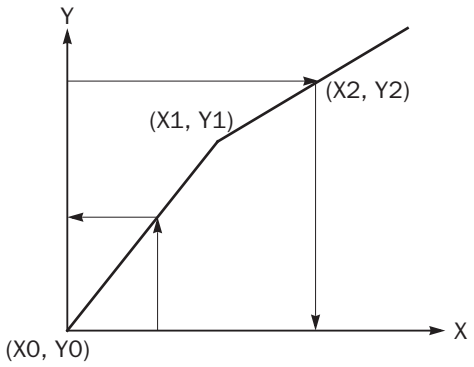
IOREF immediately writes the output Q4 internal memory status to actual output Q4.

Program execution returns to the main program.

19: COORDINATE CONVERSION INSTRUCTIONS

Introduction

The coordinate conversion instructions convert one data point to another value, using a linear relationship between values of X and Y.



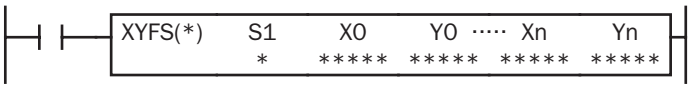
Upgrade Information

Upgraded CPU modules can use an expanded range of X and Y values. Word and integer data types can be designated for the Y values. Applicable CPU modules and system program version are shown in the table below. For the procedure to confirm the system program version of the CPU module, see page 29-1.

CPU Module	All-in-One Type			Slim Type	
	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
System Program Version	—	—	204 or higher	204 or higher	203 or higher

Use WindLDR ver. 4.50 or higher to program the upgraded coordinate conversion instructions.

XYFS (XY Format Set)



When input is on, the format for XY conversion is set. The number of XY coordinates, defining the linear relationship between X and Y, can be 2 to 5 points. ($0 \leq n \leq 4$)

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Format number	—	—	—	—	—	—	—	0 to 5	—
X0 through Xn	X value	X	X	X	X	X	X	X	0 to 32767 0 to 65535	—
Y0 through Yn	Y value	X	X	X	X	X	X	X	0 to 65535 –32768 to 32767	—

For the valid operand number range, see pages 6-1 and 6-2.

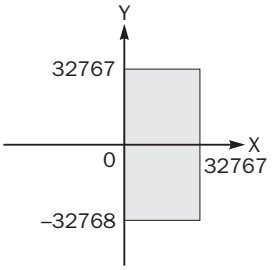
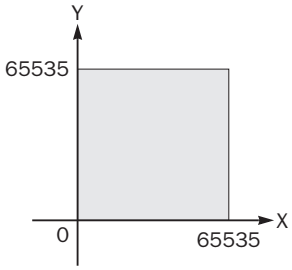
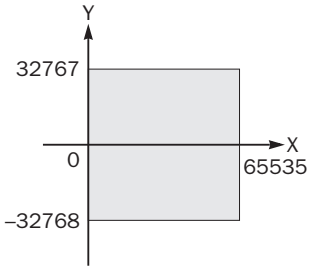
When T (timer) or C (counter) is used as X0 through Xn or Y0 through Yn, the timer/counter current value is read out.

S1 (Format number)

Select a format number 0 through 5. A maximum of 6 formats for XY conversion can be set.

Xn (X value), Yn (Y value)

Enter values for the X and Y coordinates. Three different data ranges are available depending on the system program version and the data type.

System Program	Old System Program Versions	Upgraded System Program Versions	
Data Type	Integer	Word	Integer
Xn (X value)	0 to 32767	0 to 65535	0 to 65535
Yn (Y value)	–32768 to 32767	0 to 65535	–32768 to 32767
Valid Coordinates	 If the X value becomes negative, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.		

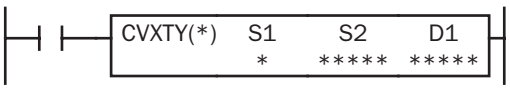
Valid Data Types

W (word)	I (integer)
W	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as Xn or Yn, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as Xn or Yn, 1 point is used.

CVXTY (Convert X to Y)



When input is on, the X value designated by operand S2 is converted into corresponding Y value according to the linear relationship defined in the XYFS instruction. Operand S1 selects a format from a maximum of six XY conversion formats. The conversion result is set to the operand designated by D1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Format number	—	—	—	—	—	—	—	0 to 5	—
S2 (Source 2)	X value	X	X	X	X	X	X	X	0 to 32767 0 to 65535	—
D1 (Destination 1)	Destination to store results	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

S1 (Format number)

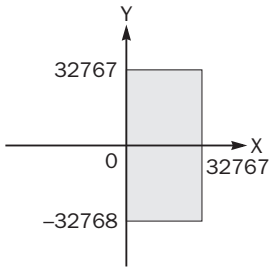
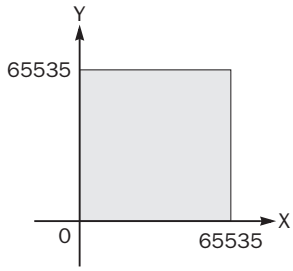
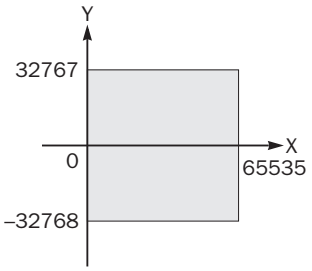
Select a format number 0 through 5 which have been set using the XYFS instruction. When an XYFS instruction with the corresponding format number is not programmed, or when XYFS and CVXTY instructions of the same format number have different data type designations, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

S2 (X value)

Enter a value for the X coordinate to convert, within the range specified in the XYFS instruction. Any value out of the range specified in the XYFS results in a user program execution error, turning on special internal relay M8004 and the ERR LED. Three different data ranges are available depending on the system program version and the data type.

D1 (Destination to store results)

The conversion result of the Y value is stored to the destination. The data range depends on the available data type.

System Program	Old System Program Versions	Upgraded System Program Versions	
Data Type	Integer	Word	Integer
S2 (X value)	0 to 32767	0 to 65535	0 to 65535
D1 (Y value)	–32768 to 32767	0 to 65535	–32768 to 32767
Valid Coordinates			

Valid Data Types

W (word)	I (integer)
X	X

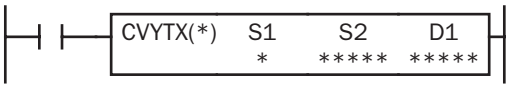
When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as S2 or D1, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as S2 or D1, 1 point is used.

Data Conversion Error

The data conversion error is ±0.5.

CVYTX (Convert Y to X)



When input is on, the Y value designated by operand S2 is converted into corresponding X value according to the linear relationship defined in the XYFS instruction. Operand S1 selects a format from a maximum of six XY conversion formats. The conversion result is set to the operand designated by D1.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Format number	—	—	—	—	—	—	—	0 to 5	—
S2 (Source 2)	Y value	X	X	X	X	X	X	X	0 to 65535 –32768 to 32767	—
D1 (Destination 1)	Destination to store results	—	X	▲	X	X	X	X	—	—

For the valid operand number range, see pages 6-1 and 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

When T (timer) or C (counter) is used as S2, the timer/counter current value is read out. When T (timer) or C (counter) is used as D1, the data is written in as a preset value which can be 0 through 65535.

S1 (Format number)

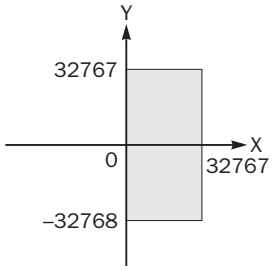
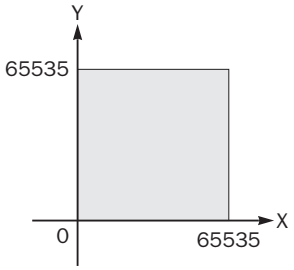
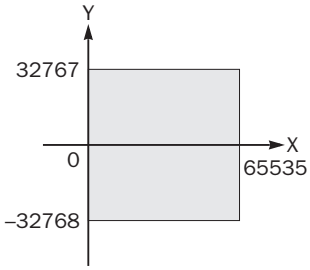
Select a format number 0 through 5 which have been set using the XYFS instruction. When an XYFS instruction with the corresponding format number is not programmed, or when XYFS and CVYTX instructions of the same format number have different data type designations, a user program execution error will result, turning on special internal relay M8004 and the ERR LED.

S2 (Y value)

Enter a value for the Y coordinate to convert, within the range specified in the XYFS instruction. Any value out of the range specified in the XYFS results in a user program execution error, turning on special internal relay M8004 and the ERR LED. Three different data ranges are available depending on the system program version and the data type.

D1 (Destination to store results)

The conversion result of the X value is stored to the destination. The data range depends on the available data type.

System Program	Old System Program Versions	Upgraded System Program Versions	
Data Type	Integer	Word	Integer
S2 (Y value)	–32768 to 32767	0 to 65535	–32768 to 32767
D1 (X value)	0 to 32767	0 to 65535	0 to 65535
Valid Coordinates			

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as S2 or D1, 16 points are used.

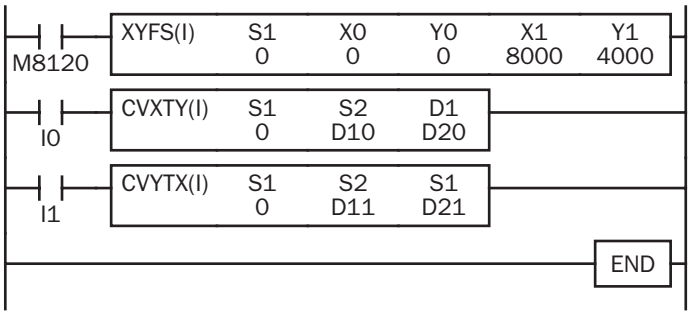
When a word operand such as T (timer), C (counter), or D (data register) is designated as S2 or D1, 1 point (integer data type) is used.

Data Conversion Error

The data conversion error is ±0.5.

Example: Linear Conversion

The following example demonstrates setting up two coordinate points to define the linear relationship between X and Y. The two points are (X0, Y0) = (0, 0) and (X1, Y1) = (8000, 4000). Once these are set, there is an X to Y conversion, as well as a Y to X conversion.

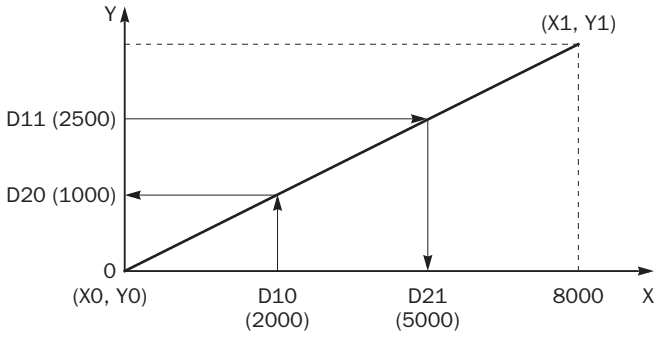


M8120 is the initialize pulse special internal relay.

At startup, XYFS specifies two points.

When input I0 is on, CVXTY converts the value in D10 and stores the result in D20.

When input I1 is on, CVYTX converts the value in D11 and stores the result in D21.



The graph shows the linear relationship that is defined by the two points:

$$Y = \frac{1}{2}X$$

If the value in data register D10 is 2000, the value assigned to D20 is 1000.

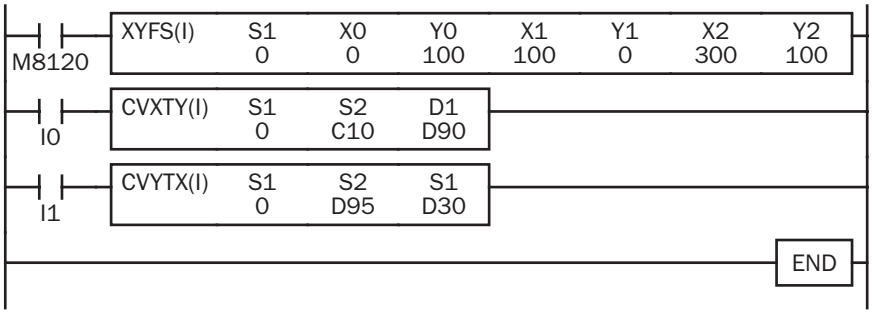
For Y to X conversion, the following equation is used:

$$X = 2Y$$

If the value in data register D11 is 2500, the value assigned to D21 is 5000.

Example: Overlapping Coordinates

In this example, the XYFS instruction sets up three coordinate points, which define two different linear relationships between X and Y. The three points are: (X0, Y0) = (0, 100), (X1, Y1) = (100, 0), and (X2, Y2) = (300, 100). The two line segments define overlapping coordinates for X. That is, for each value of Y within the designated range, there would be two X values assigned.

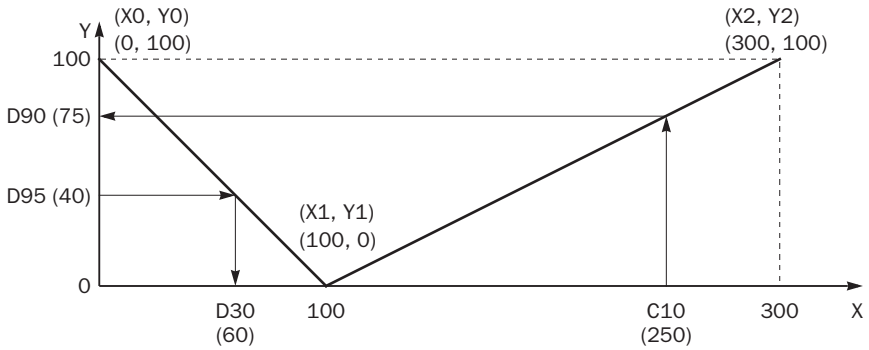


M8120 is the initialize pulse special internal relay.

At startup, XYFS specifies three points.

CVXTY converts the value in C10 and stores the result in D90.

CVYTX converts the value in D95 and stores the result in D30.



The first line segment defines the following relationship for X to Y conversion:

$$Y = -X + 100$$

The second line segment defines another relationship for X to Y conversion:

$$Y = \frac{1}{2}X - 50$$

For X to Y conversion, each value of X has only one corresponding value for Y. If the current value of counter C10 is 250, the value assigned to D90 is 75.

For Y to X conversion, the XYFS instruction assigns two possible values of X for each value of Y. The relationship defined by the first two points has priority in these cases. The line between points (X0, Y0) and (X1, Y1), that is, the line between (0, 100) and (100, 0), has priority in defining the relationship for Y to X conversion (X = –Y + 100).

Therefore, if the value in data register D95 is 40, the value assigned to D30 is 60, not 180.

Exactly the same two line segments might also be defined by the XYFS instruction, except that the point (300, 100) could be assigned first, as (X0, Y0), and the point (100, 0) could be defined next, as (X1, Y1). In this case, this linear relationship would have priority.

In this case, if the value in data register D95 is 40, the value assigned to D30 is 180, not 60.

20: PULSE INSTRUCTIONS

Introduction

The PULS (pulse output) instruction is used to generate pulse outputs of 10 Hz through 20,000 Hz which can be used to control pulse motors for simple position control applications.

The PWM (pulse width modulation) instruction is used to generate pulse outputs of 6.81, 27.26, or 217.86 Hz with a variable pulse width ratio between 0% and 100%, which can be used for illumination control.

The RAMP instruction is used for trapezoidal control, and the ZRN instruction for zero-return control.

The PULS, PWM, and RAMP instructions can be used on all slim type CPU modules, and the ZRN instructions on the FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3 only.

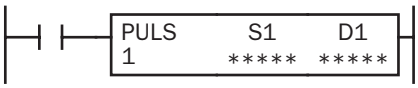
Upgrade Information

Upgraded CPU modules have an additional option of operation mode 3 to select a frequency range from 10 Hz to 20 kHz for the PULS and RAMP instructions, and also have special data registers D8055 and D8056 to store the current output pulse frequency of the PULS and RAMP instructions. Applicable CPU modules and system program versions are shown in the table below. For the procedure to confirm the system program version of the CPU module, see page 29-1.

CPU Module	All-in-One Type			Slim Type	
	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
System Program Version	—	—	—	204 or higher	202 or higher

Use WindLDR ver. 4.50 or higher to program the additional option for the PULS and RAMP instructions.

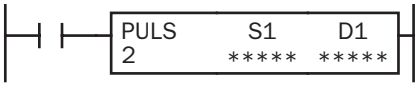
PULS1 (Pulse Output 1)



When input is on, the PULS1 instruction sends out a pulse output from output Q0. The output pulse frequency is determined by source operand S1. The output pulse width ratio is fixed at 50%.

PULS1 can be programmed to generate a predetermined number of output pulses. When pulse counting is disabled, PULS1 generates output pulses while the start input for the PULS1 instruction remains on.

PULS2 (Pulse Output 2)



When input is on, the PULS2 instruction sends out a pulse output from output Q1. The output pulse frequency is determined by source operand S1. The output pulse width ratio is fixed at 50%.

PULS2 generates output pulses while the start input for the PULS2 instruction remains on. PULS2 cannot be programmed to generate a predetermined number of output pulses.

Note: The PULS1 and PULS2 instructions can be used only once in a user program. When PULS1 or PULS2 is not used, unused output Q0 or Q1 can be used for another pulse instruction or ordinary output.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Control register	—	—	—	—	—	—	X	—	—
D1 (Destination 1)	Status relay	—	—	X	—	—	—	—	—	—

Source operand S1 (control register) uses 8 data registers starting with the operand designated as S1. Data registers D0 through D1292 and D2000 through D7992 can be designated as S1. For details, see the following pages.

Destination operand D1 (status relay) uses 3 internal relays starting with the operand designated as D1. Internal relays M0 through M1270 can be designated as D1. The least significant digit of the internal relay number designated as D1 must be 0. Special internal relays cannot be designated as D1. For details, see page 6-2.

Source Operand S1 (Control Register)

Store appropriate values to data registers starting with the operand designated by S1 before executing the PULS instruction as required, and make sure that the values are within the valid range. Operands S1+5 through S1+7 are for read only.

Operand	Function	Description	R/W
S1+0	Operation mode	0: 10 to 1,000 Hz 1: 100 to 10,000 Hz 2: 1,000 to 20,000 Hz 3: 10 to 20,000 Hz (upgraded CPU only)	R/W
S1+1	Output pulse frequency	When S1+0 (operation mode) = 0 or 1: 1 to 100 (%) (1% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 2: 1 to 20 (×5%) (5% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 3: 10 to 20,000 (Hz)	R/W
S1+2	Pulse counting	0: Disable pulse counting 1: Enable pulse counting (PULS1 only)	R/W
S1+3	Preset value (high word)	1 to 100,000,000 (05F5 E100h) (PULS1 only)	R/W
S1+4	Preset value (low word)		
S1+5	Current value (high word)	1 to 100,000,000 (05F5 E100h) (PULS1 only)	R
S1+6	Current value (low word)		
S1+7	Error status	0 to 5	R

S1+0 Operation Mode

The value stored in the data register designated by operand S1+0 determines the frequency range of the pulse output.

0: 10 to 1,000 Hz

1: 100 to 10,000 Hz

2: 1,000 to 20,000 Hz

3: 10 to 20,000 Hz (upgraded CPU only)

S1+1 Output Pulse Frequency

When S1+0 is set to 0 through 2, the value stored in the data register designated by operand S1+1 specifies the frequency of the pulse output in percent of the maximum of the frequency range selected by S1+0. When S1+0 is set to 0 (10 to 1,000 Hz) or 1 (100 to 10,000 Hz), valid values for operand S1+1 are 1 through 100, thus the output pulse frequency can be 10 to 1,000 Hz or 100 to 10,000 Hz, respectively. When S1+0 is set to 2 (1,000 to 20,000 Hz), valid values for operand S1+1 are 1 through 20 and the S1+1 value multiplied by 5 determines the output pulse frequency, thus the output pulse frequency can be 1,000 to 20,000 Hz.

When S1+0 is set to 3, the value stored in the data register designated by operand S1+1 determines the frequency of the pulse output directly. Valid values are 10 through 20,000.

Operation Mode	Output Pulse Frequency (Hz)
0 or 1	Maximum frequency (Hz) selected by S1+0 × S1+1 value (%)
2	Maximum frequency (Hz) selected by S1+0 × S1+1 value (×5%)
3	Output pulse frequency (Hz) selected by S1+1

S1+2 Pulse Counting

Pulse counting can be enabled for the PULS1 instruction only. With pulse counting enabled, PULS1 generates a predetermined number of output pulses as designated by operands S1+3 and S1+4. With pulse counting disabled, PULS1 or PULS2 generates output pulses while the start input for the PULS instruction remains on.

0: Disable pulse counting

1: Enable pulse counting (PULS1 only)

When programming PULS2, store 0 to the data register designated by S1+2.

S1+3 Preset Value (High Word)
S1+4 Preset Value (Low Word)

With pulse counting enabled as described above, PULS1 generates a predetermined number of output pulses as designated by operands S1+3 and S1+4. The preset value can be 1 through 100,000,000 (05F5 E100h) stored in two consecutive data registers designated by S1+3 (high word) and S1+4 (low word).

When pulse counting is disabled for PULS1 or when programming PULS2, store 0 to data registers designated by S1+3 and S1+4.

S1+5 Current Value (High Word)
S1+6 Current Value (Low Word)

While the PULS1 instruction is executed, the output pulse count is stored in two consecutive data registers designated by operands S1+5 (high word) and S1+6 (low word). The current value can be 1 through 100,000,000 (05F5 E100h) and is updated in every scan.

S1+7 Error Status

When the start input for the PULS1 or PULS2 instruction is turned on, operand values are checked. When any error is found in the operand values, the data register designated by operand S1+7 stores an error code.

Error Code	Operation Modes 0 through 2	Operation Mode 3
0	Normal	
1	Operation mode designation error (S1+0 stores other than 0 through 2)	Operation mode designation error (S1+0 stores other than 0 through 3)
2	Output pulse frequency designation error (S1+1 stores other than 1 through 100)	Output pulse frequency designation error (S1+1 stores other than 10 through 20,000)
3	Pulse counting designation error (S1+2 stores other than 0 and 1)	
4	Preset value designation error (S1+3 and S1+4 store other than 1 through 100,000,000)	
5	Invalid pulse counting designation for PULS2 (S1+2 stores 1)	

Destination Operand D1 (Status Relay)

Three internal relays starting with the operand designated by D1 indicate the status of the PULS instruction. These operands are for read only.

Operand	Function	Description	R/W
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	R
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	R
D1+2	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred (PULS1 only)	R

D1+0 Pulse Output ON

The internal relay designated by operand D1+0 remains on while the PULS instruction generates output pulses. When the start input for the PULS instruction is turned off or when the PULS1 instruction has completed generating a predetermined number of output pulses, the internal relay designated by operand D1+0 turns off.

D1+1 Pulse Output Complete

The internal relay designated by operand D1+1 turns on when the PULS1 instruction has completed generating a predetermined number of output pulses or when either PULS instruction is stopped to generate output pulses. When the start input for the PULS instruction is turned on, the internal relay designated by operand D1+1 turns off.

D1+2 Pulse Output Overflow

The internal relay designated by operand D1+2 turns on when the PULS1 instruction has generated more than the predetermined number of output pulses. When the start input for the PULS instruction is turned on, the internal relay designated by operand D1+2 turns off.

Special Data Register for Pulse Outputs

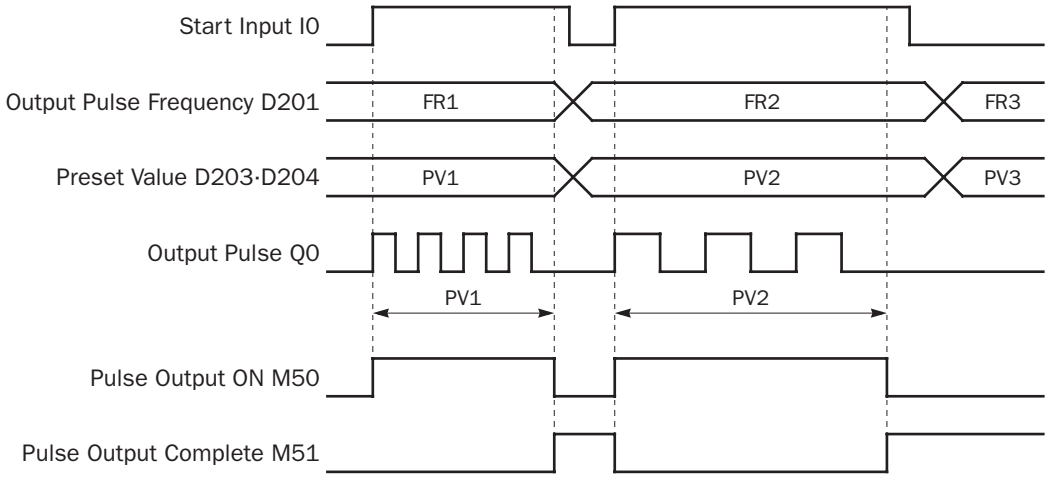
Upgraded CPU modules have two additional special data registers to store the current frequency of pulse outputs.

Allocation No.	Function	Description
D8055	Current Pulse Frequency of PULS1 or RAMP (Q0)	While the PULS1 or RAMP instruction is executed, D8055 stores the current pulse frequency of output Q0. The value is updated every scan.
D8056	Current Pulse Frequency of PULS2 or RAMP (Q1)	While the PULS2 or RAMP (reversible control dual-pulse output) instruction is executed, D8056 stores the current pulse frequency of output Q1. The value is updated every scan.

Timing Chart for Enable Pulse Counting

This program demonstrates a timing chart of the PULS1 instruction when pulse counting is enabled.

D202 = 1 (enable pulse counting)

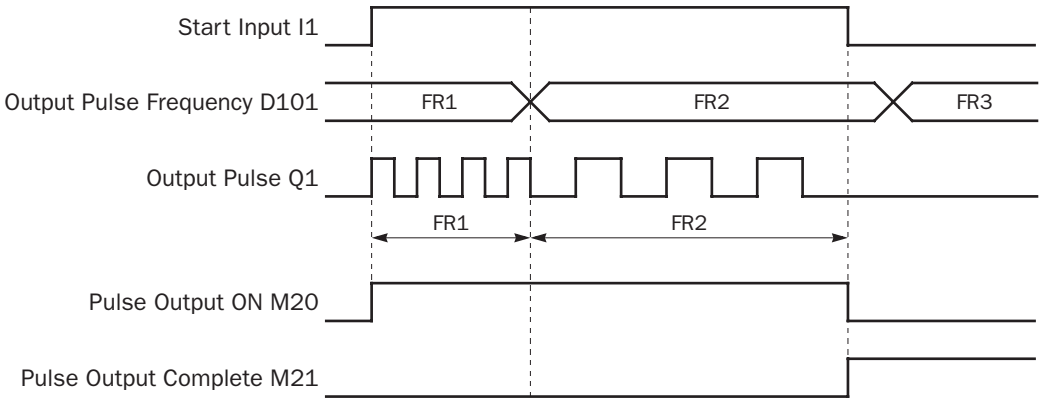


- When input I0 is turned on, PULS1 starts to generate output pulses at the frequency designated by the value stored in data register D201. While the output pulses are sent out from output Q0, internal relay M50 remains on.
- When the quantity of generated output pulses reaches the preset value designated by data registers D203 and D204, PULS1 stops generating output pulses. Then internal relay M50 turns off, and internal relay M51 turns on.
- If the output pulse frequency value in D201 is changed while generating output pulses, the change takes effect in the next scan. When changing the pulse frequency, make sure that the timing of the change is much slower than the output pulse frequency, so that the pulse frequency is changed successfully.
- If input I0 is turned off before reaching the preset value, PULS1 stops generating output pulses immediately, then internal relay M50 turns off and internal relay M51 turns on.

Timing Chart for Disable Pulse Counting

This program demonstrates a timing chart of the PULS2 instruction without pulse counting.

D102 = 0 (disable pulse counting)



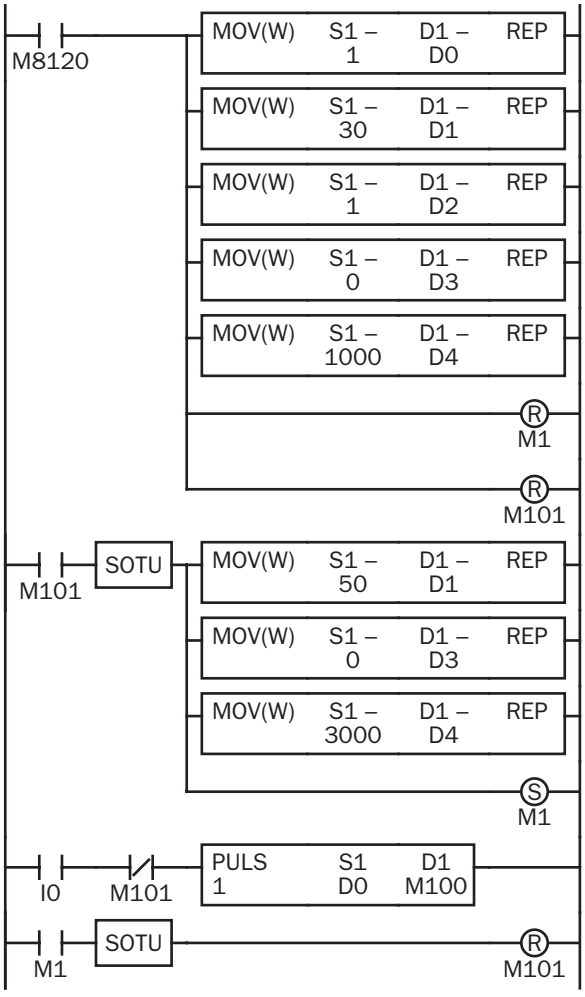
- When input I1 is turned on, PULS2 starts to generate output pulses at the frequency designated by the value stored in data register D101. While the output pulses are sent out from output Q1, internal relay M20 remains on.
- When input I1 is turned off, PULS2 stops generating output pulses immediately, then internal relay M20 turns off and internal relay M21 turns on.
- If the output pulse frequency value in D101 is changed while generating output pulses, the change takes effect in the next scan. When changing the pulse frequency, make sure that the timing of the change is much slower than the output pulse frequency, so that the pulse frequency is changed successfully.

Sample Program: PULS1

This program demonstrates a user program of the PULS1 instruction to generate 1,000 pulses at a frequency of 3 kHz from output Q0, followed by 3,000 pulses at a frequency of 5 kHz.

Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+0	Operation mode	Frequency range 100 to 10,000 Hz	D0 (1)
S1+1	Output pulse frequency	10,000 Hz × 30% (50%) = 3,000 Hz (5,000 Hz)	D1 (30) → (50)
S1+2	Pulse counting	Enable pulse counting	D2 (1)
S1+3	Preset value (high word)	1,000 (3,000)	D3 (0)
S1+4	Preset value (low word)		D4 (1000) → (3000)
S1+5	Current value (high word)	0 to 3,000	D5
S1+6	Current value (low word)		D6
S1+7	Error status		D7
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	M100
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	M101
D1+2	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred (PULS1 only)	M102



M8120 is the initialize pulse special internal relay.

When the CPU starts, five MOV(W) instructions store first-stage parameters to data registers D0 through D4.

D0 (operation mode): 1 (100 to 10,000 Hz)

D1 (output pulse frequency): 30 (10,000 Hz × 30% = 3,000 Hz)

D2 (pulse counting): 1 (enable pulse counting)

D3 (preset value high word): 0

D4 (preset value low word): 1,000

Pulse data update flag M1 is reset (pulse data not updated).

Pulse output complete flag M101 is turned off.

When M101 is turned on, three MOV (W) instructions store second-stage parameters to data registers D1, D3, and D4.

D1 (output pulse frequency): 50 (10,000 Hz × 50% = 5,000 Hz)

D3 (preset value high word): 0

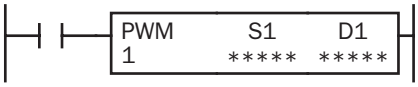
D4 (preset value low word): 3,000

Pulse data update flag M1 is set (pulse data updated).

When start input I0 is turned on, PULS1 starts to generate 3,000Hz output pulses in the first stage.

Pulse output complete M101 is turned off.

PWM1 (Pulse Width Modulation 1)

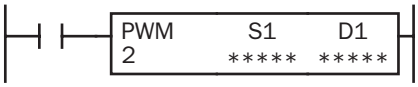


When input is on, the PWM1 instruction generates a pulse output. The output pulse frequency is selected from 6.81, 27.26, or 217.86 Hz, and the output pulse width ratio is determined by source operand S1.

PWM1 sends out output pulses from output Q0.

PWM1 can be programmed to generate a predetermined number of output pulses. When pulse counting is disabled, PWM1 generates output pulses while the start input for the PWM1 instruction remains on.

PWM2 (Pulse Width Modulation 2)



When input is on, the PWM2 instruction generates a pulse output. The output pulse frequency is selected from 6.81, 27.26, or 217.86 Hz, and the output pulse width ratio is determined by source operand S1.

PWM2 sends out output pulses from output Q1.

PWM2 generates output pulses while the start input for the PWM2 instruction remains on. PWM2 cannot be programmed to generate a predetermined number of output pulses.

Note: The PWM1 and PWM2 instructions can be used only once in a user program. When PWM1 or PWM2 is not used, unused output Q0 or Q1 can be used for another pulse instruction or ordinary output.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Control register	—	—	—	—	—	—	X	—	—
D1 (Destination 1)	Status relay	—	—	X	—	—	—	—	—	—

Source operand S1 (control register) uses 8 data registers starting with the operand designated as S1. Data registers D0 through D1292 and D2000 through D7992 can be designated as S1. For details, see below.

Destination operand D1 (status relay) uses 3 internal relays starting with the operand designated as D1. Internal relays M0 through M1270 can be designated as D1. The least significant digit of the internal relay number designated as D1 must be 0. Special internal relays cannot be designated as D1. For details, see page 6-2.

Source Operand S1 (Control Register)

Store appropriate values to data registers starting with the operand designated by S1 before executing the PWM instruction as required, and make sure that the values are within the valid range. Operands S1+5 through S1+7 are for read only.

Operand	Function	Description	R/W
S1+0	Output pulse frequency	0: 6.81 Hz 1: 27.26 Hz 2: 217.86 Hz	R/W
S1+1	Pulse width ratio	1 to 100 (1% to 100% of the period determined by output pulse frequency S1+0)	R/W
S1+2	Pulse counting	0: Disable pulse counting 1: Enable pulse counting (PWM1 only)	R/W
S1+3	Preset value (high word)	1 to 100,000,000 (05F5 E100h) (PWM1 only)	R/W
S1+4	Preset value (low word)		
S1+5	Current value (high word)	1 to 100,000,000 (05F5 E100h) (PWM1 only)	R
S1+6	Current value (low word)		
S1+7	Error status	0 to 5	R

S1+0 Output Pulse Frequency

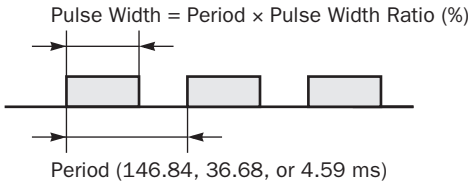
The value stored in the data register designated by operand S1+0 determines the pulse output frequency.

0: 6.81 Hz (146.84 ms period)
1: 27.26 Hz (36.68 ms period)
2: 217.86 Hz (4.59 ms period)

S1+1 Pulse Width Ratio

The value stored in the data register designated by operand S1+1 specifies the pulse width ratio of the pulse output in percent of the period determined by the output pulse frequency selected with S1+0. Valid values for operand S1+1 are 1 through 100.

$$\text{Pulse Width} = \text{Period} \times \frac{\text{Pulse Width Ratio}}{100}$$
$$= \frac{1}{\text{Output Pulse Frequency}} \times \frac{\text{Pulse Width Ratio}}{100}$$



S1+2 Pulse Counting

Pulse counting can be enabled for the PWM1 instruction only. With pulse counting enabled, PWM1 generates a predetermined number of output pulses as designated by operands S1+3 and S1+4. With pulse counting disabled, PWM1 or PWM2 generates output pulses while the start input for the PWM instruction remains on.

0: Disable pulse counting
1: Enable pulse counting (PWM1 only)

When programming PWM2, store 0 to the data register designated by S1+2.

S1+3 Preset Value (High Word)
S1+4 Preset Value (Low Word)

With pulse counting enabled as described above, PWM1 generates a predetermined number of output pulses as designated by operands S1+3 and S1+4. The preset value can be 1 through 100,000,000 (05F5 E100h) stored in two consecutive data registers designated by S1+3 (high word) and S1+4 (low word).

When pulse counting is disabled for PWM1 or when programming PWM2, store 0 to data registers designated by S1+3 and S1+4.

S1+5 Current Value (High Word)
S1+6 Current Value (Low Word)

While the PWM1 instruction is executed, the output pulse count is stored in two consecutive data registers designated by operands S1+5 (high word) and S1+6 (low word). The current value can be 1 through 100,000,000 (05F5 E100h) and is updated in every scan.

S1+7 Error Status

When the start input for the PWM1 or PWM2 instruction is turned on, operand values are checked. When any error is found in the operand values, the data register designated by operand S1+7 stores an error code.

Error Code	Description
0	Normal
1	Output pulse frequency designation error (S1+0 stores other than 0 through 2)
2	Pulse width ratio designation error (S1+1 stores other than 1 through 100)
3	Pulse counting designation error (S1+2 stores other than 0 and 1)
4	Preset value designation error (S1+3 and S1+4 store other than 1 through 100,000,000)
5	Invalid pulse counting designation for PWM2 (S1+2 stores 1)

Destination Operand D1 (Status Relay)

Three internal relays starting with the operand designated by D1 indicate the status of the PWM instruction. These operands are for read only.

Operand	Function	Description	R/W
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	R
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	R
D1+2	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred (PWM1 only)	R

D1+0 Pulse Output ON

The internal relay designated by operand D1+0 remains on while the PWM instruction generates output pulses. When the start input for the PWM instruction is turned off or when the PWM1 instruction has completed generating a predetermined number of output pulses, the internal relay designated by operand D1+0 turns off.

D1+1 Pulse Output Complete

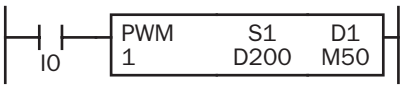
The internal relay designated by operand D1+1 turns on when the PWM1 instruction has completed generating a predetermined number of output pulses or when either PWM instruction is stopped to generate output pulses. When the start input for the PWM instruction is turned on, the internal relay designated by operand D1+1 turns off.

D1+2 Pulse Output Overflow

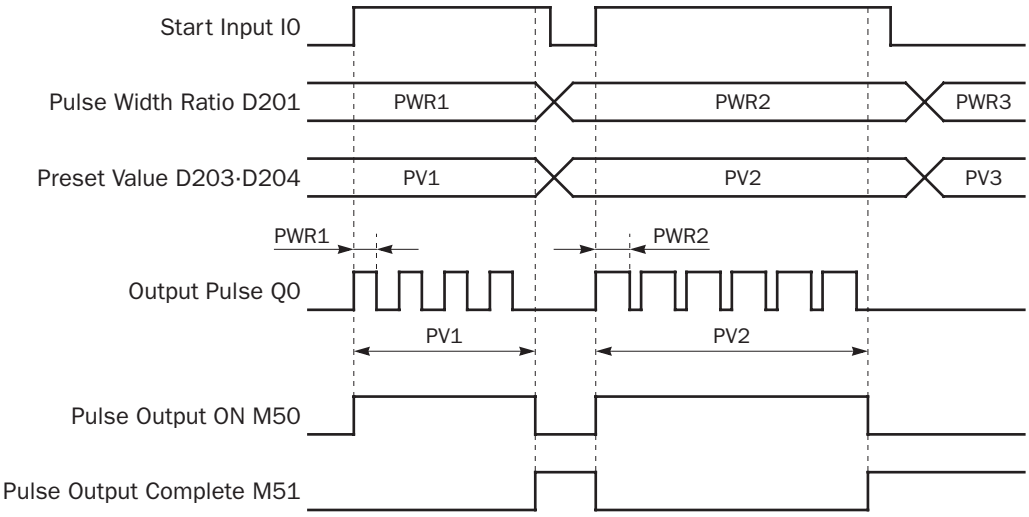
The internal relay designated by operand D1+2 turns on when the PWM1 instruction has generated more than the predetermined number of output pulses. When the start input for the PWM instruction is turned on, the internal relay designated by operand D1+2 turns off.

Timing Chart for Enable Pulse Counting

This program demonstrates a timing chart of the PWM1 instruction when pulse counting is enabled.



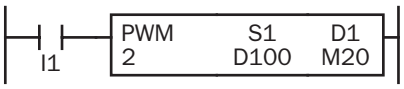
D202 = 1 (enable pulse counting)



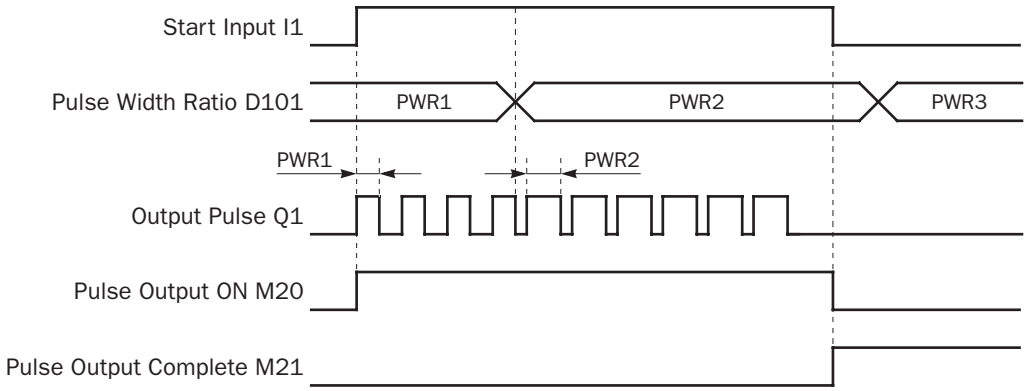
- When input I0 is turned on, PWM1 starts to generate output pulses at the frequency designated by the value stored in data register D200. The pulse width is determined by the value stored in data register D201. While the output pulses are sent out from output Q0, internal relay M50 remains on.
- When the quantity of generated output pulses reaches the preset value designated by data registers D203 and D204, PWM1 stops generating output pulses. Then internal relay M50 turns off, and internal relay M51 turns on.
- If the pulse width ratio value in D201 is changed while generating output pulses, the change takes effect in the next scan. When changing the pulse width ratio, make sure that the timing of the change is much slower than the output pulse frequency, so that the pulse width ratio is changed successfully.
- If input I0 is turned off before reaching the preset value, PWM1 stops generating output pulses immediately, then internal relay M50 turns off and internal relay M51 turns on.

Timing Chart for Disable Pulse Counting

This program demonstrates a timing chart of the PWM2 instruction without pulse counting.



D102 = 0 (disable pulse counting)



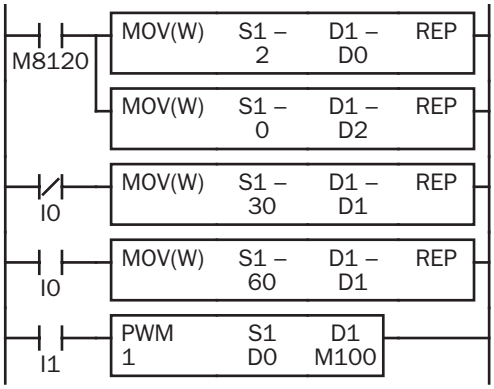
- When input I1 is turned on, PWM2 starts to generate output pulses at the frequency designated by the value stored in data register D100. The pulse width is determined by the value stored in data register D101. While the output pulses are sent out from output Q1, internal relay M20 remains on.
- When input I1 is turned off, PWM2 stops generating output pulses immediately, then internal relay M20 turns off and internal relay M21 turns on.
- If the pulse width ratio value in D101 is changed while generating output pulses, the change takes effect in the next scan. When changing the pulse width ratio, make sure that the timing of the change is much slower than the output pulse frequency, so that the pulse width ratio is changed successfully.

Sample Program: PWM1

This program demonstrates a user program of the PWM1 instruction to generate pulses from output Q0, with an ON/OFF ratio of 30% while input I0 is off or 60% when input I0 is on.

Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+0	Output pulse frequency	217.86 Hz	D0 (2)
S1+1	Pulse width ratio	30% or 60%	D1 (30 or 60)
S1+2	Pulse counting	Disable pulse counting	D2 (0)
S1+3	Preset value (high word)	Not used	D3
S1+4	Preset value (low word)		D4
S1+5	Current value (high word)	Not used	D5
S1+6	Current value (low word)		D6
S1+7	Error status		D7
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	M100
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	M101
D1+2	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred (PWM1 only)	M102



M8120 is the initialize pulse special internal relay.

When the CPU starts, MOV(W) instructions store parameters to data registers D0 and D2.

D0 (output pulse frequency): 2 (217.86 Hz)

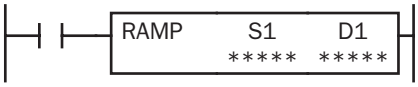
D2 (pulse counting): 0 (disable pulse counting)

When input I0 is off, D1 (pulse width ratio) stores 30 (30%).

When input I0 is on, D1 (pulse width ratio) stores 60 (60%).

When input I1 is on, PWM1 generates output pulses of a 30% or 60% pulse width ratio from output Q0 depending whether input I0 is off or on, respectively.

RAMP (Ramp Control)



When input is on, the RAMP instruction sends out a predetermined number of output pulses whose frequency changes in a trapezoidal pattern determined by source operand S1. After starting the RAMP instruction, the output pulse frequency increases linearly to a predetermined constant value, remains constant at this value for some time, and then decreases linearly to the original value.

The frequency change rate or the frequency change time can be selected for acceleration and deceleration of the movement.

When input is off, the pulse output remains off. When input is turned on again, the RAMP instruction starts a new cycle of generating output pulses.

Note: The RAMP instruction can be used only once in a user program. When RAMP is used with reversible control disabled, unused output Q1 can be used for another pulse instruction PULS2, PWM2, or ZRN2 or ordinary output.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	X	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Control register	—	—	—	—	—	—	X	—	—
D1 (Destination 1)	Status relay	—	—	X	—	—	—	—	—	—

Source operand S1 (control register) uses 11 data registers starting with the operand designated as S1. Data registers D0 through D1289 and D2000 through D7989 can be designated as S1. For details, see the following pages.

Destination operand D1 (status relay) uses 4 internal relays starting with the operand designated as D1. Internal relays M0 through M1270 can be designated as D1. The least significant digit of the internal relay number designated as D1 must be 0, otherwise the RAMP instruction does not operate correctly. Special internal relays cannot be designated as D1. For details, see page 6-2.

Source Operand S1 (Control Register)

Store appropriate values to data registers starting with the operand designated as S1 before executing the RAMP instruction as required, and make sure that the values are within the valid range. Operands S1+8 through S1+10 are for read only.

Operand	Function	Description	R/W
S1+0	Operation mode	0: 10 to 1,000 Hz 1: 100 to 10,000 Hz 2: 1,000 to 20,000 Hz 3: 10 to 20,000 Hz (upgraded CPU only)	R/W
S1+1	Steady pulse frequency	When S1+0 (operation mode) = 0 or 1: 1 to 100 (%) (1% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 2: 1 to 20 (×5%) (5% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 3: 10 to 20,000 (Hz)	R/W
S1+2	Initial pulse frequency	When S1+0 (operation mode) = 0 or 1: 1 to 100 (%) (1% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 2: 1 to 20 (×5%) (5% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 3: 10 to 20,000 (Hz)	R/W
S1+3	Frequency change rate	When S1+0 (operation mode) = 0 or 1: 1 to 100 (%) (1% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (operation mode) = 2: 1 to 20 (×5%) (5% to 100% of the maximum frequency of selected mode S1+0)	R/W
	Frequency change time	When S1+0 (operation mode) = 3: 10 to 10,000 (ms)	

Operand	Function	Description	R/W
S1+4	Reversible control enable	0: Reversible control disabled 1: Reversible control (single-pulse output) 2: Reversible control (dual-pulse output)	R/W
S1+5	Control direction	0: Forward 1: Reverse	R/W
S1+6	Preset value (high word)	1 to 100,000,000 (05F5 E100h)	R/W
S1+7	Preset value (low word)		
S1+8	Current value (high word)	1 to 100,000,000 (05F5 E100h)	R
S1+9	Current value (low word)		
S1+10	Error status	0 to 10	R

S1+0 Operation Mode

The value stored in the data register designated by operand S1+0 determines the frequency range of the pulse output.

0: 10 to 1,000 Hz

1: 100 to 10,000 Hz

2: 1,000 to 20,000 Hz

3: 10 to 20,000 Hz (upgraded CPU only)

S1+1 Steady Pulse Frequency

When S1+0 is set to 0 through 2, the value stored in the data register designated by operand S1+1 specifies the frequency of the steady pulse output in percent of the maximum of the frequency range selected by S1+0. When S1+0 is set to 0 (10 to 1,000 Hz) or 1 (100 to 10,000 Hz), valid values for operand S1+1 are 1 through 100, thus the steady pulse frequency can be 10 to 1,000 Hz or 100 to 10,000 Hz, respectively. When S1+0 is set to 2 (1,000 to 20,000 Hz), valid values for operand S1+1 are 1 through 20 and the S1+1 value multiplied by 5 determines the steady pulse frequency, thus the steady pulse frequency can be 1,000 to 20,000 Hz.

When S1+0 is set to 3, the value stored in the data register designated by operand S1+1 determines the frequency of the steady pulse output directly. Valid values are 10 through 20,000.

Operation Mode	Steady Pulse Frequency (Hz)
0 or 1	Maximum frequency (Hz) selected by S1+0 × S1+1 value (%)
2	Maximum frequency (Hz) selected by S1+0 × S1+1 value (×5%)
3	Steady pulse frequency (Hz) selected by S1+1

S1+2 Initial Pulse Frequency

When S1+0 is set to 0 through 2, the value stored in the data register designated by operand S1+2 specifies the frequency of the initial pulse output in percent of the maximum of the frequency range selected by S1+0. When S1+0 is set to 0 (10 to 1,000 Hz) or 1 (100 to 10,000 Hz), valid values for operand S1+2 are 1 through 100, thus the initial pulse frequency can be 10 to 1,000 Hz or 100 to 10,000 Hz, respectively. When S1+0 is set to 2 (1,000 to 20,000 Hz), valid values for operand S1+2 are 1 through 20 and the S1+2 value multiplied by 5 determines the initial pulse frequency, thus the initial pulse frequency can be 1,000 to 20,000 Hz.

When S1+0 is set to 3, the value stored in the data register designated by operand S1+2 determines the frequency of the initial pulse output directly. Valid values are 10 through 20,000.

Operation Mode	Initial Pulse Frequency (Hz)
0 or 1	Maximum frequency (Hz) selected by S1+0 × S1+2 value (%)
2	Maximum frequency (Hz) selected by S1+0 × S1+2 value (×5%)
3	Initial pulse frequency (Hz) selected by S1+1

S1+3 Frequency Change Rate / Frequency Change Time

When S1+0 is set to 0 through 2, the value stored in the data register designated by operand S1+3 specifies the rate of pulse output frequency change for a period of 10 ms in percent of the maximum of the frequency range selected by S1+0. When S1+0 is set to 0 (10 to 1,000 Hz) or 1 (100 to 10,000 Hz), valid values for operand S1+3 are 1 through 100, thus the frequency change rate can be 10 to 1,000 Hz or 100 to 10,000 Hz, respectively. When S1+0 is set to 2 (1,000 to 20,000 Hz), valid values for operand S1+3 are 1 through 20 and the S1+3 value multiplied by 5 determines the frequency change rate, thus the frequency change rate can be 1,000 to 20,000 Hz.

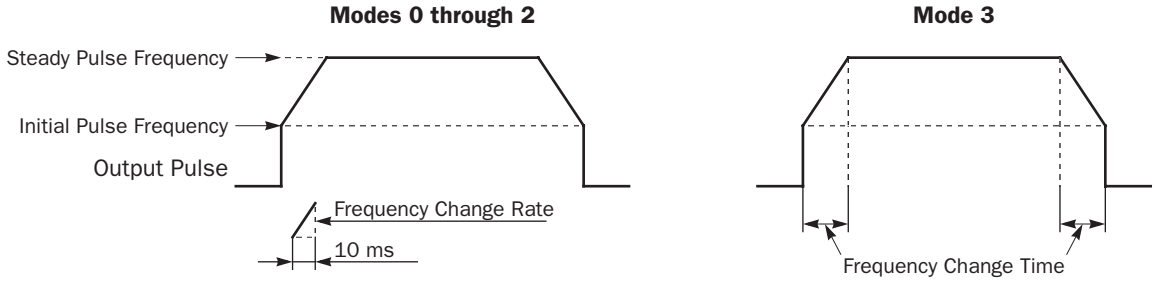
When S1+0 is set to 3, the value stored in the data register designated by operand S1+3 determines the frequency change time. Valid values are 10 through 10,000 in increments of 10, thus the frequency change time can be 10 to 10,000 ms.

Mode 0 or 1: Frequency change rate in 10 ms (Hz) = Maximum frequency (Hz) selected by S1+0 × S1+3 value (%)

Mode 2: Frequency change rate in 10 ms (Hz) = Maximum frequency (Hz) selected by S1+0 × S1+3 value (×5%)

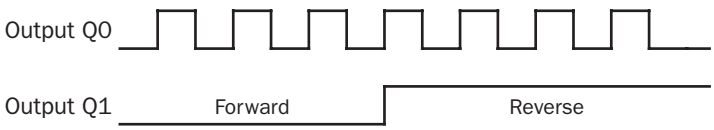
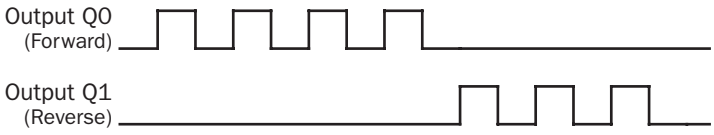
Mode 3: Frequency change time (ms) = Frequency change time (ms) selected by S1+3

The same frequency change rate and frequency change time apply to the accelerating and decelerating periods of the trapezoidal frequency change pattern.



S1+4 Reversible Control Enable

The value stored in the data register designated by operand S1+4 specifies one of the output modes.

S1+4 Value	Reversible Control	Description
0	Reversible control disabled	Output Q0 generates output pulses; used for single-direction control.  Output Q1 can be used for PULS2, PWM2, ZRN2, or ordinary output.
1	Reversible control (Single-pulse output)	Output Q0 generates output pulses, and output Q1 generates a direction control signal.  Output Q1 turns on or off depending on the value stored in data register designated by operand S1+5 (control direction): 0 for forward or 1 for reverse.
2	Reversible control (Dual-pulse output)	Output Q0 generates forward output pulses, and output Q1 generates reverse output pulses.  Output Q0 or Q1 generates output pulses alternately depending on the value stored in data register designated by operand S1+5 (control direction): 0 for forward or 1 for reverse.

If the value stored in the data register designated by operand S1+4 is changed after the start input for the RAMP instruction has been turned on, the change can take effect only after the CPU starts again.

S1+5 Control Direction

When S1+4 is set to 1 or 2 to enable reversible control, the value stored in the data register designated by operand S1+5 specifies the control direction.

0: Forward

1: Reverse

S1+6 Preset Value (High Word)
S1+7 Preset Value (Low Word)

The RAMP instruction generates a predetermined number of output pulses as designated by operands S1+6 and S1+7. The preset value can be 1 through 100,000,000 (05F5 E100h) stored in two consecutive data registers designated by S1+6 (high word) and S1+7 (low word).

S1+8 Current Value (High Word)
S1+9 Current Value (Low Word)

While the RAMP instruction is executed to generate output pulses from output Q0 or Q1, the output pulse count is stored in two consecutive data registers designated by operands S1+8 (high word) and S1+9 (low word). The current value can be 1 through 100,000,000 (05F5 E100h) and is updated in every scan.

S1+10 Error Status

When the start input for the RAMP instruction is turned on, operand values are checked. When any error is found in the operand values, the data register designated by operand S1+10 stores an error code.

Error Code	Operation Modes 0 through 2	Operation Mode 3
0	Normal	
1	Operation mode designation error (S1+0 stores other than 0 through 2)	Operation mode designation error (S1+0 stores other than 0 through 3)
2	Initial pulse frequency designation error (S1+2 stores other than 1 through 100)	Initial pulse frequency designation error (S1+2 stores other than 10 through 20,000)
3	Preset value designation error (S1+6 and S1+7 store other than 1 through 100,000,000) The number of pulses for the frequency change area calculated from the steady pulse frequency (S1+1), initial pulse frequency (S1+2), and frequency change rate (S1+3) is 0.	Preset value designation error (S1+6 and S1+7 store other than 1 through 100,000,000)
4	Steady pulse frequency designation error (S1+1 stores other than 1 through 100)	Steady pulse frequency designation error (S1+1 stores other than 10 through 20,000)
5	Frequency change rate designation error (S1+3 stores other than 1 through 100)	Frequency change time designation error (S1+3 stores other than 10 through 10,000)
6	Reversible control enable designation error (S1+4 stores other than 0 through 2)	
7	Control direction designation error (S1+5 stores other than 0 and 1)	
8	The number of pulses for the frequency change areas calculated from the steady pulse frequency (S1+1), initial pulse frequency (S1+2), and frequency change rate (S1+3) exceeds the preset value (S1+6/7) of the total output pulses. To correct this error, reduce the value of the steady pulse frequency (S1+1) or initial pulse frequency (S1+2), or increase the frequency change rate (S1+3).	
9	The initial pulse frequency (S1+2) is larger than the steady pulse frequency (S1+1). Reduce the initial pulse frequency (S1+2) to a value smaller than the steady pulse frequency (S1+1).	
10	The frequency change rate (S1+3) is larger than the difference between the initial pulse frequency (S1+2) and the steady pulse frequency (S1+1). Reduce the frequency change rate (S1+3) or the initial pulse frequency (S1+2).	—

Destination Operand D1 (Status Relay)

Four internal relays starting with the operand designated by D1 indicate the status of the RAMP instruction. These operands are for read only.

Operand	Function	Description	R/W
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	R
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	R
D1+2	Pulse output status	0: Steady pulse output 1: Changing output pulse frequency	R
D1+3	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred	R

D1+0 Pulse Output ON

The internal relay designated by operand D1+0 remains on while the RAMP instruction generates output pulses. When the start input for the RAMP instruction is turned off or when the RAMP instruction has completed generating a predetermined number of output pulses, the internal relay designated by operand D1+0 turns off.

D1+1 Pulse Output Complete

The internal relay designated by operand D1+1 turns on when the RAMP instruction has completed generating a predetermined number of output pulses or when the RAMP instruction is stopped to generate output pulses. When the start input for the RAMP instruction is turned on, the internal relay designated by operand D1+1 turns off.

D1+2 Pulse Output Status

The internal relay designated by operand D1+2 turns on while the output pulse frequency is increased or decreased, and turns off when the output pulse frequency reaches the steady pulse frequency (S1+1). While the pulse output is off, the internal relay designated by operand D1+2 remains off.

D1+3 Pulse Output Overflow

The internal relay designated by operand D1+3 turns on when the RAMP instruction has generated more than the predetermined number of output pulses (S1+6/7). When an overflow occurs, the current value (S1+8/9) stops at the preset value (S1+6/7). When the start input for the RAMP instruction is turned on, the internal relay designated by operand D1+3 turns off.

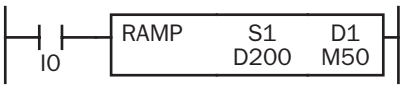
Special Data Register for Pulse Outputs

Upgraded CPU modules have two additional special data registers to store the current frequency of pulse outputs.

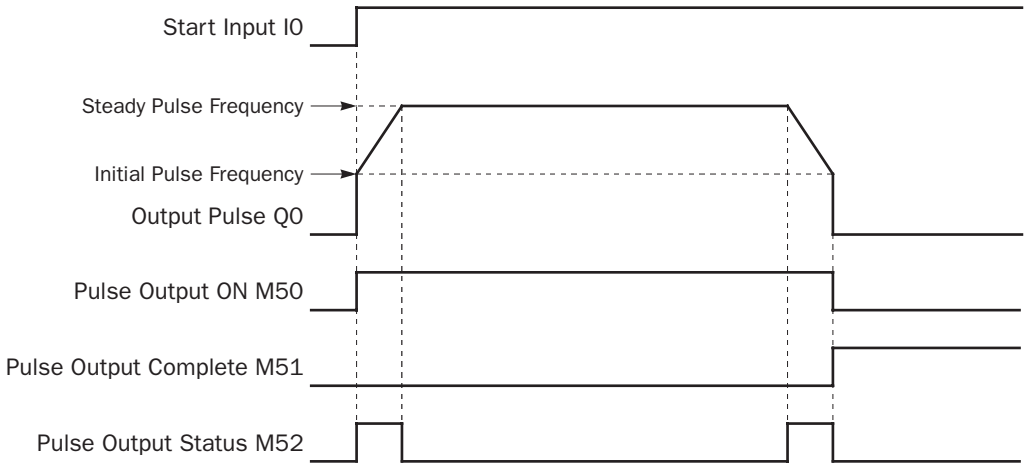
Allocation No.	Function	Description
D8055	Current Pulse Frequency of PULS1 or RAMP (Q0)	While the PULS1 or RAMP instruction is executed, D8055 stores the current pulse frequency of output Q0. The value is updated every scan.
D8056	Current Pulse Frequency of PULS2 or RAMP (Q1)	While the PULS2 or RAMP (reversible control dual-pulse output) instruction is executed, D8056 stores the current pulse frequency of output Q1. The value is updated every scan.

Timing Chart for Reversible Control Disabled

This program demonstrates a timing chart of the RAMP instruction when reversible control is disabled.



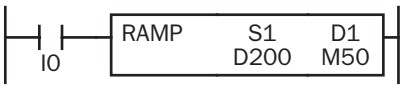
D204 = 0 (reversible control disabled)



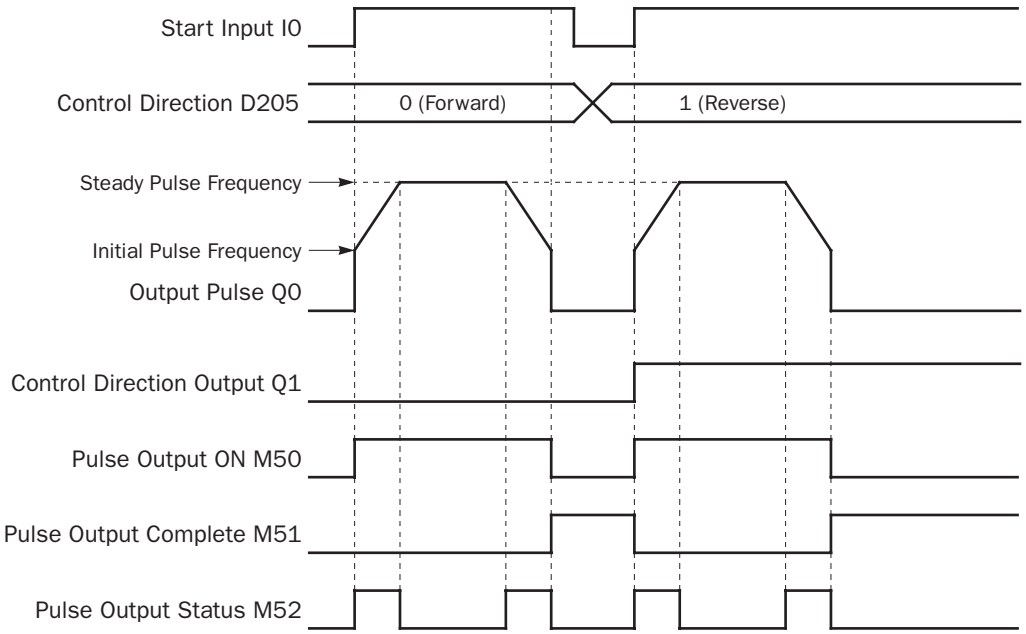
- When input I0 is turned on, RAMP generates output pulses starting at the initial frequency designated by the value stored in data register D202. While the output pulses are sent out from output Q0, internal relay M50 remains on.
- Operation modes 0 through 2: The pulse frequency increases according to the frequency change rate value stored in data register D203.
- Operation mode 3: The pulse frequency increases as long as the frequency change time stored in data register D203.
- While the output pulse frequency is on the increase, internal relay M52 remains on.
- When the output pulse frequency reaches the steady pulse frequency designated by the value stored in data register D201, internal relay M52 turns off. When the output pulse frequency starts to decrease, internal relay M52 turns on again.
- When the quantity of generated output pulses reaches the preset value designated by data registers D206 and D207, RAMP stops generating output pulses. Then internal relay M50 and M52 turn off, and internal relay M51 turns on.
- If the parameter values in D200 through D207 (except for D204) are changed while generating output pulses, the change takes effect when start input I0 is turned on for the next cycle.
- If the value stored in D204 is changed after start input I0 has been turned on, the change can take effect only after the CPU starts again.
- If start input I0 is turned off before reaching the preset value, RAMP stops generating output pulses immediately, then internal relay M50 turns off and internal relay M51 turns on. When input I0 is turned on again, RAMP restarts to generate output pulses for another cycle, starting at the initial pulse frequency.

Timing Chart for Reversible Control with Single Pulse Output

This program demonstrates a timing chart of the RAMP instruction when reversible control is enabled with single pulse output.



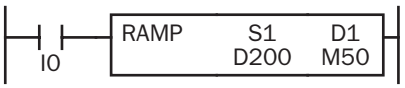
D204 = 1 (reversible control with single pulse output)



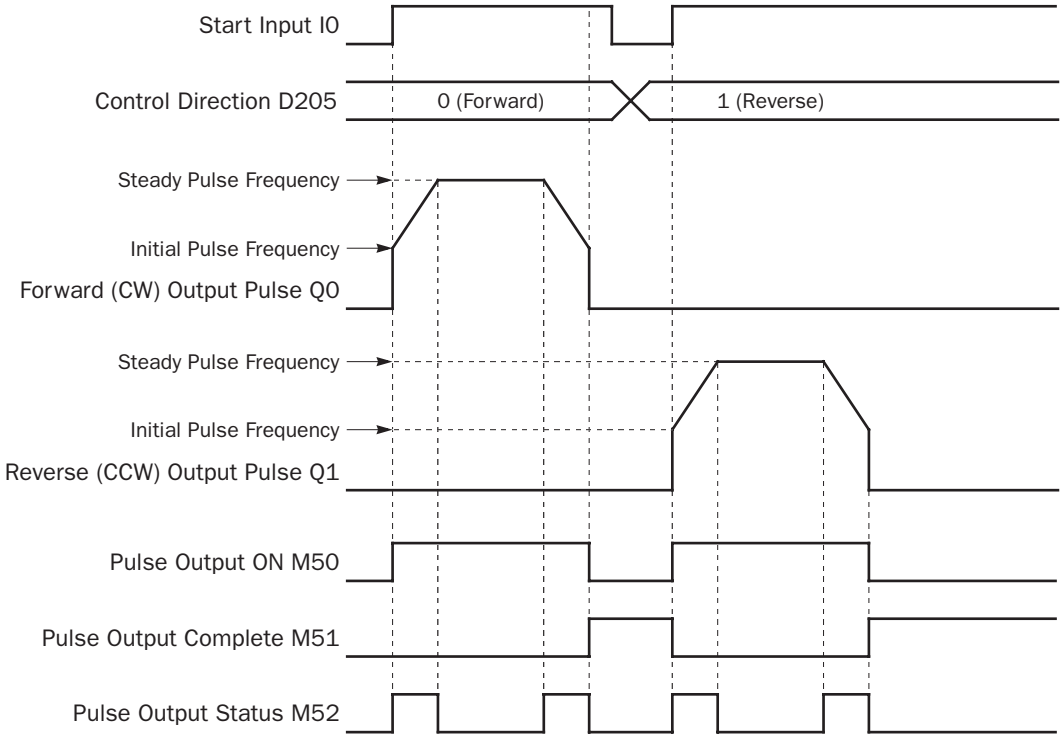
- When input I0 is turned on, RAMP generates output pulses starting at the initial frequency designated by the value stored in data register D202. While the output pulses are sent out from output Q0, internal relay M50 remains on.
- Operation modes 0 through 2: The pulse frequency increases according to the frequency change rate value stored in data register D203.
- Operation mode 3: The pulse frequency increases as long as the frequency change time stored in data register D203.
- While the output pulse frequency is on the increase, internal relay M52 remains on.
- Depending on the control direction designated by the value stored in data register D205, control direction output Q1 turns off or on while D205 stores 0 (forward) or 1 (reverse), respectively.
- When the output pulse frequency reaches the steady pulse frequency designated by the value stored in data register D201, internal relay M52 turns off. When the output pulse frequency starts to decrease, internal relay M52 turns on again.
- When the quantity of generated output pulses reaches the preset value designated by data registers D206 and D207, RAMP stops generating output pulses. Then internal relay M50 and M52 turn off, and internal relay M51 turns on.
- If the parameter values in D200 through D207 (except for D204) are changed while generating output pulses, the change takes effect when start input I0 is turned on for the next cycle.
- If the value stored in D204 is changed after start input I0 has been turned on, the change can take effect only after the CPU starts again.
- If start input I0 is turned off before reaching the preset value, RAMP stops generating output pulses immediately, then internal relay M50 turns off and internal relay M51 turns on. When input I0 is turned on again, RAMP restarts to generate output pulses for another cycle, starting at the initial pulse frequency.

Timing Chart for Reversible Control with Dual Pulse Output

This program demonstrates a timing chart of the RAMP instruction when reversible control is enabled with dual pulse output.



D204 = 2 (reversible control with dual pulse output)



- When input I0 is turned on, RAMP generates output pulses starting at the initial frequency designated by the value stored in data register D202. While the output pulses are sent out from output Q0 or Q1, internal relay M50 remains on.
- Operation modes 0 through 2: The pulse frequency increases according to the frequency change rate value stored in data register D203.
- Operation mode 3: The pulse frequency increases as long as the frequency change time stored in data register D203.
- While the output pulse frequency is on the increase, internal relay M52 remains on.
- Depending on the control direction designated by the value stored in data register D205, output Q0 or Q1 sends out output pulses while D205 stores 0 (forward) or 1 (reverse), respectively.
- When the output pulse frequency reaches the steady pulse frequency designated by the value stored in data register D201, internal relay M52 turns off. When the output pulse frequency starts to decrease, internal relay M52 turns on again.
- When the quantity of generated output pulses reaches the preset value designated by data registers D206 and D207, RAMP stops generating output pulses. Then internal relay M50 and M52 turn off, and internal relay M51 turns on.
- If the parameter values in D200 through D207 (except for D204) are changed while generating output pulses, the change takes effect when start input I0 is turned on for the next cycle.
- If the value stored in D204 is changed after start input I0 has been turned on, the change can take effect only after the CPU starts again.
- If start input I0 is turned off before reaching the preset value, RAMP stops generating output pulses immediately, then internal relay M50 turns off and internal relay M51 turns on. When input I0 is turned on again, RAMP restarts to generate output pulses for another cycle, starting at the initial pulse frequency.

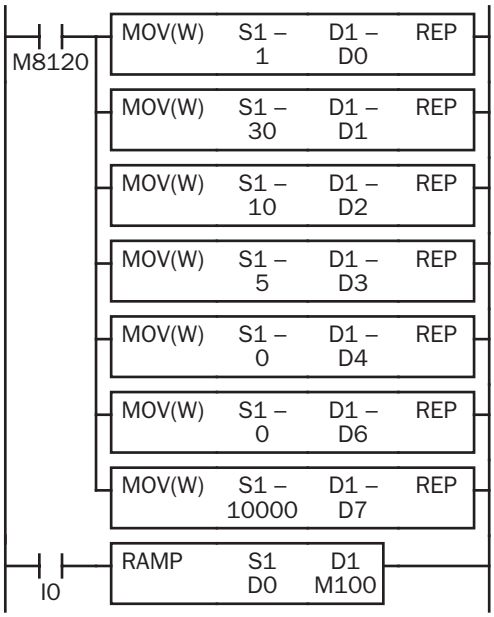
Sample Program: RAMP — Reversible Control Disabled

This program demonstrates a user program of the RAMP instruction to generate 10,000 pulses from output Q0.

Steady pulse frequency:	3,000 Hz
Initial pulse frequency:	1,000 Hz
Frequency change rate:	500 Hz / 10 ms
Reversible control enable:	Reversible control disabled
Preset value:	10,000 pulses total

Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+0	Operation mode	Frequency range 100 to 10,000 Hz	D0 (1)
S1+1	Steady pulse frequency	10,000 Hz × 30% = 3,000 Hz	D1 (30)
S1+2	Initial pulse frequency	10,000 Hz × 10% = 1,000 Hz	D2 (10)
S1+3	Frequency change rate	10,000 Hz × 5% = 500 Hz	D3 (5)
S1+4	Reversible control enable	Reversible control disabled	D4 (0)
S1+5	Control direction	Not used (no effect)	D5
S1+6	Preset value (high word)	10,000	D6 (0)
S1+7	Preset value (low word)		D7 (10000)
S1+8	Current value (high word)	0 to 10,000	D8
S1+9	Current value (low word)		D9
S1+10	Error status		D10
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	M100
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	M101
D1+2	Pulse output status	0: Steady pulse output 1: Changing output pulse frequency	M102
D1+3	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred	M103



M8120 is the initialize pulse special internal relay.

When the CPU starts, seven MOV(W) instructions store parameters to data registers D0 through D7.

D0 (operation mode): 1 (100 to 10,000 Hz)

D1 (steady pulse frequency): 30 (10,000 Hz × 30% = 3,000 Hz)

D2 (initial pulse frequency): 10 (10,000 Hz × 10% = 1,000 Hz)

D3 (frequency change rate): 5 (10,000 Hz × 5% = 500 Hz)

D4 (reversible control enable): 0 (reversible control disabled)

D6 (preset value high word): 0

D7 (preset value low word): 10,000

When start input I0 is turned on, RAMP starts to generate 10,000 output pulses.

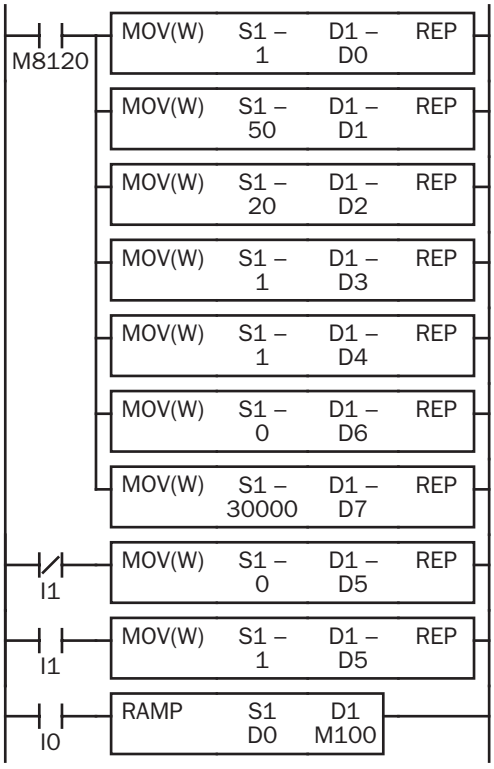
Sample Program: RAMP — Reversible Control with Single Pulse Output

This program demonstrates a user program of the RAMP instruction to generate 30,000 pulses from output Q0. Control direction output Q1 turns off or on while input I1 is off or on to indicate the forward or reverse direction, respectively.

Steady pulse frequency:	5,000 Hz
Initial pulse frequency:	2,000 Hz
Frequency change rate:	100 Hz / 10 ms
Reversible control enable:	Reversible control with single pulse output
Preset value:	30,000 pulses total

Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+0	Operation mode	Frequency range 100 to 10,000 Hz	D0 (1)
S1+1	Steady pulse frequency	10,000 Hz × 50% = 5,000 Hz	D1 (50)
S1+2	Initial pulse frequency	10,000 Hz × 20% = 2,000 Hz	D2 (20)
S1+3	Frequency change rate	10,000 Hz × 1% = 100 Hz	D3 (1)
S1+4	Reversible control enable	Reversible control with single output	D4 (1)
S1+5	Control direction	0 (forward) or 1 (reverse)	D5 (0 or 1)
S1+6	Preset value (high word)	30,000	D6 (0)
S1+7	Preset value (low word)		D7 (30000)
S1+8	Current value (high word)	0 to 30,000	D8
S1+9	Current value (low word)		D9
S1+10	Error status		D10
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	M100
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	M101
D1+2	Pulse output status	0: Steady pulse output 1: Changing output pulse frequency	M102
D1+3	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred	M103



M8120 is the initialize pulse special internal relay.

When the CPU starts, seven MOV(W) instructions store parameters to data registers D0 through D7.

D0 (operation mode): 1 (100 to 10,000 Hz)

D1 (steady pulse frequency): 50 (10,000 Hz × 50% = 5,000 Hz)

D2 (initial pulse frequency): 20 (10,000 Hz × 20% = 2,000 Hz)

D3 (frequency change rate): 1 (10,000 Hz × 1% = 100 Hz)

D4 (reversible control enable): 1 (reversible control with single output)

D6 (preset value high word): 0

D7 (preset value low word): 30,000

When input I1 is off, D5 (control direction) stores 0 (forward).

When input I1 is on, D5 (control direction) stores 1 (reverse).

When start input I0 is turned on, RAMP starts to generate 30,000 output pulses.

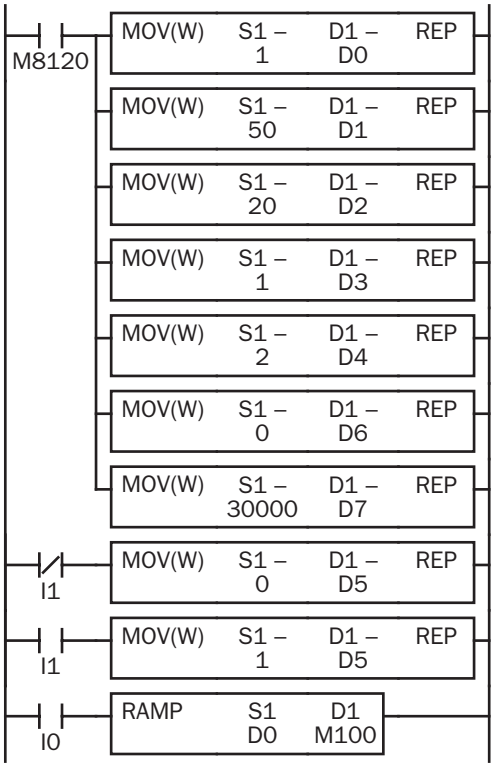
Sample Program: RAMP — Reversible Control with Dual Pulse Output

This program demonstrates a user program of the RAMP instruction to generate 30,000 pulses from output Q0 (forward pulse) or Q1 (reverse pulse) while input I1 is off or on, respectively.

Steady pulse frequency:	5,000 Hz
Initial pulse frequency:	2,000 Hz
Frequency change rate:	100 Hz / 10 ms
Reversible control enable:	Reversible control with dual pulse output
Preset value:	30,000 pulses total

Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+0	Operation mode	Frequency range 100 to 10,000 Hz	D0 (1)
S1+1	Steady pulse frequency	10,000 Hz × 50% = 5,000 Hz	D1 (50)
S1+2	Initial pulse frequency	10,000 Hz × 20% = 2,000 Hz	D2 (20)
S1+3	Frequency change rate	10,000 Hz × 1% = 100 Hz	D3 (1)
S1+4	Reversible control enable	Reversible control with dual output	D4 (2)
S1+5	Control direction	0 (forward) or 1 (reverse)	D5 (0 or 1)
S1+6	Preset value (high word)	30,000	D6 (0)
S1+7	Preset value (low word)		D7 (30000)
S1+8	Current value (high word)	0 to 30,000	D8
S1+9	Current value (low word)		D9
S1+10	Error status		D10
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	M100
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	M101
D1+2	Pulse output status	0: Steady pulse output 1: Changing output pulse frequency	M102
D1+3	Pulse output overflow	0: Overflow not occurred 1: Overflow occurred	M103



M8120 is the initialize pulse special internal relay.

When the CPU starts, seven MOV(W) instructions store parameters to data registers D0 through D7.

D0 (operation mode): 1 (100 to 10,000 Hz)

D1 (steady pulse frequency): 50 (10,000 Hz × 50% = 5,000 Hz)

D2 (initial pulse frequency): 20 (10,000 Hz × 20% = 2,000 Hz)

D3 (frequency change rate): 1 (10,000 Hz × 1% = 100 Hz)

D4 (reversible control enable): 2 (reversible control with dual output)

D6 (preset value high word): 0

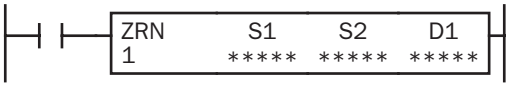
D7 (preset value low word): 30,000

When input I1 is off, D5 (control direction) stores 0 (forward).

When input I1 is on, D5 (control direction) stores 1 (reverse).

When start input I0 is turned on, RAMP starts to generate 30,000 output pulses.

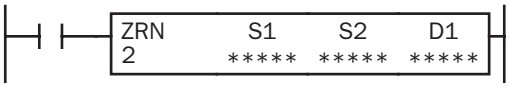
ZRN1 (Zero Return 1)



When input is on, the ZRN1 instruction sends out a pulse output of a predetermined high frequency from output Q0. When a deceleration input turns on, the output frequency decreases to a creep frequency. When the deceleration input turns off, the ZRN1 instruction stops generating output pulses.

The output pulse width ratio is fixed at 50%.

ZRN2 (Zero Return 2)



When input is on, the ZRN2 instruction sends out a pulse output of a predetermined high frequency from output Q1. When a deceleration input turns on, the output frequency decreases to a creep frequency. When the deceleration input turns off, the ZRN2 instruction stops generating output pulses.

The output pulse width ratio is fixed at 50%.

Note: The ZRN1 and ZRN2 instructions can be used only once in a user program. When ZRN1 or ZRN2 is not used, unused output Q0 or Q1 can be used for another pulse instruction or ordinary output.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
S1 (Source 1)	Control register	—	—	—	—	—	—	X	—	—
S2 (Source 2)	Deceleration input	X	—	▲	—	—	—	—	—	—
D1 (Destination 1)	Status relay	—	—	▲	—	—	—	—	—	—

Source operand S1 (control register) uses 5 data registers starting with the operand designated as S1. Data registers D0 through D1295 and D2000 through D7995 can be designated as S1. For details, see the following pages.

Source operand S2 (deceleration input) can designate inputs I0 through I307 and internal relays M0 through M1277. Special internal relays cannot be designated as S2.

Destination operand D1 (status relay) uses 2 internal relays starting with the operand designated as D1. Internal relays M0 through M1270 can be designated as D1. Special internal relays cannot be designated as D1. For details, see page 6-2.

Source Operand S1 (Control Register)

Store appropriate values to data registers starting with the operand designated by S1 before executing the ZRN instruction as required, and make sure that the values are within the valid range. Operand S1+4 is for read only.

Operand	Function	Description	R/W
S1+0	Initial operation mode	0: 10 to 1,000 Hz 1: 100 to 10,000 Hz 2: 1,000 to 20,000 Hz	R/W
S1+1	Initial pulse frequency	When S1+0 (initial operation mode) = 0 or 1: 1 to 100 (%) (1% to 100% of the maximum frequency of selected mode S1+0) When S1+0 (initial operation mode) = 2: 1 to 20 (×5%) (5% to 100% of the maximum frequency of selected mode S1+0)	R/W
S1+2	Creep operation mode	0: 10 to 1,000 Hz 1: 100 to 10,000 Hz 2: 1,000 to 20,000 Hz	R/W
S1+3	Creep pulse frequency	When S1+2 (creep operation mode) = 0 or 1: 1 to 100 (%) (1% to 100% of the maximum frequency of selected mode S1+2) When S1+2 (creep operation mode) = 2: 1 to 20 (×5%) (5% to 100% of the maximum frequency of selected mode S1+2)	R/W
S1+4	Error status	0 to 2	R

S1+0 Initial Operation Mode

The value stored in the data register designated by operand S1+0 determines the frequency range of the high-frequency initial pulse output.

0: 10 to 1,000 Hz
1: 100 to 10,000 Hz
2: 1,000 to 20,000 Hz

S1+1 Initial Pulse Frequency

The value stored in the data register designated by operand S1+1 specifies the frequency of the initial pulse output in percent of the maximum of the frequency range selected by S1+0. When S1+0 is set to 0 (10 to 1,000 Hz) or 1 (100 to 10,000 Hz), valid values for operand S1+1 are 1 through 100, thus the initial pulse frequency can be 10 to 1,000 Hz or 100 to 10,000 Hz, respectively. When S1+0 is set to 2 (1,000 to 20,000 Hz), valid values for operand S1+1 are 1 through 20 and the S1+1 value multiplied by 5 determines the initial pulse frequency, thus the initial pulse frequency can be 1,000 to 20,000 Hz.

Initial Operation Mode	Initial Pulse Frequency (Hz)
0 or 1	Maximum frequency (Hz) selected by S1+0 × S1+1 value (%)
2	Maximum frequency (Hz) selected by S1+0 × S1+1 value (×5%)

S1+2 Creep Operation Mode

The value stored in the data register designated by operand S1+2 determines the frequency range of the low-frequency creep pulse output.

0: 10 to 1,000 Hz
1: 100 to 10,000 Hz
2: 1,000 to 20,000 Hz

S1+3 Creep Pulse Frequency

The value stored in the data register designated by operand S1+3 specifies the frequency of the creep pulse output in percent of the maximum of the frequency range selected by S1+2. When S1+2 is set to 0 (10 to 1,000 Hz) or 1 (100 to 10,000 Hz), valid values for operand S1+3 are 1 through 100, thus the creep pulse frequency can be 10 to 1,000 Hz or 100 to 10,000 Hz, respectively. When S1+2 is set to 2 (1,000 to 20,000 Hz), valid values for operand S1+3 are 1 through 20 and the S1+3 value multiplied by 5 determines the creep pulse frequency, thus the creep pulse frequency can be 1,000 to 20,000 Hz.

Creep Operation Mode	Creep Pulse Frequency (Hz)
0 or 1	Maximum frequency (Hz) selected by S1+2 × S1+3 value (%)
2	Maximum frequency (Hz) selected by S1+2 × S1+3 value (×5%)

S1+4 Error Status

When the start input for the ZRN1 or ZRN2 instruction is turned on, operand values are checked. When any error is found in the operand values, the data register designated by operand S1+4 stores an error code.

Error Code	Description
0	Normal
1	Operation mode designation error (S1+0 or S1+2 stores other than 0 through 2)
2	Output pulse frequency designation error (S1+1 or S1+3 stores other than 1 through 100)

Source Operand S2 (Deceleration Input)

When the deceleration input turns on while the ZRN instruction is generating output pulses of the initial pulse frequency, the pulse frequency is changed to the creep pulse frequency. When the deceleration input turns off, the ZRN instruction stops generating output pulses.

When using the ZRN1 and ZRN2 instructions, designate different input or internal relay numbers as deceleration inputs for the ZRN1 and ZRN2 instructions. If the same deceleration input is used and both the ZRN1 and ZRN2 instructions are executed at the same time, the pulse outputs may not turn off when the deceleration input turns on.

The deceleration input is available in two types depending on the designated operand number.

Operand	Function	Description
S2	High-speed deceleration input	I2, I3, I4, I5
	Normal deceleration input	I0, I1, I6 through I307, M0 through M1277

High-speed Deceleration Input (I2, I3, I4, I5)

The high-speed deceleration input uses interrupt processing to read the deceleration input signal immediately without regard to the scan time.

When I2 through I5 are used as a deceleration input for the ZRN instruction, designate these input numbers as normal inputs in the Function Area Settings. If I2 through I5 used as a deceleration input are designated as an interrupt input, catch input, or high-speed counter input in the Function Area Settings, the inputs work as a deceleration input for the ZRN instruction; the designation in the Function Area Settings will have no effect.

When using a high-speed deceleration input, make sure that the input contact does not bounce. If the input signal contains chatter, the pulse output will be stopped immediately.

Normal Deceleration Input (I0, I1, I6 through I307, M0 through M1277)

The normal deceleration input reads the deceleration input signal when the input data is updated at the END processing, so the timing of accepting the deceleration input depends on the scan time.

Destination Operand D1 (Status Relay)

Two internal relays starting with the operand designated by D1 indicate the status of the ZRN instruction. These operands are for read only.

Operand	Function	Description	R/W
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	R
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	R

D1+0 Pulse Output ON

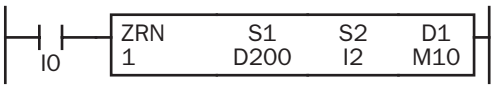
The internal relay designated by operand D1+0 remains on while the ZRN instruction generates output pulses. When the start input or deceleration input for the ZRN instruction is turned off to stop generating output pulses, the internal relay designated by operand D1+0 turns off.

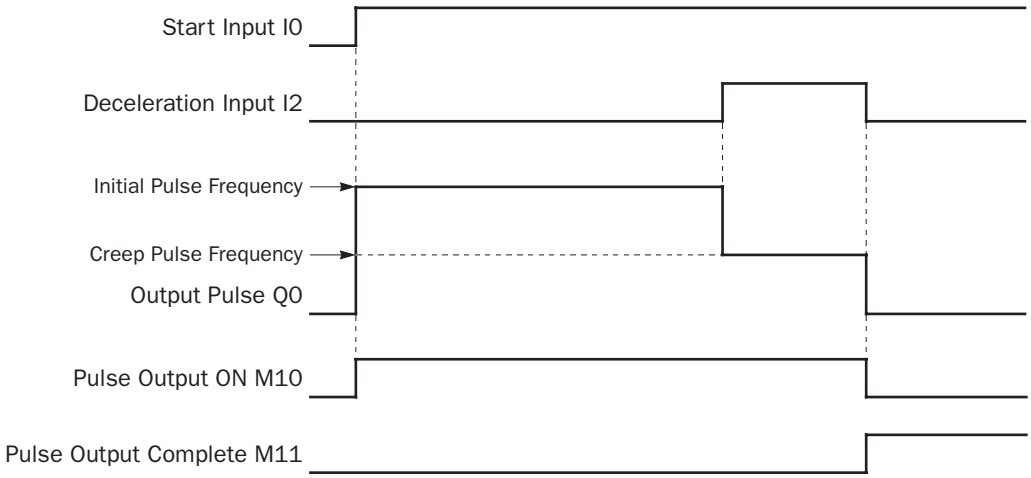
D1+1 Pulse Output Complete

The internal relay designated by operand D1+1 turns on when the deceleration input for the ZRN instruction is turned off to stop generating output pulses. When the start input for the ZRN instruction is turned on, the internal relay designated by operand D1+1 turns off.

Timing Chart for Zero-return Operation

This program demonstrates a timing chart of the ZRN1 instruction when input I2 is used for a high-speed deceleration input.





- When input I0 is turned on, ZRN1 starts to generate output pulses of the initial pulse frequency designated by the value stored in data register D201. While the output pulses are sent out from output Q0, internal relay M10 remains on.
- When deceleration input I2 is turned on, the output pulse frequency immediately reduces to the creep pulse frequency designated by the value stored in data register D203.
- When deceleration input I2 is turned off, ZRN1 stops generating output pulses immediately. Then internal relay M10 turns off, and internal relay M11 turns on.
- If parameter values in D200 through D203 are changed while generating output pulses, the change takes effect when start input I0 is turned on for the next cycle.
- If start input I0 is turned off while generating output pulses of either initial or creep pulse frequency, ZRN1 stops generating output pulses, then internal relay M10 turns off and internal relay M11 turns on. When input I0 is turned on again, ZRN1 restarts to generate output pulses for another cycle, starting at the initial pulse frequency.
- If deceleration input I2 is already on when start input I0 turns on, ZRN1 starts to generate pulse outputs of the creep pulse frequency.

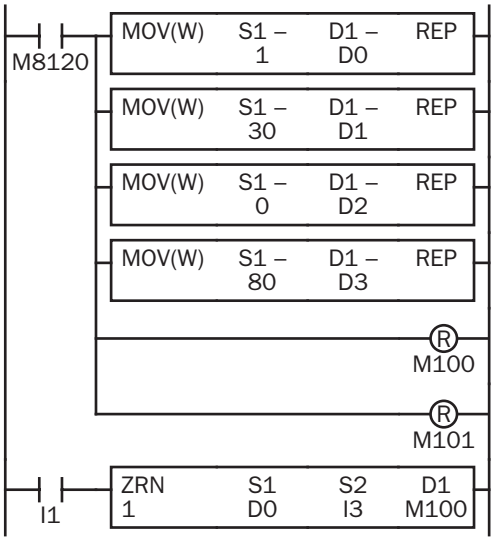
Sample Program: ZRN1

This program demonstrates a user program of the ZRN1 instruction used for zero-return operation to generate output pulses of 3 kHz initial pulse frequency from output Q0 while input I1 is on. When deceleration input I3 is turned on, the output pulse frequency reduces to the creep pulse frequency of 800 Hz. When deceleration input I3 is turned off, ZRN1 stops generating output pulses.

Initial pulse frequency: 3,000 Hz
Creep pulse frequency: 800 Hz
Deceleration input: I3 (high-speed deceleration input)

Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+0	Initial operation mode	Frequency range 100 to 10,000 Hz	D0 (1)
S1+1	Initial pulse frequency	10,000 Hz × 30% = 3,000 Hz	D1 (30)
S1+2	Creep operation mode	Frequency range 10 to 1,000 Hz	D2 (0)
S1+3	Creep pulse frequency	1,000 Hz × 80% = 800 Hz	D3 (80)
S1+4	Error status		D4
S2	Deceleration input	High-speed deceleration input	I3
D1+0	Pulse output ON	0: Pulse output OFF 1: Pulse output ON	M100
D1+1	Pulse output complete	0: Pulse output not complete 1: Pulse output complete	M101



M8120 is the initialize pulse special internal relay.

When the CPU starts, four MOV(W) instructions store parameters to data registers D0 through D3.

D0 (initial operation mode): 1 (100 to 10,000 Hz)

D1 (initial pulse frequency): 30 (10,000 Hz × 30% = 3,000 Hz)

D2 (creep operation mode): 0 (10 to 1,000 Hz)

D3 (creep pulse frequency): 80 (1,000 Hz × 80% = 800 Hz)

Pulse output ON flag M100 is turned off.

Pulse output complete flag M101 is turned off.

When start input I1 is turned on, ZRN1 starts to generate output pulses from output Q0.

21: PID INSTRUCTION

Introduction

The PID instruction implements a PID (proportional, integral, and derivative) algorithm with built-in auto tuning to determine PID parameters, such as proportional gain, integral time, derivative time, and control action automatically. The PID instruction is primarily designed for use with an analog I/O module to read analog input data, and turns on and off a designated output to perform PID control in applications such as temperature control described in the application example on page 21-14. In addition, when the output manipulated variable is converted, the PID instruction can also generate an analog output using an analog I/O module.

⚠ Warning

- Special technical knowledge about the PID control is required to use the PID function of the MicroSmart. Use of the PID function without understanding the PID control may cause the MicroSmart to perform unexpected operation, resulting in disorder of the control system, damage, or accidents.
- When using the PID instruction for feedback control, emergency stop and interlocking circuits must be configured outside the MicroSmart. If such a circuit is configured inside the MicroSmart, failure of inputting the process variable may cause equipment damage or accidents.

Upgrade Information

Upgraded CPU modules of all-in-one 24-I/O types and slim types have an additional option for the integral start coefficient (S1+10) to execute an integral action within the proportional band. Applicable CPU modules and system program versions are shown in the table below. For the procedure to confirm the system program version of the CPU module, see page 29-1.

CPU Module	All-in-One Type			Slim Type	
	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
System Program Version	—	—	202 or higher	202 or higher	201 or higher

PID (PID Control)

When input is on, auto tuning and/or PID action is executed according to the value (0 through 2) stored in a data register operand assigned for operation mode.

Applicable CPU Modules and Quantity of PID Instructions

A maximum of 8 or 14 PID instructions can be used in a user program, depending on the CPU module type.

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X (8)	X (8)	X (14)

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant
S1 (Source 1)	Control register	—	—	—	—	—	—	D0-D7973	—
S2 (Source 2)	Control relay	—	Q0-Q300	M0-M1270	—	—	—	—	—
S3 (Source 3)	Set point	—	—	—	—	—	—	D0-D7999	0-4095
S4 (Source 4)	Process variable (before conversion)	—	—	—	—	—	—	D0-D7999	—
D1 (Destination 1)	Manipulated variable	—	—	—	—	—	—	D0-D7999	—

For the valid operand number range, see pages 6-1 and 6-2.

Source operand S1 (control register) uses 27 data registers starting with the operand designated by S1. Data registers D0 through D1273 and D2000 through D7973 can be designated by S1. For details, see the following pages.

Source operand S2 (control relay) uses 8 points of outputs or internal relays starting with the operand designated by S2. Outputs Q0 through Q300 and internal relays M0 through M1270 can be designated by S2. For details, see page 21-11.

Source operand S3 (set point): When the linear conversion is disabled (S1+4 set to 0), the valid range of the set point (S3) is 0 through 4095 which can be designated using a data register or constant. When the linear conversion is enabled (S1+4 set to 1), the valid range is –32768 to 32767 that is a value after linear conversion. Use a data register to designate a negative value for a set point when the linear conversion is used. For details, see page 21-12.

Source operand S4 (process variable) is designated using a data register allocated as analog input data of the connected analog I/O module, such as D760 or D766. See page 24-8. To read input data from an analog I/O module, designate a proper data register number depending on the slot position of the analog I/O module and the analog input channel number connected to the analog input source. For details, see page 21-13.

Destination operand D1 (manipulated variable) stores –32768 through 32767 that is a calculation result of the PID action. For details, see page 21-13.

Source Operand S1 (Control Register)

Store appropriate values to data registers starting with the operand designated by S1 before executing the PID instruction as required, and make sure that the values are within the valid range. Operands S1+0 through S1+2 are for read only, and operands S1+23 through S1+26 are reserved for the system program.

Operand	Function	Description	R/W
S1+0	Process variable (after conversion)	When S1+4 (linear conversion) = 1 (enable linear conversion): Stores the process variable after conversion. When S1+4 (linear conversion) = 0 (disable linear conversion): Stores the process variable without conversion.	R
S1+1	Output manipulated variable	Stores the output manipulated variable (manual mode output variable and AT output manipulated variable) in percent. 0 to 100 (0% to 100%)	R
S1+2	Operating status	Stores the operating or error status of the PID instruction.	R
S1+3	Operation mode	0: PID action 1: AT (auto tuning) + PID action 2: AT (auto tuning)	R/W
S1+4	Linear conversion	0: Disable linear conversion 1: Enable linear conversion	R/W
S1+5	Linear conversion maximum value	–32768 to +32767	R/W
S1+6	Linear conversion minimum value	–32768 to +32767	R/W
S1+7	Proportional gain	1 to 10000 (0.01% to 100.00%) 0 designates 0.01%, ≥10001 designates 100.00%	R/W
S1+8	Integral time	1 to 65535 (0.1 sec to 6553.5 sec), 0 disables integral action	R/W
S1+9	Derivative time	1 to 65535 (0.1 sec to 6553.5 sec), 0 disables derivative action	R/W
S1+10	Integral start coefficient	1 to 100 (1% to 100%), 0 and ≥101 (except 200) designate 100% 200 executes integral action within the proportional band (upgraded CPU only)	R/W
S1+11	Input filter coefficient	0 to 99 (0% to 99%), ≥100 designates 99%	R/W
S1+12	Sampling period	1 to 10000 (0.01 sec to 100.00 sec) 0 designates 0.01 sec, ≥10001 designates 100.00 sec	R/W
S1+13	Control period	1 to 500 (0.1 sec to 50.0 sec) 0 designates 0.1 sec, ≥501 designates 50.0 sec	R/W
S1+14	High alarm value	When S1+4 (linear conversion) = 0: 0 to 4095 (≥4096 designates 4095) When S1+4 = 1: Linear conversion min. ≤ High alarm ≤ Linear conversion max. When S1+14 < S1+6 (linear conversion min.), S1+6 becomes high alarm. When S1+14 > S1+5 (linear conversion max.), S1+5 becomes high alarm.	R/W
S1+15	Low alarm value	When S1+4 (linear conversion) = 0: 0 to 4095 (≥4096 designates 4095) When S1+4 = 1: Linear conversion min. ≤ Low alarm ≤ Linear conversion max. When S1+15 < S1+6 (linear conversion min.), S1+6 becomes low alarm. When S1+15 > S1+5 (linear conversion max.), S1+5 becomes low alarm.	R/W
S1+16	Output manipulated variable upper limit	0 to 100, 10001 to 10099 (other values designate 100)	R/W
S1+17	Output manipulated variable lower limit	0 to 100 (≥101 designates 100)	R/W
S1+18	Manual mode output manipulated variable	0 to 100 (≥101 designates 100)	R/W
S1+19	AT sampling period	1 to 10000 (0.01 sec to 100.00 sec) 0 designates 0.01 sec, ≥10001 designates 100.00 sec	R/W
S1+20	AT control period	1 to 500 (0.1 sec to 50.0 sec) 0 designates 0.1 sec, ≥501 designates 50.0 sec	R/W
S1+21	AT set point	When S1+4 (linear conversion) = 0: 0 to 4095 (≥4096 designates 4095) When S1+4 = 1: Linear conversion min. ≤ AT set point ≤ Linear conversion max.	R/W
S1+22	AT output manipulated variable	0 to 100 (≥101 designates 100)	R/W
S1+23 S1+24 S1+25 S1+26	— Reserved for processing the PID instruction —		

Note: The value stored in the data register designated by S1+3 (operation mode) is checked only when the start input for the PID instruction is turned on. Values in all other control registers are refreshed in every scan.

S1+0 Process Variable (after conversion)

When the linear conversion is enabled (S1+4 set to 1), the data register designated by S1+0 stores the linear conversion result of the process variable (S4). The process variable (S1+0) takes a value between the linear conversion minimum value (S1+6) and the linear conversion maximum value (S1+5).

When the linear conversion is disabled (S1+4 is set to 0), the data register designated by S1+0 stores the same value as the process variable (S4).

S1+1 Output Manipulated Variable

While the PID action is in progress, the data register designated by S1+1 holds 0 through 100 read from the manipulated variable, –32768 through 32767, stored in the data register designated by D1, omitting values less than 0 and greater than 100. The percent value in S1+1 determines the ON duration of the control output (S2+6) in proportion to the control period (S1+13).

While manual mode is enabled with the auto/manual mode control relay (S2+1) set to on, S1+1 stores 0 through 100 read from the manual mode output manipulated variable (S1+18).

While auto tuning (AT) is in progress, S1+1 stores 0 through 100 read from the AT output manipulated variable (S1+22).

S1+2 Operating Status

The data register designated by S1+2 stores the operating or error status of the PID instruction.

Status codes 1X through 6X contain the time elapsed after starting auto tuning or PID action. X changes from 0 through 9 in 10-minute increments to represent 0 through 90 minutes. The time code remains 9 after 90 minutes has elapsed. When the operation mode (S1+3) is set to 1 (AT+PID), the time code is reset to 0 at the transition from AT to PID.

Status codes 100 and above indicate an error, stopping the auto tuning or PID action. When these errors occur, a user program execution error will result, turning on the ERR LED and special internal relay M8004 (user program execution error). To continue operation, enter correct parameters and turn on the start input for the PID instruction.

Status Code	Description	Operation
1X	AT in progress	AT is normal.
2X	AT completed	
5X	PID action in progress	PID action is normal.
6X	PID set point (S3) is reached. Status code changes from 5X to 6X once the PID set point is reached.	
100	The operation mode (S1+3) is set to a value over 2.	PID action or AT is stopped because of incorrect parameter settings.
101	The linear conversion (S1+4) is set to a value over 1.	
102	When the linear conversion is enabled (S1+4 to 1), the linear conversion maximum value (S1+5) and the linear conversion minimum value (S1+6) are set to the same value.	
103	The output manipulated variable upper limit (S1+16) is set to a value smaller than the output manipulated variable lower limit (S1+17).	
104	When the linear conversion is enabled (S1+4 set to 1), the AT set point (S1+21) is set to a value larger than the linear conversion maximum value (S1+5) or smaller than the linear conversion minimum value (S1+6).	
105	When the linear conversion is disabled (S1+4 set to 0), the AT set point (S1+21) is set to a value larger than 4095.	
106	When the linear conversion is enabled (S1+4 set to 1), the set point (S3) is set to a value larger than the linear conversion maximum value (S1+5) or smaller than the linear conversion minimum value (S1+6).	
107	When the linear conversion is disabled (S1+4 set to 0), the set point (S3) is set to a value larger than 4095.	
200	The current control action (S2+0) differs from that determined at the start of AT. To restart AT, set correct parameters referring to the probable causes listed below: • The manipulated variable (D1) or the control output (S2+6) is not outputted to the control target correctly. • The process variable is not stored to the operand designated by S4. • The AT output manipulated variable (S1+22) is not set to a large value so that the process variable (S4) can change sufficiently. • A large disturbance occurred.	AT is stopped because of AT execution error.
201	AT failed to complete normally because the process variable (S4) fluctuated excessively. To restart AT, set the AT sampling period (S1+19) or the input filter coefficient (S1+11) to a larger value.	

S1+3 Operation Mode

When the start input for the PID instruction is turned on, the CPU module checks the value stored in the data register designated by S1+3 and executes the selected operation. The selection cannot be changed while executing the PID instruction.

0: PID action
The PID action is executed according to the designated PID parameters such as proportional gain (S1+7), integral time (S1+8), derivative time (S1+9), and control action (S2+0).

1: AT (auto tuning) + PID action
Auto tuning is first executed according to the designated AT parameters such as AT sampling period (S1+19), AT control period (S1+20), AT set point (S1+21), and AT output manipulated variable (S1+22). As a result of auto tuning, PID parameters are determined such as proportional gain (S1+7), integral time (S1+8), derivative time (S1+9), and control direction (S2+0), then PID action is executed according to the derived PID parameters.

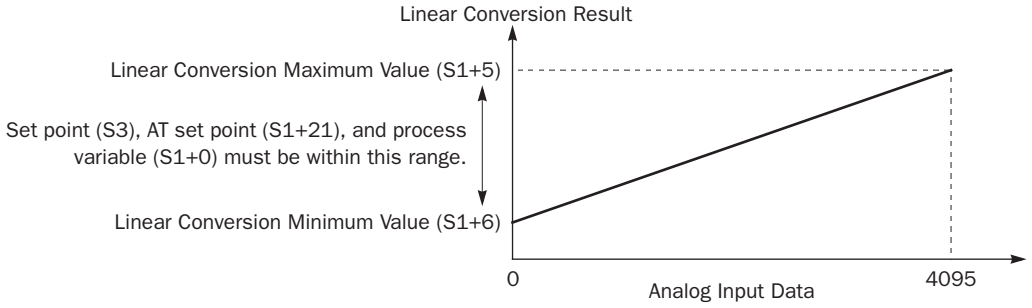
2: AT (auto tuning)
Auto tuning is executed according to designated AT parameters to determine PID parameters such as proportional gain (S1+7), integral time (S1+8), derivative time (S1+9), and control direction (S2+0); PID action is not executed.

S1+4 Linear Conversion

0: Disable linear conversion
Linear conversion is not executed. When the linear conversion is disabled (S1+4 set to 0), the analog input data (0 through 4095) from the analog I/O module is stored to the process variable (S4), and the same value is stored to the process variable (S1+0) without conversion.

1: Enable linear conversion
The linear conversion function is useful for scaling the process variable to the actual measured value in engineering units.

When the linear conversion is enabled (S1+4 set to 1), the analog input data (0 through 4095) from the analog I/O module is linear-converted, and the result is stored to the process variable (S1+0). When using the linear conversion, set proper values to the linear conversion maximum value (S1+5) and linear conversion minimum value (S1+6) to specify the linear conversion output range. When using the linear conversion function in a temperature control application, temperature values can be used to designate the set point (S3), high alarm value (S1+14), low alarm value (S1+15), and AT set point (S1+21), and also to read the process variable (S1+0).



S1+5 Linear Conversion Maximum Value

When the linear conversion is enabled (S1+4 set to 1), set the linear conversion maximum value to the data register designated by S1+5. Valid values are –32768 through 32767, and the linear conversion maximum value must be larger than the linear conversion minimum value (S1+6). Select an appropriate value for the linear conversion maximum value to represent the maximum value of the input signal to the analog I/O module.

When the linear conversion is disabled (S1+4 set to 0), you don't have to set the linear conversion maximum value (S1+5).

S1+6 Linear Conversion Minimum Value

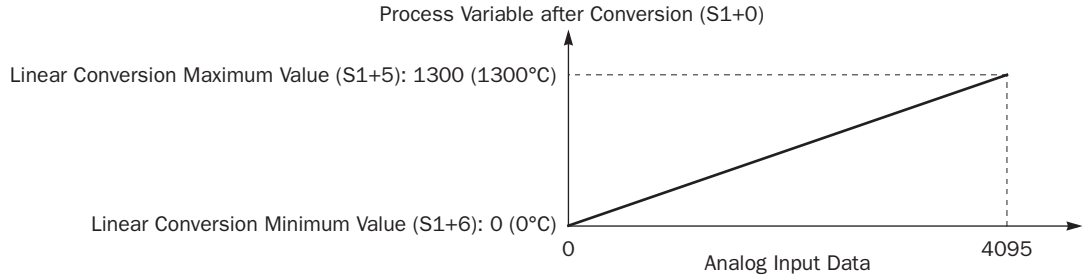
When the linear conversion is enabled (S1+4 set to 1), set the linear conversion minimum value to the data register designated by S1+6. Valid values are –32768 through 32767, and the linear conversion minimum value must be smaller than the linear conversion maximum value (S1+5). Select an appropriate value for the linear conversion minimum value to represent the minimum value of the input signal to the analog I/O module.

When the linear conversion is disabled (S1+4 set to 0), you don't have to set the linear conversion minimum value (S1+6).

Example:

When type K thermocouple is connected, the analog input data ranges from 0 through 4095. To convert the analog input data to actual measured temperature values, set the following parameters.

Linear conversion (S1+4):	1 (enable linear conversion)
Linear conversion maximum value (S1+5):	1300 (1300°C)
Linear conversion minimum value (S1+6):	0 (0°C)



S1+7 Proportional Gain

The proportional gain is a parameter to determine the amount of proportional action in the proportional band.

When auto tuning is used by setting the operation mode (S1+3) to 1 (AT+PID) or 2 (AT), a proportional gain is determined automatically and does not have to be specified by the user.

When auto tuning is not used by setting the operation mode (S1+3) to 0 (PID), set a required value of 1 through 10000 to specify a proportional gain of 0.01% through 100.00% to the data register designated by S1+7. When S1+7 stores 0, the proportional gain is set to 0.01%. When S1+7 stores a value larger than 10000, the proportional gain is set to 100.00%.

When the proportional gain is set to a large value, the proportional band becomes small and the response becomes fast, but overshoot and hunching will be caused. In contrast, when the proportional gain is set to a small value, overshoot and hunching are suppressed, but response to disturbance will become slow.

While the PID action is in progress, the proportional gain value can be changed by the user.

S1+8 Integral Time

When only the proportional action is used, a certain amount of difference (offset) between the set point (S3) and the process variable (S1+0) remains after the control target has reached a stable state. An integral action is needed to reduce the offset to zero. The integral time is a parameter to determine the amount of integral action.

When auto tuning is used by setting the operation mode (S1+3) to 1 (AT+PID) or 2 (AT), an integral time is determined automatically and does not have to be specified by the user.

When auto tuning is not used by setting the operation mode (S1+3) to 0 (PID), set a required value of 1 through 65535 to specify an integral time of 0.1 sec through 6553.5 sec to the data register designated by S1+8. When S1+8 is set to 0, the integral action is disabled.

When the integral time is too short, the integral action becomes too large, resulting in hunching of a long period. In contrast, when the integral time is too long, it takes a long time before the process variable (S1+0) reaches the set point (S3).

While the PID action is in progress, the integral time value can be changed by the user.

S1+9 Derivative Time

The derivative action is a function to adjust the process variable (S1+0) to the set point (S3) by increasing the manipulated variable (D1) when the set point (S3) is changed or when the difference between the process variable (S1+0) and the set point (S3) is increased due to disturbance. The derivative time is a parameter to determine the amount of derivative action.

When auto tuning is used by setting the operation mode (S1+3) to 1 (AT+PID) or 2 (AT), a derivative time is determined automatically and does not have to be specified by the user.

When auto tuning is not used by setting the operation mode (S1+3) to 0 (PID), set a required value of 1 through 65535 to specify a derivative time of 0.1 sec through 6553.5 sec to the data register designated by S1+9. When S1+9 is set to 0, the derivative action is disabled.

When the derivative time is set to a large value, the derivative action becomes large. When the derivative action is too large, hunching of a short period is caused.

While the PID action is in progress, the derivative time value can be changed by the user.

S1+10 Integral Start Coefficient

The integral start coefficient is a parameter to determine the point, in percent of the proportional term, where to start the integral action. Normally, the data register designated by S1+10 (integral start coefficient) stores 0 to select an integral start coefficient of 100% and the integral start coefficient disable control relay (S2+3) is turned off to enable integral start coefficient. When the PID action is executed according to the PID parameters determined by auto tuning, proper control is ensured with a moderate overshoot and no offset.

It is also possible to set a required value of 1 through 100 to start the integral action at 1% through 100% to the data register designated by S1+10. When S1+10 stores 0 or a value larger than 100 (except for 200), the integral start coefficient is set to 100%.

Another optional value of 200 is available on upgraded CPU modules with system program ver. 202 (FC4A-C24R2, FC4A-C24R2C, FC4A-D20K3, and FC4A-D20S3) and system program ver. 201 (FC4A-D20RK1, FC4A-D20RS1, FC4A-D40K3, and FC4A-D40S3) or higher.

When 200 is set to S1+10 in these upgraded CPU modules, the integral action is enabled only while the process variable (S4) is within the proportional band. When the process variable runs off the proportional band due to disturbance or changing of the set point, the integral action is disabled, so that adjustment of the output manipulated variable (S1+1) is improved with little overshoot and undershoot.

To enable the integral start coefficient, turn off the integral start coefficient disable control relay (S2+3). When S2+3 is turned on, the integral start coefficient is disabled and the integral term takes effect at the start of the PID action.

When the integral term is enabled at the start of the PID action, a large overshoot is caused. The overshoot can be suppressed by delaying the execution of the integral action in coordination with the proportional term. The PID instruction is designed to achieve proper control with a small or moderate overshoot when the integral start coefficient is set to 100%. Overshoot is most suppressed when the integral start coefficient is set to 1% and is least suppressed when the integral start coefficient is set to 100%. When the integral start coefficient is too small, overshoot is eliminated but offset is caused.

S1+11 Input Filter Coefficient

The input filter has an effect to smooth out fluctuations of the process variable (S4). Set a required value of 0 through 99 to specify an input filter coefficient of 0% through 99% to the data register designated by S1+11. When S1+11 stores a value larger than 99, the input filter coefficient is set to 99%. The larger the coefficient, the larger the input filter effect.

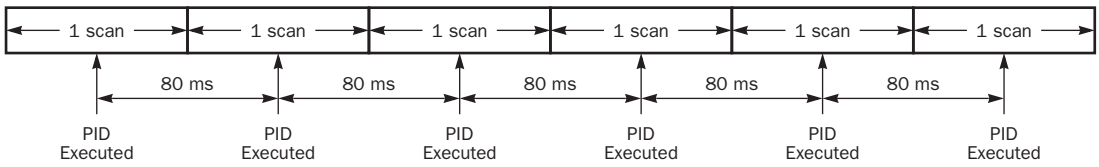
The input filter is effective for reading a process variable (S4) such as temperature data when the value changes at each sampling time. The input filter coefficient is in effect during auto tuning and PID action.

S1+12 Sampling Period

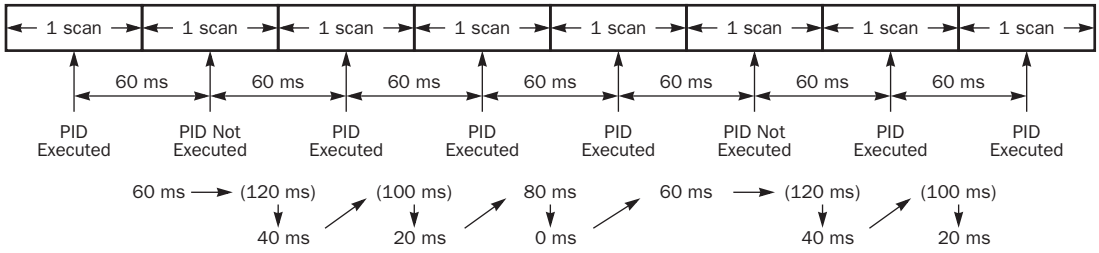
The sampling period determines the interval to execute the PID instruction. Set a required value of 1 through 10000 to specify a sampling period of 0.01 sec through 100.00 sec to the data register designated by S1+12. When S1+12 stores 0, the sampling period is set to 0.01 sec. When S1+12 stores a value larger than 10000, the sampling period is set to 100.00 sec.

When a sampling period is set to a value smaller than the scan time, the PID instruction is executed every scan.

Example – Sampling period: 40 ms, Scan time: 80 ms (Sampling period ≤ Scan time)



Example – Sampling period: 80 ms, Scan time: 60 ms (Sampling period > Scan time)

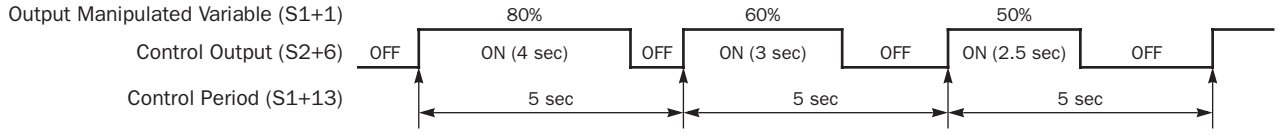


S1+13 Control Period

The control period determines the duration of the ON/OFF cycle of the control output (S2+6) that is turned on and off according to the output manipulated variable (S1+1) calculated by the PID action or derived from the manual mode output manipulated variable (S1+18). Set a required value of 1 through 500 to specify a control period of 0.1 sec through 50.0 sec to the data register designated by S1+13. When S1+13 stores 0, the control period is set to 0.1 sec. When S1+13 is set to a value larger than 500, the control period is set to 50.0 sec.

The ON pulse duration of the control output (S2+6) is determined by the product of the control period (S1+13) and the output manipulated variable (S1+1).

Example – Control period: 5 sec (S1+13 is set to 50)



S1+14 High Alarm Value

The high alarm value is the upper limit of the process variable (S1+0) to generate an alarm. When the process variable is higher than or equal to the high alarm value, the high alarm output control relay (S2+4) is turned on. When the process variable is lower than the high alarm value, the high alarm output control relay (S2+4) is turned off.

When the linear conversion is disabled (S1+4 set to 0), set a required high alarm value of 0 through 4095 to the data register designated by S1+14. When S1+14 stores a value larger than 4095, the high alarm value is set to 4095.

When the linear conversion is enabled (S1+4 set to 1), set a required high alarm value of –32768 through 32767 to the data register designated by S1+14. The high alarm value must be larger than or equal to the linear conversion minimum value (S1+6) and must be smaller than or equal to the linear conversion maximum value (S1+5). If the high alarm value is set to a value smaller than the linear conversion minimum value (S1+6), the linear conversion minimum value will become the high alarm value. If the high alarm value is set to a value larger than the linear conversion maximum value (S1+5), the linear conversion maximum value will become the high alarm value.

S1+15 Low Alarm Value

The low alarm value is the lower limit of the process variable (S1+0) to generate an alarm. When the process variable is lower than or equal to the low alarm value, the low alarm output control relay (S2+5) is turned on. When the process variable is higher than the low alarm value, the low alarm output control relay (S2+5) is turned off.

When the linear conversion is disabled (S1+4 set to 0), set a required low alarm value of 0 through 4095 to the data register designated by S1+15. When S1+15 stores a value larger than 4095, the low alarm value is set to 4095.

When the linear conversion is enabled (S1+4 set to 1), set a required low alarm value of –32768 through 32767 to the data register designated by S1+15. The low alarm value must be larger than or equal to the linear conversion minimum value (S1+6) and must be smaller than or equal to the linear conversion maximum value (S1+5). If the low alarm value is set to a value smaller than the linear conversion minimum value (S1+6), the linear conversion minimum value will become the low alarm value. If the low alarm value is set to a value larger than the linear conversion maximum value (S1+5), the linear conversion maximum value will become the low alarm value.

S1+16 Output Manipulated Variable Upper Limit

The value contained in the data register designated by S1+16 specifies the upper limit of the output manipulated variable (S1+1) in two ways: direct and proportional.

S1+16 Value 0 through 100

When S1+16 contains a value 0 through 100, the value directly determines the upper limit of the output manipulated variable (S1+1). If the manipulated variable (D1) is greater than or equal to the upper limit value (S1+16), the upper limit value is outputted to the output manipulated variable (S1+1). Set a required value of 0 through 100 for the output manipulated variable upper limit to the data register designated by S1+16. When S1+16 stores a value larger than 100 (except 10001 through 10099), the output manipulated variable upper limit (S1+16) is set to 100. The output manipulated variable upper limit (S1+16) must be larger than the output manipulated variable lower limit (S1+17).

To enable the manipulated variable upper limit, turn on the output manipulated variable limit enable control relay (S2+2). When S2+2 is turned off, the output manipulated variable upper limit (S1+16) has no effect.

S1+16 Value 10001 through 10099 (disables Output Manipulated Variable Lower Limit S1+17)

When S1+16 contains a value 10001 through 10099, the value minus 10000 determines the ratio of the output manipulated variable (S1+1) in proportion to the manipulated variable (D1) of 0 through 100. The output manipulated variable (S1+1) can be calculated by the following equation:

$$\text{Output manipulated variable (S1+1)} = \text{Manipulated variable (D1)} \times (N - 10000)$$

where N is the value stored in the output manipulated variable upper limit (S1+16), 10001 through 10099.

If the manipulated variable (D1) is greater than or equal to 100, 100 multiplied by (N – 10000) is outputted to the output manipulated variable (S1+1). If D1 is less than or equal to 0, 0 is outputted to S1+1.

To enable the manipulated variable upper limit, turn on the output manipulated variable limit enable control relay (S2+2). When S2+2 is turned off, the output manipulated variable upper limit (S1+16) has no effect.

When S1+16 is set to a value 10001 through 10099, the output manipulated variable lower limit (S1+17) is disabled.

S1+17 Output Manipulated Variable Lower Limit

The value contained in the data register designated by S1+17 specifies the lower limit of the output manipulated variable (S1+1). Set a required value of 0 through 100 for the output manipulated variable lower limit to the data register designated by S1+17. When S1+17 stores a value larger than 100, the output manipulated variable lower limit is set to 100. The output manipulated variable lower limit (S1+17) must be smaller than the output manipulated variable upper limit (S1+16).

To enable the output manipulated variable lower limit, turn on the output manipulated variable limit enable control relay (S2+2), and set the output manipulated variable upper limit (S1+16) to a value other than 10001 through 10099. When the manipulated variable (D1) is smaller than or equal to the specified lower limit, the lower limit value is outputted to the output manipulated variable (S1+1).

When the output manipulated variable limit enable control relay (S2+2) is turned off, the output manipulated variable lower limit (S1+17) has no effect.

S1+18 Manual Mode Output Manipulated Variable

The manual mode output manipulated variable specifies the output manipulated variable (0 through 100) for manual mode. Set a required value of 0 through 100 for the manual mode output manipulated variable to the data register designated by S1+18. When S1+18 stores a value larger than 100, the manual mode output manipulated variable is set to 100.

To enable the manual mode, turn on the auto/manual mode control relay (S2+1). While in manual mode, the PID action is disabled. The specified value of the manual mode output manipulated variable (S1+18) is outputted to the output manipulated variable (S1+1), and the control output (S2+6) is turned on and off according to the control period (S1+13) and the manual mode output manipulated variable (S1+18).

S1+19 AT Sampling Period

The AT sampling period determines the interval of sampling during auto tuning. When using auto tuning, set a required value of 1 through 10000 to specify an AT sampling period of 0.01 sec through 100.00 sec to the data register designated by S1+19. When S1+19 stores 0, the AT sampling period is set to 0.01 sec. When S1+19 stores a value larger than 10000, the AT sampling period is set to 100.00 sec.

Set the AT sampling period to a long value to make sure that the current process variable is smaller than or equal to the previous process variable during direct control action (S2+0 is on) or that the current process variable is larger than or equal to the previous process variable during reverse control action (S2+0 is off).

S1+20 AT Control Period

The AT control period determines the duration of the ON/OFF cycle of the control output (S2+6) during auto tuning. For operation of the control output, see Control Period on page 21-8.

When using auto tuning, set a required value of 1 through 500 to specify an AT control period of 0.1 sec through 50.0 sec to the data register designated by S1+20. When S1+20 stores 0, the AT control period is set to 0.1 sec. When S1+20 stores a value larger than 500, the AT control period is set to 50.0 sec.

S1+21 AT Set Point

While auto tuning is executed, the AT output manipulated variable (S1+22) is outputted to the output manipulated variable (S1+1) until the process variable (S1+0) reaches the AT set point (S1+21). When the process variable (S1+0) reaches the AT set point (S1+21), auto tuning is complete and the output manipulated variable (S1+1) is reduced to zero. When PID action is selected with operation mode (S1+3) set to 1 (AT+PID), the PID action follows immediately.

When the linear conversion is disabled (S1+4 set to 0), set a required AT set point of 0 through 4095 to the data register designated by S1+21. When S1+21 stores a value larger than 4095, the AT set point is set to 4095.

When the linear conversion is enabled (S1+4 set to 1), set a required AT set point of –32768 through 32767 to the data register designated by S1+21. The AT set point must be larger than or equal to the linear conversion minimum value (S1+6) and must be smaller than or equal to the linear conversion maximum value (S1+5).

In the direct control action (see page 21-11), set the AT set point (S1+21) to a value sufficiently smaller than the process variable (S4) at the start of the auto tuning. In the reverse control action, set the AT set point (S1+21) to a value sufficiently larger than the process variable (S4) at the start of the auto tuning.

S1+22 AT Output Manipulated Variable

The AT output manipulated variable specifies the amount of the output manipulated variable (0 through 100) during auto tuning. When using auto tuning, set a required AT output manipulated variable of 0 through 100 to the data register designated by S1+22. When S1+22 stores a value larger than 100, the AT output manipulated variable is set to 100.

While auto tuning is executed, the specified value of the AT output manipulated variable (S1+22) is outputted to the output manipulated variable (S1+1), and the control output (S2+6) is turned on and off according to the AT control period (S1+20) and the AT output manipulated variable (S1+22). To keep the control output (S2+6) on during auto tuning, set 100 to S1+22.

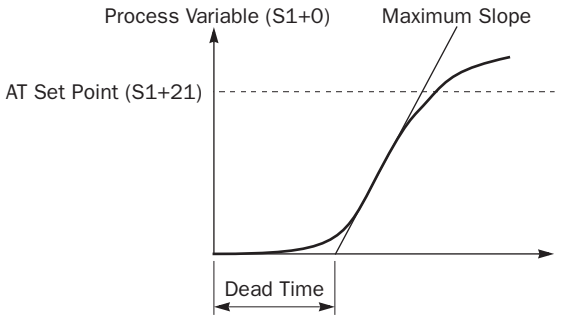
Auto Tuning (AT)

When auto tuning is selected with the operation mode (S1+3) set to 1 (AT+PID) or 2 (AT), the auto tuning is executed before starting PID control to determine PID parameters, such as proportional gain (S1+7), integral time (S1+8), derivative time (S1+9), and control action (S2+0) automatically. The MicroSmart uses the step response method to execute auto tuning. To enable auto tuning, set four parameters for auto tuning before executing the PID instruction, such as AT sampling period (S1+19), AT control period (S1+20), AT set point (S1+21), and AT output manipulated variable (S1+22).

Step Response Method

The MicroSmart uses the step response method to execute auto tuning and determine PID parameters such as proportional gain (S1+7), integral time (S1+8), derivative time (S1+9), and control action (S2+0) automatically. The auto tuning is executed in the following steps:

1. Calculate the maximum slope of the process variable (S1+0) before the process variable reaches the AT set point (S1+21).
2. Calculate the dead time based on the derived maximum slope.
3. Based on the maximum slope and dead time, calculate the four PID parameters.



Source Operand S2 (Control Relay)

Turn on or off appropriate outputs or internal relays starting with the operand designated by S2 before executing the PID instruction as required. Operands S2+4 through S2+7 are for read only to reflect the PID and auto tuning statuses.

Operand	Function	Description	R/W
S2+0	Control action	ON: Direct control action OFF: Reverse control action	R/W
S2+1	Auto/manual mode	ON: Manual mode OFF: Auto mode	R/W
S2+2	Output manipulated variable limit enable	ON: Enable output manipulated variable upper and lower limits (S1+16 and S1+17) OFF: Disable output manipulated variable upper and lower limits (S1+16 and S1+17)	R/W
S2+3	Integral start coefficient disable	ON: Disable integral start coefficient (S1+10) OFF: Enable integral start coefficient (S1+10)	R/W
S2+4	High alarm output	ON: When process variable (S1+0) ≥ high alarm value (S1+14) OFF: When process variable (S1+0) < high alarm value (S1+14)	R
S2+5	Low alarm output	ON: When process variable (S1+0) ≤ low alarm value (S1+15) OFF: When process variable (S1+0) > low alarm value (S1+15)	R
S2+6	Control output	Goes on and off according to the AT parameters or PID calculation results	R
S2+7	AT complete output	Goes on when AT is complete or failed, and remains on until reset	R

S2+0 Control Action

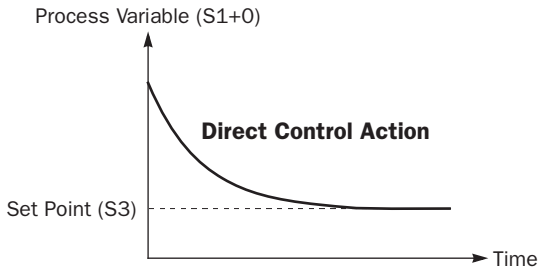
When auto tuning is executed with the operation mode (S1+3) set to 1 (AT+PID) or 2 (AT), the control action is determined automatically. When auto tuning results in a direct control action, the control action control relay designated by S2+0 is turned on. When auto tuning results in a reverse control action, the control action control relay designated by S2+0 is turned off. The PID action is executed according to the derived control action, which remains in effect during the PID action.

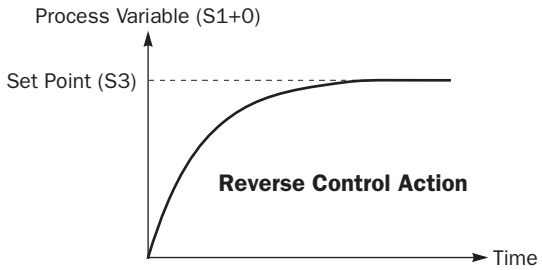
When auto tuning is not executed with the operation mode (S1+3) set to 0 (PID), turn on or off the control action control relay (S2+0) to select a direct or reverse control action, respectively, before executing the PID instruction.

In the direct control action, the manipulated variable (D1) is increased while the process variable (S1+0) is larger than the set point (S3). Temperature control for cooling is executed in the direct control action.

In the reverse control action, the manipulated variable (D1) is increased while the process variable (S1+0) is smaller than the set point (S3). Temperature control for heating is executed in the reverse control action.

In either the direct or reverse control action, the manipulated variable (D1) is increased while the difference between the process variable (S1+0) and the set point (S3) increases.





S2+1 Auto/Manual Mode

To select auto mode, turn off the auto/manual mode control relay designated by S2+1 before or after starting the PID instruction. In auto mode, the PID action is executed and the manipulated variable (D1) stores the PID calculation result. The control output (S2+6) is turned on and off according to the control period (S1+13) and the output manipulated variable (S1+1).

To select manual mode, turn on the auto/manual mode control relay (S2+1). When using manual mode, set a required value to the manual mode output manipulated variable (S1+18) before enabling manual mode. In manual mode, the output manipulated variable (S1+1) stores the manual mode output manipulated variable (S1+18), and the control output (S2+6) is turned on and off according to the control period (S1+13) and the manual mode output manipulated variable (S1+18).

While auto tuning is in progress, manual mode cannot be enabled. Only after auto tuning is complete, auto or manual mode can be enabled. Auto/manual mode can also be switched while executing the PID instruction.

S2+2 Output Manipulated Variable Limit Enable

The output manipulated variable upper limit (S1+16) and the output manipulated variable lower limit (S1+17) are enabled or disabled using the output manipulated variable limit enable control relay (S2+2).

To enable the output manipulated variable upper/lower limits, turn on S2+2.

To disable the output manipulated variable upper/lower limits, turn off S2+2.

S2+3 Integral Start Coefficient Disable

The integral start coefficient (S1+10) is enabled or disabled using the integral start coefficient disable control relay (S2+3).

To enable the integral start coefficient (S1+10), turn off S2+3; the integral term is enabled as specified by the integral start coefficient (S1+10).

To disable the integral start coefficient (S1+10), turn on S2+3; the integral term is enabled at the start of the PID action.

S2+4 High Alarm Output

When the process variable (S1+0) is higher than or equal to the high alarm value (S1+14), the high alarm output control relay (S2+4) goes on. When S1+0 is lower than S1+14, S2+4 is off.

S2+5 Low Alarm Output

When the process variable (S1+0) is lower than or equal to the low alarm value (S1+15), the low alarm output control relay (S2+5) goes on. When S1+0 is higher than S1+15, S2+5 is off.

S2+6 Control Output

During auto tuning in auto mode with the auto/manual mode control relay (S2+1) set to off, the control output (S2+6) is turned on and off according to the AT control period (S1+20) and AT output manipulated variable (S1+22).

During PID action in auto mode with the auto/manual mode control relay (S2+1) set to off, the control output (S2+6) is turned on and off according to the control period (S1+13) and the output manipulated variable (S1+1) calculated by the PID action.

In manual mode with the auto/manual mode control relay (S2+1) set to on, the control output (S2+6) is turned on and off according to the control period (S1+13) and the manual mode output manipulated variable (S1+18).

S2+7 AT Complete Output

The AT complete output control relay (S2+7) goes on when auto tuning is complete or failed, and remains on until reset. Operating status codes are stored to the operating status control register (S1+2). See page 21-4.

Source Operand S3 (Set Point)

The PID action is executed to adjust the process variable (S1+0) to the set point (S3).

When the linear conversion is disabled (S1+4 set to 0), set a required set point value of 0 through 4095 to the operand designated by S3. Valid operands are data register and constant.

When the linear conversion is enabled (S1+4 set to 1), designate a data register as operand S3 and set a required set point value of –32768 through 32767 to the data register designated by S3. Since the PID instruction uses the word data type, negative constants cannot be entered directly to operand S3. Use the MOV instruction with the integer (I) data type to store a negative value to a data register. The set point value (S3) must be larger than or equal to the linear conversion minimum value (S1+6) and smaller than or equal to the linear conversion maximum value (S1+5).

When an invalid value is designated as a set point, the PID action is stopped and an error code is stored to the data register designated by S1+2. See Operating Status on page 21-4.

Source Operand S4 (Process Variable before Conversion)

The PID instruction is designed to use analog input data from an analog I/O module as process variable. The analog I/O module converts the input signal to a digital value of 0 through 4095, and stores the digital value to a data register depending on the mounting position of the analog I/O module and the analog input channel connected to the analog input source. Designate a data register as source operand S4 to store the process variable.

For example, when the analog I/O module is mounted in the first slot from the CPU module among all analog I/O modules (not including digital I/O modules) and when the analog input signal is connected to analog input channel 0 of the analog I/O module, designate D760 as source operand S4. When the analog input is connected to analog input channel 1 of analog I/O module No. 3, designate D806 as source operand S4. For details about data register allocation numbers for analog I/O modules, see page 24-8.

Allocation Numbers for Source Operand S4

Channel	Analog I/O Module No.						
	1	2	3	4	5	6	7
Analog Input Ch 0	D760	D780	D800	D820	D840	D860	D880
Analog Input Ch 1	D766	D786	D806	D826	D846	D866	D886

When an analog I/O module is not used for supplying data to source operand S4, make sure that the S4 data takes a value between 0 and 4095. When S4 stores a value larger than 4095, the process variable is set to 4095.

Destination Operand D1 (Manipulated Variable)

The data register designated by destination operand D1 stores the manipulated variable of –32768 through 32767 calculated by the PID action. When the calculation result is less than –32768, D1 stores –32768. When the calculation result is greater than 32767, D1 stores 32767. While the calculation result is less than –32768 or greater than 32767, the PID action still continues.

When the output manipulated variable limit is disabled (S2+2 set to off) while the PID action is in progress, the data register designated by S1+1 holds 0 through 100 of the manipulated variable (D1), omitting values less than 0 and greater than 100. The percent value in S1+1 determines the ON duration of the control output (S2+6) in proportion to the control period (S1+13).

When the output manipulated variable limit is enabled (S2+2 set to on), the manipulated variable (D1) is stored to the output manipulated variable (S1+1) according to the output manipulated variable upper limit (S1+16) and the output manipulated variable lower limit (S1+17) as summarized in the table below.

While manual mode is enabled with the auto/manual mode control relay (S2+1) set to on, S1+1 stores 0 through 100 of the manual mode output manipulated variable (S1+18), and D1 stores an indefinite value.

While auto tuning is in progress, S1+1 stores 0 through 100 of the AT output manipulated variable (S1+22), and D1 stores an indefinite value.

Examples of Output Manipulated Variable Values

Output Manipulated Variable Limit Enable (S2+2)	Output Manipulated Variable Upper Limit (S1+16)	Output Manipulated Variable Lower Limit (S1+17)	Manipulated Variable (D1)	Output Manipulated Variable (S1+1)
OFF (disabled)	—	—	≥ 100	100
			1 to 99	1 to 99
			≤ 0	0
ON (enabled)	50	25	≥ 50	50
			26 to 49	26 to 49
			≤ 25	25
	10050	—	≥ 100	50
			1 to 99	(1 to 99) × 0.5
			≤ 0	0

Application Example

This application example demonstrates a PID control for a heater to keep the temperature at 200°C.

In this example, when the program is started, the PID instruction first executes auto tuning according to the designated AT parameters, such as AT sampling period, AT control period, AT set point, and AT output manipulated variable, and also the temperature data inputted to the analog input module. The control output remains on to keep the heater on until the temperature reaches the AT set point of 150°C. Auto tuning determines PID parameters such as proportional gain, integral time, derivative time, and control action.

When the temperature reaches 150°C, PID action starts to control the temperature to 200°C using the derived PID parameters. The heater is turned on and off according to the output manipulated variable calculated by the PID action. When the heater temperature is higher than or equal to 250°C, an alarm light is turned on by the high alarm output.

The analog input module data is also monitored to force off the heater power switch and force on the high alarm light.

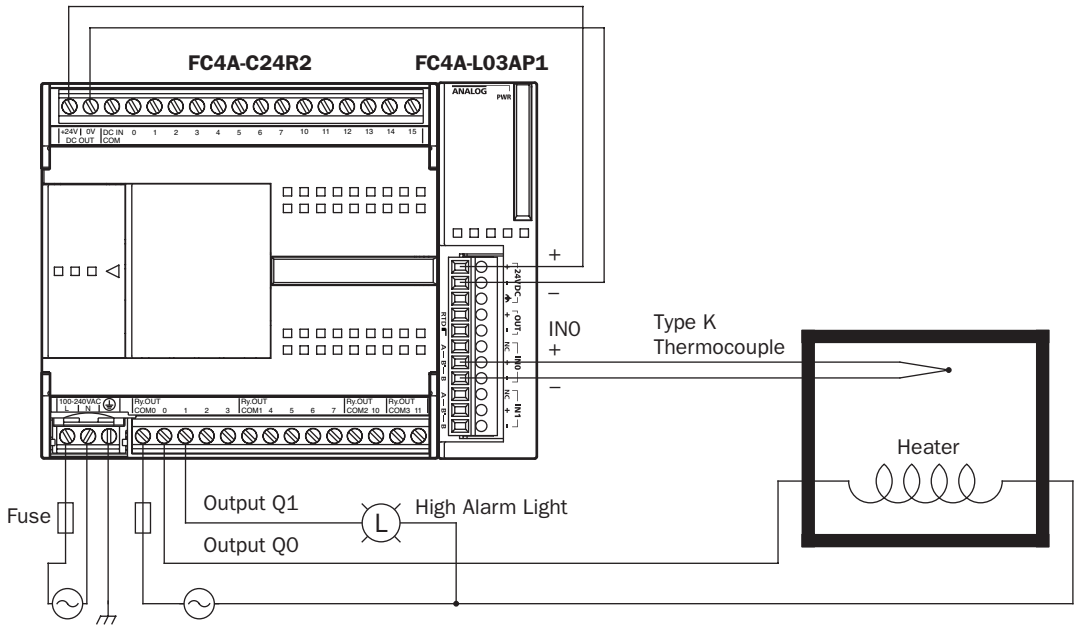
Operand Settings

Operand	Function	Description	Allocation No. (Value)
S1+3	Operation mode	AT (auto tuning) + PID action	D3 (1)
S1+4	Linear conversion	Enable linear conversion	D4 (1)
S1+5	Linear conversion maximum value	1300°C	D5 (13000)
S1+6	Linear conversion minimum value	0°C	D6 (0)
S1+10	Integral start coefficient	100%	D10 (0)
S1+11	Input filter coefficient	70%	D11 (70)
S1+12	Sampling period	500 ms	D12 (50)
S1+13	Control period	1 sec	D13 (10)
S1+14	High alarm value	250°C	D14 (2500)
S1+15	Low alarm value	0°C	D15 (0)
S1+19	AT sampling period	1.5 sec	D19 (150)
S1+20	AT control period	3 sec	D20 (30)
S1+21	AT set point	150°C	D21 (1500)
S1+22	AT output manipulated variable	100% (Note 1)	D22 (100)
S2+1	Auto/manual mode	Auto mode	M1 (OFF)
S2+2	Output manipulated variable limit enable	Disable output manipulated variable limits	M2 (OFF)
S2+3	Integral start coefficient disable	Enable integral start coefficient (S1+10)	M3 (OFF)
S2+4	High alarm output	ON: When temperature ≥ 250°C OFF: When temperature < 250°C	M4
S2+6	Control output	Remains on during auto tuning; Goes on and off according to the control period (S1+13) and output manipulated variable (S1+1) during PID action	M6
S3	Set point	200°C	D100 (2000)
S4	Process variable	Analog input data of analog I/O module 1, analog input channel 0; stores 0 through 4095	D760
	Analog input operating status	Stores 0 through 5	D761
	Analog input signal type	Type K thermometer	D762 (2)
	Analog input data type	12-bit data (0 to 4095) (Note 2)	D763 (0)
D1	Manipulated variable	Stores PID calculation result	D102
	PID start input	Starts to execute the PID instruction	I0
	Monitor input	Starts to monitor the analog input module data for high alarm and operating status	I1
	Heater power switch	Turned on and off by control output M6	Q0
	High alarm light	Turned on and off by high alarm output M4	Q1

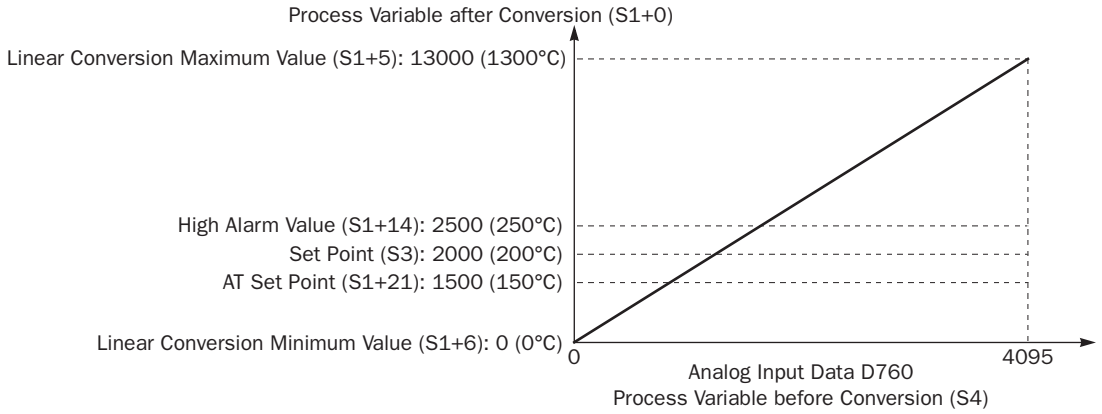
Note 1: The output manipulated variable during auto tuning is a constant value. In this example, the AT output manipulated variable is set to the maximum value of 100 (100%), so the control output (S2+6) remains on during auto tuning.

Note 2: When analog I/O module FC4A-L03AP1 is used for the PID instruction, select the binary data to make sure that the process variable takes a value of 0 through 4095.

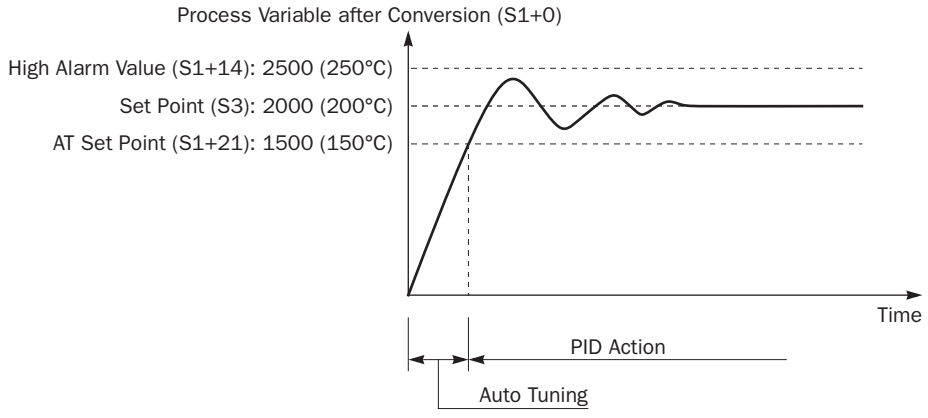
System Setup



Analog Input Data vs. Process Variable after Conversion

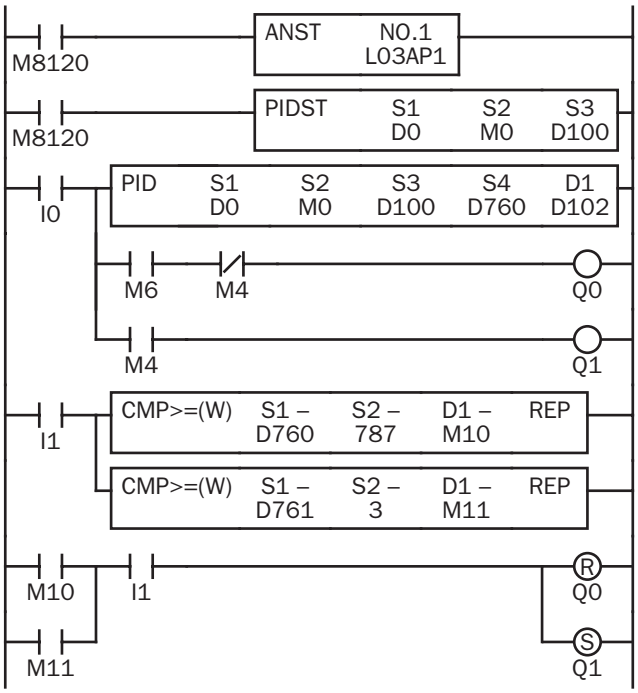


Temperature Control by Auto Tuning and PID Action



Ladder Program

The ladder diagram shown below describes an example of using the PID instruction. The user program must be modified according to the application and simulation must be performed before actual operation.



M8120 is the initialize pulse special internal relay.

When the CPU starts, the ANST (analog macro) instruction stores parameters for the analog I/O module function.

The PIDST (PID macro) instruction also stores parameters for the PID function.

While input I0 is on, the PID instruction is executed.

When internal relay M6 (control output) is turned on, output Q0 (heater power switch) is turned on.

When internal relay M4 (high alarm output) is turned on, output Q1 (high alarm light) is turned on.

While monitor input I1 is on, the temperature is monitored.

When the temperature is higher than or equal to 250°C, M10 is turned on.

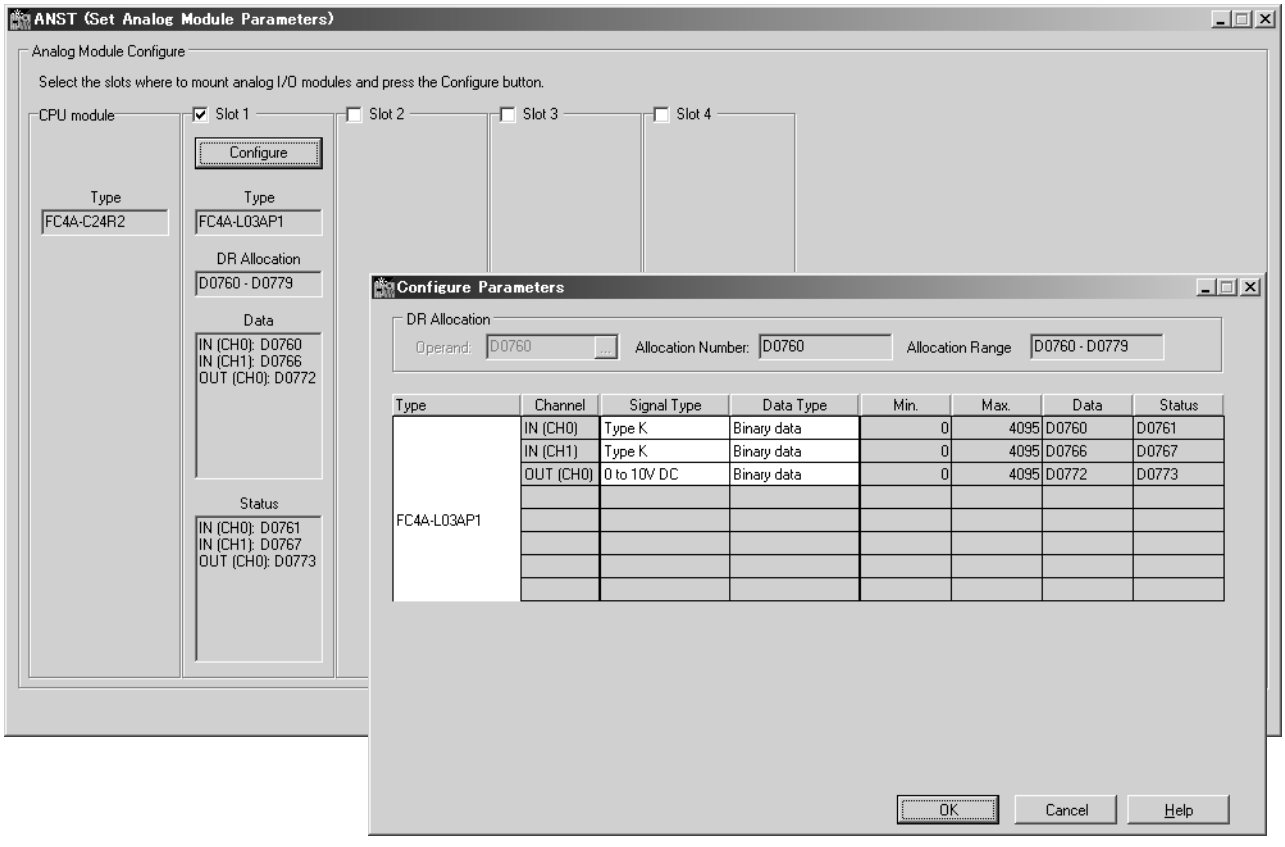
$$4095 \times 250/1300 = 787.5$$

While the analog input operating status (D761) is 3 or higher, M11 is turned on.

When M10 or M11 is on while monitor input I1 is on, Q0 (heater power switch) is forced off and Q1 (high alarm light) is forced on.

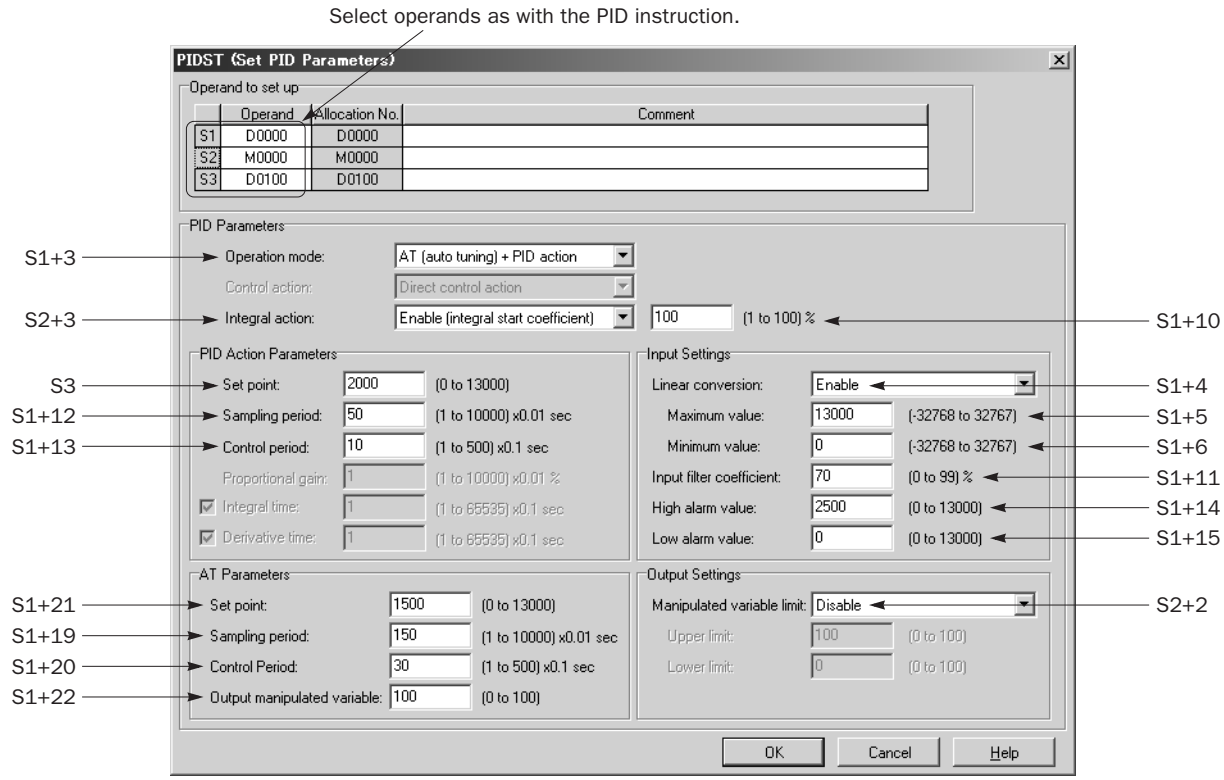
Set Analog Module Parameters (ANST) Dialog Box

WindLDR has a macro to program parameters for analog I/O modules. Place the cursor where to insert the ANST instruction, click the right mouse button, and select **Macro Instructions** > **ANST (Set Analog Module Parameters)**. In the ANST dialog box, press the **Configure** button under Slot 1, and program as shown below.



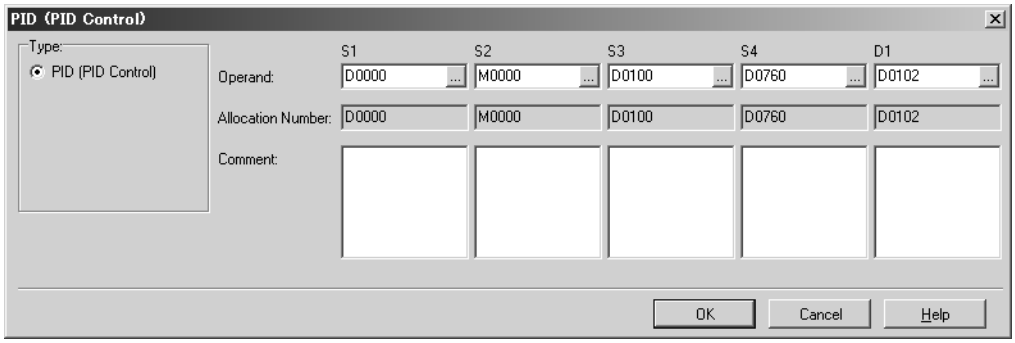
Set PID Parameters (PIDST) Dialog Box

Place the cursor where to insert the PIDST instruction, click the right mouse button, and select **Macro Instructions** > **PIDST (Set PID Parameters)**. In the PIDST dialog box, program as shown below.



S1+3	Operation mode AT + PID action	S1+14	High alarm value
S1+4	Linear conversion enable	S1+15	Low alarm value
S1+5	Linear conversion maximum value	S1+19	AT sampling period
S1+6	Linear conversion minimum value	S1+20	AT control period
S1+10	Integral start coefficient	S1+21	AT set point
S1+11	Input filter coefficient	S1+22	AT output manipulated variable
S1+12	Sampling period	S2+2	Output manipulated variable limit disable
S1+13	Control period	S2+3	Integral start coefficient enable
		S3	Set point

PID Control (PID) Dialog Box



Notes for Using the PID Instruction:

- Since the PID instruction requires continuous operation, keep on the start input for the PID instruction.
- The high alarm output (S2+4) and the low alarm output (S2+5) work while the start input for the PID instruction is on. These alarm outputs, however, do not work when a PID instruction execution error occurs (S1+2 stores 100 through 107) due to data error in control data registers S1+0 through S1+26 or while the start input for the PID instruction is off. Provide a program to monitor the process variable (S4) separately.
- When a PID execution error occurs (S1+2 stores 100 through 107) or when auto tuning is completed, the manipulated variable (D1) stores 0 and the control output (S2+6) turns off.
- Do not use the PID instruction in program branching instructions: LABEL, LJMP, LCAL, LRET, JMP, JEND, MCS, and MCR. The PID instruction may not operate correctly in these instructions.
- The PID instruction, using the difference between the set point (S3) and process variable (S4) as input, calculates the manipulated variable (D1) according to the PID parameters, such as proportional gain (S1+7), integral time (S1+8), and derivative time (S1+9). When the set point (S3) or process variable (S4) is changed due to disturbance, overshoot or undershoot will be caused. Before putting the PID control into actual application, perform simulation tests by changing the set point and process variable (disturbance) to anticipated values in the application.
- The PID parameters, such as proportional gain (S1+7), integral time (S1+8), and derivative time (S1+9), determined by the auto tuning may not always be the optimum values depending on the actual application. To make sure of the best results, adjust the parameters. Once the best PID parameters are determined, perform only the PID action in usual operation unless the control object is changed.
- When a feedback control is executed using the control output (S2+6), the optimum control may not be achieved depending on the controlled object. If this is the case, use of the manipulated variable (D1) in the feedback control is recommended.

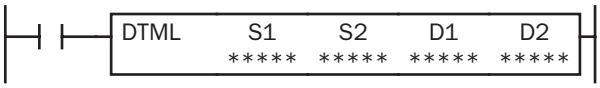
22: DUAL / TEACHING TIMER INSTRUCTIONS

Introduction

Dual timer instructions generate ON/OFF pulses of required durations from a designated output, internal relay, or shift register bit. Four dual timers are available and the ON/OFF duration can be selected from 1 ms up to 65535 sec.

Teaching timer instruction measures the ON duration of the start input for the teaching timer instruction and stores the measured data to a designated data register, which can be used as a preset value for a timer instruction.

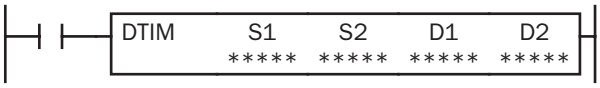
DTML (1-sec Dual Timer)



While input is on, destination operand D1 repeats to turn on and off for a duration designated by operands S1 and S2, respectively.

The time range is 0 through 65535 sec.

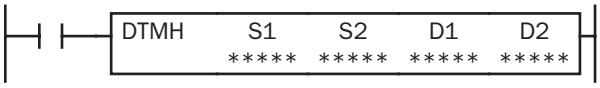
DTIM (100-ms Dual Timer)



While input is on, destination operand D1 repeats to turn on and off for a duration designated by operands S1 and S2, respectively.

The time range is 0 through 6553.5 sec.

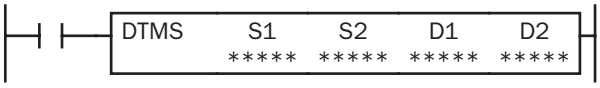
DTMH (10-ms Dual Timer)



While input is on, destination operand D1 repeats to turn on and off for a duration designated by operands S1 and S2, respectively.

The time range is 0 through 655.35 sec.

DTMS (1-ms Dual Timer)



While input is on, destination operand D1 repeats to turn on and off for a duration designated by operands S1 and S2, respectively.

The time range is 0 through 65.535 sec.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant
S1 (Source 1)	ON duration	—	—	—	—	—	—	X	0-65535
S2 (Source 2)	OFF duration	—	—	—	—	—	—	X	0-65535
D1 (Destination 1)	Dual timer output	—	X	▲	X	—	—	—	—
D2 (Destination 2)	System work area	—	—	—	—	—	—	D0-D7998	—

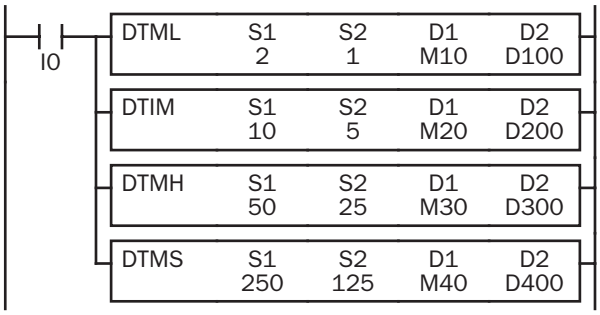
For the valid operand number range, see page 6-2.

▲ Internal relays M0 through M1277 can be designated as D1. Special internal relays cannot be designated as D1.

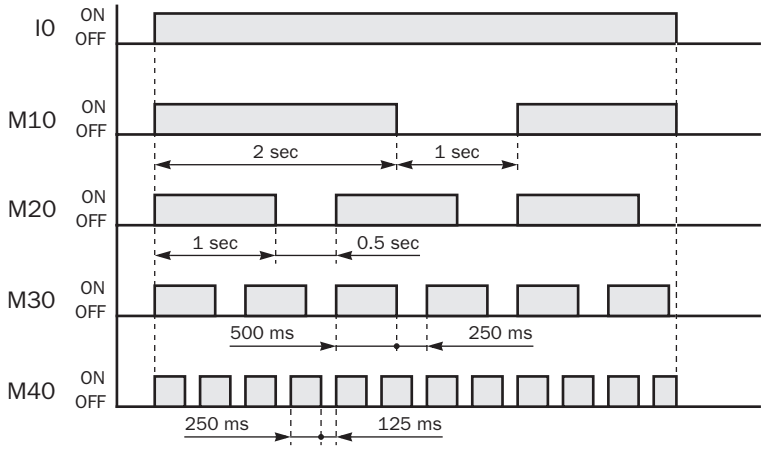
Destination operand D2 (system work area) uses 2 data registers starting with the operand designated as D2. Data registers D0 through D1298 and D2000 through D7998 can be designated as D2. The two data registers are used for a system work area. Do not use these data registers for destinations of other advanced instructions, and do not change values of these data registers using the Point Write function on WindLDR. If the data in these data registers are changed, the dual timer does not operate correctly.

The dual timer instructions cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Examples: DTML, DTIM, DTMH, DTMS



While input I0 is on, four dual timer instructions turn on and off the destination operands according to the on and off durations designated by source operands S1 and S2.



Instruction	Increments	S1	ON duration	S2	OFF duration
DTML	1 sec	2	1 sec × 2 = 2 sec	1	1 sec × 1 = 1 sec
DTIM	100 ms	10	100 ms × 10 = 1 sec	5	100 ms × 5 = 0.5 sec
DTMH	10 ms	50	10 ms × 50 = 500 ms	25	10 ms × 25 = 250 ms
DTMS	1 ms	250	1 ms × 250 = 250 ms	125	1 ms × 125 = 125 ms

For the timer accuracy of timer instructions, see page 7-8.

TTIM (Teaching Timer)

While input is on, the ON duration is measured in units of 100 ms and the measured value is stored to a data register designated by destination operand D1.

The measured time range is 0 through 6553.5 sec.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	—	—	X

Valid Operands

Operand	Function	I	Q	M	R	T	C	D	Constant
D1 (Destination 1)	Measured value	—	—	—	—	—	—	D0-D7997	—

For the valid operand number range, see page 6-2.

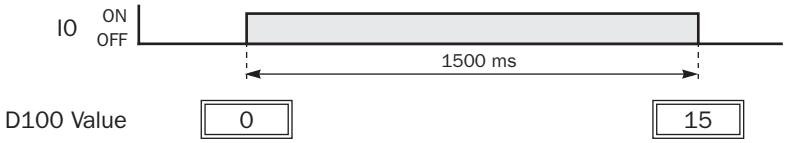
Destination operand D1 (measured value) uses 3 data registers starting with the operand designated as D1. Data registers D0 through D1297 and D2000 through D7997 can be designated as D1. Subsequent two data registers starting with destination operand D1+1 are used for a system work area. Do not use these two data registers for destinations of other advanced instructions, and do not change values of these data registers using the Point Write function on WindLDR. If the data in these data registers are changed, the teaching timer does not operate correctly.

The teaching timer instruction cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

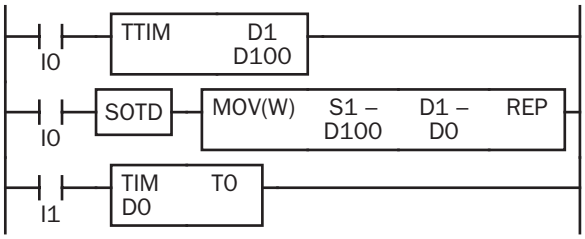
Examples: TTIM

When input I0 is turned on, TTIM resets data register D100 to zero and starts to store the ON duration of input I0 to data register D100, measured in units of 100 ms.

When input I0 is turned off, TTIM stops the measurement, and data register D100 maintains the measured value of the ON duration.



The following example demonstrates a program to measure the ON duration of input I0 and to use the ON duration as a preset value for 100-ms timer instruction TIM.



While input I0 is on, TTIM measures the ON duration of input I0 and stores the measured value in units of 100 ms to data register D100.

When input I0 is turned off, MOV(W) stores the D100 value to data register D0 as a preset value for timer T0.

When input I1 is turned on, 100-ms timer T0 starts to operate with a preset value stored in data register D0.

23: INTELLIGENT MODULE ACCESS INSTRUCTIONS

Introduction

Intelligent module access instructions are used to read or write data between the CPU module and a maximum of seven intelligent modules while the CPU module is running or when the CPU module is stopped.

Upgrade Information

Upgraded CPU modules can use the intelligent module access instructions. Applicable CPU modules and system program version are shown in the table below. For the procedure to confirm the system program version of the CPU module, see page 29-1.

CPU Module	All-in-One Type			Slim Type	
	FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
System Program Version	—	—	204 or higher	204 or higher	203 or higher

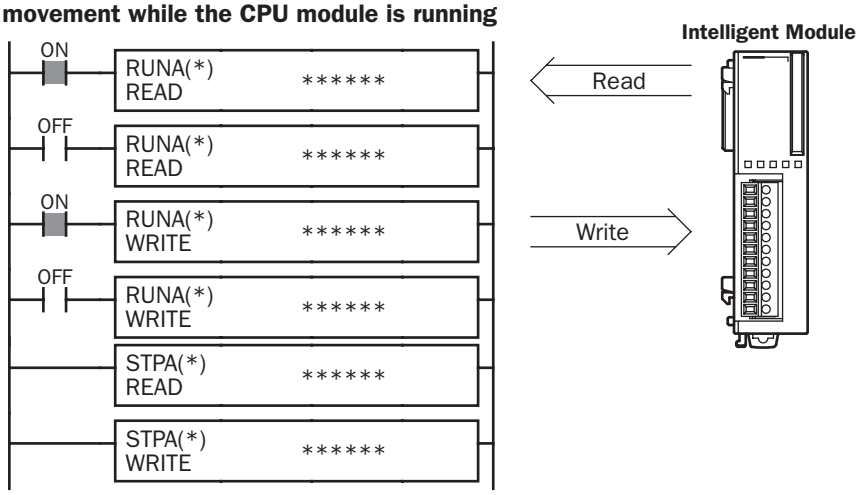
Use WindLDR ver. 4.50 or higher to program the intelligent module access instructions.

Intelligent Module Access Overview

The Run Access Read instruction reads data from the designated address in the intelligent module and stores the read data to the designated operand while the CPU module is running. The Run Access Write instruction writes data from the designated operand to the designated address in the intelligent module while the CPU module is running.

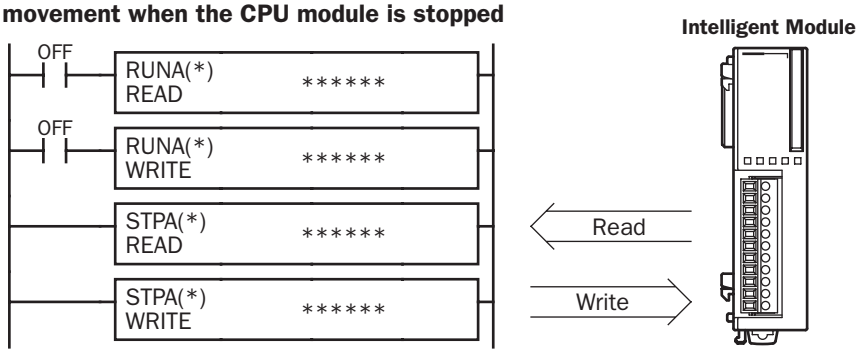
The Stop Access Read instruction reads data from the designated address in the intelligent module and stores the read data to the designated operand when the CPU module is stopped. The Stop Access Write instruction writes data from the designated operand to the designated address in the intelligent module when the CPU module is stopped.

Data movement while the CPU module is running



While the CPU module is running and the input is on, RUNA READ is executed to read data from the intelligent module, and RUNA WRITE to write data to the intelligent module.

Data movement when the CPU module is stopped



When the CPU module is stopped, STPA READ is executed to read data from the intelligent module, and STPA WRITE to write data to the intelligent module.

RUNA READ (Run Access Read)

While input is on, data is read from the area starting at ADDRESS in the intelligent module designated by SLOT and stored to the operand designated by DATA.

BYTE designates the quantity of data to read.

STATUS stores the operating status code.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands (Run Access Read)

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
DATA	First operand number to store read data	—	X	▲	X	X	X	X	—	—
STATUS	Operating status code	—	—	—	—	—	—	X	—	—
SLOT	Intelligent module slot number	—	—	—	—	—	—	—	1-7	—
ADDRESS	First address in intelligent module to read data from	—	—	—	—	—	—	—	0-127	—
BYTE	Bytes of data to read	—	—	—	—	—	—	—	1-127	—

For the valid operand number range, see pages 6-1 and 6-2.

DATA: Specify the first operand number to store the data read from the intelligent module.

▲ Internal relays M0 through M1277 and AS-Interface internal relays M1300 through M1997 can be designated as DATA. Special internal relays cannot be designated as DATA.

When T (timer) or C (counter) is used as DATA for Run Access Read, the data read from the intelligent module is stored as a preset value which can be 0 through 65535.

All data registers, including special data registers, AS-Interface data registers, and expansion data registers, can be designated as DATA.

STATUS: Specify a data register to store the operating status code. Only data registers D0 through D1299 can be designated as STATUS. Special data registers, AS-Interface data registers, and expansion data registers *cannot* be designated whether the AS-Interface master module is used or not. For status code description, see page 23-6.

SLOT: Enter the slot number where the intelligent module is mounted. A maximum of seven intelligent modules can be used.

ADDRESS: Specify the first address in the intelligent module to read data from.

BYTE: Specify the quantity of data to read in bytes.

The RUNA READ instruction cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as DATA, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as DATA, 1 point is used.

RUNA WRITE (Run Access Write)

While input is on, data in the area starting at the operand designated by DATA is written to ADDRESS in the intelligent module designated by SLOT.

BYTE designates the quantity of data to write.

STATUS stores the operating status code.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands (Run Access Write)

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
DATA	First operand number to extract data from	X	X	X	X	X	X	X	X	X
STATUS	Operating status code	—	—	—	—	—	—	X	—	—
SLOT	Intelligent module slot number	—	—	—	—	—	—	—	1-7	—
ADDRESS	First address in intelligent module to write data to	—	—	—	—	—	—	—	0-127	—
BYTE	Bytes of data to write	—	—	—	—	—	—	—	1-127	—

For the valid operand number range, see pages 6-1 and 6-2.

DATA: Specify the first operand number to extract the data to write to the intelligent module.

When T (timer) or C (counter) is used as DATA for Run Access Write, the timer/counter current value is written to the intelligent module.

All data registers, including special data registers, AS-Interface data registers, and expansion data registers, can be designated as DATA.

When a constant is designated as DATA, Repeat cannot be selected. For details about the data movement with or without Repeat, see page 23-7.

STATUS: Specify a data register to store the operating status code. Only data registers D0 through D1299 can be designated as STATUS. Special data registers, AS-Interface data registers, and expansion data registers *cannot* be designated whether the AS-Interface master module is used or not. For status code description, see page 23-6.

SLOT: Enter the slot number where the intelligent module is mounted. A maximum of seven intelligent modules can be used.

ADDRESS: Specify the first address in the intelligent module to store the data.

BYTE: Specify the quantity of data to write in bytes.

The RUNA WRITE instruction cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as DATA, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as DATA, 1 point is used.

STPA READ (Stop Access Read)

Start input is not needed for this instruction.

When the CPU module stops, data is read from the area starting at ADDRESS in the intelligent module designated by SLOT and stored to the operand designated by DATA.

BYTE designates the quantity of data to read.

STATUS stores the operating status code.

Note: STPA READ and STPA WRITE instructions can be used 64 times in a user program. When more than 64 STPA READ and STPA WRITE instructions are used in a user program, the excess instructions are not executed and error code 7 is stored in the data register designated as STATUS.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands (Stop Access Read)

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
DATA	First operand number to store read data	—	X	▲	X	X	X	X	—	—
STATUS	Operating status code	—	—	—	—	—	—	X	—	—
SLOT	Intelligent module slot number	—	—	—	—	—	—	—	1-7	—
ADDRESS	First address in intelligent module to read data from	—	—	—	—	—	—	—	0-127	—
BYTE	Bytes of data to read	—	—	—	—	—	—	—	1-127	—

For the valid operand number range, see pages 6-1 and 6-2.

DATA: Specify the first operand number to store the data read from the intelligent module.

▲ Internal relays M0 through M1277 and AS-Interface internal relays M1300 through M1997 can be designated as DATA. Special internal relays cannot be designated as DATA.

When T (timer) or C (counter) is used as DATA for Stop Access Read, the data read from the intelligent module is stored as a preset value which can be 0 through 65535.

All data registers, including special data registers, AS-Interface data registers, and expansion data registers, can be designated as DATA.

STATUS: Specify a data register to store the operating status code. Only data registers D0 through D1299 can be designated as STATUS. Special data registers, AS-Interface data registers, and expansion data registers *cannot* be designated whether the AS-Interface master module is used or not. For status code description, see page 23-6.

SLOT: Enter the slot number where the intelligent module is mounted. A maximum of seven intelligent modules can be used.

ADDRESS: Specify the first address in the intelligent module to read data from.

BYTE: Specify the quantity of data to read in bytes.

The STPA READ instruction cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

If a STPA READ instruction is programmed between MCS and MCR instructions, the STPA READ instruction is executed when the CPU module is stopped regardless whether the input condition for the MCS instruction is on or off. For MCS and MCR instructions, see page 7-23.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as DATA, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as DATA, 1 point is used.

STPA WRITE (Stop Access Write)

Start input is not needed for this instruction.

When the CPU module stops, data in the area starting at the operand designated by DATA is written to ADDRESS in the intelligent module designated by SLOT.

BYTE designates the quantity of data to write.

STATUS stores the operating status code.

Note: STPA READ and STPA WRITE instructions can be used 64 times in a user program. When more than 64 STPA READ and STPA WRITE instructions are used in a user program, the excess instructions are not executed and error code 7 is stored in the data register designated as STATUS.

Applicable CPU Modules

FC4A-C10R2/C	FC4A-C16R2/C	FC4A-C24R2/C	FC4A-D20K3/S3	FC4A-D20RK1/RS1 & FC4A-D40K3/S3
—	—	X	X	X

Valid Operands (Run Access Write)

Operand	Function	I	Q	M	R	T	C	D	Constant	Repeat
DATA	First operand number to extract data from	X	X	X	X	X	X	X	X	X
STATUS	Operating status code	—	—	—	—	—	—	X	—	—
SLOT	Intelligent module slot number	—	—	—	—	—	—	—	1-7	—
ADDRESS	First address in intelligent module to write data to	—	—	—	—	—	—	—	0-127	—
BYTE	Bytes of data to write	—	—	—	—	—	—	—	1-127	—

For the valid operand number range, see pages 6-1 and 6-2.

DATA: Specify the first operand number to extract the data to write to the intelligent module.

When T (timer) or C (counter) is used as DATA for Stop Access Write, the timer/counter current value is written to the intelligent module.

All data registers, including special data registers, AS-Interface data registers, and expansion data registers, can be designated as DATA.

When a constant is designated as DATA, Repeat cannot be selected. For details about the data movement with or without Repeat, see page 23-7.

STATUS: Specify a data register to store the operating status code. Only data registers D0 through D1299 can be designated as STATUS. Special data registers, AS-Interface data registers, and expansion data registers *cannot* be designated whether the AS-Interface master module is used or not. For status code description, see page 23-6.

SLOT: Enter the slot number where the intelligent module is mounted. A maximum of seven intelligent modules can be used.

ADDRESS: Specify the first address in the intelligent module to store the data.

BYTE: Specify the quantity of data to write in bytes.

The STPA WRITE instruction cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the CPU module.

If a STPA WRITE instruction is programmed between MCS and MCR instructions, the STPA WRITE instruction is executed when the CPU module is stopped regardless whether the input condition for the MCS instruction is on or off. For MCS and MCR instructions, see page 7-23.

Valid Data Types

W (word)	I (integer)
X	X

When a bit operand such as I (input), Q (output), M (internal relay), or R (shift register) is designated as DATA, 16 points are used.

When a word operand such as T (timer), C (counter), or D (data register) is designated as DATA, 1 point is used.

Intelligent Module Access Status Code

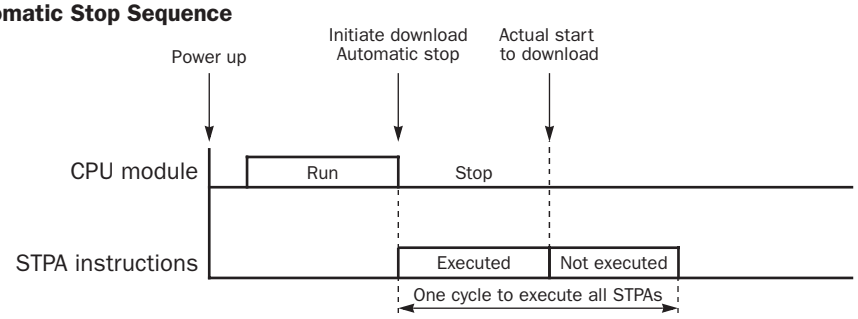
The data register designated as STATUS stores a status code to indicate the operating status and error of the intelligent module access operation. When status code 1, 3, or 7 is stored, take a corrective measure as described in the table below:

Status Code	Status	Description	RUNA	STPA
0	Normal	Intelligent module access is normal.	X	X
1	Bus error	The intelligent module is not installed correctly. Power down the MicroSmart modules, and re-install the intelligent module correctly.	X	X
3	Invalid module number	The designated module number is not found. Confirm the intelligent module number and correct the program.	X	X
7	Excessive multiple usage	More than 64 STPA READ and STPA WRITE instructions are used. Eliminate the excess instructions.	—	X

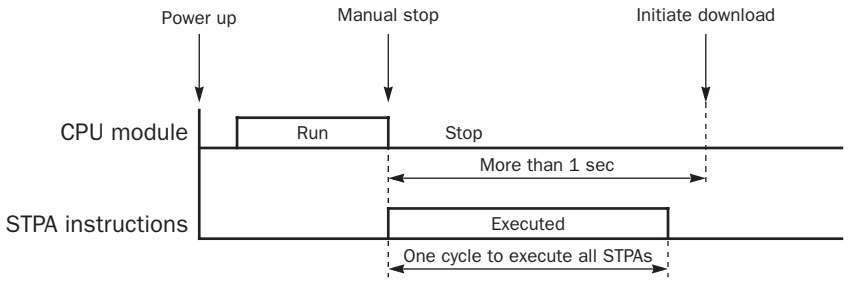
STPA Execution during Program Download

When downloading a user program, the CPU module is automatically stopped as default. Depending on the timing of the initiation of the download and the total time to execute all STPA Read and Write instructions, some of the STPA instructions may not be executed. If this is the case, manually stop the CPU module. After more than 1 second, initiate user program download as shown in the chart below.

Automatic Stop Sequence

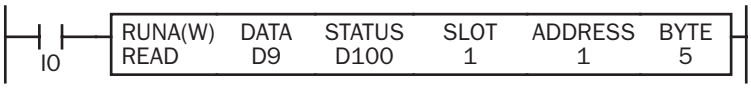


Manual Stop Sequence



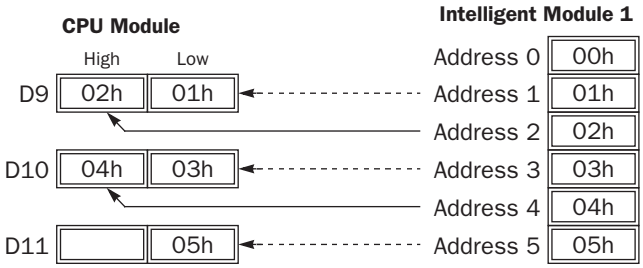
Example: RUNA READ

The following example illustrates the data movement of the RUNA READ instruction. The data movement of the STPA READ is the same as the RUNA READ instruction.



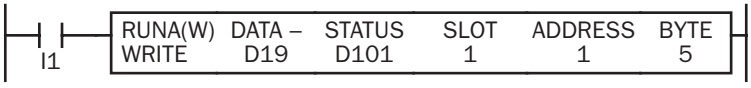
While input I0 is on, data of 5 bytes is read from the area starting at address 1 in intelligent module 1 and stored to the 5-byte area in data registers starting at D9.

Status code is stored in data register D100.



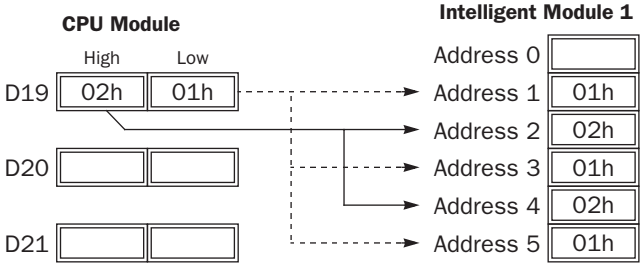
Example: RUNA WRITE without Repeat

The following example illustrates the data movement of the RUNA WRITE instruction without repeat designation. The data movement of the STPA WRITE is the same as the RUNA WRITE instruction.



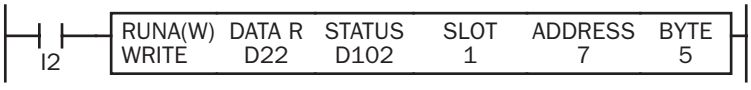
While input I1 is on, data in data register D19 is written to the 5-byte area starting at address 1 in intelligent module 1.

Status code is stored in data register D101.



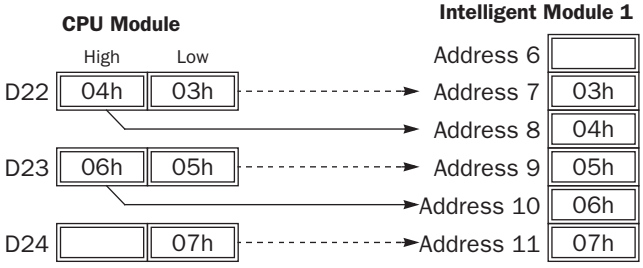
Example: RUNA WRITE with Repeat

The following example illustrates the data movement of the RUNA WRITE instruction with repeat designation. The data movement of the STPA WRITE is the same as the RUNA WRITE instruction.



While input I2 is on, data in 5-byte area starting at data register D22 is written to the 5-byte area starting at address 7 in intelligent module 1.

Status code is stored in data register D102.



24: ANALOG I/O CONTROL

Introduction

The MicroSmart provides analog I/O control capabilities of 12- through 16-bit resolution using analog I/O modules.

This chapter describes the system setup for using analog I/O modules, WindLDR programming procedures, data register allocation numbers for analog I/O modules, and application examples.

For specifications of analog I/O modules, see page 2-43.

Applicable CPU Modules

END refresh type analog I/O modules as many as listed below can be used with any FC4A MicroSmart CPU module system program versions.

Ladder refresh type analog I/O modules can be used with the FC4A MicroSmart CPU module system program versions as listed below.

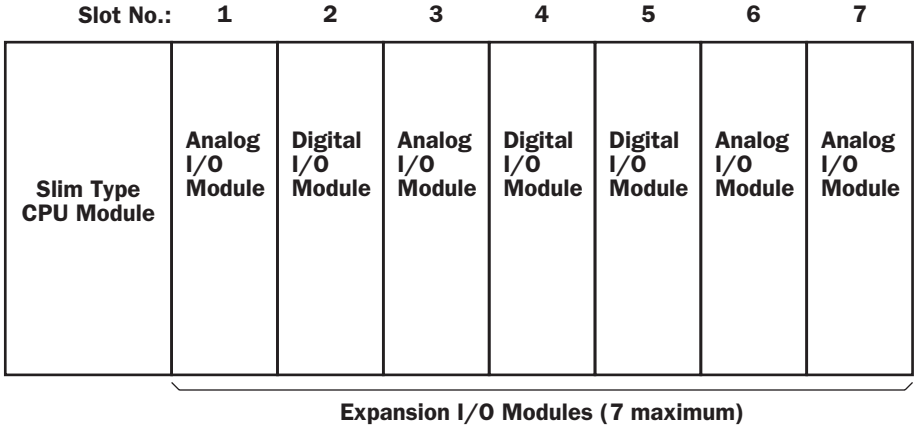
All-in-one 10- and 16-I/O type CPU modules cannot use either END refresh or ladder refresh type analog I/O modules.

FC4A MicroSmart CPU Module		All-in-One Type			Slim Type	
		FC4A-C10R2 FC4A-C10R2C	FC4A-C16R2 FC4A-C16R2C	FC4A-C24R2 FC4A-C24R2C	FC4A-D20K3 FC4A-D20S3	FC4A-D20RK1 FC4A-D20RS1 FC4A-D40K3 FC4A-D40S3
Applicable CPU System Program Version	End Refresh	—	—	Any	Any	Any
	Ladder Refresh	—	—	204 or higher	204 or higher	203 or higher
Quantity of Analog I/O Modules		—	—	4	7	7

System Setup

The FC4A MicroSmart CPU modules can be used with a maximum of seven expansion I/O modules, which include digital I/O modules and analog I/O modules.

System Setup Example



• Slot No.

Indicates the position where the expansion module is mounted. The slot number starts with 1 next to the CPU module up to a maximum of 7.

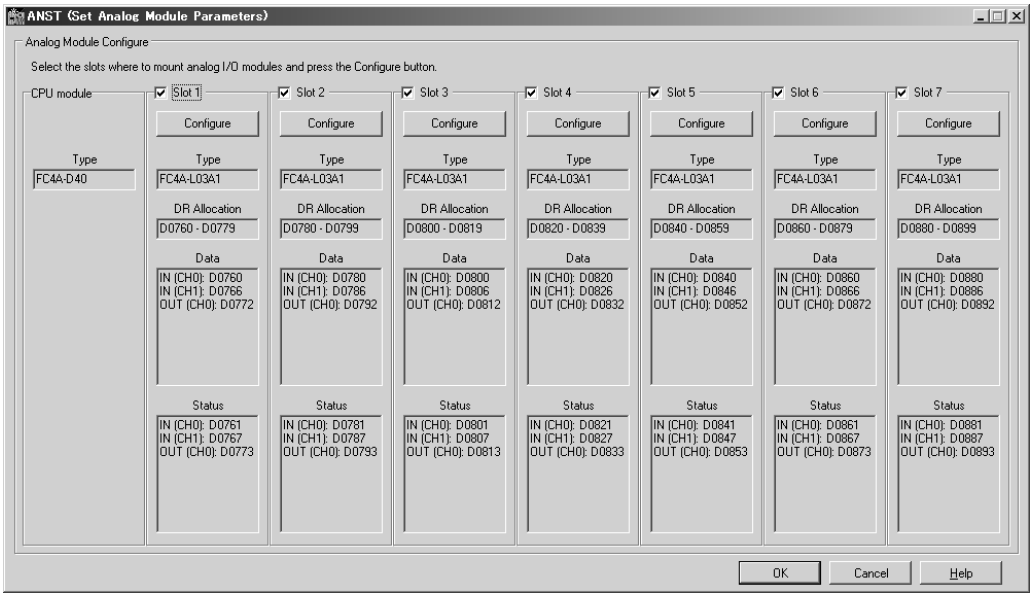
Programming WindLDR

Use WindLDR ver. 5.0 or later which has the ANST (Set Analog Module Parameters) macro for easy programming of analog I/O modules. For a start input of the ANST macro, use special internal relay M8120 (initialize pulse) to execute the ANST macro only once after starting the CPU.

1. Click the **ANST** icon from the WindLDR tool bar, then place the cursor where you want to insert the ANST instruction on the ladder editing screen, and click the mouse.

   Or, place the cursor where you want to insert the ANST instruction on the ladder editing screen, and type **ANST**.

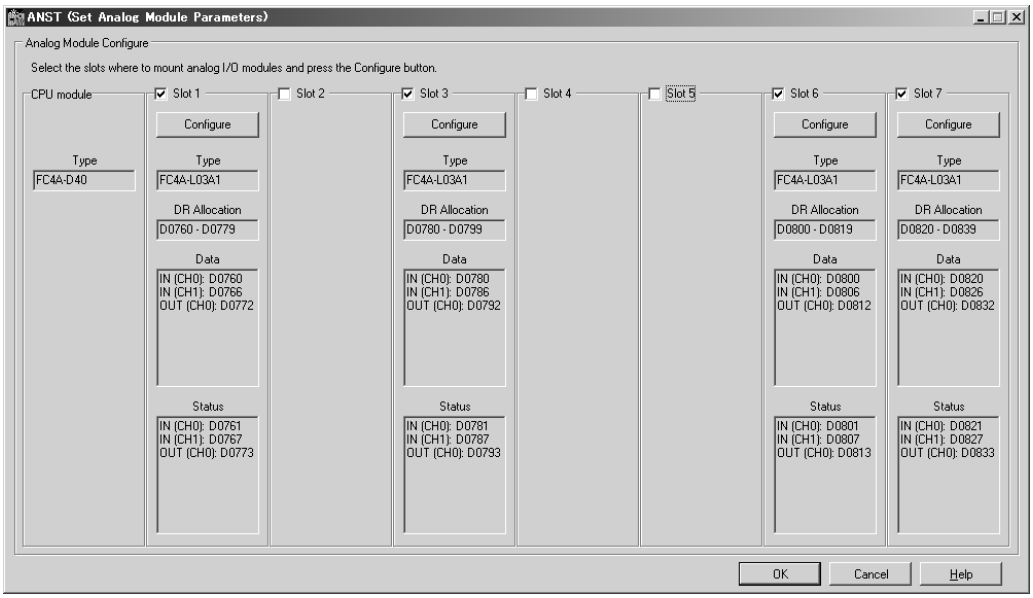
The Set Analog Module Parameters dialog box appears.



2. Select the slots where analog I/O modules are mounted.

All slots are selected to use seven analog I/O modules as default. Click the check box to deselect slots where analog I/O modules are *not* mounted.

When using analog I/O modules on Slots 1, 3, 6, and 7, deselect Slots 2, 4, and 5 as shown below.

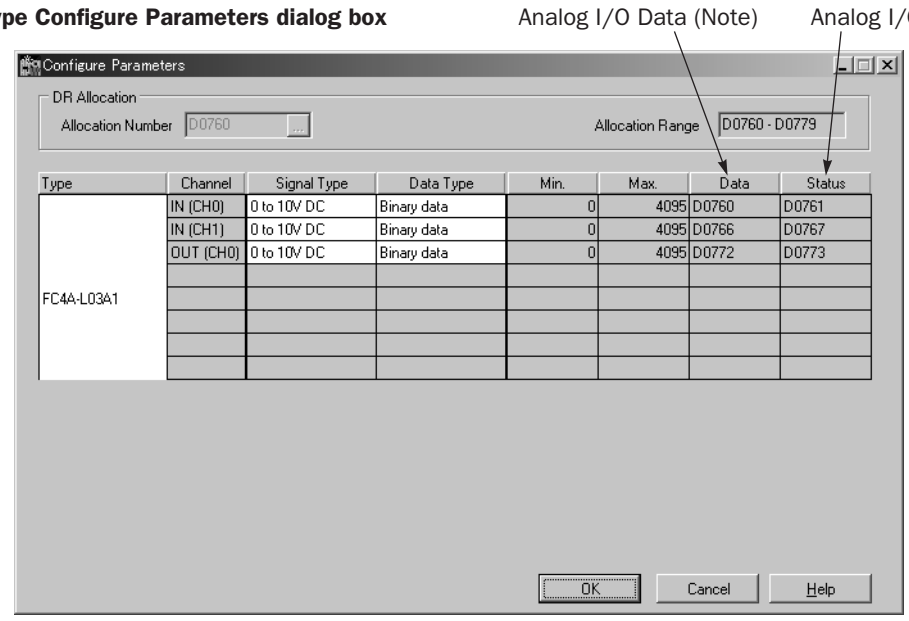


3. Click the **Configure** button under the selected slots.

The Configure Parameters dialog box appears. All parameters for analog I/O control can be set in this dialog box. Available parameters vary with the type of the analog I/O module.

END Refresh Type Configure Parameters dialog box

FC4A-L03A1
FC4A-L03AP1
FC4A-J2A1
FC4A-K1A1



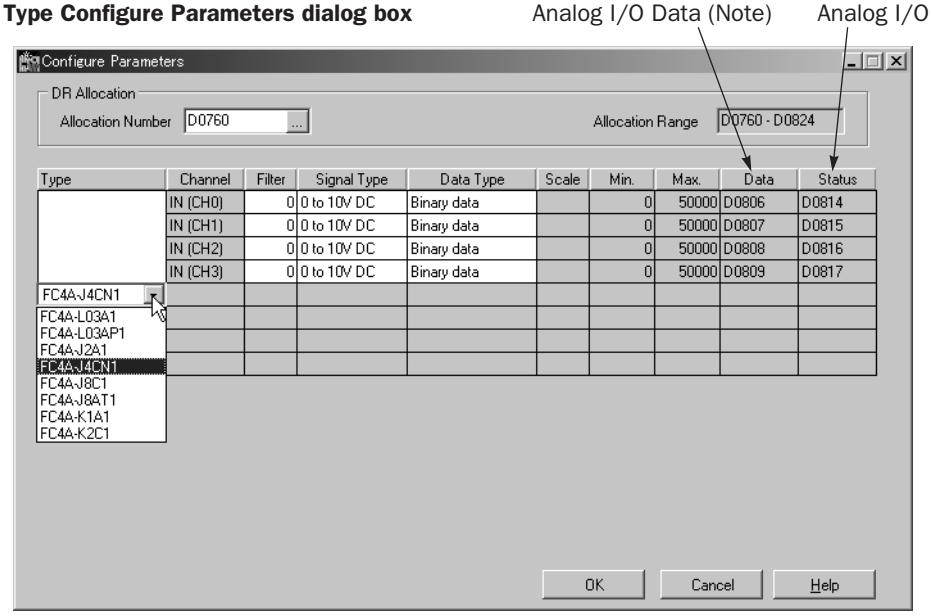
4. Select the type of the analog I/O module.

Click on the right of the analog I/O module Type No., then a pull-down list shows eight available modules.

Depending on the selected analog I/O module, other parameters available for the selected module are shown.

Ladder Refresh Type Configure Parameters dialog box

FC4A-J4CN1
FC4A-J8C1
FC4A-J8AT1
FC4A-K2C1



In the Configure Parameters dialog box, parameters in white cells are selectable while gray cells indicate default parameters. In the white cells, optional values can be selected from a pull-down list or entered by typing required values.

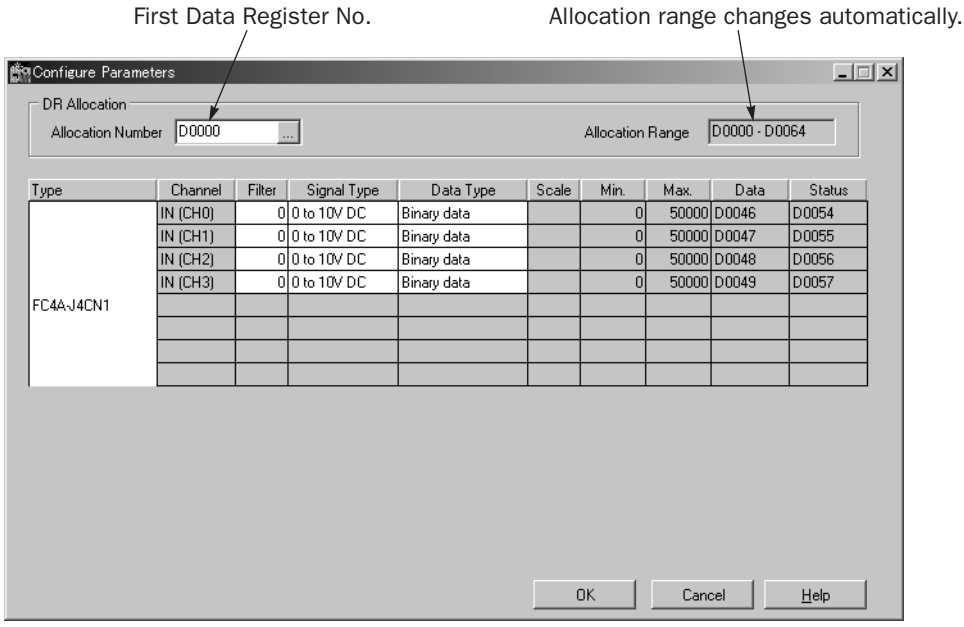
Note for PID Instruction Source Operand S4 (process variable)

When using the PID instruction, specify the data register number shown under Data in the Configure Parameters dialog box as source operand S4 (process variable) of the PID instruction. The analog input data in the selected data register is used as the process variable of the PID instruction.

5. Select a DR allocation number (Ladder refresh type only).

CPU Module	DR Allocation
END Refresh Type FC4A-L03A1 FC4A-L03AP1 FC4A-J2A1 FC4A-K1A1	DR allocation starts with D760 as default, and the first DR number cannot be changed. One analog I/O module occupies 20 data registers. When a maximum of seven analog I/O modules are used, data registers D760 through D899 are used for analog I/O control.
Ladder Refresh Type FC4A-J4CN1 FC4A-J8C1 FC4A-J8AT1 FC4A-K2C1	The first data register can be selected as required. Enter the first DR number used for analog I/O control. One analog input module occupies a maximum of 65 data registers. One analog output module occupies 15 data registers.

Ladder Refresh Type Configure Parameters dialog box



6. Enter a filter value (Ladder refresh type analog input modules only).

The filter function is available for the FC4A-J4CN1, FC4A-J8C1, and FC4A-J8AT1 only. Filtering ensures smooth input of analog data into the CPU module.

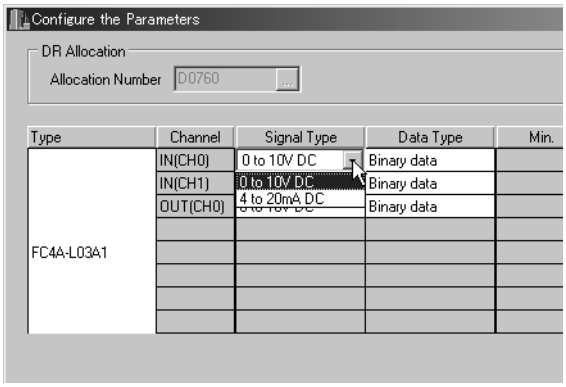
Filter Value	Description
0	Without filter function
1 to 255	The average of N pieces of analog input data is read as analog input data, where N is the designated filter value. $\text{Analog input data} = \frac{(\text{Previous analog input data}) \times (\text{Filter value}) + (\text{Current analog input data})}{(\text{Filter value}) + 1}$

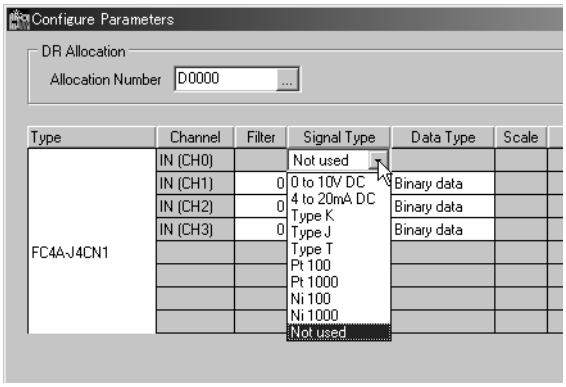
7. Select a signal type for each channel.

Click on the right of the Signal Type field, then a pull-down list appears to show all available input or output signal types.

When you do not use any input or output signal, select the default value or **Not used** for the channel.

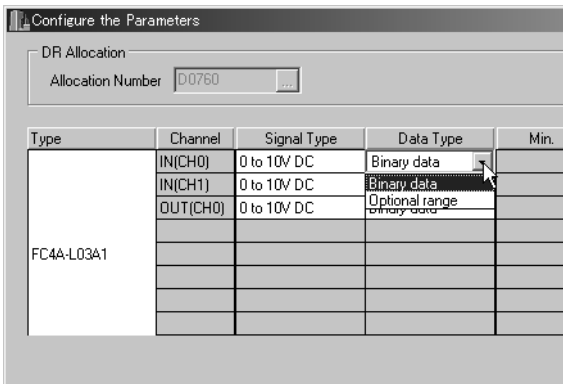
Analog I/O Module		For unused channel, select
END Refresh Type	FC4A-L03A1, FC4A-J2A1	0 to 10V DC
	FC4A-L03AP1	Type K
Ladder Refresh Type	FC4A-J4CN1, FC4A-J8C1, FC4A-J8AT1, FC4A-K2C1	Not used

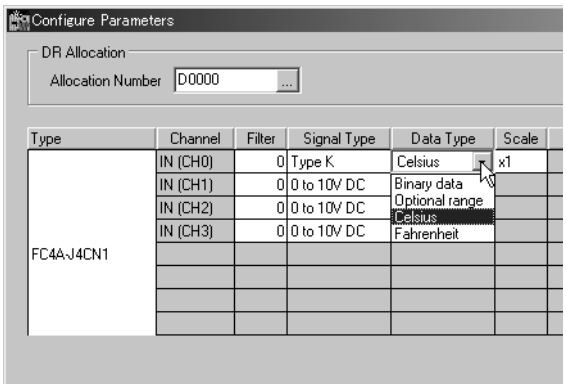




8. Select a data type for each channel.

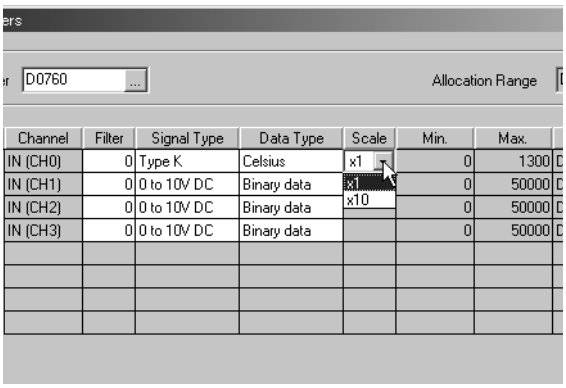
Click on the right of the Data Type field, then a pull-down list appears to show all available input or output data types.

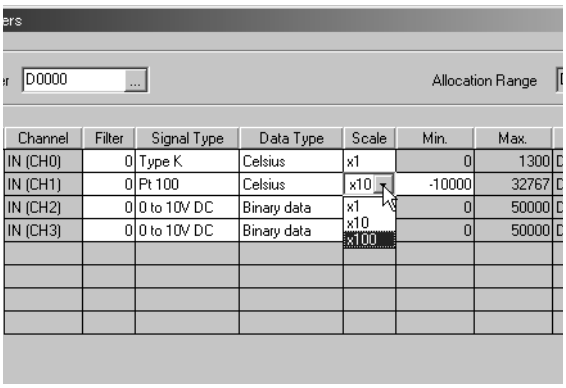




9. Select a scale value (Ladder refresh type analog input modules only).

When Celsius or Fahrenheit is selected for thermocouple, resistance thermometer, or thermistor signal types on ladder refresh type analog input modules, the scale value can be selected from ×1, ×10, or ×100 depending on the selected signal type. Using this function, the analog input data can be multiplied to ensure precise control.



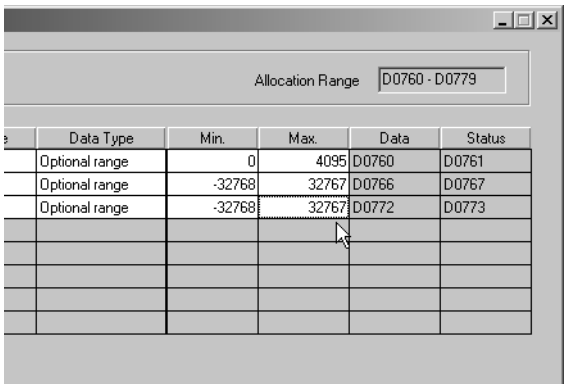


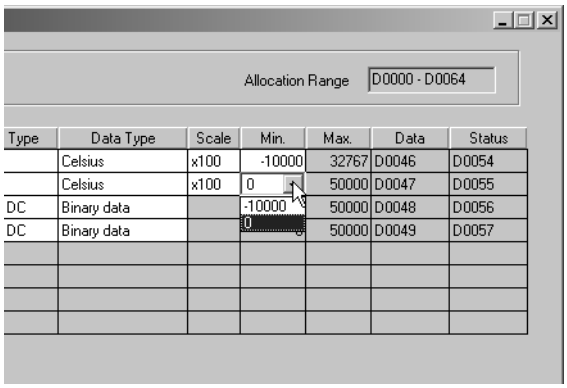
10. Select maximum and minimum values.

For analog input values, when Optional range is selected for the Data Type, designate the analog input data minimum and maximum values which can be –32,768 through 32,767.

In addition, when using resistance thermometers (Pt100, Pt1000, Ni100, or Ni1000) with the Celsius or Fahrenheit Data Type and the ×100 scale, select the analog input data minimum value from 0 or another value in the pull-down list. The maximum value is changed automatically according to the selected minimum value.

For analog output values, when Optional range is selected for the Data Type, designate the analog output data minimum and maximum values which can be –32,768 through 32,767.





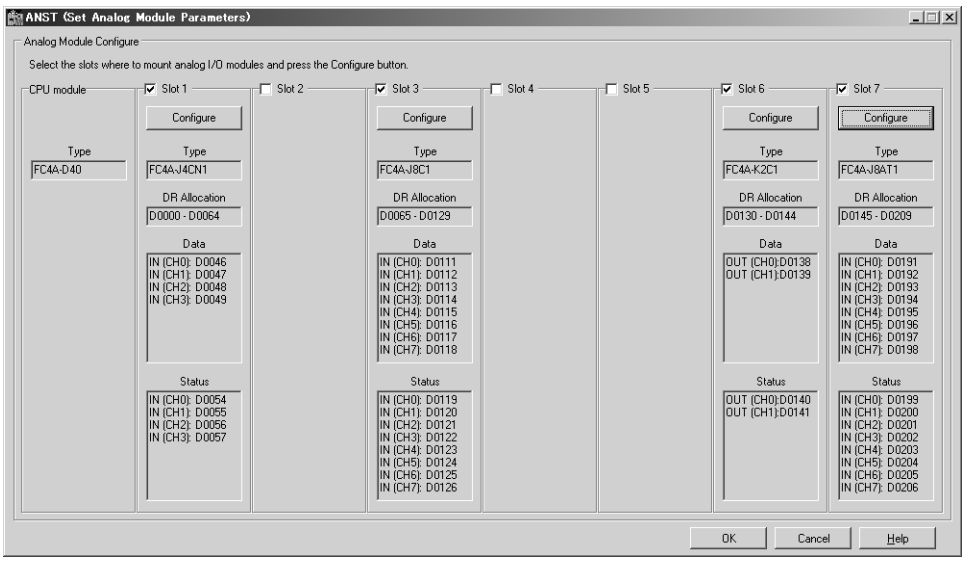
11. View the data register numbers allocated to Data and Status.

Parameter		DR Allocation
Data	**Analog I/O Data** Stores the digital data converted from an analog input signal or converted into an analog output signal. Designated as source operand S4 (process variable) of the PID instruction.	**END Refresh Type** Data registers are automatically allocated depending on the slot where the analog I/O module is mounted. **Ladder Refresh Type** Data registers are automatically allocated depending on the number designated in the DR Allocation Number field.
Status	**Analog I/O Operating Status** Stores an analog I/O operating status code. See pages 24-13 and 24-15.	

12. Click the **OK** button to save changes and exit the Configure Parameter dialog box.

13. Repeat the same steps for other slots.

14. When finished, click the **OK** button to save changes and exit the Set Analog Module Parameters dialog box.



Analog I/O Control Parameters

Available parameters for analog I/O control depend on the type of analog I/O modules as summarized in the following table. Designate the parameters in the Configure Parameters dialog box of the ANST macro as required by your application.

Parameter	Analog I/O Module		Analog Input Module				Analog Output Module	
	END Refresh Type		Ladder Refresh Type				END	Ladder
	FC4A-L03A1	FC4A-L03AP1	FC4A-J2A1	FC4A-J4CN1	FC4A-J8C1	FC4A-J8AT1	FC4A-K1A1	FC4A-K2C1
Analog Input Signal Type	X	X	X	X	X	X	—	—
	Page 24-11		Page 24-11				—	
Analog Input Data Type	X	X	X	X	X	X	—	—
	Page 24-11		Page 24-11				—	
Analog Input Data Minimum/Maximum Values	X	X	X	X	X	X	—	—
	Page 24-13		Page 24-13				—	
Filter Value	—	—	—	X	X	X	—	—
	—		—	Page 24-13			—	
Thermistor Parameter	—	—	—	—	—	X	—	—
	—		—			24-13	—	
Analog Input Data	X	X	X	X	X	X	—	—
	Page 24-13		Page 24-13				—	
Analog Input Operating Status	X	X	X	X	X	X	—	—
	Page 24-13		Page 24-13				—	
Analog Output Signal Type	X	X	—	—	—	—	X	X
	Page 24-15		—				Page 24-15	
Analog Output Data Type	X	X	—	—	—	—	X	X
	Page 24-15		—				Page 24-15	
Analog Output Data Minimum/Maximum Values	X	X	—	—	—	—	X	X
	Page 24-15		—				Page 24-15	
Analog Output Data	X	X	—	—	—	—	X	X
	Page 24-15		—				Page 24-15	
Analog Output Operating Status	X	X	—	—	—	—	X	X
	Page 24-15		—				Page 24-15	

Data Register Allocation Numbers for Analog I/O Modules

Analog I/O modules are numbered from 1 through 7, in the order of increasing distance from the CPU module. Data registers are allocated to each analog I/O module depending on the analog I/O module number. END refresh type analog I/O modules and ladder refresh type analog I/O modules have different data register allocation.

END Refresh Type Analog I/O Modules

Each END refresh type analog I/O module is automatically allocated 20 data registers to store parameters for controlling analog I/O operation, starting with D760 through D779 for analog I/O module No. 1, up to D880 through D899 for analog I/O module No. 7. When a maximum of seven analog I/O modules are *not* used, data registers allocated to the unused analog I/O module numbers can be used as ordinary data registers.

When a maximum of seven END refresh type analog I/O modules are mounted, data registers D760 through D899 are allocated to analog modules 1 through 7 as shown below. The ANST macro is used to program data registers for the analog I/O module configuration. The CPU module checks the analog I/O configuration only once when the CPU starts to run. If you have changed the parameter while the CPU is running, stop and restart the CPU to enable the new parameter.

The END refresh type analog I/O module number starts with 1 next to the CPU module up to a maximum of 7.

The run-time program download and test program download cannot be used to change analog I/O parameters.

Channel	Function	END Refresh Type Analog I/O Module No.							R/W
		1	2	3	4	5	6	7	
Analog Input Ch 0	Analog input data	D760	D780	D800	D820	D840	D860	D880	R
	Analog input operating status	D761	D781	D801	D821	D841	D861	D881	R
	Analog input signal type	D762	D782	D802	D822	D842	D862	D882	R/W
	Analog input data type	D763	D783	D803	D823	D843	D863	D883	R/W
	Analog input data minimum value	D764	D784	D804	D824	D844	D864	D884	R/W
	Analog input data maximum value	D765	D785	D805	D825	D845	D865	D885	R/W
Analog Input Ch 1	Analog input data	D766	D786	D806	D826	D846	D866	D886	R
	Analog input operating status	D767	D787	D807	D827	D847	D867	D887	R
	Analog input signal type	D768	D788	D808	D828	D848	D868	D888	R/W
	Analog input data type	D769	D789	D809	D829	D849	D869	D889	R/W
	Analog input data minimum value	D770	D790	D810	D830	D850	D870	D890	R/W
	Analog input data maximum value	D771	D791	D811	D831	D851	D871	D891	R/W
Analog Output	Analog output data	D772	D792	D812	D832	D852	D872	D892	R/W
	Analog output operating status	D773	D793	D813	D833	D853	D873	D893	R
	Analog output signal type	D774	D794	D814	D834	D854	D874	D894	R/W
	Analog output data type	D775	D795	D815	D835	D855	D875	D895	R/W
	Analog output data minimum value	D776	D796	D816	D836	D856	D876	D896	R/W
	Analog output data maximum value	D777	D797	D817	D837	D857	D877	D897	R/W
– Reserved –		D778	D798	D818	D838	D858	D878	D898	R/W
		D779	D799	D819	D839	D859	D879	D899	R/W

Note: Data registers allocated to the unused analog I/O module numbers can be used as ordinary data registers.

Ladder Refresh Type Analog I/O Modules

When using a ladder refresh type analog input or output module, the first data register number can be designated in the ASNT macro dialog box. The quantity of required data registers depends on the model of the ladder refresh type analog input or output module.

Analog I/O Module	FC4A-J4CN1	FC4A-J8C1	FC4A-J8AT1	FC4A-K2C1
Quantity of Data Registers for Analog I/O Operation	65	65	65	15

Data register numbers and parameters are shown in the table below.

Ladder Refresh Type Analog Input Module Data Register Allocation (FC4A-J4CN1, FC4A-J8C1, and FC4A-J8AT1)

Data Register Number Offset	Data Size (word)	Parameter	Channel	Default	R/W
+0 (Low Byte)	1	Analog input signal type	CH0	FFh	R/W
+0 (High Byte)		— Reserved —	All channels	00h	
+1	4	Analog input data configuration	CH0	0	R/W
+5	1	Analog input signal type	CH1	00FFh	R/W
+6	4	Analog input data configuration		0	R/W
+10	1	Analog input signal type	CH2	00FFh	R/W
+11	4	Analog input data configuration		0	R/W
+15	1	Analog input signal type	CH3	00FFh	R/W
+16	4	Analog input data configuration		0	R/W
+20	1	Analog input signal type	CH4 *	00FFh	R/W
+21	4	Analog input data configuration		0	R/W
+25	1	Analog input signal type	CH5 *	00FFh	R/W
+26	4	Analog input data configuration		0	R/W
+30	1	Analog input signal type	CH6 *	00FFh	R/W
+31	4	Analog input data configuration		0	R/W
+35	1	Analog input signal type	CH7 *	00FFh	R/W
+36	4	Analog input data configuration		0	R/W
+40	3	Thermistor parameters (FC4A-J8AT1 only)	CH0 to CH3	0	R/W
+43	3		CH4 to CH7 *	0	R/W
+46	1	Analog input data	CH0	—	R
+47	1		CH1	—	R
+48	1		CH2	—	R
+49	1		CH3	—	R
+50	1		CH4 *	—	R
+51	1		CH5 *	—	R
+52	1		CH6 *	—	R
+53	1		CH7 *	—	R
+54	1	Analog input operating status	CH0	—	R
+55	1		CH1	—	R
+56	1		CH2	—	R
+57	1		CH3	—	R
+58	1		CH4 *	—	R
+59	1		CH5 *	—	R
+60	1		CH6 *	—	R
+61	1		CH7 *	—	R
+62	3	— Reserved —	All channels	—	R

* Data registers for channels 4 through 7 are reserved on the FC4A-J4CN1.

Ladder Refresh Type Analog Output Module Data Register Allocation (FC4A-K2C1)

Data Register Number Offset	Data Size (word)	Parameter	Channel	Default	R/W
+0 (Low Byte)	1	Analog output signal type	CH0	FFh	R/W
+0 (High Byte)		— Reserved —	All channels	00h	
+1	3	Analog output data configuration	CH0	0	R/W
+4	1	Analog output signal type	CH1	00FFh	R/W
+5	3	Analog output data configuration		0	R/W
+8	1	Analog output data	CH0	0	R/W
+9	1		CH1	0	R/W
+10	1	Analog output operating status	CH0	—	R
+11	1		CH1	—	R
+12	3	— Reserved —	All channels	—	R

Analog Input Parameters

Analog input parameters include the analog input signal type, analog input data type, analog input minimum and maximum values, filter value, thermistor parameter, analog input data, and analog input operating status. This section describes these parameters in detail.

Analog Input Signal Type

A total of 11 analog input signal types are available, depending on the analog I/O or analog input module. Select an analog input signal type for each analog input channel. When a channel is not used, select the default value or **Not used** for the channel.

Parameter		FC4A-L03A1	FC4A-L03AP1	FC4A-J2A1	FC4A-J4CN1	FC4A-J8C1	FC4A-J8AT1
0	Voltage input (0 to 10V DC)	X	—	X	X	X	—
1	Current input (4 to 20 mA DC)	X	—	X	X	X	—
2	Type K thermocouple	—	X	—	X	—	—
3	Type J thermocouple	—	X	—	X	—	—
4	Type T thermocouple	—	X	—	X	—	—
5	Pt 100 resistance thermometer	—	X	—	X	—	—
6	Pt 1000 resistance thermometer	—	—	—	X	—	—
7	Ni 100 resistance thermometer	—	—	—	X	—	—
8	Ni 1000 resistance thermometer	—	—	—	X	—	—
9	NTC type thermistor	—	—	—	—	—	X
10	PTC type thermistor	—	—	—	—	—	X
255	Not used	—	—	—	X	X	X

Analog Input Data Type

A total of five analog input data types are available, depending on the analog I/O or analog input module. Select an analog input data type for each analog input channel.

Parameter		FC4A-L03A1	FC4A-L03AP1	FC4A-J2A1	FC4A-J4CN1	FC4A-J8C1	FC4A-J8AT1
0	Binary data	X	X	X	X	X	X
1	Optional range	X	X	X	X	X	X
2	Celsius	—	X	—	X	—	NTC only
3	Fahrenheit	—	X	—	X	—	NTC only
4	Resistance	—	—	—	—	—	X

Binary Data

When Binary data is selected as an analog input data type, the analog input is linearly converted into digital data in the range described in the table below.

Type No.	FC4A-L03A1 FC4A-L03AP1 FC4A-J2A1	FC4A-J4CN1		FC4A-J8C1	FC4A-J8AT1
		Analog Input Signal Type	Analog Input Data		
Analog Input Data	0 to 4095	Voltage/Current: Thermocouple: Pt100, Ni100: Pt1000, Ni1000:	0 to 50,000 0 to 50,000 0 to 6,000 0 to 60,000	0 to 50000	0 to 4000

Optional Range

When Optional range is selected as an analog input data type, the analog input is linearly converted into digital data in the range between the minimum and maximum values designated in the Configure Parameters dialog box.

Type No.	FC4A-L03A1	FC4A-L03AP1	FC4A-J2A1	FC4A-J4CN1	FC4A-J8C1	FC4A-J8AT1
Analog Input Data	Analog input data minimum value to maximum value (–32768 to 32767)					

Celsius and Fahrenheit

When Celsius or Fahrenheit is selected as an analog input data type, the analog input data range depends on the analog input signal type, scale value, and the type of the analog input module, FC4A-L03AP1, FC4A-J4CN1, and FC4A-J8AT1.

• FC4A-L03AP1

Analog Input Signal Type	Celsius		Fahrenheit	
	Temperature (°C)	Analog Input Data	Temperature (°F)	Analog Input Data
Type K thermocouple	0 to 1300	0 to 13000	32 to 2372	320 to 23720
Type J thermocouple	0 to 1200	0 to 12000	32 to 2192	320 to 21920
Type T thermocouple	0 to 400	0 to 4000	32 to 752	320 to 7520
Pt100 resistance thermometer	–100.0 to 500.0	–1000 to 5000	–148.0 to 932.0	–1480 to 9320

• FC4A-J4CN1

Analog Input Signal Type	Scale	Celsius		Fahrenheit	
		Temperature (°C)	Analog Input Data	Temperature (°F)	Analog Input Data
Type K thermocouple	×1	0 to 1300	0 to 1300	32 to 2372	32 to 2372
	×10	0.0 to 1300.0	0 to 13000	32.0 to 2372.0	320 to 23720
Type J thermocouple	×1	0 to 1200	0 to 1200	32 to 2192	32 to 2192
	×10	0.0 to 1200.0	0 to 12000	32.0 to 2192.0	320 to 21920
Type T thermocouple	×1	0 to 400	0 to 400	32 to 752	32 to 752
	×10	0.0 to 400.0	0 to 4000	32.0 to 752.0	320 to 7520
Pt100, Pt1000 resistance thermometer	×1	–100 to 500	–100 to 500	–148 to 932	–148 to 932
	×10	–100.0 to 500.0	–1000 to 5000	–148.0 to 932.0	–1480 to 9320
	×100	0.00 to 500.00 –100.00 to 327.67	0 to 50000 –10000 to 32767	0.00 to 655.35 –148.00 to 327.67	0 to 65535 –14800 to 32767
Ni100, Ni1000 resistance thermometer	×1	–60 to 180	–60 to 180	–76 to 356	–76 to 356
	×10	–60.0 to 180.0	–600 to 1800	–76.0 to 356.0	–760 to 3560
	×100	–60.00 to 180.00	–6000 to 18000	0.00 to 356.00 –76.00 to 327.67	0 to 35600 –7600 to 32767

• FC4A-J8AT1

Analog Input Signal Type	Scale	Celsius		Fahrenheit	
		Temperature (°C)	Analog Input Data	Temperature (°F)	Analog Input Data
NTC thermistor	×1	–50 to 150	–50 to 150	–58 to 302	–58 to 302
	×10	–50.0 to 150.0	–500 to 1500	–58.0 to 302.0	–580 to 3020

Resistance

When Resistance is selected as an analog input data type, the analog input is linearly converted into digital data in the range described in the table below. This option is available only when NTC or PTC type thermistor is selected for the FC4A-J8AT1.

• FC4A-J8AT1

Analog Input Signal Type	Resistance	
	Resistance (Ω)	Analog Input Data
NTC/PTC thermistor	0 to 100000	0 to 10000

Analog Input Minimum/Maximum Values

For analog input values, when Optional range is selected for the Data Type, designate the analog input data minimum and maximum values which can be –32,768 through 32,767.

In addition, when using resistance thermometers (Pt100, Pt1000, Ni100, or Ni1000) with the Celsius or Fahrenheit Data Type and the ×100 scale, select the analog input data minimum value from 0 or another value in the pull-down list. The maximum value is changed automatically according to the selected minimum value.

Filter Value

The filter function is available for the ladder input type FC4A-J4CN1, FC4A-J8C1, and FC4A-J8AT1 only. Filtering ensures smooth input of analog data into the CPU module. For the filtering function of analog input signals, see page 24-4.

Valid values are 0 through 255.

Thermistor Parameter

Thermistor parameters are enabled when selecting NTC thermistor for the analog input type of the FC4A-J8AT1. The same parameters are specified for four channels: CH0 to CH3 and CH4 to CH7.

Channel	NTC Thermistor Parameters (Values indicated on the thermistor)		Valid Range
CH0 to CH3 CH4 to CH7	R0:	Thermistor resistance value at the temperature (°C)	0 to 65535
	T0:	Temperature (°C)	–32768 to 32767
	B:	Thermistor B parameter (Kelvin)	0 to 65535

For NTC type thermistors, analog input data can be calculated from the following formula:

$$\text{Analog Input Data} = \frac{B \times T0}{B + T0 \times \log(r/R0)}$$

where, r = thermistor resistance (Ω)

For PTC type thermistors, linearize the analog input data using the XYFS instruction.

Analog Input Data

The analog input signal is converted into a digital value within the range specified by the analog input data type and applicable parameters, and is stored to a data register allocated to analog input data. The analog input data register number is shown under Data in the Configure Parameters dialog box.

END Refresh Type

The analog input signal is converted into a digital value and stored to a data register, such as D760 or D766, allocated to analog input channel 1 or 2 on analog module number 1 through 7 depending on the mounting position.

The analog input data stored in the allocated data register is updated whether the CPU module is running or stopped. When the CPU module is running, the update occurs at the END processing of every scan or 10 ms, whichever is longer. When the CPU module is stopped, the update occurs every 10 ms.

Ladder Refresh Type

The analog input signal is converted into a digital value and stored to a data register determined by the data register number selected in the Configure Parameters dialog box of the ANST macro. The analog input data stored in the allocated data register is updated when the RUNA instruction contained in the ANST macro is executed.

When a certain channel of a ladder refresh type analog input module is not used, data registers allocated to the unused channel will store indefinite values if the values are read out of the analog input module. Do not use the allocated data registers for other purposes.

Only when the analog input status code is 0, the analog input data is assured. Make sure that a user program reads analog input data only when the analog input status code is 0.

Analog Input Operating Status

The operating status of each analog input channel is stored to a data register allocated to analog input operating status. While the analog input is operating normally, the data register stores 0. The analog input operating status data register number is shown under Status in the Configure Parameters dialog box.

END Refresh Type

The operating status of each analog input channel is stored to a data register, such as D761 or D767, allocated to analog input channel 1 or 2 on analog module number 1 through 7 depending on the mounting position.

The analog input operating status data is updated whether the CPU module is running or stopped. When the CPU module is running, the update occurs at the END processing of every scan or 10 ms, whichever is longer. When the CPU module is stopped, the update occurs every 10 ms.

Status Code	Analog Input Operating Status (END refresh type)
0	Normal operation
1	Converting data (during the first data conversion after power-up)
2	Initializing
3	Invalid parameter or analog input channel not available on the installed analog module
4	Hardware failure (external power supply failure)
5	Incorrect wiring (input data over valid range)
6	Incorrect wiring (input data below valid range or current loop open)

Ladder Refresh Type

The operating status of each analog input channel is stored to a data register determined by the data register number selected in the Configure Parameters dialog box of the ANST macro.

Operating Status Bit		Analog Input Operating Status (Ladder refresh type)	
Bit 0	0	Operating status bit	Normal operation
	1		Initializing, changing configuration, hardware initialization error
Bit 1	0	Parameter bit	Parameter configuration normal
	1		Parameter configuration error
Bit 2	0	External power supply bit	External power supply normal
	1		External power supply error
Bit 3	0	Maximum value over bit	Within the maximum value
	1		Maximum value over error
Bit 4	0	Minimum value over bit	Within the minimum value
	1		Minimum value under error
Bit 5 to Bit 15	0	Reserved	Normal operation

Analog Output Parameters

Analog output parameters include the analog output signal type, analog output data type, analog output minimum and maximum values, analog output data, and analog output operating status. This section describes these parameters in detail.

Analog Output Signal Type

A total of three analog output signal types are available, depending on the analog I/O or analog output module. Select an analog output signal type for each analog output channel. When a channel is not used, select the default value or **Not used** for the channel.

Parameter		FC4A-L03A1	FC4A-L03AP1	FC4A-K1A1	FC4A-K2C1
0	Voltage output	0 to 10V DC			–10 to +10V DC
1	Current output	4 to 20 mA DC			
255	Not used	—	—	—	X

Analog Output Data Type

A total of two analog output data types are available, depending on the analog I/O or analog output module. Select an analog output data type for each analog output channel.

Parameter			FC4A-L03A1	FC4A-L03AP1	FC4A-K1A1	FC4A-K2C1
0	Binary data	Voltage	0 to 4095			–25000 to 25000
		Current				0 to 50000
1	Optional range	Voltage	Analog output data minimum value to maximum value (–32768 to 32767)			
		Current				

Analog Output Minimum/Maximum Values

For analog output values, when Optional range is selected for the Data Type, designate the analog output data minimum and maximum values which can be –32,768 through 32,767.

Analog Output Data

The analog output data is converted into an analog output signal within the range specified by the analog output data type and applicable parameters. The analog output data register number is shown under Data in the Configure Parameters dialog box.

END Refresh Type

The analog output data stored in a data register, such as D772, is converted into an analog output signal of voltage output (0 to 10V DC) or current output (4 to 20 mA) as designated by the value stored in the data register allocated to analog output signal type, such as D774.

While the CPU module is running, the analog output data stored in the allocated data register is updated at the END processing of every scan or 10 ms, whichever is longer. While the CPU module is stopped, the analog output data remains at 0 or the designated analog output data minimum value, so the generated analog output signal remains at the minimum value of 0V DC or 4 mA DC.

Ladder Refresh Type

While the CPU module is running, the analog output data stored in the allocated data register is updated when the RUNA instruction contained in the ANST macro is executed. While the CPU module is stopped, the analog output data is not updated. But the analog output signal can be changed by using the STPA instruction. For details, see page 24-21.

Analog Output Operating Status

The operating status of each analog output channel is stored to a data register allocated to analog output operating status. While the analog output is operating normally, the data register stores 0. The analog output operating status data register number is shown under Status in the Configure Parameters dialog box.

END Refresh Type

The operating status of each analog output is stored to a data register, such as D773. While the analog output is operating normally, the data register stores 0. The analog output operating status data is updated whether the CPU module is running or stopped. The update occurs at the END processing of every scan or 10 ms, whichever is longer.

Status Code	Analog Output Operating Status (END refresh type)
0	Normal operation
1	(reserved)
2	Initializing
3	Invalid parameter or analog output channel not available on the installed analog module
4	Hardware failure (external power supply failure)

Ladder Refresh Type

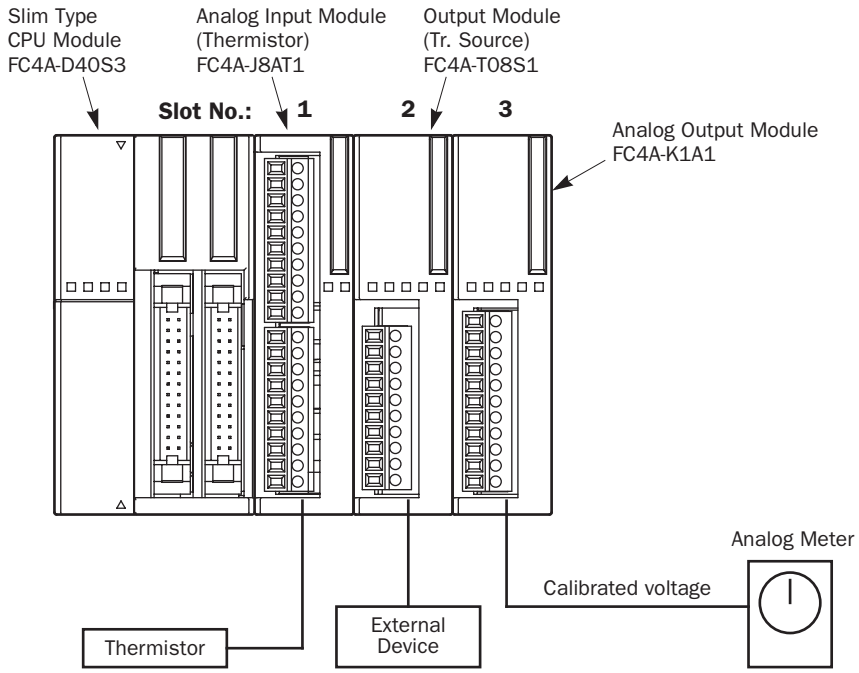
The operating status of each analog output channel is stored to a data register determined by the data register number selected in the Configure Parameters dialog box of the ANST macro.

Operating Status Bit		Analog Output Operating Status (Ladder refresh type)	
Bit 0	0	Operating status bit	Normal operation
	1		Initializing, changing configuration, hardware initialization error
Bit 1	0	Parameter bit	Parameter configuration normal
	1		Parameter configuration error
Bit 2	0	External power supply bit	External power supply normal
	1		External power supply error
Bit 3	0	Output data error bit	Output data normal
	1		Output data range error
Bit 4 to Bit 15	0	Reserved	Normal operation

Example: Analog I/O

The following example demonstrates a program of analog I/O control using an NTC thermistor. Two analog I/O modules are mounted in the slots shown below.

System Setup

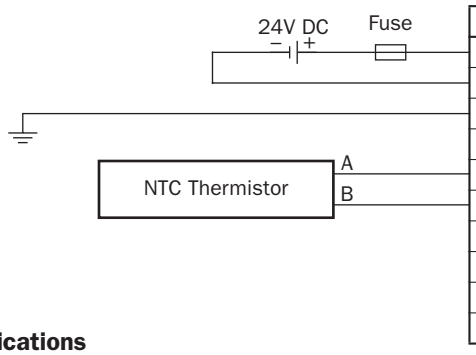


Operation

In this example, the input value from the NTC thermistor is calibrated. When the temperature reaches the preset value, the output is turned off. The thermistor temperature is monitored on an analog meter.

Wiring Diagram

FC4A-J8AT1 (Analog Input Module)

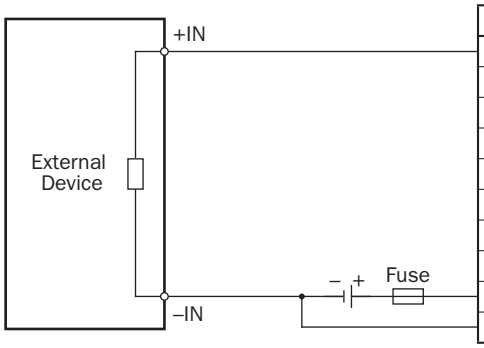


Terminal No.	Channel
24V	24V DC
0V	
⏚	
NC	—
A	IN0
B	
A	IN1
B	
A	IN2
B	
A	IN3
B	
A	IN4
B	
A	IN5
B	
A	IN6
B	
A	IN7
B	

• Thermistor Specifications

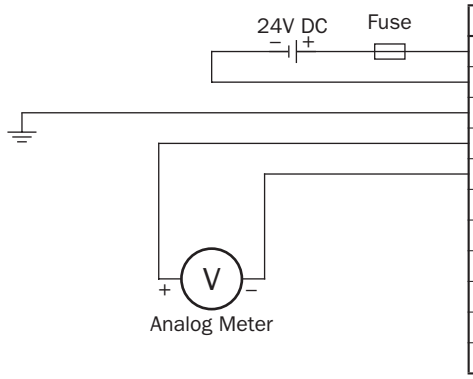
Type No.	NT731ATTD103K38J (KOA)
Type	NTC
RO	10,000Ω
T0	25°C
B Parameter	3,800K

FC4A-T08S1 (8-point Transistor Source Output Module)



Terminal No.	Output
0	Q0
1	Q1
2	Q2
3	Q3
4	Q4
5	Q5
6	Q6
7	Q7
COM(+)	COM(+)
–V	–V

FC4A-K1A1 (Analog Output Module)

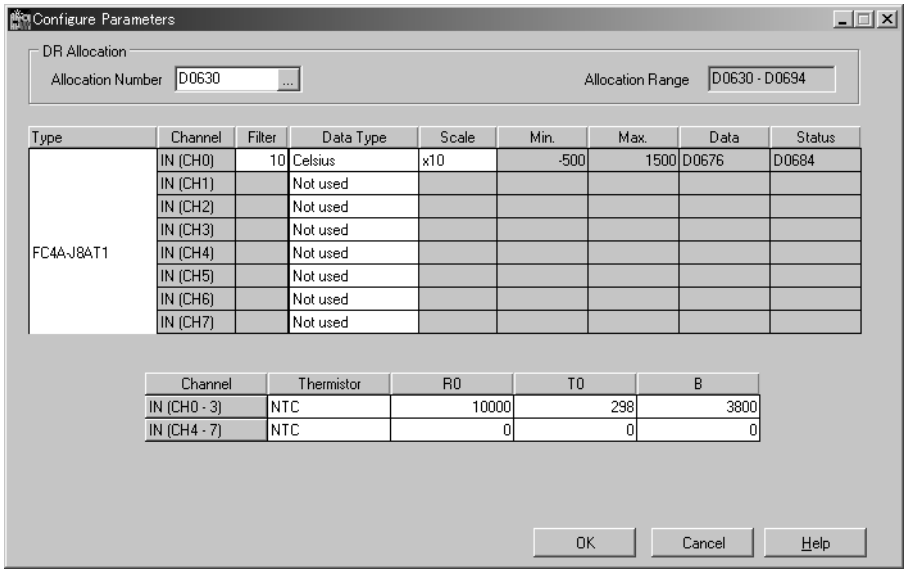


Terminal No.	Channel
+	24V DC
–	
⏚	
+	OUT
–	
NC	—
NC	
NC	
NC	—
NC	
NC	

WindLDR Programming

Analog I/O modules are programmed using the ANST macro in WindLDR. Program the ANST macro as shown below.

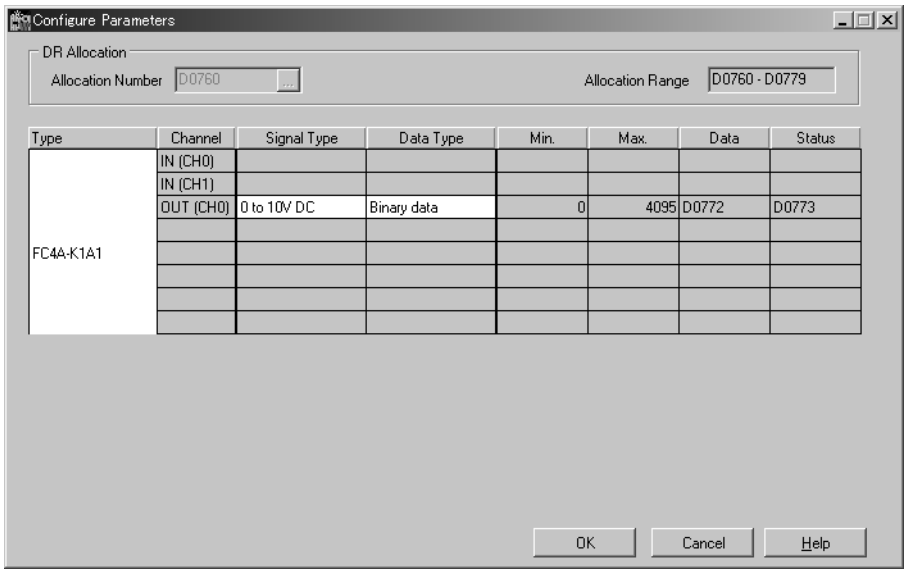
• Analog Input Module FC4A-J8AT1 on Slot 1



DR Allocation Range			Designation	Description
D630 - D694			D630	Optional range allocation, 65 words
I/O	**Channel**	**Item**	**Designation**	**Description**
IN	CH0	Filter	10	Averages input values
		Data Type	Celsius	Analog input range –50 to 150°C
		Scale	×10	Analog input data –500 to 1500
	CH1	Data Type	Not used	Unused channel
	CH2	Data Type	Not used	Unused channel
	CH3	Data Type	Not used	Unused channel
	CH4	Data Type	Not used	Unused channel
	CH5	Data Type	Not used	Unused channel
	CH6	Data Type	Not used	Unused channel
	CH7	Data Type	Not used	Unused channel
	CH0 - CH3	Thermistor Type	NTC	NTC thermistor
		R0	10,000	Resistance value at the absolute temperature = 10 kΩ
		T0	25	Temperature = 25°C
		B	3,800	B parameter = 3,800K

Note: When CH4 through CH7 are not used, thermistor settings are not required.

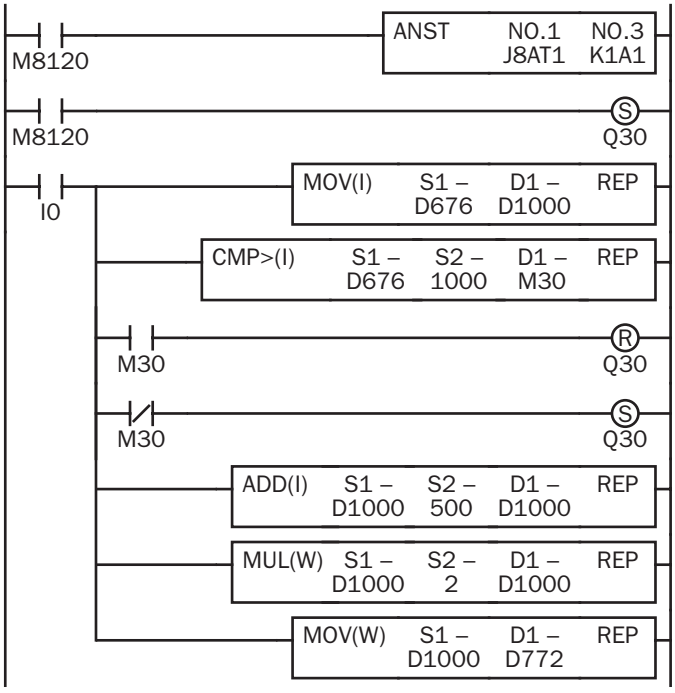
- **Analog Output Module FC4A-K1A1 on Slot 3**



DR Allocation Range			Designation	Description
D760 - D779			—	Automatic range allocation, 20 words
I/O	**Channel**	**Item**	**Designation**	**Description**
OUT	CH0	Signal Type	0 to 10V DC	Voltage output
		Data Type	Binary data	0 to 4095

Ladder Diagram

As shown in the ladder diagram below, when initialize pulse special internal relay M8120 is used for the ANST macro in parallel with another instruction, load M8120 again for the other instruction.



M8120 is the initialize pulse special internal relay.

When the CPU starts to run, ANST stores parameters to data registers to configure analog I/O modules and Q30 is turned on.

When I0 is turned on, analog input data is moved from D676 to D1000.

The temperature is compared with the alarm temperature of 100°C.

When the temperature is higher than 100°C, Q30 is turned off.

When the temperature is not higher than 100°C, Q30 is turned on.

Analog input data of –500 to +1500 is converted to 0 to 2000.

Analog input data of 0 to 2000 is converted to 0 to 4000.

Analog input data of 0 to 4000 is moved to D772 (analog output data) of the analog output module.

Note: The above ladder diagram is only an example and should be modified as required.

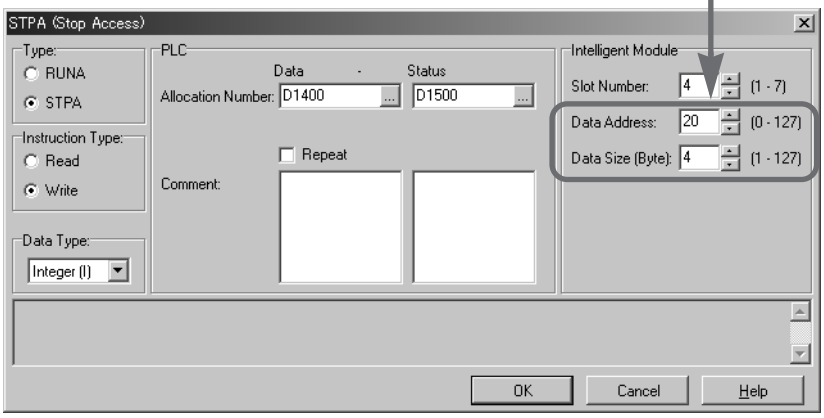
Changing Analog Output While CPU is Stopped

When using the FC4A-K2C1 analog output module, the analog output value can be changed while the CPU module is stopped. To change the analog output value, store a required output value to the memory addresses allocated to the analog output data.

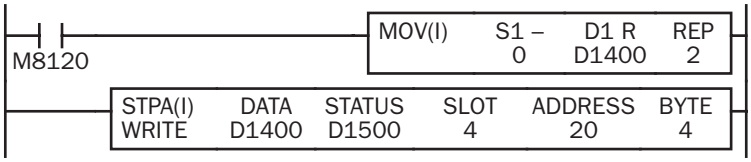
Example: Memory Allocation of Ladder Refresh Type Analog Output Module FC4A-K2C1

Memory Address (data address used for STPA)	Data Size (bytes)	R/W	Parameter	
+20	2	R/W	Analog Output Data	CH0
+22	2	R/W		CH1

STPA instruction when FC4A-K2C1 is mounted on slot 4



Ladder Diagram



M8120 is the initialize pulse special internal relay.

MOV stores output values at the OFF state.

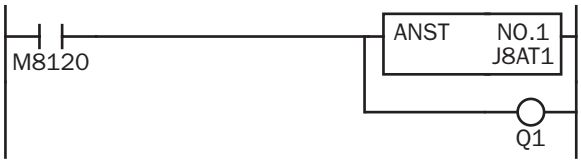
When the CPU stops, STPA updates the analog output value of the analog output module.

Note: The above ladder diagram is only an example and should be modified as required.

Precautions for Programming ANST Macro

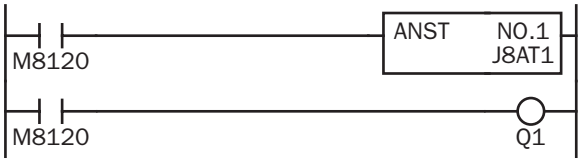
When using the ANST macro, do not make a branch from the ladder line of the ANST macro.

Incorrect



Delete the branch from the ANST macro, and start another line by inserting a LOD instruction.

Correct



25: DATA LINK COMMUNICATION

Introduction

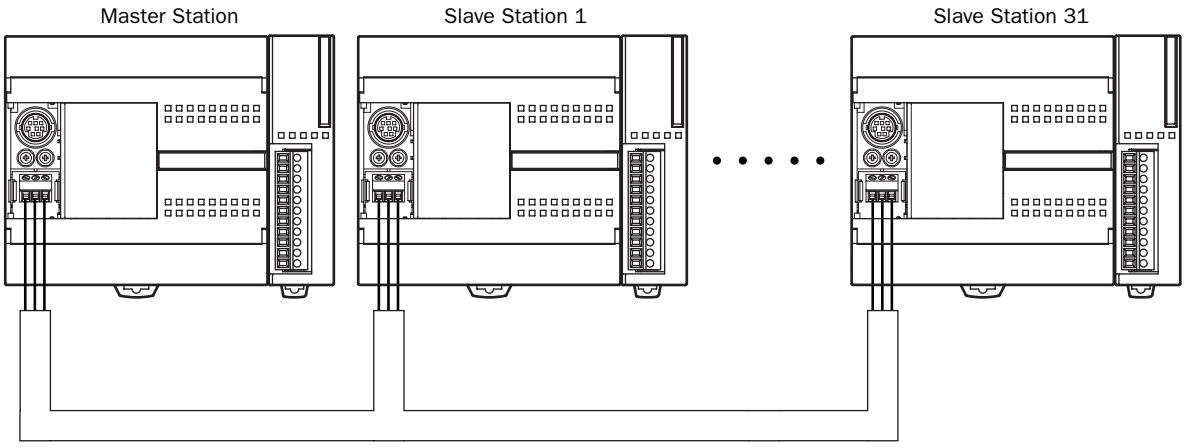
This chapter describes the data link communication function used to set up a distributed control system.

A data link communication system consists of one master station and a maximum of 31 slave stations, each station comprising a 16- or 24-I/O type MicroSmart CPU module or any slim type CPU module. When the data link communication is enabled, the master station has 12 data registers assigned for each slave station, and each slave station has 12 data registers for communication with the master station. Using these data registers, the master station can send and receive data of 6 data registers to and from each slave station. No particular program is required for sending or receiving data in the data link communication system.

When data of inputs, outputs, internal relays, timers, counters, or shift registers are moved to data registers using the move instructions in the user program, these data can also be exchanged between the master and slave stations.

The OpenNet Controller, MICRO^3, MICRO^{3}C, and FA-3S series PLCs can also be connected to the data link communication system.

The all-in-one 10-I/O type MicroSmart CPU module does not have data link communication capabilities.



Data Link Specifications

Electric Specifications	Compliance with EIA-RS485
Baud Rate	19,200 or 38,400 bps
Synchronization	Start-stop synchronization Start bit: 1 Data bits: 7 Parity: Even Stop bit: 1
Communication Cable	Shielded twisted pair cable, core wire 0.3 mm^2
Maximum Cable Length	200m (656 feet) total
Maximum Slave Stations	31 slave stations
Refresh Mode	Separate refresh
Transmit/Receive Data	0 through 6 words each for transmission and receiving per slave station
Special Internal Relay	M8005-M8007: communication control and error M8080-M8116: communication completion for each slave station M8117: communication completion for all slave stations
Data Register	D900-D1271: transmit/receive data
Special Data Register	D8069-D8099: communication error code

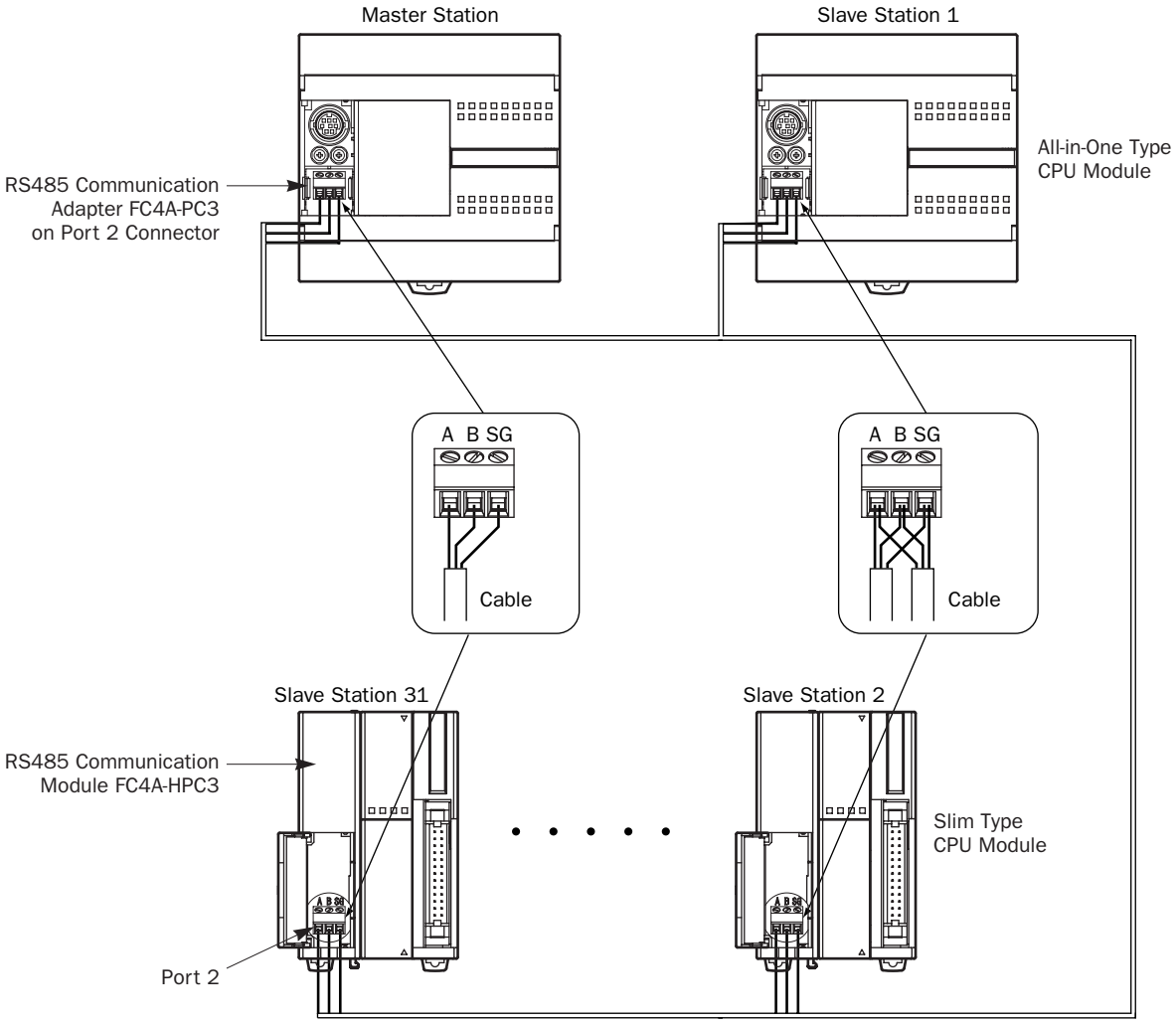
Data Link System Setup

To set up a data link system, install the RS485 communication adapter (FC4A-PC3) to the port 2 connector on the all-in-one 16- or 24-I/O type CPU module.

When using the slim type CPU module, mount the RS485 communication module (FC4A-HPC3) next to the CPU module.

When using the optional HMI module with the slim type CPU module (not shown below), install the RS485 communication adapter (FC4A-PC3) to the port 2 connector on the HMI base module.

Connect the RS485 terminals A, B, and SG on every CPU module using a shielded twisted pair cable as shown below. The total length of the cable for the data link system can be extended up to 200 meters (656 feet).



Shielded twisted pair cable 200 meters (656 feet) maximum
Core wire 0.3mm^2

Data Register Allocation for Transmit/Receive Data

The master station has 12 data registers assigned for data communication with each slave station. Each slave station has 12 data registers assigned for data communication with the master station. When data is set in data registers at the master station assigned for data link communication, the data is sent to the corresponding data registers at a slave station. When data is set in data registers at a slave station assigned for data link communication, the data is sent to the corresponding data registers at the master station.

Master Station

Slave Station Number	Data Register	Transmit/Receive Data	Slave Station Number	Data Register	Transmit/Receive Data
Slave 1	D900-D905	Transmit data to slave 1	Slave 17	D1092-D1097	Transmit data to slave 17
	D906-D911	Receive data from slave 1		D1098-D1103	Receive data from slave 17
Slave 2	D912-D917	Transmit data to slave 2	Slave 18	D1104-D1109	Transmit data to slave 18
	D918-D923	Receive data from slave 2		D1110-D1115	Receive data from slave 18
Slave 3	D924-D929	Transmit data to slave 3	Slave 19	D1116-D1121	Transmit data to slave 19
	D930-D935	Receive data from slave 3		D1122-D1127	Receive data from slave 19
Slave 4	D936-D941	Transmit data to slave 4	Slave 20	D1128-D1133	Transmit data to slave 20
	D942-D947	Receive data from slave 4		D1134-D1139	Receive data from slave 20
Slave 5	D948-D953	Transmit data to slave 5	Slave 21	D1140-D1145	Transmit data to slave 21
	D954-D959	Receive data from slave 5		D1146-D1151	Receive data from slave 21
Slave 6	D960-D965	Transmit data to slave 6	Slave 22	D1152-D1157	Transmit data to slave 22
	D966-D971	Receive data from slave 6		D1158-D1163	Receive data from slave 22
Slave 7	D972-D977	Transmit data to slave 7	Slave 23	D1164-D1169	Transmit data to slave 23
	D978-D983	Receive data from slave 7		D1170-D1175	Receive data from slave 23
Slave 8	D984-D989	Transmit data to slave 8	Slave 24	D1176-D1181	Transmit data to slave 24
	D990-D995	Receive data from slave 8		D1182-D1187	Receive data from slave 24
Slave 9	D996-D1001	Transmit data to slave 9	Slave 25	D1188-D1193	Transmit data to slave 25
	D1002-D1007	Receive data from slave 9		D1194-D1199	Receive data from slave 25
Slave 10	D1008-D1013	Transmit data to slave 10	Slave 26	D1200-D1205	Transmit data to slave 26
	D1014-D1019	Receive data from slave 10		D1206-D1211	Receive data from slave 26
Slave 11	D1020-D1025	Transmit data to slave 11	Slave 27	D1212-D1217	Transmit data to slave 27
	D1026-D1031	Receive data from slave 11		D1218-D1223	Receive data from slave 27
Slave 12	D1032-D1037	Transmit data to slave 12	Slave 28	D1224-D1229	Transmit data to slave 28
	D1038-D1043	Receive data from slave 12		D1230-D1235	Receive data from slave 28
Slave 13	D1044-D1049	Transmit data to slave 13	Slave 29	D1236-D1241	Transmit data to slave 29
	D1050-D1055	Receive data from slave 13		D1242-D1247	Receive data from slave 29
Slave 14	D1056-D1061	Transmit data to slave 14	Slave 30	D1248-D1253	Transmit data to slave 30
	D1062-D1067	Receive data from slave 14		D1254-D1259	Receive data from slave 30
Slave 15	D1068-D1073	Transmit data to slave 15	Slave 31	D1260-D1265	Transmit data to slave 31
	D1074-D1079	Receive data from slave 15		D1266-D1271	Receive data from slave 31
Slave 16	D1080-D1085	Transmit data to slave 16	—		
	D1086-D1091	Receive data from slave 16			

If any slave stations are not connected, master station data registers which are assigned to the vacant slave stations can be used as ordinary data registers.

Slave Station

Data	Data Register	Transmit/Receive Data
Slave Station Data	D900-D905	Transmit data to master station
	D906-D911	Receive data from master station

Slave station data registers D912 through D1271 can be used as ordinary data registers.

Special Data Registers for Data Link Communication Error

In addition to data registers assigned for data communication, the master station has 31 special data registers and each slave station has one special data register to store data link communication error codes. If any communication error occurs in the data link system, communication error codes are set to a corresponding data register for link communication error at the master station and to data register D8069 at the slave station. For details of link communication error codes, see below.

If a communication error occurs in the data link communication system, the data is resent two times. If the error still exists after three attempts, then the error code is set to the data registers for data link communication error. Since the error code is not communicated between the master and slave stations, error codes must be cleared individually.

Master Station

Special Data Register	Data Link Communication Error Data	Special Data Register	Data Link Communication Error Data
D8069	Slave station 1 communication error	D8085	Slave station 17 communication error
D8070	Slave station 2 communication error	D8086	Slave station 18 communication error
D8071	Slave station 3 communication error	D8087	Slave station 19 communication error
D8072	Slave station 4 communication error	D8088	Slave station 20 communication error
D8073	Slave station 5 communication error	D8089	Slave station 21 communication error
D8074	Slave station 6 communication error	D8090	Slave station 22 communication error
D8075	Slave station 7 communication error	D8091	Slave station 23 communication error
D8076	Slave station 8 communication error	D8092	Slave station 24 communication error
D8077	Slave station 9 communication error	D8093	Slave station 25 communication error
D8078	Slave station 10 communication error	D8094	Slave station 26 communication error
D8079	Slave station 11 communication error	D8095	Slave station 27 communication error
D8080	Slave station 12 communication error	D8096	Slave station 28 communication error
D8081	Slave station 13 communication error	D8097	Slave station 29 communication error
D8082	Slave station 14 communication error	D8098	Slave station 30 communication error
D8083	Slave station 15 communication error	D8099	Slave station 31 communication error
D8084	Slave station 16 communication error	—	—

If any slave stations are not connected, master station data registers which are assigned to the vacant slave stations can be used as ordinary data registers.

Slave Station

Special Data Register	Data Link Communication Error Data
D8069	Slave station communication error

Note: Slave station data registers D8070 through D8099 can be used as ordinary data registers.

Data Link Communication Error Code

The data link error code is stored in the special data register allocated to indicate a communication error in the data link system. When this error occurs, special internal relay M8005 (data link communication error) is also turned on at both master and slave stations. The detailed information of general errors can be viewed using WindLDR. Select **Online > Monitor**, then select **Online > PLC Status > Error Status: Details**. See page 29-2.

Error Code	Error Details
1h (1)	Overrun error (data is received when the receive data registers are full)
2h (2)	Framing error (failure to detect start or stop bit)
4h (4)	Parity error (an error was found by the parity check)
8h (8)	Receive timeout (line disconnection)
10h (16)	BCC (block check character) error (disparity with data received up to BCC)
20h (32)	Retry cycle over (error occurred in all 3 trials of communication)
40h (64)	I/O definition quantity error (discrepancy of transmit/receive station number or data quantity)

When more than one error is detected in the data link system, the total of error codes is indicated. For example, when framing error (error code 2h) and BCC error (error code 10h) are found, error code 12h (18) is stored.

Data Link Communication between Master and Slave Stations

The master station has 6 data registers assigned to transmit data to a slave station and 6 data registers assigned to receive data from a slave station. The quantity of data registers for data link can be selected from 0 through 6 using WindLDR. The following examples illustrate how data is exchanged between the master and slave stations when 2 or 6 data registers are used for data link communication with each of 31 slave stations.

Example 1: Transmit Data 2 Words and Receive Data 2 Words

Master Station

D8069	Communication Error
D900 - D901	Transmit Data
D906 - D907	Receive Data
D8070	Communication Error
D912 - D913	Transmit Data
D918 - D919	Receive Data
D8071	Communication Error
D924 - D925	Transmit Data
D930 - D931	Receive Data
D8072	Communication Error
D936 - D937	Transmit Data
D942 - D943	Receive Data
⋮	⋮
D8098	Communication Error
D1248 - D1249	Transmit Data
D1254 - D1255	Receive Data
D8099	Communication Error
D1260 - D1261	Transmit Data
D1266 - D1267	Receive Data

Slave Stations

D8069	Communication Error	Slave Station 1
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 2
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 3
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 4
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
⋮	⋮	⋮
D8069	Communication Error	Slave Station 30
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 31
D900 - D901	Transmit Data	
D906 - D907	Receive Data	

Example 2: Transmit Data 6 Words and Receive Data 6 Words

Master Station

D8069	Communication Error
D900 - D905	Transmit Data
D906 - D911	Receive Data
D8070	Communication Error
D912 - D917	Transmit Data
D918 - D923	Receive Data
D8071	Communication Error
D924 - D929	Transmit Data
D930 - D935	Receive Data
D8072	Communication Error
D936 - D941	Transmit Data
D942 - D947	Receive Data
⋮	⋮
D8098	Communication Error
D1248 - D1253	Transmit Data
D1254 - D1259	Receive Data
D8099	Communication Error
D1260 - D1265	Transmit Data
D1266 - D1271	Receive Data

Slave Stations

D8069	Communication Error	Slave Station 1
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 2
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 3
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 4
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
⋮	⋮	⋮
D8069	Communication Error	Slave Station 30
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 31
D900 - D905	Transmit Data	
D906 - D911	Receive Data	

Special Internal Relays for Data Link Communication

Special internal relays M8005 through M8007 and M8080 through M8117 are assigned for the data link communication.

M8005 Data Link Communication Error

When an error occurs during communication in the data link system, M8005 turns on. The M8005 status is maintained when the error is cleared and remains on until M8005 is reset using WindLDR or until the CPU is turned off. The cause of the data link communication error can be checked using **Online > Monitor**, followed by **Online > PLC Status > Error Status: Details**. See page 25-4.

M8006 Data Link Communication Prohibit Flag (Master Station)

When M8006 at the master station is turned on in the data link system, data link communication is stopped. When M8006 is turned off, data link communication resumes. The M8006 status is maintained when the CPU is turned off and remains on until M8006 is reset using WindLDR.

When M8006 is on at the master station, M8007 is turned on at slave stations in the data link system.

M8007 Data Link Communication Initialize Flag (Master Station) Data Link Communication Stop Flag (Slave Station)

M8007 has a different function at the master or slave station of the data link communication system.

Master station: Data link communication initialize flag

When M8007 at the master station is turned on during operation, the link configuration is checked to initialize the data link system. When a slave station is powered up after the master station, turn M8007 on to initialize the data link system. After a data link system setup is changed, M8007 must also be turned on to ensure correct communication.

Slave station: Data link communication stop flag

When a slave station does not receive communication data from the master station for 10 seconds or more in the data link system, M8007 turns on. When a slave station does not receive data in 10 seconds after initializing the data link system, M8007 also turns on at the slave station. When the slave station receives correct communication data, M8007 turns off.

M8080-M8116 Slave Station Communication Completion Relay (Master Station)

Special internal relays M8080 through M8116 are used to indicate the completion of data refresh. When data link communication with a slave station is complete, a special internal relay assigned for the slave station is turned on for one scan time at the master station.

Special Internal Relay	Slave Station Number	Special Internal Relay	Slave Station Number	Special Internal Relay	Slave Station Number
M8080	Slave Station 1	**M8092**	Slave Station 11	**M8104**	Slave Station 21
M8081	Slave Station 2	**M8093**	Slave Station 12	**M8105**	Slave Station 22
M8082	Slave Station 3	**M8094**	Slave Station 13	**M8106**	Slave Station 23
M8083	Slave Station 4	**M8095**	Slave Station 14	**M8107**	Slave Station 24
M8084	Slave Station 5	**M8096**	Slave Station 15	**M8110**	Slave Station 25
M8085	Slave Station 6	**M8097**	Slave Station 16	**M8111**	Slave Station 26
M8086	Slave Station 7	**M8100**	Slave Station 17	**M8112**	Slave Station 27
M8087	Slave Station 8	**M8101**	Slave Station 18	**M8113**	Slave Station 28
M8090	Slave Station 9	**M8102**	Slave Station 19	**M8114**	Slave Station 29
M8091	Slave Station 10	**M8103**	Slave Station 20	**M8115**	Slave Station 30
—	—	**—**	—	**M8116**	Slave Station 31

M8080 Communication Completion Relay (Slave Station)

When data link communication with a master station is complete, special internal relay M8080 at the slave station is turned on for one scan time.

M8117 All Slave Station Communication Completion Relay

When data link communication with all slave stations is complete, special internal relay M8117 at the master station is turned on for one scan time. M8117 at slave stations does not go on.

Programming WindLDR

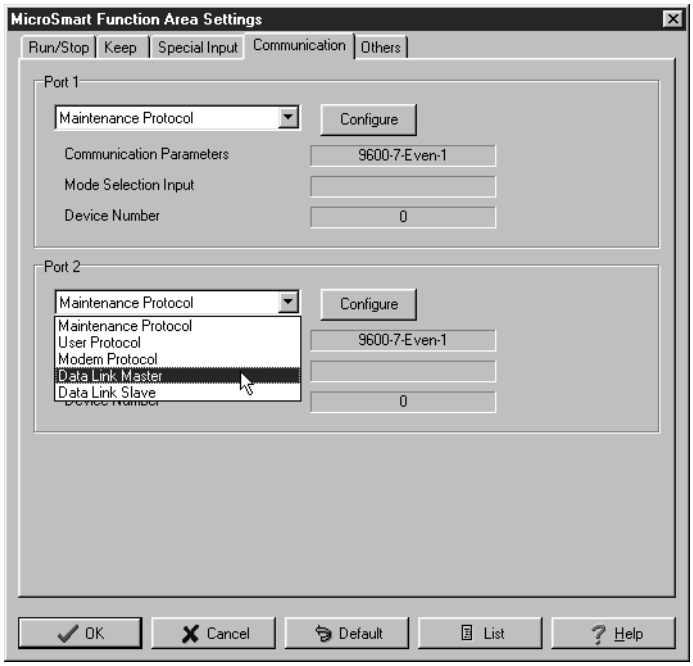
The Communication page in the Function Area Settings is used to program for the data link master and slave stations.

Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

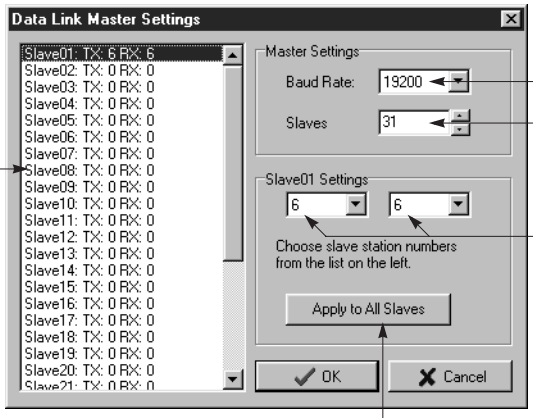
Data Link Master Station

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Setting dialog box appears.

2. Click the **Communication** tab, and select **Data Link Master** in the Port 2 pull-down list.



3. The Data Link Master Settings dialog box appears. Select a baud rate and the quantity of slave stations. Select a slave station number from the list on the left and make settings as shown below.



Baud Rate
19200 or 38400 bps

Quantity of Slave Stations
1 through 31

Slave Station Number
01 through 31

TX: Transmit from master
RX: Receive to master

Selected data quantity
0 through 6 words

Transmit/Receive Data Quantity (Words)
Select the quantity of data registers for transmit and receive data per slave station: 0 through 6 words

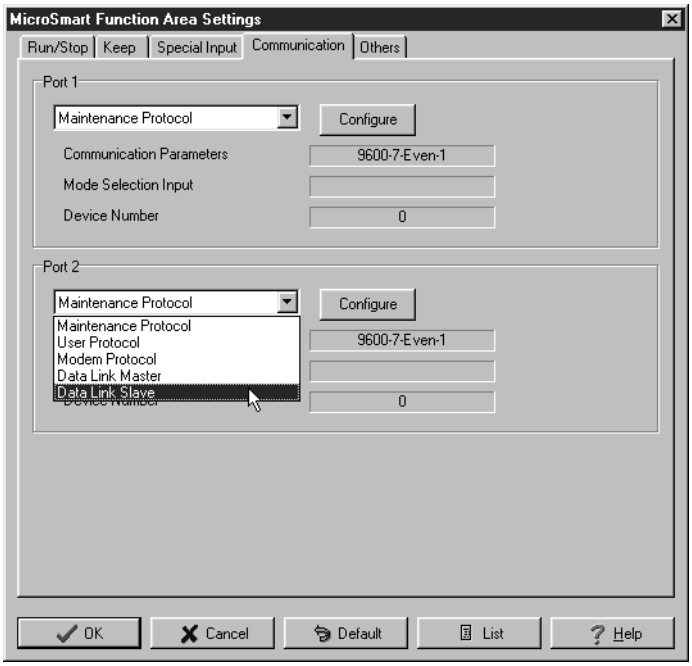
Selects the same quantities of transmit and receive data for all slave stations.

Note: When the data link system includes the MICRO^3 or MICRO^{3}C, select 19200 bps baud rate, and select 2 words of transmit/receive data for MICRO^3 or MICRO^{3}C.

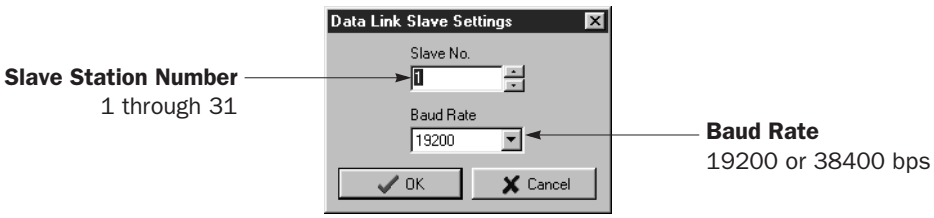
4. Click the **OK** button.

Data Link Slave Station

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Setting dialog box appears.
2. Click the **Communication** tab, and select **Data Link Slave** in the Port 2 pull-down list.



3. The Data Link Slave Settings dialog box appears. Select a slave station number and baud rate.



4. Click the **OK** button.

Refresh Mode

In the data link communication, the master station sends data to a slave station and receives data from the slave station one after another. After receiving data from slave stations, the master station stores the data into data registers allocated to each slave station. The process of updating data into data registers is called refresh. The master station refreshes the received data in the separate refresh mode as illustrated below:

Mode	Separate Refresh Mode
Master Station Scan Time	Since the master station refreshes received data at the END processing of the user program, the scan time in the master station is affected.
Master Station Refresh Timing	Data received from one slave station is refreshed at each END processing.
Applicable Master Station	MicroSmart, OpenNet Controller, MICRO^3, MICRO^{3}C, FA-3S (PF3S-SIF4)
Applicable Slave Station	MicroSmart, OpenNet Controller, MICRO^3, MICRO^{3}C, FA-3S (PF3S-SIF4)

When the data link system contains the MicroSmart and MICRO^3/MICRO^{3}C, set the baud rate to 19200 bps and transmit/receive data quantity to 2 words in the Function Area Settings for the MicroSmart to communicate with MICRO^3/MICRO^{3}C stations.

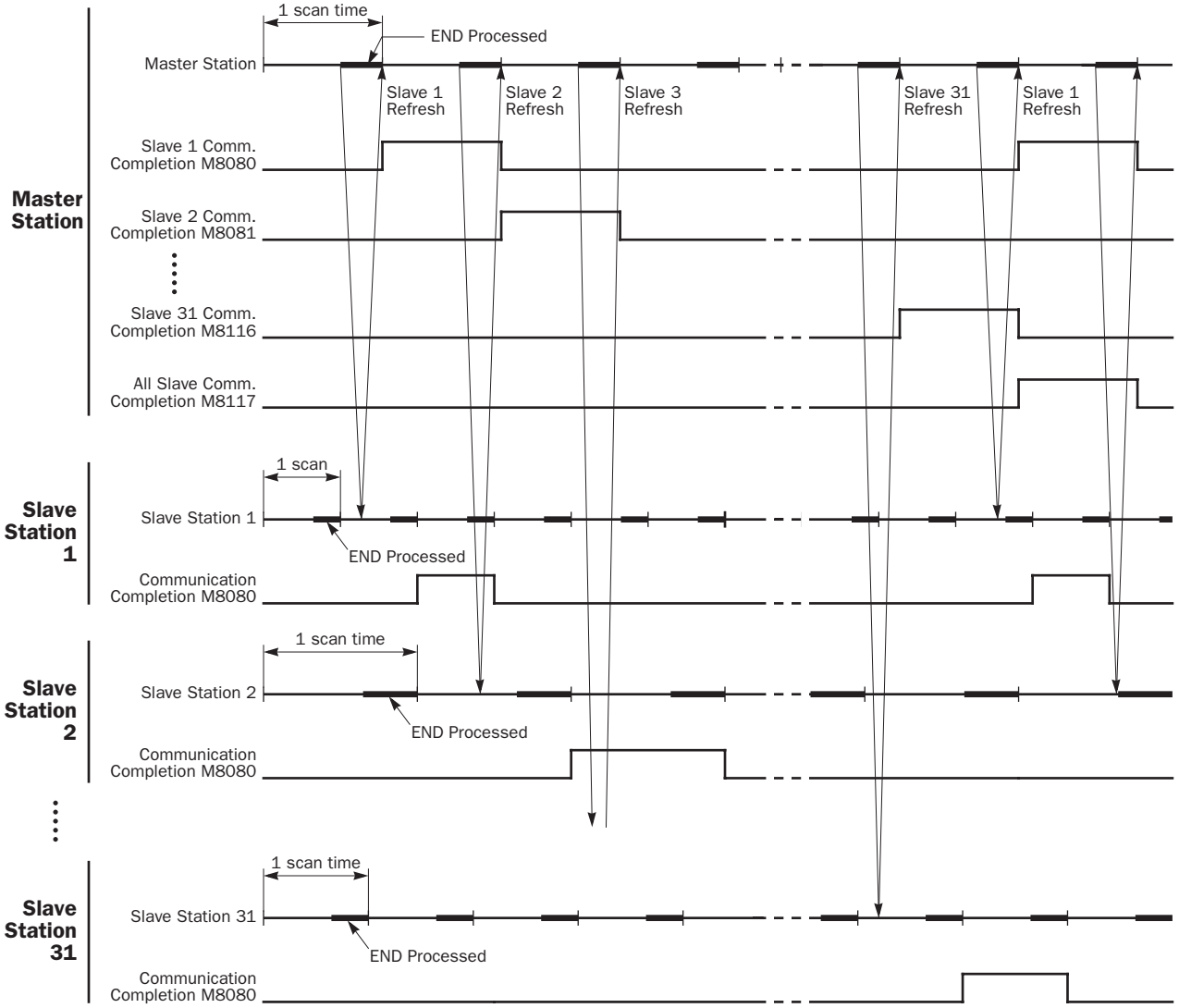
Separate Refresh Mode Communication Sequence

The master station can communicate with only one slave station in one scan time. When a slave station receives a communication from the master station, the slave station returns data stored in data registers assigned for data link communication. When the maximum 31 slave stations are connected, the master station requires 31 scans to communicate with all slave stations.

Both master and slave stations refresh communication data in the END processing at each station. When data refresh is complete, communication completion special internal relays M8080 through M8116 (slave station communication completion relay) go on at the master station for one scan time after the data refresh. At each slave station, special internal relay M8080 (communication completion relay) goes on.

When the master station completes communication with all slave stations, special internal relay M8117 (all slave station communication completion relay) goes on at the master station for one scan time.

The communication sequence in the separate refresh mode is shown below:



Refresh Time at Master Station for Communication with One Slave Station (Trf)

The master station requires the following time to refresh the transmit and receive data for communication with one slave station.

[Baud Rate 19200 bps] Trf = 4.2 ms + 2.4 ms × (Transmit Words + Receive Words) + 1 scan time

[Baud Rate 38400 bps] Trf = 2.2 ms + 1.3 ms × (Transmit Words + Receive Words) + 1 scan time

Total Refresh Time at Master Station for Communication with All Slave Stations (Trfn)

The master station requires the following time to refresh the transmit and receive data for communication with all slave stations, that is the total of refresh times.

[Baud Rate 19200 bps] $Trfn = \sum Trf = \sum$ {4.2 ms + 2.4 ms × (Transmit Words + Receive Words) + 1 scan time}

[Baud Rate 38400 bps] $Trfn = \sum Trf = \sum$ {2.2 ms + 1.3 ms × (Transmit Words + Receive Words) + 1 scan time}

Example: Refresh Time

When data link communication is performed with such parameters as transmit words 6, receive words 6, slave stations 8, and average scan time 20 ms, then the total refresh time Trf8 for communication with all eight slave stations will be:

[Baud Rate 19200 bps] Trf8 = {4.2 ms + 2.4 ms × (6 + 6) + 20 ms} × 8 = 424.0 ms

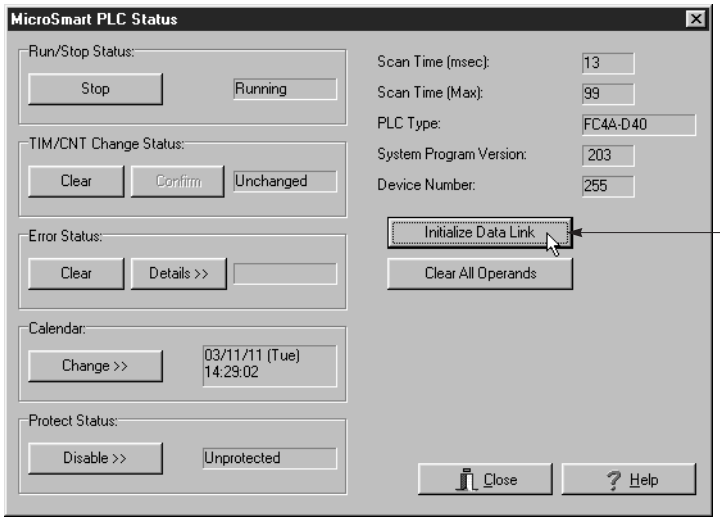
[Baud Rate 38400 bps] Trf8 = {2.2 ms + 1.3 ms × (6 + 6) + 20 ms} × 8 = 302.4 ms

Operating Procedure for Data Link System

To set up and use a data link system, complete the following steps:

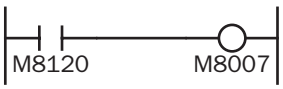
1. Connect the MicroSmart CPU modules at the master station and all slave stations as illustrated on page 25-2.
2. Create user programs for the master and slave stations. Different programs are used for the master and slave stations.
3. Using WindLDR, access **Configure > Function Area Settings > Communication** and make settings for the master and slave stations. For programming WindLDR, see pages 25-7 and 25-8.
4. Download the user programs to the master and slave stations.
5. To start data link communication, power up slave stations first, and power up the master station at least 1 second later. Monitor the data registers used for data link at the master and slave stations.

Note: To enable data link communication, power up slave stations first. If a slave station is powered up later than or at the same time with the master station, the master station does not recognize the slave station. To make the master station recognize the slave station in this case, turn on special internal relay M8007 (data link communication initialize flag) at the master station (see page 25-6), or in WindLDR select **Online** > **Monitor**, followed by **Online** > **PLC Status** and click the **Initialize Data Link** button.



Data Link Initialization Program

If the master station does not recognize the slave station when the master station is powered up, include the following program into the user program for the master station.



M8120 is the initialize pulse special internal relay.

M8007 is the data link communication initialize flag.

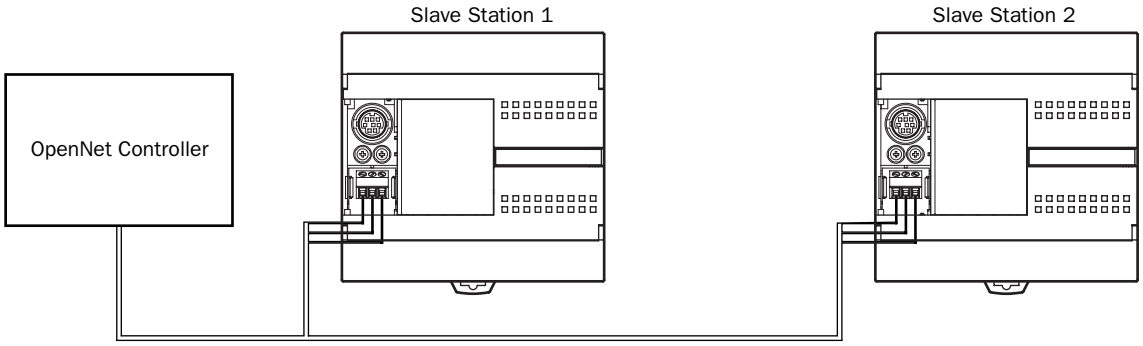
When the master station CPU module starts to run, M8120 turns on M8007 for one scan to initialize the data link communication. The master station will recognize the slave station.

Data Link with Other PLCs

The data link communication system can include IDEC's OpenNet Controller, MICRO^3/MICRO^{3}C micro programmable controllers, and FA-3S programmable controllers using serial interface modules.

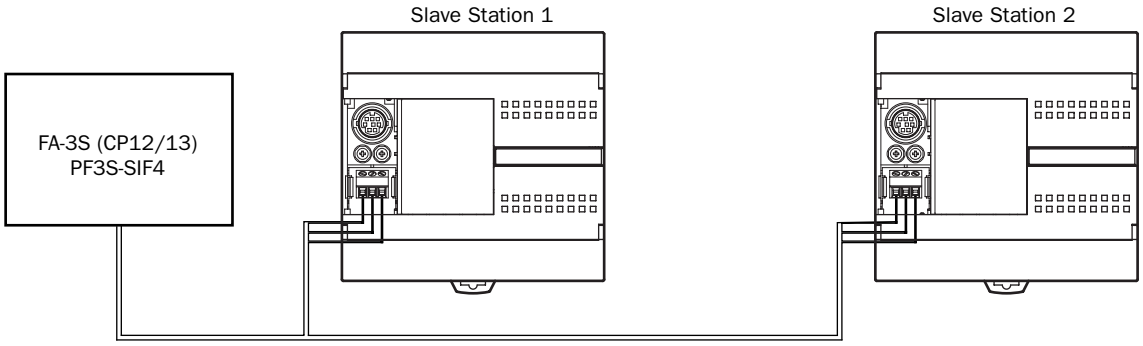
Data Link with OpenNet Controller

OpenNet Controller Settings	MicroSmart Settings	MicroSmart Settings
Transmit data: 6 words Receive data: 6 words Baud rate: 19200 or 38400 bps	Slave station number 1	Slave station number 2



Data Link with FA-3S High-performance CPU using Serial Interface Module PF3S-SIF4

FA-3S (PF3S-SIF4) Settings	MicroSmart Settings	MicroSmart Settings
Transmit data: 6 words Receive data: 6 words Baud rate: 19200 or 38400 bps	Slave station number 1	Slave station number 2



26: COMPUTER LINK COMMUNICATION

Introduction

When the MicroSmart CPU module is connected to a computer, operating status and I/O status can be monitored on the computer, data in the CPU module can be monitored or updated, and user programs can be downloaded and uploaded. The CPU module can also be started and stopped from the computer. A maximum of 32 all-in-one 16- and 24-I/O type CPU modules or slim type CPU modules can be connected to one computer in the 1:N computer link system. The all-in-one 10-I/O type CPU module can be used in the 1:1 computer link system only.

The maximum communication speed in the 1:1 or 1:N computer link system is 19,200 bps.

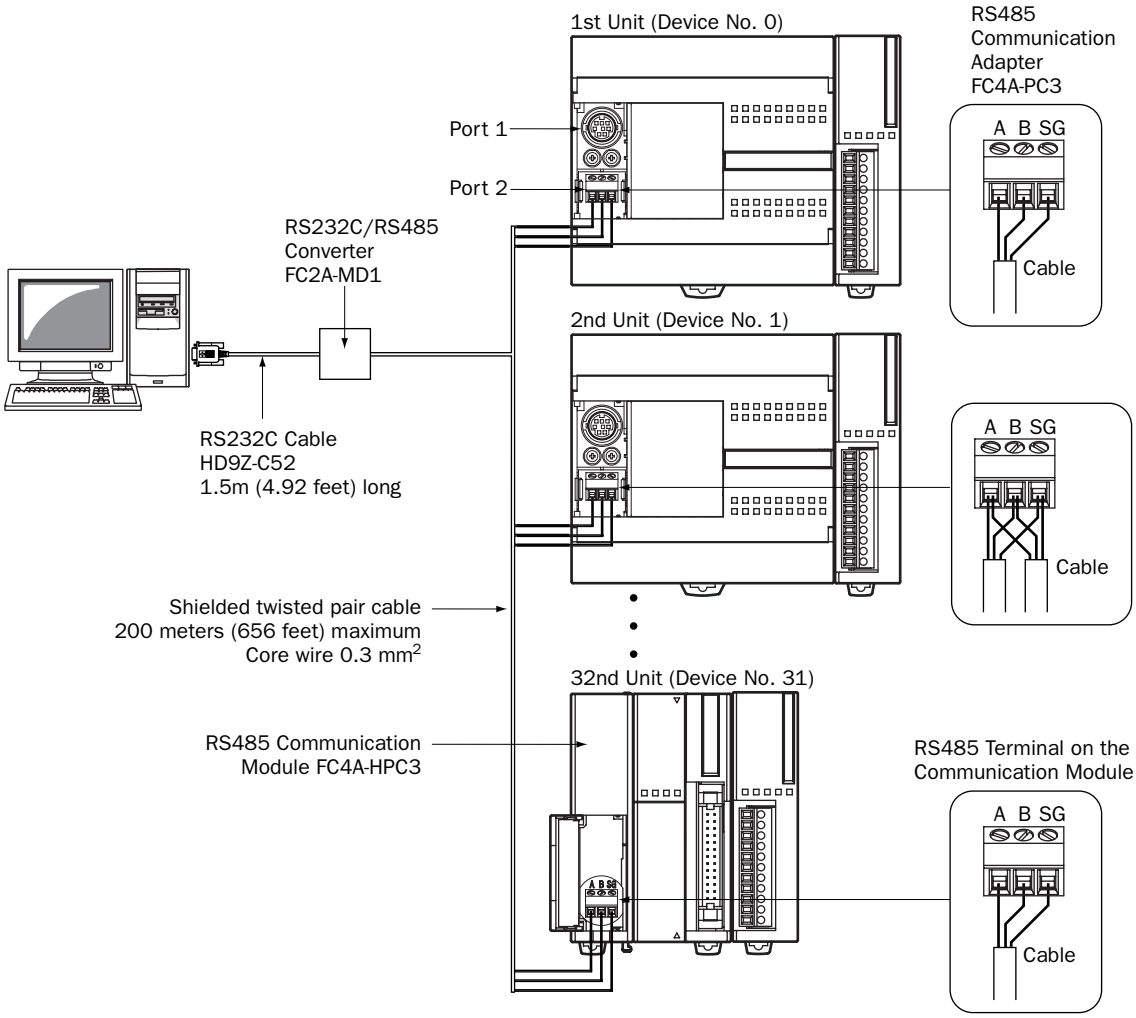
This chapter describes the 1:N computer link system. For the 1:1 computer link system, see page 4-1.

Computer Link System Setup (1:N Computer Link System)

To set up a 1:N communication computer link system, install the RS485 communication adapter (FC4A-PC3) to the port 2 connector on the all-in-one 16- or 24-I/O type CPU module, or mount the RS485 communication module (FC4A-HPC3) next to the slim type CPU module. Connect the RS232C/RS485 converter to the RS485 terminals A, B, and SG on every CPU module using a shielded twisted pair cable as shown below. The total length of the cable for the computer link system can be extended up to 200 meters (656 feet).

Connect the RS232C port on the computer to the RS232C/RS485 converter using the RS232C cable HD9Z-C52. The RS232C cable has a D-sub 9-pin female connector for connection with a computer.

OpenNet Controllers, MICRO^3, and MICRO^{3}C can be connected to the same 1:N computer link system.



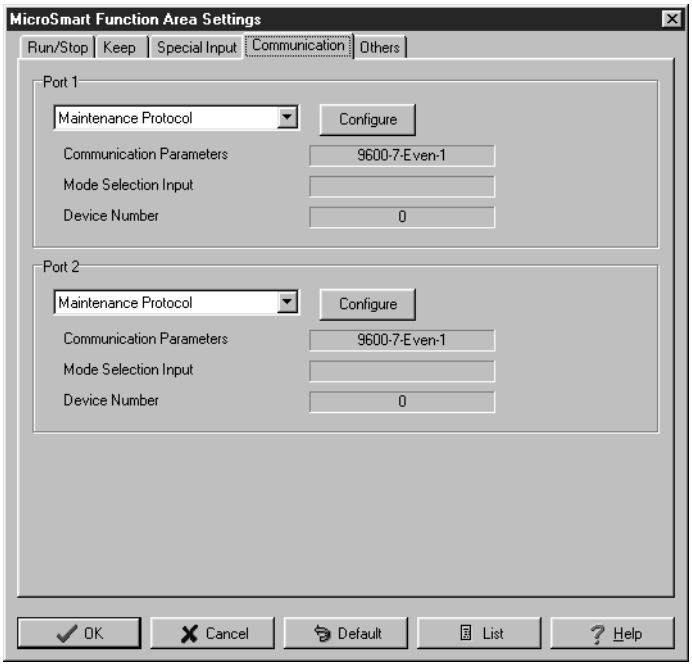
Programming WindLDR

In the 1:1 computer link system, a computer can be connected to either port 1 or 2 on the MicroSmart CPU module. In the 1:N computer link system, a computer must be connected to port 2 on the CPU module and every CPU module must have a unique device number 0 through 31. The Communication page in the Function Area Settings must be programmed for each station in the computer link system. If required, communication parameters can also be changed.

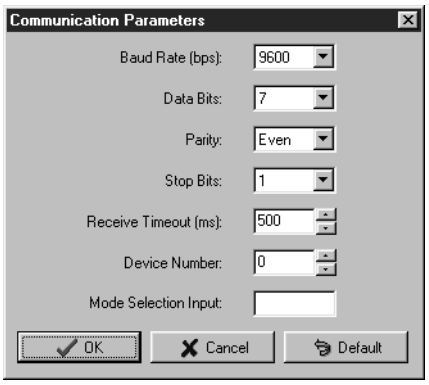
Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Setting dialog box appears.

2. Click the **Communication** tab, and select **Maintenance Protocol** in the Port 1 or 2 pull-down list.



3. Click the **Configure** button. The Communication Parameters dialog box appears. Change settings, if required.



Baud Rate (bps)	1200, 2400, 4800, 9600, 19200
Data Bits	7 or 8
Parity	None, Odd, Even
Stop Bits	1 or 2
Receive Timeout (ms)	10 to 2540 (10-msec increments) (Receive timeout is disabled when 2550 is selected.)
Device Number	0 to 31
Mode Selection Input	Any input number

Note: Only when the mode selection input is turned on, the selected communication parameters are enabled. Otherwise, default communication parameters take effect; 9600 bps, 7 data bits, even parity, 1 stop bit, receive timeout 500 msec.

4. Click the **OK** button.

Assigning Device Numbers

When assigning a unique device number of 0 through 31 to each CPU module for the 1:N computer link network, download the user program containing the device number setting to each CPU module in the 1:1 computer link system, then the new device number is assigned to the CPU module. Make sure that there is no duplication of device numbers in a 1:N computer link network.

Communication Settings

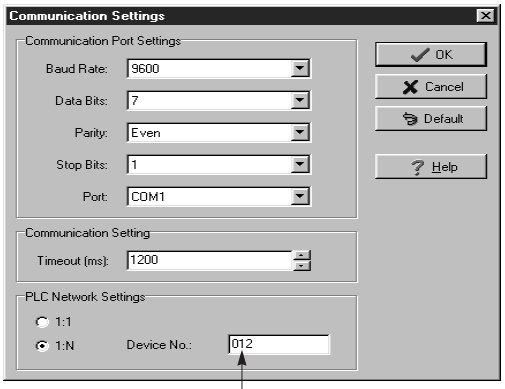
When monitoring the MicroSmart operation or downloading a user program using WindLDR, make sure that the same communication settings are selected for the CPU module and WindLDR, so that the computer communicate with the MicroSmart in either the 1:1 or 1:N computer link system. To change the communication settings for WindLDR, access the **Communication Settings** dialog box from the **Configure** menu as shown below.

When communicating in the 1:N computer link system for monitoring or downloading, select the device number of the CPU module also in the **Communication Settings** dialog box.

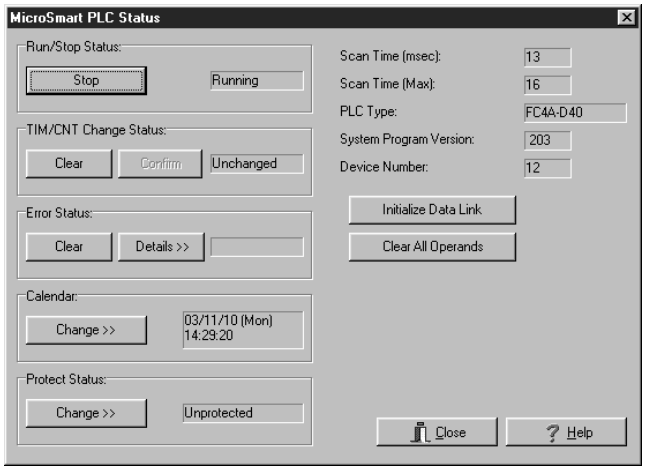
Monitoring PLC Status

The following example describes the procedures to monitor the operating status of the MicroSmart assigned with device number 12 in a 1:N communication computer link system.

1. From the WindLDR menu bar, select **Configure > Communication Settings**. The Communication Settings dialog box appears.
2. Under PLC Network Settings, click the **1:N** button to select 1:N communication, and enter 12 in the **Device No.** field.
3. From the WindLDR menu bar, select **Online > Monitor**. The ladder diagram on the screen enters the monitor mode.
4. From the WindLDR menu bar, select **Online > PLC Status**. The PLC Status dialog box appears.



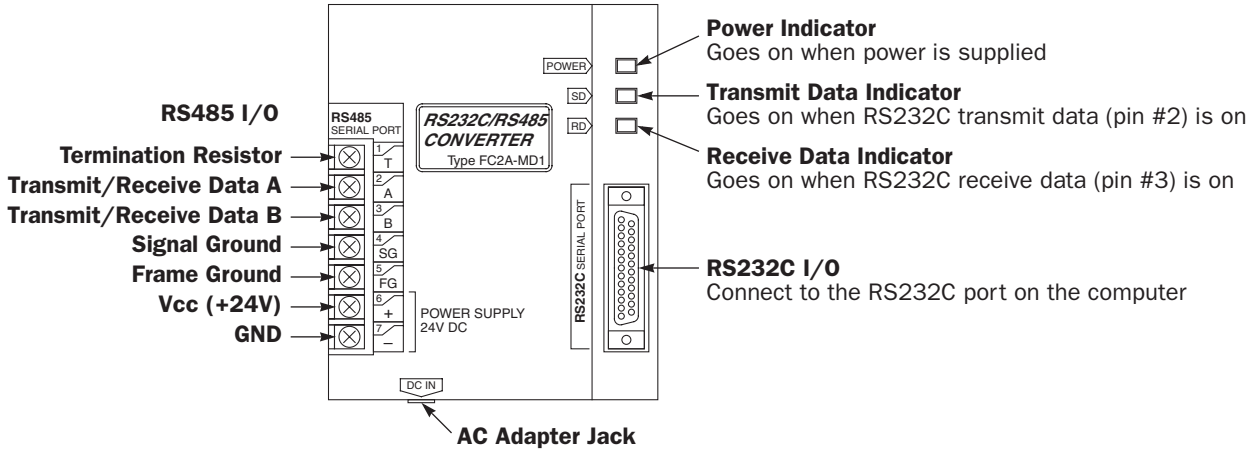
Device No.:
Enter 12 to select a device number to communicate with.



RS232C/RS485 Converter FC2A-MD1

The RS232C/RS485 converter FC2A-MD1 is used to convert data signals between EIA RS232C and EIA RS485. This converter makes it possible to connect a host device with RS232C interface to multiple MicroSmart CPU modules using one cable.

Parts Description



Note: Connect 24V DC to POWER SUPPLY + and – terminals or connect an AC adapter with 9V DC, 350mA output to the AC adapter jack.

Note: The FC2A-MD1 contains a 220Ω termination resistor on the RS485 line, eliminating the need for an external termination resistor. To use the internal termination resistor, connect terminal T to terminal B. When the termination resistor is not needed, disconnect terminal T from terminal B.

Specifications

General Specifications

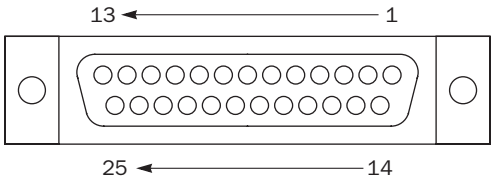
Rated Power Voltage	Power terminals: 24V DC ±20% (ripple 10% maximum) DC IN adapter jack: 9V DC, 350mA supplied from AC adapter
Current Draw	Power terminals: Approx. 40 mA at the rated voltage
Operating Temperature	0 to 60°C
Storage Temperature	–20 to +70°C
Operating Humidity	45 to 85% RH (no condensation)
Vibration Resistance	5 to 55 Hz, 60 m/sec^2, 2 hours each in 3 axes
Shock Resistance	300 m/sec^2, 3 shocks each in 3 axes
Dielectric Strength	1500V AC, 1 minute between live parts and dead parts
Insulation Resistance	10 MΩ minimum between live parts and dead parts (500V DC megger)
Noise Resistance	Power terminals: ±1 kV, 1 µs (using noise simulator)
Weight	Approx. 550g

Serial Interface Specifications

Standards in Compliance	EIA standard RS232C (D-sub 25-pin female connector) EIA standard RS485 (screw terminals)
Communication Method	Half-duplex
Communication Configuration	1:N (N ≤ 32)
Communication Cable	Shielded twisted-pair cable
Communication Baud Rate	9600 bps (fixed)
Slave Stations	32 slave stations maximum (RS485 line)
Maximum Cable Length	RS232C: 15m (49.2 ft.) RS485: Total 200m (656 ft.)

RS232C Connector Pinouts

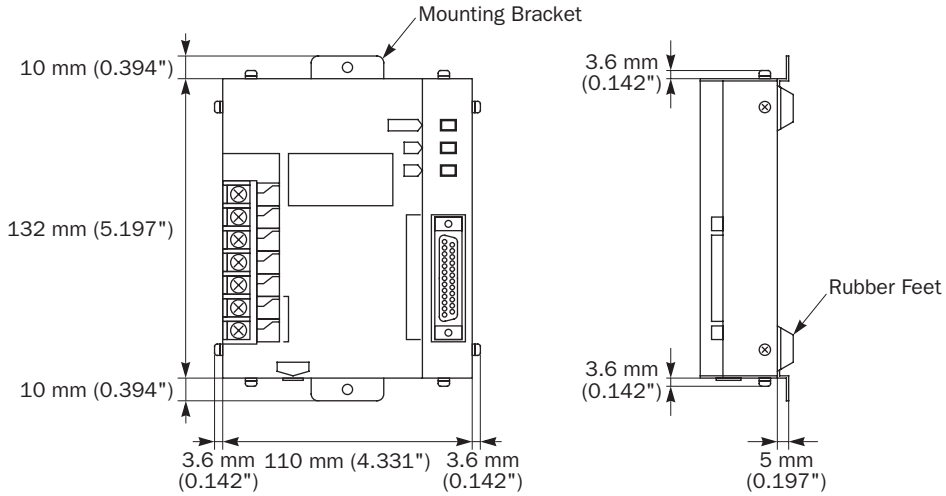
D-sub 25-pin Female Connector

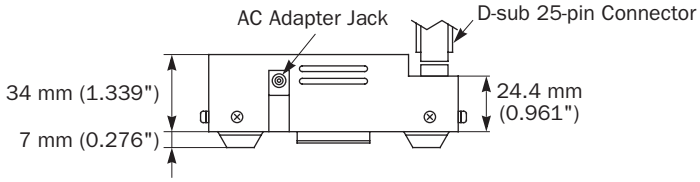


Note: Terminals 4 and 5 are connected together internally.

Pin No.	Description	
1	GND	Frame Ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	(RTS)	Unused
5	(CTS)	Unused
6	(NC)	Unused
7	GND	Signal Ground
8-25	(NC)	Unused

Dimensions





Note: When mounting the RS232C/RS485 converter on a panel surface, remove the rubber feet; then attach the supplied mounting brackets on the bottom of the converter using screws.

RS232C Cable HD9Z-C52

Connector for RS232C/RS485 Converter

Description		Pin No.
GND	Frame Ground	**1**
TXD	Transmit Data	**2**
RXD	Receive Data	**3**
RTS	Request to Send	**4**
CTS	Clear to Send	**5**
DSR	Data Set Ready	**6**
DCD	Data Carrier Detect	**8**
DTR	Data Terminal Ready	**20**
GND	Signal Ground	**7**

D-sub 25-pin male connector

1.5m (4.92 ft.) long

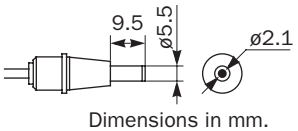
Connector for Computer

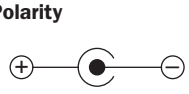
Pin No.	Symbol
1	DCD
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

D-sub 9-pin female connector

AC Adapter

The RS232C/RS485 converter is powered by a 24V DC source or an AC adapter with 9V DC, 350mA output capacity.





27: MODEM MODE

Introduction

This chapter describes the modem mode designed for communication between the MicroSmart and another MicroSmart or any data terminal equipment through telephone lines. Using the modem mode, the MicroSmart can initialize a modem, dial a telephone number, send an AT command, enable the answer mode to wait for an incoming call, and disconnect the telephone line. These operations can be performed simply by turning on a start internal relay dedicated to each operation.

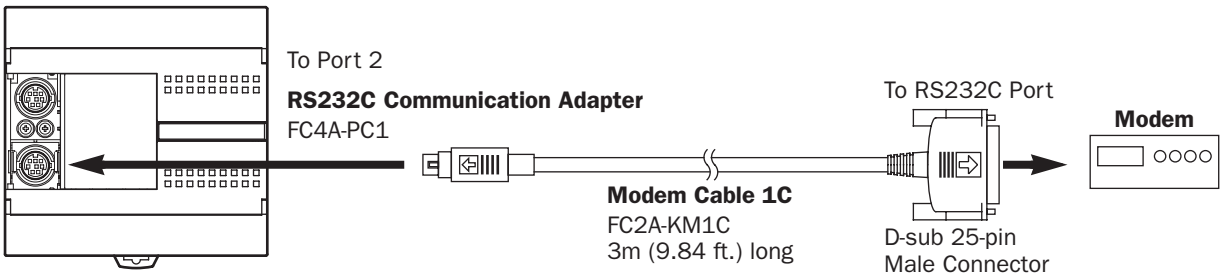
⚠ Caution

- The modem mode provides for a simple modem control function so that the MicroSmart can initialize a modem, dial a destination telephone number, or answer an incoming call. The performance of the modem communication using the modem mode depends on the modem functions and telephone line situations. The modem mode does not prevent intrusion or malfunctions of other systems. For practical applications, confirm the communication function using the actual system setup and include safety provisions.
- While communicating through modems, the telephone line may be disconnected unexpectedly or receive data errors may occur. Provisions against such errors must be included in the user program.

System Setup

To connect a modem to the MicroSmart, install the RS232C communication adapter (FC4A-PC1) to the port 2 connector on the all-in-one 16- or 24-I/O type CPU module, or mount the RS232C communication module (FC4A-HPC1) next to the slim type CPU module, and use the modem cable 1C (FC2A-KM1C). To enable the modem mode, select Modem Protocol for Port 2 using WindLDR (**Configure > Function Area Settings > Communication**). The all-in-one 10-I/O type CPU module does not have the modem communication capability.

16- or 24-I/O Type CPU Module



Mini DIN Connector Pinouts

Description		Pin
Shield		Cover
RTS	Request to Send	1
DTR	Data Terminal Ready	2
TXD	Transmit Data	3
RXD	Receive Data	4
DSR	Data Set Ready	5
SG	Signal Ground	6
SG	Signal Ground	7
NC	No Connection	8

D-sub 25-pin Connector Pinouts

Pin	Description	
1	FG	Frame Ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	RTS	Request to Send
5	NC	No Connection
6	NC	No Connection
7	SG	Signal Ground
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready

⚠ Caution

- Do not connect the NC (no connection) pin to any line; otherwise, the MicroSmart or modem may be damaged.
- Modem cables for Apple Macintosh computers cannot be used for the MicroSmart.
- Do not connect the cable to the port 1 or port 2 (RS485); otherwise, the MicroSmart or modem may be damaged.

Applicable Modems

Any Hayes compatible modem can be used. Modems with a communications rate of 9600 bps or more between modems are recommended. Use modems of the same make and model at both ends of the communication line.

Special Internal Relays for Modem Mode

Special internal relays M8050-M8077 are allocated to the modem mode. M8050-M8056 are used to send an AT command or disconnect the telephone line. M8060-M8066 and M8070-M8076 turn on to indicate the results of the command. M8057, M8067, and M8077 are used to indicate the status of the modem mode.

All completion and failure internal relays are turned off when another start internal relay is turned on.

Start and Result Internal Relays

Mode	Command	Start IR	Completion IR	Failure IR	Data Register
Originate Mode	Initialization String	M8050	M8060	M8070	D8145-D8169
	ATZ	M8051	M8061	M8071	—
	Dialing	M8052	M8062	M8072	D8170-D8199
Disconnect Mode	Disconnect Line	M8053	M8063	M8073	—
AT General Command Mode	AT Command	M8054	M8064	M8074	D8130-D8144
Answer Mode	Initialization String	M8055	M8065	M8075	D8145-D8169
	ATZ	M8056	M8066	M8076	—

When one of start internal relays M8050-M8056 is turned on, a corresponding command is executed once. To repeat the command, reset the start internal relay and turn the internal relay on again.

Completion or failure of a command is determined as described below:

Completion: The command is transmitted repeatedly as many as the retry cycles specified in data register D8109. When the command is completed successfully, the completion IR is turned on and the command is not executed for the remaining cycles.

Failure: The command is transmitted repeatedly but failed in all trials as many as the retry cycles specified in data register D8109.

Status Internal Relays

Status IR	Status	Description
M8057	AT Command Execution	ON: AT command is in execution (start IR is on) OFF: AT command is not in execution (completion or failure IR is on)
M8067	Operational State	ON: Command mode OFF: On-line mode
M8077	Line Connection	ON: Telephone line connected (Note) OFF: Telephone line disconnected

Note: While M8077 (line connection) is off, the MicroSmart cannot send and receive maintenance communication and user communication through port 2. When M8077 is turned on, maintenance communication or user communication is enabled depending on the value stored in data register D8103 (on-line mode protocol selection).

Special Data Registers for Modem Mode

Special data registers D8103 and D8109-D8199 are allocated to the modem mode. When the MicroSmart starts to run, D8109 and D8110 store the default values, and D8145-D8169 store the default initialization string.

Data Register	Stored Data	Description
D8103	On-line Mode Protocol Selection	The D8103 value selects the protocol for the RS232C port 2 after telephone line is connected. 0 (other than 1): Maintenance protocol 1: User protocol
D8109	Retry Cycles (default = 3)	The D8109 value selects how many retries will be made until the operation initiated by a start internal relay M8050-M8056 is completed. 0: No retry 1-65535: Executes a specified number of retries
D8110	Retry Interval (default = 90 sec)	The D8110 value specifies the interval to start a retry of dialing when a dialing fails with the retry cycles set to a value more than 1. (Other start commands are repeated continuously as many as the retry cycles.) Valid value: 0 to 65535 (seconds) If a telephone line is not connected within the retry interval, the MicroSmart starts a retry. Consequently, if the retry interval is set to a too small value, the telephone line can not be connected correctly.
D8111	Modem Mode Status	Modem mode status is stored (see page 27-7). When not in the modem mode, D8111 stores 0.
D8115-D8129	AT Command Result Code	AT command result codes returned from modem are stored. When the result code exceeds 30 bytes, first 30 bytes are stored.
D8130-D8144	AT Command String	AT command string for the AT general command mode is stored. Enter an AT command string to these data registers to send by turning on M8054 (AT command start internal relay). "AT" and LF (0Ah) are appended automatically.
D8145-D8169	Initialization String	Initialization string for the originate and answer modes is stored. To change the initialization string, enter a new value to these data registers. The new value is sent by turning on M8050 or M8055. "AT" and LF (0Ah) are appended automatically.
D8170-D8199	Telephone Number	Telephone number for dialing in the originate mode is stored. "ATD" and LF (0Ah) are appended automatically.

Originate Mode

The originate mode is used to send an initialization string to the modem, issue the ATZ command to reset the modem, and dial the telephone number. To execute a command, turn on one of start internal relays M8050-M8052. If two or more start internal relays are turned on simultaneously, an error will result and error code 61 is stored in modem mode status data register D8111 (see page 27-7). When a start internal relay is turned on, a corresponding sequence of commands is executed once as described below. When the start command fails, the same command is repeated as many as the retry cycles specified by D8109.

M8050: Send an initialization string, send the ATZ command, and dial the telephone number

M8051: Send the ATZ command and dial the telephone number

M8052: Dial the telephone number

Initialization String in Originate Mode

When the modem mode is enabled as described on page 27-1 and the MicroSmart is started to run, the default initialization string is stored to data registers D8145-D8169 at the END processing of the first scan. To send the initialization string from the MicroSmart to the modem, turn M8050 on; then the ATZ command is issued and the telephone number is dialed successively.

Default Initialization String: ATE0Q0V1&D2&C1\V0X4&K3\A0\N5S0=2&W[CR][LF]

AT and [LF] are appended at the beginning and end of the initialization string automatically by the system program and are not stored in data registers.

DR	8145	8146	8147	8148	8149	8150	8151	8152	8153	8154	8155	8156	8157	8158	8159	8160	8161
AT	E0	Q0	V1	&D	2&	C1	\V	0X	4&	K3	\A	0\	N5	S0	=2	&W	0D00 [LF]

Depending on your modem and telephone line, the initialization string may have to be modified. Consult the manual for your modem.

Changes can be made by entering required values to data registers D8145-D8169. Store two characters in one data register; the first character at the upper byte and the second character at the lower byte in the data register. AT and [LF] need not be stored in data registers. Use the MOV (move) instructions on WindLDR to set the initialization string characters and ASCII value 0Dh for [CR] at the end. Program the MOV instructions to replace the default values in D8145-D8169 stored in the first scan and execute the MOV instructions in a subsequent scan. For essential commands which must be included in the initialization string, see page 27-8. After the new values are stored, turn on M8050 to send the new initialization string to the modem.

When the initialization string has been sent successfully, internal relay M8060 is turned on. If the initialization string fails, internal relay M8070 is turned on. When the subsequent commands of ATZ and dialing are also completed successfully, M8061 and M8062 will also be turned on.

The default initialization string or the modified initialization string stored in D8145-D8169 is also used for the initialization in the answer mode.

ATZ (Resetting the Modem) in Originate Mode

The default initialization string specifies to be stored in the non-volatile memory of the modem, using the &W command. The initialization string is restored when the modem is powered up or when the ATZ command is issued. The MicroSmart sends the ATZ command to the modem, following the initialization string when M8050 is turned on. The ATZ command can also be issued separately by turning M8051 on, followed by the dial command to be executed automatically.

ATZ Command: ATZ[CR][LF]

When the ATZ command has been completed successfully, internal relay M8061 is turned on. If the ATZ command fails, internal relay M8071 is turned on. When the subsequent dialing is also completed successfully, M8062 will also be turned on.

If the initialization string has been stored in the non-volatile memory of the modem, M8050 may be skipped. Start with M8051 to send the ATZ command.

Dialing the Telephone Number

Data registers D8170-D8199 are allocated to the telephone number. Before turning on one of the start internal relays M8050-M8052 for the originate mode, store the telephone number in data registers starting with D8170. One data register stores two characters: the first character at the upper byte and the second character at the lower byte in the data register. Since 30 data registers are allocated to the telephone number, up to 60 characters can be stored, as many as the modem capacity allows. Use the MOV (move) instructions on WindLDR to set the telephone number and execute the MOV instructions before turning on start internal relays M8050-M8052.

Example of Dial Command: ATD1234[CR][LF]

ATD and [LF] are appended at the beginning and end of the dial command automatically by the system program and need not be stored in data registers. To program the telephone number of the example above, store the telephone number and ASCII value 0Dh for [CR] to data registers starting with D8170. It is also possible to store character T for touch-tone phone or P for pulse or rotary phone.

D8170	3132h	31h = “1” 32h = “2”
D8171	3334h	33h = “3” 34h = “4”
D8172	0D00h	0Dh = [CR] All characters subsequent to [CR] are ignored.

As described above, when start internal relay M8050 is turned on, the initialization string is sent, followed by the ATZ command and the dial command. When start internal relay M8051 is turned on, the ATZ command is sent, followed by the dial command. The dial command can also be sent separately by turning on start internal relay M8052.

If retry cycles are set to data register D8109, the dial command is repeated at retry intervals specified by D8110 (default 90 seconds) as many as the specified retry cycles (default 3 cycles) until the telephone line is connected.

When the dial command has been completed successfully, internal relay M8062 is turned on. If the dial command fails, internal relay M8072 is turned on.

The dial command is determined successful when the DCD signal is turned on.

Note: When the MicroSmart is powered down while the telephone line is connected, the telephone line is disconnected because the DTR signal is turned off. This method should not be used for disconnecting the telephone line. Always use M8053 to disconnect the telephone line as described below.

RS232C Port Communication Protocol

Before the telephone line is connected in the modem mode after power-up, the RS232C port 2 can only send out an AT command by turning on a start internal relay M8050-M8056. The communication protocol for the RS232C port 2 after the telephone line is connected is selected by the value stored in data register D8103.

D8103 Value	RS232C Port 2 Communication Protocol in the On-Line Mode
0 (other than 1)	Maintenance protocol
1	User protocol

When the telephone line is disconnected, the RS232C port 2 restores the state as before the telephone line was connected, whether D8103 is set to 0 or 1.

When using a TXD or RXD instruction in the user communication mode while the telephone line is connected, insert internal relay M8077 (line connection) as an input condition for the TXD or RXD instruction. After the telephone line is connected, make sure of an approximately 1-second interval before executing the TXD or RXD instruction until the telephone line connection stabilizes.

Note: When the MicroSmart is stopped while the telephone line is connected, the RS232C port 2 protocol changes to the maintenance protocol even if D8103 is set to 1 (user protocol in the on-line mode); then the telephone line remains connected. When the MicroSmart is restarted, the user protocol is enabled again.

Disconnect Mode

The disconnect mode includes only one command to disconnect the telephone line. To disconnect the telephone line, turn on internal relay M8053. The telephone line is disconnected by turning off the DTR signal since the initialization string includes the &D2 command.

While a modem command is executed, another command cannot be executed. If two or more start internal relays are turned on simultaneously, an error will result and error code 61 is stored in modem mode status data register D8111 (see page 27-7).

When the disconnect command has been completed successfully, internal relay M8063 is turned on. If the disconnect command fails, internal relay M8073 is turned on.

The disconnect command is determined successful when the DCD signal is turned off.

After the telephone line is disconnected, the RS232C port 2 restores the state as before the telephone line was connected whether D8103 is set to 0 or 1 so that the RS232C port 2 can be controlled by turning on a start internal relay M8050-M8056.

AT General Command Mode

Data registers D8130-D8144 are allocated to the AT command string. Before turning on start internal relay M8054 for the AT general command mode, store an AT command string in data registers starting with D8130. One data register stores two characters: the first character at the upper byte and the second character at the lower byte in the data register. Use the MOV (move) instructions on WindLDR to set the AT command string and execute the MOV instructions before turning M8054 on.

Example of AT Command: ATE0Q0V1[CR][LF]

AT and [LF] are appended at the beginning and end of the AT general command string automatically by the system program and need not be stored in data registers. To program the AT command string of the example above, store the command characters and ASCII value 0Dh for [CR] to data registers starting with D8130.

D8130	4530h	45h = "E"	30h = "0"
D8131	5130h	51h = "Q"	30h = "0"
D8132	5631h	56h = "V"	31h = "1"
D8133	0D00h	0Dh = [CR]	All characters subsequent to [CR] are ignored.

When the AT general command has been completed successfully, internal relay M8064 is turned on. If the AT general command fails, internal relay M8074 is turned on.

The AT general command is determined successful when result code [CR][LF]OK[CR][LF] returned from the modem is received.

Answer Mode

The answer mode is used to send an initialization string to the modem and to issue the ATZ command to reset the modem. To execute a command, turn on one of start internal relays M8055 or M8056. If two or more start internal relays are turned on simultaneously, an error will result and error code 61 is stored in modem mode status data register D8111 (see page 27-7). When a start internal relay is turned on, a corresponding sequence of commands is executed once as described below.

M8055: Send initialization string and send the ATZ command

M8056: Send the ATZ command

Initialization String in Answer Mode

When the modem mode is enabled as described on page 27-1 and the MicroSmart is started to run, the default initialization string is stored to data registers D8145-D8169 at the END processing of the first scan. To send the initialization string from the data registers to the modem, turn M8055 on; then the ATZ command is issued subsequently.

Default Initialization String: ATE0Q0V1&D2&C1\V0X4&K3\A0\N5S0=2&W[CR][LF]

As described in the Originate Mode, the initialization string can be modified to match your modem. For details of modifying the initialization string, see page 27-3.

When the initialization string has been sent successfully, internal relay M8065 is turned on. If the initialization string fails, internal relay M8075 is turned on. When the subsequent ATZ command is also completed successfully, M8066 will also be turned on.

ATZ (Resetting the Modem) in Answer Mode

The default initialization string specifies to be stored in the non-volatile memory of the modem, using the &W command. The initialization string is restored when the modem is powered up or when the ATZ command is issued. The MicroSmart sends the ATZ command to the modem following the initialization string when M8055 is turned on. The ATZ command can also be issued separately by turning M8056 on.

ATZ Command: ATZ[CR][LF]

When the ATZ command has been completed successfully, internal relay M8066 is turned on. If the ATZ command fails, internal relay M8076 is turned on.

If the initialization string has been stored in the non-volatile memory of the modem, M8055 may be skipped. Start with M8056 to send the ATZ command.

Modem Mode Status Data Register

When the modem mode is enabled, data register D8111 stores a modem mode status or error code.

D8111 Value	Status	Description
0	Not in the modem mode	Modem mode is not enabled.
10	Ready for connecting line	Start internal relays except for disconnecting line can be turned on.
20	Sending initialization string (originate mode)	A start internal relay is in operation in the first try or subsequent retrial.
21	Sending ATZ (originate mode)	
22	Dialing	
23	Disconnecting line	
24	Sending AT command	
25	Sending initialization string (answer mode)	
26	Sending ATZ (answer mode)	
30	Waiting for resending initialization string (originate mode)	The command started by a start internal relay was not completed and is waiting for retrial.
31	Waiting for resending ATZ (originate mode)	
32	Waiting for re-dialing	
33	Waiting for re-disconnecting line	
34	Waiting for resending AT command	
35	Waiting for resending initialization string (answer mode)	
36	Waiting for resending ATZ (answer mode)	
40	Line connected	Telephone line is connected. Only M8053 (disconnect line) can be turned on.
50	AT command completed successfully	Command started by M8054-M8056 is completed successfully.
60	AT command program error	Invalid character is included in the initialization string, dial number, or AT command string. Correct the program to include 0Dh in the AT command.
61	Simultaneous start of commands	Two or more start internal relays are on. Correct the user program so that only one start internal relay goes on at a time.
62	Invalid command in on-line mode	A start IR other than M8053 (disconnect line) is turned on while the telephone line is connected. Correct the program so that only the disconnect command is sent while the line is connected.
63	AT command execution error	Command failed in the first and all retry cycles.

Initialization String Commands

The built-in initialization string (see page 27-3) include the commands shown below. For details of modem commands, see the user's manual for your modem. When you make an optional initialization string, modify the initialization string to match your modem.

E0	Characters NOT echoed. The modem mode of the MicroSmart operates without echo back. Without the E0 command, the MicroSmart misunderstands an echo for a result code. An error will be caused although a command is executed correctly. This command must be included in the initialization string.
Q0	Result codes displayed. The modem mode of the MicroSmart is configured to use result codes. Without the Q0 command, a timeout error will be caused although a command is executed correctly. This command must be included in the initialization string.
V1	Word result code. The modem mode of the MicroSmart is configured to use word result codes. Without the V1 command, result codes are regarded as invalid and a timeout error will be caused although a command is executed correctly. This command must be included in the initialization string.
&D2	Hang up and disable auto-answer on DTR detection. When the DTR signal turns off, the telephone line is disconnected. The MicroSmart uses this function to disconnect the telephone line. This command must be included in the initialization string.
&C1	DCD ON with carrier from remote modem. DCD tracks the state of the data carrier from the remote modem. An ON condition of DCD indicates the presence of a carrier. This command must be included in the initialization string.
\V0	MNP result codes disabled. Conventional result codes are used and reliable link result codes are not used.
X4	Enables dial tone and busy detection.
&K3	Enables hardware flow control. The software flow control (XON/XOFF) cannot be used for the MicroSmart modem mode. This command must be included in the initialization string.
\A0	Set MNP maximum block size to 64 bytes.
\N5	MNP auto-reliable mode
S0=2	Ring to answer ON. Specifies the ring on which the modem will pick up the telephone line. S0=2 specifies that the modem answers an incoming call when detecting 2 ring calls. S0=0 disables the auto-answer function.
&W	Write active profile. The current configuration profile is saved to a non-volatile memory of the modem.

Preparation for Using Modem

Before using a modem, read the user's manual for your modem.

The required initialization string depends on the model and make of the modem. When the MicroSmart starts to run the user program, the default modem initialization strings is stored to D8145-D8169. See page 27-3.

Default Initialization String: ATE0Q0V1&D2&C1\V0X4&K3\A0\N5S0=2&W[CR][LF]

Programming Data Registers and Internal Relays

To enable the modem mode and communicate through the telephone line, the following settings are needed.

1. If the default initialization string does not match your modem, program a proper initialization string and enter the ASCII values to data registers starting with D8145 (initialization string). To send out the new initialization string, turn on internal relay M8050 (initialization string start IR) after the new values have been stored to the data registers.
2. Program to move 0 or 1 to data register D8103 (on-line mode protocol selection) to select maintenance protocol or user protocol for the RS232C port 2 after telephone line is connected.
3. Program the destination telephone number if dialing is required. Enter the ASCII values of the telephone number to data registers starting with D8170 (telephone number). Store two characters each in one data register. Enter 0Dh at the end of the telephone number. See page 27-4.
4. If you want to change the default value of 3 retry cycles, program to move a required value to data register D8109.
5. Include internal relays M8050-M8077 in the user program to control the modem communication as required.

Setting Up the CPU Module

1. Install the RS232C communication adapter (FC4A-PC1) to the port 2 connector on the all-in-one 16- or 24-I/O type CPU module. The 10-I/O type CPU module cannot be used for modem communication.

   When using any slim type CPU module, mount the RS232C communication module (FC4A-HPC1) next to the slim type CPU module, and use the port 2 on the RS232C communication module.

   When using the HMI base module with any slim type CPU module, install the RS232C communication adapter (FC4A-PC1) to the port 2 connector on the HMI base module.
2. Connect the MicroSmart CPU module port 2 to a modem using the modem cable 1C (FC2A-KM1C) as shown on page 27-1.

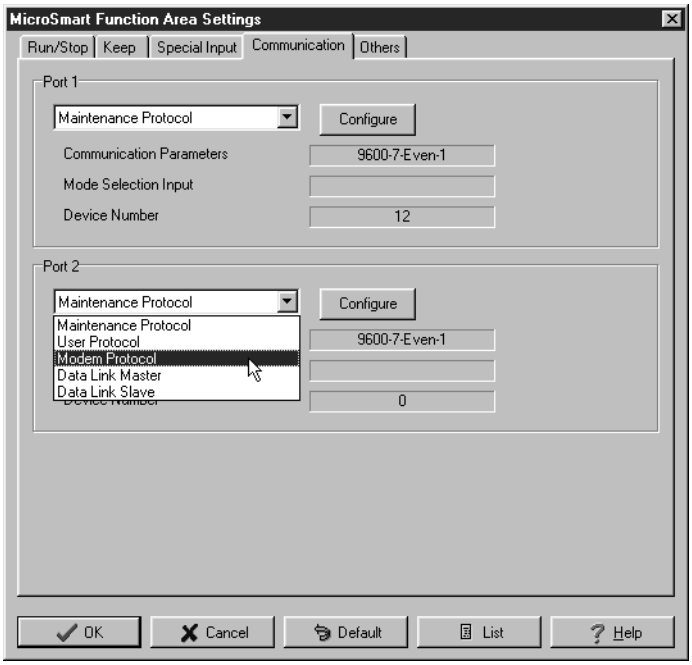
Programming WindLDR

The Communication page in the Function Area Settings must be programmed to enable the modem communication for port 2. If required, communication parameters of the CPU module port 2 can also be changed.

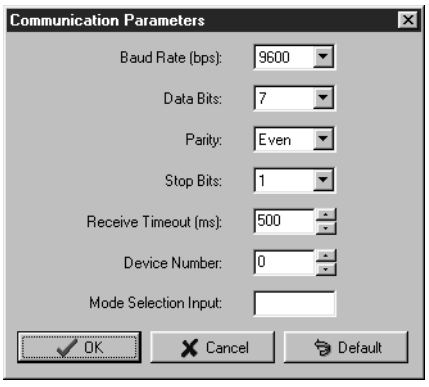
Since these settings relate to the user program, the user program must be downloaded to the MicroSmart after changing any of these settings.

1. From the WindLDR menu bar, select **Configure > Function Area Settings**. The Function Area Setting dialog box appears.

2. Click the **Communication** tab, and select **Modem Protocol** in the Port 2 pull-down list.



3. Click the **Configure** button. The Communication Parameters dialog box appears. Change settings, if required.



Baud Rate (bps)	1200, 2400, 4800, 9600, 19200
Data Bits	7 or 8
Parity	None, Odd, Even
Stop Bits	1 or 2
Receive Timeout (ms)	10 to 2540 (10-ms increments) (Receive timeout is disabled when 2550 is selected.)
Device Number	0 to 31

The default communication parameters shown below are recommended.

Baud rate	9600 bps
Start bit	1
Data bits	7
Parity	Even
Stop bit	1
Total	10 bits

Only when the modem connected on the communication line uses different communication parameters than the default values of the MicroSmart, set the matching communication parameters. Since the total of modem communication parameters is 10 bits, set the value to a total of 10 bits.

4. Click the **OK** button.

Operating Procedure for Modem Mode

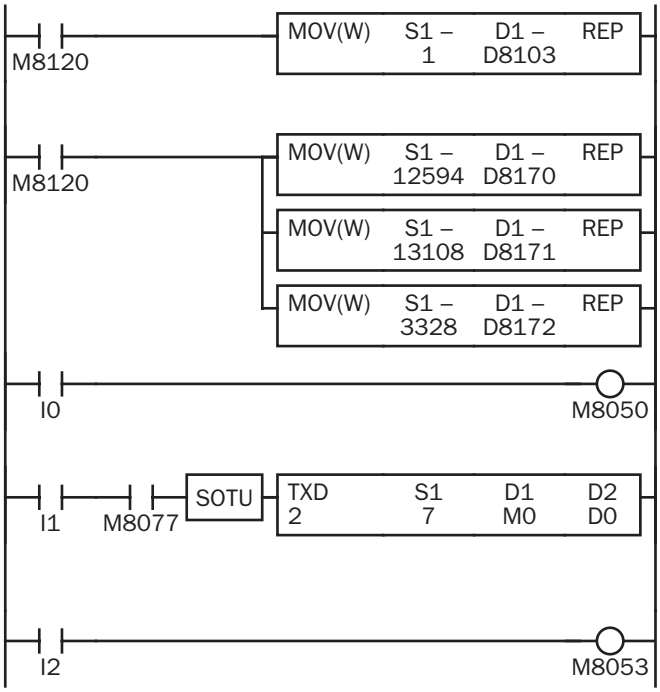
1. After completing the user program including the Function Area Settings, download the user program to the MicroSmart from a computer running WindLDR.
2. Start the MicroSmart to run the user program.
3. Turn on start internal relay M8050 or M8055 to initialize the modem.

   When originating the modem communication, turn on M8050 to send the initialization string, the ATZ command, and the dial command. If the initialization string has been stored in the non-volatile memory of the modem, turn on M8051 to start with the ATZ command followed by the dial command.

   When answering an incoming call, turn on M8055 to send the initialization string and the ATZ command. If the initialization string has been stored in the non-volatile memory of the modem, turn on M8056 to send the ATZ command only.
4. Transmit or receive communication through the modem.
5. Turn on start internal relay M8053 to disconnect the telephone line.

Sample Program for Modem Originate Mode

This program demonstrates a user program for the modem originate mode to move values to data registers assigned to the modem mode, initialize the modem, dial the telephone number, and disconnect the telephone line. While the telephone line is connected, user communication instruction TXD2 sends a character string "Connect."



M8120 is the initialize pulse special internal relay.

The MOV instruction stores 1 to D8103 to enable user protocol after telephone line is connected.

MOV instructions set a dial command ATD1234[CR][LF].

"12" (3132h = 12594) → D8170

"34" (3334h = 13108) → D8171

"CR" (0D00h = 3328) → D8172 to enter [CR] at the end of the telephone number.

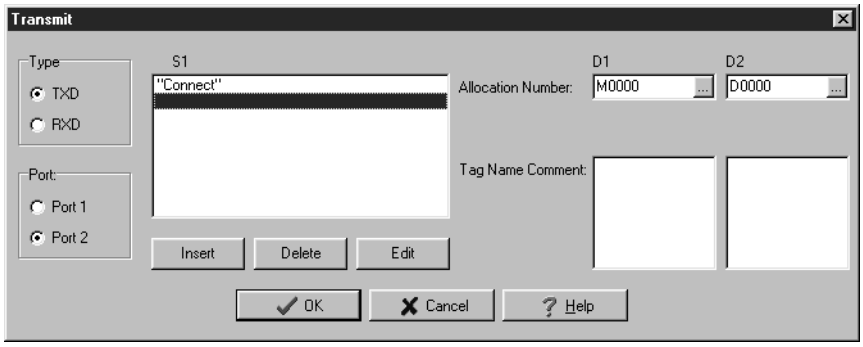
When input I0 is turned on, M8050 (initialization string) is turned on to send the initialization string, ATZ, and dial command to the modem.

M8077 (line connection status) is on while telephone line is connected.

When I1 is turned on, TXD2 sends seven characters "Connect." See the WindLDR dialog box shown below.

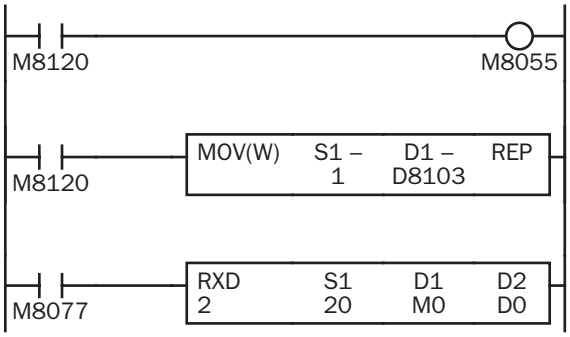
When input I2 is turned on, M8053 (disconnect line) is turned on to disconnect the telephone line.

The TXD2 instruction in the sample program for the modem originate mode is programmed using WindLDR with parameters shown below:



Sample Program for Modem Answer Mode

This program demonstrates a user program for the modem answer mode to move a value to a data register assigned to the modem mode and initialize the modem. While the telephone line is connected, user communication instruction RXD2 is executed to receive an incoming communication.



M8120 is the initialize pulse special internal relay.

When the MicroSmart starts to run, M8055 is turned on to send the initialization string for the modem answer mode.

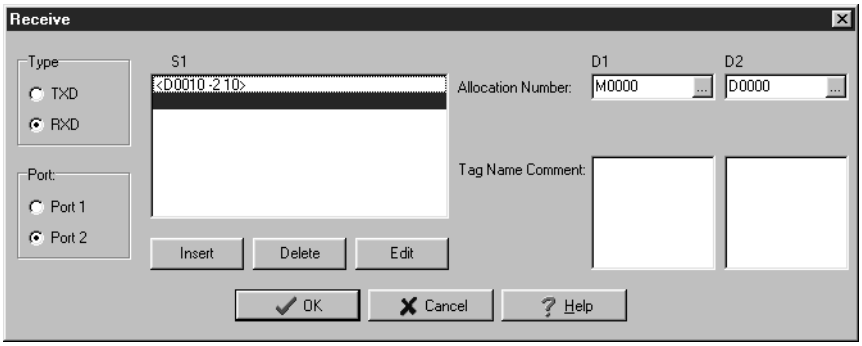
The MOV instruction stores 1 to D8103 to enable user protocol after telephone line is connected.

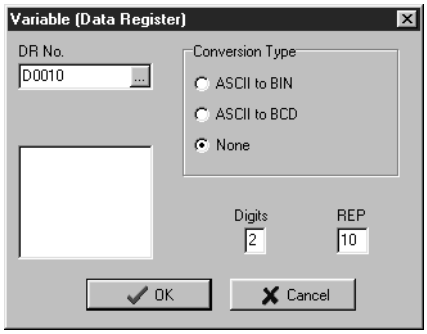
M8077 (line connection status) is on while telephone line is connected.

RXD2 receives incoming communication and stores received data to data registers starting with D10.

The RXD2 instruction is programmed using WindLDR with parameters shown below:

Source S1: Data register D10, No conversion, 2 digits, Repeat 10





Troubleshooting in Modem Communication

When a start internal relay is turned on, the data of D8111 (modem mode status) changes, but the modem does not work.

Cause: A wrong cable is used or wiring is incorrect.

Solution: Use the modem cable 1C (FC2A-KM1C).

The DTR or ER indicator on the modem does not turn on.

Cause: A wrong cable is used or wiring is incorrect.

Solution: Use the modem cable 1C (FC2A-KM1C).

When a start internal relay is turned on, the data of D8111 (modem mode status) does not change.

Cause: Modem protocol is not selected for port 2.

Solution: Select Modem Protocol for Port 2 using WindLDR (**Configure > Function Area Settings > Communication**) and download the user program to the CPU module.

When an initialization string is sent, a failure occurs, but sending ATZ completes successfully.

Cause: The initialization string is not valid for the modem.

Solution: Refer to the user's manual for the modem and correct the initialization string.

When a dial command is sent, a result code "NO DIALTONE" is returned and the telephone line is not connected.

Cause 1: The modular cable is not connected.

Solution 1: Connect the modular cable to the modem.

Cause 2: The modem is used in a PBX environment.

Solution 2: Add X0 or X3 to the initialization string stored in data registers D8145-D8169, and try initialization again.

Dialing completes successfully, but the telephone line is disconnected in a short period of time.

Cause 1: The modem settings at the both ends of the line are different.

Solution 1: Make the same settings for the modems at the both ends.

Cause 2: The model of the modems at the both ends of the line is different.

Solution 2: Use the same modems at the both ends.

Cause 3: The quality of the telephone line is low.

Solution 3: Decrease the baud rate of the MicroSmart to lower than 9600 bps.

28: AS-INTERFACE MASTER COMMUNICATION

Introduction

This chapter describes general information about the Actuator-Sensor-Interface, abbreviated AS-Interface, and detailed information about using the AS-Interface master module.

About AS-Interface

AS-Interface is a type of field bus that is primarily intended to be used to control sensors and actuators. AS-Interface is a network system that is compatible with the IEC62026 standard and is not proprietary to any one manufacturer. A master device can communicate with slave devices such as sensors, actuators, and remote I/Os, using digital and analog signals transmitted over the AS-Interface bus.

The AS-Interface system is comprised of the following three major components:

- One master, such as the MicroSmart AS-Interface master module (FC4A-AS62M)
- One or more slave devices, such as sensors, actuators, switches, and indicators
- Dedicated 30V DC AS-Interface power supply (26.5 to 31.6V DC)

These components are connected using a two-core cable for both data transmission and AS-Interface power supply. AS-Interface employs a simple yet efficient wiring system and features automatic slave address assignment function, while installation and maintenance are also very easy.

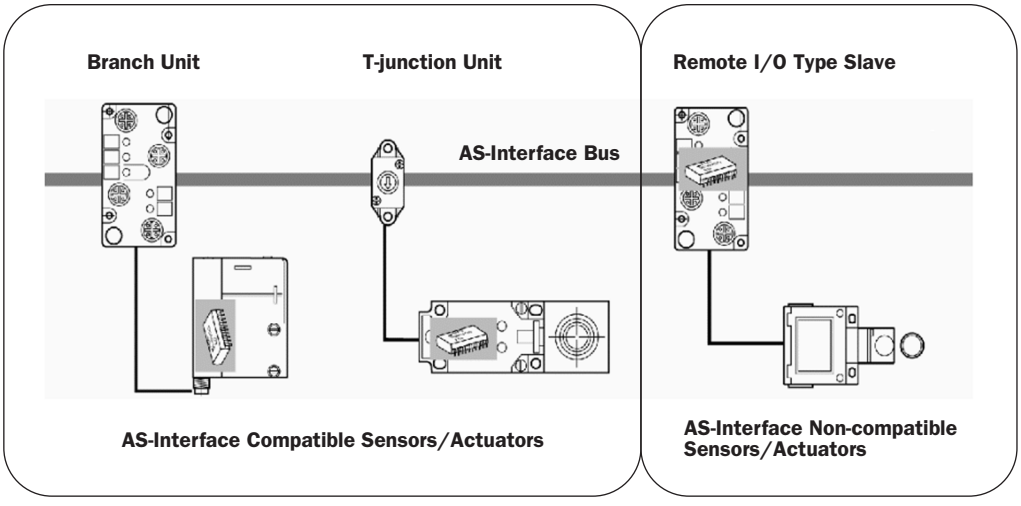
Applicable Sensors and Actuators for AS-Interface

AS-Interface Compatible Sensors and Actuators

AS-Interface compatible sensors and actuators communicate using the built-in AS-Interface function, and serve as AS-Interface slaves when connected directly to the AS-Interface bus via a branch unit or a T-junction unit.

Sensors/Actuators Not Compatible with AS-Interface

Conventional sensors and actuators that are not compatible with the AS-Interface can also be connected to the AS-Interface bus using a remote I/O slave and be handled in the same way as devices that are compatible with the AS-Interface.



Maximum I/O points when using AS-Interface master module

AS-Interface Master Module	1 module
Maximum Slaves	62 slaves
Maximum I/O Points	434 (248 inputs / 186 outputs)
Maximum Communication Distance	Without repeater: 100m With 2 repeaters: 300m

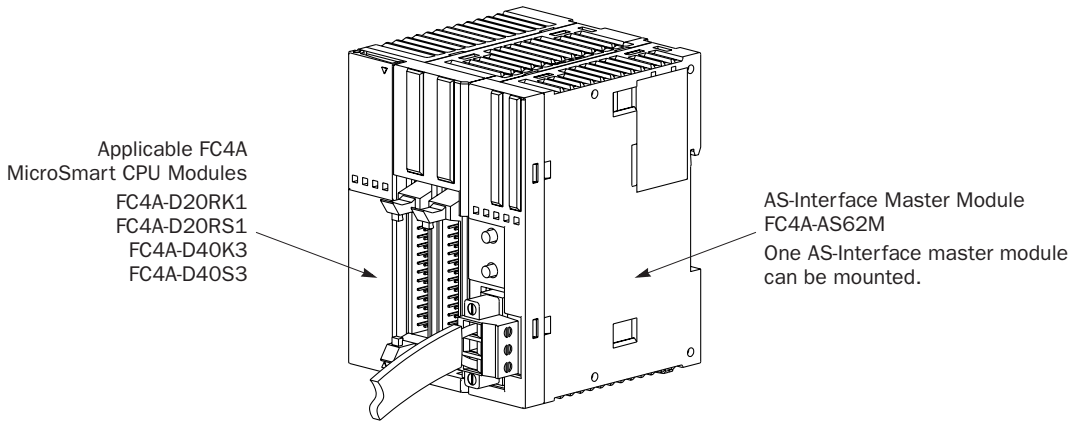
AS-Interface System Requirements

Master

The AS-Interface master controls and monitors the status of slave devices connected to the AS-Interface bus.

Normally, the AS-Interface master is connected to a PLC (sometimes called 'host') or a gateway. For example, the MicroSmart AS-Interface master module is connected to the MicroSmart CPU module.

The FC4A MicroSmart CPU module can be used with one AS-Interface master module to set up one AS-Interface network.



The AS-Interface master module can connect a maximum of 62 digital I/O slaves. A maximum of seven analog I/O slaves can also be connected to the AS-Interface master module (compliant with AS-Interface ver. 2.1 and analog slave profile 7.3).

⚠ Caution

- The AS-Interface master module cannot be connected to the all-in-one 10-I/O and 16-I/O type CPU modules.
- One AS-Interface master module can be connected to the FC4A MicroSmart CPU module. If more than one AS-Interface master module is connected, an error occurs and special data register D8037 (quantity of expansion I/O modules) stores error code 40 (hex).
- Normally, a maximum of four expansion I/O modules can be connected to the all-in-one 24-I/O type CPU module. But when an AS-Interface master module is connected, only a total of three expansion modules can be connected, including the AS-Interface master module. Do not connect more than three expansion modules due to the amount of heat generated. If more than three expansion modules, including the AS-Interface master module, are connected, an error occurs and special data register D8037 (quantity of expansion I/O modules) stores error code 20 (hex).
- Similarly, slim type CPU modules can normally connect a maximum of seven expansion I/O modules, but can connect a maximum of six expansion modules including one AS-Interface master module. If more than six expansion modules, including the AS-Interface master module, are connected, an error occurs and special data register D8037 (quantity of expansion I/O modules) stores error code 20 (hex).
- The AS-Interface master module can connect a maximum of seven analog I/O slaves. When more than seven analog I/O slaves are connected, the AS-Interface system will not operate correctly.

Slaves

Various types of slave devices can be connected to the AS-Interface bus, including sensors, actuators, and remote I/O devices. Analog slaves can also be connected to process analog data.

Slaves are available in standard slaves and A/B slaves. Standard slaves have an address of 1 trough 31 in the standard address range. A/B slaves have an address of 1A through 31A in the standard address range or 1B through 31B in the expanded address range. Among the A/B slaves, slaves with an address of 1A through 31A are called A slaves, and slaves with an address of 1B through 31B are called B slaves.

AS-Interface Power Supply

The AS-Interface bus uses a dedicated 30V DC power supply (AS-Interface power supply), which is indicated with the AS-Interface mark. General-purpose power supply units cannot be used for the AS-Interface bus.

AS-Interface Marks

Caution • Use a VLSV (very low safety voltage) to power the AS-Interface bus. The normal output voltage of the AS-Interface power supply is 30V DC.

Recommended IDEC AS-Interface Power Supplies

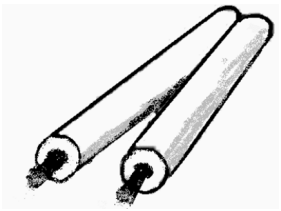
Input Voltage	Output Voltage	Output Wattage	Type No.
100 to 240V AC	30.5V DC	73W	PS2R-Q30ABL
		145W	PS2R-F30ABL

Cables

The AS-Interface bus uses only one cable to transmit signals and power. Use one of the following cable types (the wire does not have to be stranded).

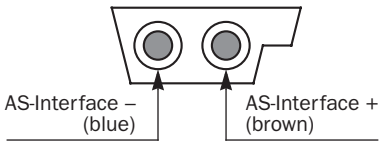
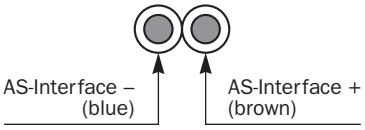
- Standard yellow unshielded AS-Interface cable (with polarity)
- Ordinary two-wire flat cable

AS-Interface Cable



Two-wire Flat Cable

Applicable Cable Specifications

Cable Type	Cable Size/Manufacturer	Cross-sectional View
AS-Interface Standard Cable	Cable sheath color: Yellow Conductor cross section: 1.5 mm^2 LAPP's Cables Type No: 2170228 (sheath material EPDM) Type No: 2170230 (sheath material TPE)	
2-wire Flat Cable or **Single Wires** (See Note)	Conductor cross section Stranded wire: 0.5 to 1.0 mm^2 Solid wire: 0.75 to 1.5 mm^2 AWG: 20 to 16	

Note: When using single wires, the maximum cable length is 200 mm. See "Maximum Communication Distance" on page 28-1.

Main Features of AS-Interface V2 with Slave Expansion Capability

The AS-Interface is a reliable bus management system in which one master periodically monitors each slave device connected on the AS-Interface bus in sequence. The master manages the I/O data, parameters, and identification codes of each slave in addition to slave addresses. The management data depends on the type of the slave as follows:

Standard Slaves

- A maximum of four inputs and four outputs for each slave
- Four parameters for setting a slave's operation mode (P3, P2, P1, P0)
- Four identification codes (ID code, I/O code, ID2 code, and ID1 code)

A/B Slaves

- A maximum of four inputs and three outputs for each slave
- Three parameters for setting a slave's operation mode (P2, P1, P0)
- Four identification codes (ID code, I/O code, ID2 code, and ID1 code)

Note 1: Parameters P3 through P0 are used to set an operation mode of the slave. For details, see the user's manual for the slave.

Note 2: The slaves connected to the AS-Interface bus are distinguished from each other by the ID code and I/O code contained in each slave. Some slaves have ID2 code and ID1 code to indicate the internal functions of the slave. For example, analog slaves use the ID2 code to represent the channel number of the slave.

Note 3: The MicroSmart AS-Interface master module is also compatible with AS-Interface ver. 2.1 and earlier slaves.

Slave Addresses

Each standard slave connected to the AS-Interface bus can be allocated an address of 1 through 31. Each A/B slave can be allocated an address of 1A through 31A or 1B through 31B. All slaves are set to address 0 at factory before shipment. The address of a slave can be changed using the "addressing tool." Using WindLDR, the addresses of slaves connected to the AS-Interface master module can also be changed (see page 28-31).

When a slave fails during operation and needs to be replaced, if the auto addressing function is enabled on the master module, just replace the slave with a new one (with address 0 and the same identification codes). The new slave will automatically be allocated the same address as the slave that was removed, and you do not have to set the address again. For details of the ASI command to enable auto addressing, see page 28-28.

Slave Identification

Slaves have the following four identification codes. The master checks the identification codes to determine the type and feature of the slave connected on the AS-Interface bus.

ID Code

The ID code consists of 4 bits to indicate the type of the slave, such as sensor, actuator, standard slave, or A/B slave. For example, the ID code for a standard remote I/O is 0, and that for an A/B slave is A (hex).

I/O Code

The I/O code consists of 4 bits to indicate the quantity and allocation of I/O points on a slave.

I/O Code	Allocation	I/O Code	Allocation	I/O Code	Allocation	I/O Code	Allocation
0h	I, I, I, I	4h	I, I, B, B	8h	O, O, O, O	Ch	O, O, B, B
1h	I, I, I, O	5h	I, O, O, O	9h	O, O, O, I	Dh	O, I, I, I
2h	I, I, I, B	6h	I, B, B, B	Ah	O, O, O, B	Eh	O, B, B, B
3h	I, I, O, O	7h	B, B, B, B	Bh	O, O, I, I	Fh	(reserved)

I: input, O: output, B: input and output

ID2 Code

The ID2 code consists of 4 bits to indicate the internal function of the slave.

ID1 Code

The ID1 code consists of 4 bits to indicate additional identification of the slave. Standard slaves can have an ID1 code of 0000 through 1111 (bin). A/B slaves use the MSB to indicate A or B slave, and can have a unique value only for the lower three bits. The MSB of A slaves is set to 0, and that of B slaves is set to 1.

Quantities of Slaves and I/O Points

The quantity of slaves that can be connected to one AS-Interface master module is as follows.

- Standard slaves: 31 maximum
- A/B slaves: 62 maximum

The limits for slave quantities given above apply when the slaves are either all standard slaves or are all A/B slaves.

When 62 A/B slaves (with four inputs and three outputs) are connected, a maximum of 434 I/O points (248 inputs and 186 outputs) can be controlled by one AS-Interface master module.

When using a mix of standard slaves and A/B slaves together, the standard slaves can only use addresses 1(A) through 31(A). Also, when a standard slave takes a certain address, the B address of the same number cannot be used for A/B slaves.

AS-Interface Bus Topology and Maximum Length

The AS-Interface bus topology is flexible, and you can wire the bus freely according to your requirements.

When repeaters or extenders are not used, the bus length can be 100m (328 feet) at the maximum.

The FC4A-AS62M AS-Interface master module can use two repeaters to extend the bus length to 300m.

AS-Interface Bus Cycle Time

The AS-Interface bus cycle time is the amount of time required for a master to cycle through every slave on the bus.

The information for each slave is continuously transmitted over the bus in sequence, so the AS-Interface bus cycle time depends on the quantity of active slaves.

- When up to 19 slaves are active, the bus cycle time is 3 ms.
- When 20 to 62 slaves are active, the bus cycle time is 0.156 × (1+N) ms where N is the number of slaves.

When A slave and B slave have the same address number (e.g. 12A and 12B), the two slaves are alternately updated each cycle. Therefore, when the system consists of 31 A slaves and 31 B slaves, then the AS-Interface bus cycle time will be 10 ms.

Maximum AS-Interface Bus Cycle Time

- When 31 slaves are connected, the maximum bus cycle time is 5 ms.
- When 62 slaves are connected, the maximum bus cycle time is 10 ms.

High Reliability and Security

The AS-Interface employs a transfer process of high reliability and high security. The master monitors the AS-Interface power supply voltage and data transmitted on the bus, and detects slave failures and data errors.

Even when a slave is replaced or a new slave is added during operation, the AS-Interface master module need not be shut down and can continue uninterrupted communication with other active slaves on the bus.

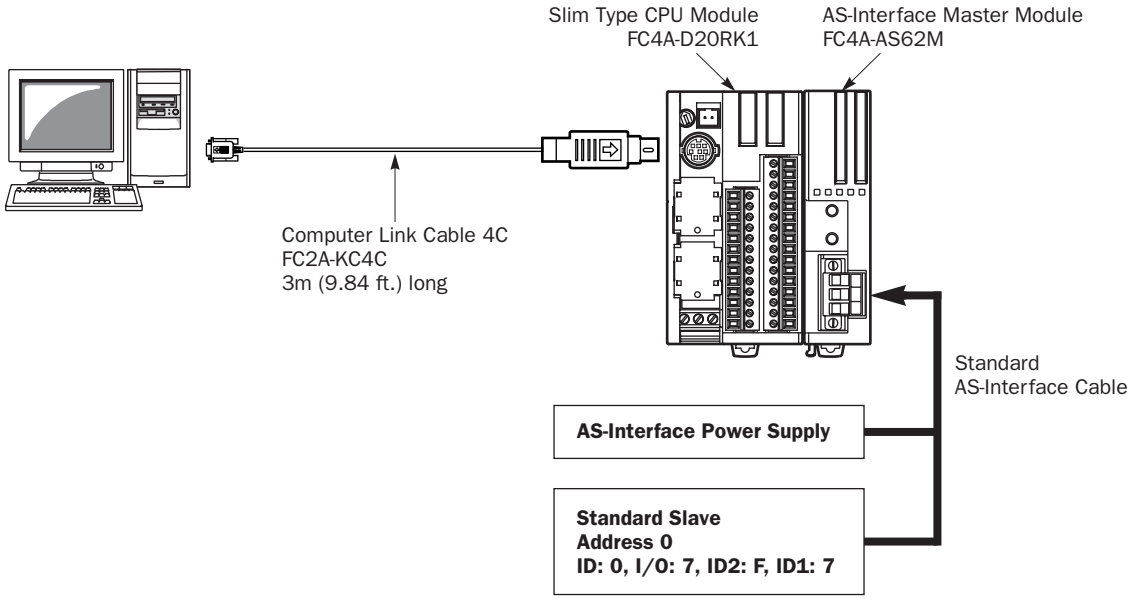
Operation Basics

This section describes simple operating procedures for the basic AS-Interface system from programming WindLDR on a computer to monitoring the slave operation.

AS-Interface System Setup

The sample AS-Interface system consists of the following devices:

Name	Type No.	Description
FC4A MicroSmart Slim Type CPU Module	FC4A-D20RK1	—
MicroSmart AS-Interface Master Module	FC4A-AS62M	—
WindLDR	FC9Y-LP2CDW	Version 5.0 or higher
AS-Interface Standard Slave	—	1 unit Address 0 ID: 0, I/O: 7, ID2: F, ID1: 7
AS-Interface Power Supply	PS2R-Q30ABL	Output 30.5V DC, 2.4A (73W)

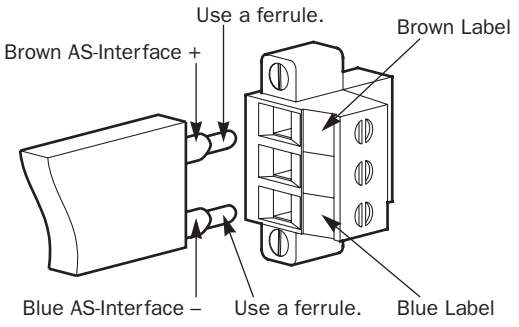


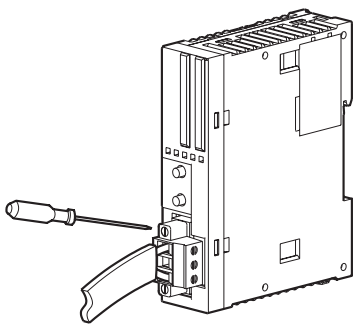
AS-Interface Cable Wiring

Before wiring the AS-Interface cable, remove the AS-Interface cable terminal block from the AS-Interface cable connector on the AS-Interface master module.

AS-Interface specifies use of brown cables for the AS-Interface + line, and blue cables for the AS-Interface – line. Connect the cables to match the color labels on the terminal block. Tighten the terminal screws to a torque of 0.5 to 0.6 N·m.

Insert the terminal block to the connector on the AS-Interface master module, and tighten the mounting screws to a torque of 0.3 to 0.5 N·m.





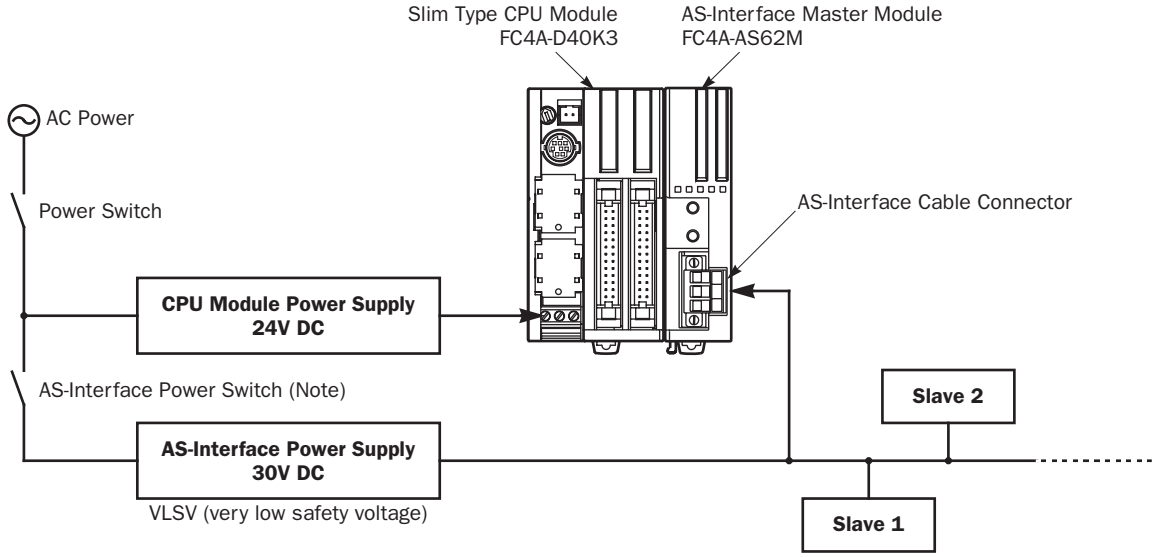
Power Supply

Caution

- When turning off the power to the CPU module, also turn off the AS-Interface power supply. If the CPU module is powered down and up while the AS-Interface power remains on, AS-Interface communication may stop due to a configuration error, resulting in a communication error.
- Turn on the AS-Interface power supply no later than the CPU module power supply, except when slave address 0 exists on the network. The two power supplies may be turned off in any order.
- Immediately after power-up, the CPU module cannot access slave I/O data in the AS-Interface master module. Make the user program so that slave I/O data are accessed after special internal relay M1945 (Normal_Operation_Active) has turned on. See page 28-24.

Power Supply Wiring Diagram

A recommended power supply wiring diagram is shown below. Use a common power switch for both the CPU module power supply and AS-Interface power supply to make sure that both power supplies are turned on and off at the same time.



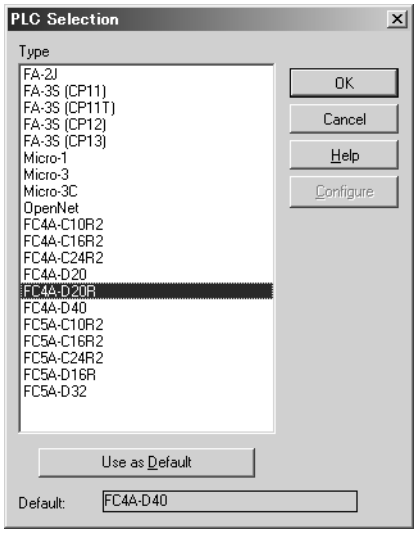
Note: A failed slave can be replaced with a new slave with address 0 without turning off the power to the CPU module and the AS-Interface line. But, if power has been turned off before replacing the slaves, install a new slave with address 0 and take one of the following steps, because the AS-Interface master module has to be initialized to enable communication.

- Disconnect the AS-Interface cable connector and turn on both power supplies. Five seconds later, connect the AS-Interface cable connector.
- Turn on the CPU module power supply first. Five seconds later, turn on the AS-Interface power supply.

Selecting the PLC Type

Start WindLDR on a computer.

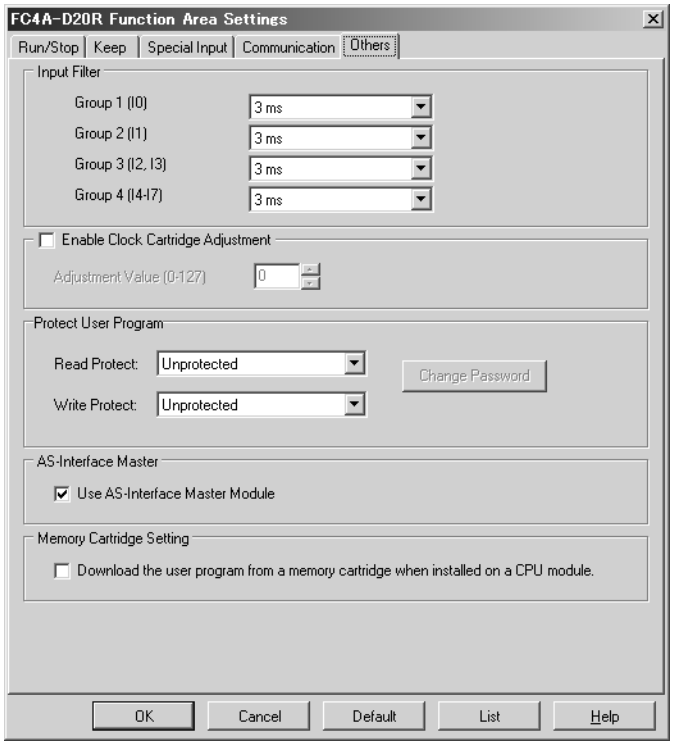
1. From the WindLDR menu bar, select **Configure** > **PLC Selection**. The PLC Selection dialog box appears.
2. Select **FC4A-D20R**.
3. Click **OK** to save changes and return to the ladder editing screen.



Function Area Settings

Use of the AS-Interface master module must be selected in the Function Area Settings dialog box.

1. From the WindLDR menu bar, select **Configure** > **Function Area Settings**. The Function Area Settings dialog box appears.
2. Select the **Others** tab.



3. Make sure of a check mark in the check box on the left of **Use AS-Interface Master Module**.

This check box is checked as default. Since this setting relates to the user program, download the user program to the CPU module after changing any of these settings.

If the ERR LED on the CPU module goes on when the AS-Interface master module is connected, download the user program to the CPU module after making the above setting.

Assigning a Slave Address

AS-Interface compatible slave devices are set to address 0 at factory. Connect the slave to the AS-Interface master module as shown on page 28-6. Do not connect two or more slaves with slave address 0, otherwise the AS-Interface master module cannot recognize slave addresses correctly.

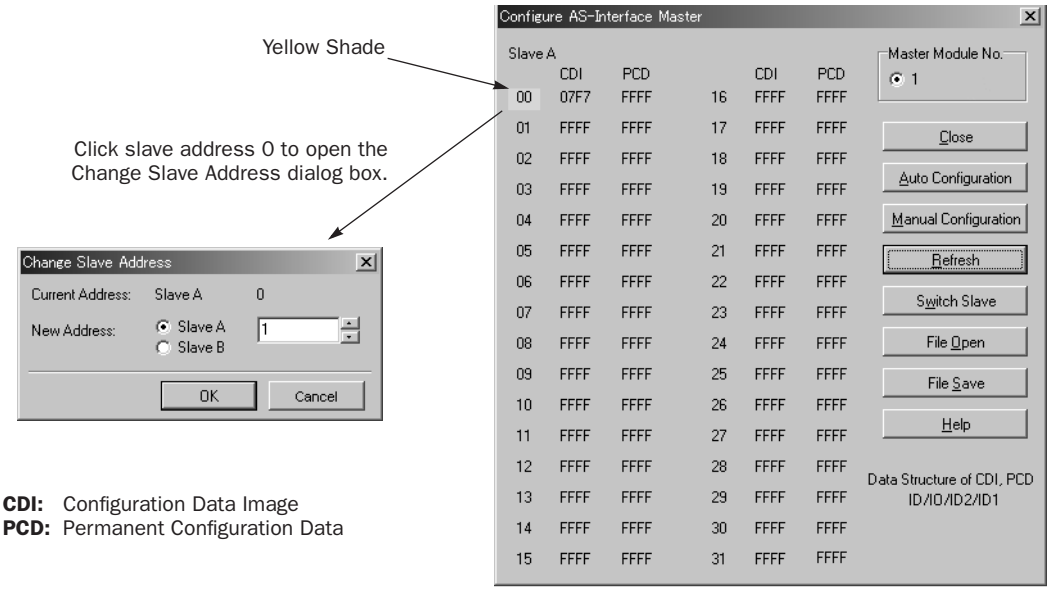
1. Power up the MicroSmart CPU module first. Approximately 5 seconds later, turn on the AS-Interface power supply.

Note: When slave address 0 is not mounted on the AS-Interface bus, the CPU module power supply and the AS-Interface power supply can be turned on at the same time. See page 28-7.

2. From the WindLDR menu bar, select **Configure** > **AS-Interface Master** to open the Configure AS-Interface Master dialog box. Press **Refresh** to collect slave information and update the screen display. (When configuration in the master module is complete, you do not have to press **Refresh** since the screen display is updated automatically.)

On the Configure AS-Interface Master dialog box, slave address 0 is shaded with yellow. This means that the master module has found slave address 0 on the AS-Interface bus. The CDI for address 0 shows 07F7 (ID: 0, I/O: 7, ID2: F, ID1: 7).

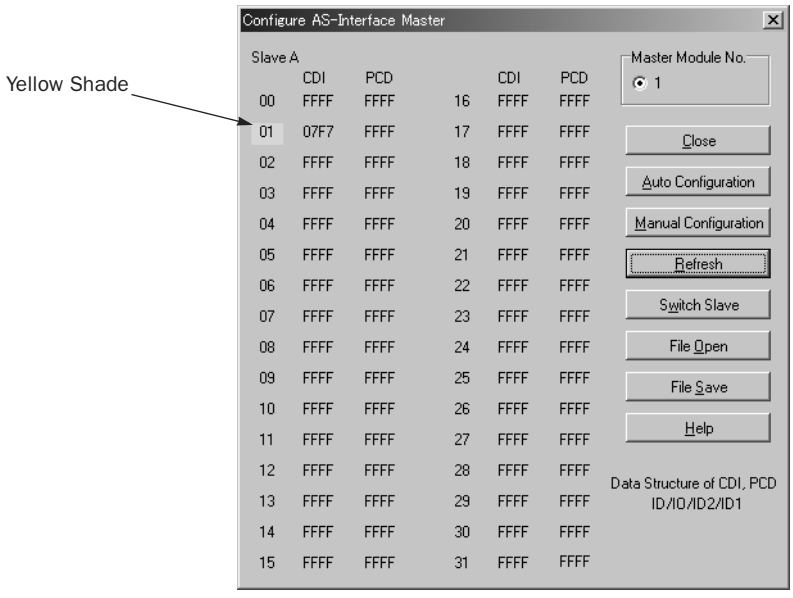
3. Click the slave address "00" to open the Change Slave Address dialog box for slave 0. To assign slave address 1 to the slave, enter **1** in the New Address field and click **OK**.



CDI: Configuration Data Image
PCD: Permanent Configuration Data

The new address "01" is shaded with yellow to indicate that the address assignment is complete.

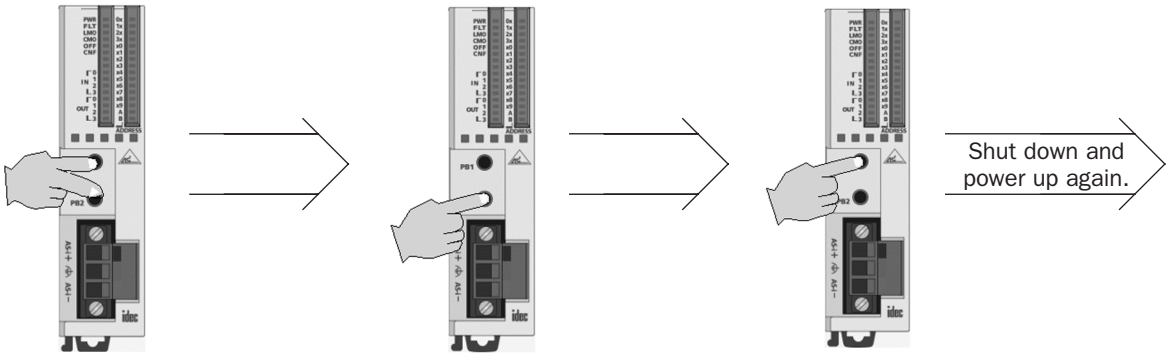
4. When changing slave addresses on other slaves, continue from step 3 if it is possible to wire the slave without turning off power, or from step 1 if the CPU module is shut down.



Configuring a Slave

Next, you have to set the slave configuration in the AS-Interface master module, either by using pushbuttons PB1 and PB2 on the AS-Interface master module or WindLDR.

Configuration Using Pushbuttons PB1 and PB2



Press PB1 and PB2. Press PB2. Press PB1.

1. Check that PWR LED and CMO LED on the AS-Interface master module are on (normal protected mode).
2. Press pushbuttons PB1 and PB2 together for 3 seconds. CMO LED turns off and LMO LED turns on (protected mode).
3. Press pushbutton PB2 for 3 seconds. CNF LED flashes (configuration mode).
4. About 5 seconds later, press pushbutton PB1 for 3 seconds. All I/O LEDs blink once to complete configuration.
5. Shut down the CPU module and AS-Interface master module, and power up again. Check that FLT LED is off, which indicates that configuration is complete.
6. Use WindLDR to view slave information on the Configure AS-Interface Master dialog box and check that all slaves are recognized correctly.

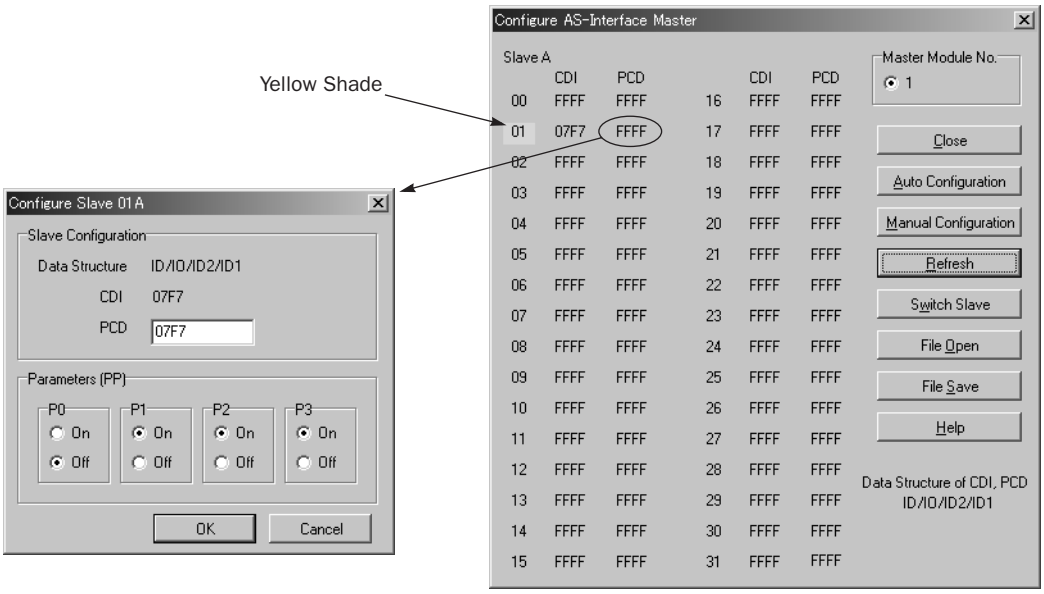
Configuration Using WindLDR

Slave configuration can be set using WindLDR in two ways; using the **Auto Configuration** or **Manual Configuration** button on the Configure AS-Interface Master dialog box.

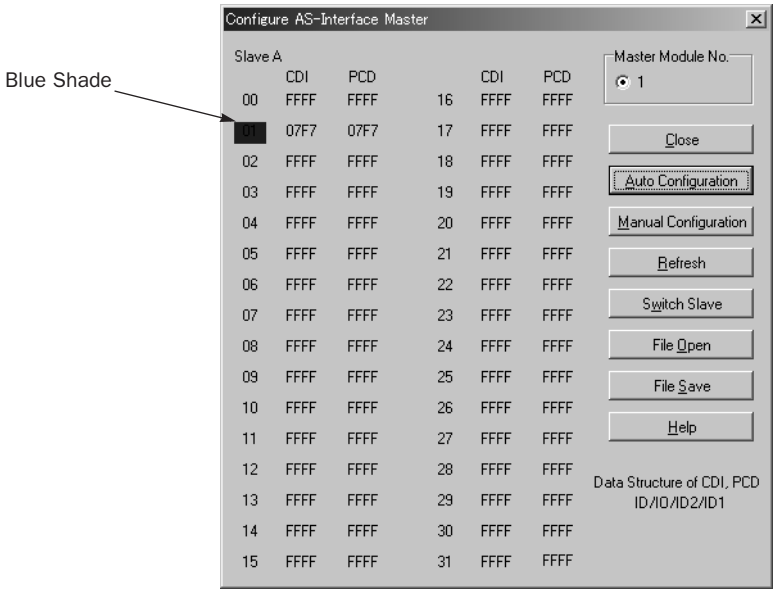
1. Click the **Auto Configuration** button to store the configuration information (LDS, CDI, PI) of the connected slaves to the EEPROM (LPS, PCD, PP) in the AS-Interface master module. For details, see page 28-32.

The auto configuration automatically stores the information of slaves found on the AS-Interface bus to the EEPROM in the master module, and this completes configuration. Another method of configuration is manual configuration as follows.

2. Click the PCD value "FFFF" of slave address 01 to open the Configure Slave 01A dialog box.
3. Enter the same value as CDI "07F7" in the PCD field. (Set FFFF to PCD values of all unused slaves.)
4. Select initial settings of parameters (PP) P0 through P3, if required.



5. Click the **Manual Configuration** button to store the selected PCD and parameter values to the master module.
6. Check that the blue shade appears at slave address 01. Now, configuration is complete.



Monitoring Digital I/O, and Changing Output Status and Parameters

While the MicroSmart is communicating with AS-Interface slaves through the AS-Interface bus, operating status of AS-Interface slaves can be monitored using WindLDR on a computer. Output statuses and parameter image (PI) of slaves connected to the AS-Interface master module can also be changed using WindLDR.

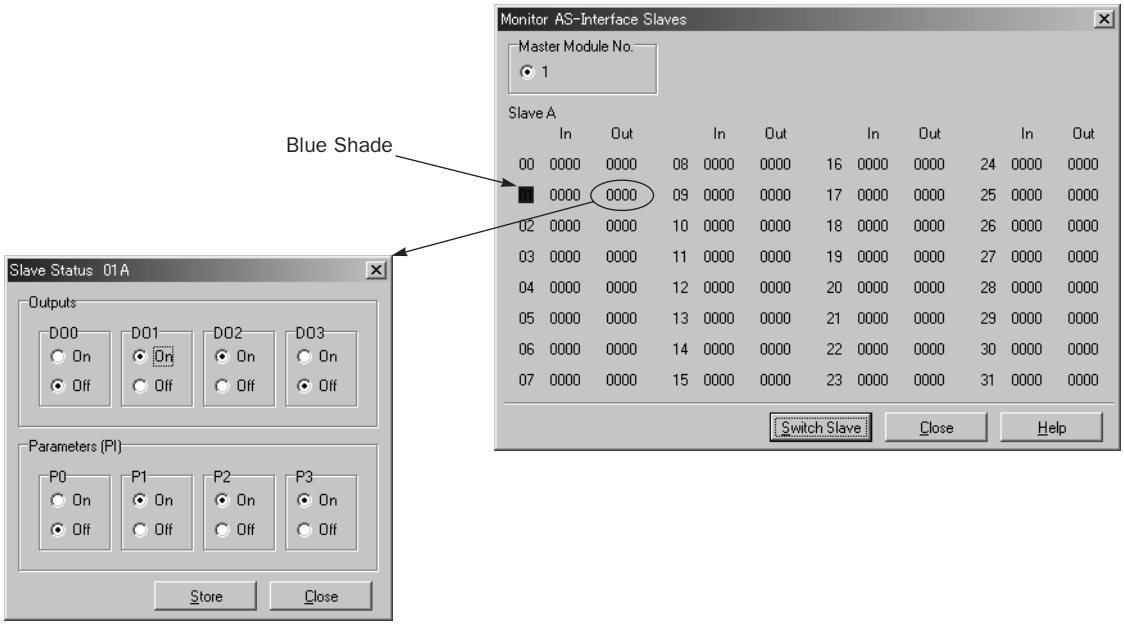
1. From the WindLDR menu bar, select **Online** > **Monitor**. From the WindLDR menu bar, select **Online**, and select **Monitor AS-Interface Slaves** in the pull-down menu. The Monitor AS-Interface Slaves dialog box appears.

Active slaves are indicated with blue shade.

Next step is to change output status of the active slave.

2. Click the output of slave address 01 to open the Slave Status 01A dialog box.

3. Click the On or Off button to change the statuses of outputs 00 through 03 and parameters (PI) P0 through P3 as required.



The selected parameters (PI) are in effect until the CPU module is shut down. When the CPU module is powered up again, the parameter values (PP) selected in the slave configuration procedure (page 28-10) will take effect. To store the changed parameter values to the AS-Interface master module EEPROM, execute the Copy PI to PP command by storing 0306, 0100, 0000, 0000, 0001 to data registers D1941 through D1945. See page 28-28.

Troubles at System Start-up

The following table summarizes possible troubles at system start-up, probable causes and actions to be taken.

Trouble	Cause and Action
PWR LED is off. (power)	• AS-Interface power is not supplied to the AS-Interface master module. Check that wiring is correct and AS-Interface power is supplied. • Power is not supplied from the CPU module to the AS-Interface master module. Check the connection between the CPU module and the AS-Interface master module.
FLT LED is on. (fault)	• Slave configuration on the bus is incorrect. Use the WindLDR slave monitor function to check that slaves are connected correctly. Perform configuration, if necessary. For the configuration method, see page 28-30. If FLT LED remains on even though slaves are connected correctly and configuration is completed, either disconnect and reconnect the AS-Interface connector, or turn off and on the AS-Interface power supply.
LMO LED is on. (local mode)	The CPU module fails to communicate with the AS-Interface master module. Check the following points. • Is the CPU module compatible with AS-Interface? Check the Type No. of the CPU module. • Is a check mark put in the check box “Use AS-Interface Master Module” in WindLDR Function Area Settings? The box is checked as default. If not, put a check mark and download the user program to the CPU module.
OFF LED is on. (offline)	• While a slave of address 0 was connected, power was turned on. After changing the slave address, power up again. For the address changing method, see page 28-31.
Slave operation is unstable.	• Check if there are two or more slaves with the same address. Each slave must have a unique address. If two slaves have the same address and same identification codes (ID, I/O, ID2, ID1), the AS-Interface master module may fail to detect an error. When changing the duplicate slave address using WindLDR, remove one of the slaves from the bus.

Pushbuttons and LED Indicators

This section describes the operation of pushbuttons PB1 and PB2 on the AS-Interface master module to change operation modes, and also explains the functions of address and I/O LED indicators.

Pushbutton Operation

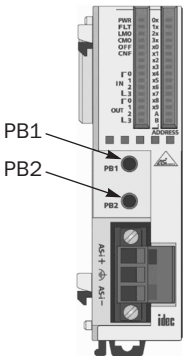
The operations performed by pushbuttons PB1 and PB2 on the front of the AS-Interface master module depend on the duration of being pressed. A "long press" switches the operation mode, and a "short press" switches the slave being monitored on the I/O LEDs. If the duration of pressing PB1 or PB2 does not correspond to either of these, the status of the AS-Interface master module does not change.

Long Press

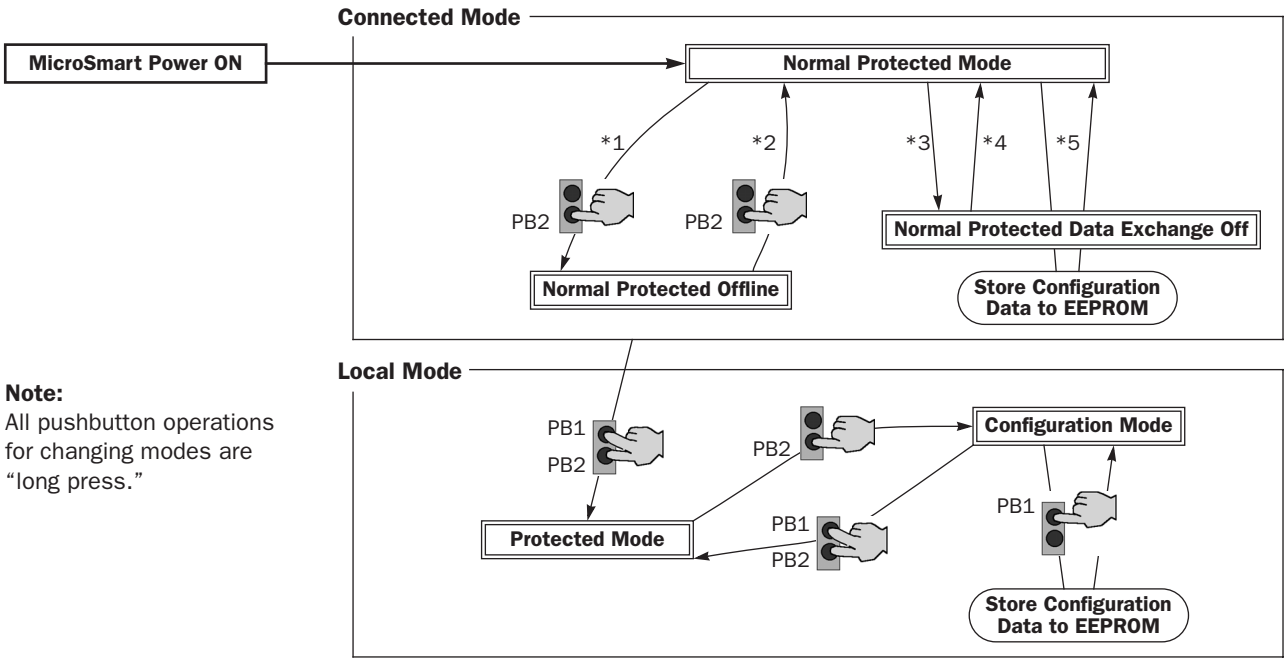
A "long press" takes effect when you press either pushbutton PB1 or PB2 or both for 3 seconds or more. Use the long press to change the operation mode of the AS-Interface master module or to save the configuration data to the AS-Interface master module EEPROM.

Short Press

A "short press" takes effect when you press either pushbutton PB1 or PB2 for 0.5 second or less. Use the short press to change the slave address when monitoring slave I/O status on the AS-Interface master module LED indicators.



Transition of AS-Interface Master Module Modes Using Pushbuttons



Note:
All pushbutton operations for changing modes are "long press."

*1 Pushbutton operation or execution of the ASI command Go to Normal Protected Offline.

*2 Pushbutton operation or execution of the ASI command Go to Normal Protected Mode.

*3 Execution of the ASI command Prohibit Data Exchange.

*4 Execution of the ASI command Enable Data Exchange.

*5 Configuration is done by clicking the Auto Configuration or Manual Configuration button in WindLDR. The configuration data is saved to the AS-Interface master module EEPROM.

AS-Interface Master Module Operation Modes

The AS-Interface master module has two modes of operation: connected mode is used for actual operation, and local mode is used for maintenance purposes.

Connected Mode

In connected mode, the CPU module communicates with the AS-Interface master module to monitor and control each slave. Connected mode is comprised of the following three modes.

- **Normal Protected Mode**

When the CPU module is powered up, the AS-Interface master module initially enters normal protected mode of connected mode if no error occurs. This is the normal operation mode for the AS-Interface master module to perform data communication with the connected slaves.

If the configuration data stored in the AS-Interface master module do not match the currently connected slave configuration, the FLT LED on the front of the AS-Interface master module goes on. Execute configuration using the pushbuttons on the AS-Interface master module. Configuration can also be done using WindLDR. See page 28-32.

- **Normal Protected Offline**

The AS-Interface master module stops communication with all slaves and enables offline operation (initialization of the master module). In this mode, the CPU module cannot monitor the slave status.

To enter normal protected offline from normal protected mode, either long-press the PB2 button or execute the ASI command Go to Normal Protected Offline. To return to normal protected mode and resume data communication, either long-press the PB2 button again or execute the ASI command Go to Normal Protected Mode. For details about the ASI commands, see page 28-28.

- **Normal Protected Data Exchange Off**

Data communication with all slaves is prohibited. To enter this mode, execute the ASI command Prohibit Data Exchange. To return to normal protected mode and resume data communication, execute the ASI command Enable Data Exchange. For details about the ASI commands, see page 28-28.

When auto configuration or manual configuration is executed on WindLDR, the AS-Interface master module enters this mode during configuration.

Local Mode

In local mode, the CPU module does not communicate with the AS-Interface master module. Local mode is used to carry out maintenance operations such as checking the configuration and slave inputs. Use the input LEDs to check the slave input data during operation.

When the CPU module is powered up, the AS-Interface master module initially enters normal protected mode of connected mode if no error occurs. To switch from any of connected mode to local mode (protected mode), long-press the PB1 and PB2 buttons simultaneously. It is not possible to switch from local mode back to connected mode using the pushbuttons. To return to connected mode, shut down the CPU module and power up again.

Local mode is comprised of two modes: protected mode and configuration mode.

- **Protected Mode**

This mode operates the slaves in accordance with the slave configuration data stored in the AS-Interface master module. If the configuration data stored in the AS-Interface master module does not match the currently connected slave configuration, the FLT LED on the front of the AS-Interface master module goes on, and slaves are not operated correctly.

To enter protected mode from any of connected mode, long-press the PB1 and PB2 buttons simultaneously.

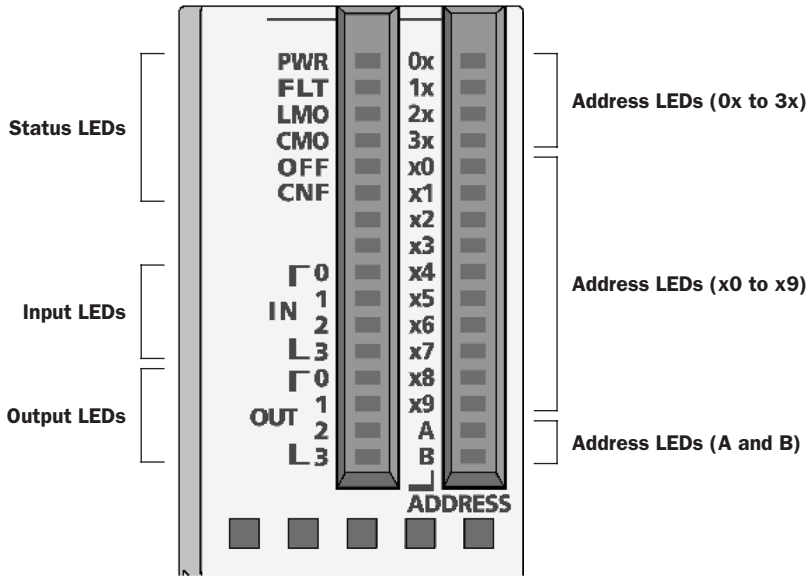
- **Configuration Mode**

This mode switches all currently connected slaves to active, regardless of the slave configuration data stored in the AS-Interface master module. To store the current slave configuration data to the AS-Interface master module EEPROM, long press the PB1 button. This way, configuration is executed.

To enter configuration mode from protected mode, long-press the PB2 button. To return to protected mode, long-press the PB1 and PB2 buttons simultaneously.

LED Indicators

The LED indicators on the AS-Interface master module consist of status LEDs, I/O LEDs, and address LEDs.



LED Indicators		Description
Status LEDs	**PWR** (AS-Interface power supply)	Indicates the status of the AS-Interface power supply for the AS-Interface master module. Goes on when the AS-Interface power is supplied sufficiently.
	FLT (Fault)	Indicates the AS-Interface configuration status. Goes on when the permanent configuration data (PCD) stored in the AS-Interface master module EEPROM does not match the current slave configuration, or configuration data image (CDI). Then, configuration is not complete or an error was found on the AS-Interface bus.
	LMO (Local mode)	Indicates the mode of the AS-Interface master module. Goes on when the AS-Interface master module is in local mode. Goes off when the AS-Interface master module is in connected mode.
	CMO (Connected mode)	Indicates the mode of the AS-Interface master module. Goes on when the AS-Interface master module is in connected mode. Goes off when the AS-Interface master module is in local mode.
	OFF (Offline)	Indicates the operating status of the AS-Interface master module. Goes on when the AS-Interface master module is in normal protected offline.
	CNF (Configuration)	Indicates the configuration status of the AS-Interface master module. Flashes when the AS-Interface master module is in configuration mode.
Input LEDs	**IN0-IN3**	Indicates the operating status of four inputs at the address indicated by the address LEDs. Goes on when the corresponding input at the indicated address is on.
Output LEDs	**OUT0-OUT3**	Indicates the operating status of four outputs at the address indicated by the address LEDs. Goes on when the corresponding output at the indicated address is on.
Address LEDs	**0x-3x** (place of 10) **x0-x9** (place of 1) **A, B** (A or B slave)	Indicates the slave address of 0A through 31B. Goes on when the selected address exists. Flashes when the selected address does not exist.

Status LEDs

The operation modes of the AS-Interface master module can be changed by pressing the pushbuttons on the front of the AS-Interface master module or by executing ASI commands. The operation modes can be confirmed on the six status LEDs on the AS-Interface master module. For details about the ASI commands, see page 28-28.

Status LED Indication

Status LED		PWR	FLT	LMO	CMO	OFF	CNF
Connected Mode	Normal Protected Mode	ON *1	OFF *2	OFF	ON	OFF	OFF
	Normal Protected Offline	ON *1	ON	OFF	ON	ON	OFF
	Normal Protected Data Exchange Off	ON *1	ON	OFF	ON	OFF	OFF
Local Mode	Protected Mode	ON *1	OFF *2	ON	OFF	OFF	OFF
	Configuration Mode	ON *1	OFF *2	ON	OFF	OFF	Flash

*1: Goes off when AS-Interface power is not supplied.
*2: Goes on when an error is found on the AS-Interface bus.

Address LEDs and I/O LEDs

The operating status and I/O status of each slave can be monitored on the address LEDs and I/O LEDs on the front of the AS-Interface master module.

Slave Operating Status

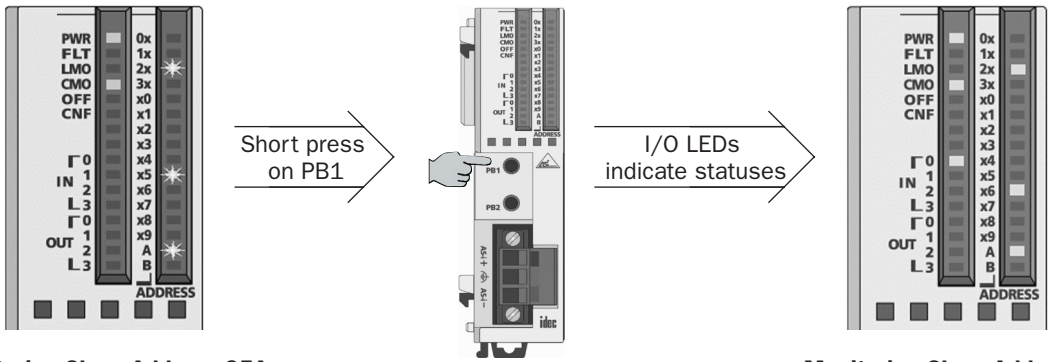
The operating status of each slave can be determined by viewing the address LEDs and I/O LEDs.

Address LED	I/O LED	Description
ON	ON or OFF	The slave at this address is active.
ON	Flash	The slave at this address is active, but has an error.
Flash	OFF	This address is not assigned a slave.
OFF	OFF	The AS-Interface bus communication is disabled because the AS-Interface power is not supplied or the AS-Interface master module is in normal protected offline.

Slave I/O Status

The I/O status of each slave can be monitored on the address LEDs and I/O LEDs. Use the short press to change the slave address when monitoring slave I/O status on the AS-Interface master module. A short press on PB1 increments the address. At the last address (31B), another short press will return to the first address (0A). A short press on PB2 decrements the address. At the first address (0A), another short press will return to the last address (31B).

The figures below illustrate what happens when you press the PB1 button while the address LEDs indicate 25A. The address LEDs increment to 26A where a slave is assigned. Note that the address LEDs flash if no slave is assigned.



Monitoring Slave Address 25A
Address LEDs are flashing since no slave is assigned.

Monitoring Slave Address 26A
Address LEDs go on and I/O LEDs indicate the statuses.

AS-Interface Operands

This chapter describes AS-Interface operands, or internal relays M1300 through M1997 and data registers D1700 through D1999, assigned in the CPU module to control and monitor the AS-Interface bus, and provides detailed description about internal relays allocated to SwitchNet™ control units for use as slaves in the AS-Interface network. Also describes ASI commands used to update AS-Interface operands in the CPU module or to control the AS-Interface master module.

AS-Interface Operand Allocation Numbers

The I/O data and parameters of slaves on the AS-Interface bus, the status of the AS-Interface bus, and various list information of the slaves are allocated to the AS-Interface master module EEPROM. This information is called AS-Interface objects, which can be accessed through the AS-Interface operands. The allocation is shown in the table below.

MicroSmart CPU Module		Precessing Time (msec) *1	Read/ Write	AS-Interface Master Module EEPROM	Operand Updated
Operand	Allocation No.			AS-Interface Object	
AS-Interface Internal Relays	M1300-M1617	3.0	R *2	Digital input (IDI) *4	Every scan
	M1620-M1937	3.0	W *2	Digital output (ODI) *4	
	M1940-M1997	1.0	R	Status information	
AS-Interface Data Registers	D1700-D1731	5.2	R	Analog input *5	
	D1732-D1763	5.2	W	Analog output *5	
	D1764-D1767	1.0	R *2	List of active slaves (LAS)	
	D1768-D1771	1.0	R *2	List of detected slaves (LDS)	
	D1772-D1775	1.0	R *2	List of peripheral fault slaves (LPF)	
	D1776-D1779	1.0	R/W *2*3	List of projected slaves (LPS)	Each time ASI command is executed
	D1780-D1843	10.4	R *2	Configuration data image (CDI)	
	D1844-D1907	10.4	R/W *2*3	Permanent configuration data (PCD)	
	D1908-D1923	3.0	R *2	Parameter image (PI)	
	D1924-D1939	3.0	R/W *2*3	Permanent parameter (PP)	
	D1940	0.7	R/W	Slave 0 ID1 code	
	D1941-D1945	—	R/W	For ASI command description	—
	D1946-D1999	—	—	(reserved)	—

*1: The time required to update the operand data. When using the AS-Interface master module, the scan time increases by a minimum of 10 ms.

*2: These AS-Interface operand data can be read or written using WindLDR. For details, see page 28-30.

*3: The LPS, PCD, and PP are set and downloaded to a PLC using WindLDR. For details, see page 28-32.

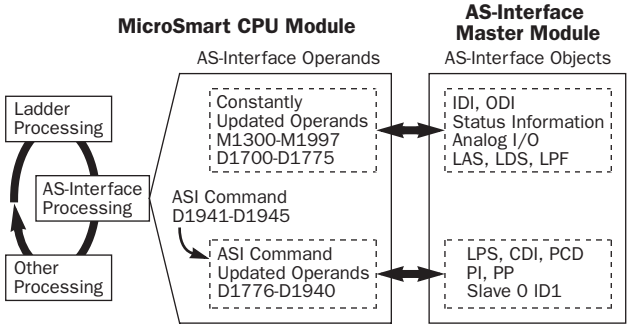
*4: IDI (input data image), ODI (output data image)

*5: The analog I/O data is updated only when an analog slave is connected to the AS-Interface bus.

Processing Time

AS-Interface internal relays for digital I/O and status information, and data registers for LAS, LDS, LPF are updated in every scan. Data registers for analog I/O operands are also updated in every scan only when analog I/O are connected to the AS-Interface bus. The processing times for these AS-Interface operands are shown in the table above.

Other AS-Interface data registers are updated when an ASI command is executed in the CPU module. For the processing times of the ASI commands, see page 28-28.



I/O Data for AS-Interface Master Module

The AS-Interface master module can process digital I/O data and analog I/O data. Digital I/O data can be a maximum of 4 digital inputs and 4 digital outputs per slave. Analog I/O data consists of 4 channels of 16-bit analog input or output data per slave.

Digital I/O Data of Standard Slaves and Expansion Slaves

The digital I/O data for standard slaves and A/B slaves (sensors and actuators) on the AS-Interface bus are allocated to the AS-Interface internal relays in the ascending order starting with slave 0. The input data image (IDI) for each slave is allocated to M1300 through M1617, and the output data image (ODI) is allocated to M1620 through M1937. For example, in the case of slave 3A, the input data is allocated to M1314 (DI0) through M1317 (DI3), and the output data is allocated to M1634 (DO0) through M1637 (DO3).

• Digital Input Data Image

Input Data Image (IDI)		Data Format							
		7 (DI3)	6 (DI2)	5 (DI1)	4 (DI0)	3 (DI3)	2 (DI2)	1 (DI1)	0 (DI0)
M1300	Byte 0	Slave 1(A)				(Slave 0)			
M1310	Byte 1	Slave 3(A)				Slave 2(A)			
M1320	Byte 2	Slave 5(A)				Slave 4(A)			
M1330	Byte 3	Slave 7(A)				Slave 6(A)			
M1340	Byte 4	Slave 9(A)				Slave 8(A)			
M1350	Byte 5	Slave 11(A)				Slave 10(A)			
M1360	Byte 6	Slave 13(A)				Slave 12(A)			
M1370	Byte 7	Slave 15(A)				Slave 14(A)			
M1380	Byte 8	Slave 17(A)				Slave 16(A)			
M1390	Byte 9	Slave 19(A)				Slave 18(A)			
M1400	Byte 10	Slave 21(A)				Slave 20(A)			
M1410	Byte 11	Slave 23(A)				Slave 22(A)			
M1420	Byte 12	Slave 25(A)				Slave 24(A)			
M1430	Byte 13	Slave 27(A)				Slave 26(A)			
M1440	Byte 14	Slave 29(A)				Slave 28(A)			
M1450	Byte 15	Slave 31(A)				Slave 30(A)			
M1460	Byte 16	Slave 1B				—			
M1470	Byte 17	Slave 3B				Slave 2B			
M1480	Byte 18	Slave 5B				Slave 4B			
M1490	Byte 19	Slave 7B				Slave 6B			
M1500	Byte 20	Slave 9B				Slave 8B			
M1510	Byte 21	Slave 11B				Slave 10B			
M1520	Byte 22	Slave 13B				Slave 12B			
M1530	Byte 23	Slave 15B				Slave 14B			
M1540	Byte 24	Slave 17B				Slave 16B			
M1550	Byte 25	Slave 19B				Slave 18B			
M1560	Byte 26	Slave 21B				Slave 20B			
M1570	Byte 27	Slave 23B				Slave 22B			
M1580	Byte 28	Slave 25B				Slave 24B			
M1590	Byte 29	Slave 27B				Slave 26B			
M1600	Byte 30	Slave 29B				Slave 28B			
M1610	Byte 31	Slave 31B				Slave 30B			

• **Digital Output Data Image**

Output Data Image (ODI)		Data Format							
		7 (DO3)	6 (DO2)	5 (DO1)	4 (DO0)	3 (DO3)	2 (DO2)	1 (DO1)	0 (DO0)
M1620	Byte 0	Slave 1(A)				(Slave 0)			
M1630	Byte 1	Slave 3(A)				Slave 2(A)			
M1640	Byte 2	Slave 5(A)				Slave 4(A)			
M1650	Byte 3	Slave 7(A)				Slave 6(A)			
M1660	Byte 4	Slave 9(A)				Slave 8(A)			
M1670	Byte 5	Slave 11(A)				Slave 10(A)			
M1680	Byte 6	Slave 13(A)				Slave 12(A)			
M1690	Byte 7	Slave 15(A)				Slave 14(A)			
M1700	Byte 8	Slave 17(A)				Slave 16(A)			
M1710	Byte 9	Slave 19(A)				Slave 18(A)			
M1720	Byte 10	Slave 21(A)				Slave 20(A)			
M1730	Byte 11	Slave 23(A)				Slave 22(A)			
M1740	Byte 12	Slave 25(A)				Slave 24(A)			
M1750	Byte 13	Slave 27(A)				Slave 26(A)			
M1760	Byte 14	Slave 29(A)				Slave 28(A)			
M1770	Byte 15	Slave 31(A)				Slave 30(A)			
M1780	Byte 16	Slave 1B				—			
M1790	Byte 17	Slave 3B				Slave 2B			
M1800	Byte 18	Slave 5B				Slave 4B			
M1810	Byte 19	Slave 7B				Slave 6B			
M1820	Byte 20	Slave 9B				Slave 8B			
M1830	Byte 21	Slave 11B				Slave 10B			
M1840	Byte 22	Slave 13B				Slave 12B			
M1850	Byte 23	Slave 15B				Slave 14B			
M1860	Byte 24	Slave 17B				Slave 16B			
M1870	Byte 25	Slave 19B				Slave 18B			
M1880	Byte 26	Slave 21B				Slave 20B			
M1890	Byte 27	Slave 23B				Slave 22B			
M1900	Byte 28	Slave 25B				Slave 24B			
M1910	Byte 29	Slave 27B				Slave 26B			
M1920	Byte 30	Slave 29B				Slave 28B			
M1930	Byte 31	Slave 31B				Slave 30B			

⚠ Caution

• Immediately after power up, the digital I/O data of standard slaves and expansion slaves cannot be accessed. Data communication between the CPU module and the connected slaves starts when special internal relay M1945 (Normal_Operation_Active) turns on. Make sure that M1945 is on before starting to access the slave I/O data.

Analog I/O Data of Analog Slaves

The I/O data for a maximum of seven analog slaves (four channels for each slave) on the AS-Interface bus is stored to AS-Interface data registers in the CPU module. The analog slave addresses (1 to 31) are in the ascending order. The input data for each analog slave is allocated to data registers D1700 to D1731, and the output data is allocated to D1732 to D1763.

The AS-Interface master module is compliant with analog slave profile 7.3.

⚠ Caution

- The maximum number of analog slaves that can be connected to the AS-Interface bus is seven. Do not connect eight or more analog slaves to one bus, otherwise the slaves will not function correctly.
- When data registers D1700 through D1731 allocated to analog inputs contain 7FFF, do not use this data for programming, because this value is reserved for a special meaning as follows:

  Unused channel on a slave allocated to analog slave. (For a channel on a slave not allocated an analog slave, the corresponding data register holds an indefinite value.)
  Data overflow.
  Communication between the master and analog slave is out of synchronism.
- When using analog slaves, read the user's manual for the analog slave to process the data properly.

• Analog Input Data

Analog Input		Channel No.	Data Format
D1700	Bytes 0 and 1	Channel 1	1st data (AI0)
D1701	Bytes 2 and 3	Channel 2	
D1702	Bytes 4 and 5	Channel 3	
D1703	Bytes 6 and 7	Channel 4	
D1704	Bytes 8 and 9	Channel 1	2nd data (AI1)
D1705	Bytes 10 and 11	Channel 2	
D1706	Bytes 12 and 13	Channel 3	
D1707	Bytes 14 and 15	Channel 4	
D1708	Bytes 16 and 17	Channel 1	3rd data (AI2)
D1709	Bytes 18 and 19	Channel 2	
D1710	Bytes 20 and 21	Channel 3	
D1711	Bytes 22 and 23	Channel 4	
D1712	Bytes 24 and 25	Channel 1	4th data (AI3)
D1713	Bytes 26 and 27	Channel 2	
D1714	Bytes 28 and 29	Channel 3	
D1715	Bytes 30 and 31	Channel 4	
D1716	Bytes 32 and 33	Channel 1	5th data (AI4)
D1717	Bytes 34 and 35	Channel 2	
D1718	Bytes 36 and 37	Channel 3	
D1719	Bytes 38 and 39	Channel 4	
D1720	Bytes 40 and 41	Channel 1	6th data (AI5)
D1721	Bytes 42 and 43	Channel 2	
D1722	Bytes 44 and 45	Channel 3	
D1723	Bytes 46 and 47	Channel 4	
D1724	Bytes 48 and 49	Channel 1	7th data (AI6)
D1725	Bytes 50 and 51	Channel 2	
D1726	Bytes 52 and 53	Channel 3	
D1727	Bytes 54 and 55	Channel 4	
D1728	Bytes 56 and 57	—	(reserved)
D1729	Bytes 58 and 59	—	
D1730	Bytes 60 and 61	—	
D1731	Bytes 62 and 63	—	

• **Analog Output Data**

Analog Output		Channel No.	Data Format
D1732	Bytes 0 and 1	Channel 1	1st data (AO0)
D1733	Bytes 2 and 3	Channel 2	
D1734	Bytes 4 and 5	Channel 3	
D1735	Bytes 6 and 7	Channel 4	
D1736	Bytes 8 and 9	Channel 1	2nd data (AO1)
D1737	Bytes 10 and 11	Channel 2	
D1738	Bytes 12 and 13	Channel 3	
D1739	Bytes 14 and 15	Channel 4	
D1740	Bytes 16 and 17	Channel 1	3rd data (AO2)
D1741	Bytes 18 and 19	Channel 2	
D1742	Bytes 20 and 21	Channel 3	
D1743	Bytes 22 and 23	Channel 4	
D1744	Bytes 24 and 25	Channel 1	4th data (AO3)
D1745	Bytes 26 and 27	Channel 2	
D1746	Bytes 28 and 29	Channel 3	
D1747	Bytes 30 and 31	Channel 4	
D1748	Bytes 32 and 33	Channel 1	5th data (AO4)
D1749	Bytes 34 and 35	Channel 2	
D1750	Bytes 36 and 37	Channel 3	
D1751	Bytes 38 and 39	Channel 4	
D1752	Bytes 40 and 41	Channel 1	6th data (AO5)
D1753	Bytes 42 and 43	Channel 2	
D1754	Bytes 44 and 45	Channel 3	
D1755	Bytes 46 and 47	Channel 4	
D1756	Bytes 48 and 49	Channel 1	7th data (AO6)
D1757	Bytes 50 and 51	Channel 2	
D1758	Bytes 52 and 53	Channel 3	
D1759	Bytes 54 and 55	Channel 4	
D1760	Bytes 56 and 57	—	(reserved)
D1761	Bytes 58 and 59	—	
D1762	Bytes 60 and 61	—	
D1763	Bytes 62 and 63	—	

For example, when analog input slaves 1, 13 and 20, analog output slaves 5 and 25, and analog I/O slaves 14 and 21 are used, the analog I/O slave data will be allocated by configuration as shown below and maintained until the next configuration is executed. Four channels (8 bytes) are always reserved for each slave.

Analog Slave Module	Data Storage	Analog Input Slave	Data Storage	Analog Output Slave
1st	D1700-D1703	**Slave 1**	D1732-D1735	Unused
2nd	D1704-D1707	Unused	D1736-D1739	**Slave 5**
3rd	D1708-D1711	**Slave 13**	D1740-D1743	Unused
4th	D1712-D1715	**Slave 14**	D1744-D1747	**Slave 14**
5th	D1716-D1719	**Slave 20**	D1748-D1751	Unused
6th	D1720-D1723	**Slave 21**	D1752-D1755	**Slave 21**
7th	D1724-D1727	Unused	D1756-D1759	**Slave 25**
(8th)	(D1728-D1731)	(reserved)	(D1760-D1763)	(reserved)

Status Information

The status information is allocated to AS-Interface internal relays M1940 through M1997. These internal relays are used to monitor the status of the AS-Interface bus. If an error occurs on the bus, you can also confirm the error with the status LEDs on the front of the AS-Interface master module in addition to these status internal relays.

• Status Information Internal Relays

Internal Relays	Status	Description	
		ON	OFF
M1940	Config_OK	Configuration is complete.	Configuration is incomplete.
M1941	LDS.0	Slave address 0 is detected on the AS-Interface bus.	Slave address 0 is not detected on the AS-Interface bus.
M1942	Auto_Address_Assign	Auto addressing is enabled.	Auto addressing is disabled.
M1943	Auto_Address_Available	Auto addressing is ready.	Auto addressing is not ready.
M1944	Configuration	Configuration mode is enabled.	Other than configuration mode.
M1945	Normal_Operation_Active	Normal protected mode is enabled.	Other than normal protected mode.
M1946	APF/not APO	AS-Interface power supply failure.	AS-Interface power supply is normal.
M1947	Offline_Ready	Normal protected offline is enabled.	Other than normal protected offline.
M1950	Periphery_OK	Peripheral devices are normal.	Peripheral devices are abnormal.
M1951-M1957	(reserved)	—	—
M1960	Data_Exchange_Active	Data exchange is enabled.	Data exchange is prohibited.
M1961	Off-line	Command to go to normal protected offline was issued by the pushbutton or WindLDR.	Command to go to normal protected offline was not issued.
M1962	Connected Mode	Connected mode is enabled.	Local mode is enabled.
M1963-M1997	(reserved)	—	—

M1940 Config_OK

M1940 indicates the configuration status. M1940 goes on when the permanent configuration data (PCD) stored in the AS-Interface master module EEPROM matches the configuration data image (CDI). When configuration is changed, e.g. a new slave is added or a slave fails, M1940 goes off. Then, the FLT LED goes on.

M1941 LDS.0

M1941 is used to check for the presence of a slave with address 0 on the AS-Interface bus. M1941 goes on when a slave with address 0 (the factory setting) is detected on the AS-Interface bus in normal protected mode or protected mode, or when a slave address is changed to 0 while the AS-Interface master module is in normal protected mode.

M1942 Auto_Address_Assign

M1942 indicates that the auto addressing function is enabled. The default setting is “enabled,” and M1942 is normally on. This setting can be changed using the ASI commands Enable Auto Addressing and Disable Auto Addressing.

Note: When the auto addressing function is enabled at the AS-Interface master module and a slave fails, you can replace the slave with a new slave which has the same identification codes without stopping the AS-Interface bus.

- If the replacement slave is assigned the same address and has the same identification codes as the failed slave, the replacement slave is automatically added to the LDS (list of detected slaves) to continue operation. If the assigned address or the identification codes of the replacement slave are different from the failed slave, the FLT LED will go on.
- When replacing a failed slave with a new slave which is assigned address 0 (factory setting) and has the same identification codes, the new slave will be assigned the address of the failed slave and added to the LDS and LAS (list of active slaves). If the identification codes of the replacement slave are different from the failed slave, the FLT LED will go on.
- The auto addressing function for a replacement slave works only when one slave has failed. This function cannot be used to replace multiple slaves.

M1943 Auto_Address_Available

M1943 indicates whether or not the conditions for the auto addressing function are satisfied. M1943 goes on when the auto addressing function is enabled and there is one faulty slave (a slave which cannot be recognized by the AS-Interface master module) on the AS-Interface bus.

M1944 Configuration

M1944 indicates whether the AS-Interface master module is in configuration mode (on) or other mode (off). While configuration mode is enabled, M1944 remains on, and the CNF LED flashes.

M1945 Normal_Operation_Active

M1945 remains on while the AS-Interface master module is in normal protected mode. M1945 is off while in other modes. When M1945 turns on, the CPU module starts to exchange data communication with the connected slaves.

M1946 APF/not APO

M1946 goes on when the AS-Interface power supply has failed, then the PWR LED goes off.

M1947 Offline_Ready

M1947 indicates that the AS-Interface master module is in normal protected offline. While in normal protected offline, M1947 remains on and the OFF LED also remains on.

M1950 Periphery_OK

M1950 remains on while the AS-Interface master module does not detect a failure in peripheral devices. When a failure is found, M1950 goes off.

M1960 Data_Exchange_Active

M1960 indicates that data exchange is enabled. While M1960 is on, the AS-Interface master module is in normal protected mode, and data exchange between the AS-Interface master module and slaves is enabled. The data exchange can be enabled and disabled using the ASI commands Enable Data Exchange and Prohibit Data Exchange.

M1961 Off-line

M1961 goes on when a command to switch to normal protected offline is issued. To switch to normal protected offline from normal protected mode, either press the PB2 button on the AS-Interface master module or issue the ASI command Go to Normal Protected Offline. M1961 remains on until normal protected offline is exited.

M1962 Connected Mode

M1962 indicates that the AS-Interface master module is in connected mode. While in connected mode, M1962 remains on. Then, LMO LED remains off and the CMO LED remains on.

Slave List Information

Data registers D1764 through D1779 are assigned to slave list information to determine the operating status of the slaves. The slave list information is grouped into four lists. List of active slaves (LAS) shows the slaves currently in operation. List of detected slaves (LDS) the slaves detected on the AS-Interface bus. List of peripheral fault slaves (LPF) the faulty slaves. List of projected slaves (LPS) the slave configuration stored in the AS-Interface master module.

List of Active Slaves (LAS)

Data registers D1764 through D1767 are allocated to read the LAS. You can check the register bits to determine the operating status of each slave. When a bit is on, it indicates that the corresponding slave is active.

LAS		Data Format	
		Bits 15 to 8	Bits 7 to 0
D1764	Bytes 0 and 1	Slaves 15(A) to 8(A)	Slaves 7(A) to 0
D1765	Bytes 2 and 3	Slaves 31(A) to 24(A)	Slaves 23(A) to 16(A)
D1766	Bytes 4 and 5	Slaves 15B to 8B	Slaves 7B to (0B)
D1767	Bytes 6 and 7	Slaves 31B to 24B	Slaves 23B to 16B

List of Detected Slaves (LDS)

Data registers D1768 through D1771 are allocated to read the LDS. You can check the register bits to determine the detection status of each slave. When a bit is on, it indicates that the corresponding slave has been detected by the master.

LDS		Data Format	
		Bits 15 to 8	Bits 7 to 0
D1768	Bytes 0 and 1	Slaves 15(A) to 8(A)	Slaves 7(A) to 0
D1769	Bytes 2 and 3	Slaves 31(A) to 24(A)	Slaves 23(A) to 16(A)
D1770	Bytes 4 and 5	Slaves 15B to 8B	Slaves 7B to (0B)
D1771	Bytes 6 and 7	Slaves 31B to 24B	Slaves 23B to 16B

List of Peripheral Fault Slaves (LPF)

Data registers D1772 through D1775 are allocated to read the LPF. You can check the register bits to determine the fault status of each slave. When a bit is on, it indicates that the corresponding slave is faulty.

LPF		Data Format	
		Bits 15 to 8	Bits 7 to 0
D1772	Bytes 0 and 1	Slaves 15(A) to 8(A)	Slaves 7(A) to 0
D1773	Bytes 2 and 3	Slaves 31(A) to 24(A)	Slaves 23(A) to 16(A)
D1774	Bytes 4 and 5	Slaves 15B to 8B	Slaves 7B to (0B)
D1775	Bytes 6 and 7	Slaves 31B to 24B	Slaves 23B to 16B

List of Projected Slaves (LPS)

Data registers D1776 through D1779 are allocated to read and write the LPS. The LPS settings are stored to the AS-Interface master module when either Auto Configuration or Manual Configuration is executed on WindLDR. The ASI command Read LPS can be used to read the LPS data to data registers D1776 through D1779. Then, you can check the register bits to determine the slave projection. When a bit is on, it indicates that the corresponding slave is set as a projected slave. After changing the LPS settings, execute the ASI command Read LPS, then you can use the updated data for program execution.

LPS		Data Format	
		Bits 15 to 8	Bits 7 to 0
D1776	Bytes 0 and 1	Slaves 15(A) to 8(A)	Slaves 7(A) to 0
D1777	Bytes 2 and 3	Slaves 31(A) to 24(A)	Slaves 23(A) to 16(A)
D1778	Bytes 4 and 5	Slaves 15B to 8B	Slaves 7B to (0B)
D1779	Bytes 6 and 7	Slaves 31B to 24B	Slaves 23B to 16B

Slave Identification Information (Slave Profile)

Data registers D1780 through D1940 are assigned to the slave identification information, or the slave profile. The slave profile includes configuration data and parameters to indicate the slave type and slave operation, respectively.

Configuration Data Image (CDI)

Data registers D1780 through D1843 are allocated to read the CDI of each slave. The CDI is the current slave configuration data collected by the AS-Interface master module at power-up and stored in the AS-Interface master module.

The CDI is made up of four codes: the ID code, I/O code, ID2 code, and ID1 code. The CDI of slaves not connected to the AS-Interface bus is FFFFh.

The ASI command Read CDI can be used to read the CDI data to data registers D1780 through D1843. Execute the ASI command Read CDI before using the CDI data for program execution.

CDI		Data Format			
		Bits 15 to 12 ID Code	Bits 11 to 8 I/O Code	Bits 7 to 4 ID2 Code	Bits 3 to 0 ID1 Code
D1780	Bytes 0 and 1	Slave 0			
D1781	Bytes 2 and 3	Slave 1(A)			
D1782	Bytes 4 and 5	Slave 2(A)			
D(1780+N)		Slave N(A)			
D1811	Bytes 62 and 63	Slave 31(A)			
D1812	Bytes 64 and 65	(unused)			
D1813	Bytes 66 and 67	Slave 1B			
D(1812+N)		Slave NB			
D1843	Bytes 126 and 127	Slave 31B			

Permanent Configuration Data (PCD)

Data registers D1844 through D1907 are allocated to read and write the PCD of each slave. Like the CDI, the PCD is made up of four codes: the ID code, I/O code, ID2 code, and ID1 code.

When auto configuration is executed, the CDI is copied to the PCD and stored in the EEPROM of the AS-Interface master module. When you execute manual configuration, you can set the PCD using the Configure Slave dialog box on WindLDR. Set the PCD of each slave to the same value as its CDI. If the PCD is different from the CDI for a slave, then that slave does not function correctly. Set FFFFh to the PCD of vacant slave numbers.

The ASI command Read PCD can be used to read the PCD data to data registers D1844 through D1907. Execute the ASI command Read PCD before using the PCD data for program execution.

PCD		Data Format			
		Bits 15 to 12 ID Code	Bits 11 to 8 I/O Code	Bits 7 to 4 ID2 Code	Bits 3 to 0 ID1 Code
D1844	Bytes 0 and 1	Slave 0			
D1845	Bytes 2 and 3	Slave 1(A)			
D1846	Bytes 4 and 5	Slave 2(A)			
D(1844+N)		Slave N(A)			
D1875	Bytes 62 and 63	Slave 31(A)			
D1876	Bytes 64 and 65	(unused)			
D1877	Bytes 66 and 67	Slave 1B			
D(1876+N)		Slave NB			
D1907	Bytes 126 and 127	Slave 31B			

Parameter Image (PI)

Data registers D1908 through D1923 are allocated to read the PI of each slave. The PI is made up of four parameters: the P3, P2, P1, and P0. The PI is the current slave parameter data collected by the AS-Interface master module at power-up and stored in the AS-Interface master module. To change the PI settings, use WindLDR (Slave Status dialog box) or execute the ASI command Change Slave PI.

The ASI command Read PI can be used to read PI data to data registers D1908 through D1923. After changing the PI settings, execute the ASI command Read PI, then you can use the updated PI data for program execution.

PI		Data Format			
		Bits 15 to 12 P3/P2/P1/P0	**Bits 11 to 8** P3/P2/P1/P0	**Bits 7 to 4** P3/P2/P1/P0	**Bits 3 to 0** P3/P2/P1/P0
D1908	Bytes 0 and 1	Slave 3(A)	Slave 2(A)	Slave 1(A)	Slave 0
D1909	Bytes 2 and 3	Slave 7(A)	Slave 6(A)	Slave 5(A)	Slave 4(A)
D1910	Bytes 4 and 5	Slave 11(A)	Slave 10(A)	Slave 9(A)	Slave 8(A)
D(1908+N/4)		Slave (N+3)(A)	Slave (N+2)(A)	Slave (N+1)(A)	Slave N(A)
D1915	Bytes 14 and 15	Slave 31(A)	Slave 30(A)	Slave 29(A)	Slave 28(A)
D1916	Bytes 16 and 17	Slave 3B	Slave 2B	Slave 1B	(unused)
D1917	Bytes 18 and 19	Slave 7B	Slave 6B	Slave 5B	Slave 4B
D(1916+N/4)		Slave (N+3)B	Slave (N+2)B	Slave (N+1)B	Slave NB
D1923	Bytes 30 and 31	Slave 31B	Slave 30B	Slave 29B	Slave 28B

Permanent Parameter (PP)

Data registers D1924 through D1939 are allocated to read and write the PP of each slave. Like the PI, the PP is made up of four parameters: the P3, P2, P1, and P0. When auto configuration is executed, the PI is copied to the PP and stored in the EEPROM of the AS-Interface master module. When you execute manual configuration, you can set the PP using the Configure Slave dialog box on WindLDR.

The ASI command Read PP can be used to read PP data to data registers D1924 through D1939. After changing the PP settings, execute the ASI command Read PP, then you can use the updated PP data for program execution.

PP		Data Format			
		Bits 15 to 12 P3/P2/P1/P0	**Bits 11 to 8** P3/P2/P1/P0	**Bits 7 to 4** P3/P2/P1/P0	**Bits 3 to 0** P3/P2/P1/P0
D1924	Bytes 0 and 1	Slave 3(A)	Slave 2(A)	Slave 1(A)	Slave 0
D1925	Bytes 2 and 3	Slave 7(A)	Slave 6(A)	Slave 5(A)	Slave 4(A)
D1926	Bytes 4 and 5	Slave 11(A)	Slave 10(A)	Slave 9(A)	Slave 8(A)
D(1924+N/4)		Slave (N+3)(A)	Slave (N+2)(A)	Slave (N+1)(A)	Slave N(A)
D1931	Bytes 14 and 15	Slave 31(A)	Slave 30(A)	Slave 29(A)	Slave 28(A)
D1932	Bytes 16 and 17	Slave 3B	Slave 2B	Slave 1B	(unused)
D1933	Bytes 18 and 19	Slave 7B	Slave 6B	Slave 5B	Slave 4B
D(1932+N/4)		Slave (N+3)B	Slave (N+2)B	Slave (N+1)B	Slave NB
D1939	Bytes 30 and 31	Slave 31B	Slave 30B	Slave 29B	Slave 28B

Changing ID1 Code of Slave 0

Data register D1940 is allocated to read and write the ID1 code of slave 0. To change the slave 0 ID1 settings, store a required value in D1940 and execute the ASI command Write Slave 0 ID1. The ASI command Read Slave 0 ID1 can be used to read slave 0 ID1 data to data register D1940. After changing the slave 0 ID1 settings, execute the ASI command Read Slave 0 ID1, then you can use the updated slave 0 ID1 data for program execution.

Slave 0 ID1 Code		Data Format			
		Bits 15 to 12	**Bits 11 to 8**	**Bits 7 to 4**	**Bits 3 to 0**
D1940	Bytes 0 and 1	—	—	—	ID1 code

ASI Commands

The ASI commands are used to update AS-Interface operands in the CPU module or to control the AS-Interface master module. Data registers D1941 through D1944 are used to store command data. D1945 is used to store a request code before executing the command. While the command is executed, D1945 stores status and result codes.

ASI Command Format

Command Part (8 bytes)				Request/Result
D1941	D1942	D1943	D1944	D1945

ASI Command Data

To execute an ASI command, store required values to data resisters D1941 through D1945 as listed in the table below:

ASI Command	Processing Time (ms)	Description	Command Data (Hexadecimal)				
			D1941	D1942	D1943	D1944	D1945
Read LPS	1.0 *3	Reads LPS to D1776-D1779	010B	084C	0000	0000	0001
Read CDI	10.4 *3	Reads CDI to D1780-D1843	010C	4050	0000	0000	0001
Read PCD	10.4 *3	Reads PCD to D1844-D1907	010E	4090	0000	0000	0001
Read PI	3.0 *3	Reads PI to D1908-D1923	0107	20D0	0000	0000	0001
Read PP	3.0 *3	Reads PP to D1924-D1939	0108	20E0	0000	0000	0001
Read Slave 0 ID1	0.7 *3	Reads slave 0 ID1 to D1940	0109	02F0	0000	0000	0001
Write Slave 0 ID1	0.7 *3	Writes D1940 to slave 0 ID1	0209	02F0	0000	0000	0001
Copy PI to PP	0.8 *4	Copies parameter image to permanent parameter	0306	0100	0000	0000	0001
Change Slave PI *1	0.8 *4	Writes PI (∗) to slave (∗∗) (Note)	0306	0102	000∗	00∗∗	0001
Go to Normal Protected Offline	0.8 *4	From normal protected mode to normal protected offline	0306	0301	0000	0000	0001
Go to Normal Protected Mode	0.8 *4	From normal protected offline to normal protected mode	0306	0300	0000	0000	0001
Prohibit Data Exchange	0.8 *4	From normal protected mode to normal protected data exchange off	0306	0401	0000	0000	0001
Enable Data Exchange	0.8 *4	From normal protected data exchange off to normal protected mode	0306	0400	0000	0000	0001
Change Slave Address *2	0.8 *4	Change slave address (∗∗) to new address (++) (Note)	0306	0500	00∗∗	00++	0001
Enable Auto Addressing	0.8 *4	Enables auto address assign (default)	0306	0800	0000	0000	0001
Disable Auto Addressing	0.8 *4	Disables auto address assign	0306	0801	0000	0000	0001

*1: WindLDR has the Slave Status dialog box to execute this command to write a PI value to a designated slave. See Sample Program on page 28-29.

*2: WindLDR has the Change Slave Address dialog box to execute this command.

*3: Completed in a scan when the five data registers store respective values. When completed, D1945 stores 4. See Request and Result Codes on page 28-29. Other commands takes several scans to complete execution.

*4: Each scan time extends by 0.8 ms. At least 1 sec is required until the ASI command takes effect.

Note: Specify the slave address in the data register as shown in the table below:

Slave Address	Data Register Value		Slave Address	Data Register Value	
	Hexadecimal	Decimal		Hexadecimal	Decimal
0(A)	00h	0	—	—	—
1(A)	01h	1	1B	21h	33
2(A)	02h	2	2B	22h	34
⋮	⋮	⋮	⋮	⋮	⋮
31(A)	1Fh	31	31B	3Fh	63

Request and Result Codes

D1945 Value Low Byte	Description	Note
00h	Initial value at power up	While D1945 lower byte stores 01h, 02h, or 08h, do not write any value to D1945, otherwise the ASI command is not executed correctly. The CPU module stores all values automatically, except for 01h.
01h	Request	
02h	Processing ASI command	
04h	Completed normally	
08h	(Executing configuration)	
14h	Peripheral device failure	
24h	ASI command error	
74h	Impossible to execute	
84h	Execution resulting in error	

Sample Program: Change Slave PI

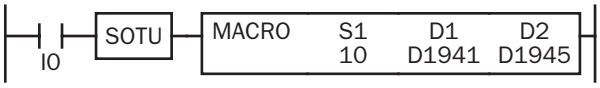
This sample program changes the PI value of slave 1A to 3. To use the ASI command Change Slave PI, store new parameter value 3 to D1943 and 1 to D1944 to designate the slave address using the MACRO instruction on WindLDR.

Program	Command Data (Hexadecimal)				
	D1941	D1942	D1943	D1944	D1945
Write PI parameter “3” to slave 1A	0306	0102	0003	0001	0001

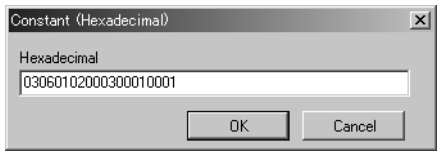
To designate slave 31A, set 001F to D1944. For slave 1B, set 0021.

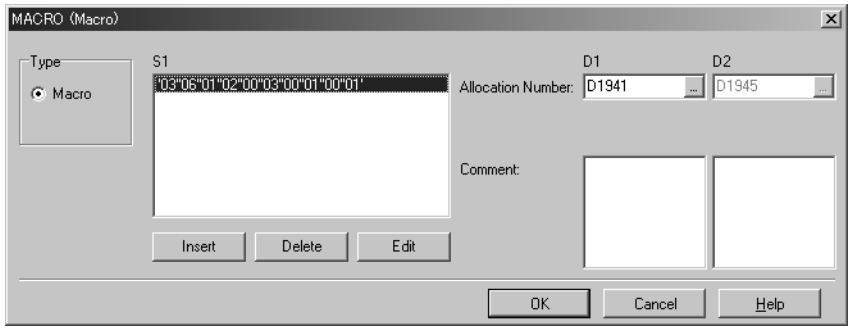
Parameters P3 through P0 are weighted as shown in the table below. When the PI parameter is set to 3, P3 and P2 are turned off, and P1 and P0 are turned on.

Parameter	P3	P2	P1	P0
Weight	8	4	2	1
ON/OFF	OFF	OFF	ON	ON



When input I0 turns on, the MACRO instruction stores hexadecimal values 0306, 0102, 0003, 0001, and 0001 to five data registers D1941 through D1945.





Using WindLDR

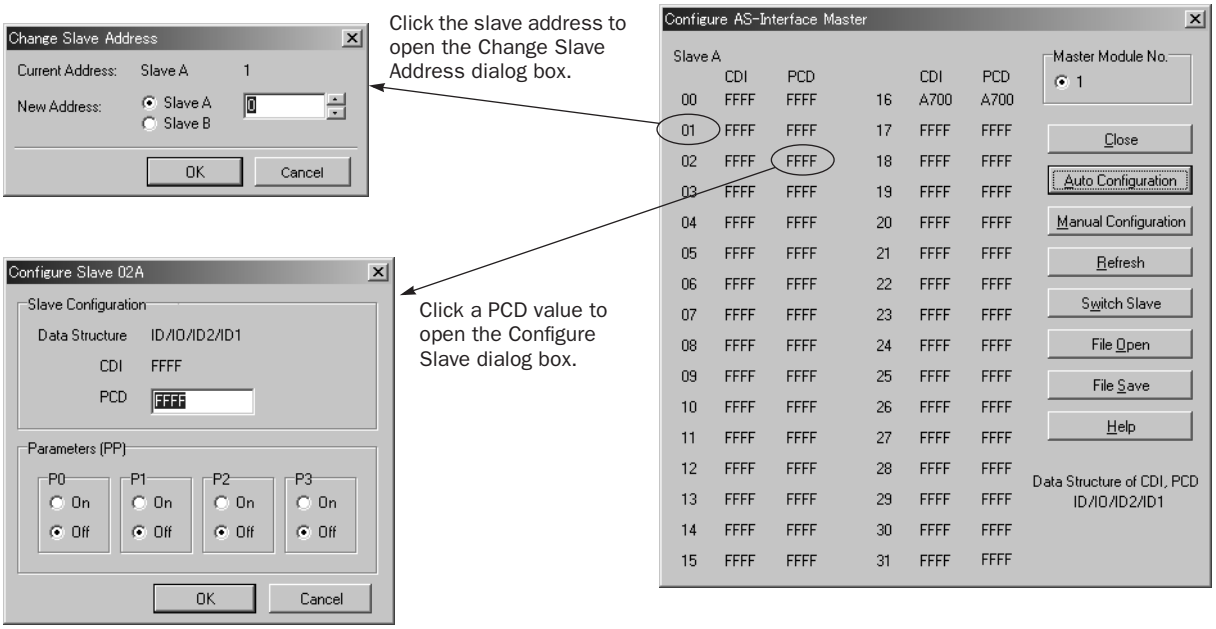
This section describes the procedures to use WindLDR for the AS-Interface system. WindLDR contains the Configure AS-Interface Master dialog box to configure slaves and to change slave addresses, and the Monitor AS-Interface Slave dialog box to monitor the slave operation.

For the procedures to select the PLC type and Function Area Settings, see page 28-8.

Configure AS-Interface Master

AS-Interface compatible slave devices are set to address 0 at factory and must be assigned a unique slave address so that the master can communicate with the slave correctly.

From the WindLDR menu bar, select **Configure** > **AS-Interface Master**. The Configure AS-Interface Master dialog box appears.



Dialog Box	Button	Description
Configure AS-Interface Master	**Auto Configuration**	Writes the currently connected AS-Interface slave configuration (LDS, CDI, PI) information to the AS-Interface master module EEPROM (LPS, PCD, PP).
	Manual Configuration	Writes the slave PCD and parameters configured by the user to the AS-Interface master module EEPROM (LPS, PCD, PP).
	Refresh	Refreshes the screen display.
	Switch Slave	Switches the dialog box for setting Slave A or Slave B.
	File Open	Opens the configuration (LPS, PCD, PP) file.
	File Save	Saves the configuration (LPS, PCD, PP) file.
	Help	Displays explanations for functions on the screen.
Change Slave Address	**OK**	Changes the slave address.
	Cancel	Discards the changes and closes the window.
Configure Slave	**OK**	Updates the PCD and PP. Not written to the master module yet.
	Cancel	Discards the changes and closes the window.

Slave Address Shading Colors

Operating status of the slave can be confirmed by viewing the shading color at the slave address on the Configure AS-Interface Master dialog box. The screen display can be updated by clicking the **Refresh** button.

Address Shading	Description	LAS List of active slaves	LDS List of detected slaves	LPF List of peripheral fault slaves	LPS List of projected slaves
No Shade	The slave is not recognized by the master.	OFF	OFF	OFF	ON/OFF
Blue Shade	The slave is active.	ON	ON	OFF	ON
Yellow Shade	The slave is recognized but not enabled to operate.	OFF	ON	OFF	OFF
Red Shade	An error was found in the slave.	ON/OFF	ON/OFF	ON	ON/OFF

Change Slave Address

When a slave is connected to the AS-Interface master module, the slave address can be changed using WindLDR.

⚠ Warning

• Duplicate slave addresses

Each slave must have a unique address. Do not connect two or more slaves with the same address, otherwise the AS-Interface master module cannot locate the slave correctly. When two slaves have the same address and different identification codes (ID, I/O, ID2, ID1), the AS-Interface master module detects an error. When two slaves have the same address and same identification codes, the AS-Interface master module cannot detect an error. Failure to observe this warning may cause severe personal injury or heavy damage to property.

⚠ Caution

• When a slave with address 0 is connected to the AS-Interface master module, power up the MicroSmart CPU module first. Approximately 5 seconds later, turn on the AS-Interface power supply. If the CPU module and AS-Interface power supply are turned on at the same time, the AS-Interface master module enters normal protected offline. In this mode, slave addresses can be changed, but the slave status cannot be confirmed on WindLDR.

To change a slave address, from the WindLDR menu bar, select **Configure > AS-Interface Master**. The Configure AS-Interface Master dialog box appears.

Click a slave address to open the Change Slave Address dialog box. Select Slave A or Slave B, enter a required address in the New Address field, and click **OK**. The Change Slave Address dialog box is closed. The new slave address is stored in the slave module nonvolatile memory.

If the command is not processed correctly, the error message "AS-Interface Master Error" and an error code will appear. See page 28-34.

The address cannot be changed in the following cases.

Error Code	Description
1	• An error was found on the expansion I/O bus.
7	• The AS-Interface master module is in local mode.
8	• The slave you are trying to change does not exist. • A slave of the designated new address already exists. • While a standard slave was set at A address, attempt was made to set an A/B slave at B address of the same number. • While an A/B slave was set at B address, attempt was made to set a standard slave at A address of the same number.

Configuration

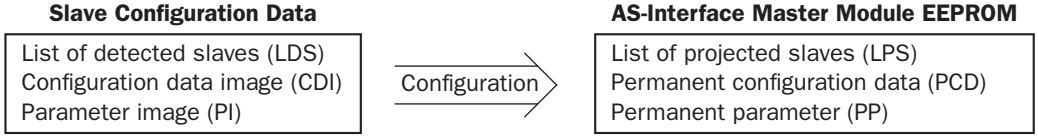
Before commissioning the AS-Interface master module, configuration must be done using either WindLDR or the pushbuttons on the front of the AS-Interface master module. This section describes the method of configuration using WindLDR. For configuration using the pushbuttons, see page 28-10. Configuration is the procedure to store the following information to the AS-Interface master module EEPROM.

- A list of slave addresses to be used
- Configuration data to specify slave types, or identification codes (ID, I/O, ID2, ID1)
- Parameters (P3, P2, P1, P0) to designate the slave operation at power-up

WindLDR provides two options for configuration: auto configuration to execute automatic configuration and manual configuration to execute configuration according to the data selected by the user.

Auto Configuration

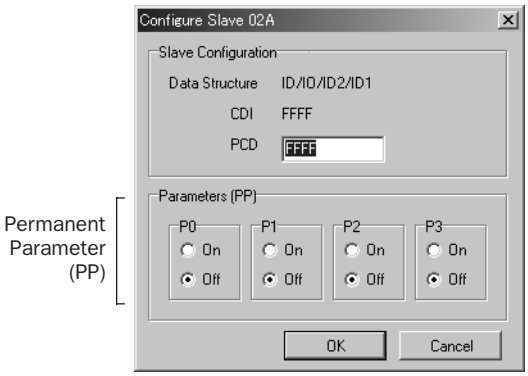
Auto configuration stores the current slave configuration data (LDS, CDI, PI) to the AS-Interface master module EEPROM (LPS, PCD, PP). To execute auto configuration, press **Auto Configuration** in the Configure AS-Interface Master dialog box. Auto configuration has the same effect as the configuration using the pushbuttons on the AS-Interface master module.



Manual Configuration

Manual configuration is the procedure to write the LPS, PCD, and PP designated on WindLDR to the AS-Interface master module EEPROM. LPS is automatically generated by WindLDR based on the value for PCD.

PCD	LPS
FFFFh	0
Other values	1



To change PCD and PP, use the Configure Slave dialog box. Set the PCD of each slave to the same value as its CDI. If the PCD is different from the CDI for a slave, then that slave does not function correctly. Set FFFFh to the PCD of vacant slave numbers.

After entering a PCD value and selecting parameter statuses, click **OK**. At this point, the configuration data are not stored to the AS-Interface master module EEPROM. To store the changes, click **Manual Configuration** on the Configure AS-Interface Master dialog box. The screen display of the Configure AS-Interface Master dialog box can be updated using **Refresh**.

If you save the configuration data to a file, you can open the file to configure other AS-Interface master modules using the same data. To save and open the configuration file, click **File Save** or **File Open**.

If the configuration command is not processed correctly, the error message "AS-Interface Master Error" and an error code will appear. See page 28-34.

If the error message "Configuration failure. Confirm the slave setup, and perform configuration again." is shown, and the FLT LED is on, then remove the cause of the error, referring to page 28-13, and repeat configuration.

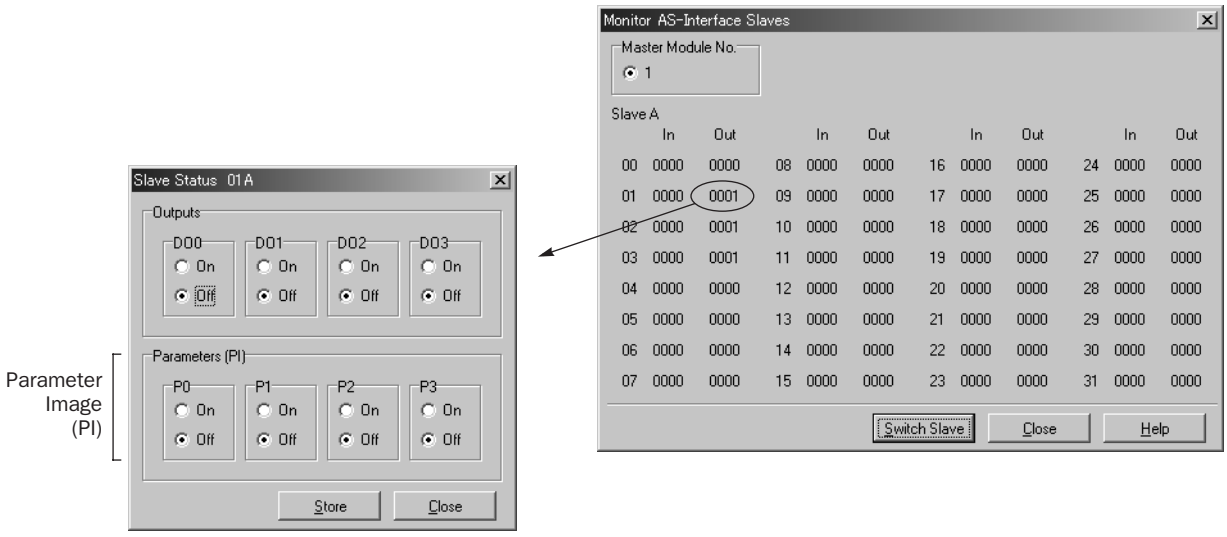
The configuration cannot be done in the following cases.

Error Code	Description
1	• An error was found on the expansion I/O bus.
2	• While the AS-Interface master module was in offline mode, attempt was made to execute auto configuration or manual configuration.
7	• While slave address 0 existed on the bus, attempt was made to execute auto configuration or manual configuration. • The AS-Interface master module is in local mode.

Monitor AS-Interface Slave

While the MicroSmart is communicating with AS-Interface slaves through the AS-Interface bus, operating status of AS-Interface slaves can be monitored using WindLDR on a computer. Output statuses and parameter image (PI) can also be changed using WindLDR.

To open the Monitor AS-Interface Slaves dialog box, from the WindLDR menu bar, select **Online** > **Monitor**. From the WindLDR menu bar, select **Online**, and select **Monitor AS-Interface Slaves** in the pull-down menu.



Dialog Box	Button	Description
Monitor AS-Interface Slaves	**Switch Slaves**	Switches between Slave A screen and Slave B screen.
	Close	Closes the window.
	Help	Displays explanations for functions on the screen.
Slave Status	**Store**	Stores output statuses and parameters to the slave.
	Close	Closes the window.

Change Slave Output Statuses and Parameters

The output statuses and parameter image (PI) of the slaves connected to the AS-Interface master module can be changed. To open the Slave Status dialog box, click the output of a required slave address in the Monitor AS-Interface Slaves dialog box. Then, click the On or Off button to change the statuses of outputs DO0 through DO3 and parameters P0 through P3 as required. Click **Store** to save the changes to the slave module.

If the command is not processed correctly, the error message "AS-Interface Master Error" and an error code will appear. See page 28-34.

The output statuses and parameters cannot be changed in the following cases.

Error Code	Description
1	• An error was found on the expansion I/O bus.
7	• The AS-Interface master module is in local mode.
8	• Attempt was made to change the parameters of a slave which did not exist.

Error Messages

When an error is returned from the AS-Interface master module, WindLDR will display an error message. The error codes and their meanings are given below.

Error Code	Description
1	• An error was found on the expansion I/O bus.
2	• While the AS-Interface master module was in offline mode, attempt was made to perform auto configuration or manual configuration. • An incorrect command was sent.
7	• While slave address 0 existed on the bus, attempt was made to perform auto configuration or manual configuration. • The AS-Interface master module is in local mode.
8	• The slave you are trying to change does not exist. • A slave of the designated new address already exists. • While a standard slave was set at A address, attempt was made to set an A/B slave at B address of the same number. • While an A/B slave was set at B address, attempt was made to set a standard slave at A address of the same number. • Attempt was made to change the parameters of a slave which did not exist.

When a reply message is not returned from the AS-Interface master module, the following error message will be displayed.

SwitchNet Data I/O Port

SwitchNet control units can be used as slaves in the AS-Interface network and are available in ø16mm L6 series and ø22mm HW series. Input signals to the MicroSmart AS-Interface master module are read to internal relays allocated to each input point designated by a slave number and a DI number. Similarly, output signals from the MicroSmart AS-Interface master module are written to internal relays allocated to each output point designated by a slave number and a DO number. When programming a ladder diagram for the MicroSmart, use internal relays allocated to input signals and output signals of SwitchNet control units.

L6 series and HW series SwitchNet control units have slightly different digital I/O data allocations.

L6 Series Digital I/O Data Allocation

Input data is sent from slaves to the AS-Interface master. Output data is sent from the AS-Interface master to slaves.

SwitchNet L6 Series Slave Unit	Used I/O	Input Data (slave send data)				Output Data (slave receive data)			
		DI3	DI2	DI1	DI0	DO3	DO2	DO1	DO0
Pushbutton	1 in	0	X1	1	1	*	—	—	—
Pilot light	1 out	0	0	1	1	*	—	—	X1
Illuminated pushbutton	1 in/1 out	0	X1	1	1	*	—	—	X1
Selector, Key selector, Lever: 2-position	1 in	0	X2	1	1	*	—	—	—
Selector, Key selector, Lever: 3-position	2 in	X3	X3	1	1	*	—	—	—
Illuminated selector: 2-position	1 in/1 out	0	X2	1	1	*	—	—	X1
Illuminated selector: 3-position	2 in/1 out	X3	X3	1	1	*	—	—	X1

Notes:

1. * The AS-Interface master uses bit DO3 for addressing A/B slaves.
2. In the above table, bits marked with X1, X2, and X3 are used for SwitchNet I/O data.
3. X1: When pushbutton is pressed, input data is 1 (on). When not pressed, input data is 0 (off). When output data is 1 (on), LED is on. When output data is 0 (off), LED is off.
4. X2: The input data from 2-position selector, key selector, and illuminated selector switches and 2-position lever switches depend on the operator position as shown below.

2-position Operator	Selector: Left / Right	Lever: Up / Down
Operator Position	Left/Down	Right/Up
DI2	0	1

5. X3: The input data from 3-position selector, key selector, and illuminated selector switches and 3-position lever switches depend on the operator position as shown below.

3-position Operator	Selector: Left / Center / Right	Lever: Up / Center / Down
Operator Position	Left/Down	Center	Right/Up
DI3	0	0	1
DI2	1	0	0

6. Unused input bits DI3 and DI2 are 0 (off), and unused input bits DI1 and DI0 are 1 (on). Slaves ignore unused output data (—) sent from the master.

• Write_Parameter Command

0	0	A4	A3	A2	A1	A0	1	$\overline{\text{Sel}}$ P3	P2	P1	P0	PB	1

• Write_Parameter Settings

LED Brightness	Settings: Output Selection	Settings: Control Data		Remarks
	P2	P1	P0	
100%	1: DO0 0: DO1	1	1	Default
50%		0	1	
25%		1	0	
12.5%		0	0	

HW Series Digital I/O Data Allocation

Input data is sent from slaves to the AS-Interface master. Output data is sent from the AS-Interface master to slaves.

SwitchNet HW Series Slave Unit	Used I/O	Communication Block Mounting Position	Input Data (slave send data)				Output Data (slave receive data)			
			DI3	DI2	DI1	DI0	DO3	DO2	DO1	DO0
Pushbutton	1 in	②	0	X1	1	1	*	—	—	—
Pilot light	1 out	②	0	0	1	1	*	—	—	X1
Illuminated pushbutton	1 in/1 out	②	0	X1	1	1	*	—	—	X1
Selector, Key selector: 2-position	1 in	②	0	X2	1	1	*	—	—	—
Selector, Key selector: 3-position	1 in	①	0	X3	1	1	*	—	—	—
	1 in	②	0	X3	1	1	*	—	—	—
Illuminated selector: 2-position	1 in/1 out	②	0	X2	1	1	*	—	—	X1
Illuminated selector: 3-position	1 in	①	0	X3	1	1	*	—	—	—
	1 in/1 out	②	0	X3	1	1	*	—	—	X1

Notes:

1. ∗ The AS-Interface master uses bit DO3 for addressing A/B slaves.
2. In the above table, bits marked with X1, X2, and X3 are used for SwitchNet I/O data.
3. X1: When pushbutton is pressed, input data is 1 (on). When not pressed, input data is 0 (off). When output data is 1 (on), LED is on. When output data is 0 (off), LED is off.
4. X2: The input data from 2-position selector, key selector, and illuminated selector switches depend on the operator position as shown below.

2-position Operator	Selector (Left / Right)	
Operator Position	Left	Right
DI2	0	1

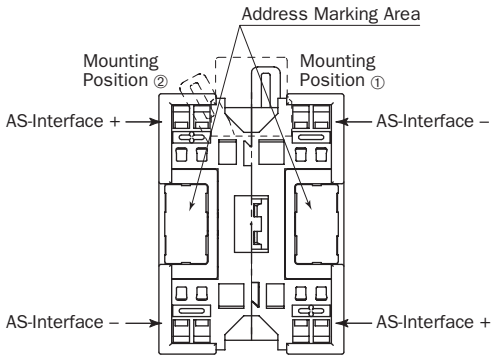
5. X3: The input data from 3-position selector, key selector, and illuminated selector switches depend on the operator position as shown below.

3-position Operator		Selector (Left / Center / Right)		
Operator Position		Left	Center	Right
Communication Block Mounting Position	Input Data Bit			
①	DI2	1	0	0
②	DI2	0	0	1

As shown in the table and figure, 3-position selector, key selector, and illuminated selector switches use two communication blocks. Each communication block must have a unique address, therefore the 3-position selectors require 2 slave addresses.

6. Unused input bits DI3 and DI2 are 0 (off), and unused input bits DI1 and DI0 are 1 (on). Slaves ignore unused output data (—) sent from the master.

Communication Block Mounting Position (Rear View)



On 3-position selector, key selector, and illuminated selector switches, communication blocks ① and ② are mounted in positions shown above.

• Write_Parameter Command

0	0	A4	A3	A2	A1	A0	1	$\overline{\text{Sel}}$ P3	P2	P1	P0	PB	1

• Write_Parameter Settings

LED Brightness	Settings			Remarks
	Output Selection	Control Data		
	P2	P1	P0	
100%	1: DO0 0: DO1	1	1	Default
50%		0	1	
25%		1	0	
12.5%		0	0	

• Internal Relays for SwitchNet Slaves

L6 Series

Slave Number	Pushbutton	Pilot Light	Illuminated Pushbutton		Selector, Key selector, Lever: 2-position
	Input DI2	Output DO0	Input DI2	Output DO0	Input DI2
(Slave 0)	M1302	M1620	M1302	M1620	M1302
Slave 1(A)	M1306	M1624	M1306	M1624	M1306
Slave 2(A)	M1312	M1630	M1312	M1630	M1312
Slave 3(A)	M1316	M1634	M1316	M1634	M1316
Slave 4(A)	M1322	M1640	M1322	M1640	M1322
Slave 5(A)	M1326	M1644	M1326	M1644	M1326
Slave 6(A)	M1332	M1650	M1332	M1650	M1332
Slave 7(A)	M1336	M1654	M1336	M1654	M1336
Slave 8(A)	M1342	M1660	M1342	M1660	M1342
Slave 9(A)	M1346	M1664	M1346	M1664	M1346
Slave 10(A)	M1352	M1670	M1352	M1670	M1352
Slave 11(A)	M1356	M1674	M1356	M1674	M1356
Slave 12(A)	M1362	M1680	M1362	M1680	M1362
Slave 13(A)	M1366	M1684	M1366	M1684	M1366
Slave 14(A)	M1372	M1690	M1372	M1690	M1372
Slave 15(A)	M1376	M1694	M1376	M1694	M1376
Slave 16(A)	M1382	M1700	M1382	M1700	M1382
Slave 17(A)	M1386	M1704	M1386	M1704	M1386
Slave 18(A)	M1392	M1710	M1392	M1710	M1392
Slave 19(A)	M1396	M1714	M1396	M1714	M1396
Slave 20(A)	M1402	M1720	M1402	M1720	M1402
Slave 21(A)	M1406	M1724	M1406	M1724	M1406
Slave 22(A)	M1412	M1730	M1412	M1730	M1412
Slave 23(A)	M1416	M1734	M1416	M1734	M1416
Slave 24(A)	M1422	M1740	M1422	M1740	M1422
Slave 25(A)	M1426	M1744	M1426	M1744	M1426
Slave 26(A)	M1432	M1750	M1432	M1750	M1432
Slave 27(A)	M1436	M1754	M1436	M1754	M1436
Slave 28(A)	M1442	M1760	M1442	M1760	M1442
Slave 29(A)	M1446	M1764	M1446	M1764	M1446
Slave 30(A)	M1452	M1770	M1452	M1770	M1452
Slave 31(A)	M1456	M1774	M1456	M1774	M1456
Slave 1B	M1466	M1784	M1466	M1784	M1466
Slave 2B	M1472	M1790	M1472	M1790	M1472
Slave 3B	M1476	M1794	M1476	M1794	M1476
Slave 4B	M1482	M1800	M1482	M1800	M1482
Slave 5B	M1486	M1804	M1486	M1804	M1486
Slave 6B	M1492	M1810	M1492	M1810	M1492
Slave 7B	M1496	M1814	M1496	M1814	M1496
Slave 8B	M1502	M1820	M1502	M1820	M1502
Slave 9B	M1506	M1824	M1506	M1824	M1506
Slave 10B	M1512	M1830	M1512	M1830	M1512
Slave 11B	M1516	M1834	M1516	M1834	M1516
Slave 12B	M1522	M1840	M1522	M1840	M1522
Slave 13B	M1526	M1844	M1526	M1844	M1526
Slave 14B	M1532	M1850	M1532	M1850	M1532
Slave 15B	M1536	M1854	M1536	M1854	M1536
Slave 16B	M1542	M1860	M1542	M1860	M1542
Slave 17B	M1546	M1864	M1546	M1864	M1546
Slave 18B	M1552	M1870	M1552	M1870	M1552
Slave 19B	M1556	M1874	M1556	M1874	M1556
Slave 20B	M1562	M1880	M1562	M1880	M1562
Slave 21B	M1566	M1884	M1566	M1884	M1566
Slave 22B	M1572	M1890	M1572	M1890	M1572
Slave 23B	M1576	M1894	M1576	M1894	M1576
Slave 24B	M1582	M1900	M1582	M1900	M1582
Slave 25B	M1586	M1904	M1586	M1904	M1586
Slave 26B	M1592	M1910	M1592	M1910	M1592
Slave 27B	M1596	M1914	M1596	M1914	M1596
Slave 28B	M1602	M1920	M1602	M1920	M1602
Slave 29B	M1606	M1924	M1606	M1924	M1606
Slave 30B	M1612	M1930	M1612	M1930	M1612
Slave 31B	M1616	M1934	M1616	M1934	M1616

L6 Series (continued)

Slave Number	Selector, Key selector, Lever: 3-position		Illuminated selector: 2-position		Illuminated selector: 3-position		
	Input DI3	Input DI2	Input DI2	Output DO0	Input DI3	Input DI2	Output DO0
(Slave 0)	M1303	M1302	M1302	M1620	M1303	M1302	M1620
Slave 1(A)	M1307	M1306	M1306	M1624	M1307	M1306	M1624
Slave 2(A)	M1313	M1312	M1312	M1630	M1313	M1312	M1630
Slave 3(A)	M1317	M1316	M1316	M1634	M1317	M1316	M1634
Slave 4(A)	M1323	M1322	M1322	M1640	M1323	M1322	M1640
Slave 5(A)	M1327	M1326	M1326	M1644	M1327	M1326	M1644
Slave 6(A)	M1333	M1332	M1332	M1650	M1333	M1332	M1650
Slave 7(A)	M1337	M1336	M1336	M1654	M1337	M1336	M1654
Slave 8(A)	M1343	M1342	M1342	M1660	M1343	M1342	M1660
Slave 9(A)	M1347	M1346	M1346	M1664	M1347	M1346	M1664
Slave 10(A)	M1353	M1352	M1352	M1670	M1353	M1352	M1670
Slave 11(A)	M1357	M1356	M1356	M1674	M1357	M1356	M1674
Slave 12(A)	M1363	M1362	M1362	M1680	M1363	M1362	M1680
Slave 13(A)	M1367	M1366	M1366	M1684	M1367	M1366	M1684
Slave 14(A)	M1373	M1372	M1372	M1690	M1373	M1372	M1690
Slave 15(A)	M1377	M1376	M1376	M1694	M1377	M1376	M1694
Slave 16(A)	M1383	M1382	M1382	M1700	M1383	M1382	M1700
Slave 17(A)	M1387	M1386	M1386	M1704	M1387	M1386	M1704
Slave 18(A)	M1393	M1392	M1392	M1710	M1393	M1392	M1710
Slave 19(A)	M1397	M1396	M1396	M1714	M1397	M1396	M1714
Slave 20(A)	M1403	M1402	M1402	M1720	M1403	M1402	M1720
Slave 21(A)	M1407	M1406	M1406	M1724	M1407	M1406	M1724
Slave 22(A)	M1413	M1412	M1412	M1730	M1413	M1412	M1730
Slave 23(A)	M1417	M1416	M1416	M1734	M1417	M1416	M1734
Slave 24(A)	M1423	M1422	M1422	M1740	M1423	M1422	M1740
Slave 25(A)	M1427	M1426	M1426	M1744	M1427	M1426	M1744
Slave 26(A)	M1433	M1432	M1432	M1750	M1433	M1432	M1750
Slave 27(A)	M1437	M1436	M1436	M1754	M1437	M1436	M1754
Slave 28(A)	M1443	M1442	M1442	M1760	M1443	M1442	M1760
Slave 29(A)	M1447	M1446	M1446	M1764	M1447	M1446	M1764
Slave 30(A)	M1453	M1452	M1452	M1770	M1453	M1452	M1770
Slave 31(A)	M1457	M1456	M1456	M1774	M1457	M1456	M1774
Slave 1B	M1467	M1466	M1466	M1784	M1467	M1466	M1784
Slave 2B	M1473	M1472	M1472	M1790	M1473	M1472	M1790
Slave 3B	M1477	M1476	M1476	M1794	M1477	M1476	M1794
Slave 4B	M1483	M1482	M1482	M1800	M1483	M1482	M1800
Slave 5B	M1487	M1486	M1486	M1804	M1487	M1486	M1804
Slave 6B	M1493	M1492	M1492	M1810	M1493	M1492	M1810
Slave 7B	M1497	M1496	M1496	M1814	M1497	M1496	M1814
Slave 8B	M1503	M1502	M1502	M1820	M1503	M1502	M1820
Slave 9B	M1507	M1506	M1506	M1824	M1507	M1506	M1824
Slave 10B	M1513	M1512	M1512	M1830	M1513	M1512	M1830
Slave 11B	M1517	M1516	M1516	M1834	M1517	M1516	M1834
Slave 12B	M1523	M1522	M1522	M1840	M1523	M1522	M1840
Slave 13B	M1527	M1526	M1526	M1844	M1527	M1526	M1844
Slave 14B	M1533	M1532	M1532	M1850	M1533	M1532	M1850
Slave 15B	M1537	M1536	M1536	M1854	M1537	M1536	M1854
Slave 16B	M1543	M1542	M1542	M1860	M1543	M1542	M1860
Slave 17B	M1547	M1546	M1546	M1864	M1547	M1546	M1864
Slave 18B	M1553	M1552	M1552	M1870	M1553	M1552	M1870
Slave 19B	M1557	M1556	M1556	M1874	M1557	M1556	M1874
Slave 20B	M1563	M1562	M1562	M1880	M1563	M1562	M1880
Slave 21B	M1567	M1566	M1566	M1884	M1567	M1566	M1884
Slave 22B	M1573	M1572	M1572	M1890	M1573	M1572	M1890
Slave 23B	M1577	M1576	M1576	M1894	M1577	M1576	M1894
Slave 24B	M1583	M1582	M1582	M1900	M1583	M1582	M1900
Slave 25B	M1587	M1586	M1586	M1904	M1587	M1586	M1904
Slave 26B	M1593	M1592	M1592	M1910	M1593	M1592	M1910
Slave 27B	M1597	M1596	M1596	M1914	M1597	M1596	M1914
Slave 28B	M1603	M1602	M1602	M1920	M1603	M1602	M1920
Slave 29B	M1607	M1606	M1606	M1924	M1607	M1606	M1924
Slave 30B	M1613	M1612	M1612	M1930	M1613	M1612	M1930
Slave 31B	M1617	M1616	M1616	M1934	M1617	M1616	M1934

HW Series

Slave Number	Pushbutton	Pilot Light	Illuminated Pushbutton		Selector, Key selector: 2-position
	Input DI2	Output DO0	Input DI2	Output DO0	Input DI2
(Slave 0)	M1302	M1620	M1302	M1620	M1302
Slave 1(A)	M1306	M1624	M1306	M1624	M1306
Slave 2(A)	M1312	M1630	M1312	M1630	M1312
Slave 3(A)	M1316	M1634	M1316	M1634	M1316
Slave 4(A)	M1322	M1640	M1322	M1640	M1322
Slave 5(A)	M1326	M1644	M1326	M1644	M1326
Slave 6(A)	M1332	M1650	M1332	M1650	M1332
Slave 7(A)	M1336	M1654	M1336	M1654	M1336
Slave 8(A)	M1342	M1660	M1342	M1660	M1342
Slave 9(A)	M1346	M1664	M1346	M1664	M1346
Slave 10(A)	M1352	M1670	M1352	M1670	M1352
Slave 11(A)	M1356	M1674	M1356	M1674	M1356
Slave 12(A)	M1362	M1680	M1362	M1680	M1362
Slave 13(A)	M1366	M1684	M1366	M1684	M1366
Slave 14(A)	M1372	M1690	M1372	M1690	M1372
Slave 15(A)	M1376	M1694	M1376	M1694	M1376
Slave 16(A)	M1382	M1700	M1382	M1700	M1382
Slave 17(A)	M1386	M1704	M1386	M1704	M1386
Slave 18(A)	M1392	M1710	M1392	M1710	M1392
Slave 19(A)	M1396	M1714	M1396	M1714	M1396
Slave 20(A)	M1402	M1720	M1402	M1720	M1402
Slave 21(A)	M1406	M1724	M1406	M1724	M1406
Slave 22(A)	M1412	M1730	M1412	M1730	M1412
Slave 23(A)	M1416	M1734	M1416	M1734	M1416
Slave 24(A)	M1422	M1740	M1422	M1740	M1422
Slave 25(A)	M1426	M1744	M1426	M1744	M1426
Slave 26(A)	M1432	M1750	M1432	M1750	M1432
Slave 27(A)	M1436	M1754	M1436	M1754	M1436
Slave 28(A)	M1442	M1760	M1442	M1760	M1442
Slave 29(A)	M1446	M1764	M1446	M1764	M1446
Slave 30(A)	M1452	M1770	M1452	M1770	M1452
Slave 31(A)	M1456	M1774	M1456	M1774	M1456
Slave 1B	M1466	M1784	M1466	M1784	M1466
Slave 2B	M1472	M1790	M1472	M1790	M1472
Slave 3B	M1476	M1794	M1476	M1794	M1476
Slave 4B	M1482	M1800	M1482	M1800	M1482
Slave 5B	M1486	M1804	M1486	M1804	M1486
Slave 6B	M1492	M1810	M1492	M1810	M1492
Slave 7B	M1496	M1814	M1496	M1814	M1496
Slave 8B	M1502	M1820	M1502	M1820	M1502
Slave 9B	M1506	M1824	M1506	M1824	M1506
Slave 10B	M1512	M1830	M1512	M1830	M1512
Slave 11B	M1516	M1834	M1516	M1834	M1516
Slave 12B	M1522	M1840	M1522	M1840	M1522
Slave 13B	M1526	M1844	M1526	M1844	M1526
Slave 14B	M1532	M1850	M1532	M1850	M1532
Slave 15B	M1536	M1854	M1536	M1854	M1536
Slave 16B	M1542	M1860	M1542	M1860	M1542
Slave 17B	M1546	M1864	M1546	M1864	M1546
Slave 18B	M1552	M1870	M1552	M1870	M1552
Slave 19B	M1556	M1874	M1556	M1874	M1556
Slave 20B	M1562	M1880	M1562	M1880	M1562
Slave 21B	M1566	M1884	M1566	M1884	M1566
Slave 22B	M1572	M1890	M1572	M1890	M1572
Slave 23B	M1576	M1894	M1576	M1894	M1576
Slave 24B	M1582	M1900	M1582	M1900	M1582
Slave 25B	M1586	M1904	M1586	M1904	M1586
Slave 26B	M1592	M1910	M1592	M1910	M1592
Slave 27B	M1596	M1914	M1596	M1914	M1596
Slave 28B	M1602	M1920	M1602	M1920	M1602
Slave 29B	M1606	M1924	M1606	M1924	M1606
Slave 30B	M1612	M1930	M1612	M1930	M1612
Slave 31B	M1616	M1934	M1616	M1934	M1616

HW Series (continued)

Slave Number	Selector, Key selector: 3-position	Illuminated selector: 2-position		Illuminated selector: 3-position	
	Input DI2 (Comm. Block ① ②)	Input DI2	Output DO0	Input DI2 (Comm. Block ① ②)	Output DO0 (Comm. Block ②)
(Slave 0)	M1302	M1302	M1620	M1302	M1620
Slave 1(A)	M1306	M1306	M1624	M1306	M1624
Slave 2(A)	M1312	M1312	M1630	M1312	M1630
Slave 3(A)	M1316	M1316	M1634	M1316	M1634
Slave 4(A)	M1322	M1322	M1640	M1322	M1640
Slave 5(A)	M1326	M1326	M1644	M1326	M1644
Slave 6(A)	M1332	M1332	M1650	M1332	M1650
Slave 7(A)	M1336	M1336	M1654	M1336	M1654
Slave 8(A)	M1342	M1342	M1660	M1342	M1660
Slave 9(A)	M1346	M1346	M1664	M1346	M1664
Slave 10(A)	M1352	M1352	M1670	M1352	M1670
Slave 11(A)	M1356	M1356	M1674	M1356	M1674
Slave 12(A)	M1362	M1362	M1680	M1362	M1680
Slave 13(A)	M1366	M1366	M1684	M1366	M1684
Slave 14(A)	M1372	M1372	M1690	M1372	M1690
Slave 15(A)	M1376	M1376	M1694	M1376	M1694
Slave 16(A)	M1382	M1382	M1700	M1382	M1700
Slave 17(A)	M1386	M1386	M1704	M1386	M1704
Slave 18(A)	M1392	M1392	M1710	M1392	M1710
Slave 19(A)	M1396	M1396	M1714	M1396	M1714
Slave 20(A)	M1402	M1402	M1720	M1402	M1720
Slave 21(A)	M1406	M1406	M1724	M1406	M1724
Slave 22(A)	M1412	M1412	M1730	M1412	M1730
Slave 23(A)	M1416	M1416	M1734	M1416	M1734
Slave 24(A)	M1422	M1422	M1740	M1422	M1740
Slave 25(A)	M1426	M1426	M1744	M1426	M1744
Slave 26(A)	M1432	M1432	M1750	M1432	M1750
Slave 27(A)	M1436	M1436	M1754	M1436	M1754
Slave 28(A)	M1442	M1442	M1760	M1442	M1760
Slave 29(A)	M1446	M1446	M1764	M1446	M1764
Slave 30(A)	M1452	M1452	M1770	M1452	M1770
Slave 31(A)	M1456	M1456	M1774	M1456	M1774
Slave 1B	M1466	M1466	M1784	M1466	M1784
Slave 2B	M1472	M1472	M1790	M1472	M1790
Slave 3B	M1476	M1476	M1794	M1476	M1794
Slave 4B	M1482	M1482	M1800	M1482	M1800
Slave 5B	M1486	M1486	M1804	M1486	M1804
Slave 6B	M1492	M1492	M1810	M1492	M1810
Slave 7B	M1496	M1496	M1814	M1496	M1814
Slave 8B	M1502	M1502	M1820	M1502	M1820
Slave 9B	M1506	M1506	M1824	M1506	M1824
Slave 10B	M1512	M1512	M1830	M1512	M1830
Slave 11B	M1516	M1516	M1834	M1516	M1834
Slave 12B	M1522	M1522	M1840	M1522	M1840
Slave 13B	M1526	M1526	M1844	M1526	M1844
Slave 14B	M1532	M1532	M1850	M1532	M1850
Slave 15B	M1536	M1536	M1854	M1536	M1854
Slave 16B	M1542	M1542	M1860	M1542	M1860
Slave 17B	M1546	M1546	M1864	M1546	M1864
Slave 18B	M1552	M1552	M1870	M1552	M1870
Slave 19B	M1556	M1556	M1874	M1556	M1874
Slave 20B	M1562	M1562	M1880	M1562	M1880
Slave 21B	M1566	M1566	M1884	M1566	M1884
Slave 22B	M1572	M1572	M1890	M1572	M1890
Slave 23B	M1576	M1576	M1894	M1576	M1894
Slave 24B	M1582	M1582	M1900	M1582	M1900
Slave 25B	M1586	M1586	M1904	M1586	M1904
Slave 26B	M1592	M1592	M1910	M1592	M1910
Slave 27B	M1596	M1596	M1914	M1596	M1914
Slave 28B	M1602	M1602	M1920	M1602	M1920
Slave 29B	M1606	M1606	M1924	M1606	M1924
Slave 30B	M1612	M1612	M1930	M1612	M1930
Slave 31B	M1616	M1616	M1934	M1616	M1934

Note: Three-position selector, key selector, and illuminated selector switches use two communication blocks, therefore require two slave addresses. For the communication block mounting position, see page 28-36.

29: TROUBLESHOOTING

Introduction

This chapter describes the procedures to determine the cause of trouble and actions to be taken when any trouble occurs while operating the MicroSmart.

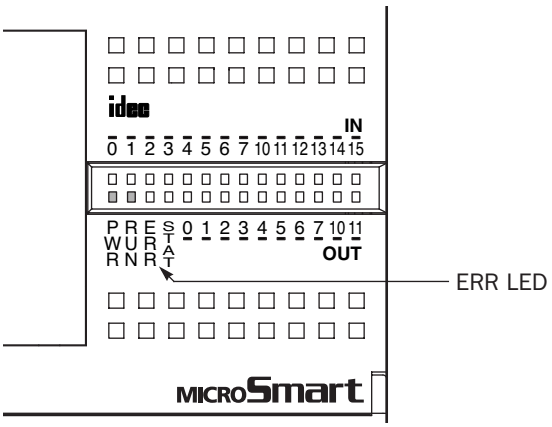
The MicroSmart has self-diagnostic functions to prevent the spread of troubles if any trouble should occur. In case of any trouble, follow the troubleshooting procedures to determine the cause and to correct the error.

Errors are checked in various stages. While editing a user program on WindLDR, incorrect operands and other data are rejected. User program syntax errors are found during compilation on WindLDR. When an incorrect program is downloaded to the MicroSmart, user program syntax errors are still checked. Errors are also checked at starting and during operation of the MicroSmart. When an error occurs, the error is reported by turning on the ERR LED on the MicroSmart and an error message can be viewed on WindLDR. Error codes can also be read on the HMI module.

ERR LED

The MicroSmart CPU module has an error indicator ERR. When an error occurs in the MicroSmart CPU module, the ERR LED is lit. See the trouble shooting diagrams on page 29-10.

For error causes to turn on the ERR LED, see page 29-4.



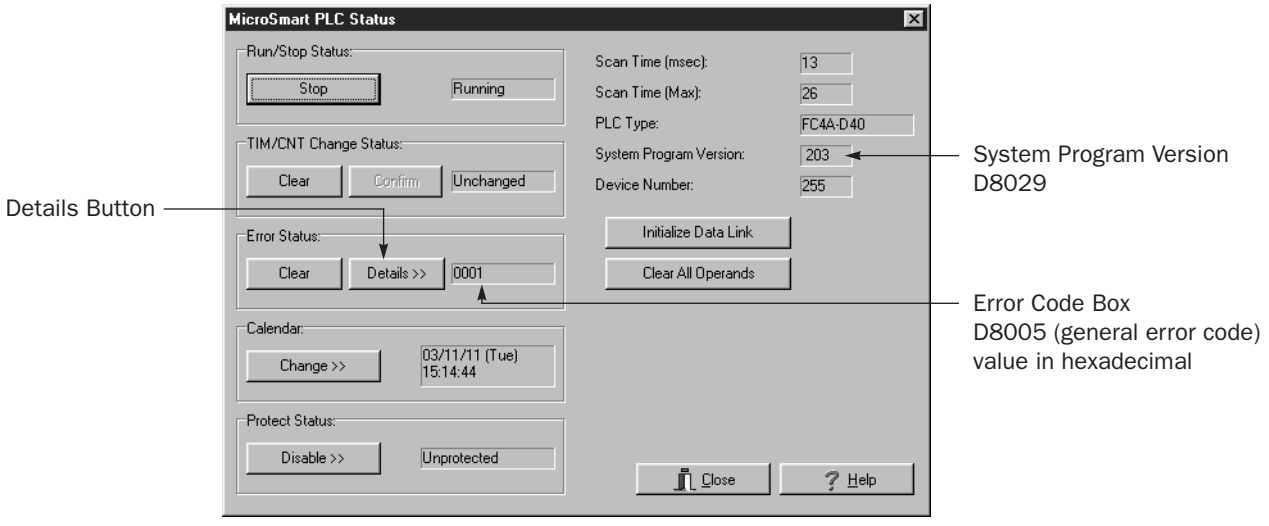
Reading Error Data

When any error occurs during the MicroSmart operation, the error codes and messages can be read out using WindLDR on a computer.

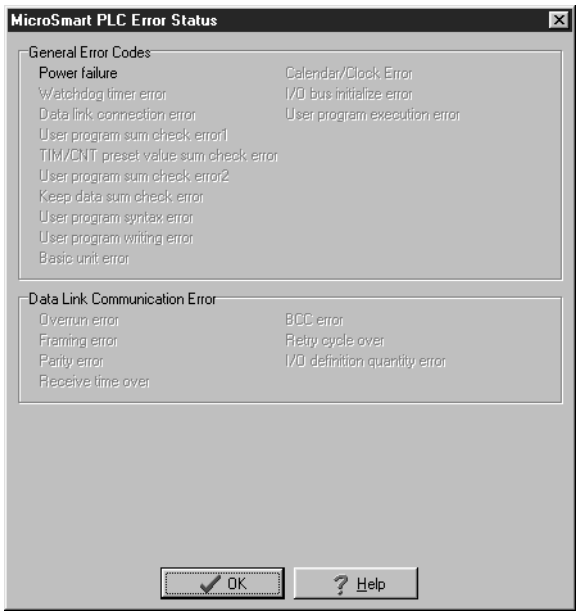
Monitoring WindLDR

1. From the WindLDR menu bar, select **Online > Monitor**. The monitor mode is enabled.
2. From the WindLDR menu bar, select **Online > PLC Status**. The PLC Status dialog box appears.

The general error code stored in special data register D8005 is displayed in the error code box.



3. Under the Error Status in the PLC Status dialog box, click the **Details** button. The PLC Error Status screen appears.

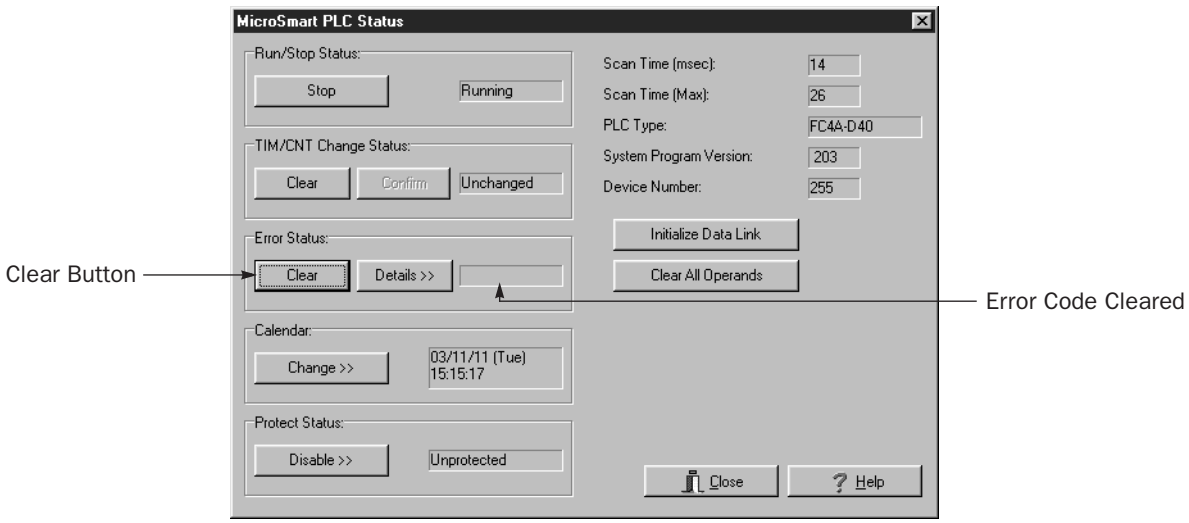


Clearing Error Codes from WindLDR

After removing the cause of the error, clear the error code using the following procedure:

1. From the WindLDR menu bar, select **Online > Monitor**. The monitor mode is enabled.
2. From the WindLDR menu bar, select **Online > PLC Status**. The PLC Status dialog box appears.
3. Under the **Error Status** in the PLC Status dialog box, click the **Clear** button.

This procedure clears the error code from special data register D8005 (general error code), and the error code is cleared from the PLC Status dialog box.



Special Data Registers for Error Information

Two data registers are assigned to store information on errors.

D8005	General Error Code
D8006	User Program Execution Error Code

General Error Codes

The general error code is stored to special data register D8005 (general error code).

When monitoring the PLC status using WindLDR, the error code is displayed in the error code box under the Error Status in the PLC Status dialog box using four hexadecimal digits 0 through F. Each digit of the error code indicates a different set of conditions requiring attention. After the error code is cleared as described on the preceding page, the error code box is left blank.

For example, the error code may read out "0021." This indicates two conditions requiring attention, "User program sum check error" from the third chart and "Power failure" from the fourth chart. If the read-out displays "000D," this indicates three conditions exist from only the fourth chart.

Error Code: Most Significant Digit	**F000**	**E000**	**D000**	**C000**	**B000**	**A000**	**9000**	**8000**	**7000**	**6000**	**5000**	**4000**	**3000**	**2000**	**1000**	**0000**
I/O bus initialize error														X		

Error Code: 2nd Digit from Left	**F00**	**E00**	**D00**	**C00**	**B00**	**A00**	**900**	**800**	**700**	**600**	**500**	**400**	**300**	**200**	**100**	**000**
User program writing error	X		X		X		X		X		X		X		X	
CPU module error	X	X			X	X			X	X			X	X		
Clock IC error	X	X	X	X					X	X	X	X				

Error Code: 3rd Digit from Left	**F0**	**E0**	**D0**	**C0**	**B0**	**A0**	**90**	**80**	**70**	**60**	**50**	**40**	**30**	**20**	**10**	**00**
TIM/CNT preset value sum check error	X		X		X		X		X		X		X		X	
User program RAM sum check error	X	X			X	X			X	X			X	X		
Keep data error	X	X	X	X					X	X	X	X				
User program syntax error	X	X	X	X	X	X	X	X								

Error Code: Least Significant Digit	**F**	**E**	**D**	**C**	**B**	**A**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Power failure	X		X		X		X		X		X		X		X	
Watchdog timer error	X	X			X	X			X	X			X	X		
Data link connection error	X	X	X	X					X	X	X	X				
User program EEPROM sum check error	X	X	X	X	X	X	X	X								

CPU Module Operating Status, Output, and ERR LED during Errors

Error Items	Operating Status	Output	ERR LED	Checked at
Power failure	Stop	OFF	OFF	Any time
Watchdog timer error	Stop	OFF	ON	Any time
Data link connection error	Stop	OFF	OFF	Initializing data link
User program EEPROM sum check error	Stop	OFF	ON	Starting operation
TIM/CNT preset value sum check error	Maintained	Maintained	OFF	Starting operation
User program RAM sum check error	Stop *1	OFF	ON	During operation
Keep data error	Maintained/ Stop *2	Maintained/ OFF *2	OFF	Turning power on
User program syntax error	Stop	OFF	ON	Downloading user program
User program writing error	Stop	OFF	ON	Downloading user program
CPU module error	Stop	OFF	ON	Turning power on
Clock IC error	Maintained	Maintained	ON	Any time
I/O bus initialize error	Stop	OFF	ON	Turning power on
User program execution error	Maintained	Maintained	ON	Executing user program

*1: When a program RAM sum check error occurs, operation is stopped momentarily for reloading the user program. After completing the reloading, operation resumes.

*2: Operation starts to run and outputs are turned on or off according to the user program as default, but it is also possible to stop operation and turn off outputs using the Function Area Settings on WindLDR. See page 5-3.

Error Causes and Actions

0001h: Power Failure

This error indicates when the power supply is lower than the specified voltage. This error is also recorded when the power is turned off. Clear the error code using the HMI module or WindLDR on a computer.

0002h: Watchdog Timer Error

The watchdog timer monitors the time required for one program cycle (scan time). When the time exceeds approximately 293 msec, the watchdog timer indicates an error. Clear the error code using the HMI module or WindLDR on a computer. If this error occurs frequently, the MicroSmart CPU module has to be replaced.

0004h: Data Link Connection Error

This error indicates that the Function Area Settings for data link communication are incorrect or the cable is not connected correctly. Make sure that slave stations are set to station numbers 1 through 31 using WindLDR. No duplication of station numbers is allowed. See page 25-7.

To correct this error, make corrections in the Function Area Settings and download the user program to each station, or connect the cable correctly. Turn power off and on again for the slave station. Then take one of the following actions:

- Turn power off and on for the master station.
- Initialize data link communication for the master station using WindLDR on a computer. See page 25-11.
- Turn on special internal relay M8007 (data link communication initialize flag) at the master station. See page 25-6.

0008h: User Program EEPROM Sum Check Error

The user program stored in the MicroSmart CPU module EEPROM is broken. Download a correct user program to the MicroSmart, and clear the error code using the HMI module or WindLDR on a computer.

When a memory cartridge is installed on the CPU module, the user program in the memory cartridge is checked.

0010h: Timer/Counter Preset Value Sum Check Error

The execution data of timer/counter preset values is broken. The timer/counter preset values are initialized to the values of the user program automatically. Note that changed preset values are cleared and that the original values are restored. Clear the error code using the HMI module or WindLDR on a computer.

0020h: User Program RAM Sum Check Error

The data of the user program compile area in the MicroSmart CPU module RAM is broken.When this error occurs, the user program is recompiled automatically, and the timer/counter preset values and expansion data register preset values are initialized to the values of the user program. Note that changed preset values are cleared and that the original values are restored. Clear the error code using the HMI module or WindLDR on a computer.

0040h: Keep Data Error

This error indicates that the data designated to be maintained during power failure is broken because of memory backup failure. Note that the "keep" data of internal relays and shift registers are cleared. Data of counters and data registers are also cleared. Clear the error code using the HMI module or WindLDR on a computer.

If this error occurs in a short period of power interruption after the battery has been charged as specified, the battery is defective and the CPU module has to be replaced.

0080h: User Program Syntax Error

This error indicates that the user program has a syntax error. Correct the user program, and download the corrected user program to the MicroSmart. The error code is cleared when a correct user program is transferred.

0100h: User Program Writing Error

This error indicates a failure of writing into the MicroSmart CPU module EEPROM when downloading a user program. The error code is cleared when writing into the EEPROM is completed successfully. If this error occurs frequently, the MicroSmart CPU module has to be replaced.

When a memory cartridge is installed on the CPU module, writing into the memory cartridge is checked.

0200h: CPU Module Error

This error is issued when the EEPROM is not found. When this error occurred, turn power off and on. Clear the error code using the HMI module or WindLDR on a computer. If this error occurs frequently, the MicroSmart CPU module has to be replaced.

0400h: Clock IC Error

This error indicates that the real time calendar/clock in the clock cartridge has lost clock backup data or has an error caused by invalid clock data.

Clear the error code and set the calendar/clock data using the HMI module or WindLDR on a computer. The clock cartridge will recover from the error. If the error continues, the clock cartridge has to be replaced. See Troubleshooting Diagram on page 29-21.

2000h: I/O Bus Initialize Error

This error indicates that an I/O module has a fault. If this error occurs frequently or normal I/O function is not restored automatically, the I/O module has to be replaced.

User Program Execution Error

This error indicates that invalid data is found during execution of a user program. When this error occurs, the ERR LED and special internal relay M8004 (user program execution error) are also turned on. The detailed information of this error can be viewed from the error code stored in special data register D8006 (user program execution error code).

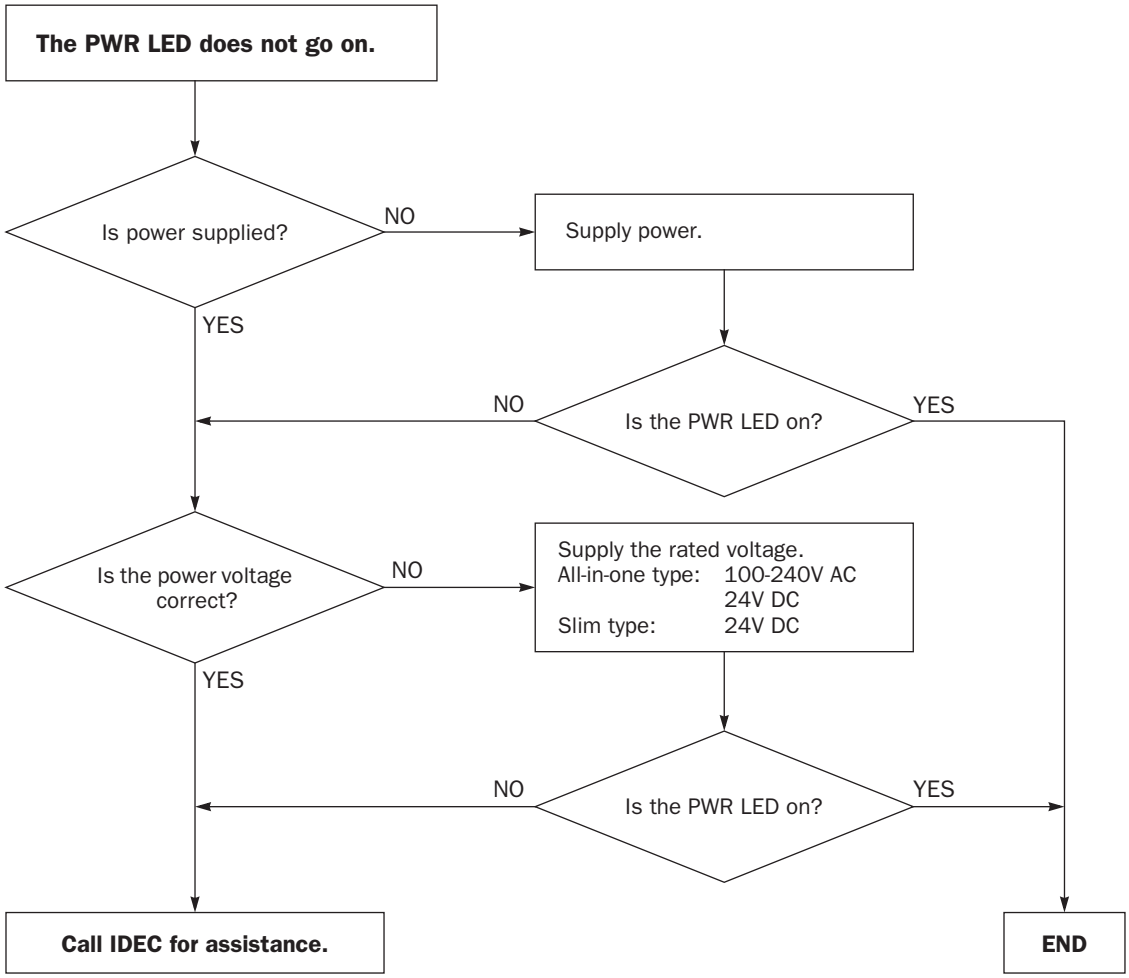
User Program Execution Error Code (D8006)	Error Details
1	Source/destination operand is out of range
2	MUL result is out of data type range.
3	DIV result is out of data type range, or division by 0.
4	BCDLS has S1 or S1+1 exceeding 9999.
5	HTOB(W) has S1 exceeding 9999.
6	BTOH has any digit of S1 exceeding 9.
7	HTOA/ATOH/BTOA/ATOB has quantity of digits to convert out of range.
8	ATOH/ATOB has non-ASCII data for S1 through S1+4.
9	WKTIM has S1, S2, and S3 exceeding the valid range. S1: 0 through 127 S2/S3: Hour data 0 through 23, minute data 0 through 59 S2/S3 can be 10000. WKTBL instruction is not programmed or WKTIM instruction is executed before WKTBL instruction when 1 (additional days in the week table) or 2 (skip days in the week table) is set for MODE in the WKTIM instruction.
10	WKTBL has S1 through Sn out of range. Month: 01 through 12 Day: 01 through 31
11	DGRD data exceeds 65535 with BCD5 digits selected.
12	CVXTY/CVYTX is executed without matching XYFS.
13	CVXTY/CVYTX has S2 exceeding the value specified in XYFS.
14	Label in LJMP/LCAL is not found.
15	TXD/RXD is executed while the RS232C port 1 or 2 is *not* set to user communication mode.
16	PID instruction execution error (see page 21-4).
17	Preset value is written to a timer or counter whose preset value is designated with a data register.
18	Attempt was made to execute an instruction that cannot be used in an interrupt program: SOTU, SOTD, TML, TIM, TMH, TMS, CNT, CDP, CUD, SFR, SFRN, ROOT, WKTIM, WKTBL, DISP, DGRD, TXD1, TXD2, RXD1, RXD2, DI, EI, XYFS, CVXTY, CVYTX, PULS1, PULS2, PWM1, PWM2, RAMP, ZRN1, ZRN2, PID, DTML, DTIM, DTMH, DTMS, and TTIM (see page 5-21).
19	Attempt was made to execute an instruction that is not available for the PLC.
20	PULS1, PULS2, PWM1, PWM2, RAMP, ZRN1, or ZRN2 has an invalid value in control registers.
21	DECO has S1 exceeding 255.
22	BCNT has S2 exceeding 256.
23	ICMP>= has S1 < S3.
24	Interrupt program execution time exceeds 670 µs when using a timer interrupt
25	BCDLS has S2 exceeding 7.
26	DI or EI is executed when interrupt input or timer interrupt is not programmed in the Function Area Settings.
27	Work area is broken when using DTML, DTIM, DTMH, DTMS, or TTIM.

Troubleshooting Diagrams

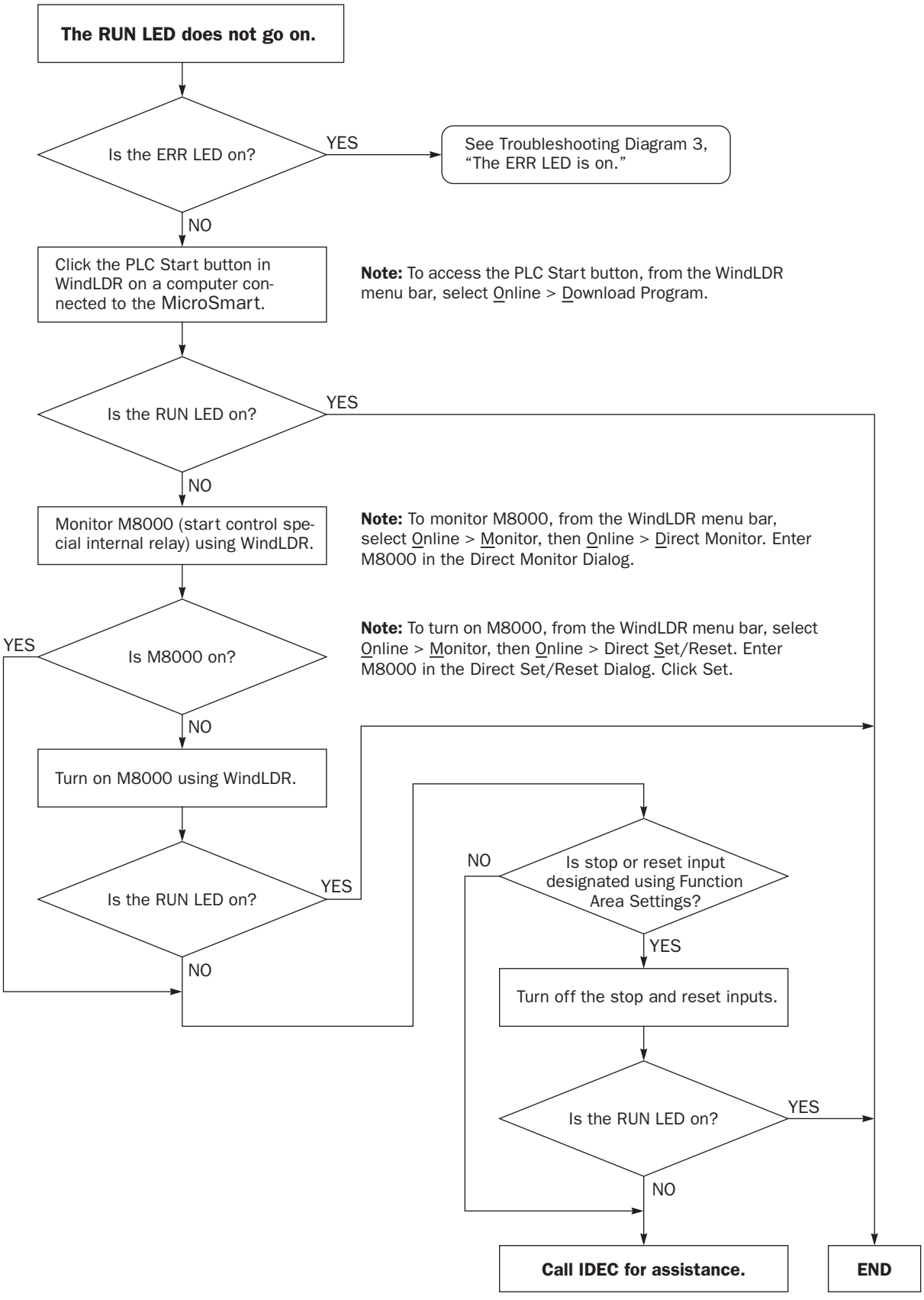
When one of the following problems is encountered, see the trouble shooting diagrams on the following pages.

Problem	Troubleshooting Diagram
The PWR LED does not go on.	Diagram 1
The RUN LED does not go on.	Diagram 2
The ERR LED is on.	Diagram 3
Input does not operate normally.	Diagram 4
Output does not operate normally.	Diagram 5
Communication between WindLDR on a computer and the MicroSmart is not possible.	Diagram 6
Cannot stop or reset operation.	Diagram 7
Data link communication is impossible.	Diagram 8
Data is not transmitted at all in the user communication mode.	Diagram 9
Data is not transmitted correctly in the user communication mode.	Diagram 10
Data is not received at all in the user communication mode.	Diagram 11
Data is not received correctly in the user communication mode.	Diagram 12
The interrupt/catch input cannot receive short pulses.	Diagram 13
The calendar/clock does not operate correctly.	Diagram 14

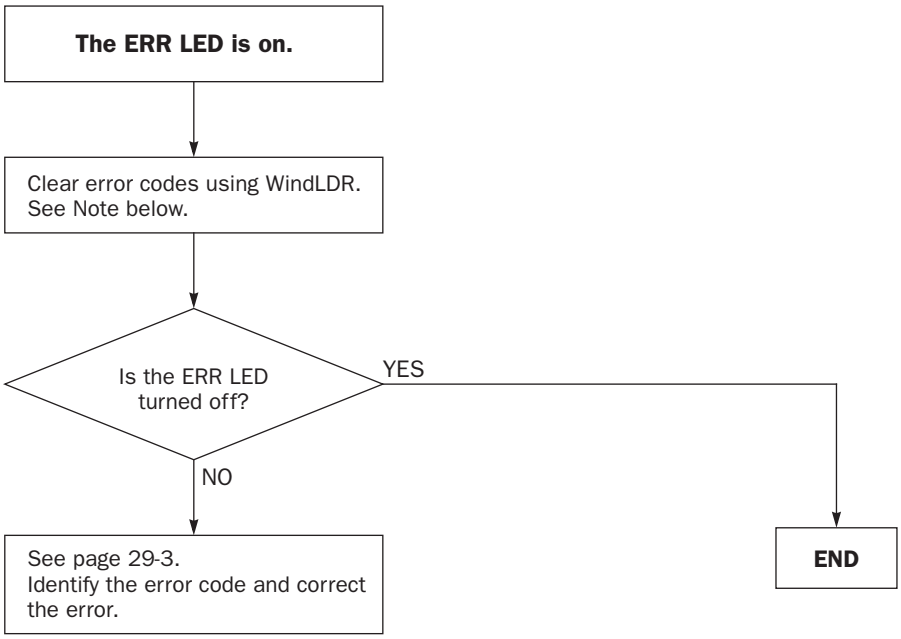
Troubleshooting Diagram 1



Troubleshooting Diagram 2

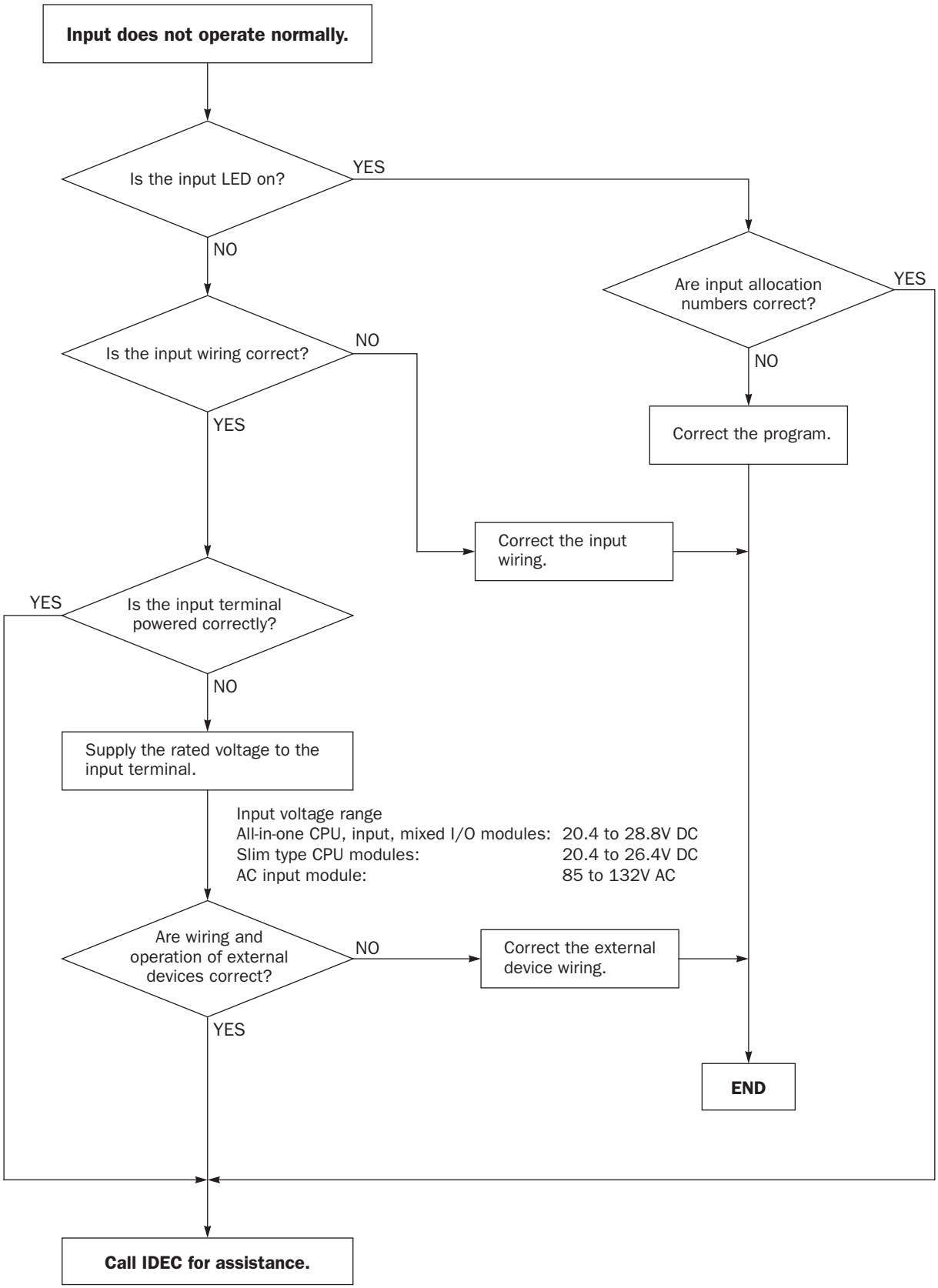


Troubleshooting Diagram 3

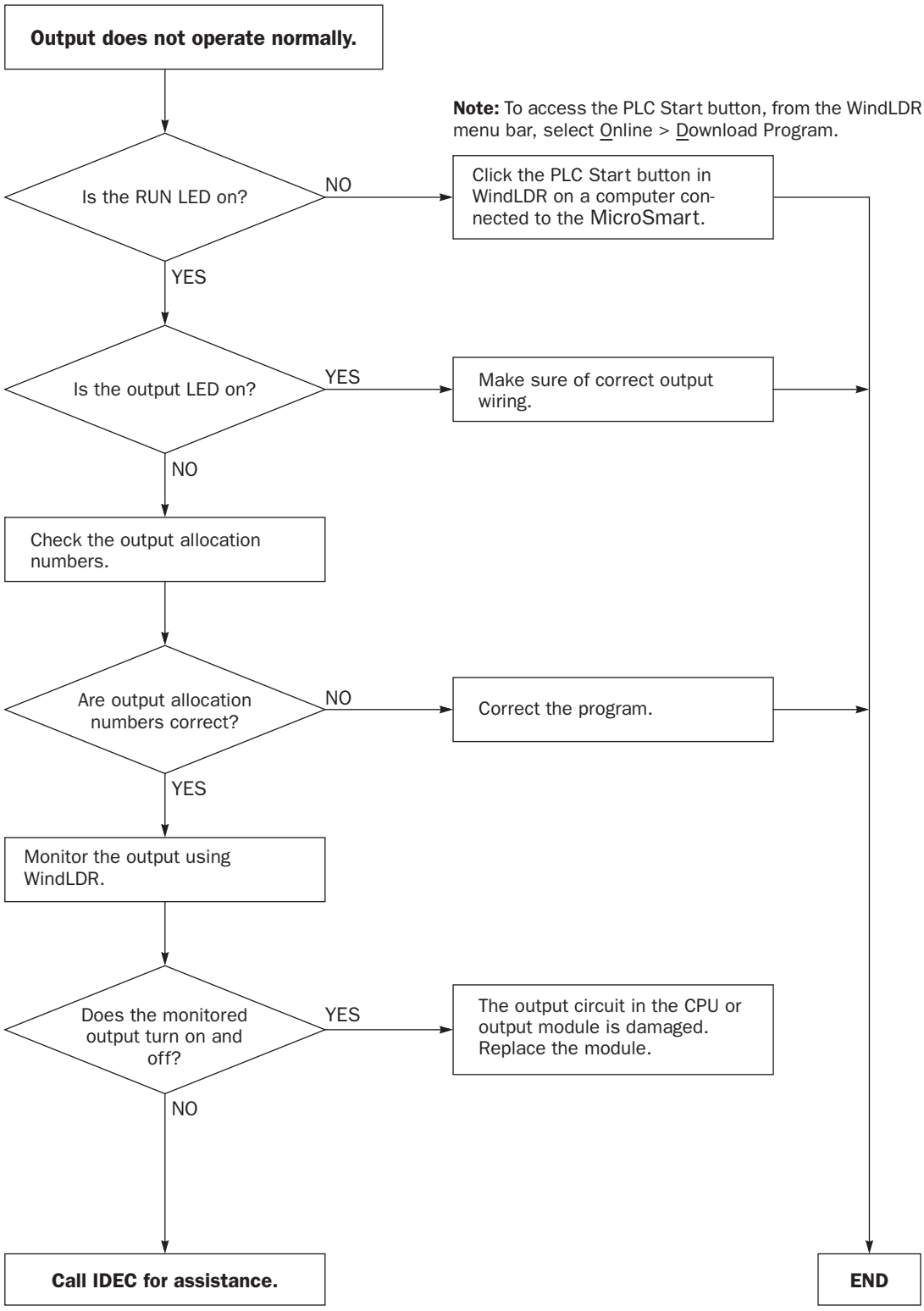


Note: Temporary errors can be cleared to restore normal operation by clearing error codes from WindLDR. See page 29-2.

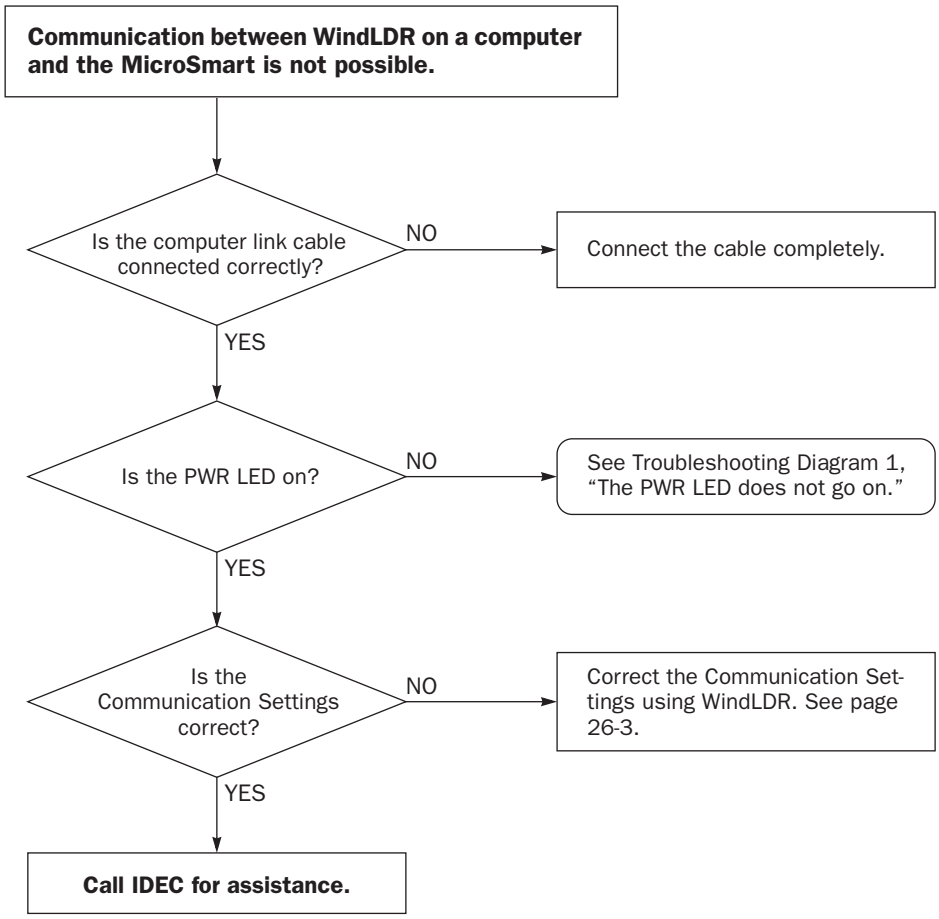
Troubleshooting Diagram 4



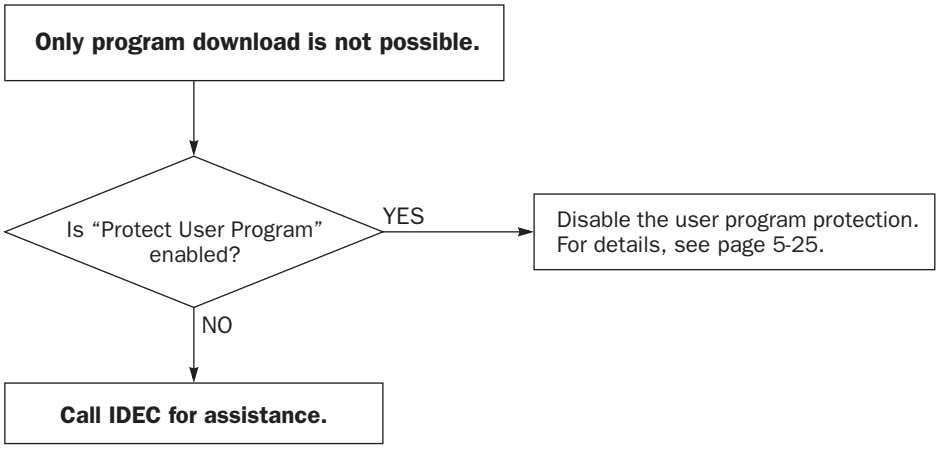
Troubleshooting Diagram 5



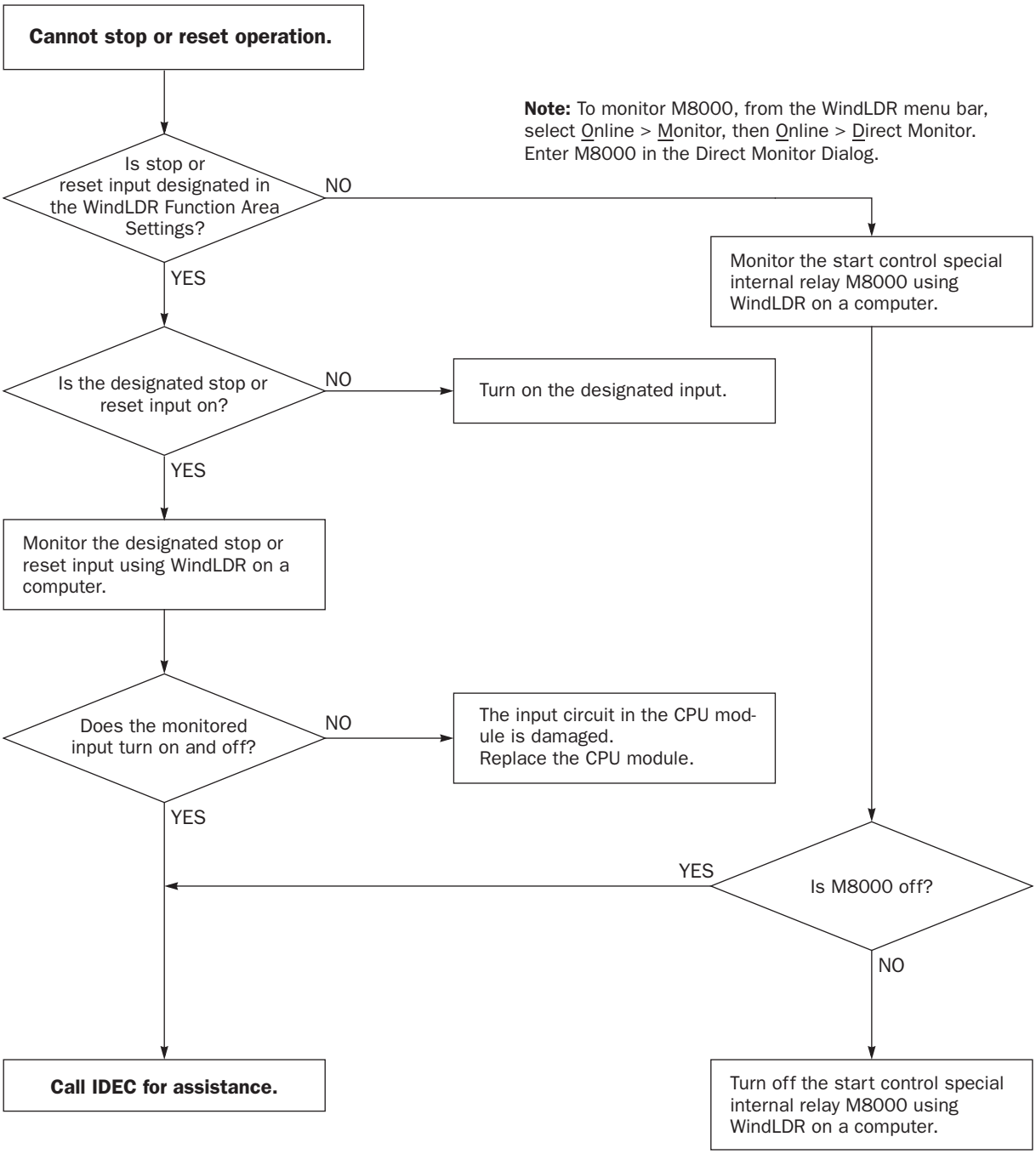
Troubleshooting Diagram 6



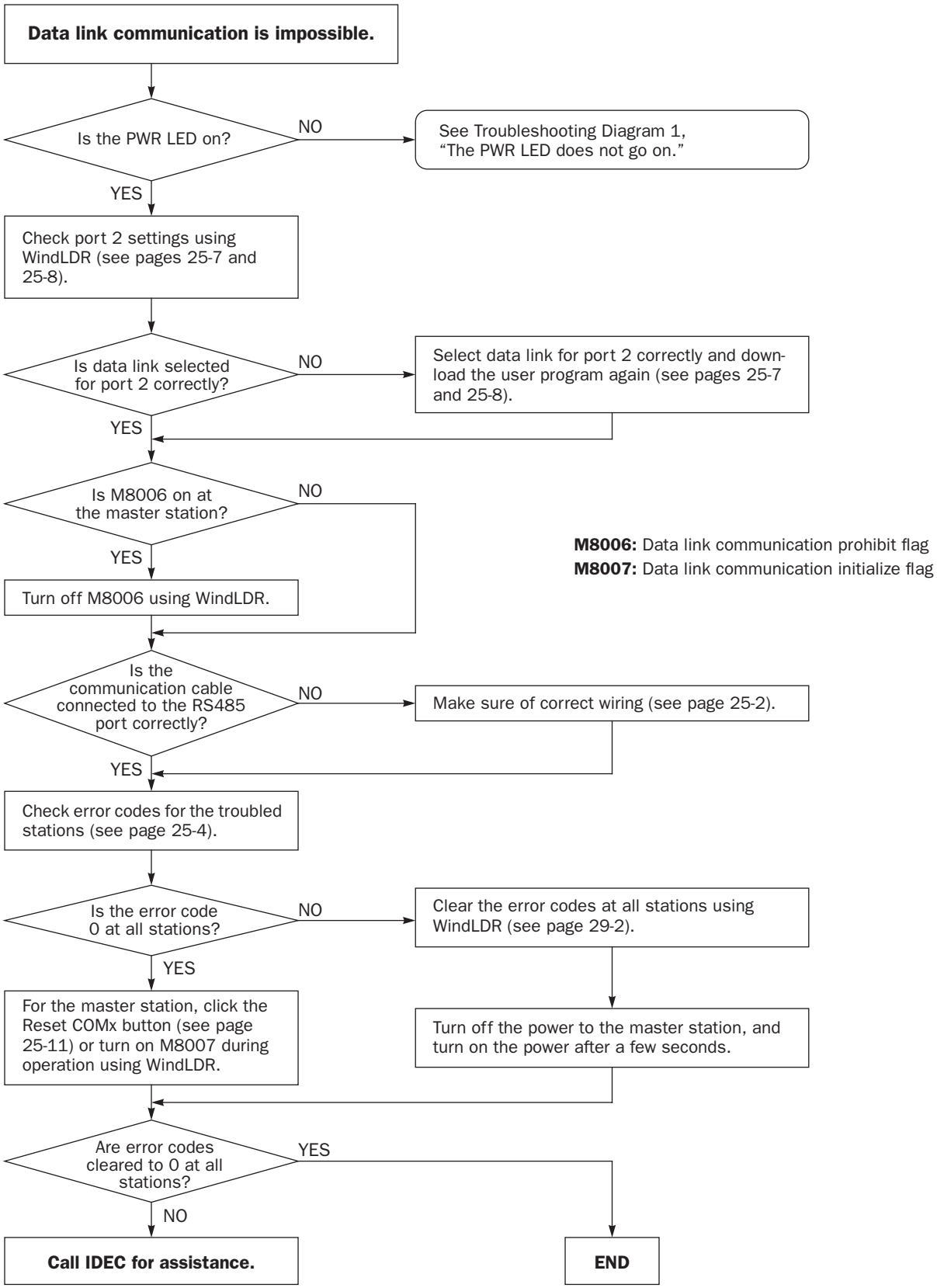
When only program download is not possible:



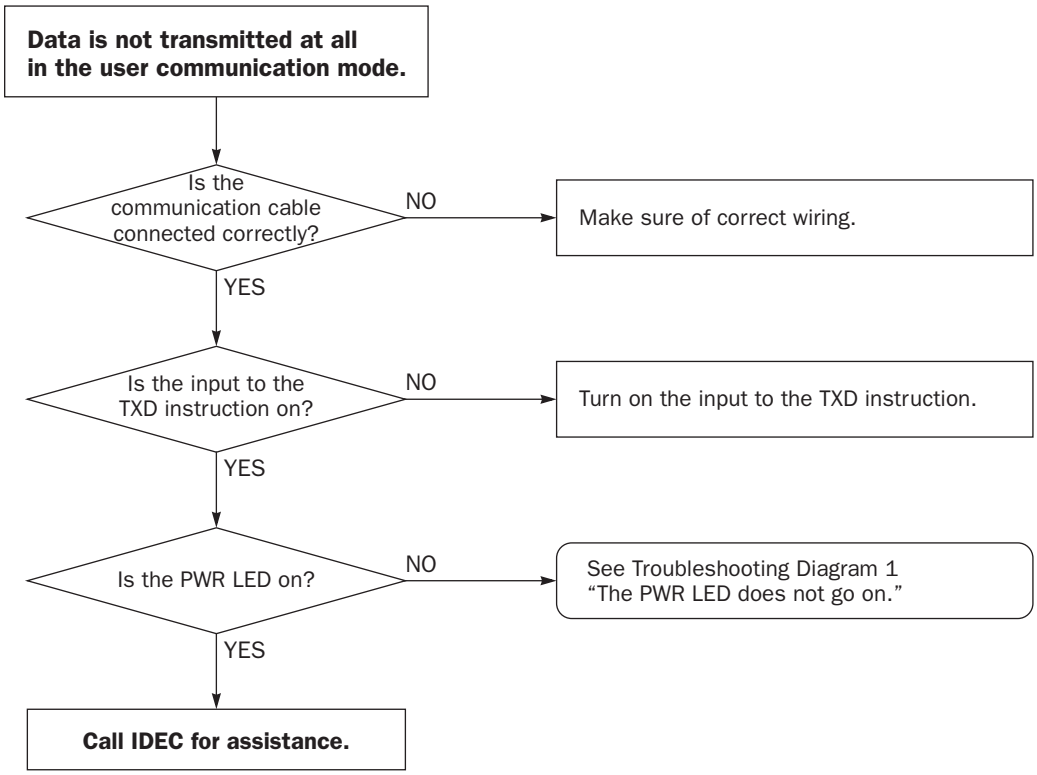
Troubleshooting Diagram 7



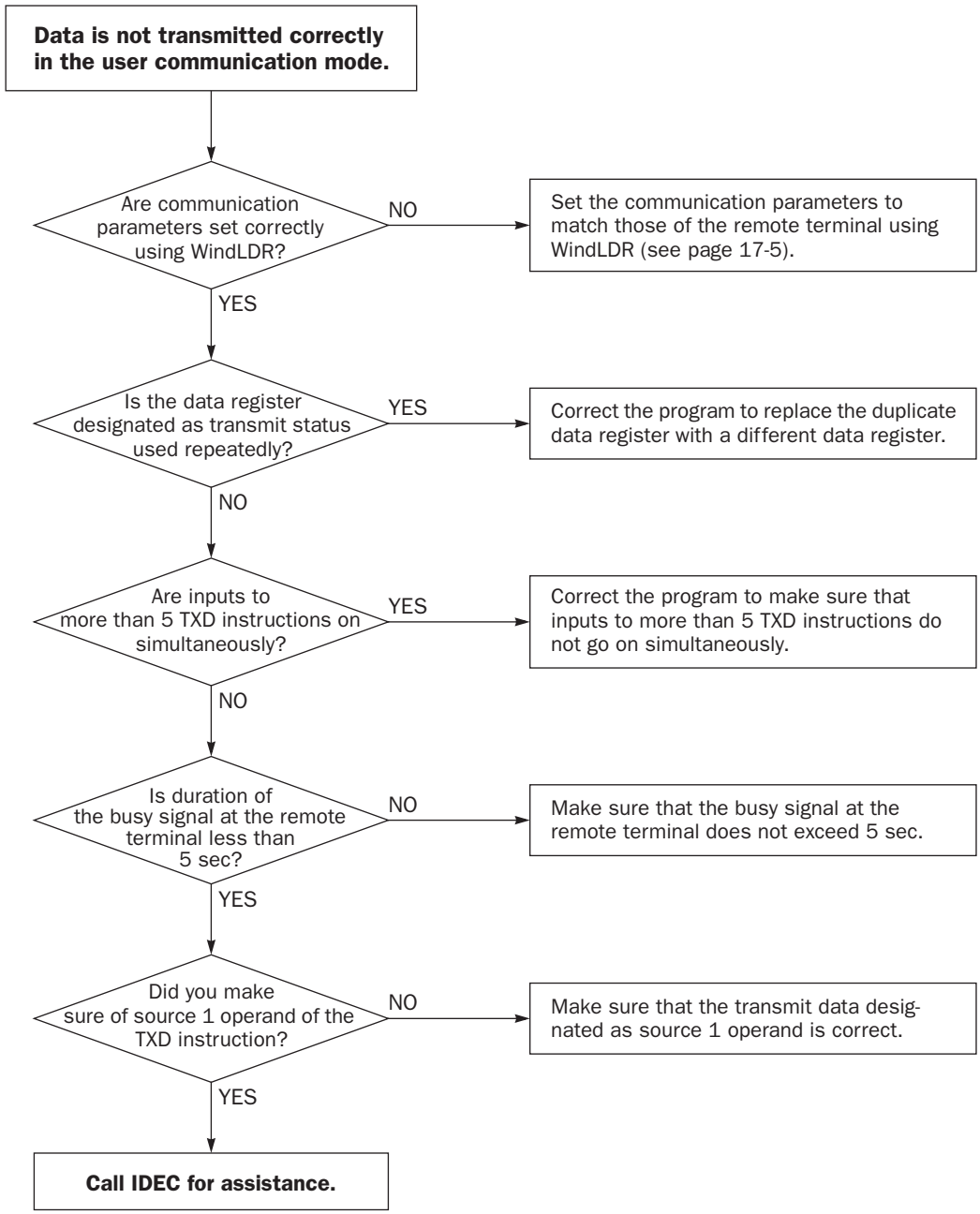
Troubleshooting Diagram 8



Troubleshooting Diagram 9

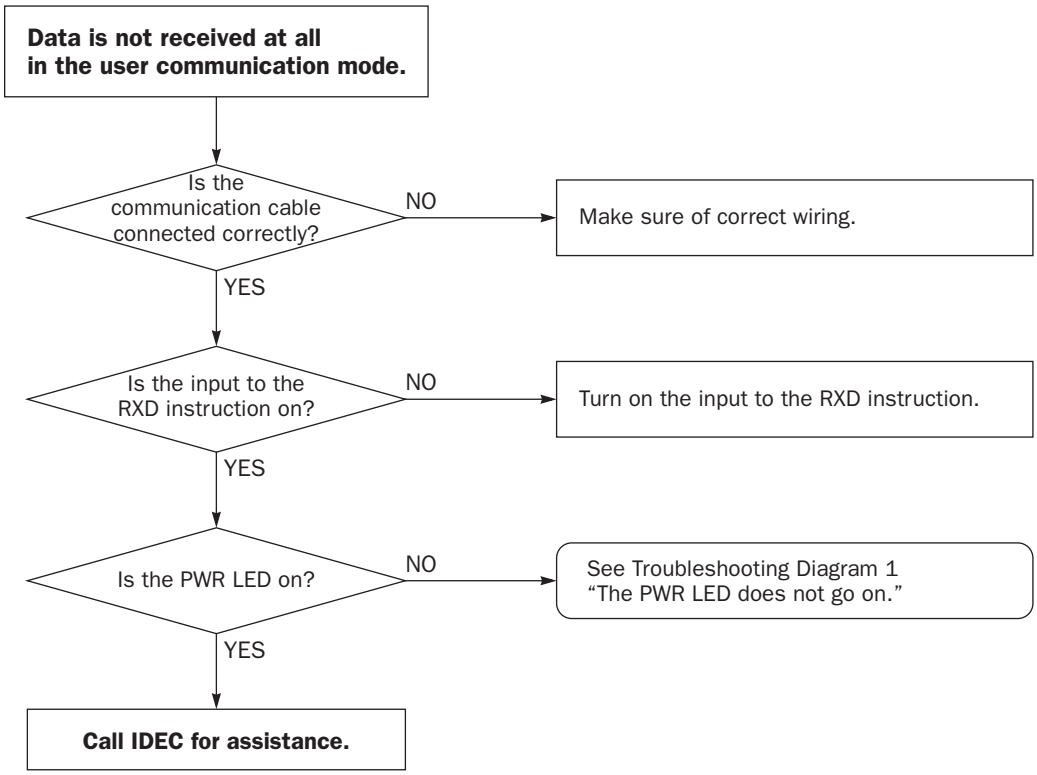


Troubleshooting Diagram 10

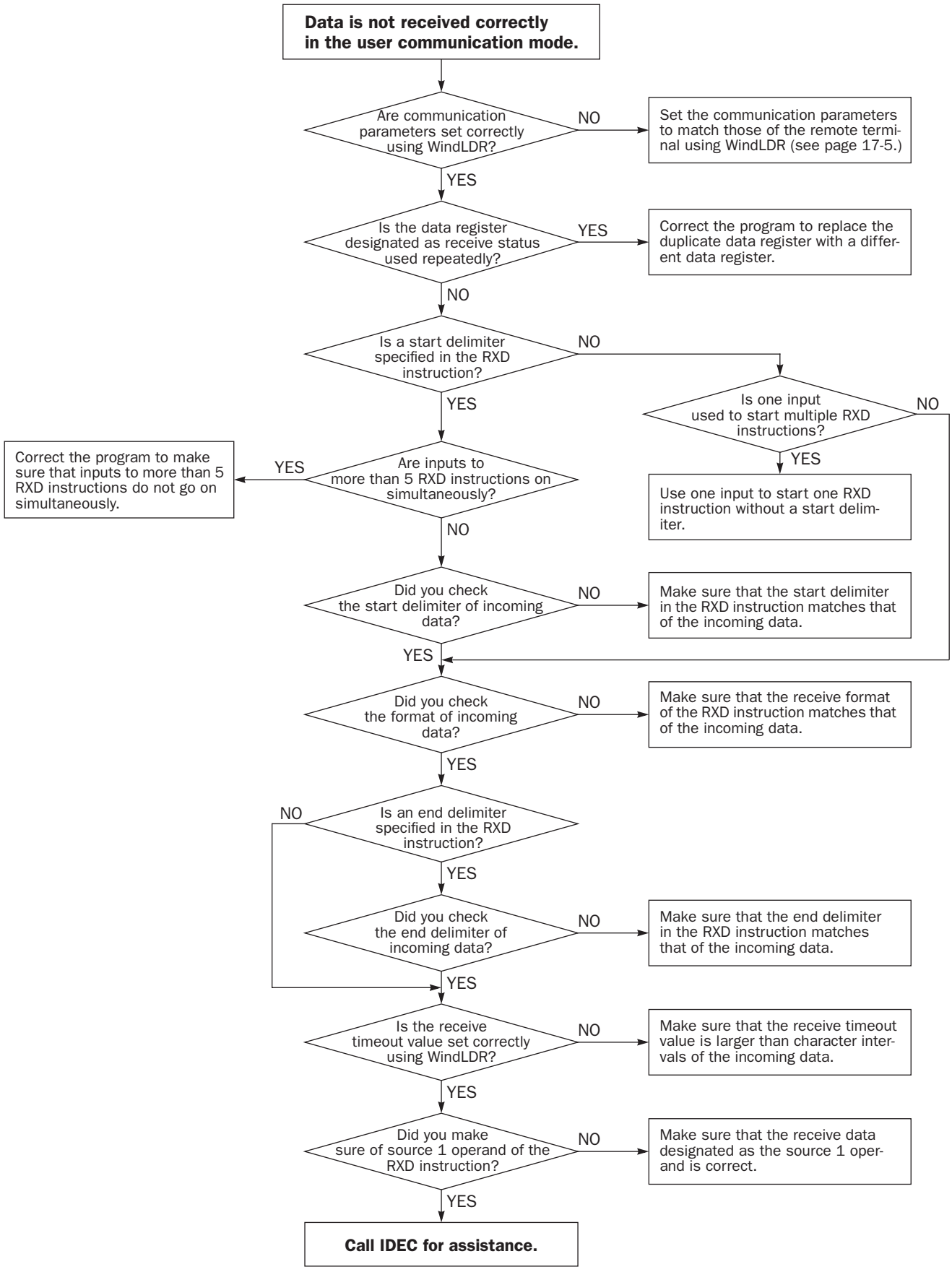


When the user communication still has a problem after completing the above procedure, also perform the procedure of Diagram 9 described on the preceding page.

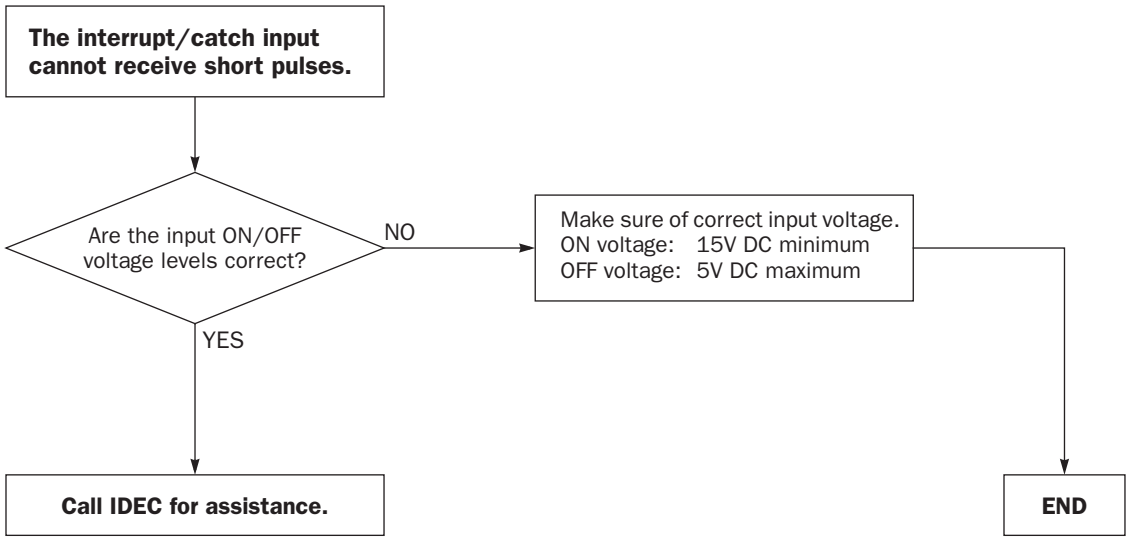
Troubleshooting Diagram 11



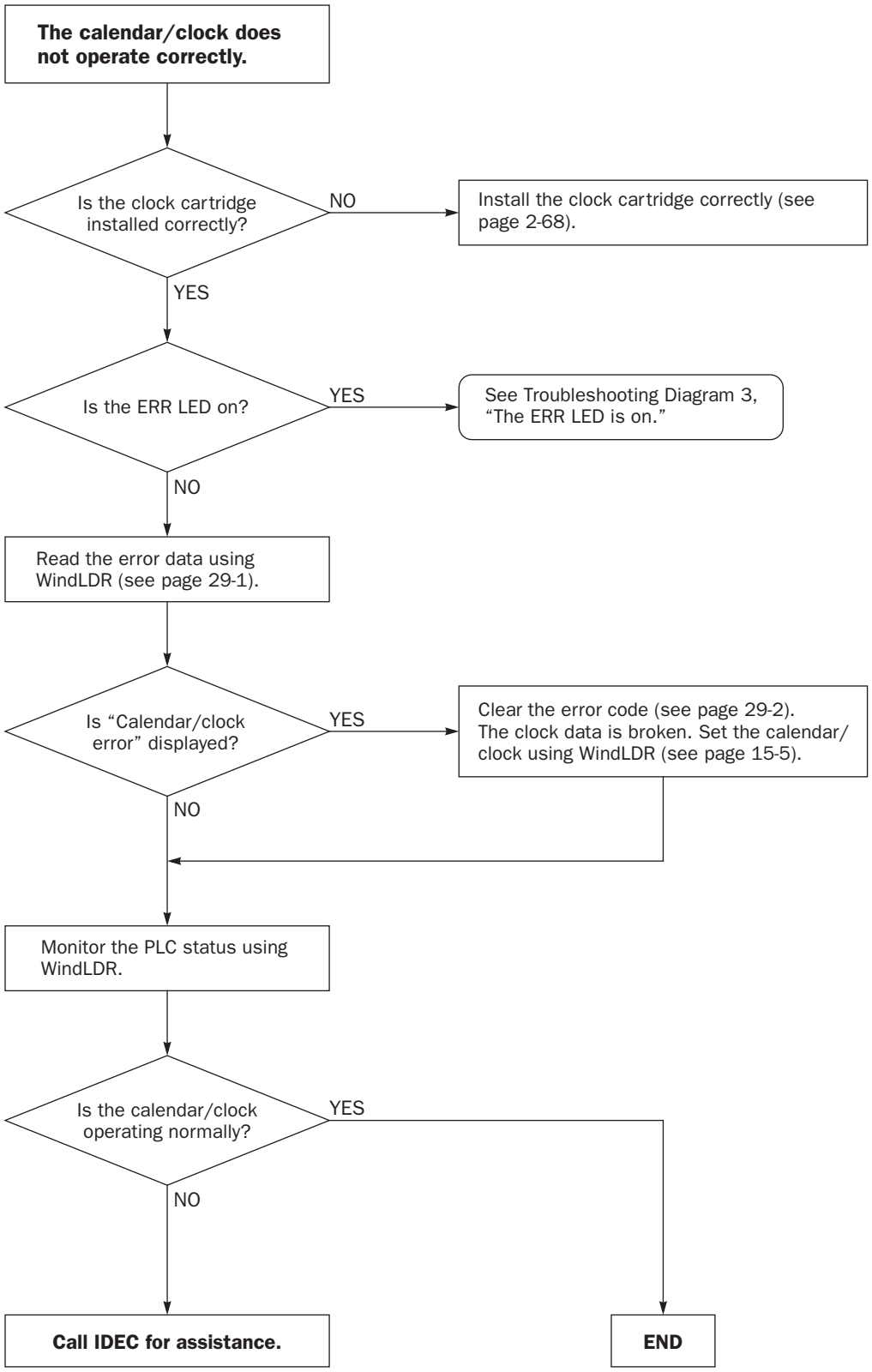
Troubleshooting Diagram 12



Troubleshooting Diagram 13



Troubleshooting Diagram 14

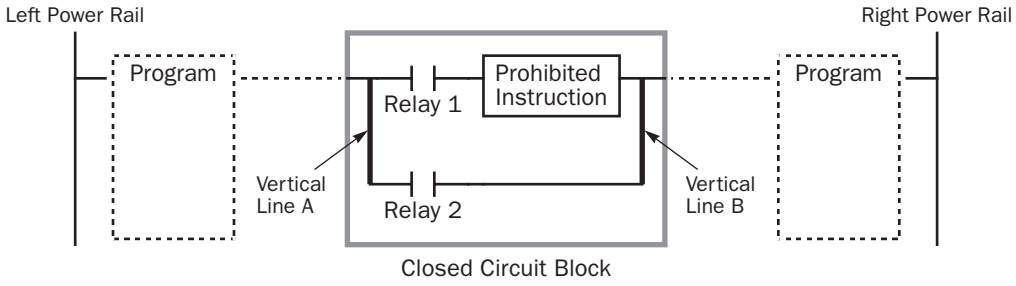


Restriction on Ladder Programming

Caution

- When using WindLDR ver. 4.4 or earlier, the restriction on ladder programming may cause an unexpected operation and possible danger.
- WindLDR ver. 4.5 or later prevents conversion of prohibited ladder program, making sure of safety.

Due to the structure of WindLDR, the following ladder diagram cannot be programmed — a closed circuit block is formed by vertical lines, except for right and left power rails, and the closed circuit block contains one or more prohibited instructions shown in the table below.



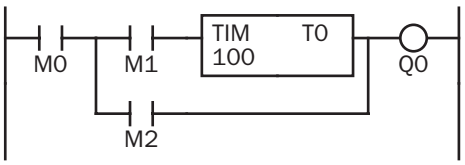
The error detection depends on the version of WindLDR. When using WindLDR ver. 4.4 or earlier, particular care is needed.

Prohibited Instructions		OUT, OUTN, SET, RST, TML, TIM, TMH, TMS, CNT, CDP, CUD, SFR, SFRN, SOTU, SOTD
Error Detection	**WindLDR Ver. 4.4 or earlier**	The ladder program is converted to incorrect mnemonics, without showing an error message. The program can be downloaded to the CPU module, causing an unexpected operation and possible danger.
	WindLDR Ver. 4.5 or later	When converting the ladder program, an error message is shown, such as "TIM follows an invalid operand." Conversion fails to create mnemonics and the program is not downloaded to the CPU module.

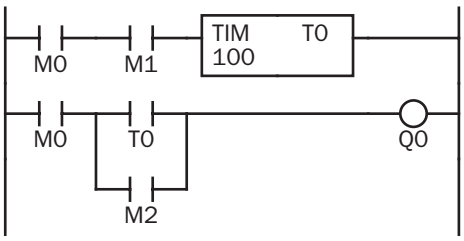
Modifying Prohibited Ladder Programs

Intended operation can be performed by modifying the prohibited ladder program as shown in the examples below:

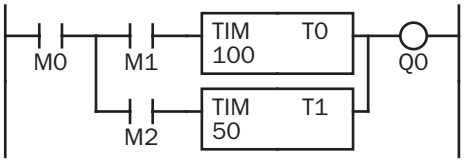
Prohibited Ladder Program 1



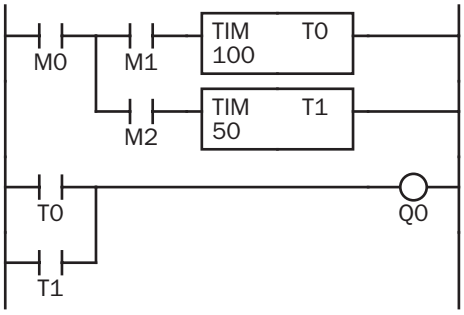
Modified Ladder Program 1



Prohibited Ladder Program 2



Modified Ladder Program 2



APPENDIX

Execution Times for Instructions

Execution times for main instructions of the MicroSmart are listed below:

Instruction	Operand and Condition	Execution Time (µs)	Note
LOD, LODN		1	
OUT, OUTN		3.1	
SET, RST		2.8	
AND, ANDN, OR, ORN		0.7	
AND LOD, OR LOD		1.2	
BPS		0.8	
BRD, BPP		0.5	
TML, TIM, TMH, TMS		24	
CNT		25	
CDP, CUD		27	
CC=, CC≥, DC=, DC≥		12	
SFR, SFRN	N bits	42 + 0.35N	
SOTU, SOTD		17	
JMP, JEND, MCS, MCR		3	
MOV, MOVN	M → M	66	Repeat is not designated on all operands.
	D → D	46	
BMOV	D → D 100 blocks	124	
CMP=, CMP<>, CMP<, CMP>, CMP<=, CMP>=	M ↔ M → M	83	Repeat is not designated on all operands.
	D ↔ D → M	66	
ICMP>=	D ↔ D ↔ D → M	78	
ADD	M + M → D	86	Repeat is not designated on all operands.
	D + D → D	69	
SUB	M – M → D	86	
	D – D → D	69	
MUL	M × M → D	97	
	D × D → D	81	
DIV	M ÷ M → D	111	
	D ÷ D → D	94	
ROOT		428	
ANDW, ORW, XORW	M · M → D	81	Repeat is not designated on all operands.
	D · D → D	63	
BCDLS	7 digits	82	
WSFT	D → D 100 blocks	2442	
HTOB	D → D	97	
BTOH	D → D	84	
HTOA	D → D	129	
ATOH	D → D	133	

Instruction	Operand and Condition	Execution Time (µs)	Note
BTOA	D → D	160	
ATOB	D → D	156	
ENCO	M → D 16 bits	92	
DECO	D → M	51	
BCNT	M → D 16 bits	180	
ALT		26	
LJMP		15	
LCAL		20	
LRET		7	
IOREF	I	52	
	Q	15	
RUNA, STPA	100-byte access	10 ms	

Note: Operands M, D, I, and Q represent internal relay, data register, input, and output, respectively.

Breakdown of END Processing Time

The END processing time depends on the MicroSmart settings and system configuration. The total of execution times for applicable conditions shown below is the actual END processing time.

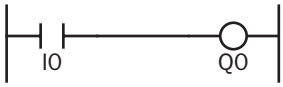
Item	Condition	Execution Time
Housekeeping (built-in I/O service)		640 µs
Expansion I/O service	IN/OUT 8/8 points	260 µs
	IN/OUT 16/16 points	340 µs
	IN/OUT 32/32 points	720 µs
Clock function processing (Note 1)		850 µs
Data link master station processing (Note 2)	When using a data link system	4.2 + 2.4 × transmit/receive words ms (at 19200 bps) See page 25-10.

Note 1: Clock function is processed once every 500 ms.
Note 2: Data link slave stations are processed in interrupt processing asynchronous to the ordinary system processing.

I/O Delay Time

In addition to processing user program instructions and END instruction, the MicroSmart system processing includes interrupt processing of various functions.

The minimum delay from a standard input to a standard output in the program below is 884.1 μs.



Instruction	Data
LOD	I0
OUT	Q0

Maximum execution time	LOD OUT	1.0 µs 3.1 µs
END processing time (without interrupt processing)	Housekeeping	640 µs
Input delay time (DC input without filter setting)	40 µs	
Output delay time (transistor output)	Approx. 200 µs	

The I/O delay time may be increased by such factors as increased END processing time (caused by frequent interrupt processing and larger program size) and input filter setting.

Instruction Steps and Applicability in Interrupt Programs

The steps and bytes of basic and advanced instructions are listed below. Applicability of advanced instructions in interrupt programs are also shown in the rightmost column.

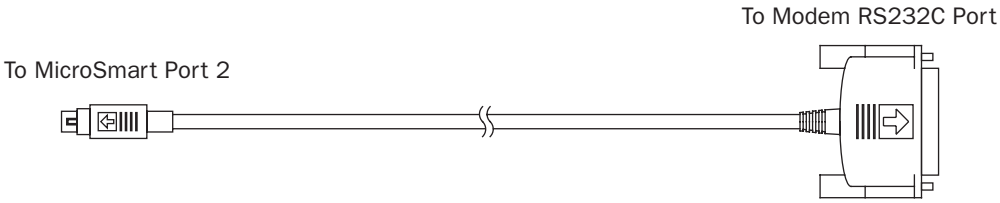
Basic Instruction	Qty of Steps	Qty of Bytes	Advanced Instruction	Qty of Steps	Qty of Bytes	Interrupt
LOD, LODN	1.00	6	NOP	0.33	2	X
OUT, OUTN	1.00	6	MOV, MOVN	2.67	16	X
SET, RST	1.00	6	IMOV, IMOVN	4.00 to 4.67	24 to 28	X
AND, ANDN, OR, ORN	0.67	4	BMOV	3.00	18	X
AND LOD, OR LOD	0.83	5	IBMV, IBMVN	4.00	24	X
BPS	0.83	5	CMP=, CMP<>, CMP<, CMP>, CMP<=, CMP>=	3.33	20	X
BRD	0.50	3				
BPP	0.33	2	ICMP>=	3.67	22	X
TML, TIM, TMH, TMS	0.67	4	ADD, SUB, MUL, DIV	3.33	20	X
CNT, CDP, CUD	0.67	4	ROOT	2.33	14	—
CC=, CC≥	1.17	7	ANDW, ORW, XORW	3.33	20	X
DC=, DC≥	1.33	8	SFTL, SFTR	2.00	12	X
SFR, SFRN	1.00	6	BCDLS	2.33	14	X
SOTU, SOTD	0.83	5	WSFT	3.00	18	X
JMP, JEND, MCS, MCR	0.67	4	ROTL, ROTR	2.00	12	X
END	0.33	2	HTOB, BTOH	2.33	14	X
			HTOA, ATOH, BTOA, ATOB	3.00	18	X
			ENCO, DECO	2.67	16	X
			BCNT	3.00	18	X
			ALT	1.67	10	X
			WKTIM	4.00	24	—
			WKTBL	2.17 to 14.83	13 to 89	—
			DISP	2.67	16	—
			DGRD	3.33	20	—
			TXD1, TXD2, RXD1, RXD2	3.50 to 136.50	21 to 819	—
			LABEL	1.33	8	X
			LJMP, LCAL	1.67	10	X
			LRET	1.00	6	X
			IOREF	2.67	16	X
			DI, EI	1.33	8	—
			XYFS	4.00 to 20.67	24 to 124	—
			CVXTY, CVYTX	3.00	18	—
			PULS1, PULS2	2.00	12	—
			PWM1, PWM2	4.00	24	—
			RAMP	2.33	14	—
			ZRN1, ZRN2	3.00	18	—
			PID	4.33	26	—
			DTML, DTIM, DTMH, DTMS	3.67	22	—
			TTIM	1.67	10	—
			RUNA, STPA	3.33	20	—

Cables

Communication cables and their connector pinouts are described in this section.

Modem Cable 1C (FC2A-KM1C)

Cable Length: 3m (9.84 feet)



Mini DIN Connector Pinouts

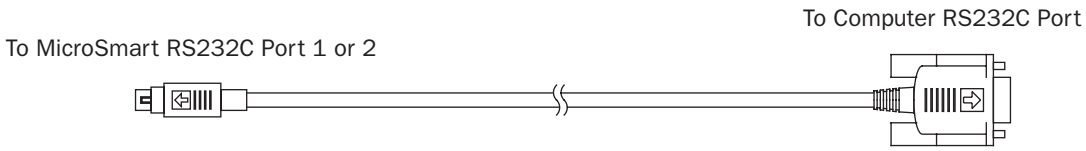
Description		Pin
Shield		Cover
RTS	Request to Send	1
DTR	Data Terminal Ready	2
TXD	Transmit Data	3
RXD	Receive Data	4
DSR	Data Set Ready	5
SG	Signal Ground	6
SG	Signal Ground	7
NC	No Connection	8

D-sub 25-pin Male Connector Pinouts

Pin	Description	
1	FG	Frame Ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	RTS	Request to Send
5	NC	No Connection
6	NC	No Connection
7	SG	Signal Ground
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready

Computer Link Cable 4C (FC2A-KC4C)

Cable Length: 3m (9.84 feet)



Mini DIN Connector Pinouts

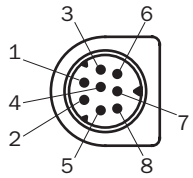
Description		Pin
Shield		Cover
TXD	Transmit Data	3
RXD	Receive Data	4
RTS	Request to Send	1
NC	No Connection	8
DSR	Data Set Ready	5
DTR	Data Terminal Ready	2
SG	Signal Ground	7
SG	Signal Ground	6

D-sub 9-pin Female Connector Pinouts

Pin	Description	
Cover	FG	Frame Ground
3	TXD	Transmit Data
2	RXD	Receive Data
6	DSR	Data Set Ready
8	CTS	Clear to Send
1	DCD	Data Carrier Detect
4	DTR	Data Terminal Ready
5	SG	Signal Ground
7	RTS	Request to Send
9	RI	Ring Indicator

User Communication Cable 1C (FC2A-KP1C)

Cable Length: 2.4m (7.87 feet)



To MicroSmart RS232C Port 1 or 2

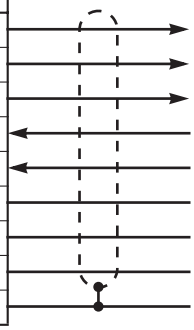
To RS232C Port

Attach a proper connector to the open end referring to the cable connector pinouts shown below.

Mini DIN Connector Pinouts

Pin	Port 1		Port 2		AWG#		Color
1	NC	No Connection	RTS	Request to Send	28	Twisted	Black
2	NC	No Connection	DTR	Data Terminal Ready	28		Yellow
3	TXD	Transmit Data	TXD	Transmit Data	28		Blue
4	RXD	Receive Data	RXD	Receive Data	28		Green
5	NC	No Connection	DSR	Data Set Ready	28		Brown
6	CMSW	Communication Switch	SG	Signal Ground	28		Gray
7	SG	Signal Ground	SG	Signal Ground	26	Twisted	Red
8	NC	No Connection	NC	No Connection	26		White
Cover	—		—		—		Shield

Signal Direction



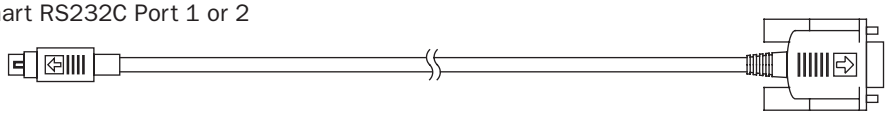
Note: When preparing a cable for port 1, keep pins 6 and 7 open. If pins 6 and 7 are connected together, user communication cannot be used.

O/I Communication Cable 1C (FC4A-KC1C)

Cable Length: 5m (16.4 feet)

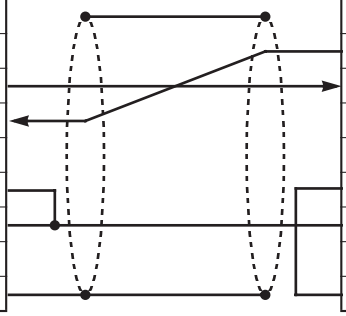
To MicroSmart RS232C Port 1 or 2

To HG1B, HG2A, or HG2C



Mini DIN Connector Pinouts

Description		Pin
NC	No Connection	**1**
NC	No Connection	**2**
TXD	Transmit Data	**3**
RXD	Receive Data	**4**
NC	No Connection	**5**
CMSW	Communication Switch	**6**
SG	Signal Ground	**7**
NC	No Connection	**8**
Shield		**Cover**

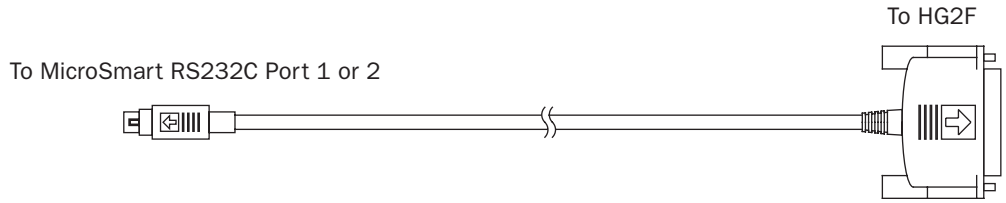


D-sub 9-pin Male Connector Pinouts

Pin	Description	
1	FG	Frame Ground
2	TXD1	Transmit Data 1
3	RXD1	Receive Data 1
4	TXD2	Transmit Data 2
5	RXD2	Receive Data 2
6	DSR	Data Set Ready
7	SG	Signal Ground
8	NC	No Connection
9	DTR	Data Terminal Ready

O/I Communication Cable 2C (FC4A-KC2C)

Cable Length: 5m (16.4 feet)



Mini DIN Connector Pinouts

Description		Pin
NC	No Connection	**1**
NC	No Connection	**2**
TXD	Transmit Data	**3**
RXD	Receive Data	**4**
NC	No Connection	**5**
CMSW	Communication Switch	**6**
SG	Signal Ground	**7**
NC	No Connection	**8**
Shield		**Cover**

D-sub 25-pin Male Connector Pinouts

Pin	Description	
1	FG	Frame Ground
2	TXD	Transmit Data
3	RXD	Receive Data
4	RTS	Request to Send
5	CTS	Clear to Send
6	DSR	Data Set Ready
7	SG	Signal Ground
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready

Type List

CPU Modules (All-in-One Type)

Power Voltage	Input Type	Output Type	I/O Points	Type No.
100-240V AC 50/60 Hz	24V DC Sink/Source	Relay Output 240V AC/30V DC, 2A	10-I/O Type (6 in / 4 out)	FC4A-C10R2
			16-I/O Type (9 in / 7 out)	FC4A-C16R2
			24-I/O Type (14 in / 10 out)	FC4A-C24R2
24V DC			10-I/O Type (6 in / 4 out)	FC4A-C10R2C
			16-I/O Type (9 in / 7 out)	FC4A-C16R2C
			24-I/O Type (14 in / 10 out)	FC4A-C24R2C

CPU Modules (Slim Type)

Power Voltage	Input Type	Output Type	High-speed Transistor Output	I/O Points	Type No.
24V DC	24V DC Sink/Source	Transistor Sink Output 0.3A		20 (12 in / 8 out)	FC4A-D20K3
		Transistor Source Output 0.3A			FC4A-D20S3
		Relay Output 240V AC/30V DC, 2A	Sink Output 0.3A	20 (12 in / 8 out) *	FC4A-D20RK1
			Source Output 0.3A		FC4A-D20RS1
		Transistor Sink Output 0.3A		40 (24 in / 16 out)	FC4A-D40K3
		Transistor Source Output 0.3A			FC4A-D40S3

Note *: Two points are transistor outputs, and six points are relay outputs.

Input Modules

Input Type	Input Points	Terminal	Type No.
24V DC Sink/Source	8 points	Removable Terminal Block	FC4A-N08B1
	16 points		FC4A-N16B1
	16 points	MIL Connector	FC4A-N16B3
	32 points		FC4A-N32B3
120V AC	8 points	Removable Terminal Block	FC4A-N08A11

Output Modules

Output Type	Output Points	Terminal	Type No.
Relay Output 240V AC/30V DC, 2A	8 points	Removable Terminal Block	FC4A-R081
	16 points		FC4A-R161
Transistor Sink Output 0.3A	8 points		FC4A-T08K1
Transistor Source Output 0.3A			FC4A-T08S1
Transistor Sink Output 0.1A	16 points	MIL Connector	FC4A-T16K3
Transistor Source Output 0.1A			FC4A-T16S3
Transistor Sink Output 0.1A	32 points		FC4A-T32K3
Transistor Source Output 0.1A			FC4A-T32S3

Mixed I/O Modules

Input Type	Output Type	I/O Points	Terminal	Type No.
24V DC Sink/Source	Relay Output 240V AC/30V DC, 2A	8 (4 in / 4 out)	Removable Terminal Block	FC4A-M08BR1
		24 (16 in / 8 out)	Non-removable Terminal Block	FC4A-M24BR2

Analog I/O Modules

Name	I/O Signal	I/O Points	Category	Terminal	Type No.
Analog I/O Module	Voltage (0 to 10V DC) Current (4 to 20mA)	2 inputs	END Refresh Type	Removable Terminal Block	FC4A-L03A1
	Voltage (0 to 10V DC) Current (4 to 20mA)	1 output			
	Thermocouple (K, J, T) Resistance thermometer (Pt100)	2 inputs			FC4A-L03AP1
	Voltage (0 to 10V DC) Current (4 to 20mA)	1 output			
Analog Input Module	Voltage (0 to 10V DC) Current (4 to 20mA)	2 inputs			FC4A-J2A1
	Voltage (0 to 10V DC) Current (4 to 20mA) Thermocouple (K, J, T) Resistance thermometer (Pt100, Pt1000, Ni100, Ni1000)	4 inputs	Ladder Refresh Type		FC4A-J4CN1
	Voltage (0 to 10V DC) Current (4 to 20mA)	8 inputs			FC4A-J8C1
	Thermistor (PTC, NTC)	8 inputs			FC4A-J8AT1
Analog Output Module	Voltage (0 to 10V DC) Current (4 to 20mA)	1 output	END Refresh		FC4A-K1A1
	Voltage (–10 to +10V DC) Current (4 to 20mA)	2 outputs	Ladder Refresh		FC4A-K2C1

AS-Interface Master Module

Name	Terminal	Type No.
AS-Interface Master Module	Removable Terminal Block	FC4A-AS62M

Optional Modules, Adapters, and Cartridges

Name	Description	Type No.
HMI Module	For displaying and changing required operands	FC4A-PH1
HMI Base Module	For mounting HMI module with slim type CPU module	FC4A-HPH1
RS232C Communication Adapter	Mini DIN connector type for all-in-one 16- and 24-I/O CPU modules *	FC4A-PC1
RS485 Communication Adapter	Mini DIN connector type for all-in-one 16- and 24-I/O CPU modules *	FC4A-PC2
	Terminal block type for all-in-one 16- and 24-I/O CPU modules *	FC4A-PC3
RS232C Communication Module	Mini DIN connector type for slim type CPU module	FC4A-HPC1
RS485 Communication Module	Mini DIN connector type for slim type CPU module	FC4A-HPC2
	Terminal block type for slim type CPU module	FC4A-HPC3
Memory Cartridge	32KB EEPROM for storing a user program	FC4A-PM32
	64KB EEPROM for storing a user program	FC4A-PM64
Clock Cartridge	Real time calendar/clock function	FC4A-PT1

Note *: RS232C or RS485 communication adapters can also be installed on the HMI base module mounted next to the slim type CPU module.

Accessories

Name	Function	Type No.
RS232C/RS485 Converter	Used for interface between a computer and the MicroSmart CPU modules in the computer link 1:N communication system or through modems	FC2A-MD1
RS232C Cable (4-wire) (1.5m/4.92 ft. long)	Used to connect the RS232C/RS485 converter to a computer, with D-sub 9-pin female connector to connect to computer	HD9Z-C52
DIN Rails (1m/3.28 ft. long)	35-mm-wide aluminum DIN rail to mount MicroSmart modules (package quantity 10)	BAA1000PN10
DIN Rails (1m/3.28 ft. long)	35-mm-wide steel DIN rail to mount MicroSmart modules (package quantity 10)	BAP1000PN10
End Clips	Used on DIN rail to fasten MicroSmart modules (package quantity 10)	BNL6PN10
Direct Mounting Strips	Used for direct mounting of slim type CPU or I/O modules on a panel (package quantity 5)	FC4A-PSP1PN05
10-position Terminal Blocks	For I/O modules (package quantity 2)	FC4A-PMT10PN02
11-position Terminal Blocks	For I/O modules (package quantity 2)	FC4A-PMT11PN02
13-position Terminal Blocks	For slim type CPU modules FC4A-D20RK1 and FC4A-D20RS1 (package quantity 2)	FC4A-PMT13PN02
16-position Terminal Blocks	For slim type CPU module FC4A-D20RK1 (package quantity 2)	FC4A-PMTK16PN02
16-position Terminal Blocks	For slim type CPU module FC4A-D20RS1 (package quantity 2)	FC4A-PMTS16PN02
20-position Connector Socket	MIL connector for I/O modules (package quantity 2)	FC4A-PMC20PN02
26-position Connector Socket	MIL connector for slim type CPU modules (package quantity 2)	FC4A-PMC26PN02
Phoenix Ferrule	Ferrule for connecting 1 or 2 wires to screw terminal	See page 3-18
Phoenix Crimping Tool	Used for crimping ferrules	See page 3-18
Phoenix Screwdriver	Used for tightening screw terminals	See page 3-18
WindLDR	Programming and monitoring software for Windows PC (CD)	FC9Y-LP2CDW
MicroSmart User's Manual	This printed manual	FC9Y-B812

BX Series I/O Terminals and Applicable Cables

MicroSmart		Cable Type No.	I/O Terminal Type No.	Connector
Module	Type No.			
CPU Module	FC4A-D20K3 FC4A-D20S3 FC4A-D40K3 FC4A-D40S3	FC9Z-H①②26	BX1D-③26A BX1F-③26A	26-pole MIL connector
Input Module	FC4A-N16B3 FC4A-N32B3	FC9Z-H①②20	BX1D-③20A BX1F-③20A BX7D-BT16A1T (16-pt relay output)	20-pole MIL connector
Output Module	FC4A-T16K3 FC4A-T16S3 FC4A-T32K3 FC4A-T32S3			

Specify required designation codes in place of ①, ②, and ③.

① Cable Length Code	② Cable Shield Code	③ Terminal Screw Style Code
050: 0.5m 100: 1m 200: 2m 300: 3m	A: Shielded cable B: Non-shielded cable	T: Touch-down terminal S: Screw terminal

Cables

Name	Function	Type No.
Modem Cable 1C (3m/9.84 ft. long)	Used to connect a modem to the MicroSmart RS232C port, with D-sub 25-pin male connector to connect to modem	FC2A-KM1C
Computer Link Cable 4C (3m/9.84 ft. long)	Used to connect a computer to the MicroSmart RS232C port (1:1 computer link), with D-sub 9-pin female connector to connect to computer	FC2A-KC4C
User Communication Cable 1C (2.4m/7.87 ft. long)	Used to connect RS232C equipment to the MicroSmart RS232C port, without a connector to connect to RS232C equipment	FC2A-KP1C
O/I Communication Cable 1C (5m/16.4 ft. long)	RS232C cable used to connect IDEC HG1B/2A/2C operator interface to MicroSmart RS232C port 1 or 2	FC4A-KC1C
O/I Communication Cable 2C (5m/16.4 ft. long)	RS232C cable used to connect IDEC HG2F operator interface to MicroSmart RS232C port 1 or 2	FC4A-KC2C
Analog Voltage Input Cable (1m/3.28 ft. long)	Used to connect an analog voltage source to the analog voltage input connector on the slim type CPU module (package quantity 2)	FC4A-PMAC2PN02
Shielded CPU Flat Cable (0.5m/1.64 ft. long)	26-wire shielded straight cable for connecting the MicroSmart slim type CPU module to an I/O terminal	FC9Z-H050A26
Shielded CPU Flat Cable (1m/3.28 ft. long)		FC9Z-H100A26
Shielded CPU Flat Cable (2m/6.56 ft. long)		FC9Z-H200A26
Shielded CPU Flat Cable (3m/9.84 ft. long)		FC9Z-H300A26
Non-shielded CPU Flat Cable (0.5m/1.64 ft. long)	26-wire non-shielded straight cable for connecting the MicroSmart slim type CPU module to an I/O terminal	FC9Z-H050B26
Non-shielded CPU Flat Cable (1m/3.28 ft. long)		FC9Z-H100B26
Non-shielded CPU Flat Cable (2m/6.56 ft. long)		FC9Z-H200B26
Non-shielded CPU Flat Cable (3m/9.84 ft. long)		FC9Z-H300B26
Shielded I/O Flat Cable (0.5m/1.64 ft. long)	20-wire shielded straight cable for connecting the MicroSmart I/O module to an I/O terminal	FC9Z-H050A20
Shielded I/O Flat Cable (1m/3.28 ft. long)		FC9Z-H100A20
Shielded I/O Flat Cable (2m/6.56 ft. long)		FC9Z-H200A20
Shielded I/O Flat Cable (3m/9.84 ft. long)		FC9Z-H300A20
Non-shielded I/O Flat Cable (0.5m/1.64 ft. long)	20-wire non-shielded straight cable for connecting the MicroSmart I/O module to an I/O terminal	FC9Z-H050B20
Non-shielded I/O Flat Cable (1m/3.28 ft. long)		FC9Z-H100B20
Non-shielded I/O Flat Cable (2m/6.56 ft. long)		FC9Z-H200B20
Non-shielded I/O Flat Cable (3m/9.84 ft. long)		FC9Z-H300B20

INDEX

UNITED STATES
IDEC CORPORATION
1175 Elko Drive, Sunnyvale, CA 94089-2209, USA
Tel: +1-408-747-0550
Toll Free: (800) 262-IDEC
Fax: +1-408-744-9055
Toll Free Fax: (800) 635-6246
E-mail: opencontact@idec.com

CANADA
IDEC CANADA LIMITED
3155 Pepper Mill Court, Unit 4,
Mississauga, Ontario, L5L 4X7, Canada
Tel: +1-905-890-8561
Toll Free: (888) 317-4332
Fax: +1-905-890-8562
E-mail: sales@ca.idec.com

AUSTRALIA
IDEC AUSTRALIA PTY. LTD.
2/3 Macro Court, Rowville, Victoria 3178, Australia
Tel: +61-3-9763-3244
Toll Free: 1800-68-4332
Fax: +61-3-9763-3255
E-mail: sales@au.idec.com

UNITED KINGDOM
IDEC ELECTRONICS LIMITED
Unit 2, Beechwood, Chineham Business Park,
Basingstoke, Hampshire RG24 8WA, UK
Tel: +44-1256-321000
Fax: +44-1256-327755
E-mail: sales@uk.idec.com

GERMANY
IDEC ELEKTROTECHNIK GmbH
Wendenstrasse 331, 20537 Hamburg, Germany
Tel: +49-40-25 30 54 - 0
Fax: +49-40-25 30 54 24
E-mail: service@idec.de

JAPAN
IDEC CORPORATION
7-31, Nishi-Miyahara 1-Chome,
Yodogawa-ku, Osaka 532-8550, Japan
Tel: +81-6-6398-2571
Fax: +81-6-6392-9731
E-mail: marketing@idec.co.jp

CHINA
IDEC (SHANGHAI) CORPORATION
Room 608-609, 6F, Gangtai Plaza, No. 700,
Yan'an East Road, Shanghai 200001, PRC
Tel: +86-21-5353-1000
Fax: +86-21-5353-1263
E-mail: idec@cn.idec.com

IDEC (BEIJING) CORPORATION
Room 211B, Tower B, The Grand Pacific Building, 8A
Guanghua Road, Chaoyang District, Beijing 100026, PRC
Tel: +86-10-6581-6131
Fax: +86-10-6581-5119

IDEC (SHENZHEN) CORPORATION
Unit AB-3B2, Tian Xiang Building, Tian'an Cyber Park,
Fu Tian District, Shenzhen, Guang Dong 518040, PRC
Tel: +86-755-8356-2977
Fax: +86-755-8356-2944

HONG KONG
IDEC IZUMI (H.K.) CO., LTD.
Units 11-15, Level 27, Tower 1, Millennium City 1,
388 Kwun Tong Road, Kwun Tong, Kowloon, Hong Kong
Tel: +852-2803-8989
Fax: +852-2565-0171
E-mail: info@hk.idec.com

TAIWAN
IDEC TAIWAN CORPORATION
8F-1, No. 79, Hsin Tai Wu Road, Sec. 1, Hsi-Chih,
Taipei County, Taiwan
Tel: +886-2-2698-3929
Fax: +886-2-2698-3931
E-mail: service@tw.idec.com

SINGAPORE
IDEC IZUMI ASIA PTE. LTD.
No. 31, Tannery Lane #05-01
HB Centre 2, Singapore 347788
Tel: +65-6746-1155
Fax: +65-6844-5995
E-mail: info@sg.idec.com

www.idec.com

Manual No. FC9Y-B812-03 06/2008

B-812(3)