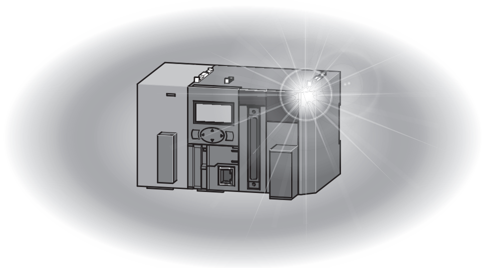
MITSUBISHI ELECTRIC

Programmable Controller

MELSEC L series

MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)

-L02SCPU
-L02SCPU-P
-L02CPU
-L02CPU-P
-L06CPU
-L06CPU-P
-L26CPU
-L26CPU-P
-L26CPU-BT
-L26CPU-PBT



●SAFETY PRECAUTIONS●

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.

In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

⚠WARNING	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠CAUTION" may lead to serious consequences.
Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠WARNING

- Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the programmable controller.
  (2) Machine OPR (Original Point Return) of the positioning function is controlled by two kinds of data: an OPR direction and an OPR speed. Deceleration starts when the near-point dog signal turns on. If an incorrect OPR direction is set, motion control may continue without deceleration. To prevent machine damage caused by this, configure an interlock circuit external to the programmable controller.
  (3) When the CPU module detects an error during control by the positioning function, the motion slows down and stops.
  (4) When the programmable controller detects an abnormal condition, it stops the operation and all outputs are:
    - Turned off if the overcurrent or overvoltage protection of the power supply module is activated.
    - Held or turned off according to the parameter setting if the self-diagnostic function of the CPU module detects an error such as a watchdog timer error.
  (5) All outputs may be turned on if an error occurs in a part, such as an I/O control part, where the CPU module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the programmable controller. For a fail-safe circuit example, refer to "General Safety Requirements" in the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).
  (6) Outputs may remain on or off due to a failure of a component such as a transistor in an output circuit. Configure an external circuit for monitoring output signals that could cause a serious accident.
- In an output circuit, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- Configure a circuit so that the programmable controller is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
- Configure a circuit so that the external power supply is turned off first and then the programmable controller. If the programmable controller is turned off first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to relevant manuals for each network. Incorrect output or malfunction due to a communication failure may result in an accident.

⚠WARNING

- When changing data from a peripheral connected to the CPU module to the running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- An absolute position restoration by the positioning function may turn off the servo-on signal (servo off) for approximately 20ms, and the motor may run unexpectedly. If this causes a problem, provide an electromagnetic brake to lock the motor during absolute position restoration.

[Design Precautions]

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100mm or more between them. Failure to do so may result in malfunction due to noise.
- During control of an inductive load such as a lamp, heater, or solenoid valve, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Therefore, use a module that has a sufficient current rating.
- After the CPU module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter settings, and/or program size. Design circuits so that the entire system will always operate safely, regardless of the time.

[Installation Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before mounting or removing a module. Failure to do so may result in electric shock or cause the module to fail or malfunction.

[Installation Precautions]

⚠CAUTION

- Use the programmable controller in an environment that meets the general specifications in the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection). Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- To interconnect modules, engage the respective connectors and securely lock the module joint levers until they click. Incorrect interconnection may cause malfunction, failure, or drop of the module.
- Do not directly touch any conductive parts and electronic components of the module. Doing so can cause malfunction or failure of the module.
- Securely connect an extension cable to the connectors of a branch module and an extension module. After connections, check that the cable is inserted completely. Poor contact may cause malfunction.
- When using an SD memory card, fully insert it into the SD memory card slot. Check that it is inserted completely. Poor contact may cause malfunction.
- Do not directly touch any conductive parts and electronic components of the module or SD memory card. Doing so can cause malfunction or failure of the module.

[Wiring Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and wiring, attach the included terminal cover to the module before turning it on for operation. Failure to do so may result in electric shock.

[Wiring Precautions]

⚠CAUTION

- Individually ground the FG terminal of the programmable controller with a ground resistance of 100Ω or less. Failure to do so may result in electric shock or malfunction.
- Use applicable solderless terminals and tighten them within the specified torque range.
  If any spade solderless terminal is used, it may be disconnected when a terminal block screw comes loose, resulting in failure.
- Check the rated voltage and terminal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause a fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered. Incomplete connections may cause short circuit, fire, or malfunction.
- Securely connect the connector to the module.
- Do not install the control lines or communication cables together with the main circuit lines or power cables. Failure to do so may result in malfunction due to noise.
- Place the cables in a duct or clamp them. If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
- Tighten the terminal block screws within the specified torque range. Undertightening can cause short circuit, fire, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, fire, or malfunction.
- When disconnecting the cable from the module, do not pull the cable by the cable part. For the cable with connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
- Prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction.
- A protective film is attached to the top of the module to prevent foreign matter, such as wire chips, from entering the module during wiring. Do not remove the film during wiring. Remove it for heat dissipation before system operation.
- To use the high-speed counter function, ground the shield cable on the encoder side (relay box). Always ground the FG and LG terminals to the protective ground conductor. Failure to do so may cause malfunction.
- Mitsubishi programmable controllers must be installed in control panels. Connect the main power supply to the power supply module in the control panel through a relay terminal block.
  Wiring and replacement of a power supply module must be performed by qualified maintenance personnel with knowledge of protection against electric shock.
  For wiring methods, refer to the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).

[Startup and Maintenance Precautions]

⚠WARNING

- Do not touch any terminal while power is on. Doing so will cause electric shock or malfunction.
- Correctly connect the battery connector. Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire. Also, do not expose it to liquid or strong shock.
  Doing so will cause the battery to produce heat, explode, ignite, or leak, resulting in injury and fire.
- Shut off the external power supply (all phases) used in the system before cleaning the module or retightening the terminal block screws or connector screws. Failure to do so may result in electric shock.

[Startup and Maintenance Precautions]

⚠CAUTION

- Before performing online operations (especially, program modification, forced output, and operating status change) for the running CPU module from the peripheral connected, read relevant manuals carefully and ensure the safety. Improper operation may damage machines or cause accidents.
- Do not disassemble or modify the module. Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) more than 25cm away in all directions from the programmable controller. Failure to do so may cause malfunction.
- Shut off the external power supply (all phases) used in the system before mounting or removing a module. Failure to do so may cause the module to fail or malfunction.
- Tighten the terminal block screws or connector screws within the specified torque range.
  Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- After the first use of the product (module, display unit, and terminal block), do not connect/disconnect the product more than 50 times (IEC 61131-2/JIS B 3502 compliant). Exceeding the limit may cause malfunction.
- After the first use of the SD memory card, do not insert/remove the memory card more than 500 times. Exceeding the limit may cause malfunction.
- Do not drop or apply shock to the battery to be installed in the module. Doing so may damage the battery, causing the battery fluid to leak inside the battery. If the battery is dropped or any shock is applied to it, dispose of it without using.
- Before handling the module, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Failure to do so may cause the module to fail or malfunction.
- Before testing the operation by the positioning function, set a low speed value for the speed limit parameter so that the operation can be stopped immediately upon occurrence of a hazardous condition.

[Disposal Precautions]

⚠CAUTION
● When disposing of this product, treat it as industrial waste. When disposing of batteries, separate them from other wastes according to the local regulations. (For details on battery regulations in EU member states, refer to the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).)

[Transportation Precautions]

⚠CAUTION
● When transporting lithium batteries, follow the transportation regulations. (For details on the regulated models, refer to the MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection).)

●CONDITIONS OF USE FOR THE PRODUCT●

(1) Mitsubishi programmable controller ("the PRODUCT") shall be used in conditions;

i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and

ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.

MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.

("Prohibited Application")

Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above restrictions, Mitsubishi may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTs are required. For details, please contact the Mitsubishi representative in your region.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC-L series programmable controllers.
This manual describes the memory maps, functions, and devices of the CPU module, and programming.

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC-L series programmable controller to handle the product correctly.
When applying the program examples introduced in this manual to an actual system, ensure the applicability and confirm that it will not cause system control problems.

■ Relevant CPU modules

CPU module	Model
LCPU	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT

Remark

This manual does not describe the details of the instructions, error codes, special relay (SM), and special register (SD).

- For the instructions, refer to the following.
  📖MELSEC-Q/L Programming Manual (Common Instruction)
- For the error codes, special relay(SM), and special register (SD), refer to the following.
  📖MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

RELEVANT MANUALS

(1) CPU module user's manual

Manual name <manual number (model code)>	Description
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection) <SH-080890ENG, 13JZ36>	Specifications of the CPU modules, power supply modules, display unit, branch module, extension module, SD memory cards, and batteries, information on how to establish a system, maintenance and inspection, and troubleshooting
MELSEC-L CPU Module User's Manual (Built-In Ethernet Function) <SH-080891ENG, 13JZ37>	The built-in Ethernet function of the CPU module
MELSEC-L CPU Module User's Manual (Built-In I/O Function) <SH-080892ENG, 13JZ38>	The general-purpose I/O function, interrupt input function, pulse catch function, positioning function, and high-speed counter function of the CPU module
QnUDVCPU/LCPU User's Manual (Data Logging Function) <SH-080893ENG, 13JZ39>	The data logging function of the CPU module

(2) Programming manual

Manual name <manual number (model code)>	Description
MELSEC-Q/L Programming Manual (Common Instruction) <SH-080809ENG, 13JW10>	Detailed description and usage of instructions used in programs
MELSEC-Q/L/QnA Programming Manual (SFC) <SH-080041, 13JF60>	System configuration, specifications, functions, programming, and error codes for SFC (MELSAP3) programs
MELSEC-Q/L Programming Manual (MELSAP-L) <SH-080076, 13JF61>	System configuration, specifications, functions, programming, and error codes for SFC (MELSAP-L) programs
MELSEC-Q/L Programming Manual (Structured Text) <SH-080366E, 13JF68>	System configuration and programming using structured text language
MELSEC-Q/L/QnA Programming Manual (PID Control Instructions) <SH-080040, 13JF59>	Dedicated instructions for PID control

(3) Operating manual

Manual name <manual number (model code)>	Description
GX Works2 Version 1 Operating Manual (Common) <SH-080779ENG, 13JU63>	System configuration, parameter settings, and online operations of GX Works2, which are common to Simple projects and Structured projects
GX Developer Version 8 Operating Manual <SH-080373E, 13JU41>	Operating methods of GX Developer, such as programming, printing, monitoring, and debugging

(4) I/O module and intelligent function module manual

Manual name <manual number (model code)>	Description
MELSEC-L I/O Module User's Manual <SH-080888ENG, 13JZ34>	Specifications and troubleshooting of the I/O module
MELSEC-L Ethernet Interface Module User's Manual (Basic) <SH-081105ENG, 13JZ73>	Specifications, procedures for data communication with external devices, line connection (open/close), fixed buffer communication, random access buffer communication, and troubleshooting of the Ethernet module
MELSEC-L Serial Communication Module User's Manual (Basic) <SH-080894ENG, 13JZ40>	System configuration, specifications, procedures before operation, data communication methods (basic), and troubleshooting of the serial communication module
MELSEC Communication Protocol Reference Manual <SH-080008, 13JF89>	Details of MELSEC communication protocol (MC protocol) that is used for data communication between a target device and a CPU module
MELSEC-L CC-Link System Master/Local Module User's Manual <SH-080895ENG, 13JZ41>	Settings, specifications, handling, data communication methods, and troubleshooting of the built-in CC-Link function of the CPU module or the CC-Link system master/local module
MELSEC-L CC-Link/LT Master Module User's Manual <SH-081012ENG, 13JZ65>	Settings, specifications, handling, data communication methods, and troubleshooting of the CC-Link/LT master module
MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual <SH-080972ENG, 13JZ54>	Overview of CC-Link IE Field Network, and specifications, procedures before operation, system configuration, installation, wiring, settings, functions, programming, and troubleshooting of the MELSEC-L series CC-Link IE Field Network master/local module
MELSEC-L Analog-Digital Converter Module User's Manual <SH-080899ENG, 13JZ42>	System configuration, specifications, settings, and troubleshooting of the analog-digital converter module
MELSEC-L Dual Channel Isolated High Resolution Analog-Digital Converter Module User's Manual <SH-081103ENG, 13JZ72>	System configuration, specifications, settings, and troubleshooting of the dual channel isolated high resolution analog-digital converter module
MELSEC-L Digital-Analog Converter Module User's Manual <SH-080900ENG, 13JZ43>	System configuration, specifications, settings, and troubleshooting of the digital-analog converter module
MELSEC-L Analog Input/Output Module User's Manual <SH-081167ENG, 13JZ87>	System configuration, specifications, settings, and troubleshooting of the analog input/output module
MELSEC-L LD75P/LD75D Positioning Module User's Manual <SH-080911ENG, 13JZ46>	System configuration specifications settings and troubleshooting of the positioning module
MELSEC-L High-Speed Counter Module User's Manual <SH-080920ENG, 13JZ49>	System configuration specifications settings and troubleshooting of the high-speed counter module
MELSEC-L Temperature Control Module User's Manual <SH-081000ENG, 13JZ64>	System configuration specifications settings and troubleshooting of the temperature control module
MELSEC-L Multiple Input (Voltage/Current/Temperature) Module User's Manual <SH-081365ENG, 13JZ76>	System configuration, specifications, settings, and troubleshooting of the multiple input module

(5) Others

Manual name <manual number (model code)>	Description
iQ Sensor Solution Reference Manual <SH-081133ENG, 13JV28>	Operating methods of iQ Sensor Solution, such as programming and monitoring
CC-Link IE Field Network Basic Reference Manual <SH-081684ENG, 13JX62>	Specifications, procedures before operation, system configuration, programming, functions, parameter settings, and troubleshooting of CC-Link IE Field Network Basic

CONTENTS

SAFETY PRECAUTIONS 1
CONDITIONS OF USE FOR THE PRODUCT 8
INTRODUCTION 9
RELEVANT MANUALS 10
MANUAL PAGE ORGANIZATION 17
TERMS 18
GENERIC TERMS AND ABBREVIATIONS 18

PART 1 PROGRAMMING

CHAPTER 1 BASIC PROCEDURE FOR PROGRAMMING 20

1.1 System Configuration Example 21
1.2 Creating a Project 22
1.3 Creating a Program 23
1.3.1 Prior knowledge for creating a program 23
1.3.2 How to create a program 24
1.4 Converting a Program 25
1.5 Writing a Project to the CPU Module 26
1.5.1 Formatting a memory 26
1.5.2 Writing to the CPU module 27
1.6 Checking an Operation of the CPU Module 28
1.7 Saving a Project 30

CHAPTER 2 APPLICATION OF PROGRAMMING 31

2.1 Memory and Files 31
2.1.1 Memory configuration 31
2.1.2 Parameter-valid drive 34
2.1.3 Files 35
2.2 I/O Number 41
2.2.1 Concept of I/O number assignment 41
2.2.2 I/O number assignment 43
2.3 Scan Time Structure 45
2.3.1 Initial processing 45
2.3.2 Refresh processing of input/output modules or intelligent function modules 46
2.3.3 Program operation 46
2.3.4 END processing 47
2.4 Operation Processing of the CPU Module for Each Operating Status 48
2.5 Operation Processing During Momentary Power Failure 50
2.6 Processing of Inputs and Outputs 51
2.6.1 Refresh processing 51
2.6.2 Direct processing 54
2.7 Interrupt Program 56
2.8 Executing Multiple Programs 61
2.8.1 Initial execution type program 63
2.8.2 Scan execution type program 65

2.8.3 Standby type program 66
2.8.4 Fixed scan execution type program 69
2.8.5 Changing the program execution type 72
2.9 Boot Operation 74
2.10 Programming Language 76
2.11 Communications with Intelligent Function Modules 77

PART 2 FUNCTIONS

CHAPTER 3 CPU MODULE FUNCTIONS 82

3.1 Function List 82
3.2 Constant Scan 85
3.3 Watchdog Timer (WDT) 87
3.4 Latch Function 88
3.5 Initial Device Value 91
3.6 Service Processing Setting. 94
3.7 Output Mode at Operating Status Change (STOP to RUN) 101
3.8 Input Response Time Setting 102
3.9 Error Time Output Mode Setting. 103
3.10 PLC Operation Mode at H/W Error Setting. 104
3.11 Security Function 105
3.11.1 File Password 32 105
3.11.2 Remote Password 110
3.12 Remote Operation 112
3.12.1 Remote RUN/STOP 112
3.12.2 Remote PAUSE 114
3.12.3 Remote RESET 116
3.12.4 Remote Latch Clear 117
3.13 Scan Time Measurement 118
3.14 Program List Monitor 121
3.15 Interrupt Program List Monitor 122
3.16 Monitor condition setting. 123
3.17 Local Device Monitor/Test. 128
3.18 External Input/Output Forced On/Off 131
3.19 Executional Conditioned Device Test 135
3.20 Sampling Trace. 143
3.21 Realtime Monitor Function 150
3.22 Writing Programs in RUN Status 151
3.22.1 Online change (ladder mode). 151
3.22.2 Online change (files) 152
3.22.3 Precautions for online change 153
3.23 Debug from Multiple Programming Tools 158
3.23.1 Simultaneous monitoring from multiple programming tools 159
3.23.2 Online change from multiple programming tools 160

3.24 Self-Diagnostic Function 161
3.25 Error Clear 166
3.26 LED Control Function 169
3.26.1 Methods for turning off the LEDs 169
3.26.2 LED indication priority 170
3.27 Module Error Collection Function 171
3.28 Latch Data Backup to Standard ROM 175
3.28.1 Latch data backup 176
3.28.2 Restoring backup data 179
3.29 Writing/Reading Device Data to/from Standard ROM 180
3.30 Module Model Name Read 181
3.31 CPU Module Change Function with SD Memory Card 182
3.31.1 Backup to SD memory card 184
3.31.2 Backup data restoration 192
3.32 Clock Function 196
3.33 Battery Life-Prolonging Function 198
3.34 Memory Check Function 199
3.35 Program Cache Memory Auto Recovery Function 200
3.36 Project Data Batch Save/Load Function 202
3.36.1 Batch save 203
3.36.2 Batch load 209
3.37 Predefined Protocol Function 215
3.37.1 Communications via Ethernet 215
3.37.2 Communications via RS-232 and RS-422/485 216
3.38 Serial Communication Function 227
3.39 iQ Sensor Solution Function 234

CHAPTER 4 DISPLAY UNIT FUNCTIONS 236

4.1 Function List 240
4.2 CPU Monitor/Test 241
4.2.1 Device monitor/test 241
4.2.2 Device clear 249
4.2.3 Forced on/off 251
4.2.4 Scan time monitor 254
4.2.5 Built-in I/O function monitor 255
4.2.6 Error display/clear 259
4.3 CPU Settings 263
4.3.1 Clock setting 263
4.4 Module Monitor/Test 265
4.4.1 Buffer memory monitor/test 265
4.5 Module Settings 270
4.5.1 Initial setting change 270
4.6 User Message 273
4.7 Memory Card Operation 274
4.7.1 File list/delete 274

4.7.2 Free space display 279
4.7.3 Batch save 280
4.7.4 Batch load 281
4.8 Options 283
4.8.1 Language setting 283
4.8.2 Contrast adjustment 284
4.8.3 Lighting period setting 285
4.8.4 Pop-up display 286

PART 3 DEVICES, CONSTANTS

CHAPTER 5 DEVICES 288

5.1 Device List 288
5.2 Internal User Devices 290
5.2.1 Input (X) 293
5.2.2 Output (Y). 293
5.2.3 Internal relay (M) 294
5.2.4 Latch relay (L) 294
5.2.5 Link relay (B) 294
5.2.6 Annunciator (F) 295
5.2.7 Link special relay (SB) 298
5.2.8 Edge relay (V) 298
5.2.9 Step relay (S) 299
5.2.10 Timer (T, ST) 299
5.2.11 Counter (C) 308
5.2.12 Data register (D). 311
5.2.13 Link register (W). 312
5.2.14 Link special register (SW) 313
5.3 Internal System Devices 314
5.3.1 Function devices (FX, FY, FD) 314
5.3.2 Special relay (SM) 316
5.3.3 Special register (SD) 316
5.4 Link Direct Devices 317
5.5 Module Access Devices 320
5.5.1 Intelligent function module device 320
5.6 Index Register/Standard Device Register (Z) 322
5.6.1 Index Register (Z) 322
5.6.2 Standard device register (Z). 323
5.6.3 Saving and restoration of the index register. 324
5.7 File Register (R, ZR). 326
5.8 Extended Data Register (D) and Extended Link Register (W) 330
5.9 Nesting (N) 335
5.10 Pointer (P). 336
5.10.1 Local pointer. 337
5.10.2 Common pointer. 338

5.11 Interrupt Pointer (I) 339
5.12 Other Devices 340
5.12.1 SFC block device (BL) 340
5.12.2 I/O No. specification device (U) 340
5.12.3 Macro instruction argument device (VD) 341

CHAPTER 6 CONSTANTS 342

6.1 Decimal Constant (K) 342
6.2 Hexadecimal Constant (H) 342
6.3 Real Number (E). 343
6.4 Character String (" "). 343

CHAPTER 7 CONVENIENT USAGE OF DEVICES 344

7.1 Global Device 344
7.2 Local Device 345

APPENDICES 350

Appendix 1 Parameter Setting 350
Appendix 1.1 List of parameter numbers 350
Appendix 1.2 PLC parameters. 353
Appendix 1.3 Network parameters. 374
Appendix 1.4 Remote password setting 375
Appendix 2 Added and Changed Functions. 376
Appendix 3 CPU Module Processing Time 378
Appendix 3.1 Time required for each processing in a scan time 378
Appendix 3.2 Factors that increase the scan time 385
Appendix 3.3 Realtime monitor function processing time 388
Appendix 4 Data Used in the CPU Module 389
Appendix 5 Character Codes Available in the Display Unit 391
Appendix 6 Differences Between LCPU and QnUCPU 403
Appendix 6.1 Specifications comparison 403
Appendix 6.2 Precautions for utilizing programs 404
Appendix 7 Precautions for Using GX Works2 and Differences with GX Developer 405
Appendix 8 Device Point Assignment Sheet 406

INDEX 407

REVISIONS 410
WARRANTY 413
TRADEMARKS 414

MANUAL PAGE ORGANIZATION

In this manual, pages are organized and the symbols are used as shown below.
The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.

"" is used for screen names and items.

1. shows operating procedures.

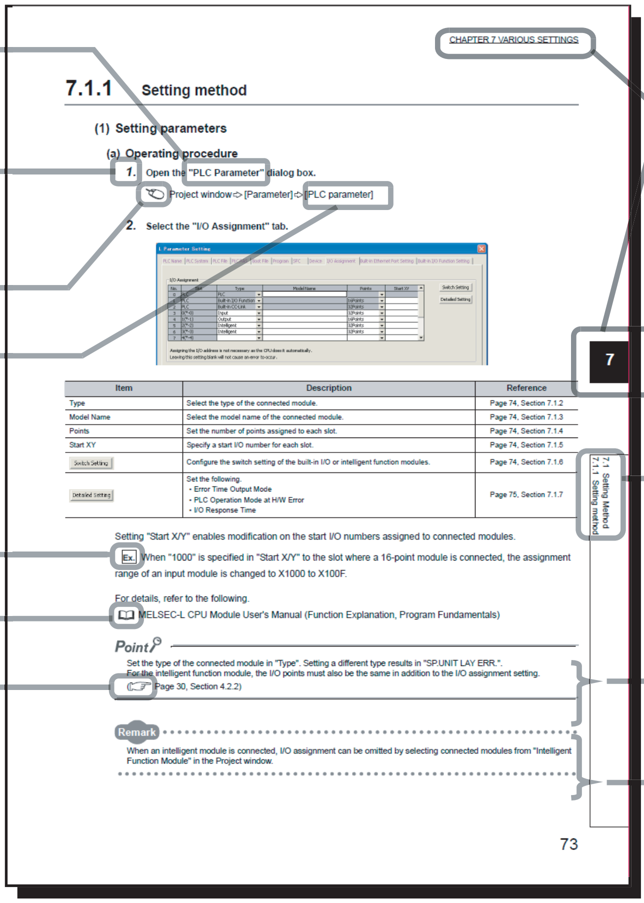
shows mouse operations.*1

[ ] is used for items in the menu bar and the project window.

Ex. shows setting or operating examples.

shows reference manuals.

shows reference pages.



The chapter of the current page is shown.

The section of the current page is shown.

Point shows notes that requires attention.

Remark shows useful information.

*1 The mouse operation example (for GX Works2) is provided below.

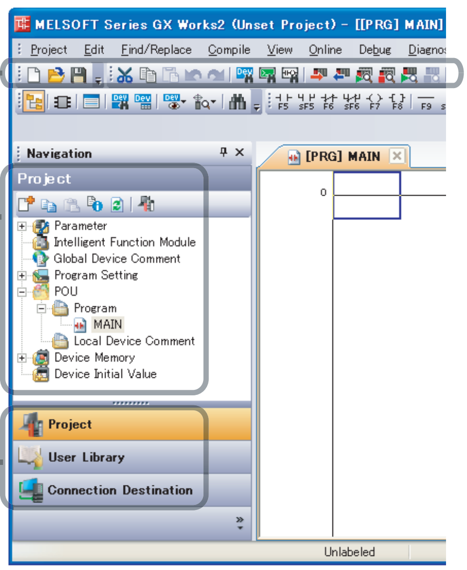
Menu bar

Ex. [Online] ⇨ [Write to PLC...]
Select [Online] on the menu bar, and then select [Write to PLC...].

A window selected in the view selection area is displayed.

Ex. Project window ⇨ [Parameter] ⇨ [PLC Parameter]
Select [Project] from the view selection area to open the Project window.
In the Project window, expand [Parameter] and select [PLC Parameter].

View selection area



TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Battery	A battery to be installed in the CPU module and used for backing up data such as the standard RAM data and latch device data in case of power failure. The Q6BAT, Q7BATN, and Q7BAT are available.
Display unit	A liquid crystal display to be attached to the CPU module
END cover	A cover to be attached to the right side of the rightmost MELSEC-L series module
Extension block	A block where an extension module is connected in an extension system
GX Works2	The product name of the software package for the MELSEC programmable controllers
GX Developer	
Main block	A block where a CPU module is connected in an extension system
SD memory card	Secure Digital Memory Card, which is a flash memory device. The NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD, L1MEM-2GBSD, and L1MEM-4GBSD are available.

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term and abbreviation	Description
Branch module	A MELSEC-L series branch module
Built-in Ethernet port LCPU	L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT
Extension module	A MELSEC-L series extension module
Programming tool	GX Works2, GX Developer
QnUCPU	A MELSEC-Q series Universal model QCPU

PART 1 PROGRAMMING

In this part, fundamental knowledge of programming is described.

CHAPTER 1 BASIC PROCEDURE FOR PROGRAMMING 19
CHAPTER 2 APPLICATION OF PROGRAMMING 31

CHAPTER 1 BASIC PROCEDURE FOR PROGRAMMING

This chapter describes the basic procedure for programming.

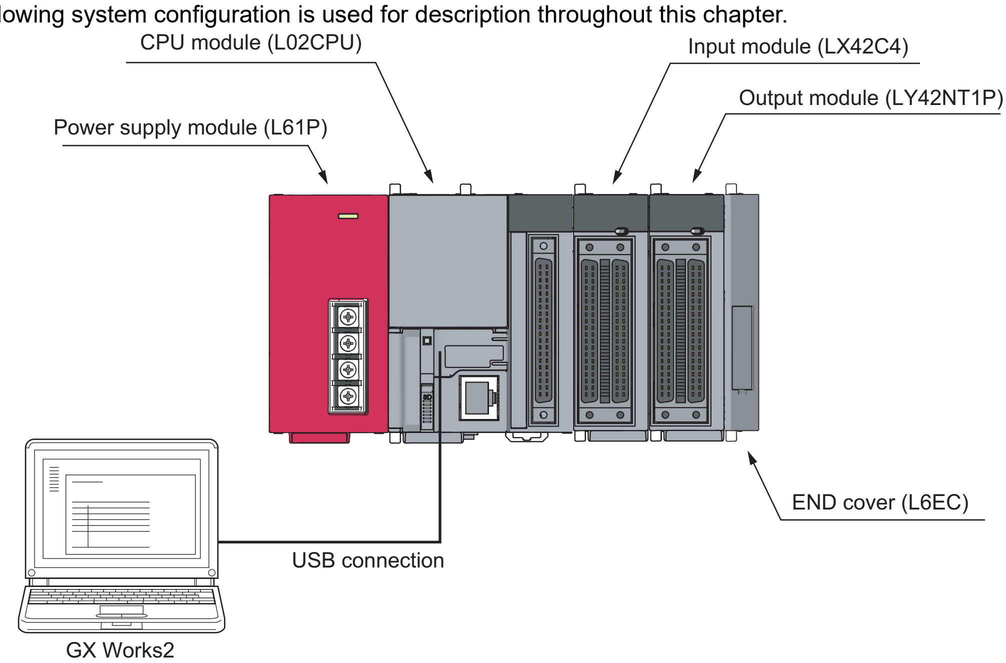
Start

Check column	Step	Description
☐	Creating projects	Create projects with GX Works2. (☞ Page 22, Section 1.2)
☐	Creating programs	Create programs. (☞ Page 23, Section 1.3)
☐	Converting programs	Convert created programs into ones that can be processed by the CPU module. (☞ Page 25, Section 1.4)
☐	Writing projects	Write the projects to the CPU module. For the first use of the CPU module, format the used memory beforehand. (☞ Page 26, Section 1.5)
☐	Checking operations	Debug the programs using the monitoring function. (☞ Page 28, Section 1.6)
☐	Saving projects	Save the projects. (☞ Page 30, Section 1.7)

End

1.1 System Configuration Example

The following system configuration is used for description throughout this chapter.



*1 Wiring of the power supply module and I/O modules are omitted in this illustration.

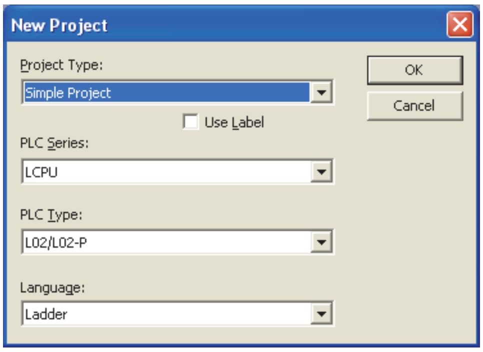
1.2 Creating a Project

A project is a set of information, such as programs and parameters, which is necessary to operate a programmable controller.
The following two projects are available.

- Simple project
- Structured project

Create a new project using GX Works2.

[Project] ⇨ [New...]



Item		Description
Project Type	—	Select a type of project to create. In this chapter, "Simple Project" is selected.
	Use Label	Select this checkbox when using a label for programming. In this chapter, this is not selected.
PLC Series		Select a series of the CPU module to use in the project. In this chapter, "LCPU" is selected.
PLC Type		Select a type of the CPU module (CPU module model) to use in the project. In this chapter, "L02/L02-P" is selected.
Language		Select a language of the program data to use for the new project. In this chapter, "Ladder" is selected.

Point

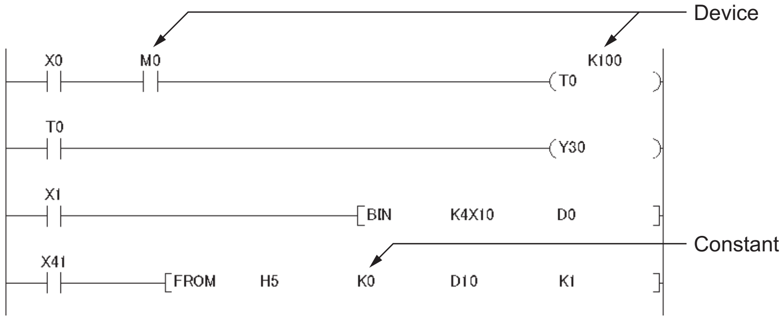
When performing communication between a programming tool and a CPU module through GOT or a network module, check the PLC type because the modules could be connected with wrong model names. If the modules are connected with wrong model names, data may not be written or read properly.

1.3 Creating a Program

1.3.1 Prior knowledge for creating a program

(1) Device and constants

Devices and constants, such as shown below, are used for creating a program. (☞ Page 288, CHAPTER 5)



(2) Concept of I/O numbers

I/O numbers are automatically assigned.

Power supply module	CPU module	Input module 64 points	Output module 64 points
	0000 to 000F	X0010 to X004F	Y0050 to Y008F

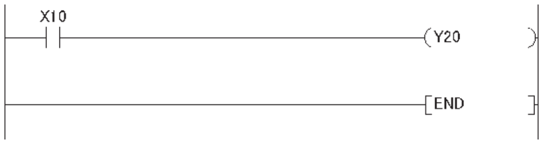
Users can also assign I/O numbers according to their purposes. (☞ Page 41, Section 2.2)

(3) Program configuration

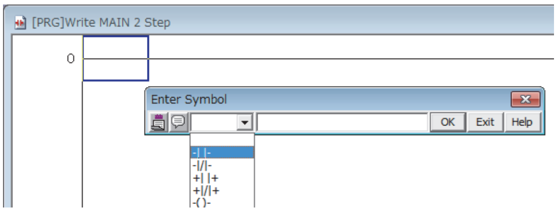
A main routing program, subroutine program, (☞ Page 46, Section 2.3.3), and interrupt program (☞ Page 56, Section 2.7) can be included in a program.

1.3.2 How to create a program

This section shows how to create the following sample program.

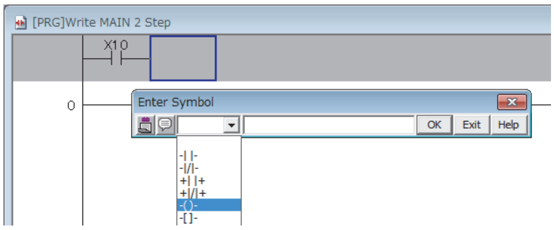


When X10 is turned on, Y20 turns on.



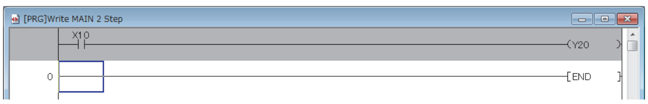
1. **To enter X10, type X10 at the original cursor position and select the contact shown in the left figure.**

↓



2. **To enter Y20, type Y20 and select the coil shown in the left figure.**

↓



The program has been created. In the next procedure, convert the program.

1.4 Converting a Program

Operation of a program is defined after converting its ladder.

[Compile] ⇨ [Build]

The program has been converted. In the next procedure, write the program to a CPU module.

Point

- To use a label, the program must be compiled.
  GX Works2 Version 1 Operating Manual (Common)
- After modifying a program, it must be compiled.

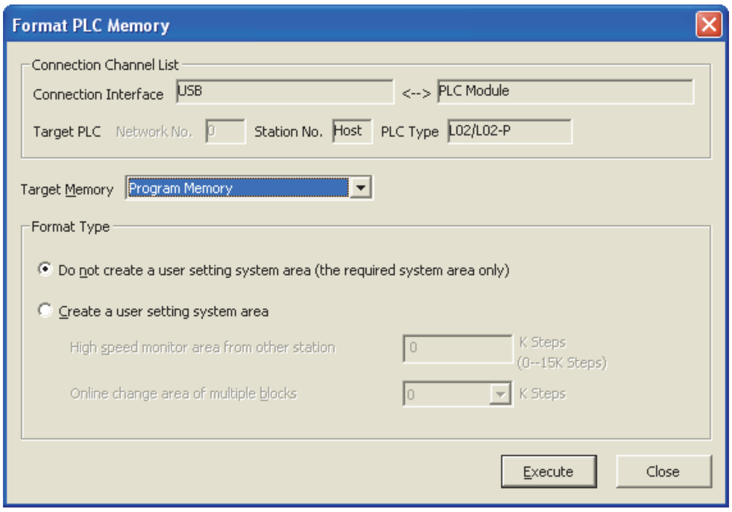
1.5 Writing a Project to the CPU Module

Write the project (the program and parameters described in Section 1.4.) to the CPU module. Note that, when the program is new, the memory (☞ Page 31, Section 2.1.1) is formatted so that a program can be written to it.

1.5.1 Formatting a memory

To format a memory, open the "Format PLC Memory" dialog box. In this chapter, a program memory is formatted so that a program can be written to it.

[Online] ⇨ [PLC Memory Operation] ⇨ [Format PLC Memory...]



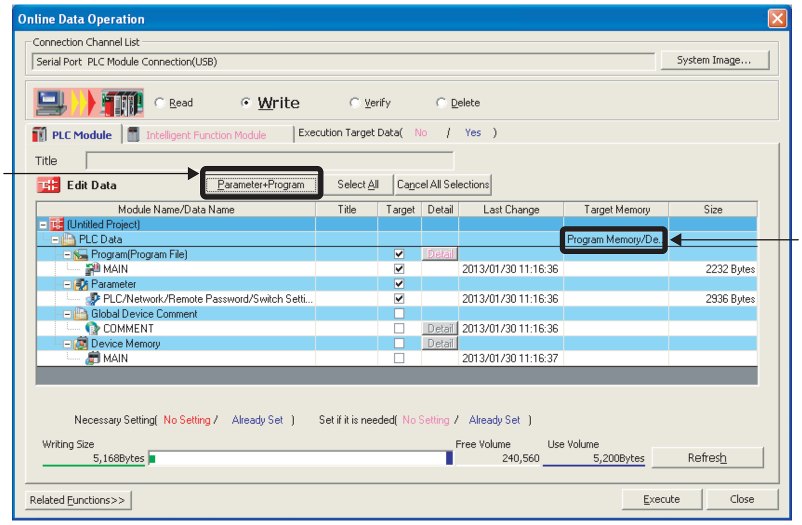
To check the capacity of the memory after formatting, open the "Online Data Operation" dialog box.

1.5.2 Writing to the CPU module

Open the "Online Data Operation" dialog box. In this chapter, a project is written to the program memory.

[Online] ⇨ [Write to PLC...]

2) Click this button to select the parameter and program checkboxes automatically.



1) Select the program memory.

The project has been written. In the next procedure, execute the program.

Point

Note that parameter setting is required to operate CPU modules. In this chapter, the procedure for parameter setting is not introduced since default values are used.(☞ Page 350, Appendix 1)

1.6 Checking an Operation of the CPU Module

To check an operation, execute the program written to the CPU module. In this chapter, operation is checked through the monitoring screen of GX Works2.

(1) Executing a program

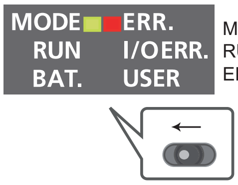
Before operating the CPU module, data written to the CPU module must be validated. To validate, power off and then on or reset the CPU module.

1. Before resetting the CPU module, check the current LED status.

MODE: On (green)
RUN : Off
ERR. : Flashing (red)

↓

2. Move the switch on the front of the CPU module to the RESET position. (One second or longer)



MODE: On (green)
RUN : Off
ERR. : Flashing (red)

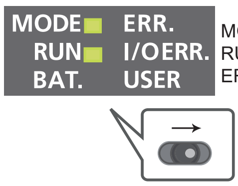
↓

3. Hold the switch until the ERR. LED turns off after flashing.

MODE: On (green)
RUN : Off
ERR. : Off

In the next procedure, run the CPU module. To run, use the switch on the CPU module.

4. Move the switch to the RUN position.



MODE: On (green)
RUN : On (green)
ERR. : Off

When the RUN LED is lit green, the program is being executed successfully.

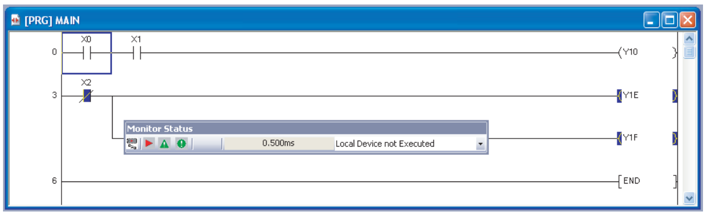
Point

By remote operation, CPU modules can be operated without using switches. (☞ Page 112, Section 3.12)

(2) Checking operation

Conductivity and power distribution status of contacts and coils can be checked by switching GX Works2 to the monitor mode.

[Online] ⇨ [Monitor] ⇨ [Start Monitoring]

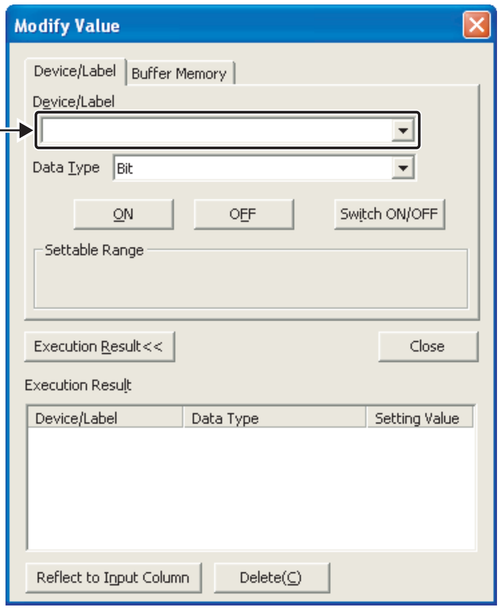


When X0 and X1 are turned on, Y10 turns on. (to turn on X0 and X1, place the cursor on them and double-click while holding the [Shift] key.) While contacts and coils are conducting, they are shown in blue.
Alternatively, device states can be checked through a display unit. (Page 241, Section 4.2.1)

Debug can be performed by forcibly turn on or off devices in the "Modify Value" dialog box.

[Debug] ⇨ [Modify Value...]

Enter a device to be turned on or off.



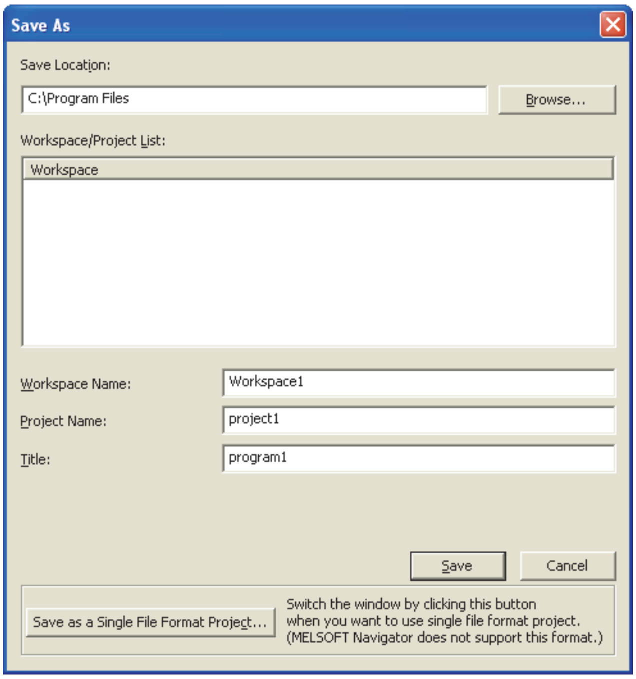
For details on current value changing, refer to the following.
GX Works2 Version 1 Operating Manual (Common)

If a program is edited during debugging, the program can be written to the CPU module even while the CPU module is in the RUN status. (Page 158, Section 3.23)

1.7 Saving a Project

To save a project, open the "Save As" dialog box.

[Project] ⇨ [Save As...]



Item	Description
Save Folder Path	Enter the storage destination folder (drive or path) of the workspace. Folders can be browsed for selection by clicking the  button.
Workspace/Project List	Select a workspace. Double-click "Workspace" to display a project list.
Workspace Name	Enter a name for the workspace.
Project Name	Enter a name for the project.
Title *1	Enter a title for the project.

*1 Projects can also be saved without titles.

CHAPTER 2 APPLICATION OF PROGRAMMING

This chapter describes applications of programming.

2.1 Memory and Files

2.1.1 Memory configuration

The following table lists the memory configuration of the CPU module.

CPU module	Memory configuration
L02SCPU, L02SCPU-P	Program memory, standard RAM, standard ROM
L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	Program memory, standard RAM, standard ROM, SD memory card

(1) Program memory

This memory stores programs and parameters required in processing of the CPU module.

(a) Processing a program

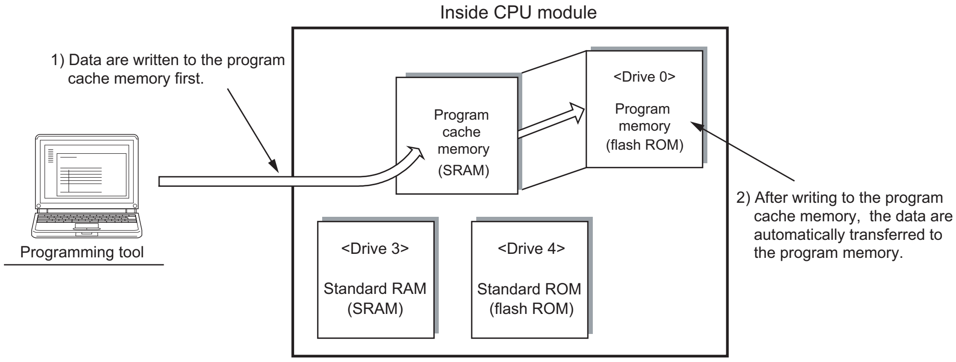
When a program is executed, data in the program memory are transferred to the program cache memory*1 at the following timings.

- Initial processing at power-on
- Initial processing at reset

*1 The program cache memory is used for program operations.

(b) Writing to the program memory

When a program is written to the program memory, it is temporarily written to the program cache memory, and then automatically transferred back to the program memory.



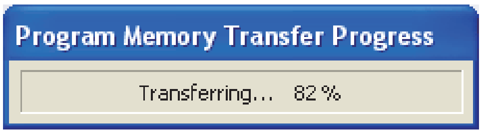
Point

While the CPU module is in the RUN status, automatic data transfer to the program memory can be disabled by setting.
(☞ Page 153, Section 3.22.3)

(c) Transfer confirmation to the program memory

Program transfer to the program memory can be checked by the following.

- On the dialog box below



- SM681 and SD681

Whether the transfer is in execution or complete can be checked by SM165.

(2) Standard RAM

This memory stores file register files, local device files, sampling trace files, and module error collection files.

(3) Standard ROM

This memory stores data such as device comments and PLC user data.

(4) SD memory card

This memory stores programs and parameters. To execute a program stored in the SD memory card, perform a boot operation. (☞ Page 74, Section 2.9)
An SD card memory is required when using the data logging function.

(5) Memory capacity

The following table shows the memory capacity of each memory.

<table>
<tr><th>CPU module</th><th>Program memory</th><th>Standard RAM</th><th>Standard ROM</th><th>SD memory card</th></tr>
<tr><td>L02SCPU, L02SCPU-P</td><td rowspan="2">80K bytes</td><td rowspan="2">128K bytes</td><td rowspan="2">512K bytes</td><td>—</td></tr>
<tr><td>L02CPU, L02CPU-P</td><td rowspan="3">Capacity of the SD memory card used</td></tr>
<tr><td>L06CPU, L06CPU-P</td><td>240K bytes</td><td rowspan="2">768K bytes</td><td>1024K bytes</td></tr>
<tr><td>L26CPU, L26CPU-P,
L26CPU-BT, L26CPU-PBT</td><td>1040K bytes</td><td>2048K bytes</td></tr>
</table>

(6) Memory and data to be stored

○: Storable, ×: Not storable

File type	Program Memory	Standard RAM	Standard ROM	SD memory card	File name and extension (any given name for ***)	Remarks
	Drive 0	Drive 3	Drive 4	Drive 2		
Parameter	○	×	○	○	PARAM.QPA	One file per drive
Intelligent function module parameters [*1]	○	×	○	○	IPARAM.QPA	One data per drive
Program	○	×	○	○	***.QPG	—
Device comment	○	×	○	○	***.QCD	—
Initial device value	○	×	○	○	***.QDI	—
File register	×	○ [*2]	×	×	***.QDR	—
Local device	×	○ [*2]	×	×	***.QDL	One file per module
Sampling trace	×	○ [*2]	×	×	***.QTD	—
PLC user data	×	×	○	○	***.CSV/BIN	—
Symbolic information[*4]	○	×	○	○	[*5]	—
Drive heading	○	×	○	○	QN.DAT	—
Device data storage file	×	×	○	×	DEVSTORE.QST	—
Module error collection file	×	○ [*2]	×	×	IERRLOG.QIE	—
Boot setting file	○	×	○	○	AUTOEXEC.QBT	—
Remote password	○	×	○	○	00000000.QTM	—
Latch data backup file	×	×	○	×	LCHDAT00.QBK	—
Backup data file	×	×	×	○	MEMBKUP0.QBP	—
System file for project batch save/load	×	×	×	○	SVLDINF.QSL	—
Device data file	×	×	×	○	DEVDATA.QDT	—
Data logging setting file	×	×	○	○	LOGCOM.QLG, LOG01 to 10.QLG	—
Data logging file	×	×	○[*3]	○	***.CSV	—
Menu definition file	×	×	○	○	MENUDEF.QDF	—
System file for the iQ Sensor Solution function (data backup/restoration)	×	×	×	○	SSBRINF.QSI	—
Backup data file for the iQ Sensor Solution function (data backup/restoration)	×	×	×	○	***.QBR[*6]	—
Predefined protocol setting file	×	×	○	○	ECPRTCL.QPT,CPRTCL.QPT	—

*1 Store parameters (PARAM.QPA) and intelligent function module parameters (IPARAM.QPA) in the same drive. Otherwise the intelligent function module parameters are invalid.
*2 Only one file can be stored.
*3 This drive cannot be selected as a storage file by the data logging function. To write data to this drive, perform Write PLC User Data.
*4 This is the data of label program configuration information.
GX Works2 Version 1 Operating Manual (Common)
*5 For Simple project (with a label): SRCINF1M.C32 and SRCINF2M.C32. For Structured project: SRCINF1I.C32 and SRCINF2I.C32.
*6 This file name depends on the connection type of the iQ Sensor Solution data backup/restoration. (iQ Sensor Solution Reference Manual)

2.1.2 Parameter-valid drive

CPU modules operate according to parameter settings. Systems automatically select parameters from those stored in the drives for CPU module operation, according to the following priority order.

[Priority order]
1) Drive 0 (program memory)
2) Drive 2 (SD memory card)
3) Drive 4 (standard ROM)

Point

- If parameters are set to be booted to an SD memory card, the above priority order is applied after the parameters are booted to the specified destination. (☞ Page 74, Section 2.9)
- The parameters used by the CPU module can be checked at "Parameter Valid Drive Information" under "PLC Status Information" on the "PLC Diagnostics" dialog box. (☞ Page 376, Appendix 2)

  [Diagnostics] ⇨ [PLC Diagnostics]

(1) Timing that the parameters take effect

The CPU module automatically searches for parameters in the following timing and operates according to the parameters stored in the drive.

- When the CPU module is powered off and then on
- When the CPU module is reset

(2) Precautions

Note that if the parameters are written while the CPU module is in operation, the timing that the parameters take effect varies.

(a) To write a parameter to the drive other than the one in operation:

The CPU module keeps operating according to the current parameters. When the CPU module is turned off and then on or is reset, newly written parameters take effect according to the priority order.

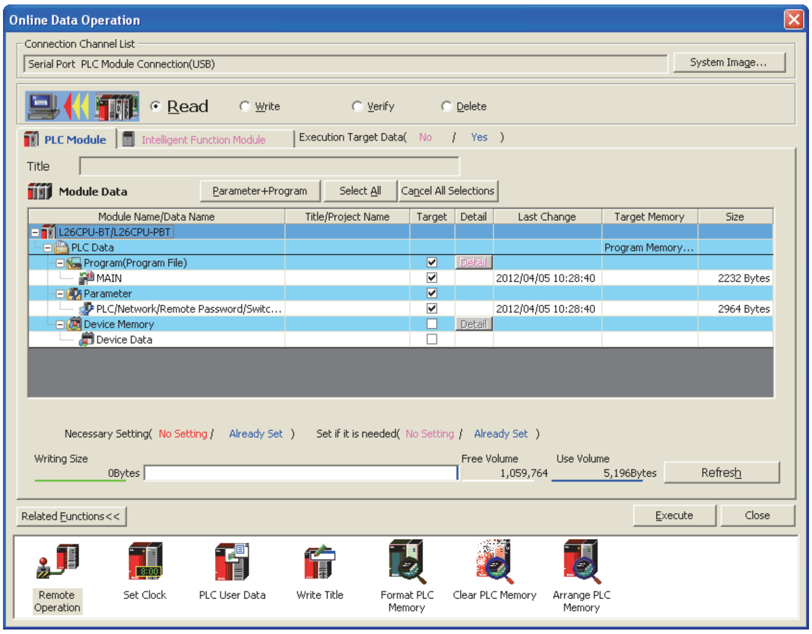
(b) To write a parameter to the drive where another parameter is currently processed:

Only the device settings become effective immediately after writing is complete. To make all the parameters effective, power off and then on or reset the CPU module.

2.1.3 Files

For the files written to the CPU module, information such as written date, file name (if created), and file size are appended to the file. By monitoring the file through Read from PLC, the file is displayed as shown below.

[Online] ⇨ [Read from PLC...]



Item	Description
File name	A file name consists of a name (up to 8 one-byte or 4 two-byte characters) and an extension. • File name: Create with uppercase characters only. • Extension: It is automatically appended according to the specified file type. The following cannot be used as a file name since they are the reserved words of Microsoft Windows®. COM1 to COM9, PRN, LPT1 to LPT9, NULL, AUX, CLOCK$, and CON When using characters, recognition of uppercase and lowercase differs depending on the memory. • Program memory, standard RAM, standard ROM: Not case-sensitive ("ABC" and "abc" are both considered to be "ABC".) • SD memory card: Case-sensitive
Update date	The date and time when the file was written to the CPU module is displayed.
Size	Except for file registers, at least 64 bytes are added to the capacity of the file created by a user. To display the latest data, click the "Refresh" button.

(1) Handling

(a) Power-off during online data operation (including reset)

Files in memory are not discarded if the CPU module is powered off or reset during online operation. However, for SD memory cards, doing so may result in data corruption. Stop accessing to an SD memory card, and then power off or reset the CPU module. All of SD memory card operations can be disabled by SM606 (SD memory card forced disable instruction). For the forced disablement of SD card, refer to the following.

MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(b) Simultaneous writing to the same file from multiple programming tools

While a file is being written, accessing the file from another programming tool is not allowed. Also, while a file is being accessed, writing data to the file from another programming tool is not allowed.
To write data to the same file from multiple programming tools, perform one by one.

(c) Simultaneous accessing to different files from multiple programming tools

Up to ten different files in a CPU module can be simultaneously accessed from multiple programming tools.

(d) File access into the SD memory card

Accessing into the SD memory card with the SP.FREAD or SP.FWRITE instruction may extend the scan time as the number of files stored in the SD memory card increases.

(2) File size

The size of a file used in the CPU module depends on the file type. Calculate the rough size of each file, referring to the following table.

File type	File size (unit: byte)
Parameter	Default • L02SCPU, L02SCPU-P: 2760 (The size can be increased by parameter setting.)[*1] • L02CPU, L02CPU-P: 2936 (The size can be increased by parameter setting.)[*1] • L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT: 2964 (The size can be increased by parameter setting.)[*1] Reference • Boot setting → 84 + (18 × (number of files))[*2] • With CC-Link setting (📖 MELSEC-L CC-Link System Master/Local Module User's Manual) • With CC-Link IE Field Network setting (📖MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual) • With Ethernet setting (📖 MELSEC-L Ethernet Interface Module User's Manual (Basic)) • With remote password setting → 92 + (the number of target modules × 10)[*2], Up to an increment of 172
Intelligent function module parameter	76 + (28 × the number of modules set) + parameter size for each utility [*2]
Program	228[*3] + 4 × ((number of steps) + (number of steps reserved for online change))
Device comment	74 + 72 + 8 + (total comment data size of each device) Comment data size per device = 10 + 10240 × a + 40 × b • a: Quotient of ((number of device points) ÷ 256) • b: remainder of ((number of device points ÷ 256)
Initial device value	66 + 44 (number of settings of the initial device value) + 2 × (total number of device points set to the initial device value) + 72 + 8
File register	2 × (number of device points)
Local device[*4]	70 + 6 × (set device type) + (Am + Av + B + Ct + Cst + Cc) × n • Am, Av = (((a1 + a2) ÷ 16) - ((a1 + 1) ÷ 16) + 1) × 2 • B = b × 2 • Ct, Cst, Cc = ((((c1 + c2) × 2) ÷ 16) - ((c1 × 2 + 1) ÷ 16) + 1) × 2 + c2 × 2 • Am, Av: Save area sizes of M (internal relay) and V (edge relay), respectively a1: Start device number of M or V a2: Number of points of M or V • B: Save area size of D (data register) and Z (index register) b: Total number of points of D and Z • Ct, Cst, Cc: Save area sizes of T (timer), ST (retentive timer), and C (counter), respectively c1: Start device number of T, ST, or C c2: Number of points of T, ST, or C • n: Number of programs (only the ones using local devices)
Sampling trace file	362 + (number of word device points + number of bit device points) × 12 + (N1 + N2 + number of word device points × 2 + (number of bit device points ÷ 16) × 2) × the number of traces (total number of executions) Apply the following values for N1 and N2 according to the items selected under Additional Information on the "Condition setting" tab of the "Trace Setting" dialog box. • N1: When "Time" is selected, apply "4". • N2: When "Program Name" is selected, apply "8".
PLC user data	Depends on the value and the number of data
Source information	Depends on the specifications of the programming tool
Drive heading	72
Device data storage file	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P: Setting value at formatting (2K to 32K) L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT: Setting value at formatting (2K to 1024K)
Module error collection file	76 + 64 (64 × (value set for the number of storable errors))
System file for project batch save/load	136 + (34 × (number of target drives) + 34 × (number of target files[*7])) + a + b + c • a: Total size of target file names (including " " . , and extension) • b: Add "34" when sampling trace is registered. • c: Add "34" when data logging is registered.
Device data file[*5]	70 + 8 × (number of target devices[*6]) + ((number of word device points) + (a ÷ 16) + (18 × b ÷ 16)) × 2 • a: Number of bit device points (other than timer, retentive timer, and counter) • b: Number of bit device points (timer, retentive timer, and counter)

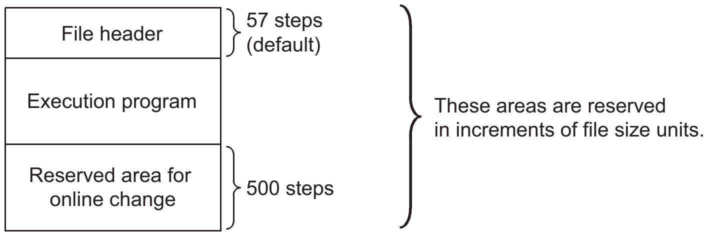
File type	File size (unit: byte)
Data logging setting file	Refer to the following. QnUDVCPU/LCPU User's Manual (Data Logging Function)
System file for the iQ Sensor Solution function (data backup/restoration)	Refer to the following. iQ Sensor Solution Reference Manual
Backup data file for the iQ Sensor Solution function (data backup/restoration)	
Predefined protocol setting file	65532

*1 The value is adjusted by the system so that the total number of bytes including the network parameter settings becomes multiple of four.
*2 The value is adjusted by the system so that the number of bytes becomes multiple of four.
*3 This value (228) is set by default. (It varies depending on parameter settings.)
*4 After the decimal point of a value obtained by a division part in the formula is rounded up.
*5 This file is used only when the project data batch save/load function is used.
*6 Count a contact and coil together as one device for the timer, retentive timer, and counter.
*7 The device data file is not included in the number of files.

(3) Program file structure

The following shows a program file structure.

Program file structure



Item	Description
File header	This area stores data such as the name, size, and created date of files. The file header size ranges from 43 to 59 steps (172 to 236 bytes) depending on the setting made in the Device tab of the PLC Parameter dialog box.
Execution program	This area stores the created program.
Reserved area for online change	This area is used when the number of steps is increased after writing data in the RUN status. (Default: 500 steps (2000 bytes) After the online change is complete, remaining number of steps for this area is displayed. The setting value can be changed in the "Online Data Operation" dialog box. (It can be changed while online change is performed.)

(4) Memory capacity

(a) File size unit for each memory area

When a file is written to the memory area, the unit of the stored file depends on the CPU module and memory area to be written. This unit is referred to as a file size unit.

<table>
<tr><th rowspan="2">CPU module</th><th colspan="4">Memory area</th></tr>
<tr><th>Program memory</th><th>Standard RAM</th><th>Standard ROM</th><th>SD memory card</th></tr>
<tr><td>L02SCPU, L02SCPU-P</td><td rowspan="3">1 step (4 bytes)</td><td rowspan="3">128 steps (512 bytes)</td><td rowspan="2">128 steps (512 bytes)</td><td>—</td></tr>
<tr><td>L02CPU, L02CPU-P, L06CPU, L06CPU-P</td><td rowspan="2">32K bytes</td></tr>
<tr><td>L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT</td><td>512 steps (2048 bytes)</td></tr>
</table>

(b) Calculation example of memory capacity

Ex. Memory capacity when the parameters and program are written to the program memory.

[Conditions]
- Program file to be written: MAIN.QPG (525 steps (2100 bytes))
- Parameter file to be written: PARAM.QPA (2936 bytes)
- Reserved area for online change: 500 steps (2000 bytes)

The memory capacity is calculated in units of file sizes of the CPU module to be written. For a program memory, the file size unit is 1 step (4 bytes).

• Calculation of program file size

The program file size can be found by the program size + reserved area for online change. Since a program is stored in units of file sizes (1 step), only the amount equal to the program size is occupied.

525 + 500 = 1025 steps (4100 bytes)

• Calculation of parameter file size

Since the parameter file capacity is 2936 bytes, it occupies 734 steps (2936 bytes) in the program memory.

[Result] Program size + parameter file capacity = 1025 + 734 = 1759 steps (7036 bytes)

Point

Memory capacity can also be calculated from a programming tool.

[Tool] ⇨ [Confirm Memory Size...]

For calculation of memory capacity using a programming tool, refer to the following.

Manual for the programming tool used

2.2 I/O Number

This section describes the I/O number assignment required for data communication between the CPU module or its built-in functions and I/O modules or intelligent function modules.

2.2.1 Concept of I/O number assignment

(1) Purpose of I/O number assignment

I/O numbers can be assigned with any given number for the following purposes.

(a) Reserving points for future module changes

The number of points can be reserved to prevent the I/O number modification when the current module is changed in the future to the one with the different number of occupied I/O points.

(b) Preventing I/O numbers from being changed

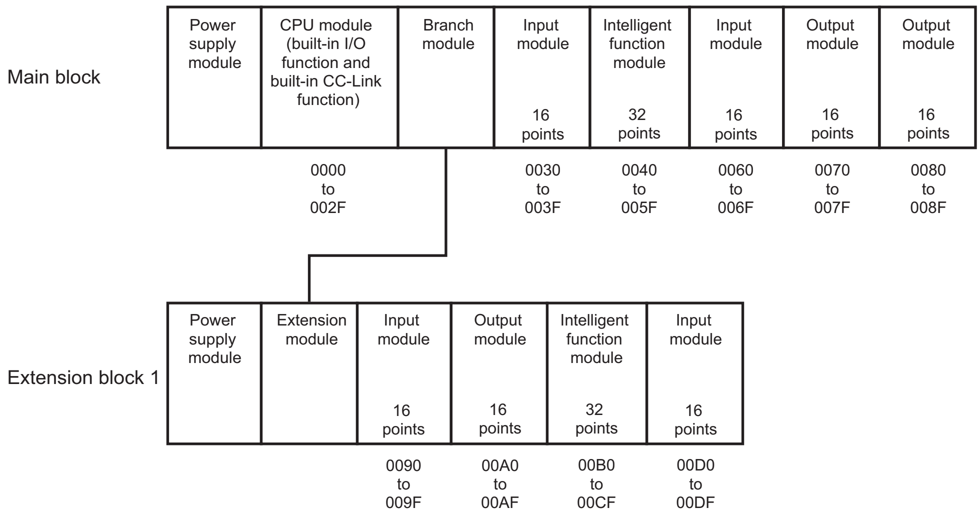
The change in the I/O numbers can be prevented when an I/O module or intelligent function module, whose occupied I/O points are other than 16, is removed due to failure.

(c) Changing the I/O numbers to those used in the program

When the I/O numbers used in the actual system differ from those in the program, the I/O numbers of each module can be changed to the ones in the program. If any of the I/O modules whose number of I/O points are other than 16 fails without I/O assignment setting, the I/O numbers assigned following to the failed module may change, leading to a malfunction. For this reason, making the I/O assignment setting is recommended.

(2) I/O number assignment

The I/O numbers are represented in hexadecimal. The following is an example of an I/O number assignment to the L26CPU-BT.



The start I/O numbers are as below.

Target	L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, L26CPU-P		L26CPU-BT, L26CPU-PBT	
	Default	**I/O assignment**	**Default**	**I/O assignment**
Built-in I/O	0000_H	Change allowed	0000_H	Change allowed
Built-in CC-Link	—	—	0010_H	Change allowed
Following modules (Main block) • The module to the right of a CPU module (when a branch module is not connected, or connected to the left of the END cover) • The module to the right of a branch module (when a branch module is connected to the right of a CPU module)	0010_H	Change allowed	0030_H	Change allowed
Following modules (Extension block) • The module to the right of an extension module (when a branch module is connected to the left of the END cover) • The module to the right of a branch module (when a branch module is connected to the right of an extension module)	Consecutive number from the previous block	Change allowed	Consecutive number from the previous block	Change allowed

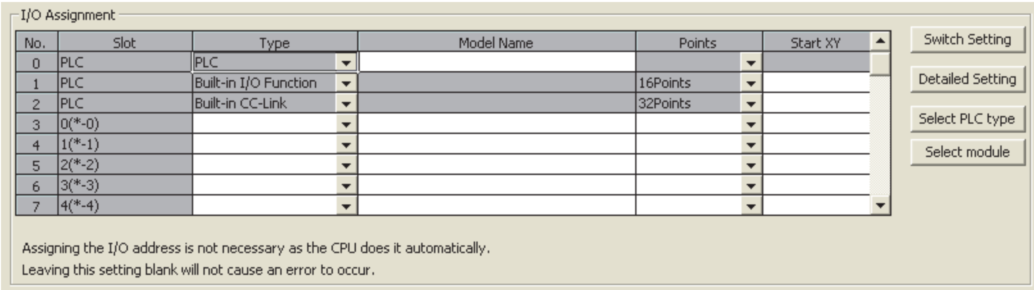
The start I/O number does not need to be assigned for the built-in Ethernet port, branch module, extension module, RS-232 adapter, RS-422/485 adapter, and END cover.

2.2.2 I/O number assignment

(1) Setting method

To assign I/O numbers, open the I/O Assignment tab. (☞ Page 353, Appendix 1.2)

Project window ⇨ [PLC Parameter] ⇨ [Parameter] ⇨ [I/O Assignment]



I/O Assignment

No.	Slot	Type	Model Name	Points	Start XY
0	PLC	PLC			
1	PLC	Built-in I/O Function		16Points	
2	PLC	Built-in CC-Link		32Points	
3	0(*-0)				
4	1(*-1)				
5	2(*-2)				
6	3(*-3)				
7	4(*-4)				

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

Point

In the I/O Assignment tab, the following settings are also available

- Response time from I/O (I/O Response Time) (☞ Page 102, Section 3.8)
- CPU module operation upon an error (☞ Page 103, Section 3.9)
- CPU module operation upon hardware failure of an intelligent function module (☞ Page 104, Section 3.10)
- Switch setting of intelligent function modules

(2) Precautions

(a) Type setting

The type set in the I/O Assignment tab must be the same as that of the connected module. Setting a different type may result as follows.

Connected modules	Type specified in the I/O Assignment tab	Results
• Input module • Output module • I/O combined module	• Intelligent • Branch • Branch (for LA1S Extension)	
Intelligent function module	• Input • Output • I/O Mix • Branch • Branch (for LA1S Extension)	"SP.UNIT.LAY.ERR." occurs.
Branch module	• Empty • Input • Output • I/O Mix • Intelligent	
Modules other than a branch module	Empty	Regarded as Empty
Other combinations	—	Error does not occur but incorrect operation may be caused. Or "PARAMETER ERROR" (error code: 3000) is detected.

For intelligent function modules, the number of I/O points must be the same.

(b) I/O points of modules

The number of I/O points for each module selected in the I/O Assignment tab is set in priority to those of connected modules.

- If the preset number of I/O points is less than those of connected I/O modules, the available points for the connected I/O module are decreased. (For example, If the number of I/O points is set to 16 points in the I/O Assignment tab to the module where a 32-point input module is connected, the second half 16 points of the 32-point input module are invalid.)
- If the preset number of I/O points exceeds those of connected I/O modules, the exceeded number of points will not be used.

Set the last I/O number within the I/O point range. Failure to do so causes "SP. UNIT LAY ERR.".

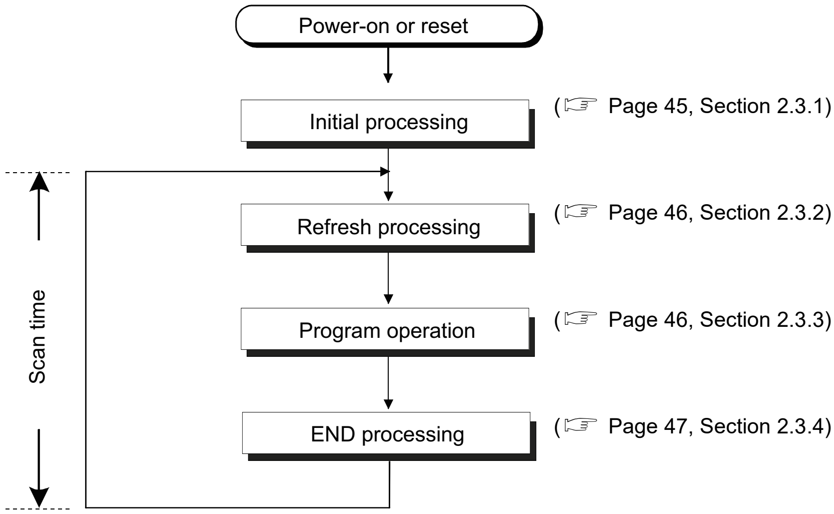
(c) Start XY setting

Note that, in the following two cases, the start XY setting may be duplicated. ("SP.UNIT.LAY.ERR." occurs if the start X/Y is duplicated.)

- Start XY values are not in the correct order.
- Modules with and without the start XY setting (automatically assigned module) are mixed.

2.3 Scan Time Structure

A CPU module sequentially performs the following processing in the RUN status. Scan time is the time required for all processing and executions to be performed.



2.3.1 Initial processing

The CPU module performs preprocessing required for program operations. The preprocessing is executed only once when any of the operations described in the following table is performed to the CPU module. When initial processing is completed, the CPU module is placed in the operation status set by the switch. (☞ Page 48, Section 2.4)

○: Performed, ×: Not performed

Initial processing item	CPU module status		
	Powered-on	Reset	Changed from STOP to RUN*1
The I/O module initialization	○	○	×
Boot from an SD memory card	○	○	×
PLC parameter check	○	○	○
Initialization of devices outside the latch range (bit device: off, word device: 0)	○	○	×
Automatic I/O number assignment to connected modules	○	○	○
CC-Link IE Field Network information setting	○	○	×
Intelligent function modules switch setting	○	○	×
CC-Link information setting	○	○	×
Ethernet information setting	○	○	×
Initial device value setting	○	○	○

*1 The operation indicates that the status is changed back to RUN without resetting the module after any program or parameter was changed in the STOP status. (The switch is set from STOP to RUN (the RUN LED will flash), then back to STOP and to RUN again.)
Note that the PLS, □P instruction (instruction for pulse conversion) may not be executed properly with the above operation. This is because the previous information may not be inherited depending on the program changes.

2.3.2 Refresh processing of input/output modules or intelligent function modules

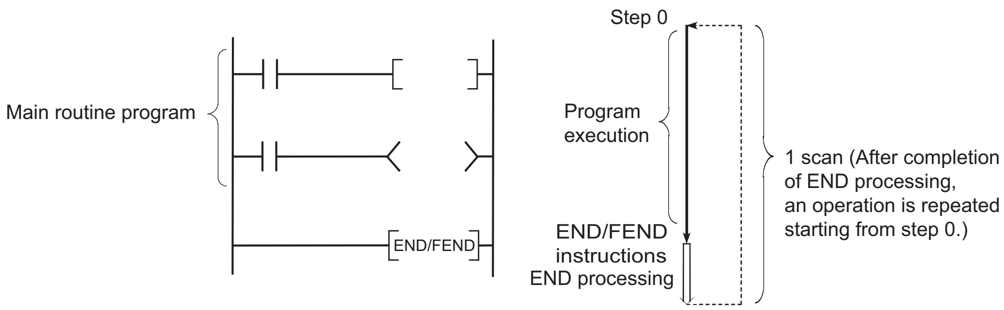
The CPU module performs the following before program operations.

- On/off data input from the input module or intelligent function module to the CPU module
- On/off data output from the CPU module to the output module or intelligent function module

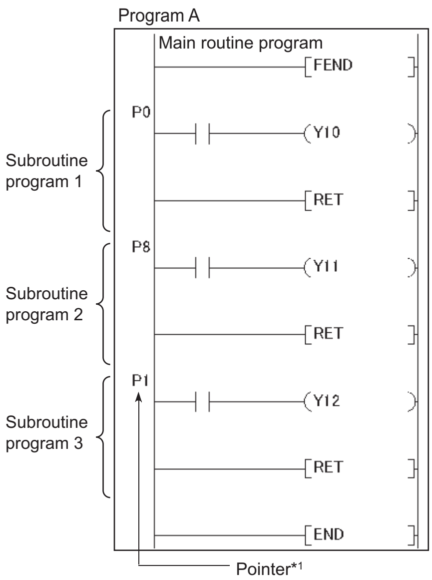
For details on refresh processing, refer to ☞ Page 51, Section 2.6.

2.3.3 Program operation

A main routine program repeatedly executes its operation from the step 0 to the END or FEND instruction. This program is executed from the step 0 for every scan.



A main routine program can be divided to main routine and subroutine programs. A subroutine program is from a pointer (P□) to the RET instruction, and is created between the FEND and END instructions. This program is executed only when it is called by a subroutine program call instruction (such as CALL(P), FCALL(P)) from a main routine program.



*1 Pointer numbers do not need to be specified in ascending order.

Use a subroutine program for the purposes such as the following.

- Handle the program that is executed several times in one scan as a subroutine program so that the entire number of steps can be reduced
- Handle the program that is executed under the specific condition as a subroutine program so that the scan time can be decreased

Point

- A subroutine programs can be managed as one separate program (standby type program). (☞ Page 66, Section 2.8.3)
- Subroutine programs can be configured with the nesting. (☞ Page 335, Section 5.9)
- Using an interrupt pointer in a subroutine program changes the program to an interrupt program. (☞ Page 56, Section 2.7)

2.3.4 END processing

END processing includes the following.

- Link refresh with CC-Link IE Field Network master/local module
- Link refresh with CC-Link IE Field Network Basic
- Auto refresh with intelligent function module
- Intelligent function module dedicated instruction processing
- Device data latch processing
- Service processing
- Watchdog time reset
- Self-diagnostics processing
- Special relay/special register value setting (only for those that should be set during END processing)

Point

When a constant scan is set, the results of processing performed in END processing are held for the period between after END processing is completed and until the next scan starts.

2.4 Operation Processing of the CPU Module for Each Operating Status

There are three types of operating status for the CPU module.

- RUN status
- STOP status
- PAUSE status

(1) Operation processing in the RUN status

RUN status is a status where program operations are repeatedly performed in a loop between the step 0 and the END (FEND) instruction.

(a) Output status when entering the RUN status

The CPU module outputs either of the following according to the output module parameter setting when its status is changed to RUN. (☞ Page 101, Section 3.7)

- Output (Y) status saved immediately before entering the STOP status
- Result of operations performed for one scan after entering the RUN status

(b) Processing time required before operations

The processing time required for the CPU module to start program operations after its operating status is changed from STOP to RUN varies depending on the system configuration and/or parameter settings. (Normally, it takes one to three seconds.)

(2) Operation processing in the STOP status

Stop status is a status where the operation is stopped. The CPU module status will be changed to STOP when a stop error occurs.

(a) Output status when entering the STOP status

When entering the STOP status, the CPU module saves data in the output (Y) and turns off all outputs. The device memory other than that of the output (Y) will be held.

(3) Operation processing in the PAUSE status

PAUSE status is a status where program operations are stopped after operations are performed for one scan, holding the output and device memory status.

(4) Operation processing when operating status of the CPU module changed

The operation processing of CPU module performs the following according to the RUN/STOP status.

RUN/STOP status	CPU module operation processing			
	Program operation processing	External output	Device memory	
			M, L, S, T, C, D	Y
RUN → STOP	The CPU module executes the program until the END instruction and stops.	The CPU module saves the output (Y) status immediately before its status is changed to STOP and turns off all the outputs.	The CPU module holds the device memory status immediately before its status is changed to STOP.	The CPU module saves the output (Y) status immediately before its status is changed to STOP and turns off all the outputs.
STOP → RUN	The CPU module executes the program from the step 0.	The CPU module outputs data according to the output mode parameter setting. (☞ Page 101, Section 3.7)	The CPU module holds the device memory status immediately before its status is changed to STOP. Note that the CPU module uses initial device values if those values are present. Local device data are cleared.	The CPU module outputs data according to the output mode parameter setting. (☞ Page 101, Section 3.7)

Point

The CPU module performs the following in any of the RUN, STOP, or PAUSE status.

- Refresh processing with I/O modules
- Refresh processing with CC-Link IE Field Network master/local module
- Refresh processing with CC-Link IE Field Network Basic
- Auto refresh processing with intelligent function modules
- Intelligent function module dedicated instruction processing (completion processing only)
- Self-diagnostics processing
- Service processing
- Special relay/special register value setting (only for those that should be set during END processing)

Even if the CPU module is in the STOP or PAUSE status, the following operations can be executed.

- I/O monitor or test operation from a programming tool
- Read/Write data from/to external devices using the MC protocol

2.5 Operation Processing During Momentary Power Failure

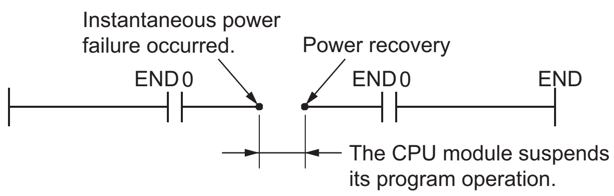
When the input voltage supplied to the power supply module drops below the specified range, the CPU module detects a momentary power failure and performs the following operation.

(1) When a momentary power failure occurs for a period shorter than the allowable power failure time

The CPU module registers error data and suspends the operation processing. The CPU module, however, continues measurement in the timer device and holds the output status.

- When resume start is specified for the SFC program, data in the system is saved.
- The CPU module restarts its operation processing.
- Even if operation processing is suspended due to a momentary power failure, the CPU module continues the measurement of the watchdog timer (WDT).

Ex. When the WDT setting of PLC parameter is 200ms and the scan time is 190ms, if a momentary power failure occurs for 15ms, "WDT ERROR" occurs.



(2) When a momentary power failure occurs for a period longer than the allowable power failure time

The CPU module starts its operations initially and the operation processing will be the same as that when either of the following is performed.

- Powering off and then on the CPU module
- Resetting the CPU module

2.6 Processing of Inputs and Outputs

The CPU module can batch-access the general-purpose I/O*1, I/O modules, and intelligent function modules before program operation (refresh processing). Note that I/O processing can be performed upon execution of an instruction by using a direct access device in the program. (direct processing)
Compared to the refresh processing, data acquisition is faster in direct processing while more time is required for processing of each instruction.

*1 The general-purpose input is an input signal received by the general-purpose input function of the built-in I/O function. The general-purpose output is an output signal sent by the general-purpose output function of the built-in I/O function.

2.6.1 Refresh processing

(1) Outline of the processing

The following describes the details of the refresh processing.

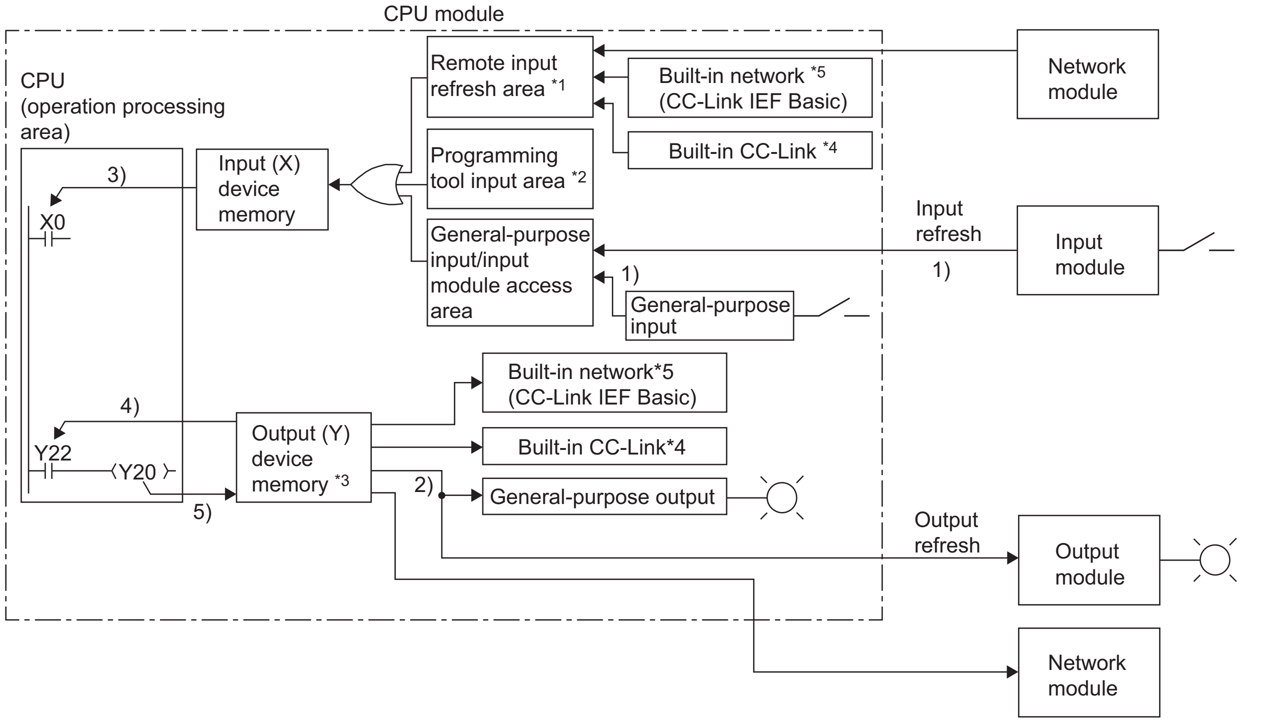
(a) Input

Before starting a program operation, the ON/OFF data in the general-purpose module and input module are collectively loaded to the General-purpose input/input module access area of the CPU module. When a program is executed, the ON/OFF data in the input (X) device memory are used for operation.

(b) Output

The program operation result for output (Y) is output to the output (Y) device memory for every operation. Before starting a program operation, the ON/OFF data in the output (Y) device memory are collectively output to the general purpose output or output module.

(c) Process flowchart



*1 The remote input refresh area indicates the area to be used when refresh is set to the input (X) in CC-Link IE Field Network, CC-Link IE Field Network Basic, or CC-Link. Data in the remote input refresh area are refreshed during END processing.

*2 Data in the programming tool input area can be turned on or off by the following:
- · Test operation of a programming tool
- · Writing data from a network module
- · Writing data from an external device using the MC protocol
- · Writing data using the simple PLC communication function

*3 Data in the output (Y) device memory can be turned on or off by the following operation.
- · Test operation of a programming tool
- · Writing data using a program
- · Writing data from a network module
- · Writing data from an external device using the MC protocol
- · Writing data using the simple PLC communication function

*4 This applies only to the L26CPU-BT and L26CPU-PBT.

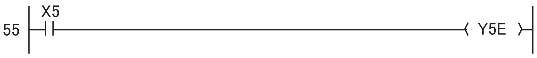
*5 This applies only to the Built-in Ethernet port LCPU.

Item	Description
Input refresh	Before program operation, input data (1)) are collectively read out from the general-purpose input and input modules, the OR processing with the programming tool input area and remote input refresh area is executed, and then the data are stored in the input (X) device memory.
Output refresh	Before program operation, data in the output (Y) device memory (2)) are collectively output to the output module.
Execution of an input contact instruction	Input data (3)) are read out from the input (X) device memory and the program is executed.
Execution of an output contact instruction	Output data (4)) are read from the output (Y) device memory and the program is executed.
Execution of the OUT instruction	The operation result of the program (5)) are stored to the output (Y) device memory.

(2) Response delay

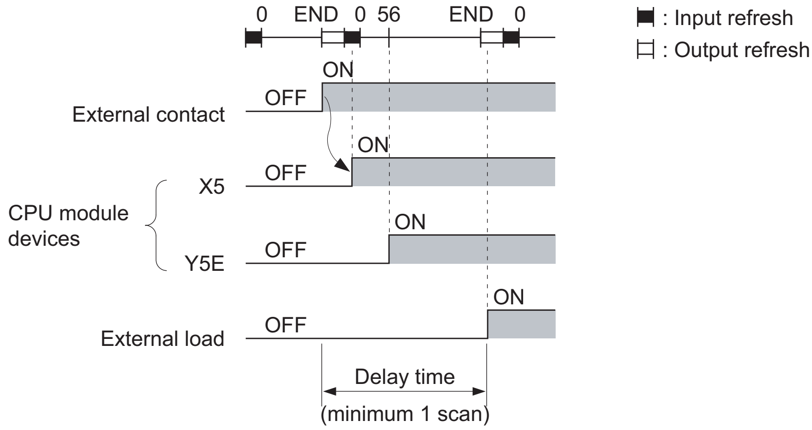
An output response which corresponds to the status change in the input module delays for two scans (maximum) depending on the on timing of an external contact. The following shows response delay examples.

[Example]

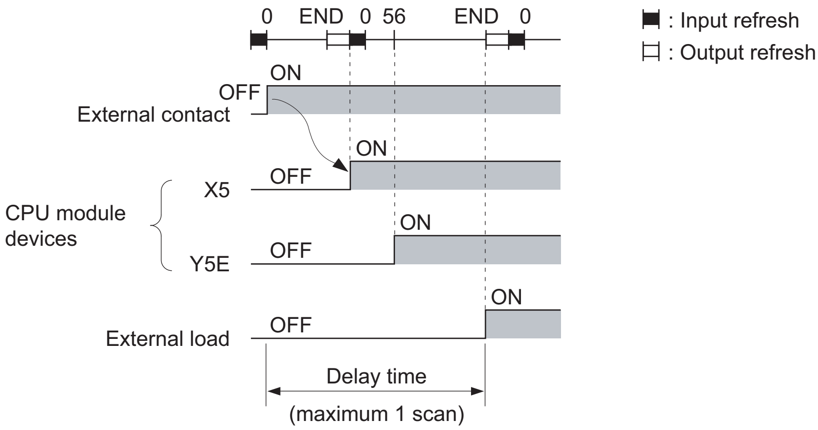


A program that turns on the output Y5E when the input X5 is turned on.

- Y5E turns on at the earliest timing



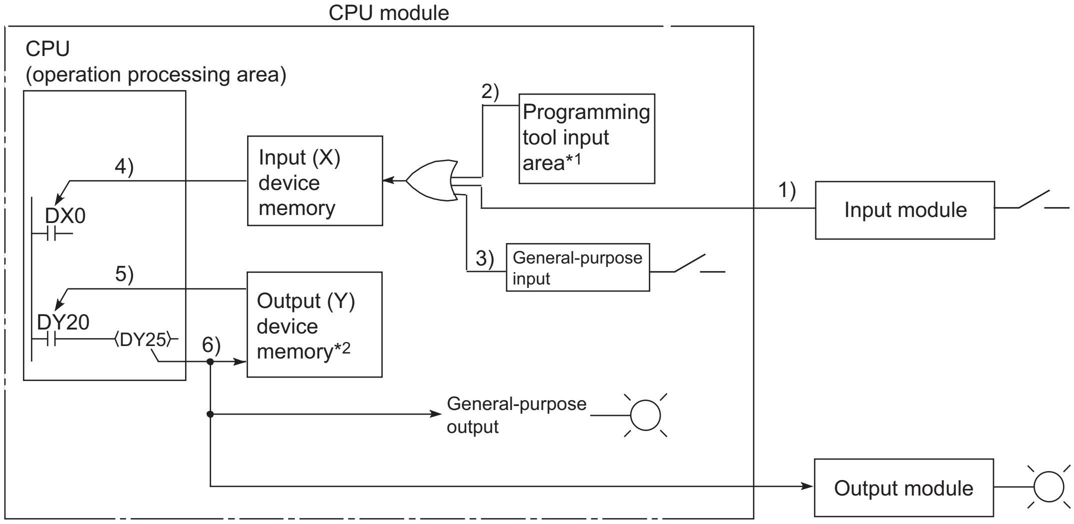
- Y5E turns on at the slowest timing



2.6.2 Direct processing

(1) Outline of the processing

Data input and output are performed using a direct access input (DX) or direct access output (DY).



*1 Data in the programming tool input area can be turned on or off by the following:
· Test operation of a programming tool
· Writing data from a network module
· Writing data from an external device using the MC protocol
· Writing data using the simple PLC communication function

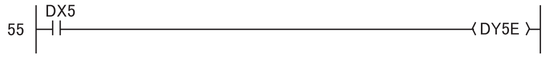
*2 Data in the output (Y) device memory can be turned on or off by the following operation.
· Test operation of a programming tool
· Writing data to a program
· Writing data from a network module
· Writing data from an external device using the MC protocol
· Writing data using the simple PLC communication function

Item	Description
Execution of an input contact instruction	The OR processing is performed between the input information of the input module (1)) and the input data of the programming tool (2)) or general-purpose input(3)). The result is stored in the input (X) device memory and is used as input data (4)) to execute the program.
Execution of an output contact instruction	Output data in the output (Y) device memory are read out (5)) and the program is executed.
Execution of the OUT instruction	The operation result of the program (6)) are output to the general-purpose output and output module, and stored in the output (Y) device memory.

(2) Response delay

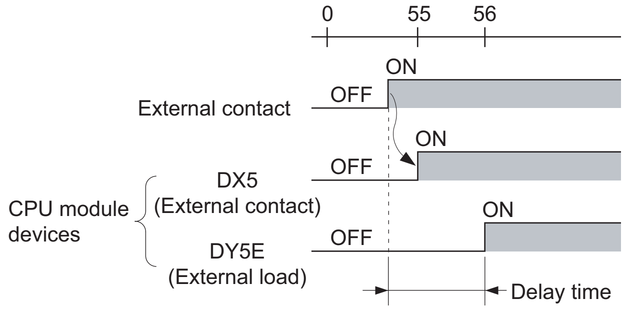
An output response which corresponds to the status change in the input module delays for one scan (maximum) depending on the on timing of an external contact. The following shows response delay examples.

[Example]

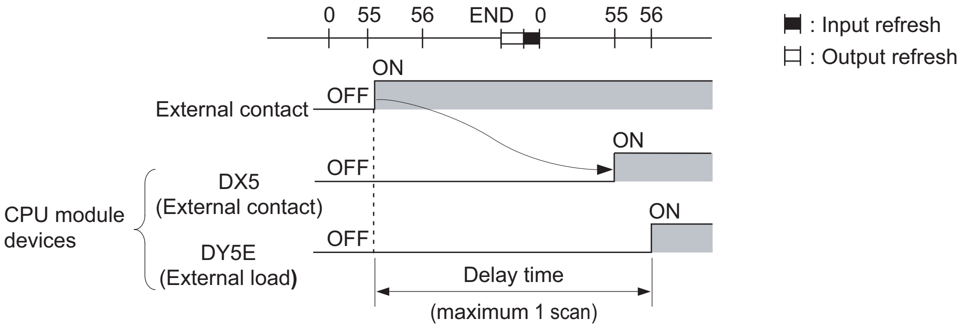


A program that turns on the output Y5E when the input X5 is turned on.

- DY5E turns on at the earliest timing

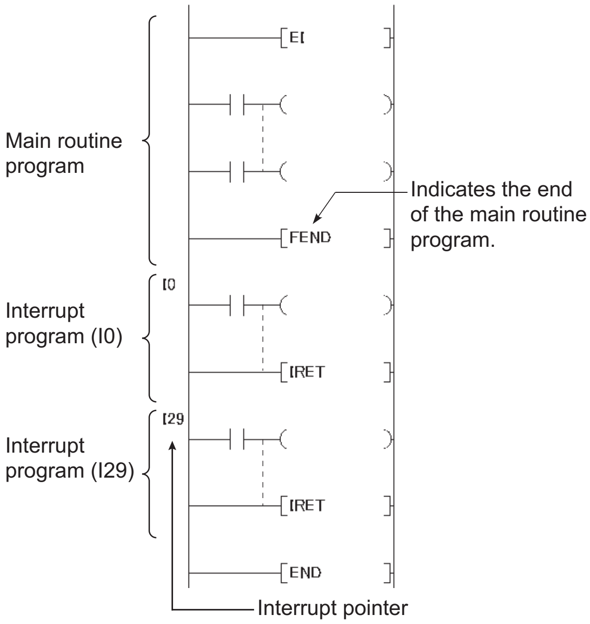


- DY5E turns on at the slowest timing

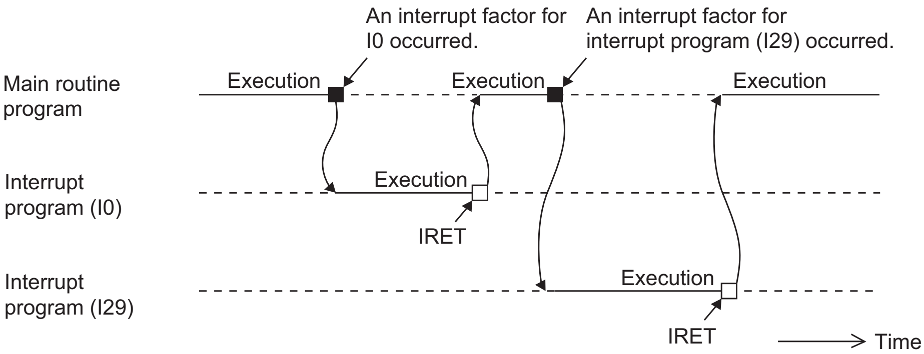


2.7 Interrupt Program

An interrupt program is from an interrupt pointer (I□) to the IRET instruction.

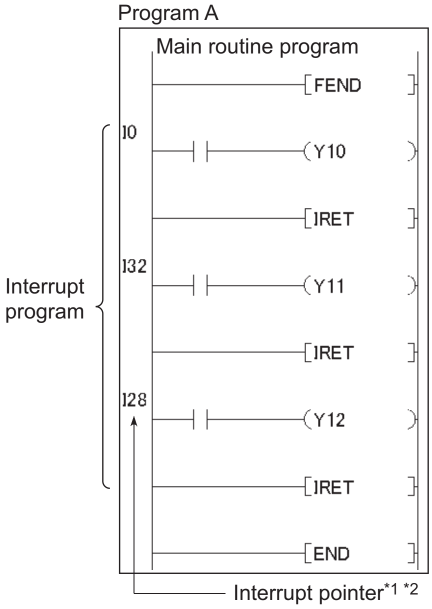


The interrupt pointer number (I□) varies depending on the interrupt factor. (☞ Page 339, Section 5.11) When an interrupt factor occurs, an interrupt program of the interrupt pointer number corresponding to that factor is executed. (Interrupt program are executed only after the corresponding interrupt factor occurs.)



(1) Creating an interrupt program

Create interrupt programs between the FEND and END instructions in the main routine program. Before executing any of interrupt programs of I0 to I15, I23 to I31, or I50 to I255, allow an interrupt by the EI instruction.



*1 The pointer numbers do not need to be specified in ascending order.
*2 Ensure that there is no overlap of interrupt pointer numbers.

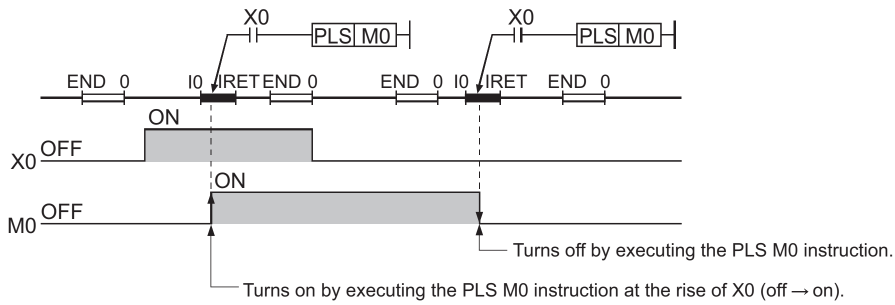
Point

Interrupt programs can be managed as one separate program (standby type program). (☞ Page 66, Section 2.8.3)

(2) Restrictions on creating an interrupt program

(a) PLS and PLF instructions

The PLS and PLF instructions perform off processing in the next scan of which the instruction is executed. Therefore, the device which is turned on by the instruction remains on until the same instruction is reexecuted.



(b) EI and DI instructions

Do not execute the EI or DI instruction during interrupt program execution.

(c) Timer (T) and counter (C)

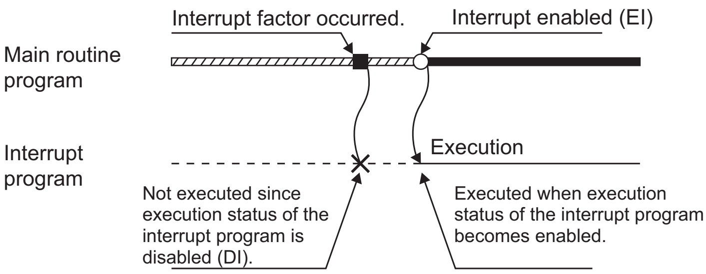
Do not use the timer (T) and counter (C) in interrupt programs.

(3) Operation when an interrupt factor occurs

The following restrictions are applied to the interrupt program depending on the interrupt factor occurrence timing.

(a) When an interrupt factor occurs before the interrupt program execution status is enabled

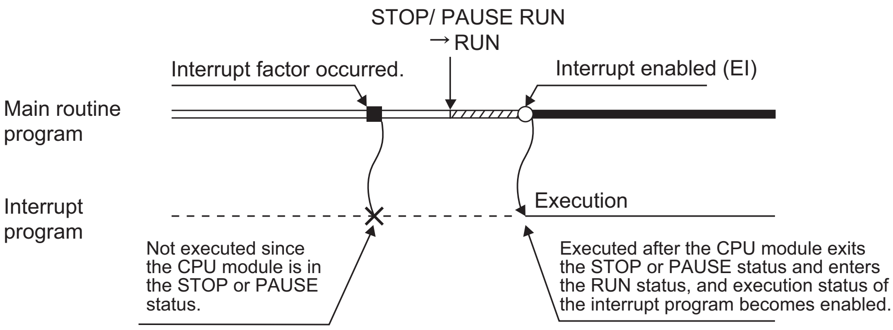
The CPU module stores the interrupt factor occurred. As soon as the interrupt program execution status is enabled, the CPU module executes the interrupt program corresponding to the stored interrupt factor.



When the same interrupt factor occurs more than one time before the interrupt program execution status is enabled, the interrupt factors of I0 to I15, I28 to I31, I45, I50 to I255, and fixed scan execution type programs are stored only once. Note that all interrupt factors occurred are discarded if they are masked by the IMASK instruction.

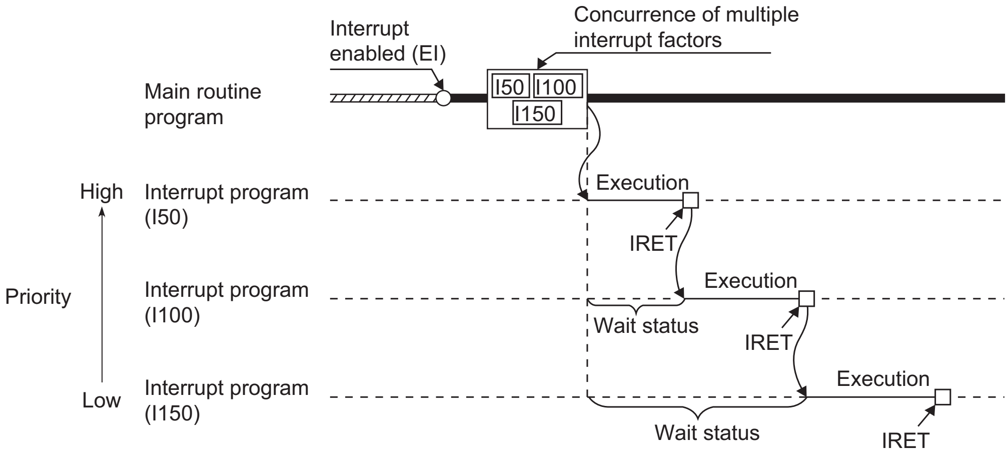
(b) When an interrupt factor occurs in the STOP or PAUSE status

The CPU module executes the interrupt program as soon as the interrupt program execution status is enabled after the CPU module status is changed to RUN.



(c) When multiple interrupt factors simultaneously occur in the interrupt program execution enabled status

The interrupt programs are executed in the order of priority of the interrupt pointers (I□) (☞ Page 339, Section 5.11).
Other interrupt programs have to wait until processing of the interrupt program being executed is completed.

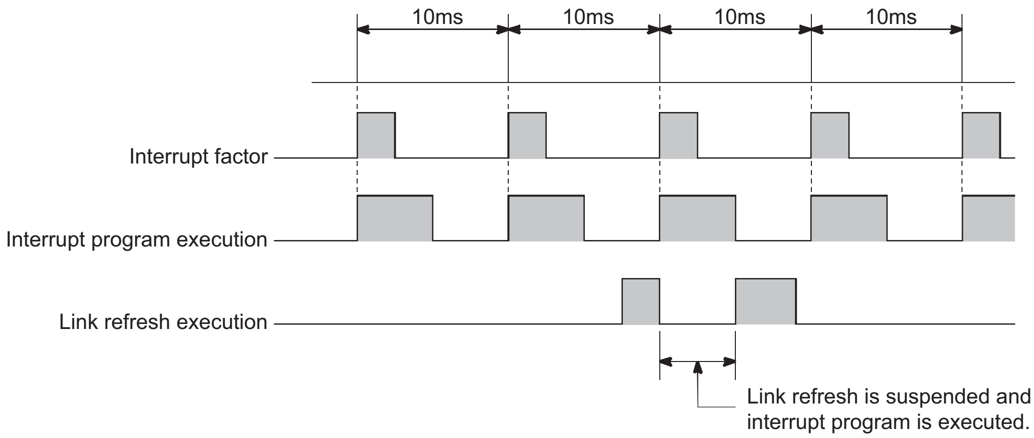


(d) When the same interrupt factor as that of the interrupt program being executed occurs

When the same interrupt factor as that of the interrupt program being executed occurs more than one time before completion of interrupt program processing, the interrupt factors of I0 to I15, I45, and I50 to I255 are stored only once, and then the interrupt program corresponding to each stored interrupt factor is executed after completion of current interrupt program execution. The interrupt factors of I28 to I31 and fixed scan execution type programs are all stored, and then all the interrupt programs corresponding to interrupt factors are executed after completion of current interrupt program execution.

(e) When an interrupt factor occurs during link refresh

The link refresh is suspended and an interrupt program is executed. Even if the Block data assurance per station setting is enabled in the CC-Link IE or MELSECNET/H network, this setting does not work when a device set as a refresh target is used in the interrupt program. In the interrupt program, do not use any refresh target device. For the Block data assurance per station setting, refer to the following.
📖MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual



(f) When an interrupt factor occurs during END processing

When the constant scan function is used and an interrupt factor occurs during the waiting time in END processing, an interrupt program corresponding to the interrupt factor is executed.

(g) When an interrupt factor occurs during access to another module

When an interrupt factor occurs during access to another module (during service processing or instruction processing), the interrupt program becomes standby status until the service processing or the instruction in execution is completed. To shorten the wait time of the interrupt, reduce the amount of data that access to other modules.

(4) Processing at program execution type change

When the program execution type is changed from the scan execution type to the interrupt, the CPU module saves and restores the following data. (☞ Page 324, Section 5.6.3)

- Data in the index register
- File register block number

Whether to save and restore the data above can be set by parameter. When the data are not saved or restored, the overhead time of the corresponding interrupt program can be shortened. (☞ Page 378, Appendix 3.1)

(5) Precautions

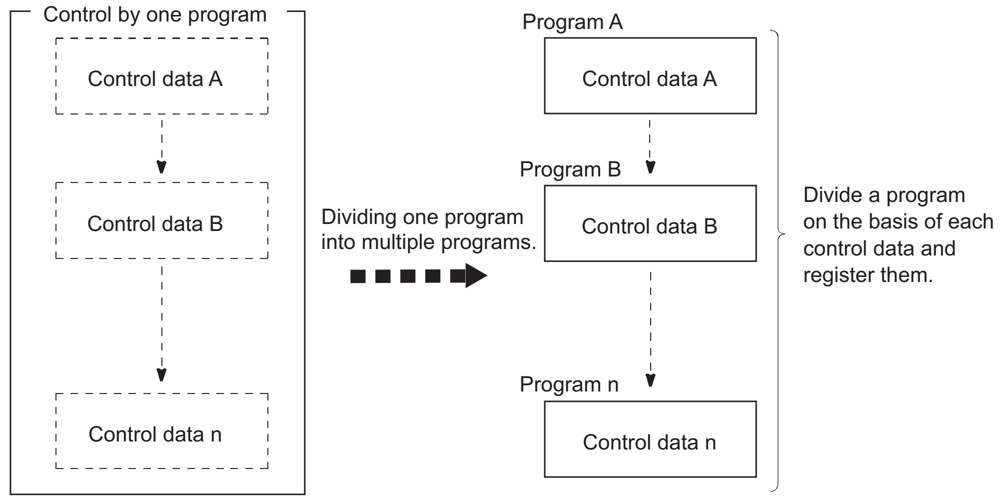
(a) When the same device is used

During execution of an instruction in a main routine program, an interrupt program may be executed, suspending the processing of the instruction being executed, resulting in a device data inconsistency. Take the following measures to prevent device data inconsistency.

- Do not directly specify the device where the data are written by the interrupt program in the main routine program. Use the data in another device by moving the data with the transfer instruction.
- Disable interrupts with the DI instruction if instructions that may cause inconvenience for the main routine program are used. Note that interrupts do not occur during accessing the device of the corresponding argument of the instruction. For this reason, data inconsistency does not occur in units of arguments.

2.8 Executing Multiple Programs

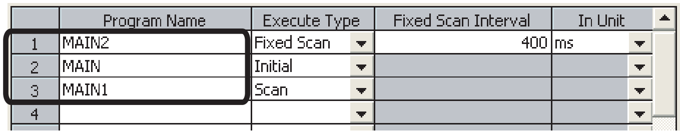
Multiple programs can be stored in the CPU module by changing the file names of programs. Dividing a program by process or function allows simultaneous programming by multiple engineers.



The following settings are required after creating multiple programs.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Program]

Programs are executed in this order.

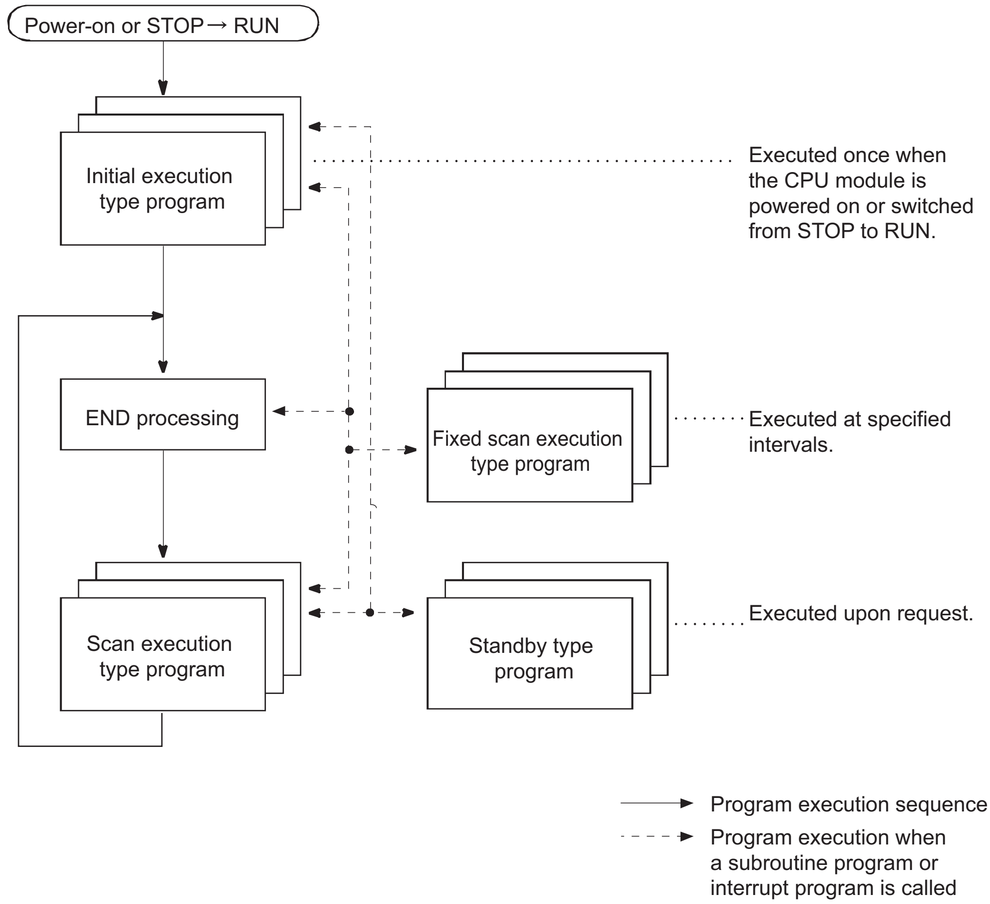


Setting item		Description	Reference
Program Name		Enter the name (file name) of the program to be executed in the CPU module.	—
Execute Type	"Initial" (initial execution type program)	This program is executed only once when the CPU module is powered on or its status is switched from STOP to RUN.	Page 63, Section 2.8.1
	"Scan" (scan execution type)	This program is executed once in every scan, starting in the next scan of which the initial execution type program is executed and later.	Page 65, Section 2.8.2
	"Wait" (standby type program)	This program is executed only when requested.	Page 66, Section 2.8.3
	"Fixed Scan" (fixed scan execution type program)	This program is executed at time intervals specified in "Fixed scan interval" and "In Unit". • Fixed Scan Interval: Enter the execution interval of fixed scan execution type program. The setting range varies depending on the setting unit. · When the unit is "ms": 0.5 to 999.5ms (in increments of 0.5ms) · When the unit is "s": 1 to 60s (in increments of 1s) • In Unit: Select the unit ("ms" or "s") of the fixed scan interval.	Page 69, Section 2.8.4

How to use a local device file can be set for each program by proceeding to another dialog box. (☞ Page 347, Section 7.2 (2))

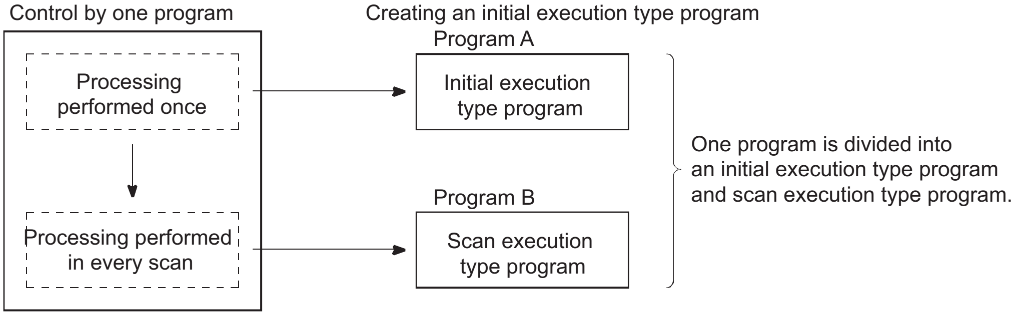
(1) Program sequence in the CPU module

The following shows the program sequence after the CPU module is powered on or its operating status is changed from STOP to RUN. Use an appropriate type of program as required.



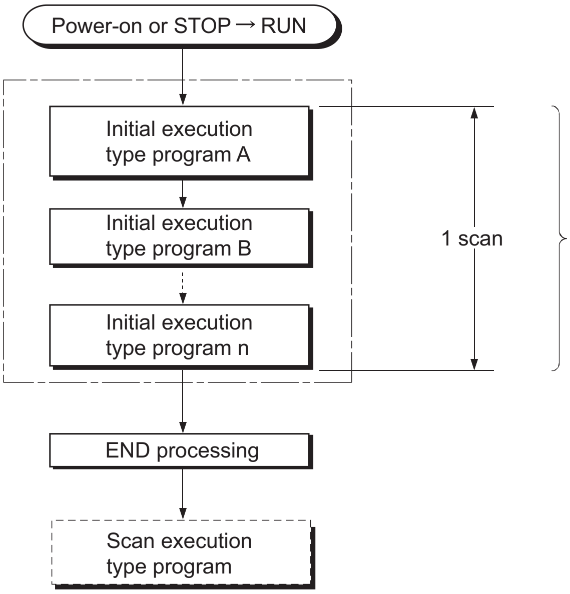
2.8.1 Initial execution type program

Initial execution type program is executed only once when the CPU module is powered on or its operating status is changed from STOP to RUN.
This type of program can be used as a program that need not be executed from the next scan and later once it is executed, like initial processing to an intelligent function module.



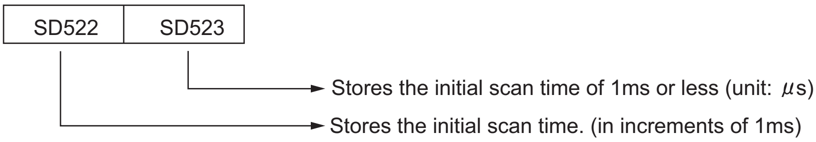
(1) Processing

After completion of all the initial execution type program execution, END processing is performed. In the next scan and later, scan execution type programs are executed.



(2) Initial scan time

Initial scan time is the execution time of initial execution type program. When multiple programs are executed, the initial scan time will be the time required for completing all the initial execution type program execution.
Since the CPU module stores the initial scan time into the special register (SD522 and SD523), the initial scan time can be checked by monitoring SD522 and SD523. (Accuracy: ±0.1ms)



Ex. If the stored values in SD522 and SD523 are 3 and 400 respectively, the initial scan time is 3.4ms. Even if the WDT instruction (instruction that resets the watchdog timer) is executed in the program, the measurement of the initial scan time continues.

Point

When an interrupt program or fixed scan execution type program is executed before completion of the initial execution type program execution, the execution time of the executed program will be added to the initial scan time.

(3) Initial execution monitoring time

Initial execution monitoring time is a timer for monitoring initial scan time.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC RAS]

The setting range is 10 to 2000ms (in increments of 10ms). No default value is set.

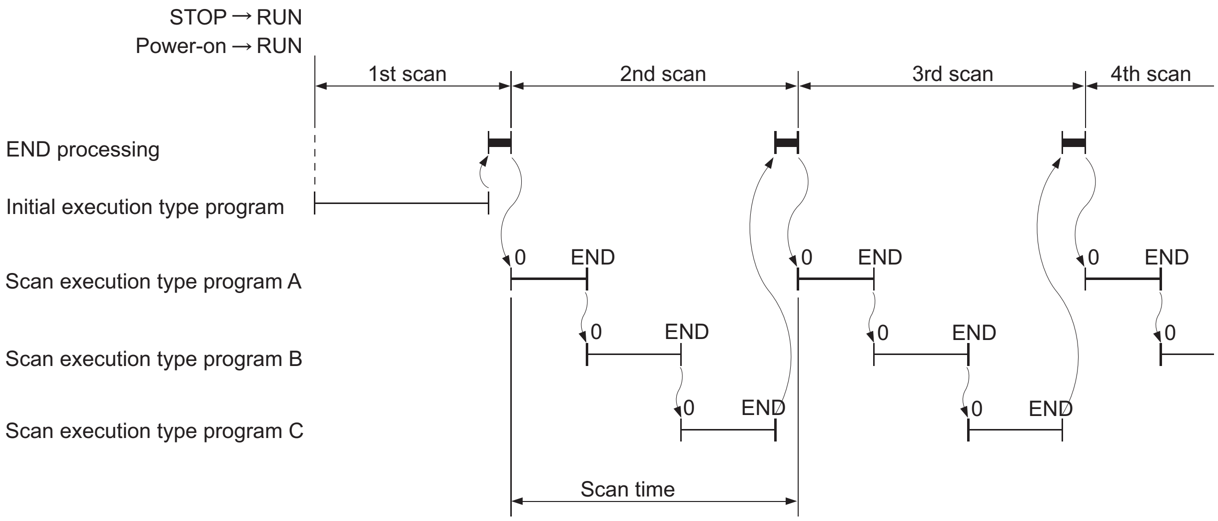
(4) Precautions on programming

Initial execution type programs do not support the instructions that require several scans (instructions with completion device).

Ex. SEND, RECV and similar instructions

2.8.2 Scan execution type program

Scan execution type program is executed once in every scan, starting in the next scan of which the initial execution type program is executed and later.



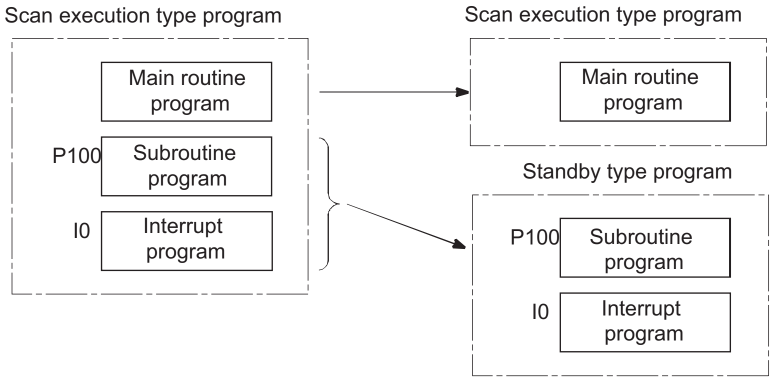
When multiple scan execution type programs are executed, the scan time will be the time required for completing all the scan execution type program execution. If an interrupt program or fixed scan execution type program is executed, execution time of the executed program will be added to the scan time.

2.8.3 Standby type program

Standby type program is executed only when its execution is requested. This type of program can be changed to any desired execution type by a program instruction. This program is mainly used for the following purposes.

- Program library

  Standby type program is used as a program library, a collection of subroutine programs and/or interrupt programs, and managed separately from a main routine program. Multiple subroutine programs and/or interrupt programs can be created and managed in a single standby type program.



- Program type change

  Standby type program is used to create and store programs available in all systems. Only required programs will be executed. For example, a program preset as a standby ("Wait") type program in the PLC Parameter dialog box can be changed to a scan execution type program and executed in the program.

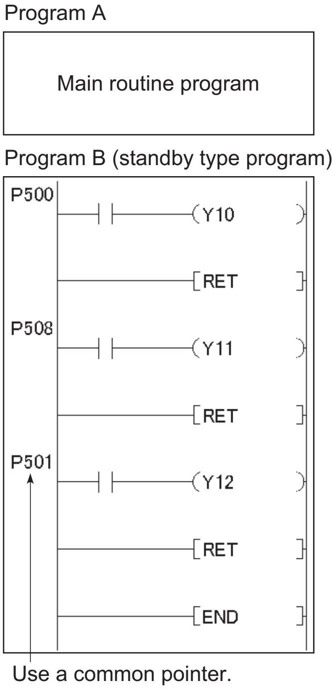
(1) Execution method

Execute standby type programs in either of the following methods.

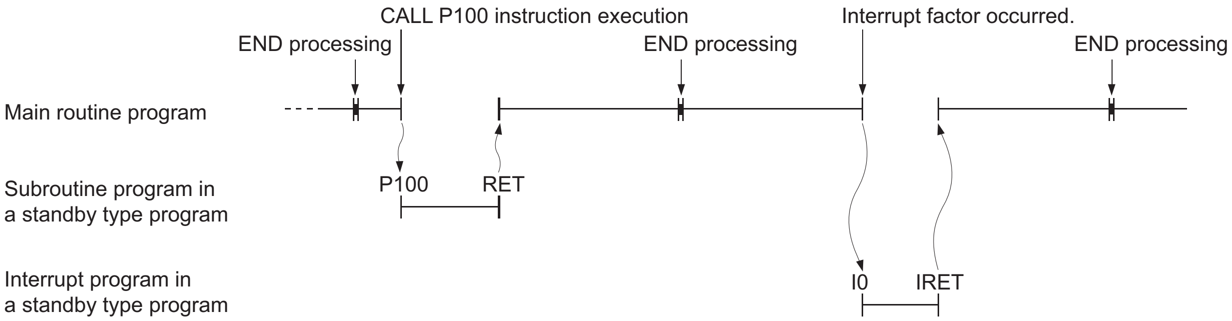
- Create subroutine and interrupt programs in a standby type program and execute the standby program upon interrupt or a call such as by a pointer.
- Change a standby type program to any other execution type using instructions.

(a) Executing upon an interrupt or a call such as by a pointer

When creating subroutine and/or interrupt programs in a single standby type program, start the program from the step 0. The FEND instruction used in creation of a subroutine or interrupt program is not required after a main routine program.



After execution of the standby type program, the CPU module re-executes the program that called a program in the standby type program.



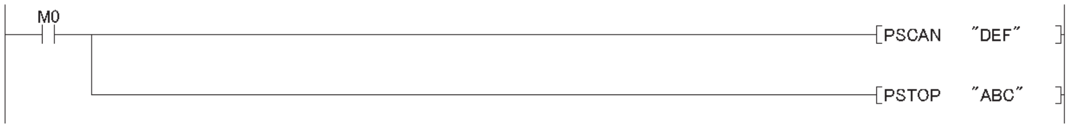
Point

Use common pointers. (☞ Page 336, Section 5.10) If local pointers are used, subroutine programs in a standby type program cannot be executed from any other program.

(b) Changing the program execution type using instructions

Use the PSCAN, PSTOP, or POFF instruction to change a program execution type. (☞ Page 72, Section 2.8.5)

In the following figure, the PSCAN instruction changes the program "DEF" to a scan execution type program and the PSTOP instruction changes the program "ABC" to a standby type program.



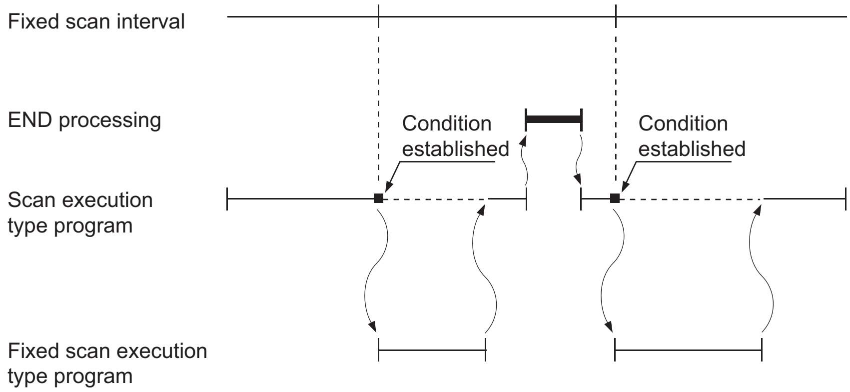
The program execution type is changed in END processing. Therefore, the execution type will not be changed in the middle of program execution. If different types are set to the same program in the same scan, the program will be changed to the type specified by the last instruction executed.

(2) Restrictions on creating a execution type program

Unavailable devices depend on the program type (subroutine program or interrupt program) or the execution type changed by an instruction.

2.8.4 Fixed scan execution type program

Fixed scan execution type program is a program executed at specified time intervals. This type of programs, unlike interrupt programs, can be interrupted in units of files without interrupt pointers or the IRET instruction. For the restrictions on programming, refer to ☞ Page 57, Section 2.7 (2). The restrictions on programming are the same as those for interrupt programs.



Point

To execute a fixed scan execution type program, execute the EI instruction in the initial execution type program or scan execution type program to enable interrupts.

(1) Processing

(a) When two or more fixed scan execution type programs exist

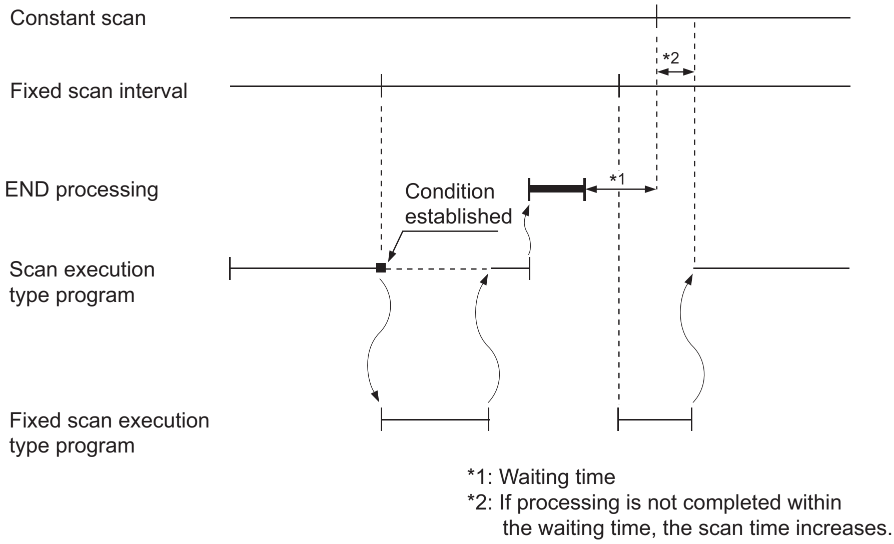
Each fixed scan execution type program is executed at specified time intervals. If two or more fixed scan execution type programs reach the specified time at the same timing, programs will be executed in ascending order of the numbers set in the Program tab of the PLC Parameter dialog box.

(b) When both fixed scan execution type program and interrupt program exist

When a fixed scan execution type program and an interrupt program (I28 to I31) reach the specified time at the same timing, the interrupt program will be given priority.

(c) When the execution condition is established during END processing

When the execution condition is established during the waiting time of the constant scan execution or the END instruction, a fixed scan execution type program is executed.



(2) Processing at program execution type change

For how to save and restore data in the index register when the program execution type is changed, refer to ☞ Page 60, Section 2.7 (4). The method is the same as that for interrupt programs.

(3) Precautions

(a) Execution interval of a fixed scan execution type program

Execution interval of a fixed scan execution type program may increase from the preset interval depending on the time set for disabling interrupts by the DI instruction (interrupt disabled time). If the interrupt disabled time by the DI instruction becomes too long, use an interrupt program by fixed scan interrupt (I28 to I31) instead of a fixed scan execution type program.

Highest common factor of fixed scan execution intervalal *1	<	Interrupt disabled time	··· Condition 1

*1 This is the highest common factor of execution interval set to multiple fixed scan execution type programs.

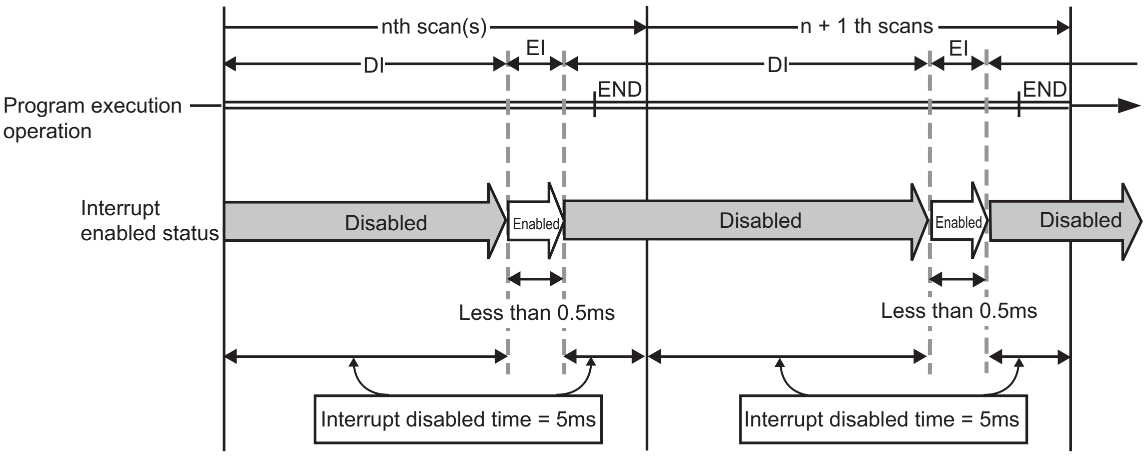
When the condition 1) is satisfied, the actual execution interval of a fixed scan execution type program may increase from the preset interval by the time shown in the expression below.

$$\frac{\text{Interrupt disabled time}}{\text{Highest common factor of fixed scan execution interval}} \times \text{Fixed scan execution interval set to the corresponding program}$$

Ex. Increase in execution time of a fixed scan execution type program.

- Fixed scan execution interval···10ms, 5ms, 1ms, 0.5ms
- Highest common factor of fixed scan execution interval···0.5ms
- Interrupt disabled time (DI)···5ms (Interrupt enabled time (EI)···less than 0.5ms)

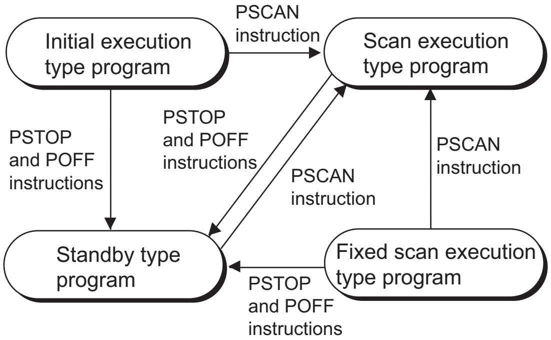
With the settings above, the condition 1) will be 0.5ms < 5ms.



The execution time of a fixed scan execution type program whose execution interval is set to 10ms increases 100ms (5 ÷ 0.5 × 10 = 100) at the most.

2.8.5 Changing the program execution type

The execution type of programs can be changed using instructions even during execution. Use the PSCAN, PSTOP, or POFF instruction to change a program execution type.

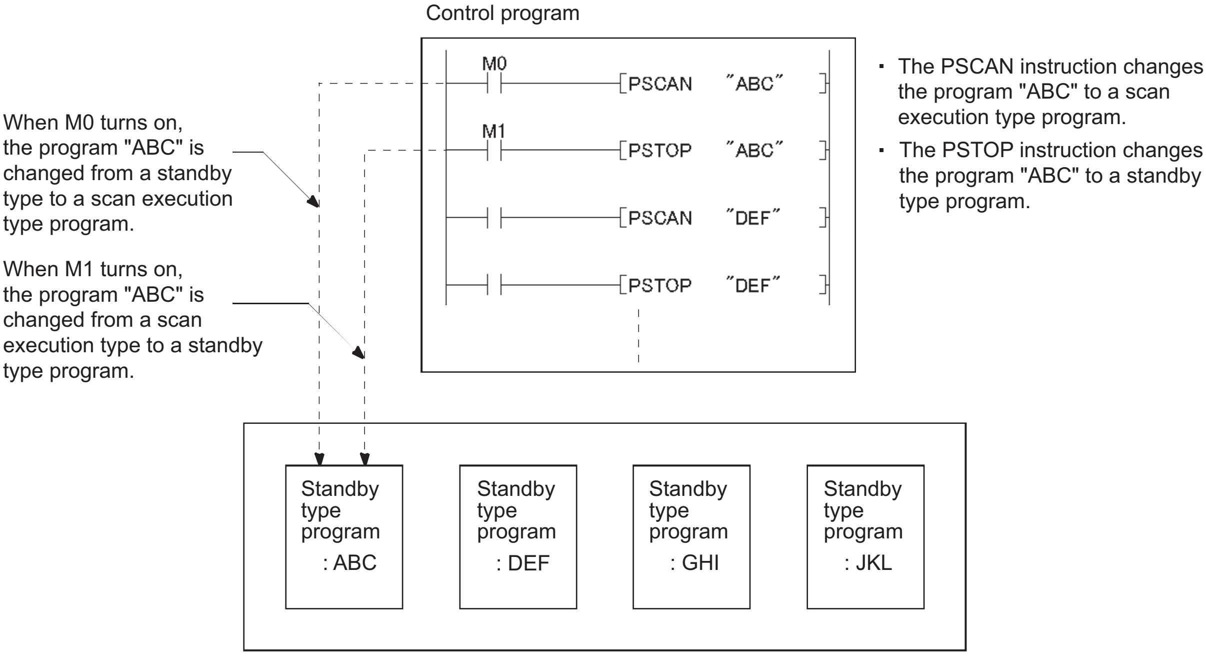


Execution type before change	Instruction		
	PSCAN	PSTOP	POFF
Scan execution type	Remains unchanged.	Changes to the standby type.	Turns off outputs in the next scan. Changes to the standby type in two scans later.
Initial execution type	Changes to the scan execution type.	Changes to the standby type.	Turns off outputs in the next scan. Changes to the standby type in two scans later.
Standby type	Changes to the scan execution type.	Remains unchanged.	No processing
Fixed scan execution type	Changes to the scan execution type.	Changes to the standby type.	Turns off outputs in the next scan. Changes to the standby type in two scans later.

Point

Once the fixed scan execution type program is changed to another execution type, the type cannot be returned to the fixed scan execution type.

Ex. In a control program, a standby type program matching the preset condition is changed to a scan execution type program in the course of program execution. (An unused scan execution type program can also be changed to a standby type program.)



2.9 Boot Operation

CPU modules can be operated according to the program and parameters stored in an SD memory card. Operating a CPU module by reading out (boot) those data are called boot operation. The files that can be booted are as follows.

○: Bootable, ×: Not bootable

File type	Boot source	Boot destination	
		Program memory	Standard ROM
Parameter*1	SD memory card	○	○
Program		○	×
Device comment		○	○
Device initial value		○	○
Symbolic information*2		○	○

*1 The intelligent function parameter is included.
*2 This is the data of label program configuration information.
📖GX Works2 Version 1 Operating Manual (Common)

Remark

The L02SCPU and L02SCPU-P do not support the boot operation.

(1) Executing a program

The programs specified in the Boot File tab are booted to the program memory when:

- the CPU module is powered off and then on or
- the CPU module is reset.

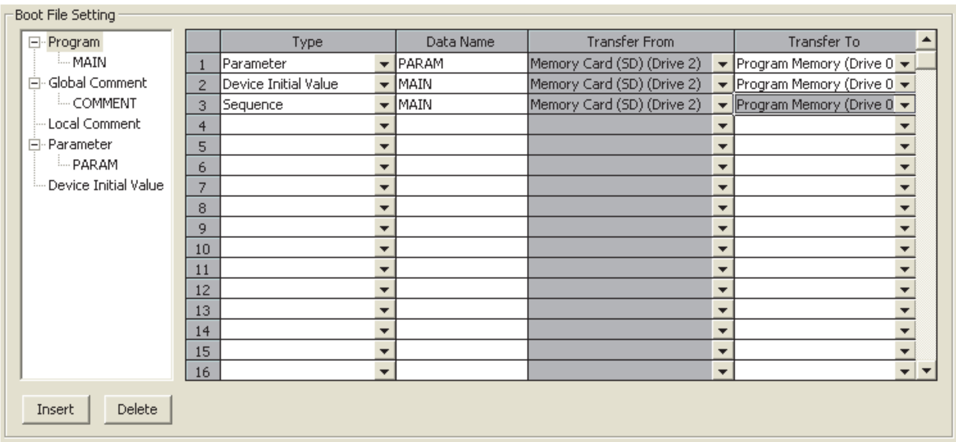
(a) Boot operation procedure

1. Create a program.

2. Configure the setting for a boot operation.

Specify the file name and transfer destination to boot.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Boot File]



3. **Insert an SD memory card.**

4. **Write the setting to the SD memory card (parameters and the files set in the Boot File tab).**

Reset the CPU module after the above operation. The boot operation is complete.

(b) Checking whether a boot is complete

The following indicates completion of boot operation.

- SM660 turns on
- There is no BOOT ERROR.
- Data in the transfer destination and in the program memory are found matched by a data verification.

[Online] ⇨ [Verify with PLC...]

(c) Operation for stopping boot operation

To stop boot operation and operate the CPU module by the parameters and program files written to the program memory, perform the following operations.

1. **Remove the memory card and write parameters without boot file setting to the program memory.**

2. **Power off and then off or rest the CPU module.**

(d) Boot operation precautions

- Set the maximum number of settable boot files in the Boot File tab of the PLC Parameter dialog box so that it is the same with the number of files storable to the program memory. However, the number of boot files is reduced by one when:
  - a heading is set
  - the parameters set in the Boot File tab of the PLC Parameter dialog box and stored in the SD memory card is booted.
- The size unit of a file stored in each memory differs between the SD memory card and the program memory. Therefore, note that files transferred from the SD memory card to the program memory differ in memory capacity between before and after the transfer.
- Set the same model name of the CPU module in the "PLC type" (model name of the CPU module) for the program written to the SD memory card (program set in the Boot File tab) and for the CPU module to boot.

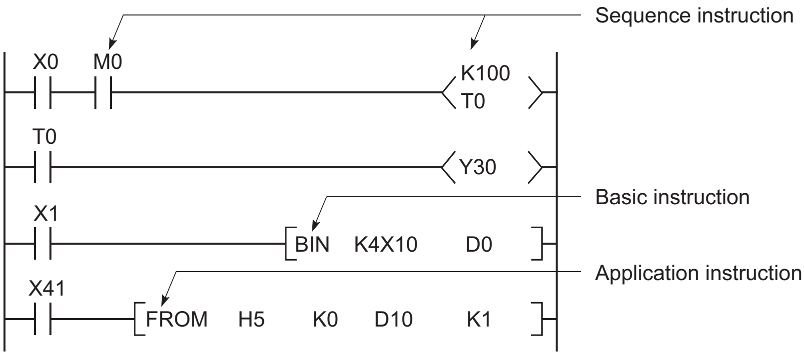
2.10 Programming Language

The following programming languages are supported in the programming tool.

- Ladder
- ST
- SFC
- Structured ladder

(1) Ladder

A graphical programming language which uses contacts and coils. For a project with a label, the inline ST function can be used in the ladder editor which allows a user to edit an ST program.



Point

Data indicating the execution status of an operation in a sequence program step is referred to as "signal flow".

(2) ST

A text language such as C language, and is preferred by computing engineers.

(3) SFC

A graphical programming language where the execution order and conditions are clearly defined for the program.

(4) Structured ladder

A graphical programming language that uses contacts and coils.

Remark

For the projects that support these programming languages, refer to the following.
Manual for the programming tool used

2.11 Communications with Intelligent Function Modules

The intelligent function module allows the CPU module to process analog quantity and high-speed pulses that cannot be processed by the I/O modules. The following is some of the intelligent modules.

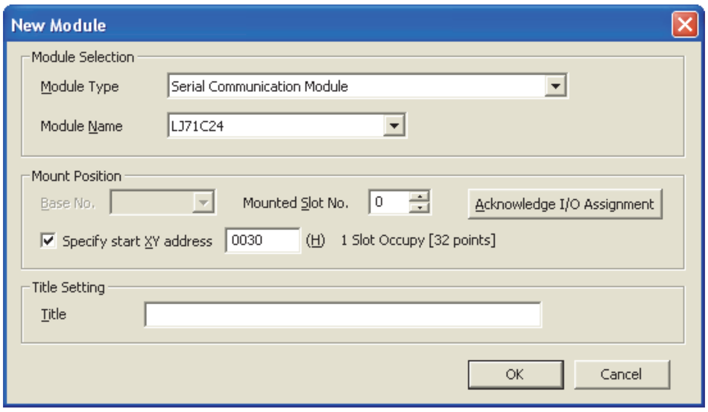
- Serial communication module
- Analog module
- Positioning module
- High-speed counter module
- Temperature control module

The intelligent function module is equipped with a memory (buffer memory) to store the data taken in from or output to external devices. The CPU module writes or reads data to or from the buffer memory of the intelligent function module.

(1) Setting method of intelligent function module parameters

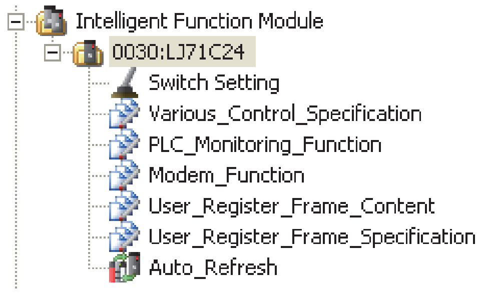
Open the "New Module" dialog box.

Project window ⇨ [Intelligent function module] ⇨ Right-click ⇨ [New Module...]



Item		Description
Module Selection	Module Type	Select a type of the CPU module.
	Module Name	Select a model name of the CPU module.
Mount Position	Mounted Slot No.	Select a slot No. where he CPU module is connected.
	Acknowledge I/O Assignment	The I/O assignment settings in the PLC Parameter dialog box can be checked.
	Specify start X/Y address	Enter the start I/O number.
Title Setting	Title	Enter a title.

Upon completion of the setting above, parameters for the intelligent function module appear in the "Project" window.



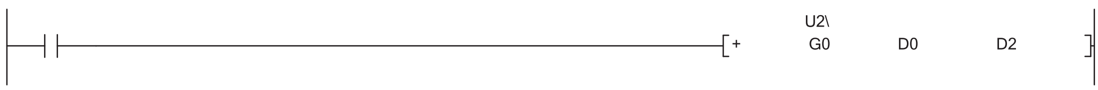
To set the intelligent module parameters, refer to the following.
📖Manual for the intelligent function module used

(2) Communications with the FROM and TO instructions

The FROM instruction stores data read from the buffer memory of the intelligent function module to the specified device.
The TO instruction writes data stored in the specified device to the buffer memory of the intelligent function module.
For details on the FROM and TO instructions, refer to the following.
📖MELSEC-Q/L Programming Manual (Common Instruction)

(3) Communications using the intelligent function module device

The intelligent function module device represents the buffer memory of the intelligent function module as one of the CPU module devices. (☞ Page 320, Section 5.5.1)
The difference from the FROM and TO instructions is that, with this device, both reading and writing data from and to the intelligent function module cab be processed with one instruction.

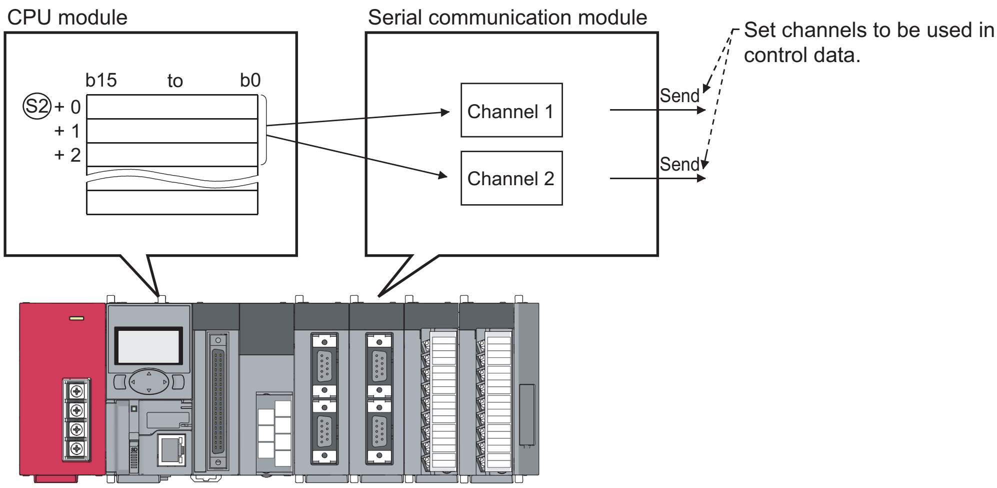


(4) Communications using the intelligent function module dedicated instruction

This instruction enables easy programming for the use of functions of the intelligent function module.

Ex. Serial communication module dedicated instruction (OUTPUT instruction)

The OUTPUT instruction allows communications with external device by nonprocedural protocol regardless of the buffer memory address of the serial communication module.



(a) Processing of the intelligent function module dedicated instruction

When using multiple intelligent function module dedicated instructions to one intelligent function module, execute the dedicated instructions one by one after the completion device turns on. This completion device turns on for one scan when an instruction is completed. If the CPU module status is changed from RUN to STOP before the completion device turns on, the completion device does not turn on until one scan after the next RUN of CPU module.

For details on the intelligent function module dedicated instructions and the completion device, refer to the following.

Manual for the intelligent function module used

Memo

PART 2 FUNCTIONS

In this part, the functions of the CPU module and display unit are described.

CHAPTER 3 CPU MODULE FUNCTIONS 82
CHAPTER 4 DISPLAY UNIT FUNCTIONS 236

CHAPTER 3 CPU MODULE FUNCTIONS

This chapter describes the functions for a CPU module.

3.1 Function List

The following table lists the functions of the LCPU.

Item		Description	Reference
Boot operation[*2]		Boots data stored in an SD memory card to the program memory or the standard ROM when the CPU module is powered off and on or reset.	Page 74, Section 2.9
Constant scan		Executes a program at a set interval regardless of its scan time.	Page 85, Section 3.2
Watchdog timer (WDT)		Monitors operational delays caused by hardware failure or program error of the CPU module.	Page 87, Section 3.3
Latch function		Holds the device data of the CPU module.	Page 88, Section 3.4
Initial device value		Registers data used in programs without programs.	Page 91, Section 3.5
Service processing setting		Specifies the service processing count or time to be executed in END processing.	Page 94, Section 3.6
Output mode at operating status change (STOP to RUN)		Sets the status of the outputs(Y) when the operation status of the CPU module is changed from STOP to RUN.	Page 101, Section 3.7
Input response time setting		Sets input response time for modules such as an input module.	Page 102, Section 3.8
Error time output mode setting		Sets whether to clear or hold the output status to each module when a stop error occurs in the CPU module.	Page 103, Section 3.9
PLC operation mode at H/W error setting		Sets a operation mode of the CPU module when a hardware error (CPU module detects SP.UNIT DOWN) occurred in an intelligent function module.	Page 104, Section 3.10
Security function	—	Protects data in the CPU module against tampering and theft by unauthorized persons.	Page 105, Section 3.11
	File password 32	Prohibits writing/reading data to/from each file in the CPU module using a programming tool. Sets write password and read password for each file stored in the CPU module.	Page 105, Section 3.11.1
	Remote Password	Prevents unauthorized access from external devices.	Page 110, Section 3.11.2
	Block password	Prevents access to program contents by setting a block password for each POU.	GX Works2 Version 1 Operating Manual (Common)
Remote RUN/STOP		Changes the CPU module status to RUN or STOP without using switches.	Page 112, Section 3.12.1
Remote PAUSE		Changes the CPU module status to PAUSE without using switches, holding the status of outputs (Y).	Page 114, Section 3.12.2
Remote RESET		Resets the CPU module when it is in the STOP status, without using switches.	Page 116, Section 3.12.3
Remote latch clear		Clears latch data of the CPU module when the module is in the STOP status, without using switches.	Page 117, Section 3.12.4
Scan time measurement		Measures the execution time of the specified steps in a program.	Page 118, Section 3.13
Program list monitor		Displays the scan time and execution status of the program being executed.	Page 121, Section 3.14
Interrupt program list monitor		Displays the number of executions of an interrupt programs.	Page 122, Section 3.15
Monitor condition setting		Monitors the CPU module under specified conditions.	Page 123, Section 3.16
Device monitor/test		Monitors and/or tests the local devices of the specified program.	Page 128, Section 3.17
External input/output forced on/off		Forcibly turns on/off the external input/output of the CPU module.	Page 131, Section 3.18
Executional conditioned device test		Changes a device value for the specified step of a program.	Page 135, Section 3.19

Item		Description	Reference
Sampling trace		Continuously collects data of the specified device at a specified timing.	Page 143, Section 3.20
Realtime monitor function*1*2		Monitors the data in the specified device of the CPU module at a specified interval or at a desired timing in real time.	Page 150, Section 3.21
Online change (ladder mode)		Writes programs while the CPU module is in the RUN status.	Page 151, Section 3.22.1
Online change (files)		Collectively writes files while the CPU module is in the RUN status.	Page 152, Section 3.22.2
Debug from multiple programming tools		Enables simultaneous debugging from multiple programming tools.	Page 158, Section 3.23
Self-diagnostic function		Self-diagnoses the CPU module to see whether an error exists or not.	Page 161, Section 3.24
Error clear		Clears continuation errors by error type. User can specify continuation errors to clear.	Page 166, Section 3.25
LED control function		Sets the timing to turn off and indication of the LEDs located on the front of the CPU module.	Page 169, Section 3.26
Module error collection function		Lists descriptions of errors occurred in the intelligent function module.	Page 171, Section 3.27
Latch data backup to standard ROM		Backs up latch data such as device data and error history without using a battery.	Page 175, Section 3.28
Writing/reading device data to/from standard ROM		Writes/reads device data to/from the standard ROM using an instruction.	Page 180, Section 3.29
Module model name read		Reads the model name of a module connected.	Page 181, Section 3.30
CPU module change function with SD memory card*2		Saves the data in a CPU module to an SD memory card when changing the CPU module.	Page 182, Section 3.31
Clock function		Controls the clock data of the CPU module.	Page 196, Section 3.32
Battery-life prolonging function		Prolongs a battery life by limiting the data to be latched to clock data only.	Page 198, Section 3.33
Memory check function*1		Checks that data in the memories of the CPU module are not changed due to excessive electric noise.	Page 199, Section 3.34
Program cache memory auto recovery function*1		Restores the error location automatically by using data in the program memory, which are stored in the flash ROM, when the memory check function detects an error in the program cache memory.	Page 200, Section 3.35
Project data batch save/load function*1*2		Saves data in the CPU module to an SD memory card, and also reads the data saved in an SD memory card to the CPU module.	Page 202, Section 3.36
Predefined protocol function*1	Communications via Ethernet*2	Sends and receives packets predefined by using GX Works2, enabling easy communications with external devices (such as measuring instruments and bar code readers).	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
	Communications via RS-232, RS-422/485*2		Page 215, Section 3.37
Serial communication function*1*2		Communicates data with external devices, such as personal computers and HMIs, connected with an RS-232 or RS-422/485 cable using the MC protocol.	Page 227, Section 3.38
Writing/reading data to/from refresh devices with the specified station number*1*2		Writes/reads data by specifying the station number of the target station, without considering the assignment of refresh devices.	MELSEC-Q/L Programming Manual (Common Instruction)
Data logging function*2		Collects data from the specified device of a CPU module at a specified timing. The data logging file can be transferred from a CPU module to the FTP server using the data logging file transfer function.	QnUDVCPU/LCPU User's Manual (Data Logging Function)
Built-in I/O function		Performs the following dedicated functions: general-purpose input function, general-purpose output function, interrupt input function, pulse catch function, positioning function, and high-speed counter function.	MELSEC-L CPU Module User's Manual (Built-In I/O Function)

Item		Description	Reference
Built-in Ethernet function*2	—	Enables MC protocol communications and the following functions by using built-in Ethernet ports.	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
	Socket communication function*2	Communicates data (using TCP/UDP) with external devices connected on the Ethernet network. The function is executed by dedicated instructions.	
	File transfer function (FTP server)	Works as a server of the FTP (File Transfer Protocol), which is used to transfer files between the CPU module and external devices. External devices with the FTP client function can directly access to the files in the CPU module.	
	File transfer function (FTP client)*1	Works as a client of the FTP (File Transfer Protocol), which is used to transfer files between the CPU module and external devices. The CPU module can transfer files to the FTP server on a LAN by using the file transfer function instructions.	
	Simple PLC communication function*1*2	Allows data communications between specified devices at the specified timing just by doing simple settings from a programming tool.	
	IP address change function	Changes the IP address of the built-in Ethernet port by storing the value in the special relay or special register, instead of setting the value in the built-in Ethernet port setting parameter.	
	IP packet transfer function*1*2	Communicates with the following devices that correspond to IP address specified via a CC-Link IE Controller Network module or CC-Link IE Field Network module, using a protocol such as the FTP or HTTP via a built-in Ethernet port from an Ethernet device such as a personal computer. • External devices on the CC-Link IE Field Network • External devices on the Ethernet network, which are connected through the built-in Ethernet ports	
	E-mail send/receive function*1	Sends/receives e-mail to/from an electronic device, such as a mobile phone, smart phone, and personal computer. Device data can be read/written by describing an MC protocol command in the body of the e-mail sent to the CPU module.	
	SLMP frame send instruction*1	Sends MC protocol messages (QnA-compatible 3E frame) from the CPU module to external devices connected on the Ethernet network.	
Built-in CC-Link function*2		Allows data communications over CC-Link.	MELSEC-L CC-Link System Master/Local Module User's Manual
iQ Sensor Solution function*1*2		A set of functions that can be used in iQ Sensor Solution	Page 234, Section 3.39 iQ Sensor Solution Reference Manual
CC-Link IE Field Network Basic function*1*3		A set of functions that can be used in CC-Link IE Field Network Basic	CC-Link IE Field Network Basic Reference Manual

*1 Availability depends on the version of the CPU module. (Page 376, Appendix 2)
*2 Some models do not support those functions. For the availability, refer to the reference sections of each function.
*3 Only the Built-in Ethernet port LCPU supports this function.

Point

For details on the special relay (SM) and special register (SD) used for each function, refer to the following.
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

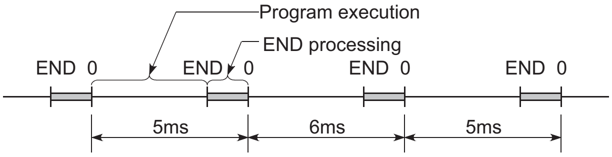
3.2 Constant Scan

Scan time of the CPU module is not constant because the processing time varies depending on the execution status of instructions used in a program. By using this function, scan time can be maintained constant.

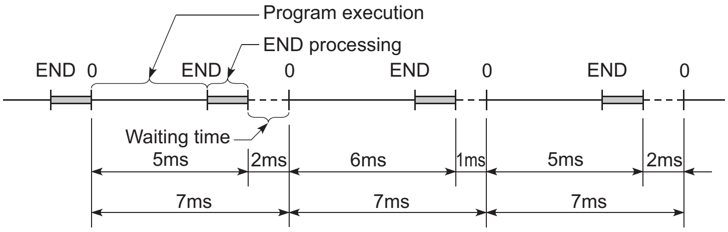
(1) Application

I/O refresh is performed before program execution. By using this function, the constant I/O refresh intervals can be maintained even the time required for program execution changes.

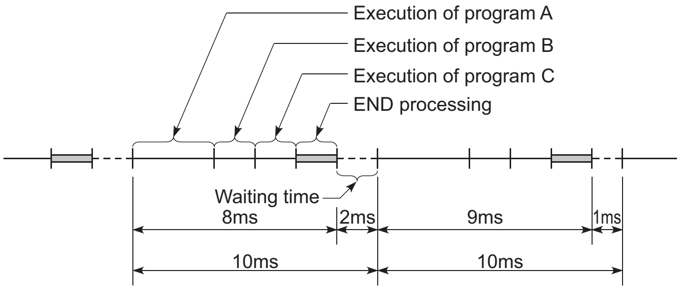
- Scan time without constant scan setting



- Scan time with constant scan setting (7ms)



- Scan time for multiple programs with constant scan setting (10ms)



(2) Constant scan time setting

The setting range is from 0.5 to 2000ms (in increments of 0.5ms).
When not using the constant scan function, leave the setting box blank.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC RAS]

The constant scan accuracy is 0.01ms. However, constant scan time may increase when an interrupt program or a fixed scan execution type program is being executed.

3

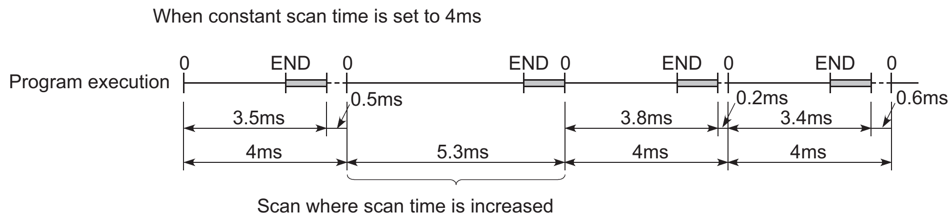
3.2 Constant Scan

(a) Condition

Set a value that satisfies the following relational expression.

WDT setting time > Constant scan setting time > Maximum scan time of the program

If the program scan time is longer than the constant scan setting time, the CPU module detects "PRG. TIME OVER" (error code: 5010). In this case, the constant scan setting is ignored and the program scan time is applied.



If the program scan time is longer than the WDT setting time, the CPU module detects "WDT ERROR". In this case, the program execution is stopped.

(3) Waiting time from when END processing is executed until next scan starts

- When an interrupt factor occurs during waiting time, either of an interrupt program or fixed scan execution type program is executed.
- When a service processing parameter is set, a communication service processing with peripherals and intelligent function modules is performed.

3.3 Watchdog Timer (WDT)

WDT is an internal timer of the CPU module that detects CPU module hardware failures and program errors.

(1) Setting

Open the tab for watchdog timer setting. The setting range is 10 to 2000ms (in increments of 10ms). 200ms is set by default.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC RAS]

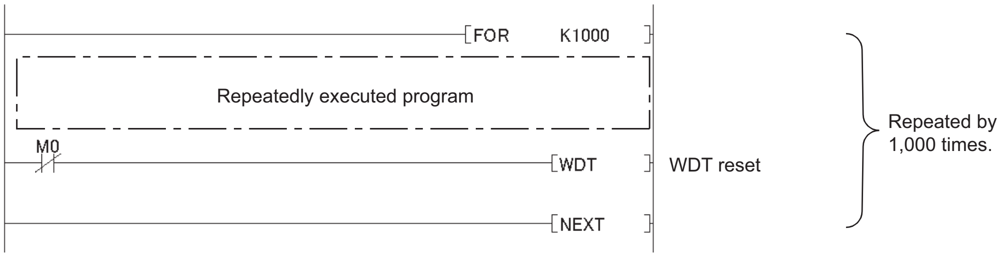
A precision error is observed within the range of 0 to 10ms.

(2) Reset

WDT is reset in END processing. The watchdog timer times up when the scan time is extended due to a factor such as an interrupt, and the END/FEND instruction were not executed within the setting value of watchdog timer. When the watchdog timer times up, "WDT ERROR" is detected and the following status occurs.

- The CPU module turns off all outputs.
- The RUN LED turns off and the ERR.LED starts flashing.
- SM1 turns on and the error codes 5000 and 5001 are stored in SD0.

The watchdog timer can be reset by executing the WDT instruction in a program. To avoid the time up of watchdog timer while a program is repeatedly executed between the FOR and NEXT instructions, reset the watchdog timer by the WDT instruction.



Note that the scan time value is not reset even after the watchdog timer is reset. The scan time is measured up to the END instruction.

3.4 Latch Function

This function holds data in each device of the CPU module when:

- the CPU module is powered off and then on,
- the CPU module is reset, or
- a power failure occurred exceeding the allowable momentary power failure time.

Data in each device of the CPU module are cleared and set back to its default (bit device: off, word device: 0) when this function is not used. By using this function, the operation can be continued with the previous data even after powering off the CPU module while the data are managed by a sequential control. Program operation does not change regardless of the latch

(1) Devices that can be latched

Set the device to latch among the following. (By default, only the latch relay is latched.)

- Latch relay (L)
- Link relay (B)
- Annunciator (F)
- Edge relay (V)
- Timer (T)
- Retentive timer (ST)
- Counter (C)
- Data register (D)
- Link register (W)

The following devices also can be set when a file register is set to be used in the PLC file.

- File register (ZR)
- Extended data register (D)
- Extended link register (W)

Point

When the battery life-prolonging function is set, the latch function is disabled. (☞ Page 198, Section 3.33)

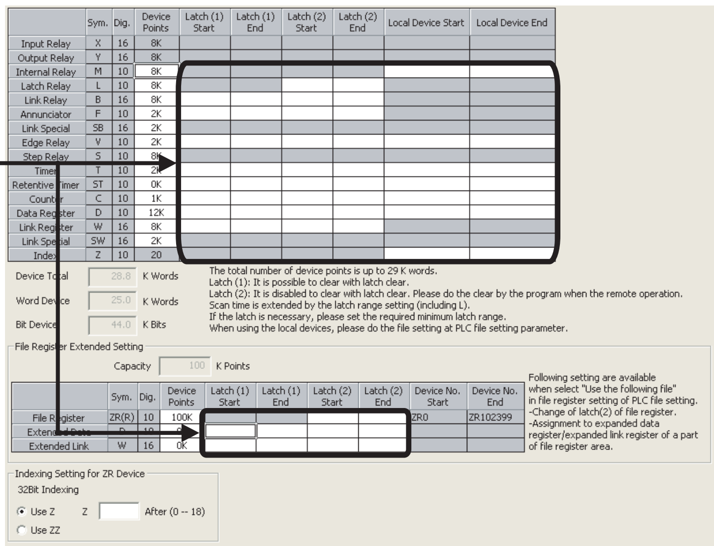
(2) Latch range setting

Set the following.

Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Device]

- Latch clear operation enable range (Latch (1) Start/End): Data in this range can be cleared by a latch clear operation.
- Latch clear operation disable range (Latch (2) Start/End): Data in this range cannot be cleared by a latch clear operation.

Set latch ranges by the start and end device numbers.



(3) Effect on the scan time

Data latch processing is performed during END processing therefore the scan time increases. Consider the effect on the scan time when latching devices. (☞ Page 378, Appendix 3.1)

Point

To reduce the scan time increase due to latch*1, minimize the number of latch points (latch (1) setting, latch (2) setting, and latch relay) as much as possible by performing the following.

- Move data to be latched to the file register.
- Store device data that is less frequently updated in the standard ROM with the SP.DEVST instruction. (The device data stored in the standard ROM can be read with the S(P).DEVLD instruction. (☞ Page 180, Section 3.29)

*1 For file registers (including an extended data register (D) and an extended link register (W)), the scan time is not increased due to latch.

(4) Latch data clear methods

(a) Data in the latch clear operation enable range (Latch (1) Start/End)

Perform either of the following.

• Remote latch clear

Perform the operation using a programming tool. (☞ Page 117, Section 3.12.4)

• Latch clear by using the special relay and special register areas

1. **Change the operating status of the CPU module to STOP.**
2. **Set "5A01$_H$" in SD339.**
3. **Turn on SM339.**

Point

Data can be cleared only when the CPU module is in STOP status. If the operation is performed when the CPU module is not in STOP status, no processing is performed (data cannot be cleared).

Remark

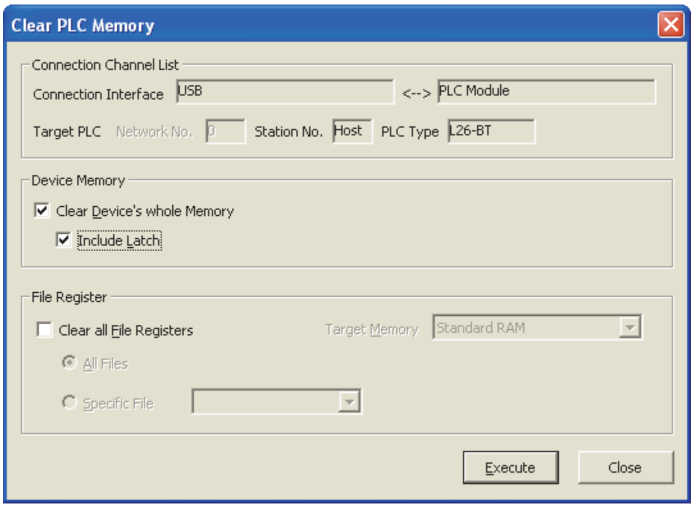
Before clearing data by using the special relay and special register areas, check the version of the CPU module used. (☞ Page 376, Appendix 2)

(b) Data in the latch clear operation disable range (Latch (2) Start/End) and in the file register

Perform any of the following.

- Perform the data clear operation using the display unit. (☞ Page 249, Section 4.2.2)
- Reset data by using the RST instruction.
- Transfer K0 by using the MOV or FMOV instruction. (📖 MELSEC-Q/L Programming Manual (Common Instruction))
- Set parameters ("Clear Device's whole Memory" or "Clear all File Registers").

[Online] ⇨ [PLC Memory Operation] ⇨ [Clear PLC Memory] ⇨ "Clear Device's whole Memory"/"Clear all File Registers"

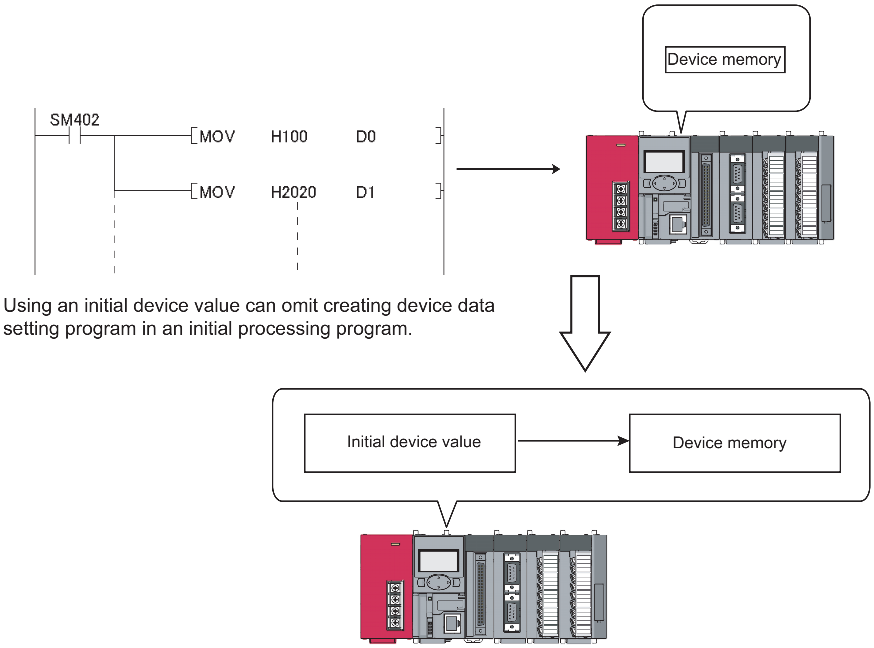


(5) Precautions

- The latch setting does not take effect if the device is specified as a local device or the initial device value is specified for the device.
- A battery is required to retain the device data that are set in the latch range. If the battery connecter is removed during power-off, the device data within the latch range are discarded and undefined values may replace. (A battery is required also in the boot operation to latch devices.)

3.5 Initial Device Value

This function registers data used in a program to the device or the buffer memory of the intelligent function module without a program. By using this function, creating a device data setting program can be omitted in the initial processing program.



(1) Timing when initial device values are written to the specified device

The CPU module writes data in the specified initial device value file to the specified device or the buffer memory of the intelligent function module when:

- the CPU module is powered off and then on,
- the CPU module is reset, or
- the CPU module status is switched from STOP to RUN.

(2) Devices that can be used*1

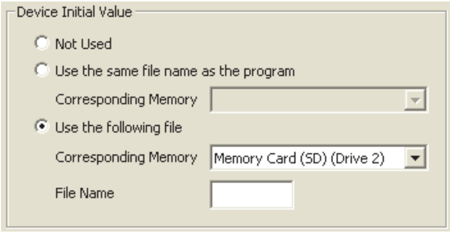
The following shows devices that can be used for initial device value.

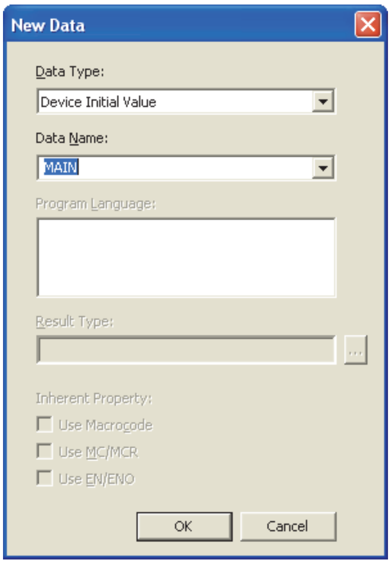
- Current timer value (T)
- Current retentive timer value (ST)
- Current counter value (C)
- Data register (D)
- Link register (W)
- Link special register (SW)
- Special register (SD)
- Intelligent function module device (U□\G□)*2
- File register (R, ZR)
- Extended data register (D)
- Extended link register (W)

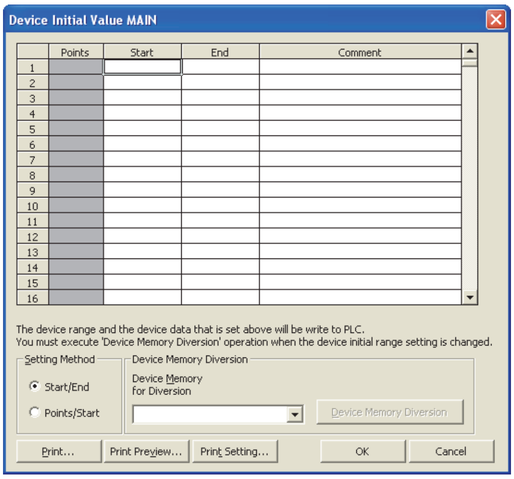
*1 The initial device value setting has a priority over the latch setting.

*2 To use an intelligent function module, select "Module Synchronization" because the initial device value may not be correctly set to the target module. (☞ Page 350, Appendix 1)

(3) Initial device value setting





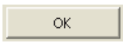


1. Specify a memory and name to store initial device values.

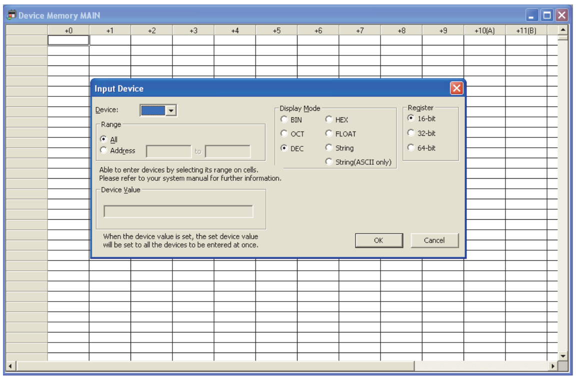
Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC File]

2. Specify a name to store initial device values.

Project Window ⇨ [Device Initial Value] ⇨ Right-click ⇨ [Add New Data...]

3. Clicking the 

button on the dialog box above opens the dialog box shown on the left. Enter the range for initial device values.



4. **Configure the device memory.**

Project Window ⇨ [Device Memory] ⇨ [MAIN] ⇨ Right-click ⇨ [Input Device...]

5. **Store the initial device value file in the program memory of the CPU module, standard ROM, or an SD memory card.**

[Online] ⇨ [Write to PLC]

Point

Whenever the initial device value range is changed, execute "Device Memory Diversion" on the "Device Initial Value MAIN" dialog box. For details on creating initial device value data, refer to the following.
Manual for the programming tool used

(4) Precautions

Initial device values also take effect when the CPU module status is switched from STOP to RUN. Therefore, they cannot be used for the data that are to be changed in a program after being set at CPU module power-on from off. Use an instruction such as the MOV instruction in the main routine program so that the initial device values are set to the specified devices. For the intelligent function module, use the TO instruction to write data to the buffer memory.

3.6 Service Processing Setting

This function allows to set the time and the number of times of service processing performed at END processing by parameters.
Processing for requests from peripherals to the CPU module are performed with this function. The processing speed for the communication response to the requests varies depending on the scan time and communication load. Set the parameter of the service processing time as follows to achieve an optimal service processing environment for the system used.

- Setting a longer time can improve the processing speed for the communication response.
- Setting a shorter time can avoid a prolonged scan time caused by the service processing.

Communications from multiple peripherals to the CPU module may slow down the processing speed for the communication response. Adjust the settings for the system by specifying a longer service processing time or modifying the parameter settings of the peripherals to set longer timeout times taking the processing speed for the communication response and increases in scan times into consideration.

Point

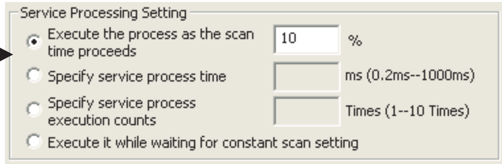
- The service processing setting can be configured for the following communications:
  - Communication via an intelligent function module (not including refresh with a network module)
  - Communication via a USB cable, RS-232 adapter, RS-422/485 adapter, and the built-in Ethernet ports (communication with a programming tool or GOT, or using the built-in Ethernet function)
- Using the COM instruction enables the service processing during program execution while the processing is normally performed during END processing. Use this function when the scan time is long.

(1) Setting method

Configure the setting of the service processing.

Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Select any one option button. Deselected parameters remain unavailable.



"Execute the process as the scan time proceeds" is selected by default.

Item	Description	Setting range	Remarks
Execute the process as the scan time proceeds*6	Set the percentage of service processing for one scan.	• Range: 1 to 99% • Unit: 1%	Default when selected = 10%
Specify service process time *1*2*3*6	Set the time of service processing for one scan.	• Range: 0.2ms to 1000ms • Unit: 0.1ms	Default when selected = 0.2ms
Specify service process execution counts*4*5	Set the number of service processing for one scan.	• Range: 1 to 10 times • Unit: 1 time	Default when selected = 1 time
Execute it while waiting for constant scan setting.*2*6	Set whether to perform service processing during waiting time for constant scan setting.	—	Even when the waiting time is 0.2ms or less, the service processing time of 0.2ms is added to the scan time.

*1 There is a margin of error between -20μs to +30μs against actual processing time.

*2 The response performance of service processing significantly reduces in the following cases. Set service processing time considering the time-out time of the peripheral.
· Service processing time is set much shorter than the scan time.
· Setting "Execute it while waiting for constant scan setting" results in increase in the scan time and decrease in the service processing time.

*3 For the following functions, scan time will be increased longer than the specified time during service processing even if the service processing time specification is set.
· Online change
· Change T/C setting
· Local device monitor
· Program memory backup
· Writing to/reading from the file register or the buffer memory of an intelligent function module (The scan time is increased when the write or read size is large.)
· Diagnostic function of a network module (CC IE Field diagnostics, CC-Link/ CC-Link/LT diagnostics)
· Monitor function (module access device, link direct device)

*4 Note that the scan time is increased much longer if the CPU module simultaneously receives multiple requests while the service processing count specification is set many.

*5 If a data communication is performed over Ethernet when "Specify service process execution counts" is selected in parameter, the scan time will increase by the service processing time (approx. 500ms). To keep the scan time 500ms or less, select an item other than "Specify service process execution counts" in parameter. (For example, select "Specify service process time." and set a time value to 500ms or less.)

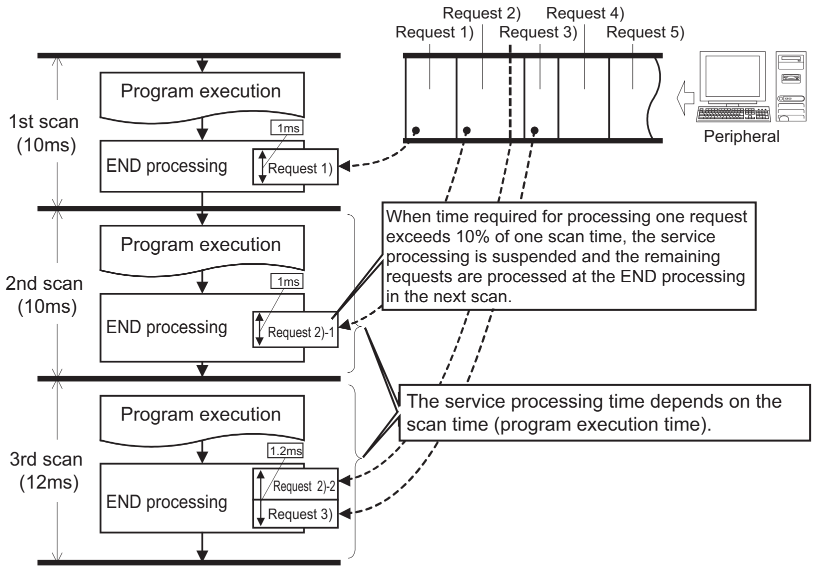
*6 If communications in the MC protocol are performed with an item other than "Specify service process execution counts" selected, data inconsistency may occur. To prevent data inconsistency, select "Specify service process execution counts".

(2) Operation for service processing setting

Operations for each service processing setting is described below.

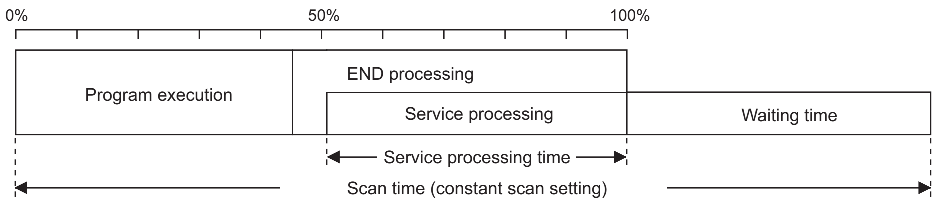
(a) Execute the process as the scan time proceeds

Operation when 10% is set is as shown below.



Point

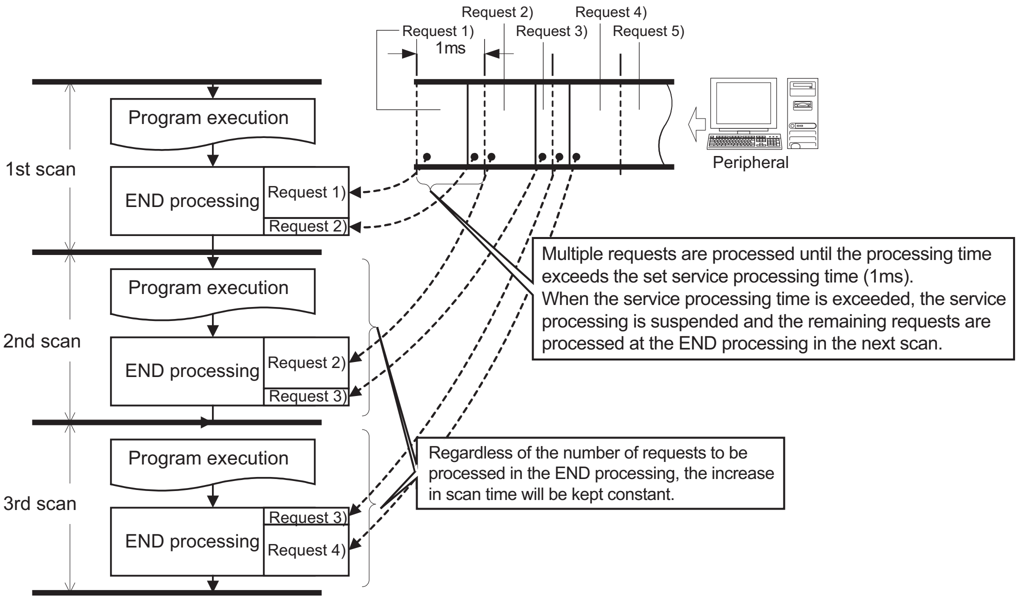
- When the constant scan is set, (☞ Page 99, Section 3.6 (2) (d)), the calculation of the service processing time is a calculation of the percentage of the time excluding the waiting time of the constant scan from the scan time, not a calculation of the percentage of the scan time.
  Example: Operation when 50% is set



- If no request data for service processing exists, END processing speeds up by the amount of request processing time. (The CPU module does not wait for requests.)

(b) Specify service process time

Operation when 1ms is set is as shown below.



Point

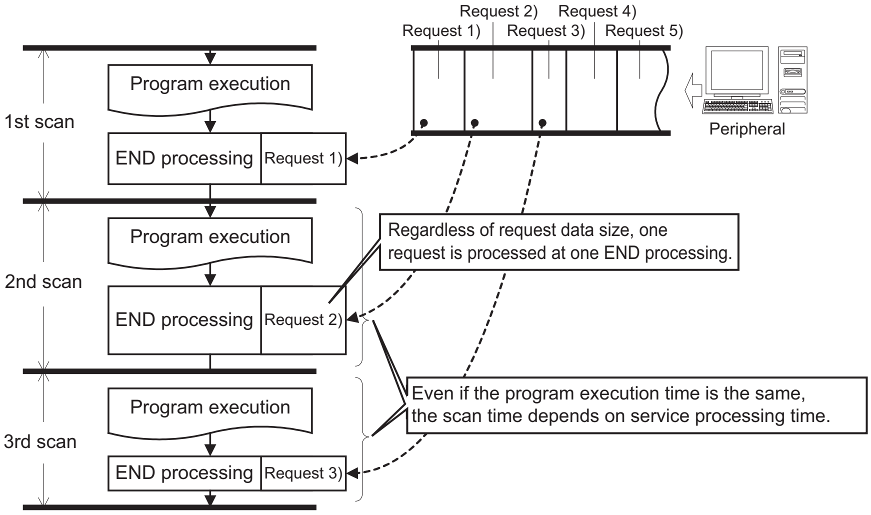
If no request data for service processing exists, END processing speeds up by the amount of request processing time. (The CPU module does not wait for requests.)

3

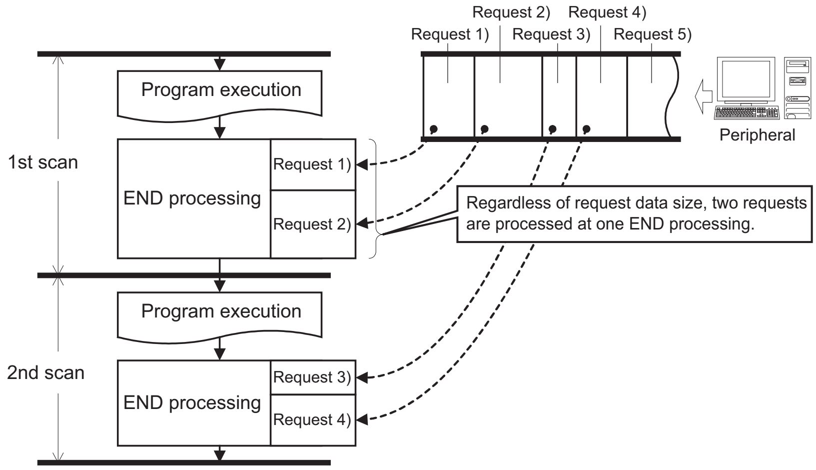
3.6 Service Processing Setting

(c) Specify service process execution counts

Operation when 1 time is set is as shown below.



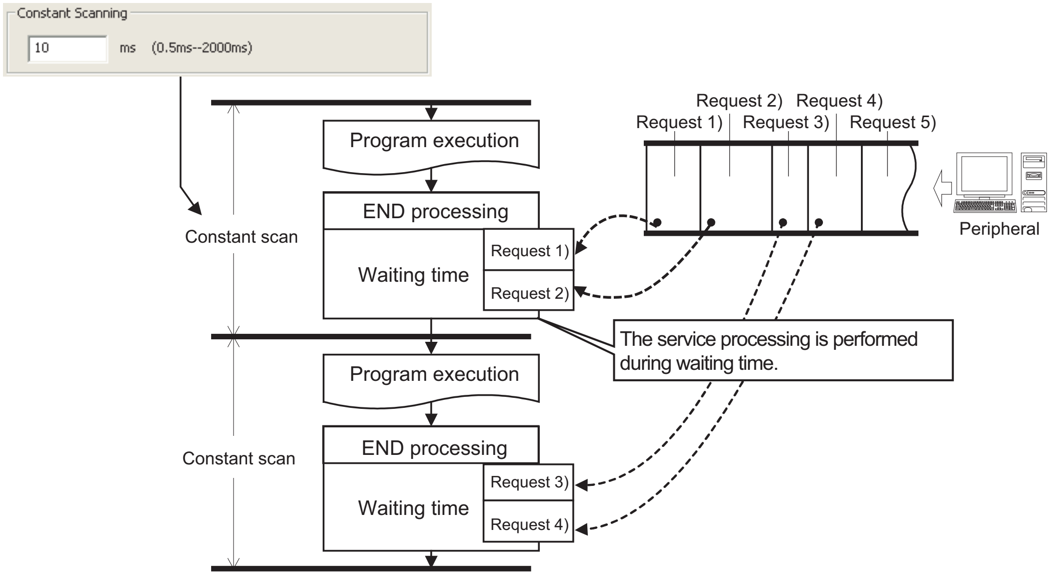
Operation when 2 times is set is as shown below.



Point

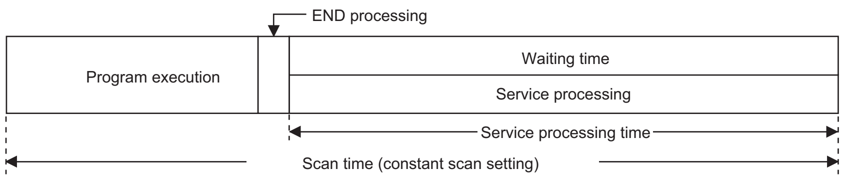
- When several devices are connected to one CPU module, each device requests service processing. When the CPU module simultaneously receives requests from several devices, a single END processing can simultaneously accept several requests if the service processing count is set to the number of connected devices. This improves response performance. Note that the scan time increases by the amount of service processing time.
- If no request data for service processing exists, END processing speeds up by the amount of request processing time. (The CPU module does not wait for requests.)

(d) Execute it (service processing) while waiting for constant scan setting.

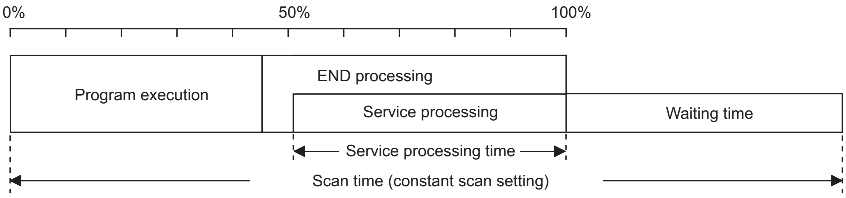


Point

- When setting the constant scan, selecting "Execute it while waiting for constant scan setting." can perform the service processing efficiently.
  - When "Execute it while waiting for constant scan setting." is selected



  - When "Execute the process as the scan time proceeds." is selected (50% is set.)



- Even when there is no waiting time, the service processing (0.2ms) is performed. Therefore, when the waiting time is less than 0.2ms, the constant scan time may be exceeded.

(e) Precautions

The following describes precautions for the service processing setting.

- In the service processing, the processing for request data from each receiving port (each intelligent function module, USB, RS-232, and built-in Ethernet) is performed one by one. If several requests are received one after another from the same receiving port, the service processing may not be performed in the same scan even if the service processing time is left. For this reason, in communications with multiple peripherals via the built-in Ethernet port, response times to each peripheral may not be improved, regardless of the service processing time setting. In this case, divide connections of peripherals into the Ethernet module and the built-in Ethernet of the CPU module to improve the response time.
- Since the processing for a request including accesses to files takes time, responding to the request necessarily takes time. The processing for request data is performed one by one in the service processing, and therefore responses to succeeding request data delays when the CPU module receives a request including accesses to files. For this reason, set a longer timeout time for peripherals when using a system that regularly send these requests.

3.7 Output Mode at Operating Status Change (STOP to RUN)

When the operating status is changed from RUN to STOP, the CPU module internally stores the outputs (Y) in the RUN status and then turns off all the outputs (Y). Therefore, status of the outputs(Y) can be selected for setting when the CPU module is set back to the RUN status.

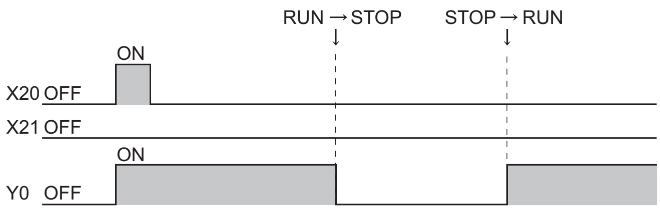
- Output the output (Y) status prior to STOP. ("Previous state")
- Clear the output (Y) status. ("Recalculate (output is 1 scan later)")

(1) Operation when the operating status is changed from STOP to RUN

Operations are explained using the holding circuit below.

(a) When outputting the output (Y) status prior to STOP

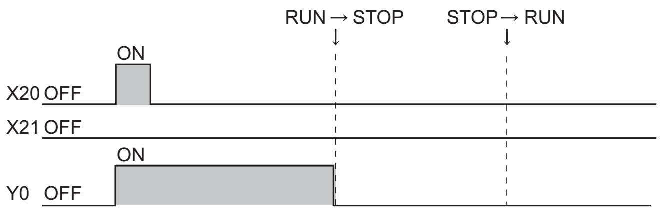
The CPU module outputs the output (Y) status immediately before the CPU module is changed to the STOP status, and then performs program operations.



When the output (Y) status is forcibly turned on while the CPU module is in the STOP status, the CPU module outputs the previous output (Y) status. The on status is not held if the output (Y) status before the CPU module is changed to STOP status is off.

(b) When clearing output (Y) status

All outputs are turned off. The CPU module outputs the output (Y) status after program operations are completed.



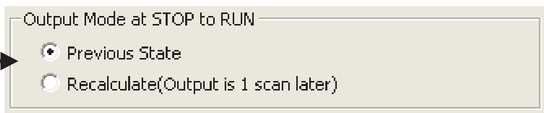
When the output (Y) status is forcibly turned on while the CPU module is in the STOP status, the CPU module holds the on status.

(2) Setting procedure

Configure the setting under "Output Mode at STOP to RUN"

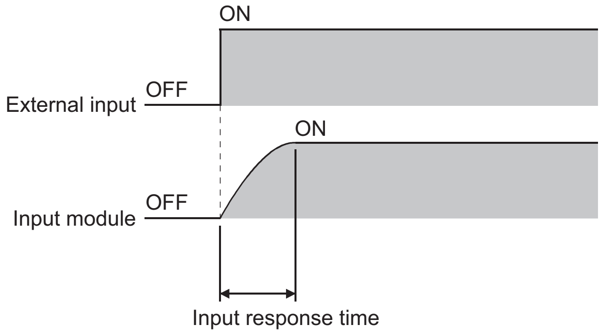
Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Select either of these.



3.8 Input Response Time Setting

This function changes the input response time for each input module. Input modules obtain external inputs within the set response time.
For the input response time setting by the built-in I/O function, refer to the following.
MELSEC-L CPU Module User's Manual (Built-In I/O Function)



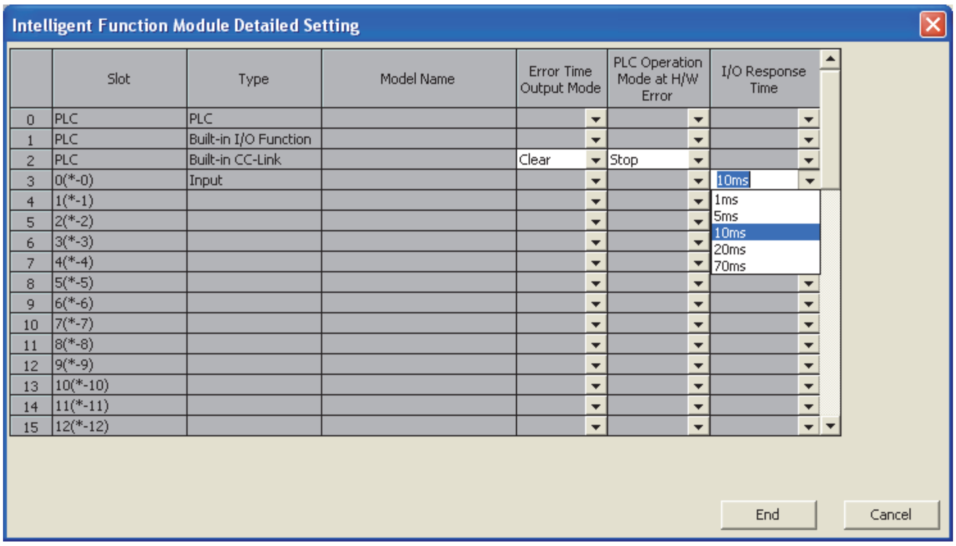
(1) Setting procedure

1. Select "Input" under "Type" in the I/O Assignment tab, and click the Detailed Setting **button.**

Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [I/O Assignment]

2. Select any of the following values for "I/O Response Time". (Default: 10ms)

1ms, 5ms, 10ms, 20ms, 70ms



Point

The shorter the input response time is, the more the CPU module is susceptible to noise. Consider the operating environment when setting input response time values.

3.9 Error Time Output Mode Setting

This function determines the output mode (clear or hold) from the CPU module to output modules and intelligent function modules when a stop error occurs in the CPU module.
For the error time output mode setting by the built-in I/O function, refer to the following.
MELSEC-L CPU Module User's Manual (Built-In I/O Function)

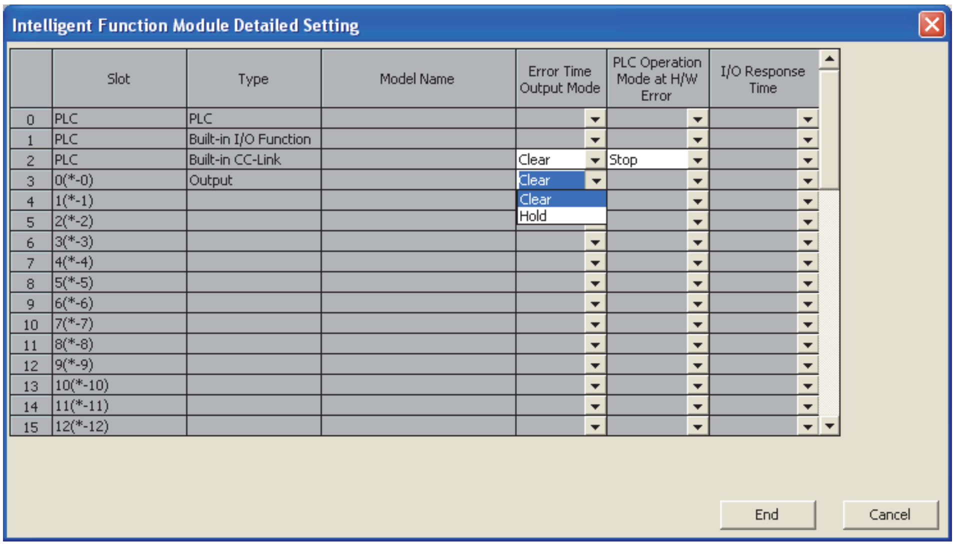
(1) Setting procedure

1. Select a target module under "Type" in the I/O Assignment tab, and click the Detailed Setting button.

Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [I/O Assignment]

2. Select either of the following items for "I/O Response Time".

"Clear" or "Hold"



3.10 PLC Operation Mode at H/W Error Setting

This function determines an operation mode of the CPU module when a hardware error (CPU module detects SP.UNIT DOWN) occurred in an intelligent function module.

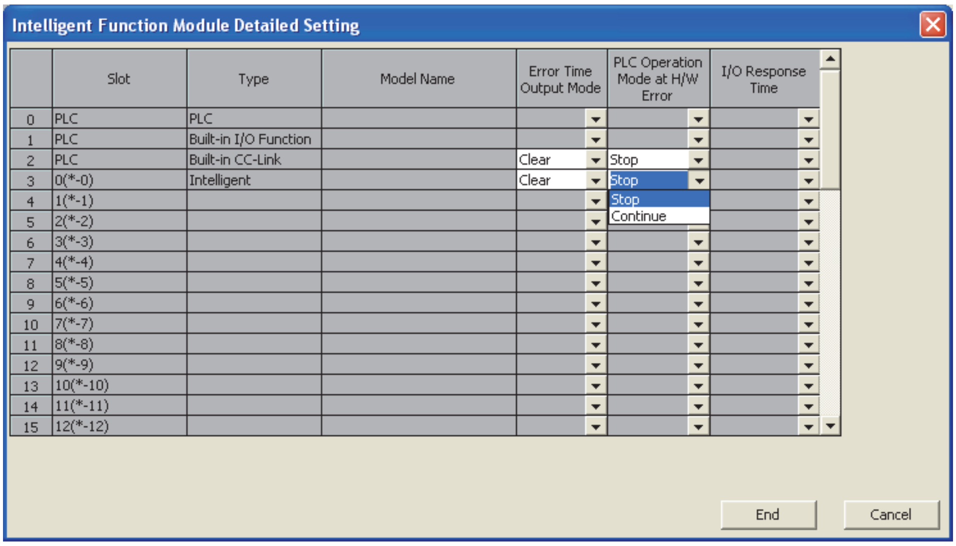
(1) Setting procedure

1. **Select "Intelligent" under "Type" in the I/O Assignment tab, and click the** Detailed Setting **button.**

Project Window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [I/O Assignment]

2. **Select either of the following items for "PLC Operation Mode at H/W error".**

"Stop" or "Continue"



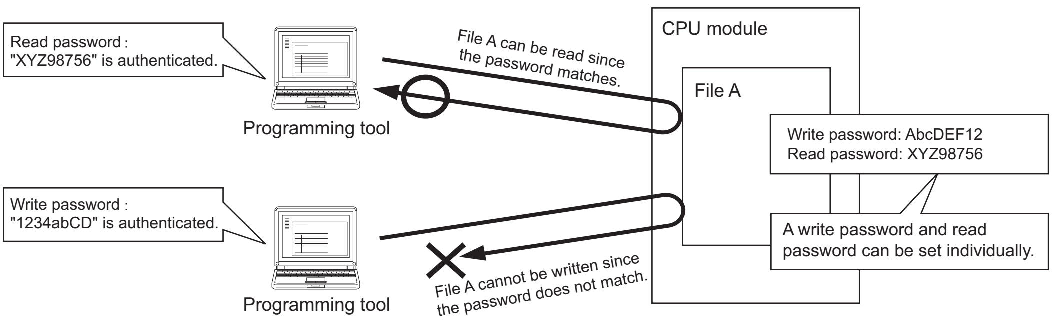
3.11 Security Function

This function protects data in the CPU module against tampering and theft by unauthorized persons. Use the following functions according to your applications and needs.

Function	Purpose	Reference
File password 32	To limit access to each file in the CPU module	Page 105, Section 3.11.1
Remote password	To limit access to the CPU module from external devices.	Page 110, Section 3.11.2
Block password	To limit access to each POU	GX Works2 Version 1 Operating Manual (Common)

3.11.1 File Password 32

This function sets write password and read password for each file stored in the CPU module so that files are protected against tampering and theft by unauthorized persons.



(1) File protection timing

File protection is enabled immediately after the passwords are registered, and it is disabled immediately after the passwords are deleted.

(2) Password target files

A password can be set to the following files.

- Program
- Device comment
- Initial device value
- Parameter
- Source information

(3) Operations that are controlled and the number of characters

A password can be set to the following operations. The minimum number of characters in the password should be 4, and the maximum number should be 32.

- Reading files
- Writing files
- Reading/writing files

Remark

For characters allowed in passwords, refer to "Explanation of Registration Conditions" on the Input Password window.

(4) Online operations that require authentication

Authentication is required to execute the following operations to password-protected files.
(☞ Page 108, Section 3.11.1 (7))

- Write to PLC (data writing)
- Read from PLC (data reading)
- Online change (data writing)
- Change TC setting value (data writing)
- Verify with PLC (data reading)
- Create/Change or Delete of a password (data reading and writing)
- Delete PLC data (data writing)

(5) Operating procedure

To change, delete, or unlock the password, refer to the following.
Manual for the programming tool used

(6) Precautions

(a) Boot from an SD memory card

The following table shows the relationship between the boot operation availability and file password 32 setting.

—: No combination available

Transfer source file		Transfer destination file		Password status	Boot operation
File	Password	File	Password		
Exist	Set	Exist	Set	Matched	Enabled
				Not matched	Disabled
			Not set	—	Disabled
		Not exist	—	—	Enabled
	Not set	Exist	Set	—	Disabled
			Not set	—	Enabled
		Not exist	—	—	Enabled
Not exist	—	—	—	—	—

If boot file settings are configured to more than one file, the files can be transferred only when all the passwords match. If all the passwords do not match, data in the SD memory card are not transferred and "BOOT ERROR" (error code: 2213) occurs.

(b) When "Clear Program Memory" is selected in parameter (Boot File tab)

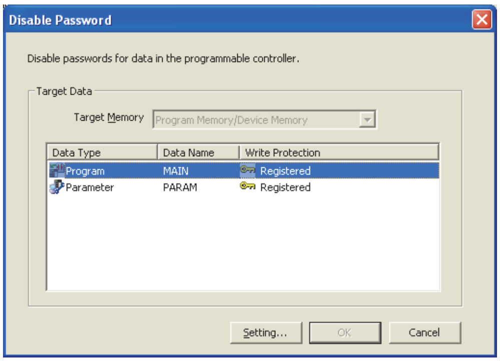
Even when a password is registered, files will be formatted.

(7) Authentication method

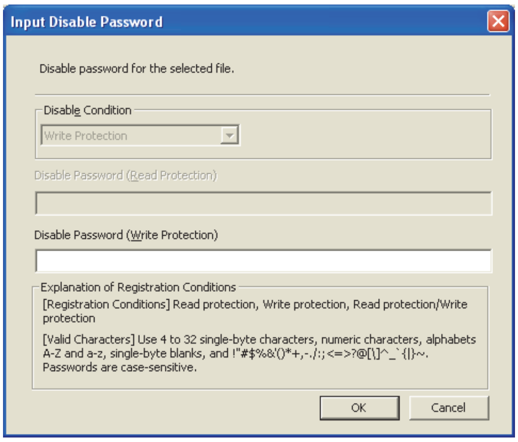
Passwords are authenticated in three ways.

- By a programming tool
- By the FTP server
- By the MC protocol

(a) Authentication by a programming tool



↓



1. **Whenever an online operation requiring password authentication is executed, the "Disable Password" window appears. Select an authentication target file, and click the Setting... button.**

2. **Enter a password in the "Input Disable Password" window.**

Point

The entered password is valid until the project is closed.

(b) Authentication by the FTP server

To access a password-protected file from external devices using the FTP server function, password authentication is required for each file. To change a file to access, password authentication is required again.

○: Authentication required, —: Authentication not required

Operation	Command	Password authentication	
		Data write	Data read
Deleting a file in a CPU module	delete	○	—
Reading a file from a CPU module	get	—	○
Deleting a file in a CPU module	mdelete	○	—
Reading a file from a CPU module	mget	—	○
Changing the file name in a CPU module	rename	○	—
Changing or displaying the file attribute in a CPU module	change	○	—

To authenticate a password, use FTP commands for password authentication.

Operation	Command name
Disable Password (Write)	quote passwd-wr <password>
Disable Password (Read)	quote passwd-rd <password>

For details on commands, refer to the following.

📖MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)

(c) Authentication by the MC protocol

To access a password-protected file from external devices using the MC protocol, the request message format of the MC protocol must be changed and a command for the file password 32 must be specified.

1. **Add "Keyword" at the end of the request message, and set a password in that area.**
2. **Authenticate the password using the password set to "Keyword"**
3. **For the commands requiring password authentication, specify 0004 (for file password 32) in the "Subcommand" area of a request message.**

Function	Command (subcommand)
File delete	1822 (0004)
File attribute change	1825 (0004)
File copy	1824 (0004)
File open	1827 (0004)

For details, refer to the following.

📖MELSEC Communication Protocol Reference Manual

3.11.2 Remote Password

This function prevents unauthorized access to the CPU module from external devices.

(1) Settable modules and the number of settable modules

The following shows the modules for which the remote password can be set and the number of settable modules. (The number of settable modules does not indicate the number of connectable modules in the system where a CPU module is used.)

- CPU module (Built-in Ethernet port): 1
- Serial communication module: 8
- Ethernet module: 8

Remark

For details on the remote password for a serial communication module or Ethernet module, refer to the manual of each module.

(2) Flow from remote password setting to reflection of the password

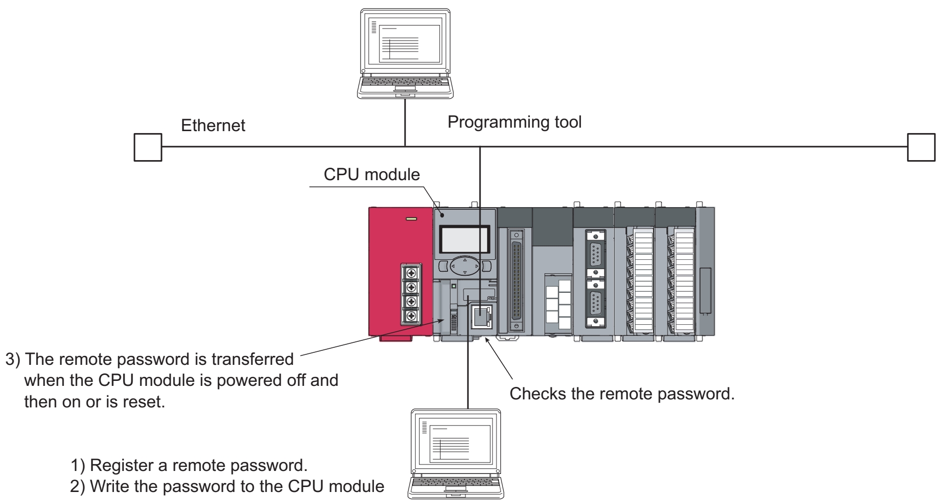
1. Register a remote password. (☞ Page 375, Appendix 1.4)

Project window ⇨ [Parameter] ⇨ [Remote Password]

2. Write it to the CPU module. (Write to PLC)

The remote password is transferred to the target module when:

- the CPU module is powered off and then on or
- the CPU module is reset.



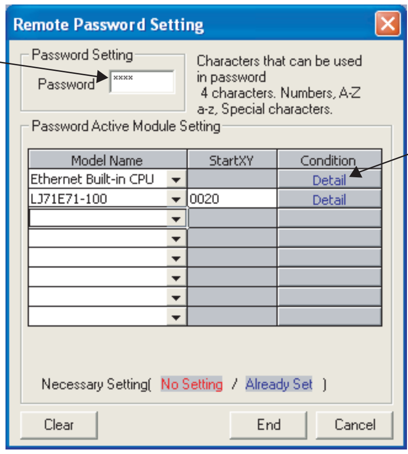
A remote password proceeds the lock processing so that the access is limited. Only when the entered remote password matches with the registered password, access is allowed (unlock processing (cancel)).

(3) Changing/deleting a remote password

Open the "Remote Password Setting" dialog box. (☞ Page 375, Appendix 1.4)

Project window ⇨ [Parameter] ⇨ [Remote Password]

Enter a remote password.



For a Ethernet Built-in CPU, configure the "Detail" setting.

- To change a remote password, write the new remote password to the CPU module.
- To delete a remote password, click the Clear button and write the remote password to the CPU module.

3.12 Remote Operation

Remote operation allows externally (by programming tool, external devices using the MC protocol, or using remote contacts) changing the operating status of the CPU module. There are four types of remote operations:

- Remote RUN/STOP (☞ Page 112, Section 3.12.1)
- Remote PAUSE (☞ Page 114, Section 3.12.2)
- Remote RESET (☞ Page 116, Section 3.12.3)
- Remote latch clear (☞ Page 117, Section 3.12.4)

(1) The relationship between a CPU module and a remote operation.

Availability of remote operations varies depending on the status of a CPU module.

CPU module status	Remote operation				
	Remote RUN	Remote STOP	Remote PAUSE	Remote RESET	Remote latch clear
RUN status	RUN status	STOP status	PAUSE status	Not operable	Not operable
STOP status	STOP status	STOP status	STOP status	Reset*1	Latch clear

*1 Including the stop error status of the CPU module.

3.12.1 Remote RUN/STOP

This operation externally changes the operating status of the CPU module to RUN or to STOP, keeping the switch of the CPU module in the RUN position.

This operation is useful when:

- the CPU module is inaccessible, or
- changing the status of a CPU module that is in a control panel to RUN or STOP by external signals.

(1) Program operation

(a) Remote RUN

The CPU module changes its operating status to RUN and executes a program from the step 0. (The remote RUN operation must be performed to the CPU module whose operating status has been changed to STOP by the remote STOP operation.)

(b) Remote STOP

The CPU module executes a program until the END instruction and changes its operating status to STOP.

(2) Executing method

The following three methods are available.

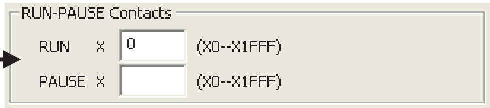
- Using a RUN contact
- Using a programming tool
- By an external device using the MC protocol
- Using the link dedicated instruction of the CC-Link IE Field Network master/local module

(a) Using a RUN contact

Set a RUN contact.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Set a device used as a contact.



The remote RUN/STOP operation can be performed by turning on/off the set RUN contact.

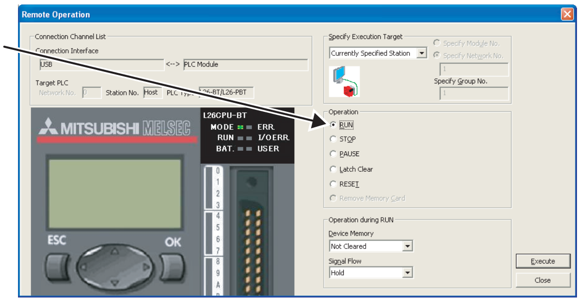
- When the RUN contact is turned off, the CPU module status changes to RUN.
- When the RUN contact is turned on, the CPU module status changes to STOP.

(b) Using a programming tool

Open the "Remote Operation" dialog box.

[Online] ⇨ [Remote Operation]

Select "RUN" or "STOP" for execution.



Point

A remote RUN operation executed only from the programming tool that has executed a remote STOP operation to the CPU module.

(c) By an external device using the MC protocol

Use MC protocol commands.

MELSEC Communication Protocol Reference Manual

(d) Using the link dedicated instruction of the CC-Link IE Field Network master/local module

Use the link dedicated instruction of the CC-Link IE Field Network master/local module.

MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual

3.12.2 Remote PAUSE

This operation externally changes the operating status of the CPU module to PAUSE, keeping the CPU module switch in the RUN position.
PAUSE status is a status where program operations in the CPU module are stopped, holding the status (on or off) of all outputs (Y).
This operation is useful, especially during the process control, to hold the on status of outputs (Y) even after the operating status of the CPU module is switched from RUN to STOP.

(1) Executing method

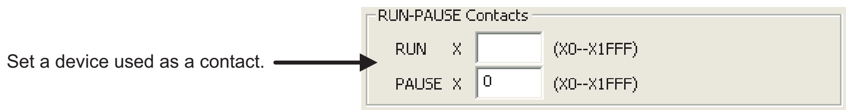
There are three methods for performing the remote PAUSE operation.

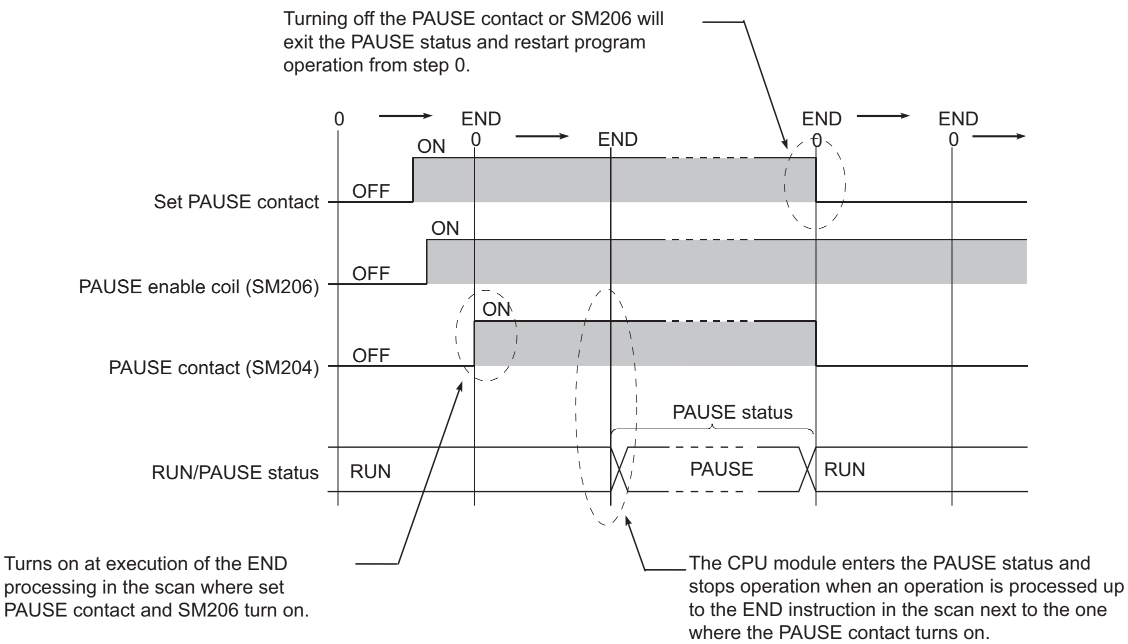
- Using a PAUSE contact
- Using a programming tool
- By an external device using the MC protocol

(a) Using a PAUSE contact

Set a PAUSE contact.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]





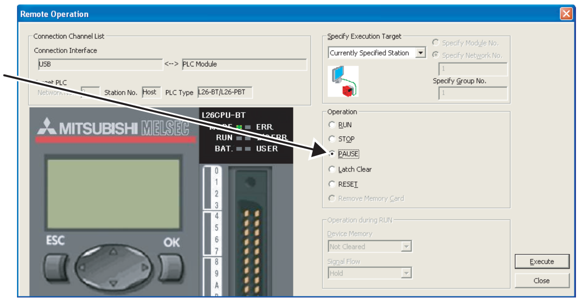
Point

When setting a PAUSE contact, set a RUN contact as well. (Setting of only a PAUSE contact is not allowed.)

(b) Using a programming tool

Open the "Remote Operation" dialog box.

[Online] ⇨ [Remote Operation]



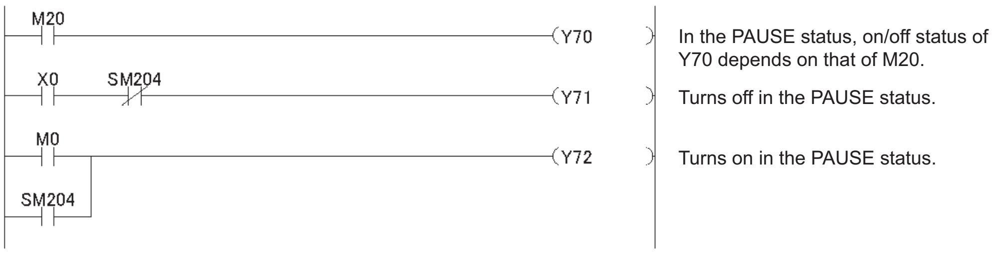
(c) By an external device using the MC protocol

Use MC protocol commands.

📖MELSEC Communication Protocol Reference Manual

(2) When forcibly keeping output status

To forcibly keep the output status (on or off) in the PAUSE status, provide an interlock with the PAUSE contact (SM204).



3.12.3 Remote RESET

This operation externally resets the CPU module when the CPU module is in the STOP status. Even if the switch of the CPU module is in the RUN position, this operation can be performed when the module is stopped due to an error. Use this function when an error occurred in the CPU module that is beyond the reach.

(1) Executing method

There are two methods for performing the remote RESET operation.

- Using a programming tool
- By an external device using the MC protocol

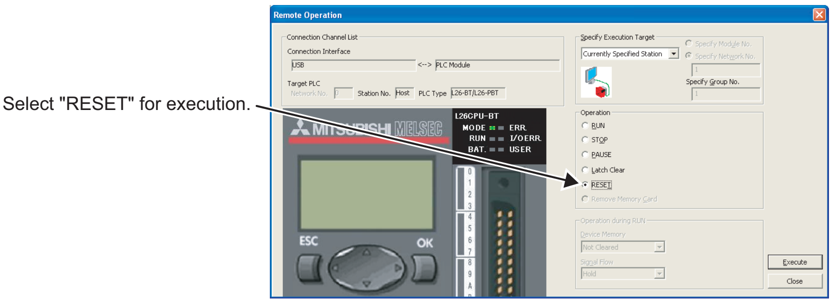
Before performing the remote RESET operation, select the "Allow" checkbox for the remote RESET operation in the PLC System tab of the PLC Parameter dialog box, and then write the parameter setting to the CPU module.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

(a) Using a programming tool

Open the "Remote Operation" dialog box.

[Online] ⇨ [Remote Operation]



(b) By an external device using the MC protocol

Use MC protocol commands.

MELSEC Communication Protocol Reference Manual

(2) Status after reset processing

When the remote RESET operation is completed, the CPU module will be placed in the operating status set by the switch. Consider the operating status of the CPU module after the reset operation if the CPU module had stopped due to an error.

(3) Precautions

Note that the CPU module may not be reset by the remote RESET operation due to noise. In this case, reset the CPU module using the switch or power off and then on the CPU module.

3.12.4 Remote Latch Clear

This function resets the device data when the CPU module is in the STOP status. *1

This operation is useful when:

- the CPU module is inaccessible or
- externally performing latch clear to the CPU module in a control panel.

*1 Device data in the range where a latch clear operation is disabled cannot be reset. (☞ Page 89, Section 3.4 (4))

(1) Executing method

The following two methods are available.

- Using a programming tool
- By an external device using the MC protocol

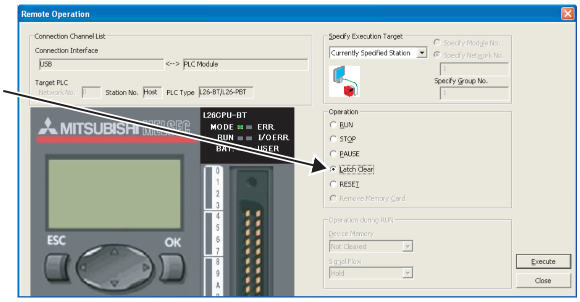
Set the CPU module in the STOP status. (Not available in the RUN status)

(a) Using a programming tool

Open the "Remote Operation" dialog box.

[Online] ⇨ [Remote Operation]

Select "Latch clear" for execution.



(b) By an external device using the MC protocol

Use MC protocol commands.

📖MELSEC Communication Protocol Reference Manual

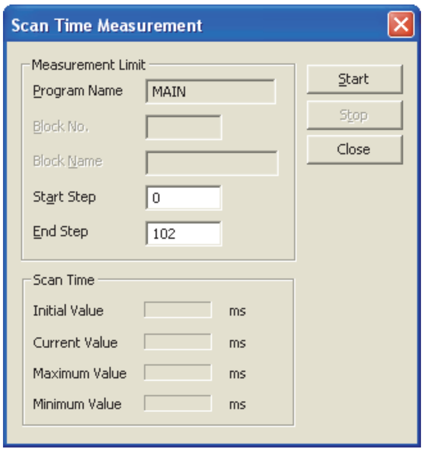
3.13 Scan Time Measurement

This function displays the processing time of set program section during ladder monitoring. The time required for the subroutine and interrupt programs can be measured.

(1) Execution

1. Open the "Scan Time Measurement" dialog box.

[Debug] ⇨ [Scan Time Measurement]



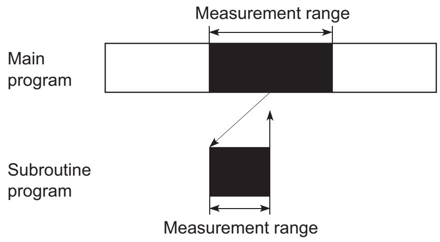
2. Specify the start and end steps and click the Start button.

Point

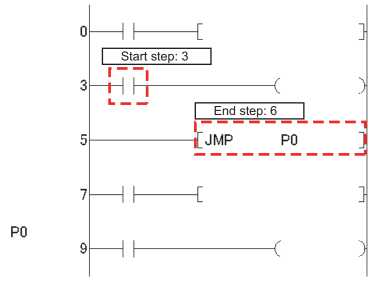
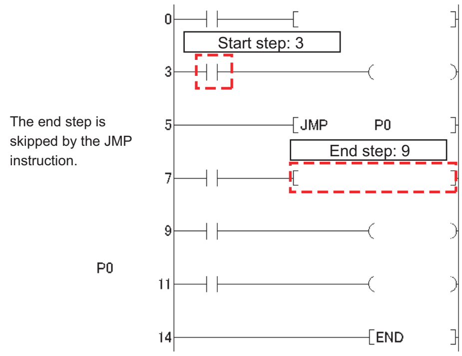
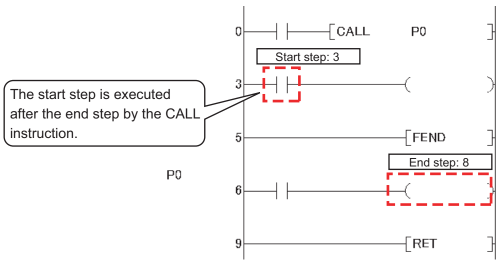
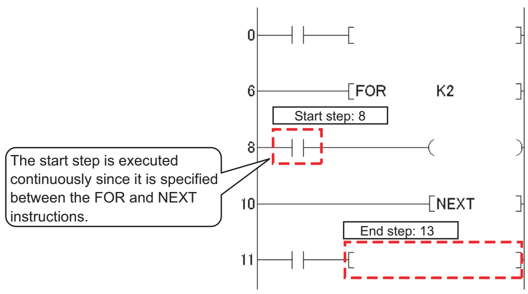
When displaying the "Scan Time Measurement" dialog box after specifying the scan time measurement range in monitor mode, the start and end steps are set in the specified ranges such as shown below.

(2) Precautions

- The minimum unit of measurement time is 0.01ms. If the measurement time is less than 0.01ms, 0.000ms is displayed.
- When between the FOR and NEXT instructions is specified, the execution time of one scan between the specified steps is displayed.
- If a subroutine program call instruction (CALL instruction) exists within the scan time measurement range, processing time for the subroutine program is included in the scan time.

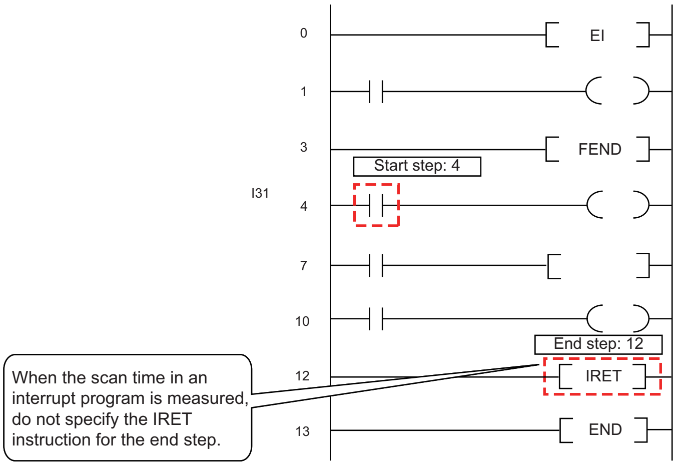


- If an interrupt/fixed scan execution type program is executed within the scan time measurement range, program execution time is added to the scan time.
- Scan time from one program file to another cannot be measured. Also, scan time cannot be updated on the "Scan Time Measurement" dialog box in the following cases.

When the branch instruction is specified in the end step	When only the start step is executed
Example: The JMP instruction is specified in the end step. 	Example: The specified end step is not executed by the JMP instruction. 
When the end step is executed before the start step	**When the start step is executed continuously**
Example: The start step is specified as the next step of the CALL instruction and the end step is specified in a subroutine program executed by the CALL instruction. 	Example: Only the start step is specified between the FOR and NEXT instructions. 

When the IRET instruction, FEND instruction, BREAK instruction, or RET instruction is specified in the end step

Example: The IRET instruction is specified in the end step of an interrupt program by I31.



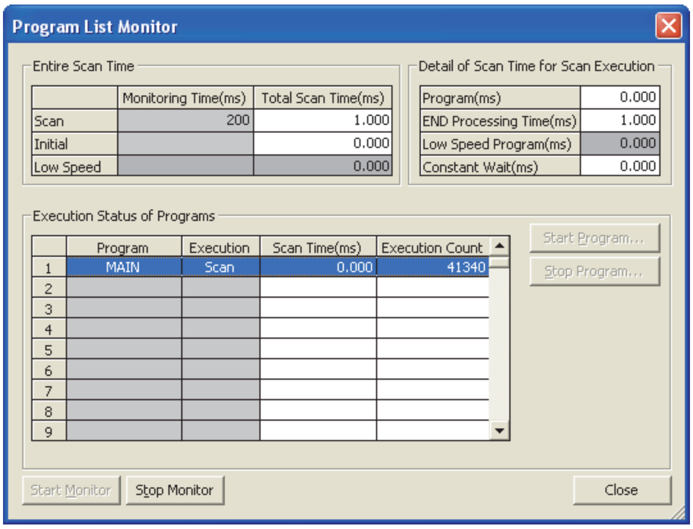
3.14 Program List Monitor

This function displays the processing time of the program being executed. The scan time, number of execution times, and processing time by item can be displayed for each program.

(1) Execution

Open the "Program List Monitor" dialog box.

[Online] ⇨ [Monitor] ⇨ [Program List...]



Item		Description
Entire Scan Time	Monitoring Time(ms)	The monitoring time of each program is displayed. If the scan time exceeds this time, the CPU module detects "WDT ERROR".
	Total Scan Time(ms)	The total time of each item in "Detail of Scan Time for Scan Execution" is displayed. When constant scan time is set, the constant scan time is displayed.
Detail of Scan Time for Scan Execution	Program(ms)	The total execution time of the scan execution type program is displayed.
	END Processing Time(ms)	The END processing time is displayed.
	Low Speed Program(ms)	Since low-speed execution type programs cannot be used "0.000" is displayed.
	Constant wait(ms)	The constant scan waiting time is displayed when the constant scan time is set.
Execution Status of Programs	Program	The execution status of a program selected at the program tab of the PLC Parameter dialog box is displayed.
	Execution	The program type set in the PLC Parameter dialog box is displayed.
	Scan Time(ms)	The actual scan time (current value) is displayed. When a program is in stop (standby) status, the scan time is displayed as 0.000 ms.
	Execution Count	The number of execution times of programs before monitoring is displayed, setting the measurement start as "0". The number of execution times is displayed up to 65535 and returns to 0 when the 65536 is measured. The execution times is held even when the program is stopped.

Point

- The scan time of a fixed scan execution type program is not displayed during its execution. "-" is displayed in the Scan time column.
- When the POFF instruction is executed, a non-execution processing is performed for one scan. The number of execution times displayed is the addition of the execution times of the non-execution processing.

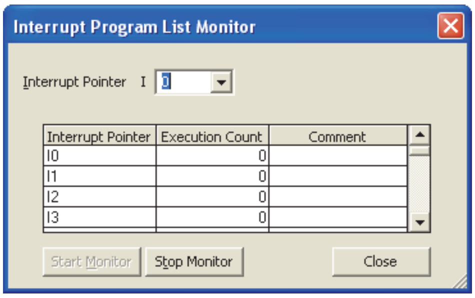
3.15 Interrupt Program List Monitor

This function displays the number of executions of an interrupt program.

(1) Execution

Open the "Interrupt Program List Monitor" dialog box.

[Online] ⇨ [Monitor] ⇨ [Interrupt Program List...]



Item	Description
Interrupt Pointer	An interrupt pointer is displayed.
Execute Count	The number of executions of an interrupt program is displayed. This function starts counting after the CPU module is in the RUN status. When the counting reaches 65536 times, it is reset to 0.
Comment	Device comments created to an interrupt pointer is displayed.

3.16 Monitor condition setting

This function monitors the CPU module under specified conditions.

Remark

Before setting the monitor condition, check the version of the programming tool used.
(☞ Page 376, Appendix 2)

(1) Setting method

The following two methods are available for the monitor condition setting.

- Monitor execution condition setting
- Monitor stop condition setting

For details on the setting method, refer to the following manuals.

📖Manual for the programming tool used

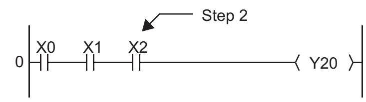
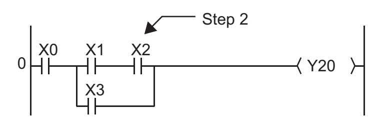
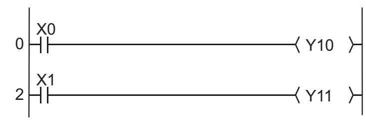
(a) When only a step number is specified

Monitor data is collected when the status immediately before execution of the specified step becomes the specified status.

- When the operation of the specified step changes from the non-execution status to the execution status: <↑>
- When the operation of the specified step changes from the execution status to the non-execution status: <↓>
- Always when the operation of the specified step is being executed only: <ON>
- Always when the operation of the specified step is not being executed only: <OFF>
- Always regardless of the status of the operation of the specified step: <Always>

Point

- If a step between the AND/OR blocks is specified as a monitor condition, monitor data is collected when the status immediately before execution of the specified step becomes the specified status by the LD instruction in the block. The monitoring timing depends on the ladder of the specified step. The following shows examples of monitoring when the step 2 is on (Step No. [2] = <ON>).

Condition	Description
When the step 2 is connected by the AND instruction	When both X0 and X1 are on, the monitor execution condition is established. Ladder mode  List mode 0 LD XO 1 AND X1 2 AND X2 3 OUT Y20
When the step 2 is connected in the middle of the AND/OR block	When X1 is on, the execution condition is established. (The on/off status of X0 does not affect the establishment of the monitor execution condition.) Ladder mode  List mode 0 LD XO 1 LD X1 2 AND X2 3 OR X3 4 ANB 5 OUT Y20
When the start of a ladder block other than the step 0 is specified for the step number as a detailed condition	Monitor data is collected when the execution status of the instruction immediately before execution becomes the specified status. If (Step No. [2] = <ON>) is specified in the following ladder, monitor data is collected when OUT Y10 turns on. Ladder mode  List mode 0 LD XO 1 OUT Y10 2 LD X1 3 OUT Y11

- When "0" is specified as the step No., set the condition to "Always".

(b) When only a device is specified

Either word device or bit device can be specified.

- When a word device is specified

Monitor data is collected when the current value of the specified word device becomes the specified value. Specify the current value in decimal or hexadecimal.

- When a bit device is specified

Monitor data is collected when the execution status of the specified bit device becomes the specified status. Select the execution condition (on the rising edge or falling edge).

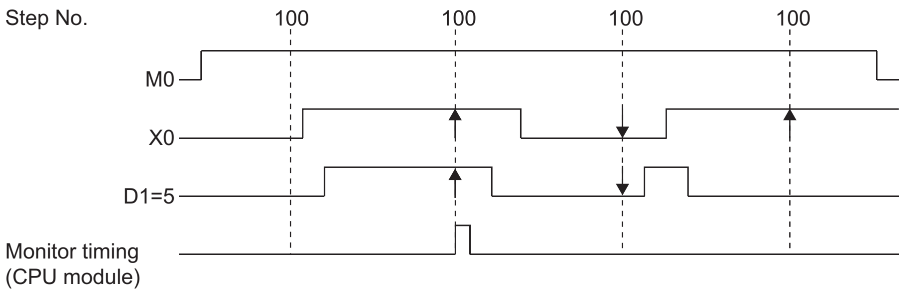
(c) When a step number and device are specified

Monitor data is collected when the status immediately before execution of the specified step becomes the specified status or the status (current value) of the specified bit device (word device) becomes the specified value.

Point

If the rise of step 100 and D1 = 5 (Step No. [100] = <↑>, Word device [D1] = [K5]) are specified as the execution condition, the monitor execution condition is established on the rising edge of the step 100 and also D1=5.

The monitor interval of a programming tool depends on the processing speed of the programming tool. For the monitor execution conditions established at the interval shorter than the monitor interval of the programming tool, monitor is executed only when the monitor execution condition is established at the monitor timing of the programming tool.



(2) Precautions

(a) Files to be monitored

When monitoring after setting the monitor condition, the file displayed on a programming tool is monitored. Match the name of the file to be monitored on the CPU module with that on the programming tool by executing [Online] -> [Read from PLC].

(b) No file register setting

If the file register is monitored when there is no file register used, "$FFFF_H$" is displayed.

(c) Device assignment

For a monitor operation, the device assignment in the CPU module and the programming tool must be the same.

(d) Monitoring by multiple users simultaneously

When monitoring the buffer memory of an intelligent function module, the scan time increases as in case when executing the FROM/TO instructions.

(e) Monitoring by multiple users simultaneously

The following precautions should be followed:

- High speed monitoring can be performed by increasing 1K step in the system area for every monitor file of other stations when formatting the program memory or setting the Boot file of the PLC Parameter.. Up to 15 stations can be set as the station monitor file, however, the program area will be reduced by the files' worth of areas.
- When the monitor condition or monitor stop condition is set, only one user can perform monitoring.

(f) Setting a monitor stop condition

A monitor stop condition can be set only in the ladder monitor.

(g) Specifying the same device as a condition

When specifying the same device as a monitor condition or monitor stop condition, set the on/off status as well.

(h) Specifying a step number as a monitor condition

If an instruction in the specified step is not executed in such cases described below, the monitor condition will not be established.

- The specified step is skipped with the CJ, SCJ, or JMP instruction.
- The specified step is the END instruction and never be executed because the FEND instruction also exists in the program.

(i) During monitor condition registration

Do not reset the CPU module while monitoring conditions are being registered.

(j) Monitor operation with monitor condition setting

When monitor operation with monitor condition setting is performed, other applications on the same personal computer cannot execute any online function using the same route for the monitor operation. The following shows examples of other applications:

- Programming tool
- Applications using MX Component
- MX Sheet

If any online function is executed by other applications using the same route for the monitor operation, the following situations may occur.

- No response comes back from the CPU module to the online function. (A time-out occurs in the online communication function with the CPU module.)
- The CPU module detects an error (error code: 4109) for the online function executed.
- Even when the monitor condition is established in the CPU module, monitoring results cannot be updated for the monitor operation with monitor condition setting.

3.17 Local Device Monitor/Test

This function enables debugging of local devices of the program that is being monitored.

(1) Monitoring local devices

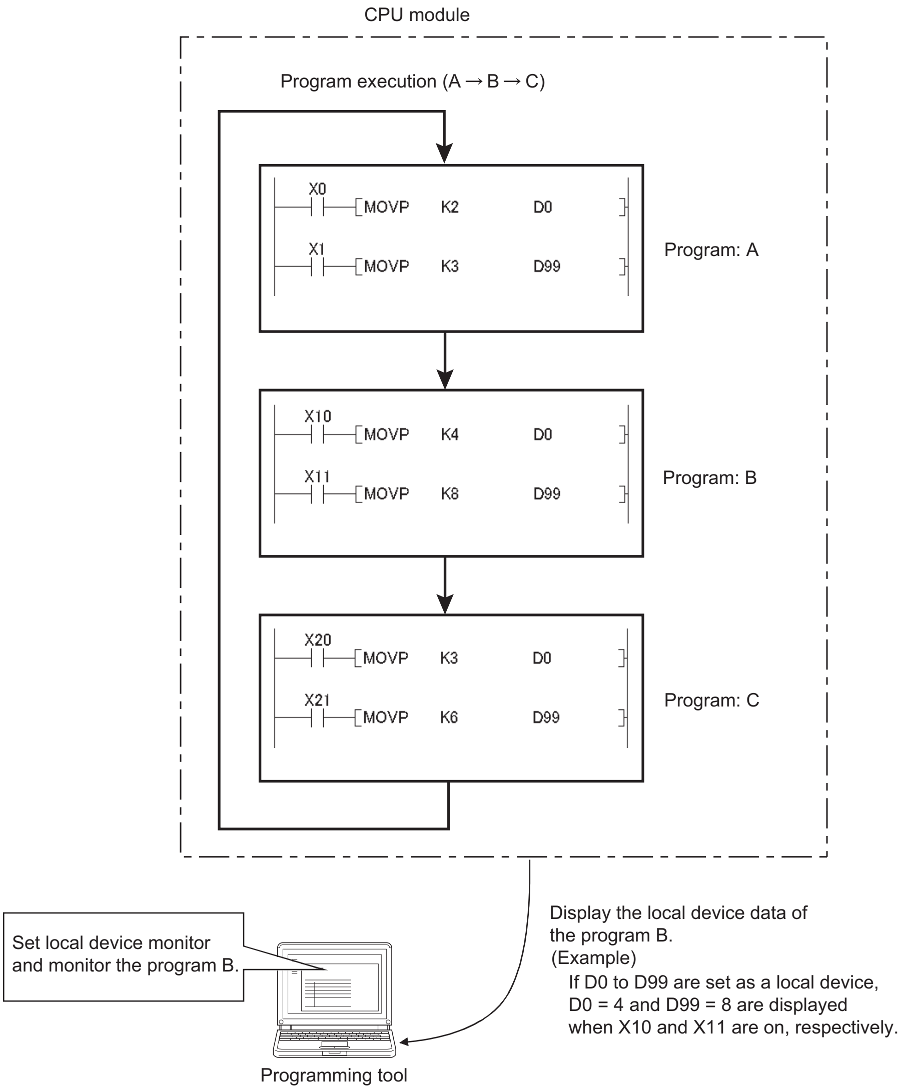
The following shows examples of monitoring local devices.

Ex. D0 to D99 are set as local devices and three programs "A", "B" and "C" are executed by the CPU module. (Three programs are to be executed in the order of A → B → C → (END processing) → A → B....)

Item	Monitored device	
	D0 (Local device)	D100 (Global device)
Local device monitor is set	The D0 value in the specified program (local device for a program) is monitored.*1	The D100 value after execution of the specified program is monitored.
Local device monitor is not set	The D0 value after execution of the program "C" is monitored.	The D100 value after execution of the program "C" is monitored.

*1 When "Not Used" is set for "Local Device" in File Usability Setting of the Program tab, the D0 value after execution of the specified program is monitored.

When local devices are set to be monitored and the program "B" is displayed for monitoring, the local device(s) used in the program "B" can be monitored.



(2) Device test procedure

1. **Connect a personal computer to the CPU module**
2. **Display a program in ladder mode**
3. **Switching to the monitor mode**

   [Online] ⇨ [Monitor] ⇨ monitor mode

4. **Select [Local device monitor] from the monitor window.**

(3) Number of programs that can be monitored/tested

Local devices of 16 programs can be simultaneously monitored or tested from multiple programming tools.

(4) Precautions

- One programming tool can monitor or test local devices in one program at a time. Local devices in multiple programs cannot be monitored or tested simultaneously.
- When local devices in a stand-by type program are monitored, data in local devices are saved and restored. For this reason, the scan time increases.
- When local devices in a fixed scan execution type program are monitored, data in local devices cannot be acquired and "0" is displayed.

3.18 External Input/Output Forced On/Off

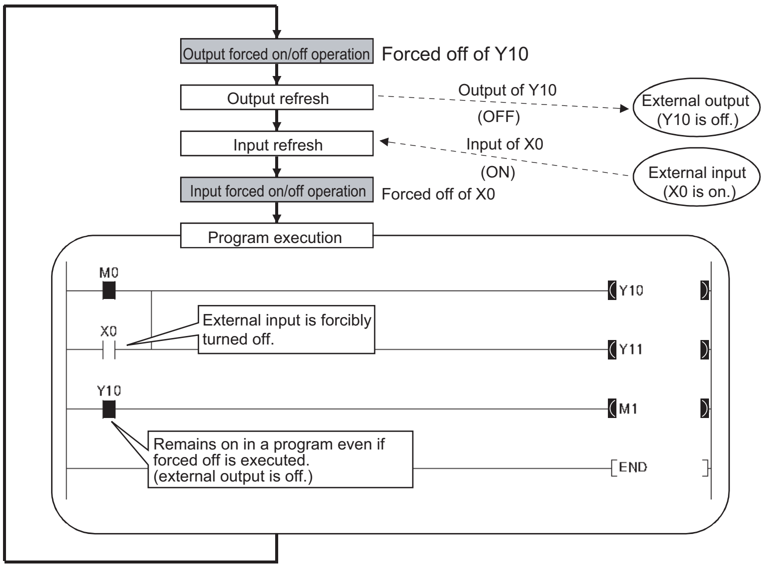
This function forcibly turns on or off the external input/output of the CPU module.

(1) Input/output operation when a forced on/off operation is performed

Three types of forced on/off operations are available as shown in the following table. The following table shows the status of input (X) and output (Y) when a forced on/off operation is performed. Note that a program operation takes priority when a contact of output (Y) is used in the program.

Operation	Input (X) operation	Output (Y) operation
Forced on registered	The CPU module performs program operations using inputs forcibly turned on.	The CPU module outputs "on" externally regardless of the results of program operations.
Forced off registered	The CPU module performs program operations using inputs forcibly turned off.	The CPU module outputs "off" externally regardless of the results of program operations.
Forced on/off cancellation	The CPU module performs program operations using external inputs.	The CPU module outputs the results of program operations externally.

The following shows the input/output operation when a forced on/off operation is performed.



(2) Specifications

(a) CPU module status

Forced on/off can be registered regardless of the operating status (RUN/STOP) of the CPU module. Note, however, that only input can be forcibly turned on/off during a stop error. The CPU module outputs on/off data only to device Y.

(b) Registerable devices

Forced on/off can be registered as many as the number of I/O device points in the CPU module.

(c) Target input/output

- Input (X) and output (Y) of modules used in the built-in I/O function.
- Input (X) and output (Y) of connected modules
- Input (X) and output (Y) of the CPU module to be refreshed from RX/RY of the CC-Link IE Field Network master/local module or the CC-Link system master/local module (including built-in CC-Link function)
- Input (X) and output (Y) of the CPU module to be refreshed from RX/RY of CC-Link IE Field Network Basic

(d) Forced on/off timing

Refresh area	Timing
• Input and output used in the built-in I/O function. • Input and output of modules connected	• During END processing (input refresh) • At execution of the COM instruction (input refresh) • At execution of an instruction using direct access input (DX) and direct access output (DY) • At execution of the RFS or MTR instruction • At execution of an instruction used for a system
Input and output of the CPU module to be refreshed from RX/RY of the CC-Link IE Field Network master/local module or the CC-Link system master/local module (including built-in CC-Link function)	• During END processing (auto refresh) • At execution of the COM instruction (auto refresh) • At execution of the ZCOM instruction (auto refresh)
Input and output of the CPU module to be refreshed from RX/RY of CC-Link IE Field Network Basic	• During END processing (auto refresh) • At execution of the COM instruction (auto refresh)

(e) Number of registerable devices

Up to 32 devices in total can be registered.

(f) Checking the execution status

- Reading the forced on/off registration status of a programming tool.
- Flashing of the MODE LED (green) (The MODE LED flashes in green when a device is registered.)
- The on status of the 1st bit in SD840 (Debug function usage).

Point

When checking by the MODE LED or SD840, remind that they are also used to check the execution status of executional conditioned device tests. (☞ Page 135, Section 3.19)

(g) Forcibly turning on/off from multiple programming tools

Forced on/off can be registered to a single CPU module from multiple programming tools. In this case, the last registration is effective. For this reason, the forced on/off status which is different from the status actually registered in the CPU module may be displayed on the screen that registered forced on/off earlier. When the forced on/off registration is performed from multiple programming tools, click the [Update Status] button to update the data, and execute the function.

(h) Status of devices after forced on/off registration data are canceled

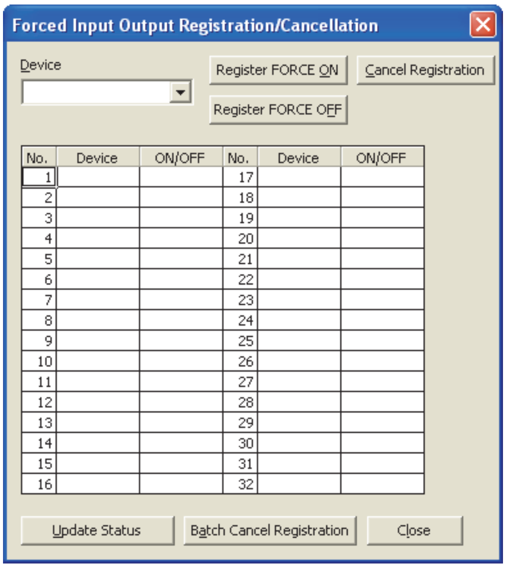
Forced on/off registered device		Program operations (on/off) performed	Program operations (on/off) not performed
Input	• Input used in the built-in I/O function • Input from connected modules on the base unit	Uses the on/off status input from modules.	
	Input of the CPU module to be refreshed from RX of the CC-Link IE Field Network master/local module or the CC-Link system master/local module (including built-in CC-Link function)	Uses the on/off status refreshed via CC-Link.	
	Input of the CPU module to be refreshed from RX of CC-Link IE Field Network Basic	Uses the on/off status refreshed from the CPU module (when CC-Link IE Field Network Basic is used)	
	Output other than above (outside of the refresh range)	Uses the results of program operations.	Holds the forced on/off status.
Output	• Output used in the built-in I/O function. • Output from connected modules on the base unit.	Outputs the results of program operations.	Holds the registered on/off status.
	Output of the CPU module to be refreshed from RY of the CC-Link IE Field Network master/local module or the CC-Link system master/local module (including built-in CC-Link function)		
	Output of the CPU module to be refreshed from RY of CC-Link IE Field Network Basic	Outputs the results of program operations.[*1]	
	Output other than above (outside of the refresh range)	Outputs the results of program operations. (The results are not output externally.)	Holds the forced on/off status.

*1 The result is output for one scan in accordance with the registered on/off state even after forced on/off registration data are canceled.

(3) Operating procedure

1. Open the "Forced Input Output Registration/Cancellation" dialog box.

[Debug] ⇨ [Forced Input Output Registration/Cancellation...]



2. Enter the target device to the "Device" column.

3. Click the button for intended operation.

Button name	Description	Button name	Description
Register FORCE ON	Registers forced on for a specified device.	Cancel Registration	Cancels forced on/off registered for the device specified.
Register FORCE OFF	Registers forced off a specified device.	Batch Cancel Registration	Cancels all forced on/off registration.

The latest on/off status can be checked by clicking the Update Status button.

3.19 Executional Conditioned Device Test

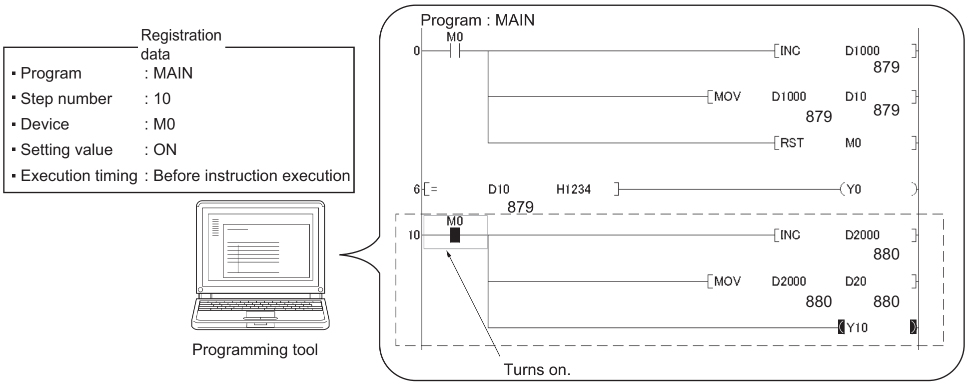
This function changes a device value for the specified step of a program. This enables debugging of the specified ladder block without modifying the program.

Point

The executional conditioned device test is not available for the SFC program.

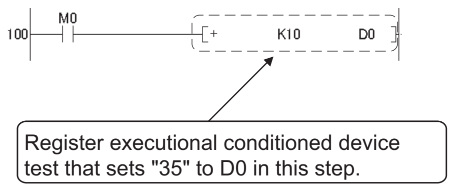
(1) Operation of the executional conditioned device test

A device value is changed based on the registration data once the executional conditioned device test setting is registered.

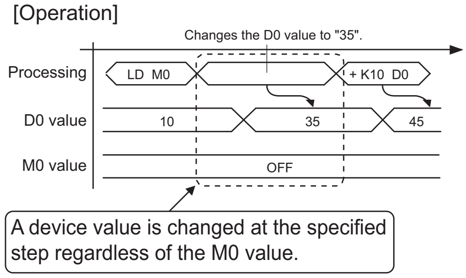


Note that a device value is changed in the specified step regardless of an execution status of the instruction in the specified step.

[Program example]



[Operation]



(2) Applicable devices

The following table shows the applicable devices and the number of settable devices.

Type	Applicable device	Number of settable devices
Bit device	X (DX), Y (DY), M, L, B, F, SB, V, SM, T (contact), ST (contact), C (contact), FX, and FY	Up to 32 (in total)
Word device	T (current value), ST (current value), C (current value), D, D (extended data register), SD, W, W (extended link register), SW, R, ZR, Z, U□\G, FD	
	Digit-specified bit device: X, Y, M, L, F, SM, V, B, SB	
	Indirect specification (@D0): D, SD, W, SW, R, ZR (devices specified with @)	

(3) How to check the execution status

- By displaying the "Check/Disable Executional Conditioned Device Test Registration" dialog box
- By the flash of the MODE LED in green
- By the on status of the first bit in SD840 (Debug function usage)

Point

When checking by the MODE LED or SD840, remind that they are also used to check the execution status of the external input/output forced on/off function. (☞ Page 335, Section 5.9)

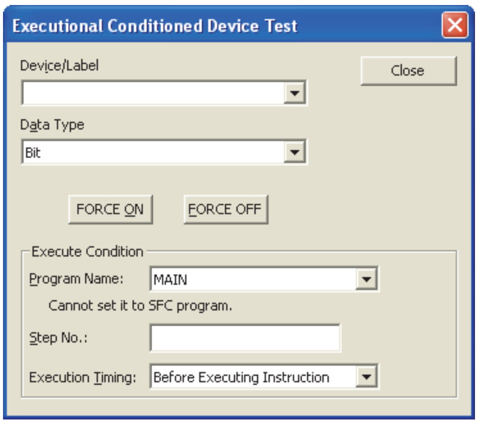
(4) Registering executional conditioned device test settings

(a) Operation method

1. **Open a dialog box to edit the program and select the step number to register.**

2. **The setting can be set by "Executional Conditioned Device Test" dialog box.**

[Debug] ⇨ [Executional Conditioned Device Test] ⇨ [Register Executional Conditioned Device Test...]



Item		Description
Device/Label		Target device or label is displayed. (Data entry is also available)
Data Type	For bit device	Forced on/off operation is executed.
	For word device	Enter a value. (For a signed value, specify decimal or hexadecimal.)
Execute Condition	Program Name	Select the name of the program that is registered in the CPU module.
	Step No.	Specify the step number (step number from 0 to END instruction) that exists in the program. (Specification of a start step of the instruction.)
	Execution Timing	Select either "Before Executing Instruction" or "After Executing Instruction".

Multiple executional conditioned device test settings can be registered for one step number.

However, if more than one executional conditioned device test settings are registered with the same device name and the same execution timing for one step number, the set data are overwritten. (More than one setting with the same device name can be registered for one step, if the execution timing differs.)

Point

- When setting a word device with a different data type, a device is regarded as the same device.
  Example: When a word device is set in the order of "D100 (16-bit integer)" and then "D100 (Real number (single precision))", "D100 (Real number (single precision))" is registered.
- When setting a device with a different modification method (such as a bit-specified word device, digit-specified bit device, or index-modified device), a device is regarded as a different device.
  Example: When a word device is set in the order of "D100.F" and then "D100Z0 (Real number (single precision)), both devices are registered.

(b) Registration from multiple programming tools

The executional conditioned test setting can be registered to a CPU module from multiple programming tools. Note, however, that if an executional conditioned device test setting is registered with the same device name in the same step, the existing data are overwritten.

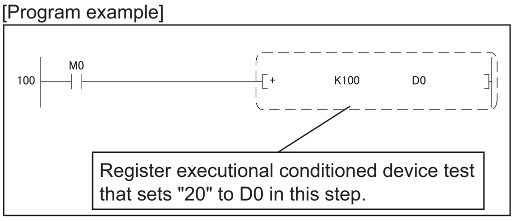
Before registering executional conditioned test settings from multiple programming tools, click the

button to update the registered data.

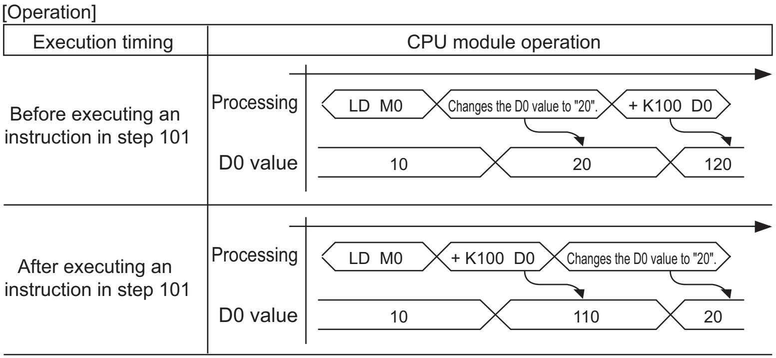
(c) Operation by different execution timing

Operation of the CPU module varies depending on the timing (either before or after an instruction of the specified step) of changing a device value.

[Program example]



[Operation]



Note that, when registering the executional conditioned device test with particular instruction specified, a device value may not be changed depending on the execution timing even after the specified step is executed. The following instructions need to be noted.

• Instructions that do not change device values

A device value is not changed by executing the executional conditioned device test when the execution timing has been set to "After executing instruction", specifying the step for instructions that do not execute the next step, such as branch instructions.

Classification	Instruction	Operation
Stop	STOP	• When the execution condition for an instruction is satisfied, a device value is not changed even when the specified step is executed. • When execution condition for an instruction is not satisfied, a device value is changed after the specified step is executed.
Jump	CJ	
	SCJ	
	GOEND	
Repeated (Loop)	BREAK(P)	
Subroutine program call	CALL(P)	
	FCALL(P)	
	ECALL(P)	
	EFCALL(P)	
	XCALL	
End	FEND	A device value is not changed even when the specified step is executed.
Jump	JMP	
Return from subroutine program	RET	
Return from interrupt program	IRET	

• FOR and NEXT instructions

When the executional conditioned device test setting is registered specifying the step for the FOR or NEXT instruction, timing of device value change is different from the timing when steps for other instructions are specified.

Instruction	Execution timing setting	
	Before Executing Instruction	After Executing Instruction
FOR	Executed once before the start of loop processing.	Executed once after the start of loop processing. (Executed before the operation of the program between the FOR and NEXT instructions.)
NEXT	Executed every loop processing. (Executed after the operation of the program between the FOR and NEXT instructions.)	Executed once after the start of loop processing.

• END instruction

If "After executing instruction" is set, the CPU module returns a registration error to the programming tool.

(d) Number of settings that can be simultaneously registered in one scan

Eight executional conditioned device test settings can be simultaneously registered into the CPU module in one scan. When nine or more executional conditioned device test settings are to be simultaneously registered, they are registered over multiple scans.

(5) Checking the executional conditioned device test

Open the "Check/Disable Executional Conditioned Device Test Registration" dialog box. (☞ Page 139, Section 3.19 (6))

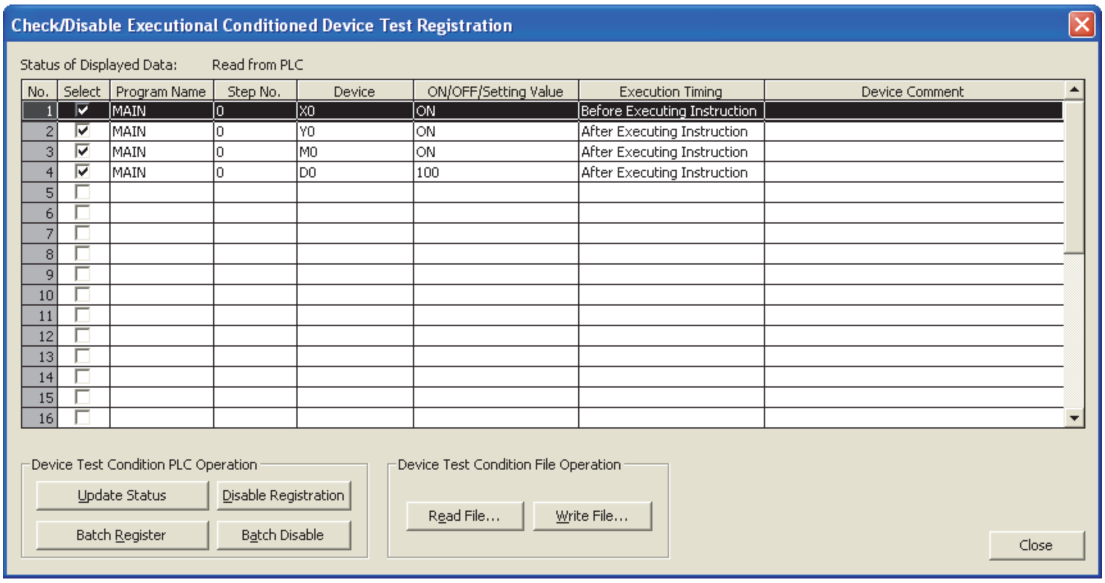
Contents can be viewed by clicking the [Update Status] button.

(6) Disabling the executional conditioned device test

(a) Operating procedure

1. Open the "Check/Disable Executional Conditioned Device Test Registration" dialog box.

[Debug] ⇨ [Executional Conditioned Device Test] ⇨ [Check/Disable Executional Conditioned Device Test...]



2. Select the checkboxes for registered data to disable, and click the [Cancel Registration] button.

The data can also be disabled by any of the following operation.

- Powering off and then on the CPU module
- Resetting the CPU module
- Writing program files stored in the program memory to the CPU module while it is in the STOP status
- Clearing the program file stored in the program memory while the CPU module is in the STOP status
- Formatting the program memory while the CPU module is in the STOP status

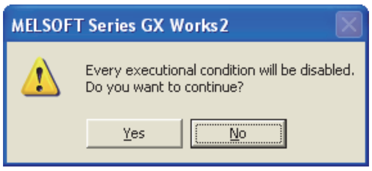
(b) Number of settings that can be simultaneously disabled in one scan

Eight executional conditioned device test settings can be simultaneously disabled in one scan. When nine or more executional conditioned device test settings are to be simultaneously disabled, they are disabled over multiple scans.

(7) Batch-disabling executional conditioned device test settings

1. Open the following message box.

[Debug] ⇨ [Executional Conditioned Device Test] ⇨ [Batch Disable Executional Conditioned Device Test]



This operation is also available from the "Check/Disable Executional Conditioned Device Test Registration" dialog box.

(8) Conditions that registration or disabling is not available

In the following cases, executional conditioned device test setting cannot be registered or disabled. When multiple device settings are to be registered, no setting can be registered if any of them is invalid.

(a) Settings cannot be registered if:

- Specified program name, step, or device does not exist or
- The number of registered settings exceeds 32.

(b) Settings cannot be disabled if:

- Specified program name, step, or device does not exist or
- No executional conditioned device test setting has been registered.

(9) Precautions

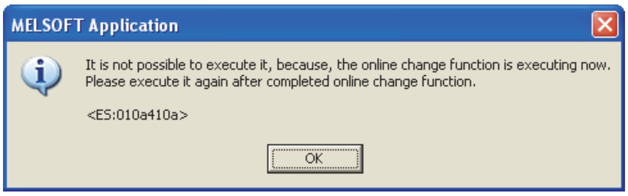
(a) Priority

If any of the following functions is set in the same step number that is specified by the executional conditioned device test setting, the executional conditioned device test takes priority over the other functions.

- Monitor condition setting
- Sampling trace function (trace point or trigger point)
- Scan time measurement (start step or end step)

(b) Executional conditioned device test and writing data to the running CPU module

- If the executional conditioned device test is executed during execution of an online change function, only the online change function is completed while the executional conditioned device test is not executed. The following message box appears. Execute the executional conditioned device test again after the online change has been completed.

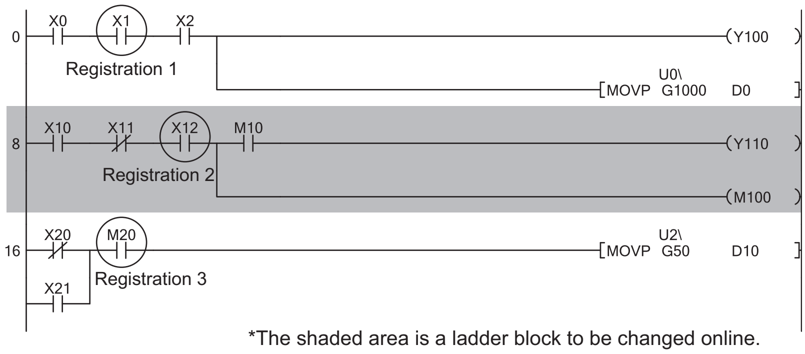


- If the online change function is executed during execution of the executional conditioned device test, only the online change function is completed. The executional conditioned device test setting registered in the ladder block that was changed online, the corresponding setting is disabled.

(c) Online change to the CPU module where the executional conditioned device test setting has been registered

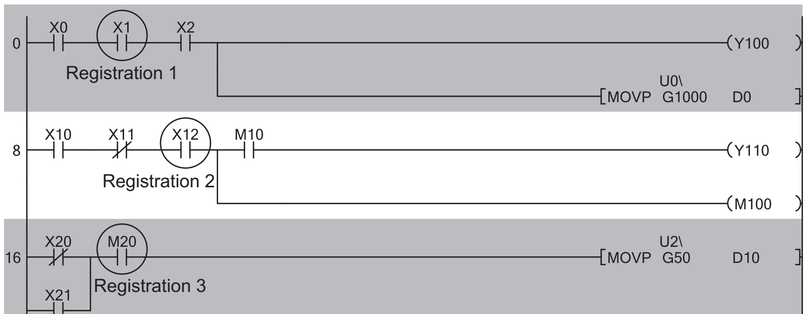
For the online module change (ladder): if any executional conditioned device test setting has been registered in the ladder block that is to be changed online, the CPU module disables the corresponding setting.

Ex. Step numbers of registrations 1 to 3 are specified in the executional conditioned device test settings. When the ladder block including the registration 2 is changed online, the registration 2 is disabled. Since the registrations 1 and 3 are not included in the change target program, they are not disabled.



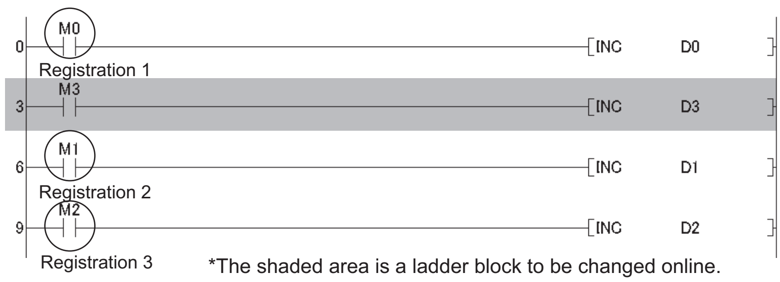
*The shaded area is a ladder block to be changed online.

Ex. When multiple ladder blocks are to be changed online, ladder blocks between the change target ladder blocks are included in the change target. In the following example, registrations 1 to 3 are all disabled.



*The shaded areas are ladder blocks to be changed online.

Ex. When a ladder block is to be added online, the executional conditioned device test setting included in the ladder block followed after the added ladder block is disabled.



*The shaded area is a ladder block to be changed online.

For the online change (files), all executional conditioned device test settings registered to the target program are disabled.

(d) Specifying a device by index modification

If an index-modified device name is specified to register the executional conditioned device test setting, the CPU module does not check whether the specified device is within the setting range. To change a device by specifying a step No., the index-modified device must be within the device range or not be on the boundary of devices, otherwise the device value is not changed.

(e) Specifying a device by indirect specification

If an indirectly-specified device name is specified to register the executional conditioned device test setting, the CPU module does not check whether the specified device is within the setting range. To change a device by specifying a step No., the index-modified device must not be on the boundary of devices, otherwise the device value is not changed.

(f) Specifying the file register

If the file register is specified to register the executional conditioned device test setting, the CPU module does not check the file register file assignment and the file register number range. A file register value is not changed within the specified step in the following cases.

- The file register file is not assigned.
- The specified file register number is out of the file register range.

3.20 Sampling Trace

This function samples the data of the specified device at a preset timing and at a preset interval (sampling cycle), and then stores the trace results in the sampling trace file. In addition, this function can be used to read the device data upon trigger condition establishment. This function is useful to check the change of the device data used in the program during debugging at a preset timing.

(1) Sampling trace file

This file stores the trace setting necessary for executing the function and trace results. Sampling trace file can be stored only in the Standard RAM.

(2) Devices that the sampling trace function is applicable

The following devices can be set up to 50 points.

Type	Description
Bit device:	X (DX), Y (DY), M, L, F, SM, V, B, SB, T (contact), T (coil), ST (contact), ST (coil), C (contact), C (coil), FX, FY, BL□\S, J□\X, J□\Y, J□\SB
Word device:	T (current value), ST (current value), C (current value), D, D (extended data register), SD, W, W (extended link register), SW, R, Z, ZR, FD, U□\G, J□\W, J□\SW

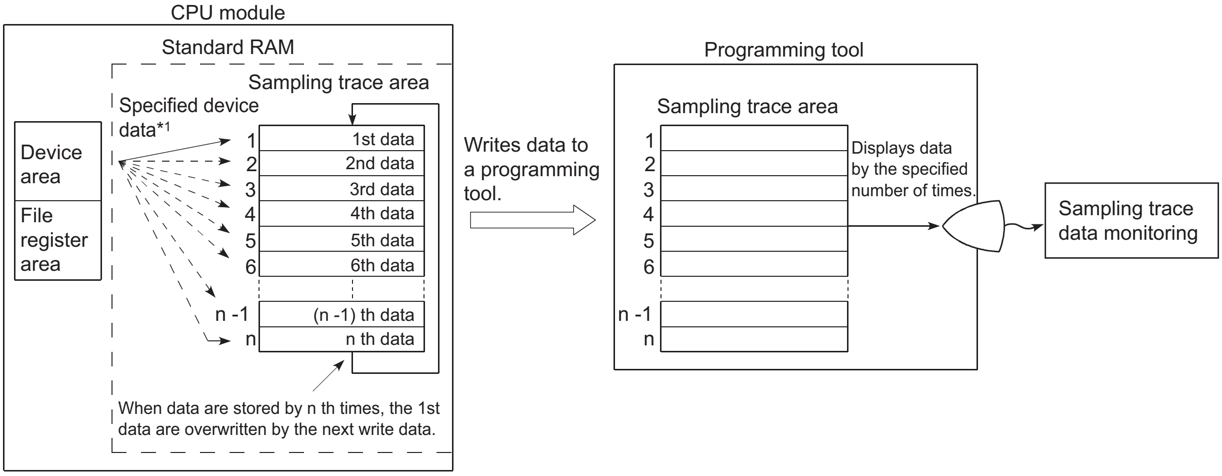
The following modifications are available for the above devices.

- Digit specification of bit device
- Bit specification of word device
- Index modification

(3) Sampling trace operation

(a) Operating of the CPU module

When a sampling trace trigger is issued, the CPU module executes traces for the preset number of times.



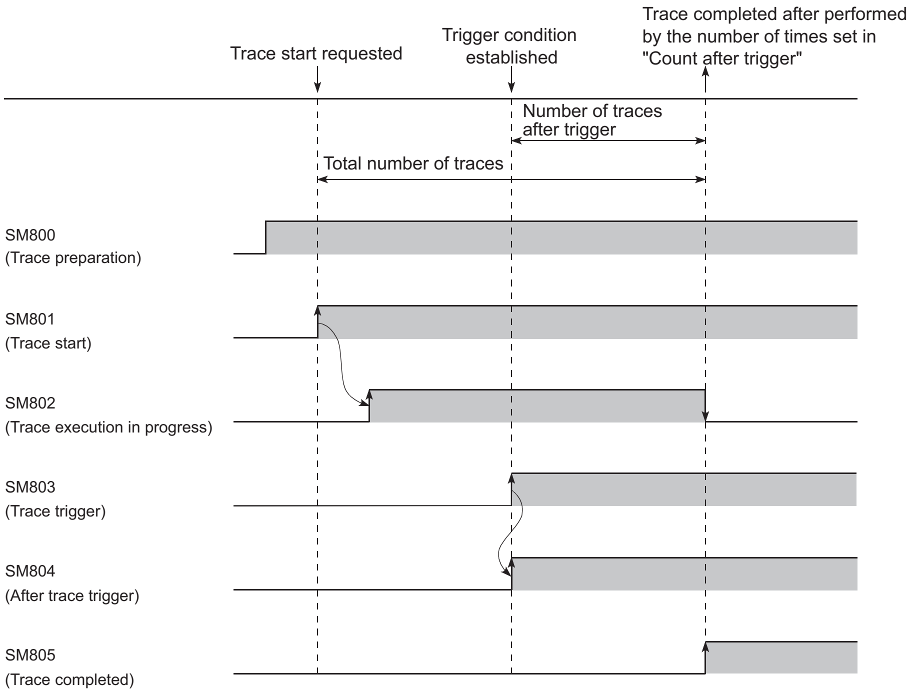
*1 When the trigger is issued, the CPU module samples data for the preset number of times and latches the data in the sampling trace area.

3

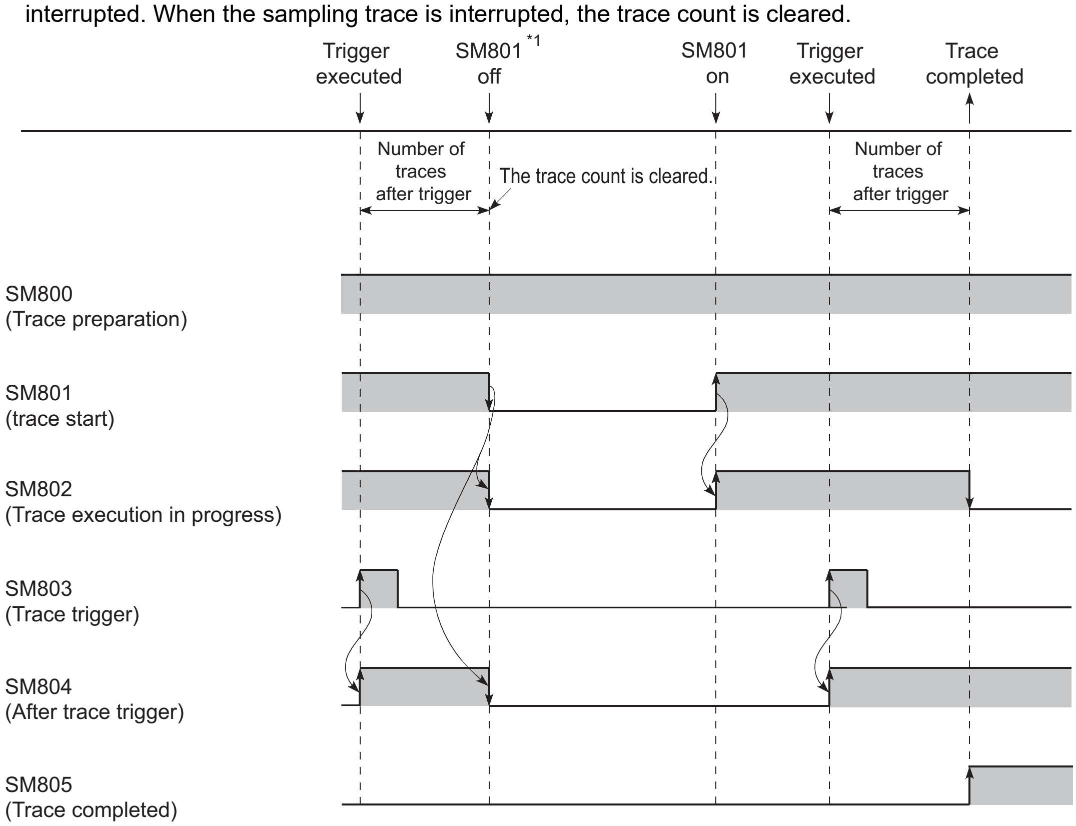
3.20 Sampling Trace

(b) On/Off of the special relay

The execution status of the sampling trace can be checked by the special relay.



Also, if SM801 (Trace start) is turned off during sampling trace, execution of the sampling trace will be interrupted. When the sampling trace is interrupted, the trace count is cleared.



*1 SM801 also turns off when the sampling trace is interrupted by a programming tool.

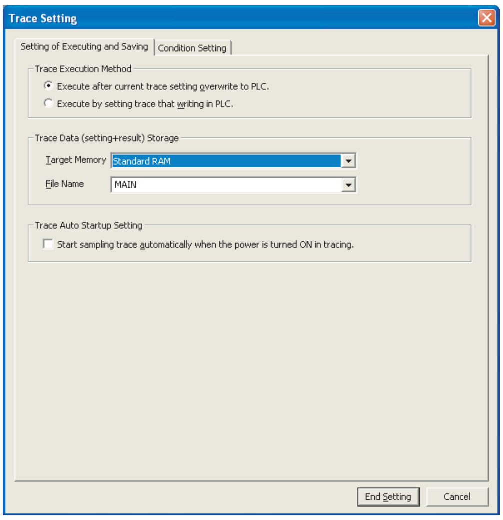
(4) Setting method

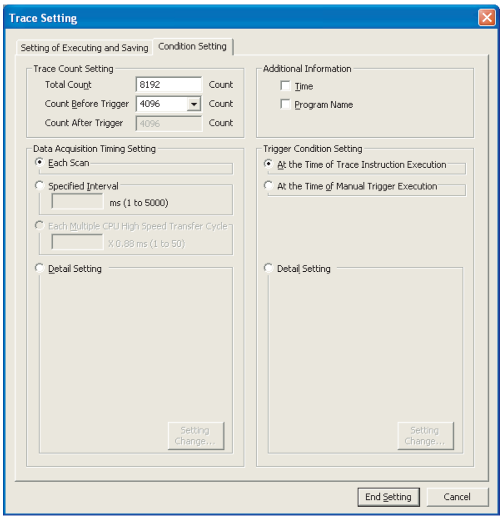
1. Open the "Sampling Trace" dialog box.

[Debug] ⇨ [Sampling Trace] ⇨ [Open Sampling Trace]

2. On the "Sampling Trace" window, open a dialog box to configure a setting.

[Debug] ⇨ [Sampling Trace] ⇨ [Trace Setting...]





Item			Description	Reference
Trace Setting	Setting of Executing and Saving	Trace Execution Method	Set the execution method of the trace.	(a)
		Trace Data (setting + result) Storage	Select a file name of the trace data.	—
		Trace Auto Startup Setting	Set whether to start the sampling trace automatically when the CPU module is powered off and then on.	—
	Condition Setting	Trace Count Setting	Set the number of traces to perform.	(b)
		Additional Information	Set whether to add clock data and program name to the sampling trace file.	—
		Data Acquisition Timing Setting	Set the timing of data acquisition.	(c)
		Trigger Condition Setting	Select the condition to activate a trigger.	(d)

(a) Trace Execution Method

- Execute after current trace setting overwrite to PLC:

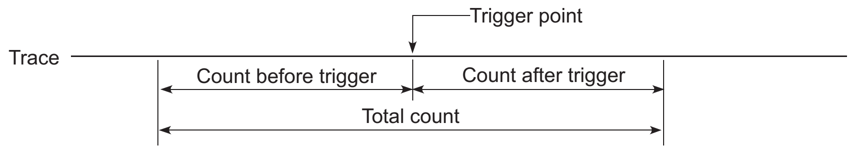
The CPU module executes the sampling trace after the trace settings are overwritten to the existing sampling trace file.

- Execute by setting trace that writing in PLC:

The CPU module executes the sampling trace with the trace settings in the sampling trace file selected for storage.

(b) Trace Count Setting

- Total Count: The number of times that data are stored in the memory
- Count Before Trigger: The number of data storages to retain before the trigger
- Count After Trigger: A value obtained by the "Count Before Trigger" value subtracted from the "Total Count" value.



(c) Data Acquisition Timing Setting

Set the timing for collecting trace data.

Item	Description
Each Scan	Collects trace data during END processing of each scan.
Specified Interval *1	Collects trace data at specified time intervals.
Detail Setting	Specify a device or label from the following. • Bit device: X (DX), Y (DY), M, L, F, SM, V, B, SB, T (contact), ST (contact), C (contact), FX, FY, BL□\S, J□\X, J□\Y, J□\SB • Word device: T (current value), ST (current value), C (current value), D, D (extended data register), SD, W, W (extended link register), SW, R, Z, ZR, FD, U□\G, J□\W, J□\SW The following modifications are available for the above devices. • Digit specification of bit device • Bit specification of word device • Indirect specification of word device • Index modification When the set conditions are met, data collection is performed.

*1 Pay attention to the sampling interval and sampling processing time for one sampling since the sampling trace is performed as interrupt processing. If the sampling processing time for one sampling is long, "WDT ERROR" may occur.

(d) Trigger Condition Setting

Select the trigger point.

Item	Description
At the Time of Trace Instruction Execution	The time of the TRACE instruction execution is regarded as a trigger.
At the Time of Manual Trigger Execution	The time of trigger execution from the programming tool is regarded as a trigger.
Detail Setting	Specify a device or label from the following. • Bit device: X (DX), Y (DY), M, L, F, SM, V, B, SB, T (contact), ST (contact), C (contact), FX, FY • Word device: T (current value), ST (current value), C (current value), D, D (extended data register), SD, W, W (extended link register), SW, R, ZR The following modifications are available for the above devices. • Bit specification of word device When the set conditions are met, this timing is set as a trigger point.

Point

GX Developer allows the specification of a step No. for the data acquisition timing and the trigger condition. (The acquisition timing and the trigger point are the timing that the state just before the execution of a specified step becomes to a specified state.) Note however that with a step No. specified, the processing time of sampling trace is added into the program execution time. The watchdog timer settings and the interrupt programs therefore should be designed with a consideration of the processing time of sampling trace (setting of the fixed scan interval).

(5) Online operation of trace data

Before execution of a sampling trace, write the created trace setting to the CPU module.

[Debug] ⇨ [Sampling Trace] ⇨ [Write to PLC...]

The trace data written to the CPU module can be read.

[Debug] ⇨ [Sampling Trace] ⇨ [Read from PLC...]

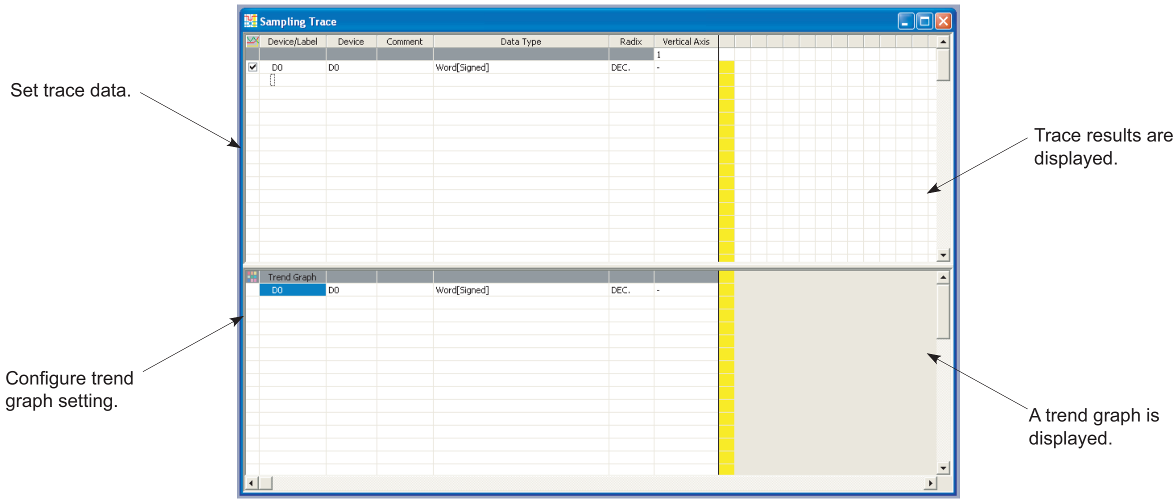
(6) Executing a sampling trace

The following describes how to execute from a programming tool.

(a) Start

1. On the "Sampling Trace" window, enter devices to trace.

[Debug] ⇨ [Sampling Trace] ⇨ [Open Sampling Trace]



2. Select "Start Trace".

[Debug] ⇨ [Sampling Trace] ⇨ [Start Trace]

(b) Stop

When a trace is stopped, the number of traces counted is cleared.
(To resume the trace, select "Start Trace" again.)

[Debug] ⇨ [Sampling Trace] ⇨ [Stop Trace]

To clear the execution status, perform a latch clear operation. (☞ Page 89, Section 3.4 (4) (a))
To perform the trace operation again after the latch clear operation, select "Start Trace".

(c) Execution of a trigger

After completing a trace, execute a trigger.

[Debug] ⇨ [Sampling Trace] ⇨ [Execute Manual Trigger]

After a trigger is completed, trace results are displayed on "Sampling Trace" window.

Point

The sampling trace can be performed from other stations in the network or serial communication modules. Note that these functions cannot be simultaneously performed from several sites.

(7) Precautions

(a) Holding and clearing the trace setting

The trace setting (sampling trace file) registered with the CPU module is latched. Even if the CPU module is powered off and then on or is reset, the sampling trace can be performed again with the trace setting at registration.

However, the previous trace result cannot be read. Also in the following cases, even when the trigger condition is established, the condition is not recognized. In these cases, the trace setting must be registered again since the latched trace setting is cleared (SM800 turns off).

- With the size of a local device in the standard RAM set to be changed, the parameters have been written to the CPU module and the CPU module is powered off and then on or it is set from STOP to RUN.
- With a sampling trace file corrupted, the CPU module is powered off and then on or it is reset.

(b) Reading trace result

The trace result cannot be read while the CPU module is in the STOP status.

(c) Registering sampling trace when the trigger condition is established.

Even if the trigger condition is established, the sampling trace setting can be registered by the following procedure.

1. **Turn on SM829 (Forced registration specification of trace setting).**
2. **Enable the forced execution registration.**

[Debug] ⇨ [Sampling Trace] ⇨ [Forced Execution Registration Effective...]

For the above cases, start the trace in the status where the trigger condition is not established. With a trigger condition established, the trigger may not be normally executed.

(d) Specifying a file register

When a file register is selected as a specified device by the detail setting of trace setting, do not change the block numbers of file register file and file register after trace registration. Trace data may not be normally sampled.

(e) Performing sampling trace during execution of another sampling trace

The first sampling trace is performed normally. The second sampling trace cannot be performed.

(f) Performing online change during execution of sampling trace

The trace point or trigger point is specified by the step number: The sampling trace is suspended but the online change is normally performed. (If neither of them is specified other than by step No., both the online change and sampling trace can be performed.)

(g) Performing sampling trace during online change

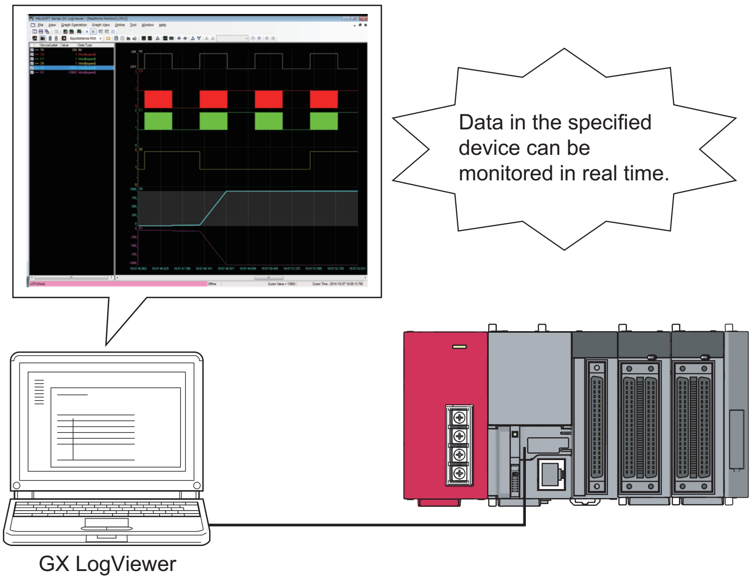
The trace point or trigger point is specified by the step number: The online change is completed normally but the sampling trace is not performed. (If neither of them is specified other than by step No., both the online change and sampling trace can be performed.)

(h) Latch clear by using the special relay and special register areas during execution of sampling trace

The latch clear processing will be performed, but the sampling trace processing stops.

3.21 Realtime Monitor Function

This function monitors the data in the specified device of the CPU module at a specified interval or at a desired timing in real time.
The function can be set with GX LogViewer, where the value changes of a specified device can be shown graphically.
Saving the set data and displayed graphs makes it possible to simplify the settings and check the graphs at a later time.



Remark

- Before executing this function, check the versions of the CPU module and GX LogViewer used. (☞ Page 376, Appendix 2)
- The L02SCPU and L02SCPU-P do not support this function.
- For details on the realtime monitor function, refer to the following.
  📖GX LogViewer Version 1 Operating Manual

3.22 Writing Programs in RUN Status

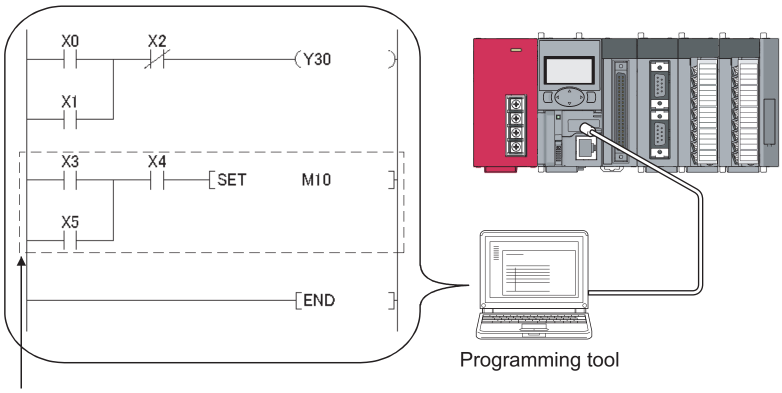
There are two types of writing programs to the CPU module while it is in the RUN status.

- Online change (ladder mode) (☞ Page 151, Section 3.22.1)
- Online change (files) (☞ Page 152, Section 3.22.2)

To perform from multiple programming tools, use a pointer so that data are relatively written to the CPU module. (☞ Page 152, Section 3.22.2)

3.22.1 Online change (ladder mode)

Programs are written by a ladder block unit.



A program is written in units of ladder blocks.

Online change can be performed only for the program memory (program cache memory).
Up to 512 steps can be batch-written.

(1) The reserved area for online change

A program file has an area designated as reserved area for online change to support the one that changes number of steps. The change in the number of steps does not affect the program file size if the change is within the reserved area for online change. If more steps are required for the reserved area for online change during performing an online change, change the setting.

3.22.2 Online change (files)

The files listed in the following table are batch-written to the CPU module.

○: Can be written, △: Cannot be written while the file is being accessed in the program, ×: Cannot be written

File name	Program memory	Standard RAM	Standard ROM	SD memory card
Parameter	×	×	×	×
Intelligent function module parameter	×	×	×	×
Program *1	○	×	○	○
Device comment	○	×	△	△
Initial device value	×	×	×	×
File register	×	△	×	△
Local device	×	×	×	×
Sampling trace file	×	○	×	○
Programmable controller user data	×	×	○	○

*1 Online change (files) of SFC programs cannot be performed.

A file can be written to the CPU module in the RUN status regardless of space of a memory to be written and the number of files to be stored.

(1) Online change (files) from multiple locations

Do not simultaneously write files to one CPU module in the RUN status from multiple locations. Doing so may delete program files.

3.22.3 Precautions for online change

(1) Effect on the scan time

Performing change increases scan time. (☞ Page 385, Appendix 3.2)

(2) Online change during boot operation

Even if the online change is executed, the status of boot source program is not changed.

(3) Operations prohibited during online change

Do not perform power-off or reset to the CPU module when changing the TC setting value or transferring data to the program memory during online change. If performed, write the data to the CPU module again.

An error is displayed if any of the following operation is performed. Execute again after completing the operation.

- Online change (ladder mode), online change (files)
- TC setting value change
- Data transfer to the program memory

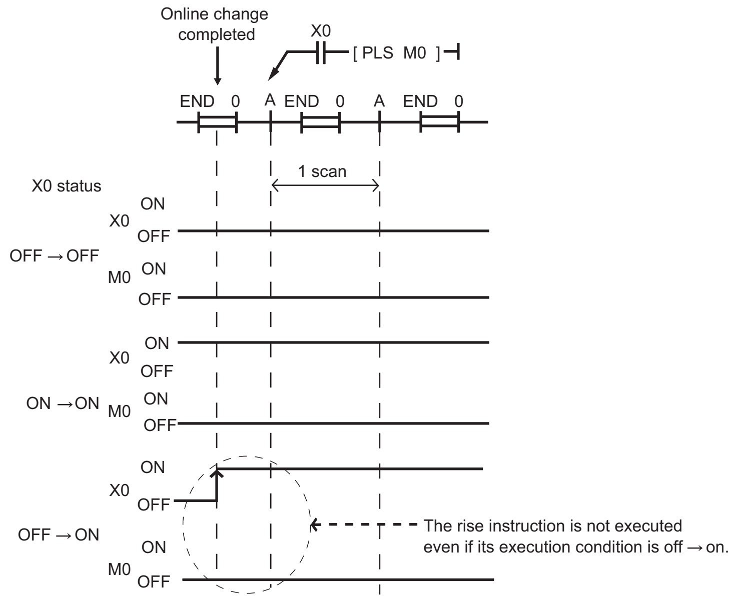
(4) Instructions do not operate normally during online change

If the following instructions are executed during online change, they do not operate normally.

- Rise instruction
- SCJ instruction
- STMR instruction

(a) Rise instruction

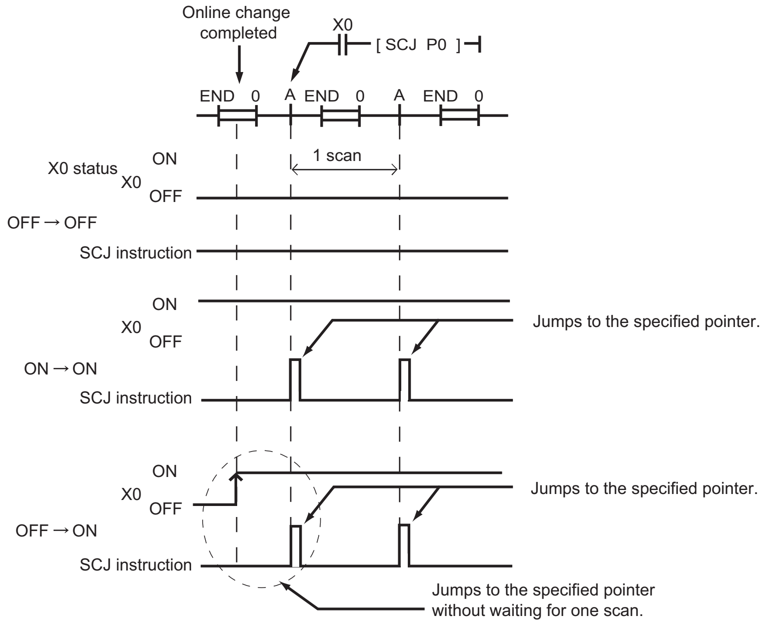
The rise instruction within the program targeted for online change will not be executed even though the execution condition of the instruction (off → on) is met at the completion of online change.



The corresponding rise instructions are PLS and □P.

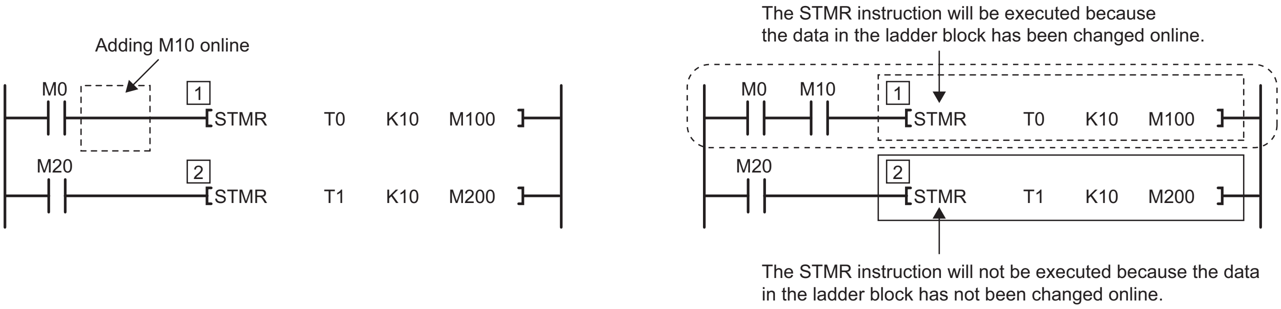
(b) SCJ instruction

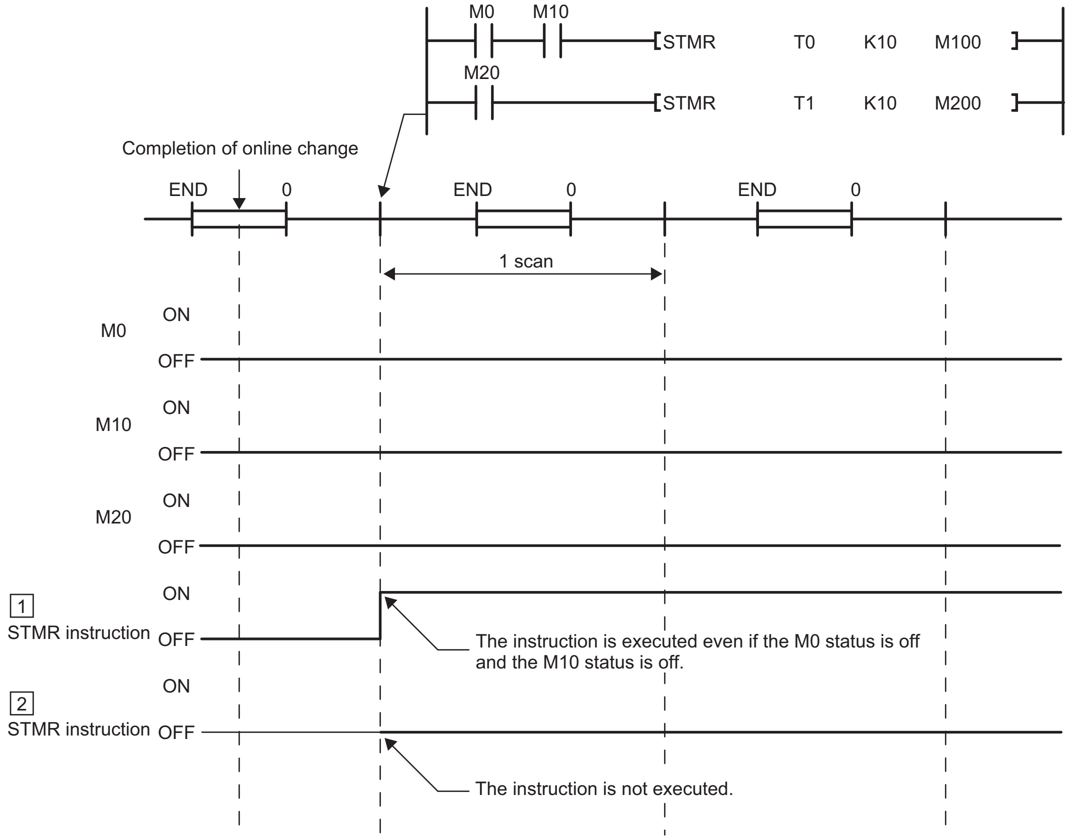
When the SCJ instruction is used within the program targeted for online change and the execution condition of the instruction is on at the completion of online change, the program jumps to the specified pointer without waiting for one scan.



(c) STMR instruction

Note that when the STMR instruction is used within the program targeted for online change, the instruction is executed.

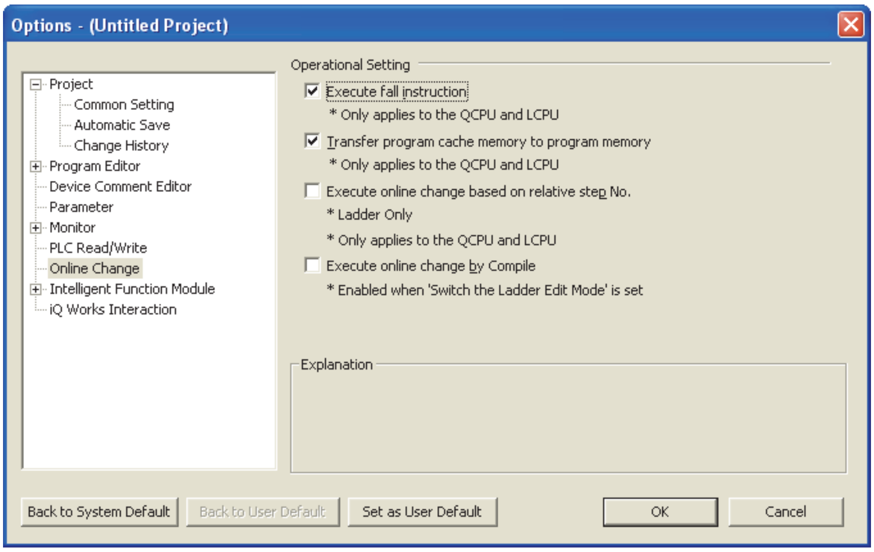




Point

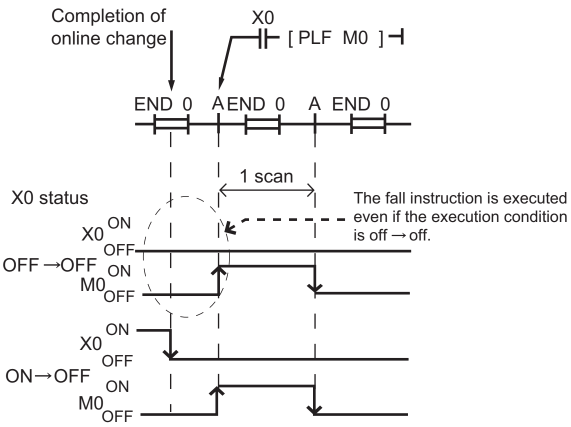
With "Execute fall instruction" selected for in the Options dialog box of the programming tool, the fall instruction is executed when the instruction is in the data written to the CPU module in the RUN status, even if the execution condition (on → off) is not met. (The same operation as High Performance model QCPU)

[Tool] ⇨ [Options...]

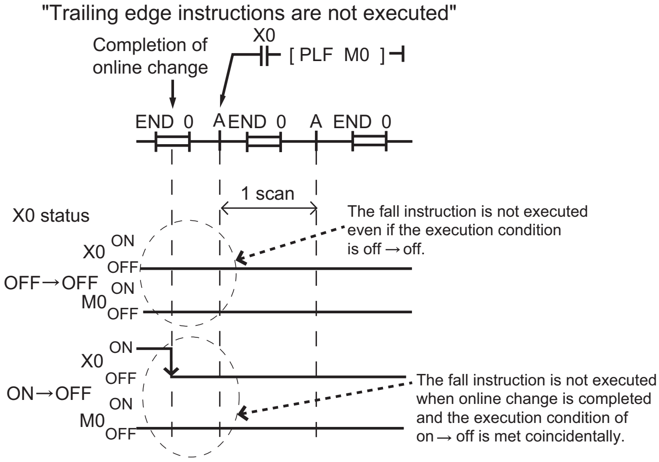


The corresponding fall instructions are LDF, ANDF, ORF, MEF, PLF, FCALLP, and EFCALLP. The following describes the operation with and without "Execute fall instruction" selected.

Operation when selecting
"Trailing edge instructions are not executed"



Operation when deselecting
"Trailing edge instructions are not executed"



(5) Automatic data transfer to the program memory

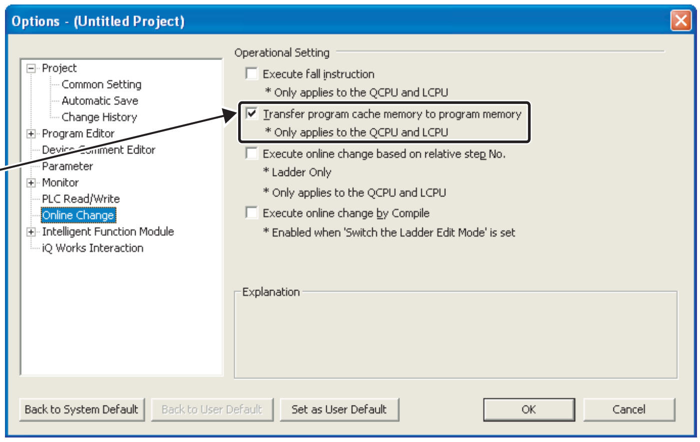
The automatic data transfer to the program memory requires time obtained from the following formula.

- L02SCPU, L02SCPU-P: (Scan time (s)) × 120.0 + 1.2 (s)
- L02CPU, L02CPU-P: (Scan time (s)) × 170.0 + 1.0 (s)
- L06CPU, L06CPU-P: (Scan time (s)) × 260 + 4.7 (s)
- L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT: (Scan time (s)) × 1100 + 15.0 (s)

The number of writes to the program memory (flash ROM) is limited to 100,000 times. When data are written to the CPU module in the RUN status or the T/C setting values are changed frequently, disable the automatic transfer to the program memory.

[Tool] ⇨ [Options...]

To avoid automatic transfer of program memory data, clear the checkbox. (Selected by default.)



When the automatic transfer is set to be disabled, the following message appears after online change. When selecting "No", the program memory must be batch transferred by a programming tool.

[Online] ⇨ [Program Memory Batch Download]

The status of the transfer to the program memory can be confirmed by SM165.

3.23 Debug from Multiple Programming Tools

This function allows debugging from multiple programming tools connected to a module such as a CPU module or serial communication module. The following table shows combinations of the debug function executable from multiple programming tools.

○: Can be simultaneously performed, △: Can be simultaneously performed but partially restricted, ×: Can not be simultaneously performed

Function in execution	Function executed later					
	Monitor	Program monitor list, Interrupt program monitor list	Online change	Scan time measurement	Sampling trace	Executional conditioned device test
Monitor*1	○	○	○	○	○	○
Program monitor list, Interrupt program monitor list	○	△*5	○	○	○	○
Online change	○	○	×*2*4	×*2	×*2	×*2
Scan time measurement	○	○	×*3	×*2	○	○
Sampling trace	○	○	△*6	○	×*2	○
Executional conditioned device test	○	○	△*7	○	×*2	○

*1 The Monitor in the above table indicates the following.
· Ladder monitor
· Entry ladder monitor
· Device block monitor
· Entry data monitor
· Local device monitor

*2 Only the function in execution is operated. Another function intended later cannot be executed.

*3 The function in execution stops and another function intended later is operated.

*4 To perform online change to one file from multiple programming tools, refer to ☞ Page 160, Section 3.23.2.

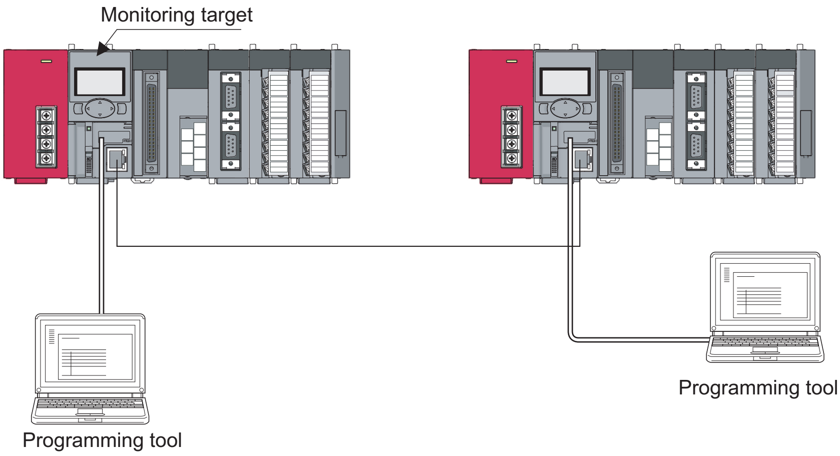
*5 The one in execution and the one intended later are different functions, these functions can be simultaneously performed. The one in execution and the one intended later are the same function, the later one is not executed.

*6 If the condition other than for step No. is set as a trace point or trigger point, these functions can be performed simultaneously. If a trace point or trigger point is specified for the step No., the one in execution stops and the one intended later is performed.

*7 The functions cannot be simultaneously performed in the following cases. In any of the following cases, the function in execution stops and another function intended later is operated.
· The data to be changed online includes the registration of an executional conditioned device test.
· When adding a ladder block by online change, registration of an executional conditioned device test is included in the ladder block immediately after the one where the ladder block is to be added.
· The program to be changed online includes registration of an executional conditioned device test.

3.23.1 Simultaneous monitoring from multiple programming tools

Creating a user setting system area allows high-speed monitoring from multiple programming tools (Setting a monitoring file for the host station is not required).

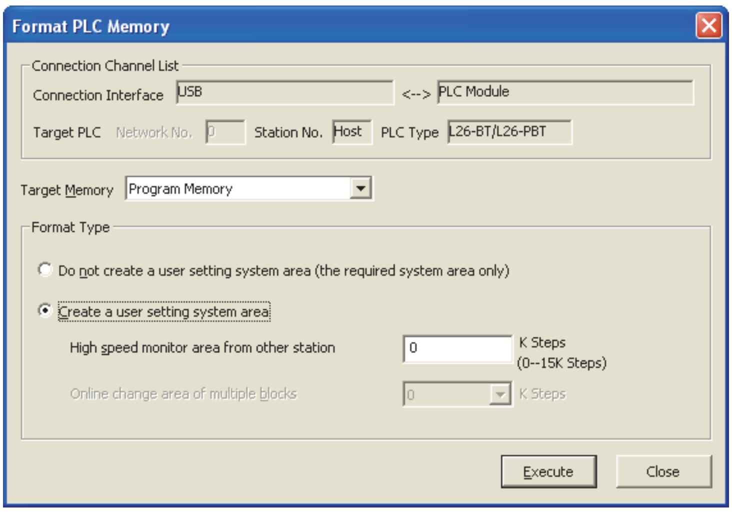


However, since the system area is stored in the program memory, the storage area is reduced by the system area size.

(1) Creating a user setting system file

1. Open the "Format PLC Memory" dialog box.

[Online] ⇨ [PLC Memory Operation] ⇨ [Format PLC Memory]



2. Select "Program memory" in "Target Memory".

3. Select "Create a user setting system area" in the "Format Type" area.

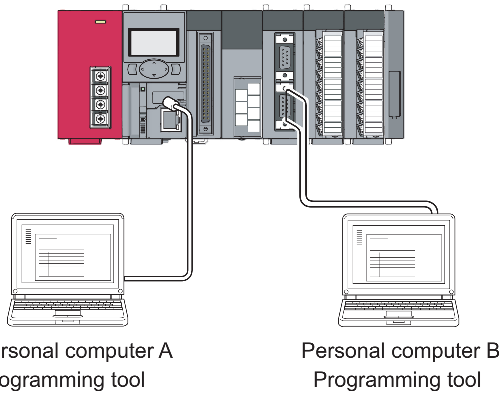
The number of programming tools that can simultaneously monitor a CPU module is the number of user setting system areas +1.

4. Set the number of steps for the system area (in increments of: 1K step).

Up to 15K steps can be set for the system area. 1K step is available for a monitoring file from another station.

3.23.2 Online change from multiple programming tools

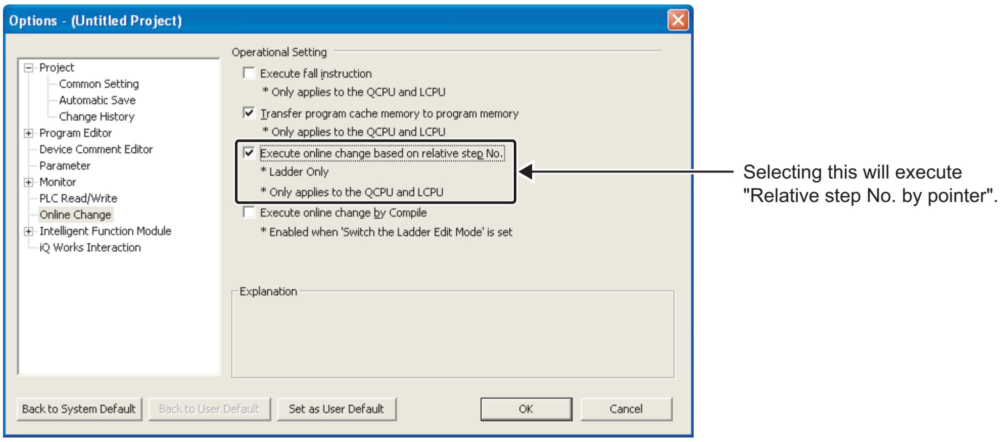
To perform online change from multiple programming tools, select "Execute online change based on relative step No.".



(1) Online change based on relative step No.

Open the "Options" dialog box.

[Tool] ⇨ [Options...]



Display the program including the specified pointer and write the changed program during RUN.

Remark

Precautions for online change from multiple programming tools are the same as those for standard online change. (☞ Page 153, Section 3.22.3)

3.24 Self-Diagnostic Function

This function allows the CPU module to diagnose itself to check for errors. This function aims to preventive measures and prevention of malfunction of the CPU module.

(1) Self-diagnostic timing

When an error occurs at power-on or during the RUN or STOP status of the CPU module, the error is detected and displayed by the self-diagnostic function, and the CPU module stops an operation. Note that errors cannot be detected by the function depending on error status or an instruction executed. When the operation is not stopped by the function, configure a safety circuit external to the programmable controller so that the entire system operates safely.

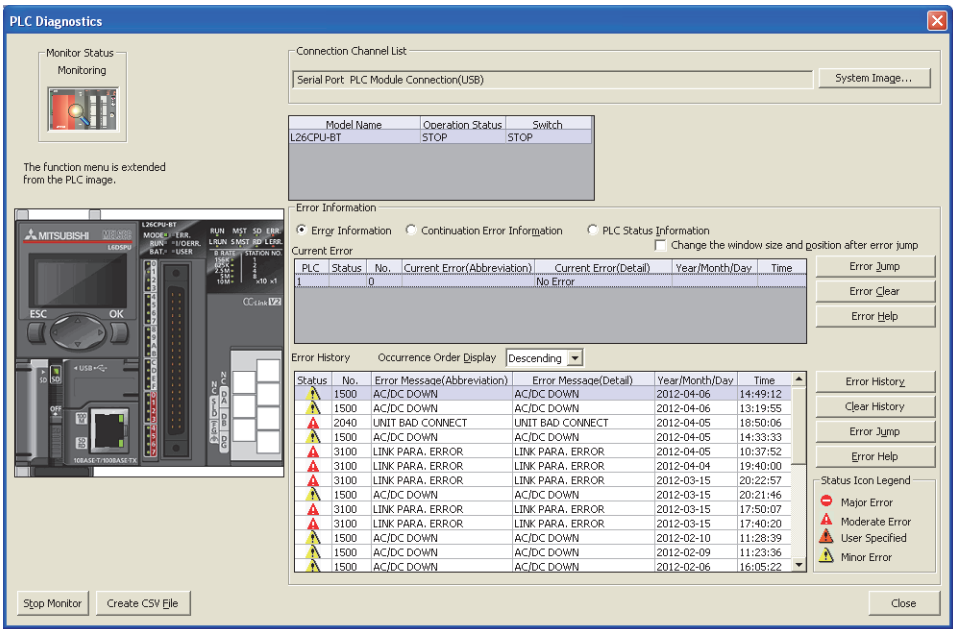
(2) Storage location of error information and error check

When the CPU module detects an error, SM0 and SM1 turn on and the error information (error code) are stored in SD0. When several errors are detected, the latest error code is stored in SD0. Use SM0, SM1, or SD0 in a program as an interlock for the programmable controller and mechanical system. Also, it can be checked by the on status of the ERR. LED.

(3) Checking error history

The latest error history (error description) can be checked in the "Error history" area.

[Diagnostics] ⇨ [PLC Diagnostics]



Errors can also be checked by the display unit. (Page 241, Section 4.2.1)

Up to 100 module errors*2 can be stored in the system memory*1.

*1 The memory is internally controlled by the system.
*2 Once the memory is full, the oldest stored log is deleted to store a new log.

To clear the module error data, click the [Clear History] button.

(4) CPU module operation at error detection

When an error is detected by the self-diagnostic function, the CPU module operates according to the specified operation mode.

- Mode that stops CPU module operation ("Stop"):

The output mode setting in the detailed setting (from the "I/O Assignment" tab)	CPU module operation
Clear	When an error is detected, the CPU module stops an operation and turns off all external outputs of the module. (Outputs (Y) of the device memory are held.)
Hold	When an error is detected, the CPU module stops an operation and holds external outputs of the module. (Outputs (Y) of the device memory are held.)

- Mode that continues CPU module operation ("Continue"):

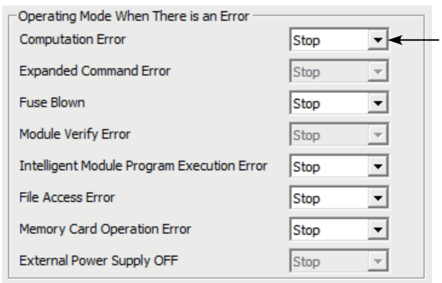
When an error is detected, the CPU module executes programs other than the one (instruction) where an error occurred.

(a) Errors selectable in the PLC RAS tab

- Computation Error (including SFC program)
- Intelligent Module Program Execution Error
- Memory Card Operation Error
- Fuse Blown
- File Access Error

Set the operation mode for the case of an error.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC RAS]



(b) Errors selectable in the I/O Assignment tab

- Intelligent function module error (☞ Page 103, Section 3.9)

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [I/O Assignment] ⇨ [Detailed Setting]

(5) Error check options

Whether to check the following errors can be selected. (All the items are selected by default.)

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC RAS]

- Carry Out Battery Check
- Carry Out Fuse Blown Check
- Check Device Range at Indexing

(6) Self-diagnostics list

○: Self-diagnostics is performed. ×: Self-diagnostics is not performed.

Diagnostics		Error message	Diagnostic timing	CPU module status	LED status		Availability of self-diagnostics
					RUN	ERR.	
Hardware failure	CPU error	CPU UNIT DOWN	• Always	Stop	Off	Flashing	○
	END instruction not executed	END NOT EXECUTE	• Execution of the END instruction	Stop	Off	Flashing	○
	SFC program execution error	SFCP. END ERROR	• Execution of a SFC program	Stop	Off	Flashing	○
	RAM check	RAM ERROR	• Power-on/reset	Stop	Off	Flashing	○
	Operation circuit check	OPE.CIRCUIT ERR.	• Power-on/reset • Execution of the END instruction	Stop	Off	Flashing	○
	I/O interrupt error	I/O INT. ERROR	• Occurrence of an interrupt	Stop	Off	Flashing	○
	LAN controller failure	LAN CTRL. DOWN	• Power-on/reset	Stop	Off	Flashing	○[*3]
	Intelligent function module error[*1]	SP.UNIT DOWN	• Power-on/reset • Execution of the FROM/TO instructions • Execution of the intelligent function module dedicated instruction • Execution of the END instruction	Stop/Continue	Off/On	Flashing /On	○
	System bus error	• BUS TIMEOUT ERR. • UNIT BUS ERR. • SYSTEM RST ERR.	• Powered-on • Execution of END processing • Execution of the FROM/TO instructions • Execution of the intelligent function module dedicated instruction • Always	Stop	Off	Flashing	○
	End cover error	END COVER ERR.	• Powered-on • Execution of END processing • Always	Stop	Off	Flashing	○
	Momentary power failure	AC/DC DOWN	• Always	Continue	On	Off	○
	Flash ROM error	FLASH ROM ERROR	• Writing to ROM	Continue	On	On	○
Handling error	Module verification	UNIT BAD CONNECT	• Execution of the END instruction	Stop	Off/On	Flashing /On	○
	Intelligent function module assignment error	SP.UNIT LAY ERR.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
	Intelligent module program execution error[*1]	SP.UNIT ERROR	• Execution of the FROM/TO instructions	Stop/Continue	Off/On	Flashing /On	○
	Intelligent function module version error	SP.UNIT VER.ERR	• Power-on/reset	Stop	Off	Flashing	○
	END cover is not attached	NO END COVER	• Power-on/reset • Execution of the END instruction	Stop	Off	Flashing	○
	Unrecognizable module connection error	SYSTEM LAY ERR	• Power-on/reset	Stop	Off	Flashing	○
	No parameter	MISSING PARA.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
	Boot error	BOOT ERROR	• Power-on/reset	Stop	Off	Flashing	○

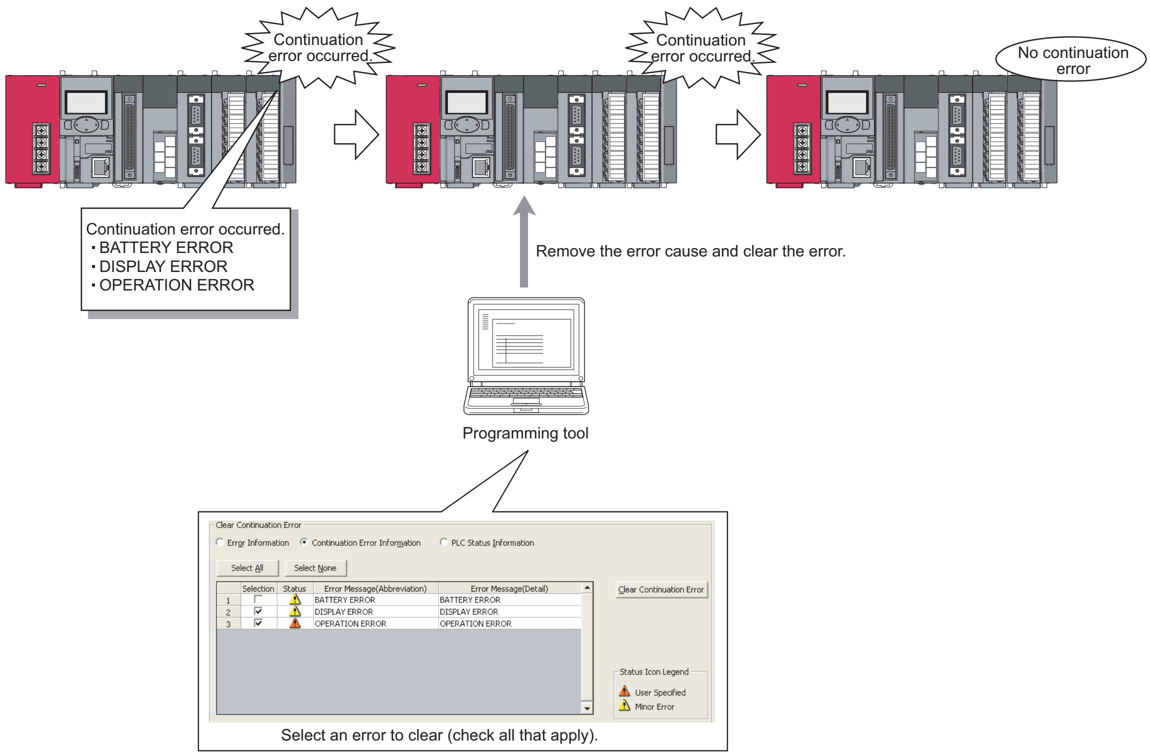
Diagnostics		Error message	Diagnostic timing	CPU module status	LED status		Availability of self-diagnostics
					RUN	ERR.	
Handling error	Backup error	RESTORE ERROR	• Power-on/reset	Stop	Off	Flashing	○
	Memory card operation error[*1]	ICM.OPE. ERROR	• Insertion/removal of the memory card	Stop/Continue	Off/On	Flashing /On	×
	File setting error	FILE SET ERROR	• Power-on/reset • Writing to programmable controller	Stop	Off	Flashing	○
	File access error[*1]	FILE OPE. ERROR	• Instruction execution	Stop/Continue	Off/On	Flashing /On	○
	Instruction execution disabled	CAN'T EXE.PRG.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
Parameter error	Parameter setting check	PARAMETER ERROR	• Power-on/reset • Switching from STOP to RUN • Writing to programmable controller	Stop	Off	Flashing	○
	SFC parameter error	SFC PARA.ERROR	• Switching from STOP to RUN • Writing to programmable controller	Stop	Off	Flashing	○
	Intelligent function module parameter error	SP.PARA. ERROR	• Power-on/reset	Stop	Off	Flashing	○
Password error		REMOTE PASS.ERR	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
Instruction code check		INSTRUCT. CODE ERR	• Power-on/reset • Switching from STOP to RUN • Instruction execution	Stop	Off	Flashing	○
No END instruction		MISSING END INS.	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
Pointer setting error		CAN'T SET(P)	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
		CAN'T SET(I)	• Power-on/reset • Switching from STOP to RUN	Stop	Off	Flashing	○
Program error	Operation error[*1*2]	OPERATION ERROR	• Instruction execution	Stop/Continue	Off/On	Flashing /On	○
	FOR to NEXT instructions structure error	FOR NEXT ERROR	• Instruction execution	Stop	Off	Flashing	○
	CALL to RET instructions structure error	CAN'T EXECUTE(P)	• Instruction execution	Stop	Off	Flashing	○
	Interrupt program error	CAN'T EXECUTE(I)	• Instruction execution	Stop	Off	Flashing	○
	Instruction execution disabled	INST. FORMAT ERR.	• Instruction execution	Stop	Off	Flashing	○
	SFC block configuration error	CAN'T SET(BL)	• Switching from STOP to RUN	Stop	Off	Flashing	○
	SFC step configuration error	CAN'T SET(S)	• Switching from STOP to RUN	Stop	Off	Flashing	○
	SFC execution error	SFC EXE. ERROR	• Switching from STOP to RUN	Stop	Off	Flashing	○
	SFC syntax error	SFCP. FORMAT ERR.	• Switching from STOP to RUN	Stop	Off	Flashing	○
	SFC block execution error	BLOCK EXE.ERROR	• Instruction execution	Stop	Off	Flashing	○
	SFC step execution error	STEP EXE.ERROR	• Instruction execution	Stop	Off	Flashing	○

Diagnostics		Error message	Diagnostic timing	CPU module status	LED status		Availability of self-diagnostics
					RUN	ERR.	
CPU error	Watchdog error supervision	WDT ERROR	• Always	Stop	Off	Flashing	○
	Program timeout	PRG.TIME OVER	• Always	Continue	On	On	○
File diagnostic check		INCORRECT FILE	• Power-on/reset • Switching from STOP to RUN • Writing to programmable controller	Stop	Off	Off	○
Annunciator check		F****	• Instruction execution	Continue	On	USER LED On	○

*1 The operating status can be changed to "Continue" with the parameter setting. ("Stop" is set by default.)
*2 This error includes an operation error when a device range is checked at index modification.
*3 For Built-in Ethernet port LCPU only.

3.25 Error Clear

Continuation errors can be selected by user to clear them by error type.



(1) Errors that can be cleared

Only the errors shown below can be cleared.

- SP.UNIT DOWN
- FLASH ROM ERROR
- FILE OPE.ERROR
- DISPLAY ERROR
- F*** (Annunciator)
- AC/DC DOWN
- SP.UNIT ERROR
- REMOTE PASS.FAIL
- OPERATION ERROR
- PID ERROR
- BATTERY ERROR
- ICM.OPE.ERROR
- SNTP OPE.ERROR
- PRG.TIME OVER

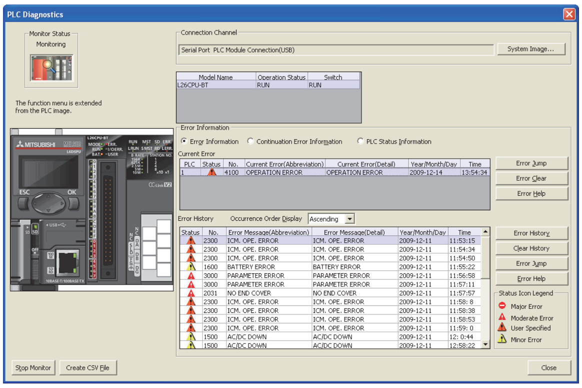
(2) Clearing errors

To clear errors by error type, use any of the following.

- Programming tool
- SM and SD
- Display unit

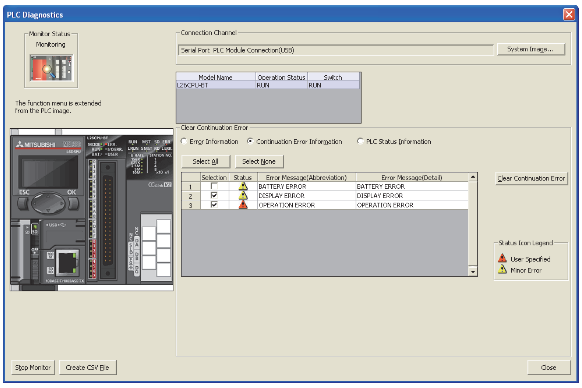
(a) Programming tool

Perform the following procedure.

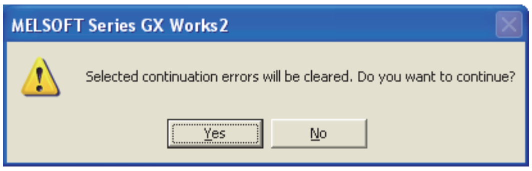


1. **On the "PLC Diagnostics" dialog box, check the continuation errors that have been detected.**

2. **Remove the error cause of the continuation errors.**



3. **Select the "Continuation Error Information" radio button and checkboxes of the errors to clear, and click the** Clear Continuation Error **button.**



4. **Click "Yes" to clear the error.**

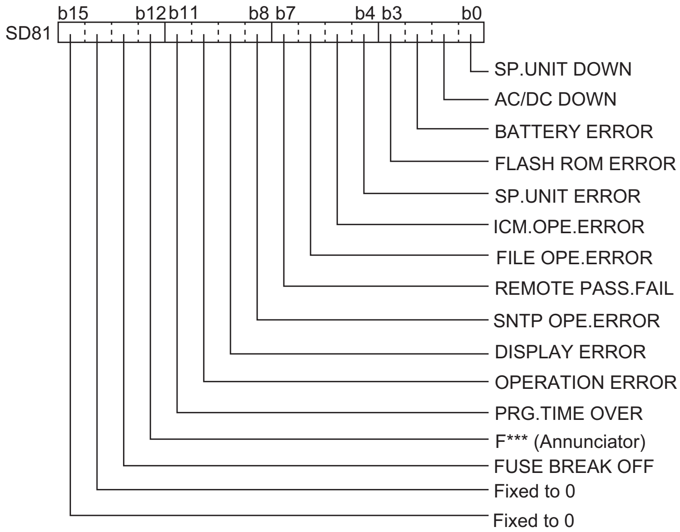
5. **Open the "PLC Diagnostics" dialog box and check that the errors have been cleared.**[*1]

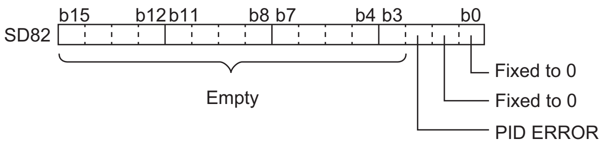
*1 Descriptions of the cleared errors are not deleted from error history data.

(b) SM and SD

Perform the following procedure.

1. **In SD81 and SD82, check the continuation errors that have been detected.**





2. **Remove the error cause of the continuation error.**

3. **In SD84 and SD85, specify the continuation errors to clear.**
(The bit patterns specified in SD84 and SD85 are same as those in SD81 and SD82)

4. **Turn off and on SM84.**

5. **In SD81 and SD82, check that the bits corresponding to the cleared errors are off.**

Point

Errors can also be cleared by storing the error code of the error to be cleared in SD50, and turning SM50 from off to on. In this method, however, errors cannot be cleared by error type.

(3) Status after error clear

When the CPU module is recovered from an error, SM, SD, and the LED that are relevant to the error return to the status before the error. If the same error occurs after clearing the error, the error is registered to the error history again.

(4) Precautions

- The error code that the user did not desire to clear may be cleared if its error message is the same, regardless of its error code.
- To clear multiple annunciators, perform error clear operation as the same time as the number of annunciators that are on.

3.26 LED Control Function

Whether to turn off the LED after an error and whether to indicate an error or not (on/off) can be set.

3.26.1 Methods for turning off the LEDs

The LEDs can be turned off by the following operations.

○: Applicable, ×: Not applicable

How to turn off	Relevant LED			
	BAT.	ERR.	I/O ERR.	USER
After resolving the error cause, execute the LEDR instruction.	○	○	○	○
After resolving the error cause, clear the error using*1 SM and SD. (Continuation error only)	○	○	○	○
Use SM202 and SD202	×	×	×	○

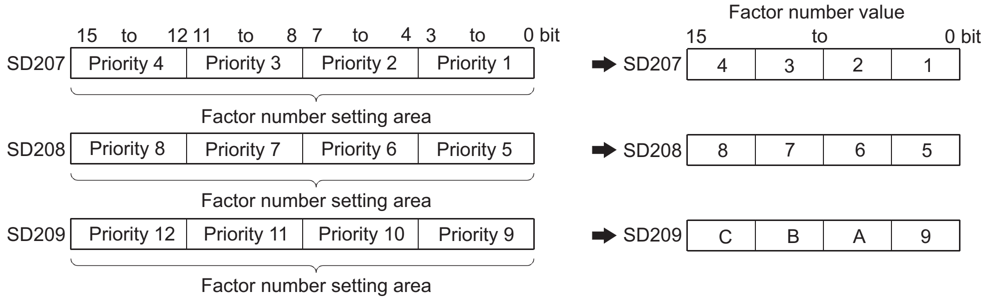
*1 For operation of SM and SD, refer to ☞ Page 166, Section 3.25. For the I/O ERR. LED, use SM1850, SM1870, SM1899, or SM1919. For details on SM and SD, refer to the following.
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

Point

For LED of the built-in CC-Link function, refer to the following.
MELSEC-L CC-Link System Master/Local Module User's Manual

3.26.2 LED indication priority

The LED indication is determined according to the factor number in SD207 to SD209 (LED indication priority). The LED indication can be disabled (off).

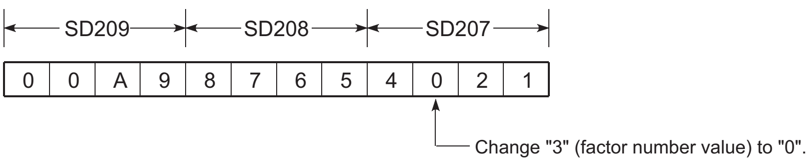


Priority order*1	Factor number (Hexadecimal)	Error message to be displayed	Remarks
1	1	AC/DC DOWN	Power-off
2	2	• FUSE BREAK OFF • SP.UNIT ERROR • SP.UNIT DOWN	• Fuse blown • Intelligent function module verification error • Intelligent function module error
3	3	• OPERATION ERROR • REMOTE PASS.FAIL • SNTP OPE.ERROR	• Operation error • Remote password error • SNTP error
4	4	• ICM.OPE.ERROR • FILE OPE.ERROR • FLASH ROM ERROR	• Memory card operation error • File access error • Number of flash ROM access exceeded
5	5	PRG.TIME OVER	Constant scan setting time exceeded
6	6	PID ERROR	PID control instruction error
7	7	Annunciator	—
8	8	—	—
9	9	BATTERY ERROR	—
10	A	—	—
11	B	—	—
12	C	DISPLAY ERROR	Display unit error

*1 When errors having the same priority simultaneously occur, the LED corresponding to the first detected error is displayed.

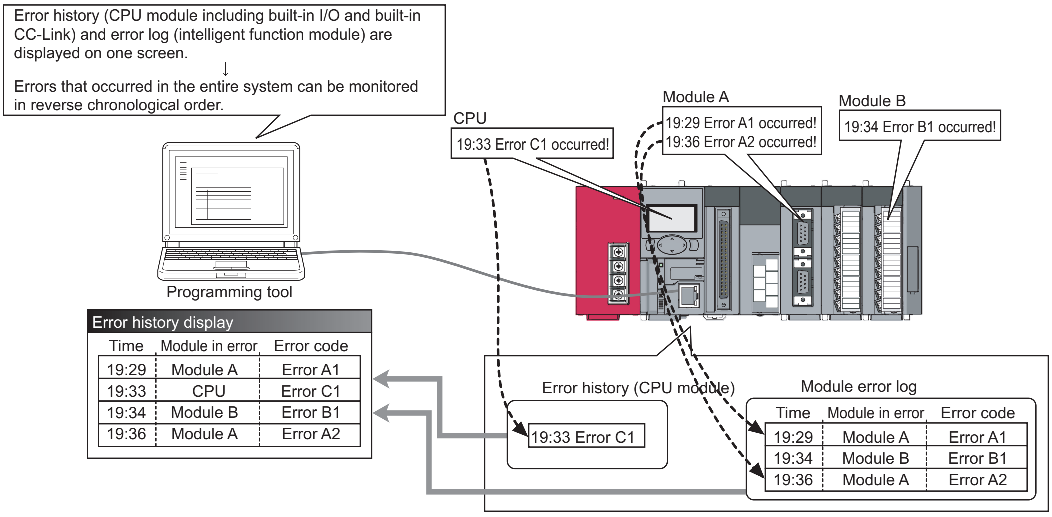
To disable (off) the LED indication for the error, set 0 to the relevant factor number in SD207 to SD209.
Note that even after disabling (off) the LED indication, SM0 and SM1 are turned on and the error codes are stored to SD0.

Ex. Disabling (off) the ERR.LED indication upon detection of a remote password error.

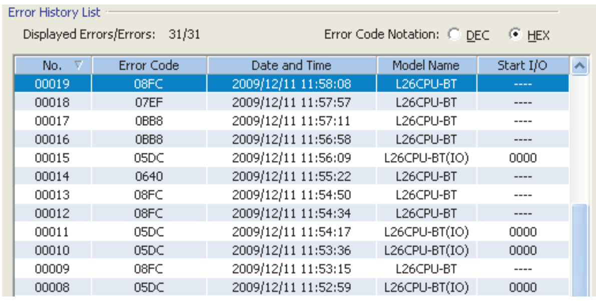


3.27 Module Error Collection Function

This function collects errors occurred in the connected intelligent function modules in the CPU module. By storing the errors in a memory that can hold data in the event of power failure, the errors can be held even after power-off or reset.



[Example of screen display]



No.	Error Code	Date and Time	Model Name	Start I/O
00019	08FC	2009/12/11 11:58:08	L26CPU-BT	----
00018	07EF	2009/12/11 11:57:57	L26CPU-BT	----
00017	0BB8	2009/12/11 11:57:11	L26CPU-BT	----
00016	0BB8	2009/12/11 11:56:58	L26CPU-BT	----
00015	05DC	2009/12/11 11:56:09	L26CPU-BT(IO)	0000
00014	0640	2009/12/11 11:55:22	L26CPU-BT	----
00013	08FC	2009/12/11 11:54:50	L26CPU-BT	----
00012	08FC	2009/12/11 11:54:34	L26CPU-BT	----
00011	05DC	2009/12/11 11:54:17	L26CPU-BT(IO)	0000
00010	05DC	2009/12/11 11:53:36	L26CPU-BT(IO)	0000
00009	08FC	2009/12/11 11:53:15	L26CPU-BT	----
00008	05DC	2009/12/11 11:52:59	L26CPU-BT(IO)	0000

Remark

When GX Developer is used, note that the error history cannot be displayed although the parameters for this function can be set.

(1) Supported module

The CPU module collects errors occurred in the connected intelligent function modules. The CPU module does not collect errors of those modules on other stations in the network.

(2) Timing when module errors are collected

Module errors are collected in END processing. Executing the COM instruction does not collect errors.

(3) Storing module errors

The module errors are stored in below, separately from error history data.

- System memory[*1]: 100 errors (fixed)
- Standard RAM: 1000 errors

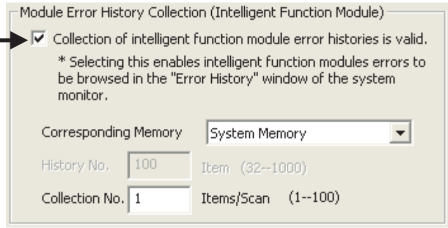
*1 The memory is managed inside the system.

(4) Setting procedure

On the "PLC RAS" tab, select "Collection of intelligent function module error histories is valid." in the "Module Error History Collection (Intelligent Function Module)" area.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC RAS]

Select any one option button. Deselected parameters remain unavailable.



Item	Setting item	Setting range	Default
Corresponding Memory	Select a storage location.	• System memory • Standard RAM *1*2	System memory
History No.	Enter the number of errors to collect only when they are stored in the standard RAM.	32 to 1000	100
Collection No.	Enter the number of errors to collect in one scan.*3	• Stored in system memory: 1 to 100 • Stored in standard RAM: 1 to 128	1

*1 When a sampling trace file is stored to the standard RAM, powering off and then on or resetting the CPU module will delete the file.

*2 The battery consumption may be increased.
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

*3 If collected module errors are frequently lost, set a greater value to "Collection No.". The recommended value is the number of intelligent function modules that support this function.

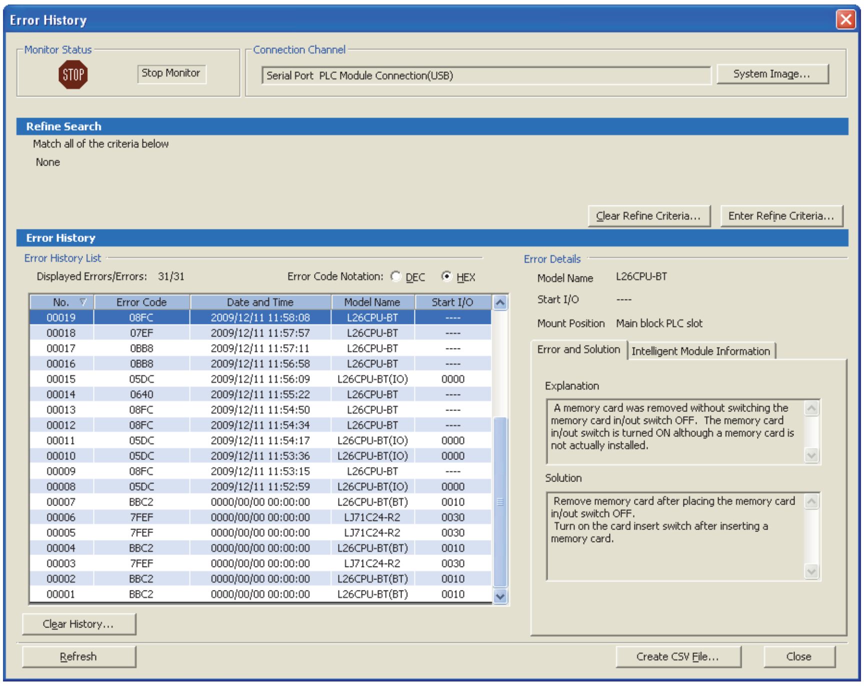
Parameter settings are enabled to the CPU module when:

- the CPU module is powered off and then on or
- the CPU module is reset.

(5) Monitoring module errors

To check the collected errors, open the "Error History" dialog box.

[Diagnostics] ⇨ [System Monitor...] ⇨ [System Error History]



Item	Description	Remarks
Error Code *1	Displays error code numbers.	—
Date and Time *2	Displays the year, month, day, hour, minute, and second when an error occurred.	The year can be displayed within the range of 1980 to 2079.
Model Name	Displays a module model name. For the built-in I/O and built-in CC-Link, the model name is displayed as follows. • Built-in I/O: CPU module model name (IO) (example: L02CPU(IO) or L26CPU-BT(IO)) • Built-in CC-Link: CPU module model name (BT) (example: L26CPU-BT(BT)	—
Start I/O	Displays the start I/O number of a module in error.	—

*1 For details on error codes, refer to the following.
Manual for the intelligent function module used

*2 If an error occurred during initial processing, its occurrence time may be stored as "0000/00/00 00:00:00" in the module error collection file. Such errors are not displayed in the order of occurrence.

Point

- The Error History dialog box can be displayed by selecting a module figure in the "Main block" area and clicking the Error History Detail button in the System Monitor dialog box. In this case, only the errors of the selected module are displayed.
GX Works2 Version 1 Operating Manual (Common)
- Errors are not displayed for modules that do not support the module error collection function.
- Errors may not be displayed when they occur frequently and successively.

(6) Clearing module error history

On the "Error History" dialog box, click the [Clear History...] button.

[Diagnostics] → [System Monitor...] → [System Error History]

Note that the errors displayed in the "Intelligent Module Information" tab are not cleared.

Point

The module error history data are cleared when the standard RAM is formatted. Note that a module error collection file cannot be deleted since it is automatically created after the CPU module is powered off and then on or is reset. To delete the file, clear the setting and then format the standard RAM.

(7) Precautions

The CPU module stops collecting errors if backing up or restoring data are performed using the CPU module change function with memory card.

3.28 Latch Data Backup to Standard ROM

This function holds (backs up) latch data, such as device data and error history, to the standard ROM without using a battery when the system is stopped for a long period. The stored data are restored when the system is restarted.

Point

When this function is used, the battery life-prolonging function is enabled even if it is set to be disabled by the parameter. The battery life-prolonging function is switched back to be disabled after executing this function (restoring stored data). The status (enabled/disabled) of the battery life-prolonging function can be checked with SD119 (battery life-prolonging factor).
For details on the battery life-prolonging function, refer to ☞ Page 198, Section 3.33.

(1) Backup target data and file size

The following table lists backup target data and sizes of files where data are stored.

Backup target data	Data description	File size (byte)
Device data	• File register (R, ZR)*1 • Extended data register (D)*1 • Extended link register (W)*1	64 + 2 x Number of file register points
	• Internal user device (M, L, B, F, V, T, ST, C, D, W) • Index register (Z)/standard device register (Z)	File sizes differ depending on the CPU module used. *4(☞ Page 175, Section 3.28 (1) (a))
Error history	• Error history information immediately before latch data backup to the standard ROM	
File transfer error history	Information on errors of the data logging file transfer function	
SFC program continuation start information	Information for starting the SFC program continuously	
Module error collection*2	Information on errors that has occurred in the intelligent function module (module error collection function)	92 + 64 x Maximum number of module errors
Trace setting (Sampling trace file)*3	Trace condition settings and trace data settings created by the sampling trace function	16 + Sampling trace file size

*1 To backup the data, check the "Transfer to Standard ROM at Latch data backup operation" checkbox on the PLC file tab.
*2 Regardless of the setting of the module error collection function, data is backed up.
*3 A storage file is created only when the trace registration has been made. The data are not backed up when trace settings are not written to the CPU module.
*4 These are sizes when the device assignment is default. Sizes differ depending on parameter settings.

(a) File size

CPU module	Serial number (first five digits)	File size (byte)
L02SCPU, L02SCPU-P	15101 or earlier	93702
	15102 or later	93710
L02CPU, L02CPU-P	12111 or earlier	93802
	12112 or later	99098
	14112 or later	106702
	15102 or later	106710
L06CPU, L06CPU-P	—	130382
	15102 or later	130390
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	12111 or earlier	117482
	12112 or later	122778
	14112 or later	130382
	15102 or later	130390

3.28.1 Latch data backup

The following two methods are available for backing up latch data to the standard ROM.

- By contacts
- By remote operation

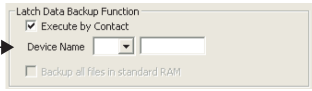
(1) Execution by contacts

(a) Setting method

Configure the setting for Latch Data Backup Operation Valid Contact. (The devices applicable to a contact are X, M, and B.)

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Set a contact.



(b) Execution method

Backup starts at the rise of a contact (off→on). After backup, the BAT.LED of the CPU module flashes (green), indicating that the CPU module is in the standby status ready to be powered off. The execution status of backup operation can be checked by SM671 or SD671 to SD675.

(c) Precautions

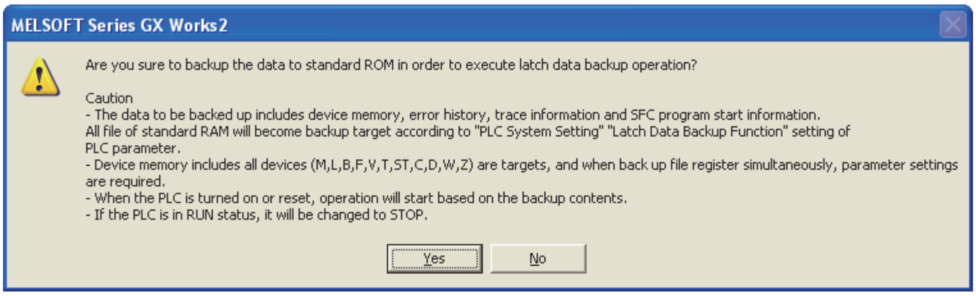
- To set the CPU module to the RUN status from the standby status, power it on again or reset after the backup operation is complete.
- The status of latch data backup valid contact to the standard ROM is checked at execution of the END instruction. Therefore, data are not backed up even if a contact is repeatedly turned on and off (on → off → on, or off → on → off) in one scan.
- When the latch data backup valid contact to the standard ROM is set to X, and the CPU module is powered off and then on or is reset without turning off the contact, the data cannot be backed up unless the latch data backup valid contact to the standard ROM is turned off and then on again.
- The latch data backup start contact to the standard ROM is set to M or B, and data are backed up by turning off and then on the contact the data cannot be backed up unless the latch data backup valid contact to the standard ROM is turned off and then on again.

(2) Execution by remote operation

(a) Execution method

Open a dialog box to execute a remote operation.

[Online] ⇨ [Latch Data Backup] ⇨ [Backup]



Data to be backed up are the data at the execution of remote operation. After backup, the BAT.LED of the CPU module flashes (green), indicating that the CPU module is in the standby status ready to be powered off.

(3) Deleting backup data

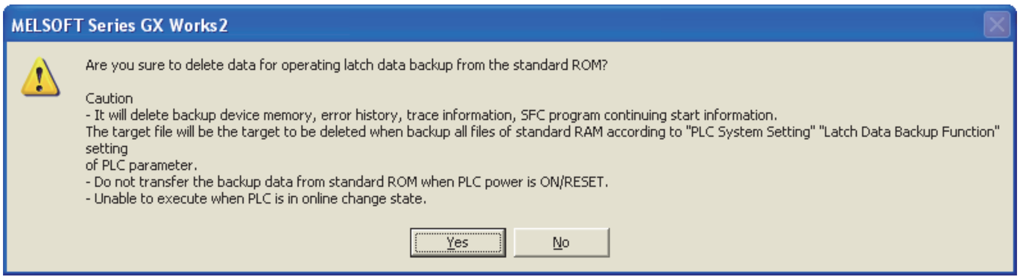
The following two methods are available for deleting the backed up data.

- Remote operation
- Formatting the standard ROM

(a) Execution by remote operation

Open a dialog box to execute a remote operation. (This operation is available only while the CPU module is in the STOP status.)

[Online] ⇨ [Latch Data Backup] ⇨ [Delete Backup Data]



Deleting backup data clears values of SM and SD.

(b) Execution by formatting the standard ROM

Open the "Format PLC Memory" dialog box.

[Online] ⇨ [PLC Memory Operation] ⇨ [Format PLC Memory]

(4) Precautions

(a) Power-off and reset of the CPU module during backup

The backup data are deleted. Performing power-off of reset of the CPU module causes "RESTORE ERROR" (error code: 2221) and data cannot be restored.

(b) Priority of backup data

- When the device initial value has been set, it takes effect. (Therefore, after reflecting the backup data, the device where the initial device value setting is configured is overwritten by the device data of the initial device value.)
- When the latch device or latch range has been set, the backup data take effect. (Therefore, even if data of latch device or latch range setting are changed after backup, it is overwritten by the data backed up when the CPU module was powered off and then on or was reset.)

(c) When using a local device

Devices where local device range setting is configured are not backed up.

(d) The number of writes to standard ROM

The number of writes to standard ROM ("FLASH ROM ERROR" (error code: 1610) is detected), data may not be normally backed up.

(e) Operations disabled during backup

The following operations cannot be performed during backup. Perform them after the backup operation.

- Format PLC memory (standard ROM only)
- Latch data backup by remote operation
- Online change (ladder mode, files, function block)

3.28.2 Restoring backup data

Backup data are automatically restored when:

- the CPU module is powered off and then on or
- the CPU module is reset.

Whether to restore data once after backup or per above operation can be set by on/off of SM676.

Status of SM676	Restoration operation
SM676 = OFF	Data are restored once when the CPU module is powered off and then on or is reset after backup.
SM676 = ON	Data are restored whenever the CPU module is powered off and then on or is reset after backup. Data are repeatedly restored until the backup data are deleted or the latch data are backed up next time.

After backup data are restored, the BAT.LED on the CPU module turns on (green) for five seconds.

Point

If the number of device points at the time of parameter configuration are different from those at backup, "RESTORE ERROR" (error code: 2220) is detected upon restoration, and the restoration cannot be completed. (Restoration is performed again when the CPU module is powered off and then on or is reset the next time.) To complete the restoration, perform any of the following operations.

- Set the parameters back to the configuration before backup.
- Delete the backup data.
- Back up the data again.

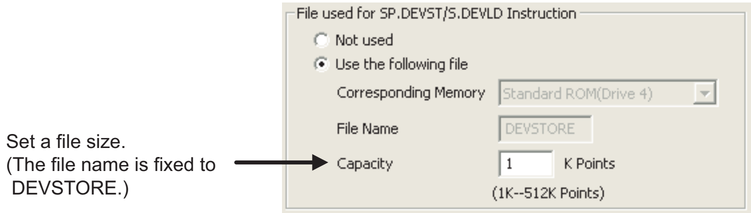
3.29 Writing/Reading Device Data to/from Standard ROM

This function writes device data to the standard ROM. Writing the fixed values for operation and operation results to the standard ROM can prevent losing data due to low battery. The data written to the standard ROM can be read at any given timing by using an instruction.

(1) Setting procedure

Set the area, where device data are stored, to the standard ROM.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC File]



(a) File size setting

The capacity that can be set varies depending on the CPU module.

CPU module	Setting range
L02SCPU, L02SCPU-P, L02CPU, L02CPU-P	1 to 16K points
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	1 to 512K points

(2) Devices that can be written

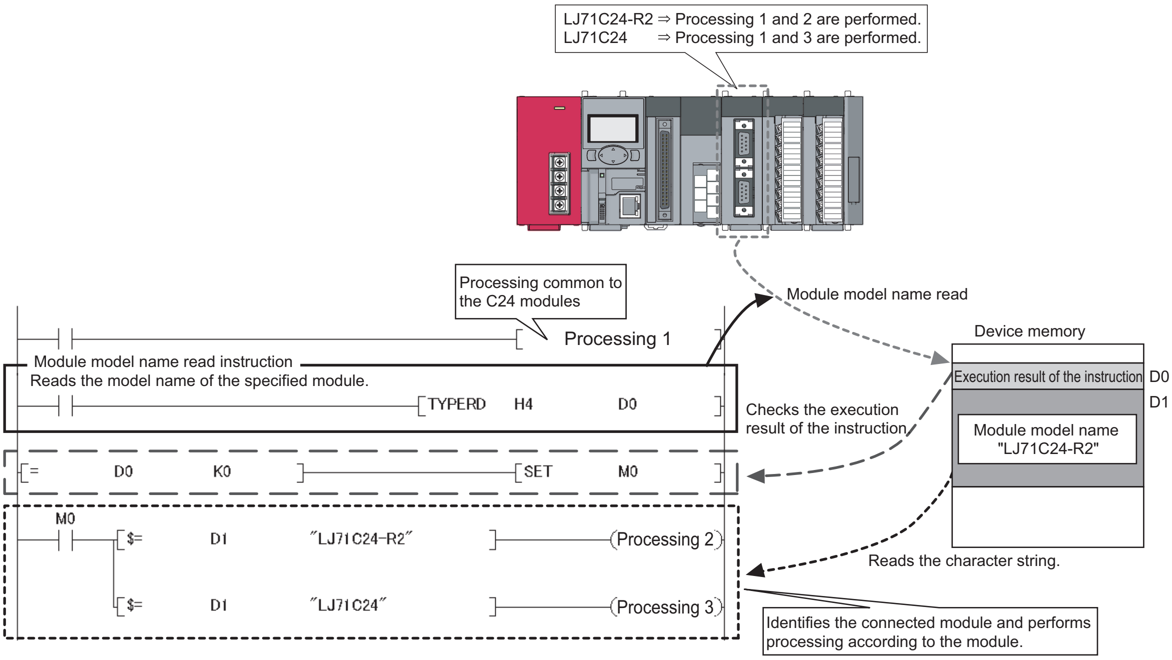
- Internal user device (X, Y, M, L, B, F, SB, V, T, ST, C, D, W, SW)
- Internal system device (SM, SD)
- File register (R, ZR)
- Extended data register (D)
- Extended link register (W)

(3) Execution method

Use the SP.DEVST instruction to write device data to the standard ROM. The device data written to the standard ROM are read to the specified device by the S(P).DEVLD instruction.
For details, refer to the following.
MELSEC-Q/L Programming Manual (Common Instruction)

3.30 Module Model Name Read

This function reads the model name of a module connected. The connected modules can be identified in a program so that processing for individual module can be performed.



(1) Execution method

Use the TYPERD instruction.

For details, refer to the following.

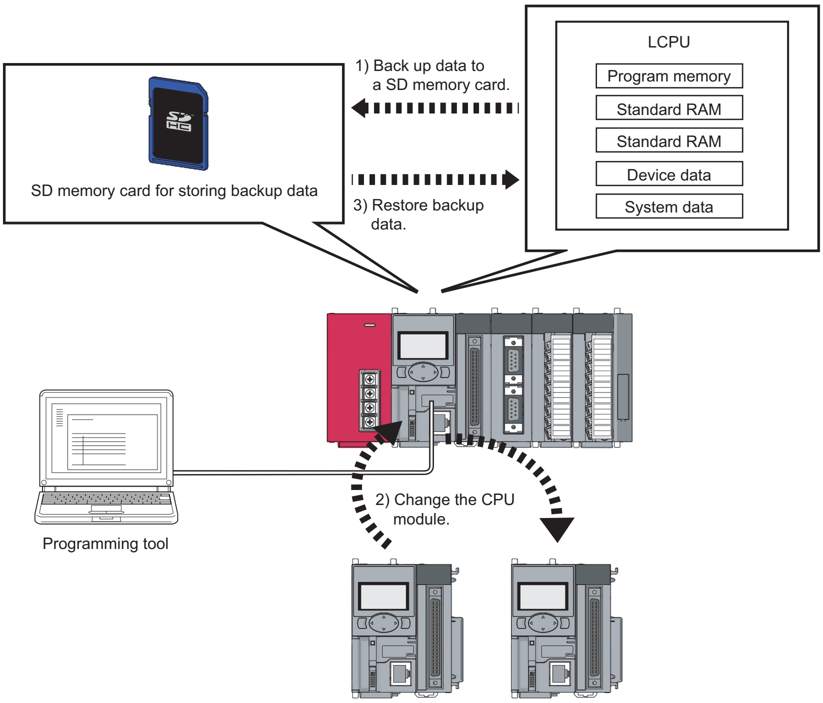
MELSEC-Q/L Programming Manual (Common Instruction)

3.31 CPU Module Change Function with SD Memory Card

Remark

The L02SCPU and L02SCPU-P do not support this function.

This function enables to pass data from a module to be changed (disconnected) to the newly-connected CPU module. By using this function, control can be continued after replacing a CPU module. To pass data, backup the data to an SD memory card before replacing a CPU module, and restore them in the newly-connected CPU module before replacing a CPU module.



(1) Backup data file

Only one backup data file can be stored to an SD memory card. If any backup data file exists in the SD memory card, the file data are overwritten.
To delete the backup data file, select "Delete PLC Data..." of a programming tool.

(2) Backup data

(a) Backup data selection

Select a drive to back up from the following table.

Backup data (drive)	Description	Backup by user
Program memory (drive 0)	All data in the program memory (drive 0) *1	Allowed
Standard RAM (drive 3)	All data in the program memory (drive 3)	
Standard ROM (drive 4)	All data in the program memory (drive 4)	
Device data *2	Internal user device (L, B, F, V, T, ST, C, D, W)	Not allowed (backed up by the system)
System data	Data managed by the system (such as module error history)	

*1 Data in the program cache memory are backed up.
*2 Latch devices or devices set in the latch range are backed up.

(b) Maximum backup data size

The following shows the maximum backup data size.

(Unit: K byte)

Backup target data (drive)	L02CPU, L02CPU-P	L06CPU, L06CPU-P	L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT
Program memory (drive 0)	82	244	1048
Standard RAM (drive 3)	130	770	770
Standard ROM (drive 4)	516	1032	2056
Device data	128	128	128
System data	53	76	76
Total (maximum)	909	2250	4078

The backup data size can be checked by the following.

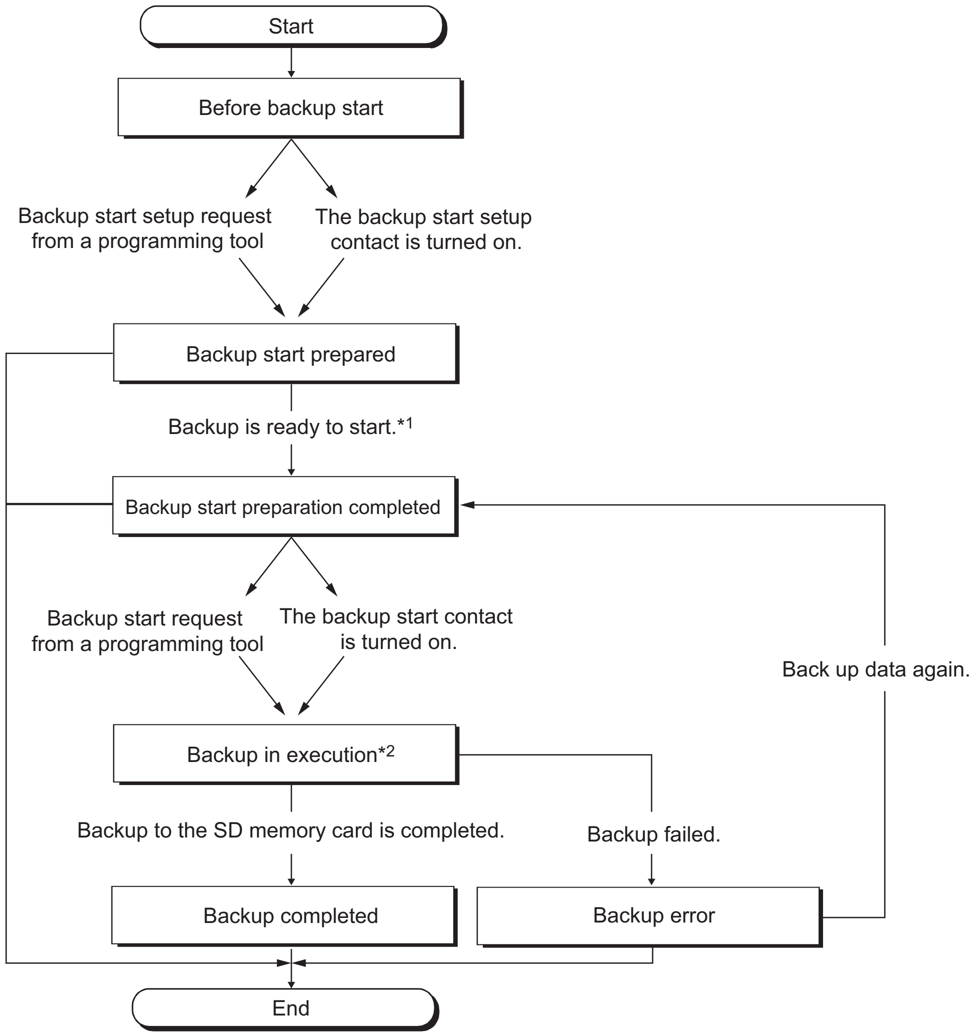
- "Create Backup Data for PLC Module Change" dialog box (☞ Page 189, Section 3.31.1 (2) (b))
- SD698 and SD699[*1]

*1 Available after starting backup.

3.31.1 Backup to SD memory card

This function can save data in the CPU module to an SD memory card. If an SD memory card is used in a running system, data can be backed up by replacing the SD memory card with the one for storing a backup data.

(1) Procedure



*1 After preparation for backup is completed, the following functions are disabled. (They are not resumed after backup.)
· Refresh of a network module
· Refresh of CC-Link IE Field Network Basic
· Auto refresh of intelligent function module
· Simple PLC communication function

*2 Do not perform the following during backup.
· Insertion and removal of an SD memory card
· Power-off
· Reset

Backup status can be checked in SD690.

SD690 value	Status	Description
0_H	Before backup start	Backup is not started
1_H	Backup start prepared	An SD memory card can be inserted/removed
2_H	Backup start preparation completed	Set data to backup
3_H	Backup in execution	Backup is in execution
4_H	Backup completed	Backup has been completed
FF_H	Backup error	Backup has failed due to an error

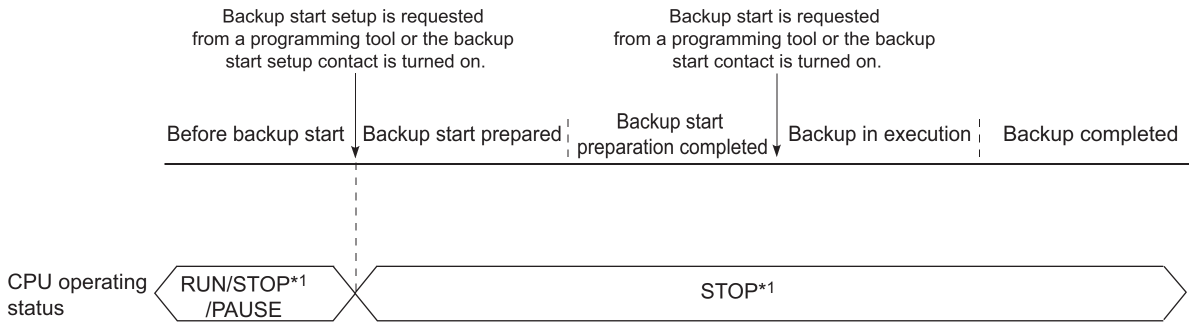
(a) Insertion and removal of SD memory card

An SD memory card can be inserted/removed after preparation for backup is completed.

(b) Operating status of the CPU module

To backup, the CPU module must be in the STOP status. (When the CPU module is either in the RUN or PAUSE status, it is switched to the STOP status after the END processing where a request to backup start preparation was received.)

After replacing the CPU module, it must be powered off and then on or reset. (Failure to do so keeps the CPU module in the STOP status even after setting the switch to the RUN position.)



*1 The status includes a stop error.

(2) Execution method

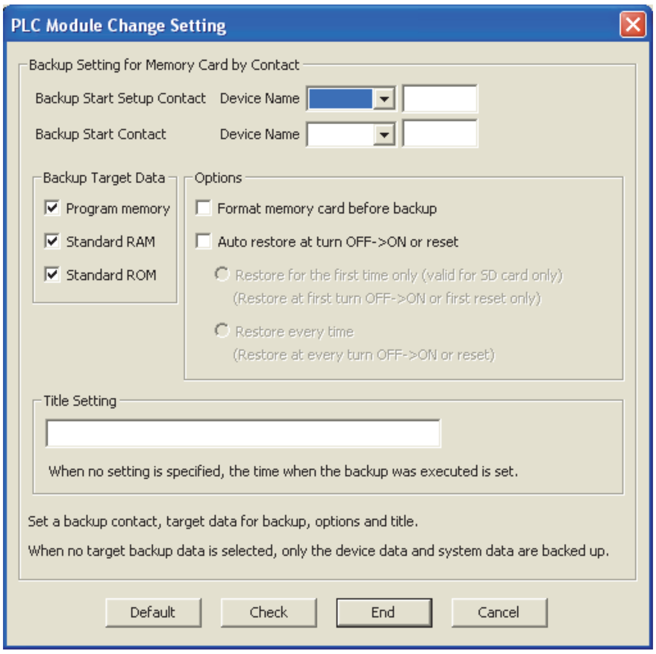
The following two ways are available for backing up data to SD memory card.

- By contacts
- By remote operation

(a) Execution by contacts

Backup is executed by turning on the device specified in the "PLC Module Change Setting" dialog box.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System] ⇨ [PLC Module Change Setting]



Item	Description	Setting range	Default
Backup Start Setup Contact *1	At the rise of the selected device, backup is ready to start.	Applicable device *2 • X (0 to 1FFF) • M (0 to 8191) *3 • B (0 to 1FFF) *3	—
Backup Start Contact	At the rise of the selected device, backup enters execution status.		
Backup Target Data	Select the data to backup.*4	• Program memory (drive 0) • Standard RAM (drive 3) • Standard ROM (drive 4)	All drives are backed up.
Format memory card before backup	Select whether to format the SD memory card before backup.	Format Not format	Not format
Title Setting*5	Set a title to append to the backup data stored in the SD memory card.	32 characters	Current time is set. (Example) If data are backed up at 12 p.m. on October 1, 2008, "20081001200" is set.

*1 The CPU module enters the STOP status at the rise of the backup start setup contact, the backup start contact cannot be turned on in the program.

*2 The backup start setup contact and backup start contact cannot be set to the same device.

*3 This indicates the default number of points. The setting range when the internal user device is set to the maximum number of points (60K points) is M (0 to 61439) and B (0 to 0EFFF).

*4 If data to be backed up is deselected in "Backup Target Data", only device data and system data are backed up.

*5 Title is used for identifying backup data. Settings for the title of the backup data stored in an SD memory card can be checked on the "Online Data Operation" dialog box with the "Delete" radio button selected.

Turn on the backup start setup contact and then the backup start contact. Data are not backed up when only the backup start contact is on.

Turn on the backup start setup contact.

↓

Preparation for backup:
1) Set the CPU module to the STOP status.
2) Set the SD memory card so that the card can be removed from the CPU module.

↓

1) Check that the preparation for backup is completed by the following methods.
 - Check the LEDs.
 - Check the special relay and special register.
   (a) SM691 is on.
   (b) SD690 stores 2h.
2) Remove the SD memory card.
3) Install a SD memory card for storing backup data.
4) Turn on the backup start contact.

↓

Back up data.

↓

Check that the backup is completed by the following methods.
 - Check the LEDs.
 - Check the special relay and special register.
   (a) SM691 is on.
   (b) SD690 stores 4h.

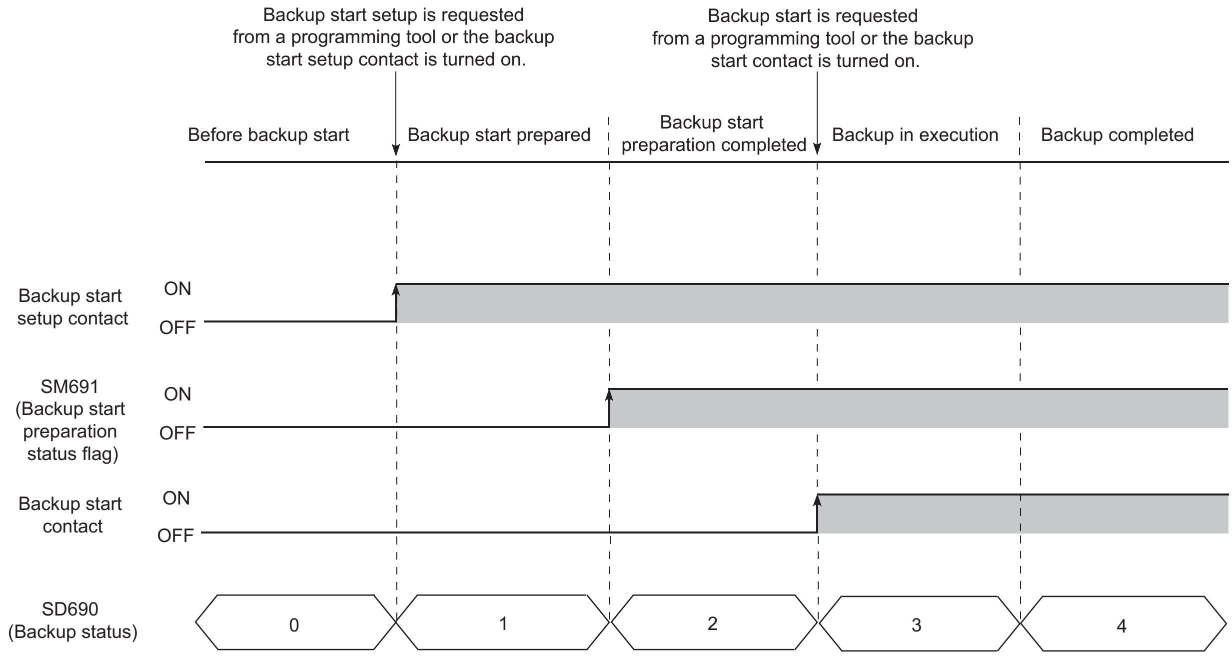
Check for backup error*6.
 - Check the LEDs.
 - Check the special relay and special register.
   (a) SM691 is on.
   (b) SD690 stores FFh.
   (c) SD689 stores an error cause.

: Operation by the CPU module

: Operation by the user

*6 Since SM691 (Backup start preparation status flag) is on, data can be backed up again by turning off and then on the backup start contact.

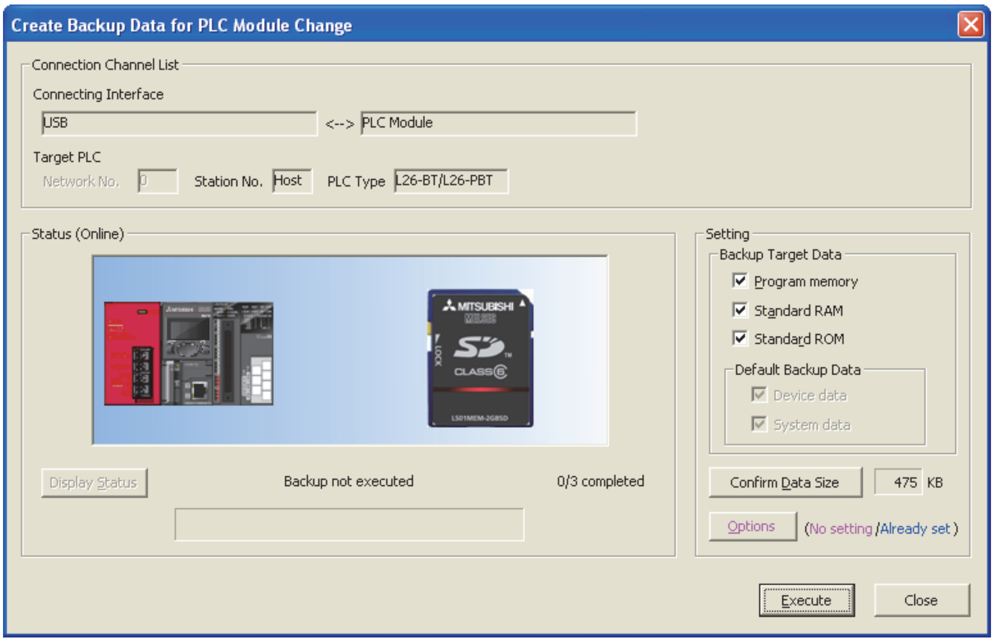
If the backup start contact is turned on while the value in SD690 is 0_H (Before backup start) or 1_H (Backup start prepared), data are not backed up. If the backup start contact is on before the value in SD690 becomes 2_H (Backup start preparation completed), turning off and then on the backup start contact again while the value in SD690 is 2_H (Backup start preparation completed) starts backup.



(b) Execution by remote operation

Open the "Create Backup Data for PLC Module Change" dialog box.

[Online] ⇨ [PLC Module Change] ⇨ [Create Backup Data...]



Point

Clicking the [Confirm Data Size] button displays the backup data size. (A value appears regardless of the CPU module connection status and SD memory card insertion status.) Data size can be checked only when the backup operation was executed by a remote operation.

(3) LEDs indicating backup status

Backup operation status can be checked by LED indication as shown below.

SD690 value	Backup status	LED indication
2_H	Backup start preparation completed	MODE: Flash (green), BAT.: Flash (yellow)
3_H	Backup in execution	The color changes as follows at intervals of 800ms. 1) MODE: Flash (green), BAT.: On (green) ↓ 2) MODE: Flash (green), BAT.: On (green), USER: On (red) ↓ 3) MODE: Flash (green), USER: On (red)
4_H	Backup completed	MODE: Flash (green), BAT.: Flash (green)
FF_H	Backup error	MODE: Flash (green), BAT.: Flash (green), USER: Flash (red)

(4) Causes of a backup error

If backup was not completed, a diagnostic error is not detected. In that case, the error cause is stored in SD689 and the error response is returned to the programming tool.

SD689 (Backup error factor) value	Error response number	Error cause
100_H*1	$41FE_H$*2	• Backup started without an SD memory card inserted or without the SD memory card lock switch slid down. • Backup preparation or backup started while an SD memory card is disabled by SM606 (SD memory card forced disable instruction).
200_H	—	Size of data to be backed up exceeds the capacity of the SD memory card inserted.
300_H	—	Write protection has been set to the SD memory card.
400_H	—	Writing data to the SD memory card has failed.
500_H	—	Reading data from a drive storing backup data was not completed. (program memory read error)
503_H	—	Reading data from a drive storing backup data was not completed. (standard RAM read error)
504_H	—	Reading data from a drive storing backup data was not completed. (standard ROM read error)
510_H	—	Reading data from a drive storing backup data was not completed. (system data read error)
600_H*1	4335_H*2	Backup preparation started while latch data were being backed up to the standard ROM.
601_H*1	$410A_H$*2	Backup preparation started while data were being written in the RUN status.
602_H*1	4336_H*2	Backup preparation started with an FTP client connected to and communicated with the CPU module.
603_H*1	4276_H*2	Backup preparation started while the data logging function was being executed.
605_H*1	$433A_H$*2	Backup preparation started while the project data batch save/load function was being executed.
606_H*1	4904_H*2	Backup preparation started while any specified file or folder was being deleted using a display unit.
—	4082_H*3, 4330_H*4	Backup preparation or backup started while another backup was in execution.
—	4333_H*2	Backup started while the CPU module was in "Before backup start" (SD690 = 0).
607_H *1	4800_H *2	Backup preparation started while the iQ Sensor Solution function (data backup/restoration) was being executed.
608_H *1	$433D_H$ *2	Backup preparation started while the file transfer function (FTP client) was being executed.
701_H *1	4426_H *2	Backup preparation started when a block password for which "Execution Program Protection Setting" was enabled was set.

*1 Only when data are backed up using contacts
*2 Only when data are backed up by remote operation
*3 When data are backed up from another boot source
*4 When data are backed up from the same boot source

(5) Functions that cannot be performed during backup

The following functions cannot be executed during backup.

Category	Function	Category	Function
Drive operation	Format PLC memory	Online change	Online change (ladder mode)
	Program memory batch download		Online change (files)
	Arrange PLC memory		Online change (multiple blocks)
File operation	Write to PLC		Change TC setting
	Delete PLC data	Trace	Sampling trace registration
	Write PLC user data	Data logging	Data logging registration
	Delete PLC user data	Remote operation	Remote latch clear
	File password 32	SM/SD operation	Latch clear by using the special relay and special register areas
	Latch data backup	Monitor	Monitor condition setup
FTP function	For all operations and commands	Device test	Executional conditioned device test

(6) Precautions

(a) During data logging

Backup operation cannot be performed during data logging. To perform backup, stop the data logging. For how to stop data logging, refer to the following.

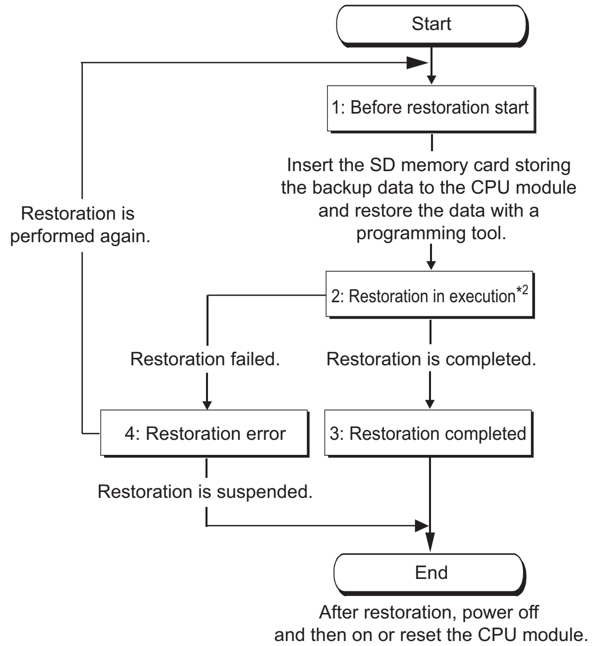
📖QnUDVCPU/LCPU User's Manual (Data Logging Function)

3.31.2 Backup data restoration

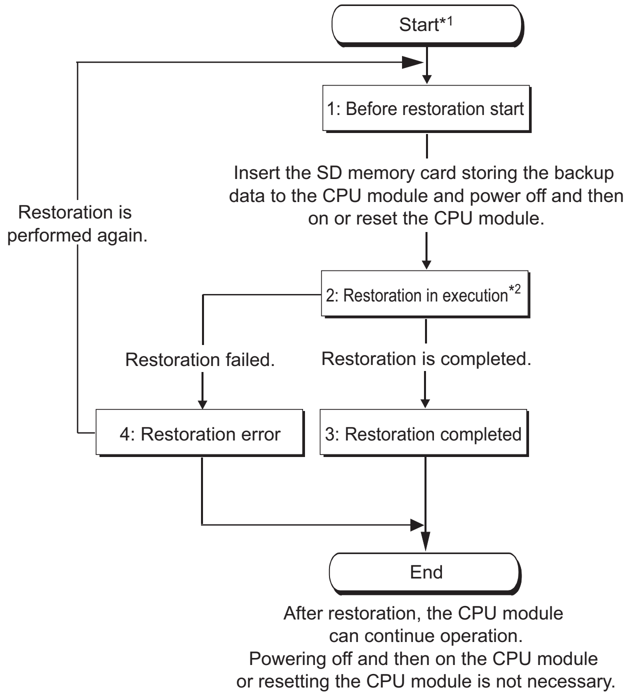
This function restores backup data in an SD memory card to the CPU module.

(1) Procedure

• Restoration by remote operation



• Automatic restoration

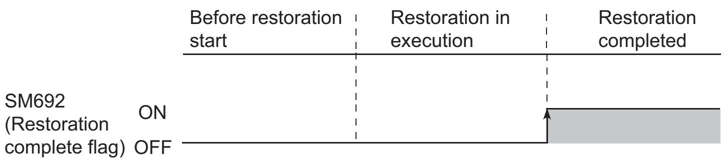


*1 Once a restoration is started, the following functions are stopped. (They are not resumed after restoration.)
· Refresh of a network module
· Refresh of CC-Link IE Field Network Basic
· Auto refresh of intelligent function module
· Simple PLC communication function

*2 Do not perform the following operations during restoration.
· Inserting or removing an SD memory card
· Power-off
· Reset

Status	Description	SD693 value
Before restoration start	Restoration is not performed.	0_H
Restoration in execution	Restoration is in execution.	1_H
Restoration completed	Restoration is completed.	2_H
Restoration error	Restoration failed due to an error.	FF_H

After restoration is completed, SM692 turns on.



(2) Execution method

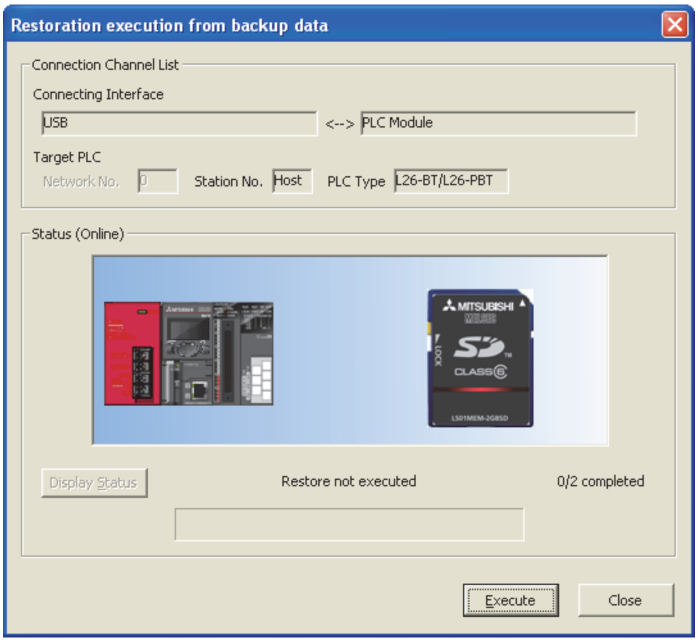
The following two methods are available for restoring backup data.

- Remote operation
- Automatic restoration

(a) Execution by remote operation

Open the "Restoration execution from backup data" dialog box.

[Online] ⇨ [PLC Module Change] ⇨ [Restore...]



To validate restored data, click "Yes" in the screen that appears after clicking the Execute button in the above dialog box and then power on or reset the CPU module.

(b) Execution by automatic restoration

On the "CPU Module Change Setting" dialog box, select "Auto restore at turn OFF -> ON or reset". (☞ Page 186, Section 3.31.1 (2) (a))

After backup, turn off and then on or reset the CPU module so that restoration starts. Restoration timing can be set from either once at the first time or every time.

Option setting item	Turn on from off or reset of the CPU module	
	First time	Second time or later
Restore for the first time only	Restoration executed	Restoration not executed (Operate as a standard SD memory card)
Restore every time	Restoration executed	Restoration executed

(3) LEDs indicating restoration status

Restoring status can be checked by LED indication as shown below.

SD693 value	Restoration status	LED indication
0_H	Before restoration start	MODE: On (green)
1_H	Restoration in execution	The color changes at intervals of 800ms as follows. 1) MODE: Flash (orange), BAT.: On (green) ↓ 2) MODE: Flash (orange), BAT.: On (green), USER: On (red) ↓ 3) MODE: Flash (orange), USER: On (red)
2_H	Restoration completed	• By remote operation···MODE: Flash (orange), BAT.: Flash (green) • By automatic restoration···MODE: On (green)
FF_H	Restoration error	• By remote operation···MODE: Flash (orange), BAT.: Flash (green), USER: Flash (red) • By automatic restoration···MODE: On (green), ERR.: Flash (red)

(4) Causes of a restoration error

If restoration was not completed, a diagnostic error is not detected. In that case, the error cause is stored in SD692 and the error response is returned to the programming tool.

SD692 value	Error response number	Error cause
800_H	—	The CPU module where data are restored is different model from the one where the backup source data are stored.
801_H	—	• Backup data files do not match. • Reading of backup data from an SD memory card was not completed.
810_H	—	Writing of backup data to the restoration destination drive was not completed.
—	4335_H*1	Restoration started while latch data were being backed up to the standard ROM.
—	$410A_H$*1	Restoration started while data were being written in the RUN status.
—	4276_H*1	Restoration started during data logging.
—	4336_H*1	Restoration started with an FTP client connected to and communicated with the CPU module.
—	4330_H*1	Restoration started during another restoration.
—	$433A_H$*1	Restoration started while the project data batch save/load function was being executed.
—	4904_H*1	Restoration started while any specified file or folder was being deleted using a display unit.
—	$41FE_H$*1	• Restoration started without an SD memory card inserted or without the SD memory card lock switch slid down. • Restoration started while an SD memory card is disabled by SM606 (SD memory card forced disable instruction).
—	4800_H *1	Restoration started while the iQ Sensor Solution function (data backup/restoration) was being executed.
—	$433D_H$*1	Restoration started while the file transfer function (FTP client) was being executed.

*1 Only when data are restored by remote operation.

If automatic restoration was not completed, "RESTORE ERROR" (error code: 2225 to 2227) occurs.

Error code	Error message	Error cause
2225	RESTORE ERROR	The CPU module where data are restored is different model from the one where the backup source data are stored.
2226		• Backup data file is corrupt (The contents of backup data file do not match with the check code). • Reading of backup data from an SD memory card was not completed. • The SRAM card has been write-protected so that "Restore for the first time only" setting can not take effect.
2227		Writing of backup data to the restoration destination drive was not completed.

(5) Functions that cannot be performed during restoration

Functions that cannot be performed during restoration are the same as those cannot be performed during backup.

(☞ Page 191, Section 3.31.1 (5))

(6) Precautions

(a) When boot settings are configured

If any parameter in the SD memory card has been set for booting, the data are overwritten according to the boot setting even restoration is executed.

(b) When file password 32 is configured

Even if a file in the CPU module is protected by password 32, restoration continues.

(c) Combination of CPU modules for restoration

If the model of the CPU module where data is restored is different from the model of the CPU module where backup data is stored, Restoration error occurs. However, using the CPU modules of the following combinations allows completion of backup and restoration.

- L02CPU and L02CPU-P
- L06CPU and L06CPU-P
- L26CPU and L26CPU-P
- L26CPU-BT and L26CPU-PBT

3.32 Clock Function

This function is used for clock data management, such as storing a date into the error history, by reading internal clock data of the CPU module. Clock data are retained the battery of a CPU module even after power-off or a momentary power failure is occurred exceeding the allowable period of time.

(1) Clock data details

Clock data	Description
Year	Four digits[*1] (from 1980 to 2079)
Month	1 to 12
Day	1 to 31 (Automatic leap year detection)
Hour	0 to 23 (24 hours)
Minute	0 to 59
Second	0 to 59
Day of the week	0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday
1/1000sec [*2]	0 to 999

*1 Storing in SD213 for the first two digits and SD210 for the last two digits of the year.
*2 Use only the expansion clock data read (S(P).DATERD) to read.
MELSEC-Q/L Programming Manual (Common Instruction)

(2) Changing clock data

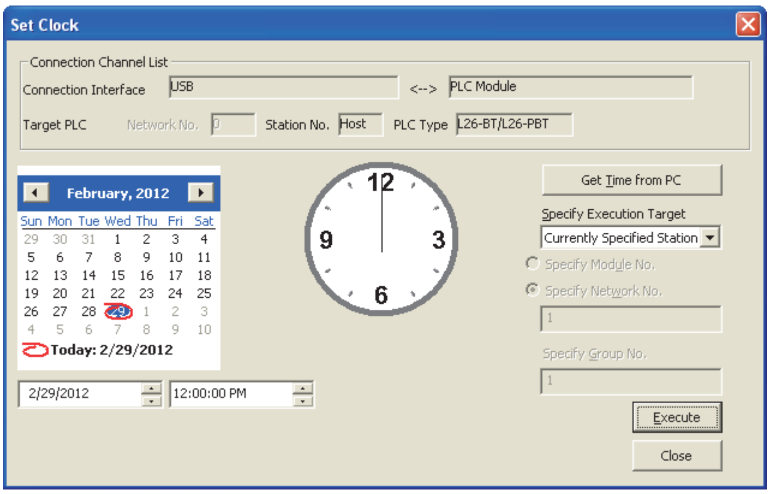
To change clock data, use any of the following.

- Programming tool
- SM and SD
- Program
- Clock setting of the display unit

(a) Programming tool

Open the "Set Clock" dialog box.

[Online] ⇨ [Set Clock...]



(b) Clock setting of the display unit

Clock data can be set and checked in the display unit. (Page 263, Section 4.3.1)

(c) SM and SD

Store clock data in SD210 to SD213. After the END processing for the scan where SM210 (Clock data set request) were turned on from off, write the values stored in SD210 to SD213 to the CPU module.

(d) Program

Use an instruction to write clock data (DATEWR).
MELSEC-Q/L Programming Manual (Common Instruction)

Point

- After changing clock data, a 1/1000 is reset to 0.
- Year data that can be set with GX Works2 is up to 2037.

(3) Clock data read

To read clock data, use either of the following.

- Program
- SM and SD

(a) Program

Use either of the following instructions. (MELSEC-Q/L Programming Manual (Common Instruction))

- Clock data read (DATERD)
- Expansion clock data read (S(P).DATERD)

(b) SM and SD

The data are read out into SD210 to SD213 in a BCD value when SM213 (clock data read request) is turned on.

(4) Precautions

(a) Initial clock data setting

Set the correct clock data because it has not been set when shipping.

(b) Correcting clock data

Whenever any part of the clock data are changed, the entire clock data must be written to the CPU module again.

(c) Clock data range

When changing clock data, write data within the range given (3) in this section. If data outside the clock range is written to the CPU module, the clock function does not operate correctly. However, errors do not occur if the value is within the following range.

Example	Write operation to the CPU module	CPU module operation
February 30	Performed	No error is detected.
Day 32 of month 13	Not performed	• Upon DATEWR instruction: "OPERATION ERROR" (error code: 4100) • Upon turn-on of SM210: SM211 (Clock data error) turns on

3.33 Battery Life-Prolonging Function

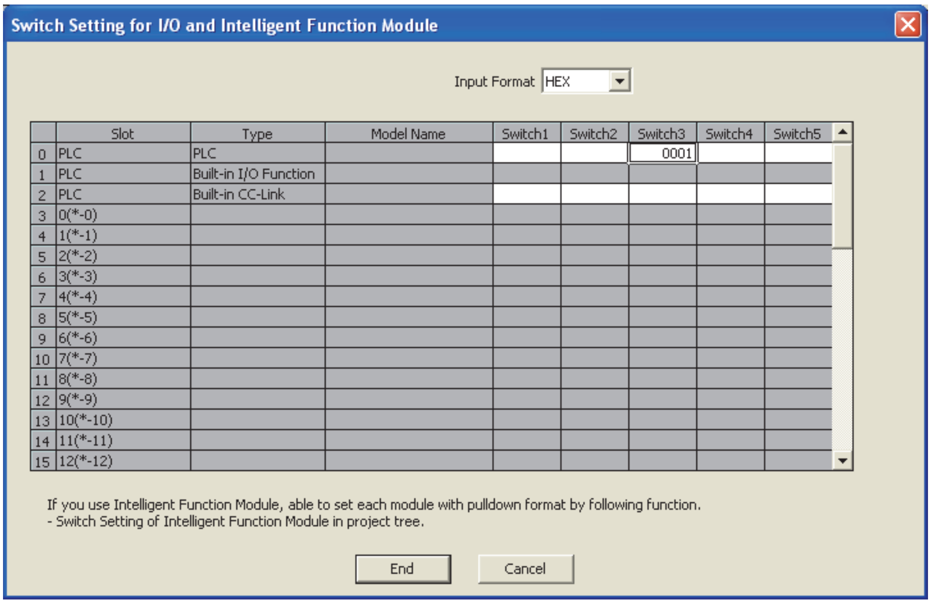
This function extends battery life of the CPU module by limiting data to retain by a battery to clock data only. The use of this function initializes all data other than the clock data when the CPU module is powered off or is reset.

Data retained by a battery	Description
Error history	The number of error history data are initialized to zero.
Latch relay (L)	Cleared to zero.
Device in the latch range	Cleared to zero.
Files in the standard RAM	Files are deleted. (Files are created again at power-on or reset.) Data are cleared to zero.

(1) Setting procedure

Enter 0001_H under Switch3 in the "Switch Setting for I/O and Intelligent Function Module" dialog box.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [I/O Assignment] ⇨ [Switch Setting]



For the life of battery installed in the CPU module when the battery life-prolonging function is used, refer to the following.

MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

3.34 Memory Check Function

This function checks that data in the memories of the CPU module are not changed due to such as excessive electric noise. Since the CPU module automatically checks a memory, setting for enabling this function is unnecessary. This function does not require processing time.

(1) Data to be checked

(a) Program

The program during execution is compared with the user program written to the program memory. If they do not match, a stop error, "RAM ERROR" (error code: 1160) is detected.

(b) Parameter

The parameters are compared with the ones written to the parameter-valid drive.

(c) Device memory

If the change of data in the device memory is detected, a stop error, "RAM ERROR" (error code: 1161) is detected. For the LCPU whose serial number (first five digits) is "14012" or later, the device information, which contains data change information, can be checked in SD927 and SD928. For details on the special register, refer to the following.
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(2) Execution timing

- Program: At program execution
- Parameter: When the CPU module is powered off and on, when the CPU module is reset, and when the CPU module is set from STOP to RUN after data are written to it
- Device memory: When device data are read

3.35 Program Cache Memory Auto Recovery Function

This function is to restore the error location automatically by using data in the program memory, which are stored in the flash ROM, when the memory check function (☞ Page 199, Section 3.34) detects an error in the program cache memory. This function enables the CPU module to continue its operation even if an error such as change of data in the program cache memory occurs due to noise.

Remark

Before executing this function, check the version of the CPU module used. (☞ Page 376, Appendix 2)

(1) Execution condition

The function is executed when the following conditions are all met.

- The CPU module is in RUN status.
- Data in the program memory match those in the program cache memory.

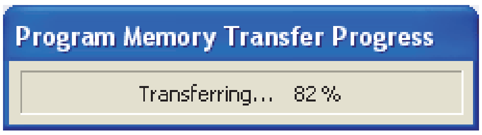
Note that the function is not executed in the following conditions and RAM ERROR (error code: 1160) is detected even though the above conditions are met:

- The change of data in the program cache memory was detected while the following operations were being performed with the CPU module during RUN.

Operation item	LCPU with a serial number (first five digits) of "13102" or later
Online change (ladder mode)	△*1
Password registration (program files)	○
Program memory batch download	△*1
Export to ROM format	○
Write to PLC (device comments)	△*1

○: The auto recovery processing is performed and no error occurs. △: An error occurs depending on the condition. ×: An error occurs.

*1 If the change of data is detected while data in the program cache memory are being transferred to the program memory (while the following window is being displayed on the programming tool), the auto recovery processing is not performed and an error occurs.



- The change of data in the program cache memory was detected by SFC program.
- The change of data in the program cache memory was detected by the dedicated instruction such as the S(P). instruction.
- The change of data in the program cache memory was detected by the rise instruction or the fall instruction.
- The data in the program cache memory was not restored due to the failure of the memory.

Point

To match the data in the program memory and those in the program cache memory, configure the setting to transfer the data of the program cache memory to the program memory from "Options" screen.*2 (☞ Page 157, Section 3.22.3 (5))

[Tool] ⇨ [Options]

*2 The transferring of the data in the program cache memory to the program memory is set by default.

Other than the above, you can also use "Program Memory Batch Download" to match the data in the program memory and those in the program cache memory.

(2) Execution timing

The execution timing of this function is described below.

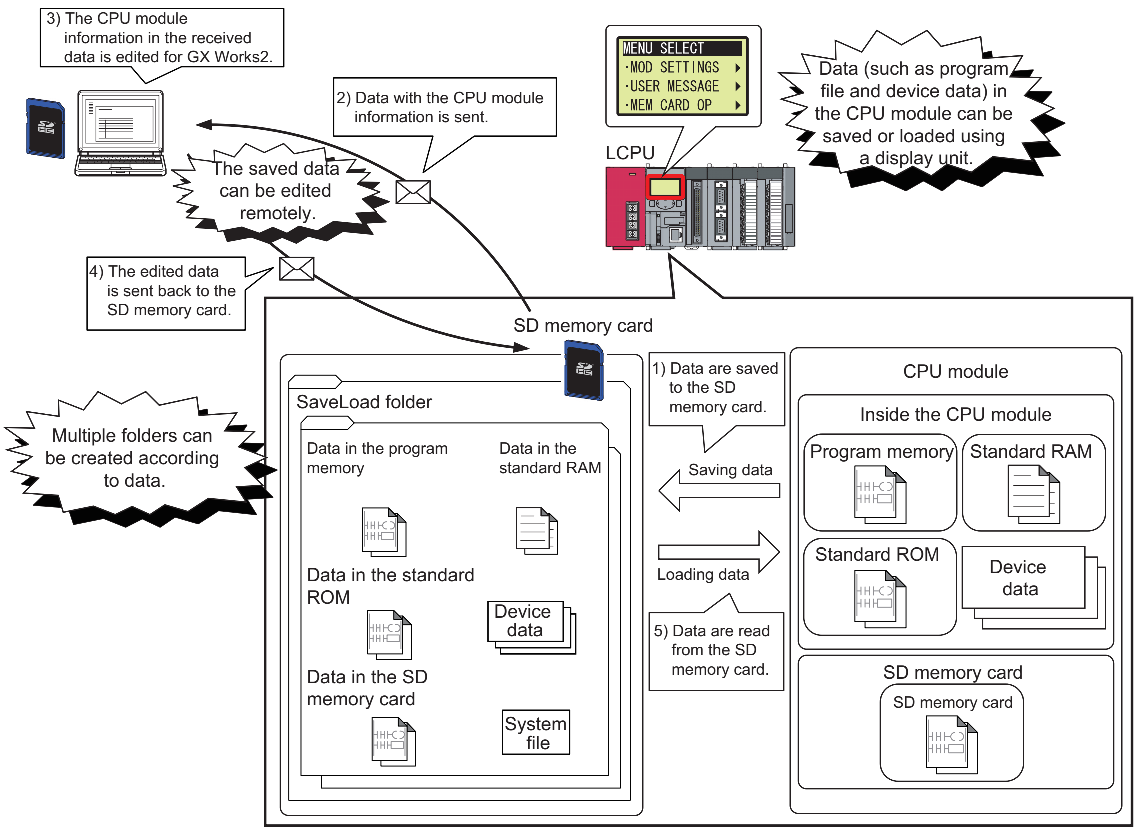
- When the program is executed
- When data are verified or data are read from the programmable controller
- When any one of the write operations listed under (1) in this section is performed*1

*1 To perform the auto recovery processing when a write operation is performed, use GX Works2 with version 1.80J or later. (GX Works2 must be used to perform the auto recovery processing at this timing because LCPU does not support GX Developer.)

3.36 Project Data Batch Save/Load Function

This function saves data in the CPU module to an SD memory card, and also reads the data saved in an SD memory card to the CPU module.

This function enables users to read/write data (such as program file and parameter file) without using a personal computer.



Remark

Before executing this function, check the version of the CPU module used. (☞ Page 376, Appendix 2)

The L02SCPU and L02SCPU-P do not support this function.

3.36.1 Batch save

This function saves data (such as program file and parameter file) in the CPU module to an SD memory card using a display unit.

(1) Execution method

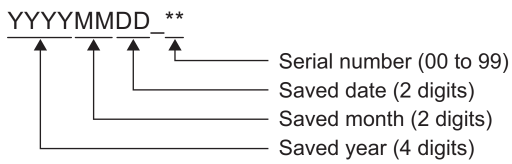
The function can be executed using a display unit. For the operating method of the display unit, refer to ☞ Page 280, Section 4.7.3.

(2) Folders to be used

When the function is executed, a new folder will be created under the "SaveLoad" folder within the root directory. If there is no "SaveLoad" folder, it will be created upon execution of the function.
The "SaveLoad" folder can store up to 100 folders.

(a) Folder name

The created folder is automatically named as follows.

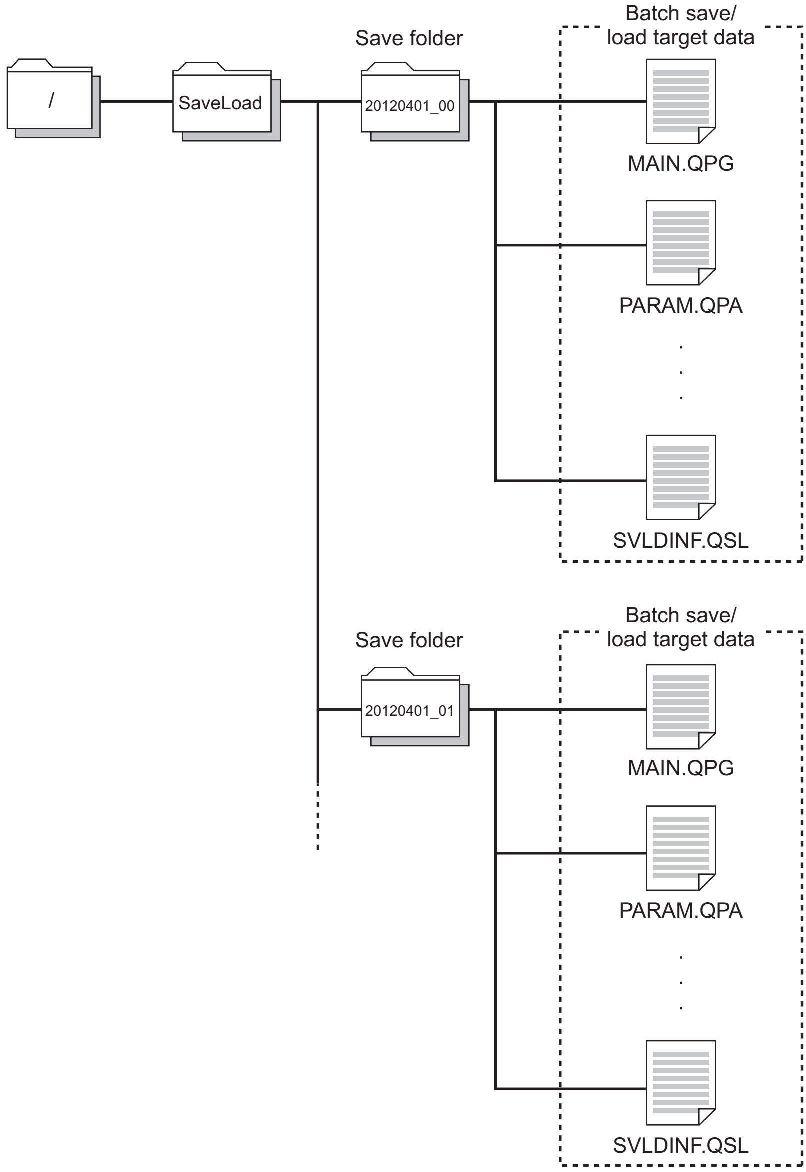


Remark

The range of the number at the end of the folder name is 00 to 99. The smallest number available will be assigned automatically. For example, if two folders, "20120401_00" and "20120401_02", have already existed and the batch save function is executed on April 2, 2012, the name of the new folder will be "20120402_01".

(b) Folder structure

The following shows the folder structure in the SD memory card.



(3) Target drives and data

(a) Target drives

The following drives are targeted.

- Drive 0 (program memory)*1
- Drive 2 (SD memory card)*2
- Drive 3 (standard RAM)
- Drive 4 (standard ROM)

*1 Data in the program cache memory are targeted.

*2 Files only within the root directory are targeted. (Note that the data logging setting file in the "LOGGING" folder is regarded as a target file.)

(b) Target data

The following table lists save-target data.

○: Saved, ×: Not saved

File type	File name and extension (any given name for ***)	Save status
Parameter	PARAM.QPA	○
Intelligent function module parameter	IPARAM.QPA	○
Program*2	***.QPG	○
Device comment	***.QCD	○
Initial device value	***.QDI	○
File register	***.QDR	○
Local device	***.QDL	○
Sampling trace file	***.QTD	○
PLC user data	***.CSV/BIN	○
Symbolic information	***.C32	○
Drive heading	QN.DAT	○
Device data storage file	DEVSTORE.QST	○
Module error collection file	IERRLOG.QIE	×
Data logging setting file	LOGCOM.QLG, LOG01 to LOG10.QLG	○
Data logging file	***.CSV	×
Menu definition file	MENUDEF.QDF	○
Predefined protocol setting file	ECPRTCL.QPT,CPRTCL.QPT	○
Latch data backup file	LCHDAT00.QBK	○
Backup file (for data backup to an SD memory card)	MEMBKUP0.QBP	○
Device data file*1	DEVDATA.QDT	○

*1 All device data set in the Device tab of the PLC parameter dialog box

*2 This file type is not saved when a block password for which "Execution Program Protection Setting" is enabled has been set.

Point

Up to 1024 files (all drives in total) can be saved with the batch save function.
If more than 1024 files are targeted, an error occurs.
When an error is detected, check the number of files in Drive 2 (SD memory card).

(4) Batch save status

The batch save status can be checked in SD635 (Project data batch save status).

SD635 value	Batch save status	Description
0_H	Batch save not executed	The batch save function is not executed.
1_H	Batch save being executed	The batch save function is being executed.
2_H	Batch save completed	The batch save processing is completed successfully.
FF_H	Batch save error	The batch save processing has failed and an error occurs.

(5) Causes of a batch save error

When the batch save processing has failed, a diagnostic error is not detected. An error message is displayed on the display unit and the error cause is stored in SD634 (Project data batch save error cause).

SD634 value	Description
100_H	• An SD memory card is not inserted. • The SD memory card lock switch of the CPU module is not slid down.
101_H	Use of an SD memory card is disabled by SM606 (SD memory card forced disable instruction).
200_H	• Size of data to be batch-saved exceeds the capacity of the SD memory card.*1 • The number of files to be batch-saved exceeds the number of storable files of the SD memory card.*1
201_H	The number of files to be batch-saved exceeds 1024.
202_H	More than 100 folders exist in the "SaveLoad" folder.
300_H	The SD memory card is write-protected.
400_H	Writing data to the SD memory card has failed.*2
401_H	The SD memory card was removed during the batch save processing.
500_H	Reading data from the target drive (program memory) has failed.*2
503_H	Reading data from the target drive (standard RAM) has failed.*2
504_H	Reading data from the target drive (standard ROM) has failed.*2
505_H	Reading data from the target drive (SD memory card) has failed.*2
510_H	Reading data from the target drive (system data) has failed.*2
600_H	The batch save function was executed during the latch data backup to the standard ROM.
601_H	The batch save function was executed during online change.
602_H	The batch save function was executed with an FTP client connected to and communicated with the CPU module.
604_H	The batch save function was executed while the CPU module change function with SD memory card was being executed.
607_H	The batch save function was executed while the iQ Sensor Solution function (data backup/restoration) was being executed.
608_H	The batch save function was executed while the file transfer function (FTP client) was being executed.
701_H	The batch save function was executed for the program for which a block password was set and "Execution Program Protection Setting" of the password was enabled.

*1 The number of save-target system files (up to five files) and its size (up to 10KB) are included.

*2 If the function is executed again and the same error occurs, the possible cause is a hardware failure of the CPU module or SD memory card. Please consult your local Mitsubishi representative.

(6) Precautions

(a) Precautions for executing the function simultaneously with another function

- The batch save function cannot be executed while any of the following function is being executed. If executed, an error occurs.
  - Latch data backup to standard ROM
  - Online change (Writing programs in RUN status)
  - File transfer function (FTP server)
  - File transfer function (FTP client)
  - CPU module change function with SD memory card
  - iQ Sensor Solution function (data backup/restoration)
- If the batch save function is executed while the data logging function is being executed, the sampling performance of the data logging function degrades. This may lead to data missing or increase the frequency of data missing.

(b) Precautions during execution

- Do not power off or reset the CPU module, or do not insert or remove the SD memory card while the function is being executed. The processing stops and project data cannot be saved properly.
- If save-target data are added or modified using a programming tool while the function is being executed, the saved data may differ from the original data or data inconsistency may occur. To prevent this, do not add or modify save-target data while the function is being executed. Also, set the operating status of the CPU module to STOP, and execute the batch save function.
- If the batch save function is executed when the CPU module is in STOP status, the following functions are suspended. After data are saved successfully, the functions automatically restart.
  - Refresh of a network module
  - Refresh of CC-Link IE Field Network Basic
  - Auto refresh of an intelligent function module
  - Simple PLC communication function (read/write)
- The following online functions cannot be executed during the batch save processing. If executed, an error response is returned to the request source.

Category	Function	Category	Function
Drive operation	Format PLC memory	Online change	Online change (ladder mode)
	Clear PLC memory (Clear all file registers)		Online change (multiple blocks)
	Write title		Change TC setting
	Program memory batch download		Online change (files)
	Arrange PLC memory	Trace	Sampling trace registration
File operation	Write to PLC	Data logging	Data logging registration
	Delete PLC data	Remote operation	Remote latch clear
	Write PLC user data	SM/SD operation	Latch clear by using the special relay and special register areas
	Delete PLC user data	Monitor	Monitor condition setup
	Password registration	Device test	Executional conditioned device test
	Latch data backup to standard ROM	Others	CPU module change function with SD memory card
File transfer function (FTP server)	For all operations and commands		Register/cancel display unit menu

(c) Predefined protocol setting file

If the communication protocol setting files are stored in Drive 2 (SD memory card) and Drive 4 (standard ROM), only the file stored in Drive 2 (SD memory card) is targeted for batch save.
When executing the batch save function, store the communication protocol setting file in the valid drive only.

(7) Editing data

Data saved in the SD memory card by the batch save function can be edited using a programming tool before the data are loaded.
For the editing method, refer to the following.
Manual for the programming tool used

Point

- When reading data from the SD memory card to a personal computer to edit the saved program file, read the parameter file together.
- Any file that has been added to the SD memory card without using a programming tool cannot be loaded.

(a) Data that are not editable

The following data cannot be edited using a programming tool.

- PLC user data
- Latch data backup file
- Backup file (for data backup to an SD memory card)
- Data logging setting file

3.36.2 Batch load

This function reads the data saved in an SD memory card to the CPU module.
The following two methods are available to execute the function.

- Batch load using a display unit
- Auto loading using the special register (SD909)

(1) Execution timing

(a) Batch load using a display unit

The function is executed when the operation is performed on the display unit.

(b) Auto loading using SD909

The function is executed when:

- the CPU module is powered off and on
- the CPU module is reset

(2) Batch load target folders

(a) Batch load using a display unit

Folders under the "SaveLoad" folder in the SD memory card are targeted.
The name of these folders can be customized within 48 one-byte (24 two-byte) characters.
If the number of characters for the name exceeds 48 (or 24), the file will be excluded from the load target.

(b) Auto loading using SD909

Folders named "AutoLoad**"[*1] under the "SaveLoad" folder in the SD memory card are targeted.

*1 When the folder number is specified at "**" within the range between 01 and 99, the desired folder can be loaded. Folders can also be loaded automatically without specifying the folder number.

Point

For the target folder structure and target data, refer to the following.

- Folder structure: ☞ Page 204, Section 3.36.1 (2) (b)
- Target data: ☞ Page 204, Section 3.36.1 (3)

(3) Procedure

For the execution procedure using a display unit, refer to ☞ Page 281, Section 4.7.4.

For auto loading, set a desired folder number in SD909 (Auto loading target folder number).
To execute auto loading without specifying the folder number, set 0 (default) in SD909. (The load target folder name is "AutoLoad".)
The auto loading procedures for each case are described below.

(a) When the folder number is not specified

1. **Save data into the SD memory card by the batch save function of a display unit or using GX Works2.**
2. **Change the name of the saved data folder to "AutoLoad" using a personal computer.**
3. **Power off and on or reset the CPU module.**
4. **Data in the "AutoLoad" folder are automatically loaded to the CPU module.**

Point

- Auto loading cannot be performed if the SD memory card is write-protected.
  Remove write protection from the SD memory card, and execute auto loading.
- When the "AutoLoad" folder exists, data in the folder are automatically loaded whenever the CPU module is powered off and on or is reset.
  To prevent this, change the folder name.
- When 0 (default) is set in SD909 and no "AutoLoad" folder exists, an error does not occur. This is because the CPU module determines that auto loading is disabled.
- If any folder or file (including read-only files) with the same name exists in the load-destination CPU module, data in the corresponding folder or file will be overwritten.

(b) When the folder number is specified

1. **Save data into the SD memory card by the batch save function of a display unit or using GX Works2.**
2. **Change the name of the saved data folder to "AutoLoad**" using a personal computer.**
   Specify any number between 01 and 99 at "**".
3. **Set the number assigned in Step 2 in SD909.**
   For example, when the folder name is "AutoLoad01", set "1" in SD909.
4. **Power off and on or reset the CPU module.**
5. **Data in the specified folder are automatically loaded to the CPU module.**

Point

- Auto loading cannot be performed if the SD memory card is write-protected.
  Remove write protection from the SD memory card, and execute auto loading.
- The value in SD909 returns to 0 (default) upon completion of auto loading.
  If the "AutoLoad" folder exists even after completion of auto loading, data in the folder will be automatically loaded again when the CPU module is powered off and on or is reset next time.
- If any folder or file (including read-only files) with the same name exists in the load-destination CPU module, data in the corresponding folder or file will be overwritten.

(4) Operation when the file password 32 is set

The following table shows the load availability when the file password 32 is set to the load-source or load-destination file.

○: Loaded, ×: Not Loaded

Password of load-source file	Load-destination file		Password match	Load availability
	Existence	Password		
Set	Exist	Set	Matched*1	○
			Not matched	×
		Not set	—	×
	Not exist	—	—	○
Not set	Exist	Set	—	×
		Not set	—	○
	Not exist	—	—	○

*1 If the file password 32s are set to more than one file, the passwords of all files have to be the same. Even one file has a different password, files cannot be loaded.

(5) Batch load status

The batch load status can be checked in SD637 (Project data batch load status).
In the case of auto loading, the status can also be checked with LED indication.

SD637 value	Batch load status	Description	LED indication
0_H	Batch load not executed	The batch load function is not executed.	MODE: On (green)
1_H	Batch load being executed	The batch load function is being executed.	The LED indication changes as follows at intervals of 800ms. 1) MODE: Flashing (orange), BAT.: On (green) ↓ 2) MODE: Flashing (orange), BAT.: On (green), USER: On (red) ↓ 3) MODE: Flashing (orange), USER: On (red)
2_H	Batch load completed	The batch load processing is completed successfully.	MODE: On (green)
FF_H	Batch load error	The batch load processing has failed and an error occurs.	MODE: On (green), ERR.: Flashing (red)

(6) Causes of a batch load error

When the batch load processing has failed, the error cause is stored in SD636 (Project data batch load error cause).

When the batch load function is executed using a display unit, an error message is displayed on the display unit.

SD636 value	Description
800_H	The model of the load-destination CPU module is different from that of the load-source CPU module.
801_H	Reading data from the SD memory card has failed.*1
802_H	The SD memory card was removed during batch load processing.
803_H	A data folder without a system file (SVLDINF.QSL) was loaded.
804_H	• The file password 32 of the load-destination file is different from that of the load-source file. • A file password 32 is not set to the load-source file while the load-destination file has a password.
805_H	• When the folder number (1 to 99) is set in SD909 (Auto loading target folder number) for auto loading, the folder with the corresponding number does not exist in the SD memory card. • The folder number out of the setting range (other than 0 to 99) is set in SD909 (Auto loading target folder number).
810_H	Loading data to the load-destination drive has failed.*1
820_H	Loading of the folder that includes a file that may be used in another application has failed.
821_H	Format was executed while any file that may be used in another application existed.
900_H	• When the batch load function is executed, the SD memory card is not inserted. • When the batch load function is executed, the SD memory card lock switch of the CPU module is not slid down. • When the folder number (1 to 99) is set in SD909 (Auto loading target folder number) for auto loading, the SD memory card is not inserted. • When the folder number (1 to 99) is set in SD909 (Auto loading target folder number) for auto loading, the SD memory card lock switch of the CPU module is not slid down.
901_H	Use of an SD memory card is disabled by SM606 (SD memory card forced disable instruction).
$A00_H$	• The size of load-target data is larger than the free space of each drive in the CPU module or SD memory card. • The number of files to be loaded exceeds the number of files can be stored in each drive in the CPU module or SD memory card.
$B00_H$	The SD memory card is write-protected.
$C00_H$	The batch load function was executed using a display unit during the latch data backup to the standard ROM.
$C01_H$	The batch load function was executed using a display unit during online change.
$C02_H$	The batch load function was executed using a display unit with an FTP client connected to and communicated with the CPU module.
$C04_H$	The batch load function was executed using a display unit while the CPU module change function with SD memory card was being executed.
$C07_H$	The batch load function was executed while the iQ Sensor Solution function (data backup/restoration) was being executed.
$C08_H$	The batch load function was executed while the file transfer function (FTP client) was being executed.
$C10_H$	The batch load function was executed using a display unit when the CPU module was in RUN or PAUSE status.

*1 If the function is executed again and the same error occurs, the possible cause is a hardware failure of the CPU module or SD memory card. Please consult your local Mitsubishi representative.

When the auto loading processing has failed, "LOAD ERROR" (error code: 2240 to 2248) occurs.

Error code	Error message	Error cause
2240	LOAD ERROR	The model of the load-destination CPU module is different from that of the load-source CPU module.
2241		Reading data from the SD memory card has failed.
2242		A system file (SVLDINF.QSL) does not exist in the load-target folder.
2243		The file password 32 of the load-destination file is different from that of the load-source file. Or, a file password 32 is not set to the load-source file while the load-destination file has a password.
2244		• When the folder number (1 to 99) is set in SD909 (Auto loading target folder number), a folder with the corresponding number does not exist in the SD memory card. • The folder number out of the setting range (other than 0 to 99) is set in SD909 (Auto loading target folder number).
2245		Loading data to the load-destination drive has failed.
2246		• When the folder number (1 to 99) is set in SD909 (Auto loading target folder number), the SD memory card is not inserted. • When the folder number (1 to 99) is set in SD909 (Auto loading target folder number), the SD memory card lock switch of the CPU module is not slid down.
2247		• The size of load-target data is larger than the capacity of the CPU module memory or SD memory card. • The number of files to be loaded exceeds the number of files can be stored in the CPU module or SD memory card.
2248		Auto loading was executed with the SD memory card write-protected.

(7) Precautions

(a) Precautions for selecting CPU module models

If the model of the load destination CPU module is different from that of the load source CPU module, the batch load function cannot be executed. However, the batch load function can complete normally in the following combinations of CPU modules.

- L02CPU and L02CPU-P
- L06CPU and L06CPU-P
- L26CPU and L26CPU-P
- L26CPU-BT and L26CPU-PBT

(b) Precautions for executing the function simultaneously with another function

- If data are saved into the SD memory card while the sampling trace function is being executed, the load-destination CPU module may misunderstand that a request is from a different source when trace start/suspension or trigger execution is requested. An alarm message is displayed at this time. However, there is no impact on the operation of the sampling trace function.
- The batch load function cannot be executed while any of the following function is being executed. If executed, an error occurs.
  - Latch data backup to standard ROM
  - Online change (Writing programs in RUN status)
  - File transfer function (FTP server)
  - File transfer function (FTP client)
  - CPU module change function with SD memory card
  - iQ Sensor Solution function (data backup/restoration)
- Do not set auto loading simultaneously with the following functions. If set, auto loading is not executed.
  - Boot operation
  - Data restoration by the CPU module change function with SD memory card[*1]
  - Latch data backup to standard ROM

*1 This applies only when automatic restoration is set.

(c) Precautions for editing data

- If a file in the load-target folder is deleted, the function corresponding to the deleted file cannot continue the same execution status after data are loaded.
- If a system file (SVLDINF.QSL) in a data folder is deleted, the corresponding folder cannot be loaded. Do not delete the system file of the load target folder.

(d) Precautions for parameter settings

- If the batch load function is executed without formatting Drive 0 (program memory) of the load-destination CPU module, the setting that has been configured originally to the load-destination CPU module is used for the "High Speed Monitor Area from Other Station" parameter on the Boot File tab of the PLC parameter dialog box.

(e) Precautions during execution

- Do not power off or reset the CPU module while the function is being executed. If performed, the processing stops and data cannot be loaded properly.
- If the batch load function is executed using a display unit, the following functions are suspended. After data are loaded successfully, the functions automatically restart.
  - Refresh of a network module
  - Refresh of CC-Link IE Field Network Basic
  - Auto refresh of an intelligent function module
  - Simple PLC communication function (read/write)
- Even if the RUN/STOP/RESET switch of the CPU module is switched from STOP to RUN, or the remote RUN or remote PAUSE function is executed while the function is being executed, the operating status of the CPU module remains in STOP. If the operating status is changed during the batch load processing, the status changes after the processing is completed.
- If the CPU module detects any file that may be used in another application during the batch load processing, an error occurs at the following timing or due to the following reason.
  - Timing: while drives are being formatted (if the format is executed)
  - Reason: files other than the one in use are loaded
- The online functions that cannot be executed during the batch load processing are the same as those that cannot be executed during the batch save processing. (☞ Page 207, Section 3.36.1 (6) (b))

3.37 Predefined Protocol Function

This function sends and receives packets predefined by using GX Works2, enabling easy communications with external devices (such as measuring instruments and bar code readers).
Data are sent and received by registering the protocol setting data to the CPU module, and executing the program for starting data communications. Use of the function helps to reduce the significant number of program steps since no program is required to communicate with external devices.
In addition, the CPU module can send and receive data that vary in each communication as well as fixed data (such as numerical values and codes) since devices can be included in the communication packets.

Remark

Before executing this function, check the version of the CPU module used. (☞ Page 376, Appendix 2)

3.37.1 Communications via Ethernet

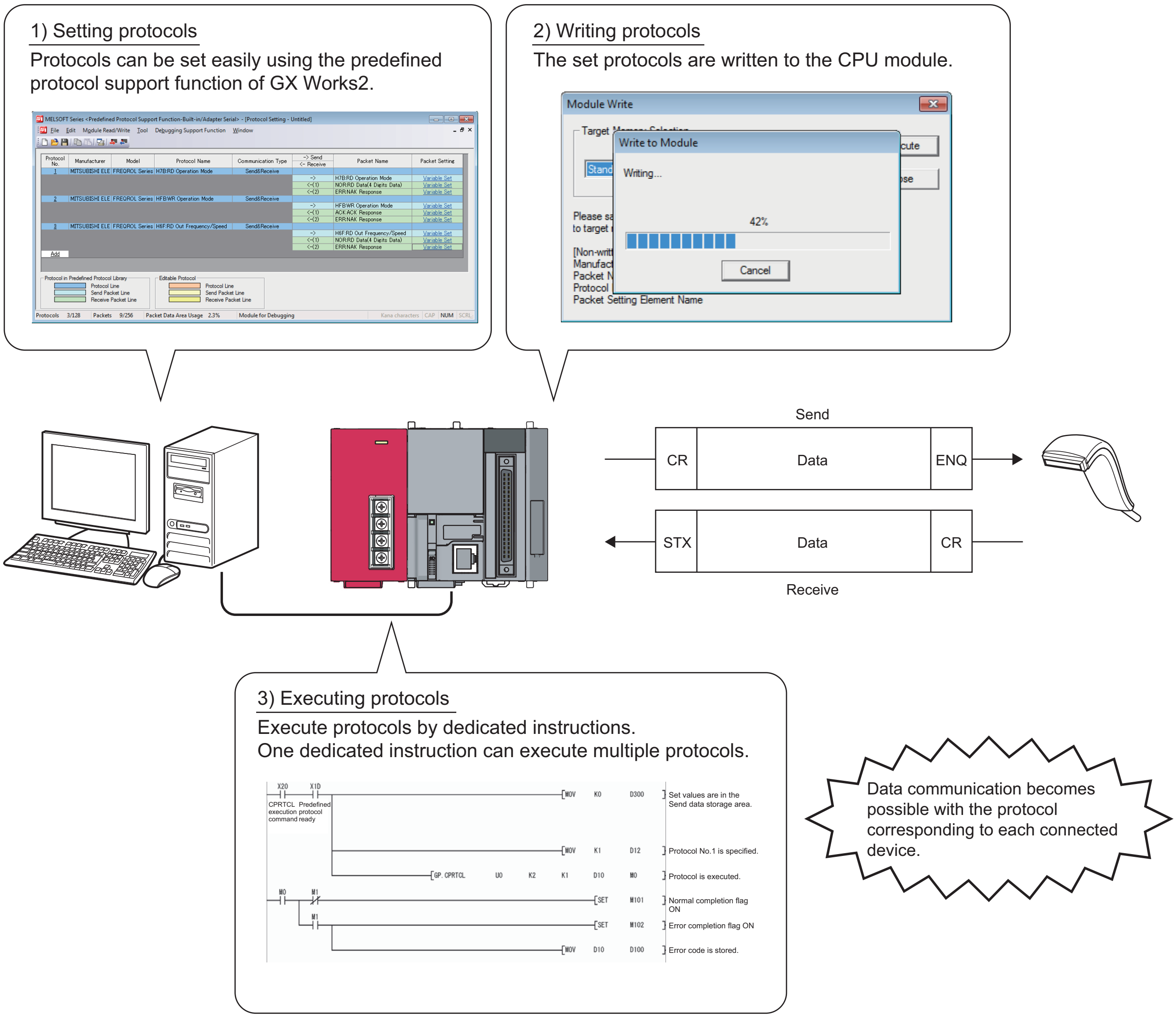
The CPU module communicates with external devices via the built-in Ethernet ports.

For details on communications via Ethernet, refer to the following.
MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)

3.37.2 Communications via RS-232 and RS-422/485

The CPU module communicates with external devices via the RS-232 interface of the CPU module, RS-232 adapter, or RS-422/485 adapter.

Set protocols required for data communications with external devices using the predefined protocol support function of GX Works2.
Protocols can be either selected from the predefined protocol library, or created and edited by users.



Remark

The CPU module with the RS-232 interface and the LCPU where the RS-232 adapter or RS-422/485 adapter can be mounted support this function. (☞ Page 376, Appendix 2)

(1) Specifications

(a) Transmission specifications

The following table lists the transmission specifications.

Item		Specifications
Communication method		Full-duplex communication
Synchronization method		Asynchronous method
Transmission speed	RS-232 interface of CPU module, RS-232 adapter	9600bps, 19200bps, 38400bps, 57600bps, 115200bps
	RS-422/485 adapter	1200bps, 2400bps, 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps
Data format	Start bit	1
	Data bit	• 7 • 8
	Parity bit	• Available • Not available
	Stop bit	• 1 • 2
Error detection	Parity check	• Available (Select odd or even number in parameter.) • Not available
	Sumcheck	• Available (Determine whether to use sumcheck codes and check method according to the protocol used.) • Not available
Line configuration (connection)	RS-232 interface of CPU module, RS-232 adapter	1:1
	RS-422/485 adapter	1:1, 1:n, n:1, m:n
Line configuration (data communication)	RS-232 interface of CPU module, RS-232 adapter	1:1
	RS-422/485 adapter	1:1, n:1

(b) Functional specifications

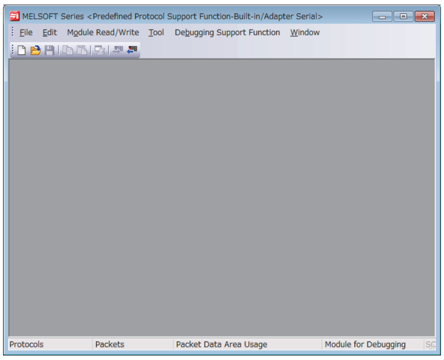
The following table lists the functional specifications.

Item		Specifications
Protocol setting data	Number of protocols*1	Up to 128
	Number of packets*2	Up to 256
	Packet area data size*3	Up to 12288 bytes
Available channel		CH1 (fixed)
Protocol execution method		S(P).CPRTCL instruction
Length of data that can be sent or received at a time		Up to 2048 bytes

*1 Number of protocols specified as the protocol setting data
*2 Total number of packets set to each protocol
*3 Size as a sum of all packets

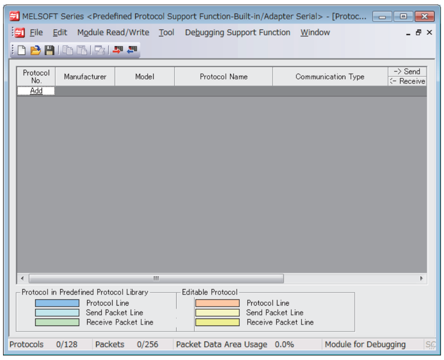
(2) Setting method

The setting required for using the predefined protocol function is described below.



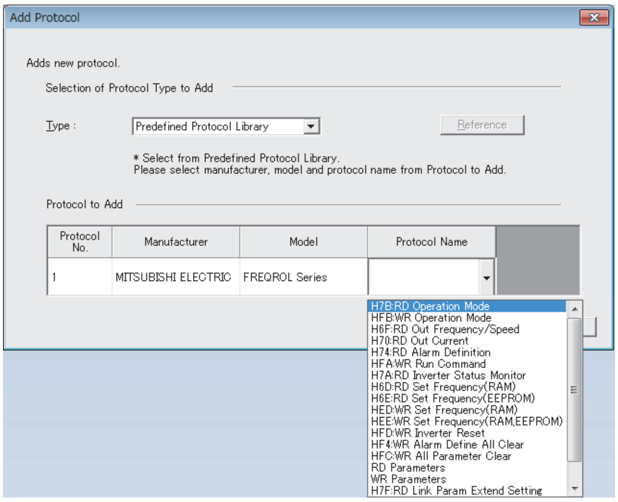
1. **Open the "Predefined Protocol Support Function" screen.**

[Tool] ⇨ [Predefined Protocol Support Function] ⇨ [Built-in/Adapter Serial]



2. **Create a new file.**

[File] ⇨ [New] ⇨ "Add"



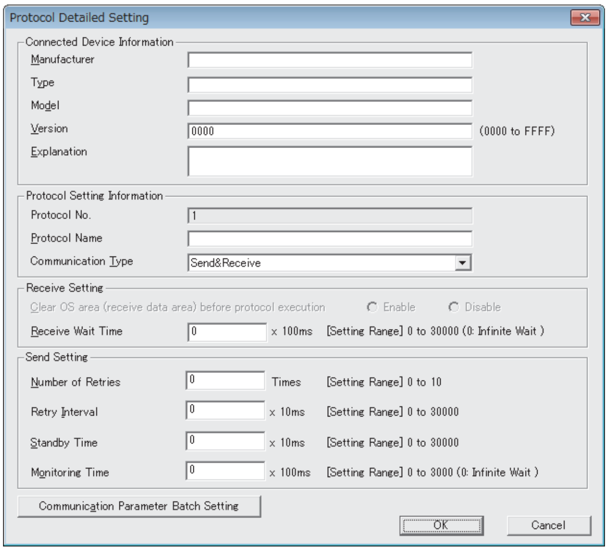
3. **In the "Add Protocol" screen, select "Predefined Protocol Library" or "Add New".**

When "Predefined Protocol Library" is selected, select a desired protocol from the library registered in GX Works2.

Point

Select "User Protocol Library" in the "Add Protocol" screen to read the protocol stored by the user. For details on the user protocol library, refer to the following.

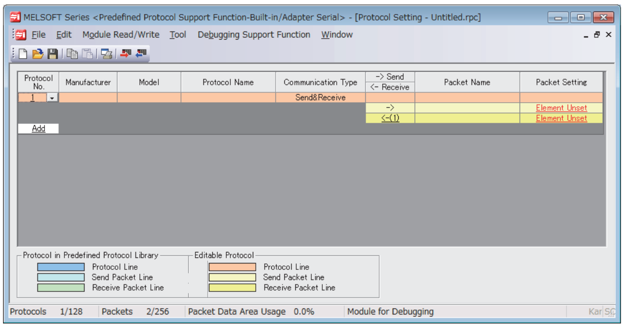
GX Works2 Version 1 Operating Manual (Intelligent Function Module)



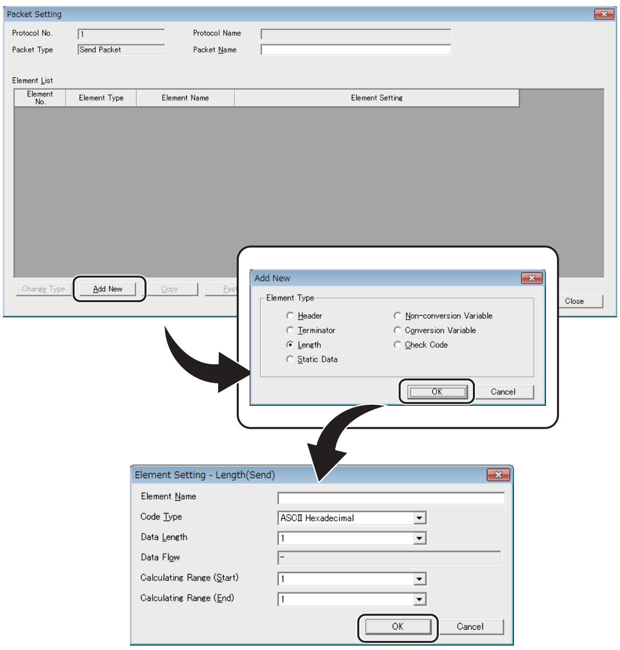
4. **Set the items required for data communications.**

In the "Protocol Detailed Setting" screen, set the communication parameters for the protocol.

"Protocol Setting" ⇨ screenSelect protocol ⇨ [Edit] ⇨ [Protocol Detailed Setting]



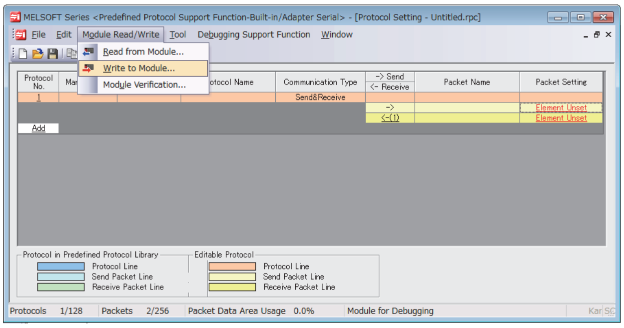
↓



5. **Set the packet configuration.**

In the "Packet Setting" screen, set the configuration of the packet to be sent or received.

"Protocol Setting" screen⇨"Variable unset" or "Element unset"



6. Write the protocol setting data to the CPU module.

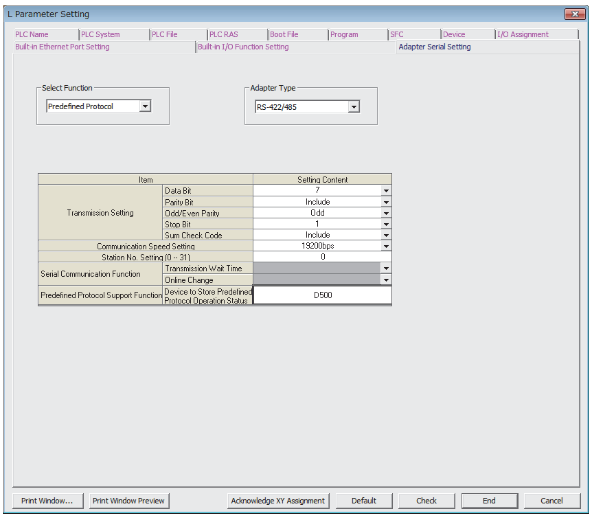
In the "Writing Protocol Setting" screen, specify the write-target drive in the CPU module and write the protocol setting data.*1

[Module Read/Write] ⇨ [Write to Module]

*1 The written protocol setting data will be enabled when
- the CPU module is powered on or is reset, or
- SM1333 (Predefined protocol setting check request) is turned on.

Point

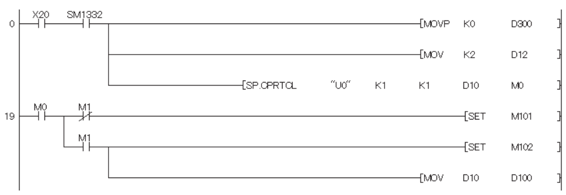
- To enable the written protocol setting data without powering off or resetting the CPU module, turn on SM1333 (Predefined protocol setting check request). Before enabling the protocol setting data, stop the CPU module and check that no instruction is being executed. Depending on the on-timing of SM1333, the instruction being executed may end abnormally.



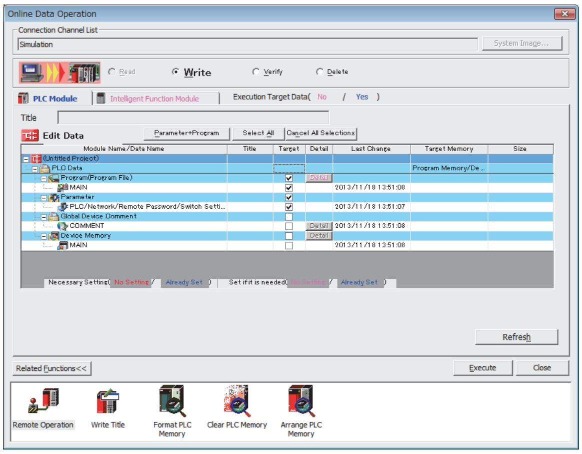
7. Set the PLC parameters.

Set "Predefined Protocol" to the "Select Function" parameter on the "Adapter Serial Setting" tab or "Built-in Serial Setting" tab.

"PLC Parameter Setting" screen⇨"Adapter Serial Setting" or "Built-in Serial Setting"



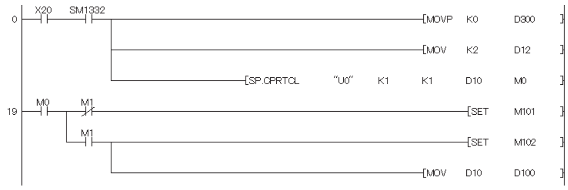
8. Create a program for starting data communications. Use the S(P).CPRTCL instruction.



9. Write the program to the CPU module.

[Online] ⇨ [Write to PLC]

10. Execute the program written to the CPU module by using the S(P).CPRTCL instruction.



For the protocol setting method, refer to the following.

📖GX Works2 Version 1 Operating Manual (Intelligent Function Module)

(3) Setting items of the predefined protocol support function

(a) Communication type

There are three protocol communication types: "Send", "Receive", and "Send & Receive".
For details on the protocol communication types, refer to the following.
MELSEC-L Serial Communication Module User's Manual (Basic)

(b) Packet elements set for "Packet Setting"

The packet elements include "Header", "Terminator", "Length", "Static Data", "Non-conversion Variable", "Conversion Variable", "Check Code", and "Non-verified Reception".
Up to 32 elements can be set in a single packet.

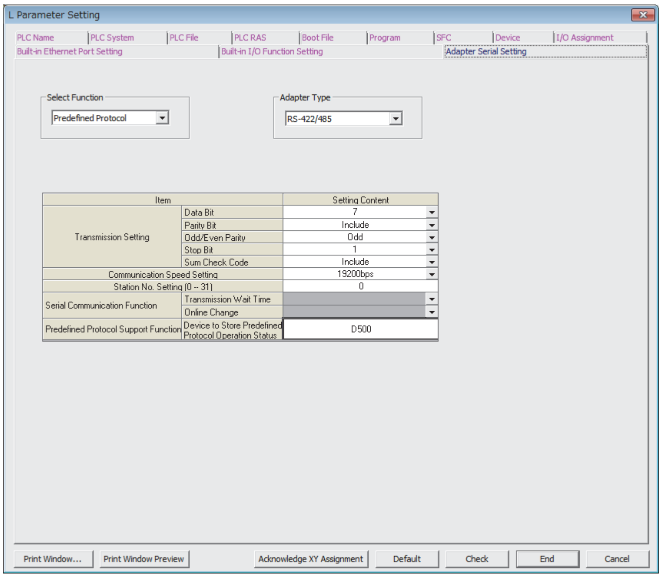
The protocol setting method and packet elements are the same as those for the serial communication module, except for some differences. (MELSEC-L Serial Communication Module User's Manual (Basic))
The following are the differences.

- Read the C24 (serial communication module) as the CPU module in the manual.
- Buffer memory cannot be set as a data storage area for conversion variables and non-conversion variables.

(4) Setting items of "Adapter Serial Setting" or "Built-in Serial Setting"

Setting items of "Adapter Serial Setting" or "Built-in Serial Setting", (items required for data communications using the predefined protocol function) are described below. For other setting items, refer to the following.

☞ Page 353, Appendix 1.2



(a) Select function

Select "Predefined Protocol".

(b) Predefined protocol support function (Device to store predefined protocol operation status)

Set the start device for storing the predefined protocol operating status. The following information is stored in the area of three words starting from the selected device.

Device for storing the predefined protocol operating status (offset)	Item	Description*1	Default
+0	Protocol cancellation	The protocol cancellation request status is stored. 0: No cancellation requested 1: Cancellation requested (set by users) 2: Cancellation completed (set by the CPU module)	0
+1	Protocol execution status	The execution status of the predefined protocol function is stored. 0: Not executed 1: Waiting for data to be sent 2: Data being sent 3: Waiting for data to be received 4: Data being received 5: Execution completed	
+2	Number of protocol executions	The number of protocol executions is stored. 0: No execution 1 to 65535: Number of executions (The value remains the same after 65535.)	

*1 Even after data communications using the predefined protocol function (executed by the S(P).CPRTCL instruction) is completed, the stored values are held.

(5) S(P).CPRTCL instruction

This instruction sends and receives packets set by the predefined protocol support function of GX Works2. Usage of the S(P).CPRTCL instruction is the same as that of the G(P).CPRTCL instruction for the serial communication module, except for some differences. For the G(P).CPRTCL instruction, refer to the following.

MELSEC-L Serial Communication Module User's Manual (Basic)

The following are the differences.

Item	Difference
Setting data Un	U0 (dummy) needs to be specified. Since this is a dummy, no change is required when replacing from the G(P).CPRTCL instruction of the serial communication module.
Setting data n1	Setting data n1 is fixed to "1: Channel 1 (CH1 side)".
Cancellation of protocol execution	Protocol execution can be cancelled by using the device for storing the predefined protocol operating status (offset+0) set in the "Adapter Serial Setting" or "Built-in Serial Setting" tab of the PLC parameter dialog box, instead of the buffer memory.
Functional protocol	The predefined protocol function of the CPU module does not support the functional protocol.

(6) Execution conditions

The predefined protocol function can be executed when the protocol setting data is enabled and SM1332 (Predefined protocol ready) is on. The written protocol setting data will be enabled when

- the CPU module is powered on or is reset, or
- SM1333 (Predefined protocol setting check request) is turned on.

SM1332 is used as an interlock signal to execute the protocol.

(a) Operation of SM1332

SM1332 (Predefined protocol ready) turns on when the protocol setting data is normal, and remains off when the protocol setting data is abnormal or not written.

If the protocol setting data is abnormal, the following occurs.

- The error details are stored in SD1337 to SD1340 (Predefined protocol setting data error information).
- The error code is stored in SD1351 (Predefined protocol function error code).

Registration of the protocol setting data can be checked in the following special register areas.

- SD1341 (Number of protocols registered)
- SD1342 to SD1349 (Protocol registration status)

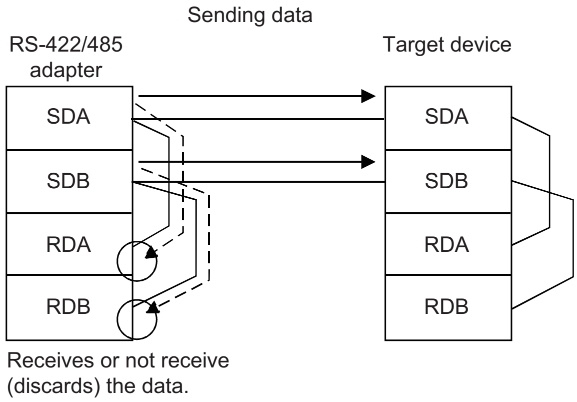
(b) Changing the protocol setting data without powering off or resetting the CPU module

The protocol setting data can be changed by turning on SM1333 (Predefined protocol setting check request). After SM1333 turns on, the following occurs.

- SM1332 (Predefined protocol ready) turns off.
- If the protocol setting data is normal, SM1332 turns on, and SM1333 turns off.
- If the protocol setting data is abnormal, SM1332 does not turn on, and SM1333 turns off. The error information is stored in SD1337 to SD1340 (Predefined protocol setting data error information) and SD1351 (Predefined protocol function error code).

(7) Enabling/disabling echo back of RS-422/485 adapter

When data are communicated via RS-485 (two-wire type) in the RS-422/485 adapter, the sent data can be echoed back to the RDA and RDB of own station.
The echo back status (receiving the sent data or not (discard the data)) can be specified.



(a) Setting method

1. **Turn on SM1334 (RS-422/485 echo back specification). (Disable echo back.)**
2. **The CPU module starts sending data.**

(b) Precautions

- The setting of SM1334 (RS-422/485 echo back specification) is enabled when a send request is received. Even if the setting is changed during the send processing, the change is not reflected to the operation.
- SM1334 can be set only when the RS-422/485 adapter is used.

(8) Operation image and data structure of the predefined protocol function

For operation image and data structure of the predefined protocol function, refer to the following. (Read the C24 (serial communication module) as the CPU module.)

MELSEC-L Serial Communication Module User's Manual (Basic)

(9) Precautions

(a) To use two predefined protocol functions simultaneously (Ethernet or RS-232/422/485)

Store the predefined protocol setting files in the same drive. If the setting files are stored separately in the SD memory card and the standard ROM, the setting stored in the SD memory card is enabled while the setting stored in the standard ROM is disabled.
To enable the predefined protocol setting stored in the standard ROM, format the SD memory card or delete the predefined protocol setting stored in the SD memory card. Then, enable the setting.
The predefined protocol setting can be checked and deleted by operating the PLC user data.

[Online] ⇨ [PLC User Data] ⇨ [Read] or [Delete]

(b) After writing the predefined protocol setting

Power on or reset the CPU module, or turn on SM1333 (Predefined protocol setting check request) to enable the written protocol setting data. If the written data are not enabled, the data may be changed unintentionally when the CPU module is powered on or is reset.

(c) Device data

Data set by the predefined protocol function are communicated during the service processing. Therefore, if the COM instruction is executed while b15 of SD778 (Refresh processing selection when the COM/CCOM instruction is executed) is on, data communications are also performed at the execution of the COM instruction.

(d) S(P).CPRTCL instruction

Even though the instruction has been completed successfully, an error code may be stored in SD1351 (Predefined protocol function error code). If data communications fail even though the instruction has been completed successfully, check SD1351 and take corrective actions.
The following are the error codes that may be stored.

- Overrun error ($7F67_H$)
- Framing error ($7F68_H$)
- Parity error ($7F69_H$)
- Buffer full error ($7F6A_H$)

(e) Time settings of the protocol detailed setting

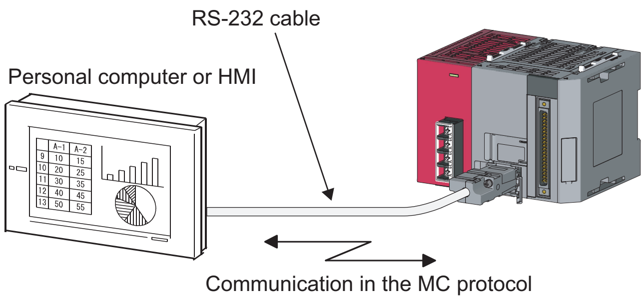
Since the passage of time, such as receive wait time, standby time, and monitoring time, is checked during the service processing, the set time may be exceeded depending on the scan time. Set the time values considering the scan time.

3.38 Serial Communication Function

This function communicates data using the MC protocol by connecting the RS-232 interface of the CPU module, the RS-232 adapter, or the RS-422/485 adapter to the following devices.

- MC protocol-compatible devices
- Personal computer
- HMI from other companies

This section describes the specifications, settings, and details of the function.



Remark

The CPU module with the RS-232 interface and the LCPU where the RS-232 adapter or RS-422/485 adapter can be mounted support this function. (☞ Page 376, Appendix 2)

Point

- A personal computer or an HMI from other companies can communicate only with a CPU module connected to it, the RS-232 adapter, and the RS-422/485 adapter. Communication with other stations in the network is not available.
- For details on MC protocol, refer to the following.
  MELSEC Communication Protocol Reference Manual
- The RS-422/485 adapter can be used to connect multiple external devices (multidrop connection).

(1) Specifications

(a) Transmission specifications

The following table lists the transmission specifications used for the serial communication function. Check that the specifications of the personal computer or HMI from other companies used match those in the following table.

Item	Setting range	Default
Communication method	—	Full-duplex communication
Synchronization method	—	Asynchronous method
Transmission speed[*1]	• When RS-232 interface of CPU module or RS-232 adapter is used: 9600bps, 19200bps, 38400bps, 57600bps, 115200bps • When RS-422/485 adapter is used: 1200bps, 2400bps, 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bps	19200bps
Data format	—	• Start bit: 1 • Data bit: 8 • Parity bit: Odd • Stop bit: 1
MC protocol format[*3] (auto identification)	—	• Format 4 (ASCII) • Format 5 (binary)
Frame[*3]	—	• QnA-compatible 3C frame • QnA-compatible 4C frame
Transmission control	—	DTR/DSR flow control[*2]
Sumcheck code[*1]	Include/Not include	Include
Transmission wait time[*1]	No wait time, 10 to 150ms (in increments of 10ms)	No wait time
Online change[*1]	Enable/Disable	Disable
Overall cable distance	—	15m
Line configuration (connection, data communication)	• When RS-232 interface of CPU module or RS-232 adapter is used: 1:1 • When RS-422/485 adapter is used: 1:1, 1:n, n:1, m:n	—

*1 The item is set in the PLC parameter of the programming tool. (☞ Page 370, Appendix 1.2 (13), Page 372, Appendix 1.2 (14))

*2 This control is performed only when data are communicated via RS-232.

*3 Relationship between the MC protocol formats and frames is shown below.

○: Available, ×: Not available

Function		Format 4	Format 5
Communication using ASCII code	QnA-compatible 3C frame	○	×
	QnA-compatible 4C frame	○	×
Communication using binary code	QnA-compatible 4C frame	×	○

(2) Commands

The following table lists the MC protocol commands that can be executed.

Function			Command	Description of processing	Number of processing points
Device memory	Batch read	In units of bits	0401 (00□1)	Reads out a bit device in units of one point.	• ASCII: 3584 points • BIN: 7168 points
		In units of words	0401 (00□0)	Reads out a bit device in units of 16 points.	480 words (7680 points)
				Reads out a word device in units of one point.	480 points
	Batch write*1	In units of bits	1401 (00□1)	Writes into a bit device in units of one point.	• ASCII: 3584 points • BIN: 7168 points
		In units of words	1401 (00□0)	Writes into a bit device in units of 16 points.	480 words (7680 points)
				Writes into a word device in units of one point.	480 points
	Random read*2*3	In units of words	0403 (00□0)	Reads out a bit device in units of 16 or 32 points by specifying a device/device number at random.	96 points
				Reads out a word device in units of one or two points by specifying a device/device number at random.	
	Test*1 (random write)	In units of bits	1402 (00□1)	Sets/resets a bit device in units of one point by specifying a device/device number at random.	94 points
		In units of words*2	1402 (00□0)	Sets/resets a bit device in units of 16 or 32 points by specifying a device/device number at random.	*5
				Writes into a word device in units of one or two points by specifying a device/device number at random.	
	Monitor registration*2*3*4	In units of words	0801 (00□0)	Registers a bit device to be monitored in units of 16 or 32 points.	96 points
				Registers a word device to be monitored in units of one or two points.	96 points
	Monitor	In units of words	0802 (00□0)	Monitors a device registered.	Number of monitor registration points

*1 To perform online change, check the "Permit" checkbox under "Online Change".

*2 Devices such as TS, TC, SS, SC, CS, and CC cannot be specified in units of words.
For the monitor registration, an error (4032_H) occurs during the monitor operation.

*3 The monitor condition specification cannot be used for these commands.

*4 Do not execute monitor registration from multiple external devices. If executed, the last monitor registration becomes valid.

*5 Set the number of processing points within the range of the following calculation formula.
(number of word access points) × 12 + (number of double word access points) × 14 ≤ 960
- One point of a bit device corresponds to 16 bits for word access or to 32 bits for double word access.
- One point of a word device corresponds to one word for word access or to two words for double word access.

(3) Accessible devices

The following table lists accessible devices with the serial communication function.

Classification	Device		Device code*1		Device number range	
			ASCII	Binary		
Internal system device	Function input		—	—	(Cannot be accessed)	Hexadecimal
	Function output		—	—		Hexadecimal
	Function register		—	—		Decimal
	Special relay		SM	91_H	Within the device number range of the CPU module accessed. Note, however, that local devices cannot be accessed.	Decimal
	Special register		SD	$A9_H$		Decimal
Internal user device	Input		X *	$9C_H$		Hexadecimal
	Output		Y *	$9D_H$		Hexadecimal
	Internal relay		M *	90_H		Decimal
	Latch relay		L *	92_H		Decimal
	Annunciator		F *	93_H		Decimal
	Edge relay		V *	94_H		Decimal
	Link relay		B *	$A0_H$		Hexadecimal
	Data register		D *	$A8_H$		Decimal
	Link register		W *	$B4_H$		Hexadecimal
	Timer	Contact	TS	$C1_H$		Decimal
		Coil	TC	$C0_H$		
		Current value	TN	$C2_H$		
	Retentive timer	Contact	SS	$C7_H$		Decimal
		Coil	SC	$C6_H$		
		Current value	SN	$C8_H$		
	Counter	Contact	CS	$C4_H$		Decimal
		Coil	CC	$C3_H$		
		Current value	CN	$C5_H$		
	Link special relay		SB	$A1_H$		Hexadecimal
	Link special register		SW	$B5_H$		Hexadecimal
	Step relay		S *	98_H		Decimal
	Direct input*2		DX	$A2_H$		Hexadecimal
	Direct output*2		DY	$A3_H$		Hexadecimal
Index register	Index register		Z *	CC_H	Within the device number range of the CPU module accessed	Decimal
File register	File register		R *	AF_H		Decimal
			ZR	$B0_H$		Decimal
Extended data register	Extended data register		D *	$A8_H$	• Binary: Within the device number range of the CPU module accessed • ASCII: 000000 to 999999 (up to 976.6K points)	Decimal
Extended link register	Extended link register		W *	$B4_H$	Specify within the device number range of the access destination CPU module.	Hexadecimal

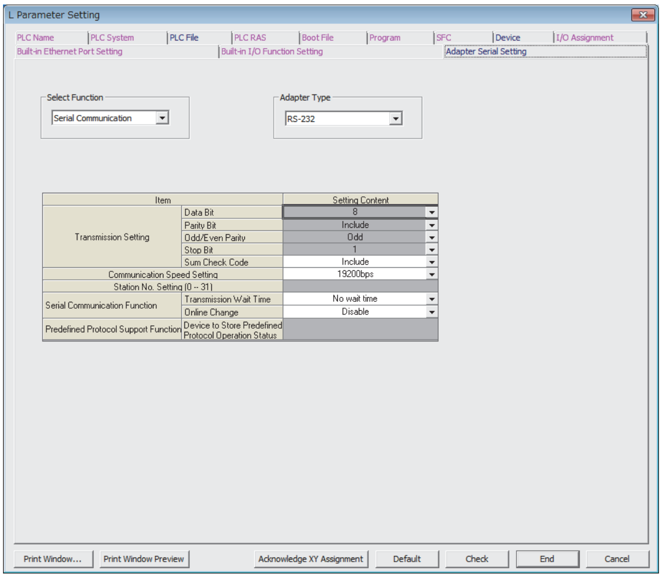
1 This is a code specified in MC protocol messages. When communicating data in ASCII code, specify the device code in two characters. If the code consists of only one character, add "" (ASCII code: $2A_H$) or a space (ASCII code: 20_H) after the character.

*2 Devices of DX/DY1000 or later are not available. Use X/Y devices to access devices of X/Y1000 or later.

(4) Setting transmission specifications

Set the transmission specifications of the serial communication function in the "Built-in Serial Setting" or "Adapter Serial Setting" tab of the PLC parameter dialog box. (☞ Page 353, Appendix 1.2)

- Set "Serial Communication" to the "Select Function" in the "Built-in Serial Setting" or "Adapter Serial Setting" tab.
- Set the adapter type*1, sum check code, communication speed setting, station number setting*2, transmission wait time, and online change status.



*1 The adapter type can be set only in the "Adapter Serial Setting" tab.
*2 The station number can be set only when the RS-422/485 adapter is used.

(5) Precautions

(a) Switching a connection from an HMI from other companies to a programming tool

A connection device can be switched from a personal computer or an HMI from other companies to a programming tool during communication. However, this operation causes a communication error in the personal computer or HMI. For a startup method of the personal computer or HMI after it is reconnected to the CPU module, refer to the manual for the device used.

(b) Transmission speed set in the Transfer setup screen

When "Use Serial Communication" is selected, the transmission speed set in the Transfer setup screen of the programming tool is ignored.

(c) Communication error

If any of the following conditions is met, no response is returned (a communication error occurs). Take a corrective action.

- The serial communication function is set not to be used.
- Communication is made at different transmission speed and data format.
- A frame to be sent has no correct starting end or terminal.
  - · 3C frame format 4: ENQ/CR + LF
  - · 4C frame format 4: ENQ/CR + LF
  - · 4C frame format 5[*1]: DLE + STX/DLE + ETX

*1 When the "Sum Check" checkbox is selected, the sumcheck code is included.
- The frame identification number of a frame to be sent is incorrect.
- The number of transmission bytes is under the header part size.

(6) Error codes during communication with the serial communication function

The following table lists the error codes (together with their descriptions and corrective actions) sent from the CPU module to the external device when an error occurs during communication using the serial communication function.

Error code (hexadecimal)	Error item	Description	Corrective action
4000_H to $4FFF_H$	—	Error detected by the CPU module (error occurred by other than the serial communication function)	Refer to the following manual and take corrective actions. MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)
7155_H	Unregistered monitor error	A monitor request was given before monitor registration.	Give a monitor request after registering a device to be monitored.
7157_H	Request target specification error	A CPU module that does not support the serial communication function is specified as a request target module or is in the specified route.	Check that the transmission message is addressed to the CPU module that supports the serial communication function. If not, correct the address and restart communication.
7164_H	Request data error	The requested data or device specification method is wrong.	Check the sent message/requested data of the external device, correct it, and restart communication.
7167_H	Disabled during RUN	A write command was specified while online change is disabled.	• Enable the online change and restart communication. • Set the CPU module to STOP and restart communication.
$7E40_H$	Command error	A command or a subcommand that does not exist is specified.	Check and correct the sent message of the external device, and restart communication.
$7E41_H$	Data length error	The number of points specified for random write/read exceeds the number of points enabled for communication.	Check and correct the sent message of the external device, and restart communication.
$7E42_H$	Data count error	The requested number of points exceeds the range of the command.	Check and correct the sent message of the external device, and restart communication.
$7E43_H$	Device error	The device specified does not exist. The device specified cannot be specified by the corresponding command.	Check and correct the sent message of the external device, and restart communication.
$7E47_H$	Continuous request error	The next request was received before the response message was returned.	Do not give continuous requests from the external device. Match the monitoring time of Timer 1 with the time-out time of the external device.
$7F21_H$	Receive header section error	The command (frame) section specified is in error. The ASCII code received cannot be converted into binary.	Check and correct the sent message of the external device, and restart communication.
$7F22_H$	Command error	The command or device specified does not exist.	Check and correct the sent message of the external device, and restart communication.
$7F23_H$	MC protocol message error	The data (such as ETX and CR + LF) specified after the character part does not exist or in error.	Check and correct the sent message of the external device, and restart communication.
$7F24_H$	Sum check error	The calculated sum check does not match the received sum check.	Review the sum check of the external device.
$7F67_H$	Overrun error	The next data was received before the CPU module completed receive processing.	Reduce the communication speed and restart communication. Check the CUP module for a momentary power failure. (For the CPU module, use the special register SD53 to check.) When a momentary power failure occurs, remove its cause.
$7F68_H$	Framing error	• The stop bit setting does not match. • Communication line became unstable by powering on/off the target device. • Noise is generated on the communication line.	• Match the setting of the CPU module with that of the external device. • Take noise reduction measures.
$7F69_H$	Parity error	• The parity bit setting does not match. • Communication line became unstable by powering on/off the target device. • Noise is generated on the communication line.	• Match the setting of the CPU module with that of the external device. • Take noise reduction measures.

3.39 iQ Sensor Solution Function

The iQ Sensor Solution function includes the following functions.

Function	Description
Automatic detection of connected device	Detects devices supporting iQ Sensor Solution connected to the CPU module, and automatically displays them on "List of devices" and "Device map area" using a programming tool.
System configuration check	Compares the system configuration information displayed on a programming tool with the actual system configuration, and checks if they match.
Reflection of the communication setting	Reflects the communication settings (such as IP addresses) of devices supporting iQ Sensor Solution on "Device map area" to the devices connected over Ethernet in the system.
Sensor parameter read/write	Reads/writes parameters from/to devices supporting iQ Sensor Solution.
Monitoring	Monitors the current values (such as measurement values and input/output values), status (error existence), and error information of devices supporting iQ Sensor Solution graphically using a programming tool.
Data backup/restoration	Backs up setting data (such as parameters) in devices supporting iQ Sensor Solution to an SD memory card. The data backed up can be restored as necessary.

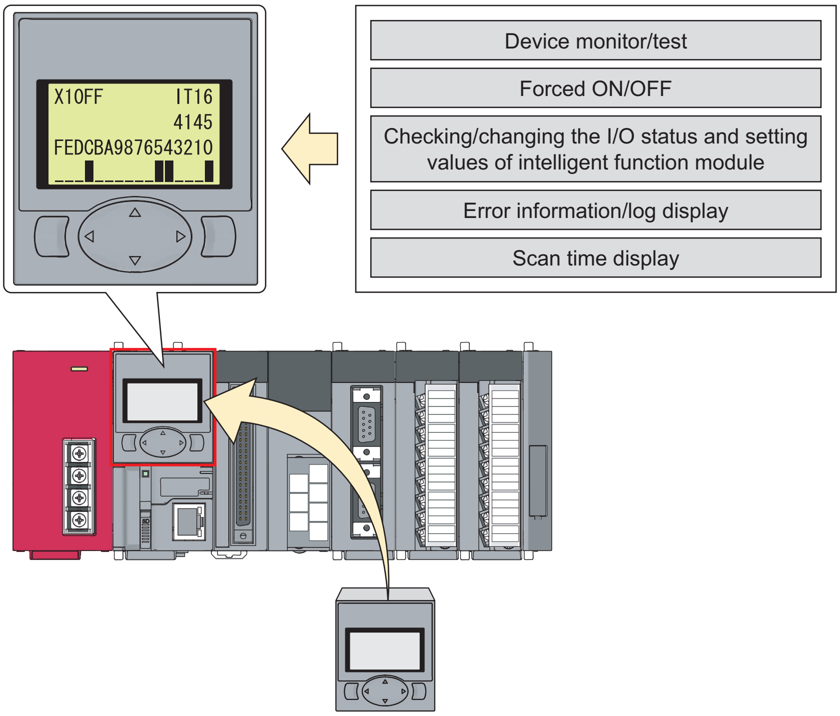
Remark

- Before using this function, check the versions of the CPU module and the programming tool used. (☞ Page 376, Appendix 2)
- The L02SCPU and L02SCPU-P do not support this function.
- For details on iQ Sensor Solution functions, refer to the following.
  iQ Sensor Solution Reference Manual

Memo

CHAPTER 4 DISPLAY UNIT FUNCTIONS

A display unit is an LCD attachable to the CPU module. Using a display unit allows checking system status and changing system setting values without software packages.
If an error occurs, the error cause can be determined by displaying the error information.



Remark

The L02SCPU and L02SCPU-P do not support the display unit functions.

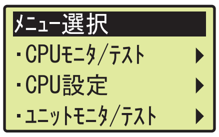
(1) Description of the buttons

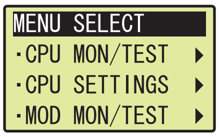
Unless otherwise specified, buttons on a display unit operate as follows.

Type	Name	Description
▲	Up arrow button	Moves the cursor up, scrolls the screen up, and increases values. In the following screens, pressing a device longer will increase a bit device value in increments of 10 bits (decimal notation) or 16 bits (hexadecimal notation), and will increase a word device value in increments of 10 words (decimal notation) or 16 words (hexadecimal notation). • Device monitor/test • Buffer memory monitor/test
▼	Down arrow button	Moves the cursor down, scrolls the screen down, and decreases values. In the following screens, pressing a device longer will increase a bit device value in increments of 10 bits (decimal notation) or 16 bits (hexadecimal notation), and will increase a word device value in increments of 10 words (decimal notation) or 16 words (hexadecimal notation). • Device monitor/test • Buffer memory monitor/test
▶	Right arrow button	Moves the cursor forward and switches screens.
◀	Left arrow button	Moves the cursor back and switches screens.
OK	OK button	Switches screens, executes functions, and accepts selection.
ESC	ESC button	Returns the display to the previously displayed screen or the initial screen for each function.

(2) Menu structure

Functions can be selected in the "function selection" screen.





In screen figures, the Japanese version is shown on the left and the English version on the right. Switch a displayed language by language setting. (☞ Page 283, Section 4.8.1)

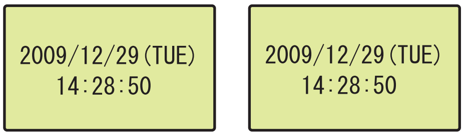
The menu structure is as shown below.

- CPU MON/TEST
  - DEV MON/TEST
  - DEVICE CLEAR
    - DEVICE MEMORY
    - FILE REGISTER
  - FORCED ON/OFF
    - SET ON/OFF
    - CLR FORCE
    - CLR ALL FORCES
    - LIST FORCES
  - SCAN TIME MON
  - BLT-IN I/O MON
    - BLT-IN POSITION
    - BLT-IN COUNTER
    - BLT-IN ERR CLR
  - ERROR MONITOR
    - MONITOR
    - LOG
      - USER MSG DSP
    - CLEAR
- CPU SETTINGS
  - CLOCK SETTING
- MOD MON/TEST
  - SPECIFY I/O No.
  - MODULE LIST
- MOD SETTINGS
  - SPECIFY I/O No.
    - INIT CHANGE*1
  - MODULE LIST
    - INIT CHANGE*1
- USER MESSAGE
  - USER MSG DSP
- MEM CARD OP
  - FILE LIST/DEL
  - FREE SPACE
  - BATCH SAVING
  - BATCH LOADING
- OPTIONS
  - LANGUAGE
  - CONTRAST
  - BACKLIGHT
  - POPUP
    - POP BLK DISABLE

*1 Menus lower than the "initial setting change" screen depend on the selected module.

(3) Standby screen

The standby screen is the initial screen of the display unit. The screen displays year, month, day, day of the week, hour, minute, and second. The internal time of the CPU module is automatically read to the display unit. Pressing any button moves you to the "function selection" screen.[*1]



*1 The ESC button is invalid.

(4) Backlight

(a) Normal status

The backlight lights up green. (When any button is not pressed within set lighting period, the backlight will turn off.) The backlight will turn on again in the following cases.

- Any button is pressed.
- A user message is popped up.
- An error is detected from the display unit by the self-diagnostic function.[*1]

*1 If this occurs, the backlight will not turn off even after elapse of set period.

(b) Error status

If an error occurs in the CPU module, the backlight lights up red. (When any button is not pressed within set lighting period, the backlight will turn off.) Check error information in the "Error list" screen and clear the error. (☞ Page 259, Section 4.2.6)

Note that the backlight status does not change in the following cases.

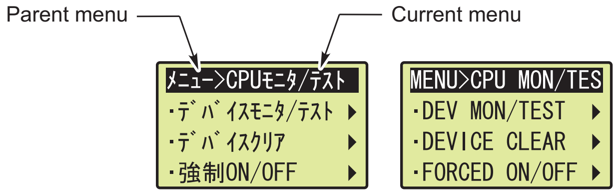
- "0" has been set to each cause number setting area of LED display priority ranking (SD207 to SD209).
- The ERR. LED, USER LED, or BAT. LED of the CPU module does not turn on.

(5) Self-diagnostic function

When a display unit detects an error by self-diagnostics, a continuation error ("DISPLAY ERROR") occurs in the CPU module, and an error message "DISPLAY UNIT ERROR" is displayed on the display unit. If the same error occurs even after the CPU module is reset, the possible cause is a hardware failure of the CPU module or the display unit. Check a 4-digit error code displayed on the display unit, and please consult your local Mitsubishi representative.

(6) Level navigation

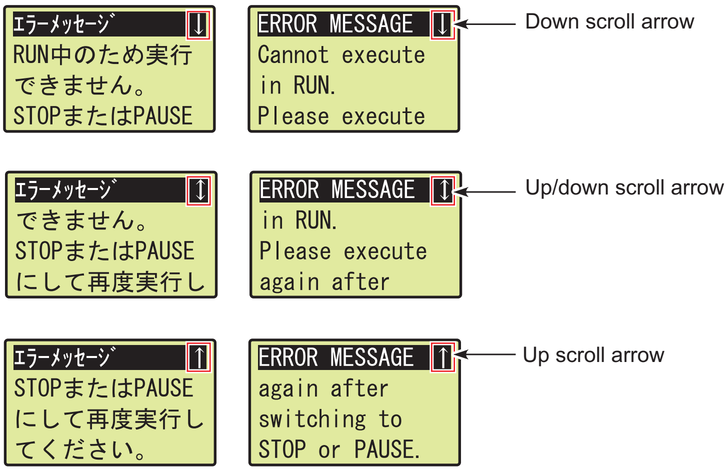
To clear where you are, level navigation is displayed on the top of the screen. The level navigation is inverted and the selected menu flashes.



Note that level navigation is not displayed in the detail setting screen of each function.

(7) Scrolling

When the display screen consists of several pages, the title line is displayed. An arrow indicating a scrollable direction is displayed on the top right of the screen.



However, when the contents of a screen with title line (such as a screen that displays an error message) can be displayed at a time, the scroll symbol (arrow) will not be displayed.

4

4.1 Function List

The following table lists the functions of a display unit.

Category	Function name		Displayed function name	Description	Reference
CPU monitor/test	Device monitor/test		DEV MON/TEST	Monitors and tests device memory values.	Page 241, Section 4.2.1
	Device clear	Device memory clear	DEVICE MEMORY	Clears all device memory values.	Page 249, Section 4.2.2
		File register clear	FILE REGISTER	Clears values of file register including extended data register and extended link register.	
	Forced on/off	Forced on/off registration	SET ON/OFF	Forcibly turns on or off devices X and Y.	Page 251, Section 4.2.3
		Forced on/off clear	CLR FORCE	Clears a forced on/off-registered device.	
		Forced on/off batch clear	CLR ALL FORCES	Clears all forced on/off-registered devices.	
		Forced on/off list	LIST FORCES	Lists forced on/off-registered devices.	
	Scan time monitor		SCAN TIME MON	Displays the current value, maximum value, and minimum value of scan time.	Page 254, Section 4.2.4
	Built-in I/O function monitor	Built-in positioning function monitor	BLT-IN POSITION	Displays the setting values of the built-in positioning function.	Page 255, Section 4.2.5
		Built-in high-speed counter function monitor	BLT-IN COUNTER	Displays the setting values of the built-in high-speed counter function.	
		Built-in I/O function error clear	BLT-IN ERR CLR	Clears errors of the built-in I/O function.	
	Error display	Current error monitor	MONITOR	Displays the current error code, message, common information, and individual information.	Page 259, Section 4.2.6
		Error log display	LOG	Displays information of error logs.	
		Error clear	CLEAR	Clears continuation errors.	
CPU settings	Clock setting		CLOCK SETTING	Sets the internal time of the CPU module.	Page 263, Section 4.3
Module monitor/test	Buffer memory monitor/test		BUF MEM MON/TES	Monitors and tests the buffer memory value of the selected module.	Page 265, Section 4.4
Module settings	Initial setting change		INIT CHANGE	Sets the initial value of the selected module.	Page 270, Section 4.5
User message	User message		USER MESSAGE	Checks a user message executed by the UMSG instruction.	Page 273, Section 4.6
Memory card operation	File list/delete		FILE LIST/DEL	Displays folders and files stored in an SD memory card in a list, and deletes a specified folder or file.	Page 274, Section 4.7.1
	Free space check		FREE SPACE	Displays the free space on an SD memory card.	Page 279, Section 4.7.2
	Batch save		BATCH SAVING	Saves data in the CPU module into an SD memory card.	Page 280, Section 4.7.3
	Batch load		BATCH LOADING	Loads data in an SD memory card to the CPU module.	Page 281, Section 4.7.4
OPTIONS	Language setting		LANGUAGE	Selects a displayed language.	Page 283, Section 4.8.1
	Contrast adjustment		CONTRAST	Sets tone of displayed characters.	Page 284, Section 4.8.2
	Lighting period setting		BACKLIGHT	Sets a period while the backlight is on.	Page 285, Section 4.8.3
	Pop-up setting		POP BLK DISABLE	Enables pop-up display.	Page 286, Section 4.8.4

4.2 CPU Monitor/Test

System status can be checked and device values can be changed with the display unit.
The CPU monitor/test includes the following functions.

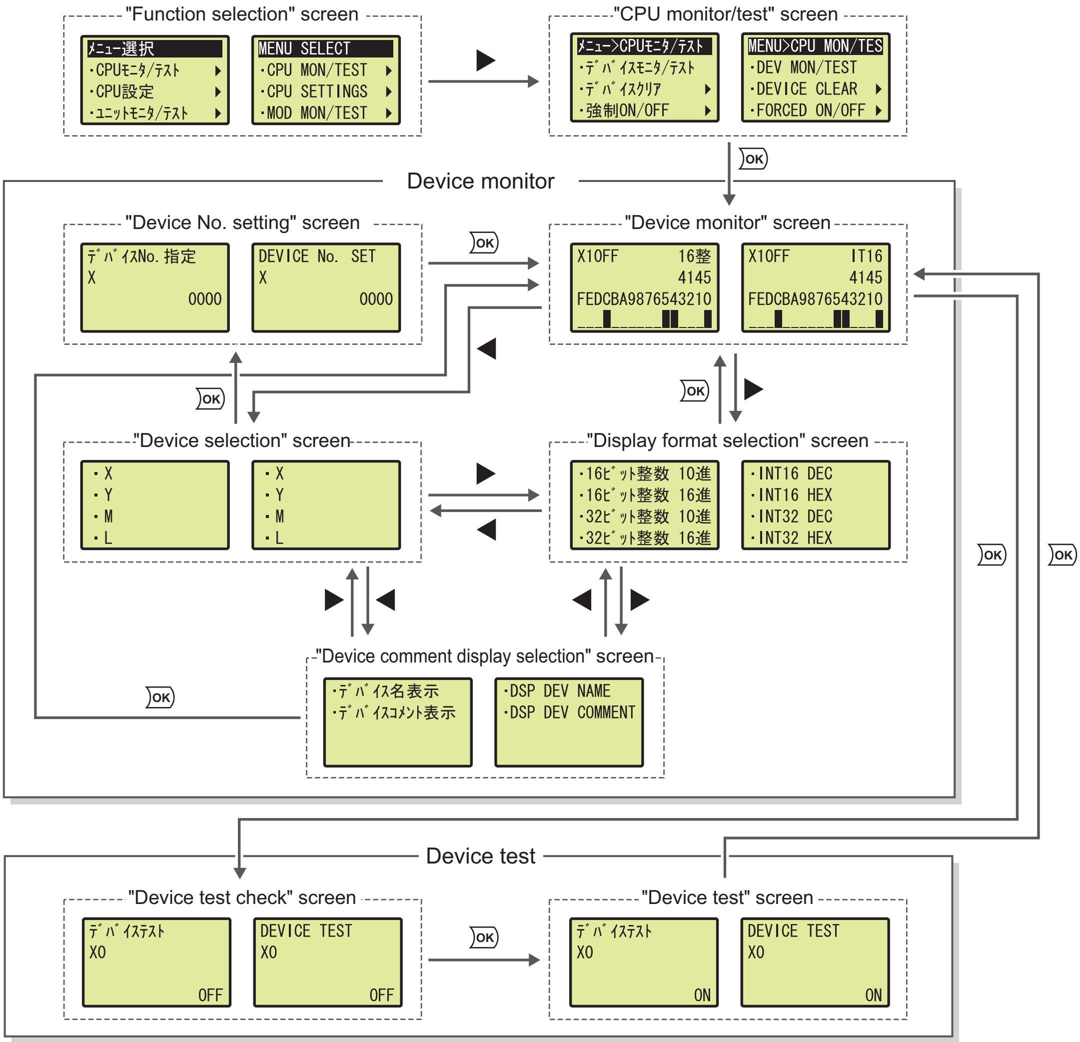
- Device monitor/device test
- Device clear
- Forced on/off
- Scan time monitor
- Built-in I/O function monitor
- Error display/clear

4.2.1 Device monitor/test

Device memory values can be monitored. Also, on/off status of bit devices and word device values can be checked and changed.

(1) Screen transition

The following shows screen transition of device monitor/test.



(2) Devices that can be monitored/tested

The following shows devices that can be monitored and tested by device monitor/test. Note that local devices cannot be monitored and tested.

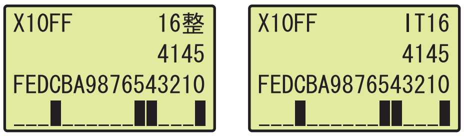
Type	Device*1
Bit device	X, Y, M, L, B, F, SB, V, SM, T (contact, coil*2), ST (contact, coil*2), C (contact, coil*2)
Word device	D, D (extended data register), W, W (extended link register), SW, SD, Z, R, ZR, T (current value), ST (current value), C (current value)

*1 Digit-specified devices, devices specified by indirect specification, and bit-specified devices cannot be set.
*2 Device test cannot be executed.

(3) Specifying a device to be monitored or tested

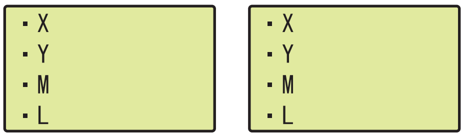
Perform the following operation.

"Device monitor" screen



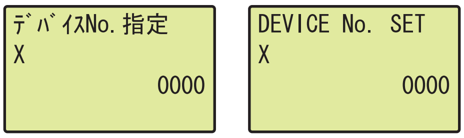
↓

"Device selection" screen



↓

"Device No. setting" screen



1. **Press the ◀ button.**

2. **Select a target device using the ▲ and ▼ buttons and press the OK button.**

3. **Move the cursor using the ◀ and ▶ buttons, increase or decrease a device number*1 using the ▲ and ▼ buttons, and press the OK button.**

*1 The device number is displayed in decimal or hexadecimal, according to the notation of each device. Regardless of the notation, the device number is displayed with the number of digits that can be set.

(4) Display format

The following table shows display format available for device monitor/test.

Format	Numeric value	Bit display	Remarks
16-bit integer	Decimal	Available	A sign is appended for negative values only.
	Hexadecimal		When a value is less than four digits, "0" is assigned for the remaining digits.
32-bit integer	Decimal	Not available	A sign is appended for minus values only.
	Hexadecimal		When a value is less than eight digits, "0" is assigned for the remaining digits.
Real number (single precision)*1	—		A value is displayed in floating point exponent (2 lines). A sign is appended for negative values only.
Real number (double precision)*1			A value is displayed in floating point exponent (2 lines). A sign is appended for negative values only.

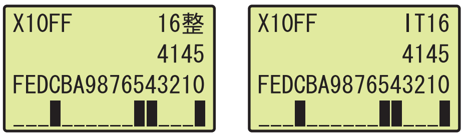
*1 Device test cannot be executed while a value is displayed in real number. (except for bit device, timer (T), retentive timer (ST), and counter (C))

Also, a device number can be displayed with device comment. Note that, however, the device comment can be displayed up to 5 two-byte characters (11 one-byte characters).

(a) Specifying display format

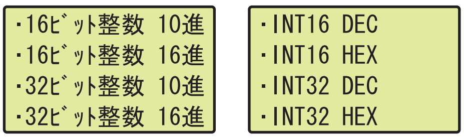
Perform the following operation.

"Device monitor" screen



↓

"Display format selection" screen



1. **Press the ▶ button.**

2. **Select display format using the ▲ and ▼ buttons and press the OK button.**

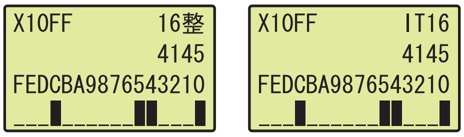
4

4.2 CPU Monitor/Test
4.2.1 Device monitor/test

(b) Specifying a device comment

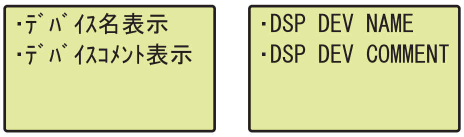
To display a device with device comment, perform the following operation.*1*2*3

"Device monitor" screen



↓

"Device comment display selection" screen



1. **Press the ▶ button twice.**

2. **Select "DSP DEV COMMENT" using the ▲ and ▼ buttons and press the OK button.**

*1 A device comment can be displayed only in the following screens.
- "Device monitor" screen
- "Device test check" screen
- "Device test" screen

*2 To display a device with device comment, preset "Comment File Used in a Command" in the PLC File tab of the PLC Parameter dialog box.

*3 The device comment can be displayed up to 5 two-byte characters (11 one-byte characters).

Point

- If a device comment file is changed while the device comment is displayed, the device comment is updated when the screen transitions.
- When several device comment files are used, the device comment in the device comment file used when the screen changes to the "device monitor" screen is displayed.

(5) Device monitor

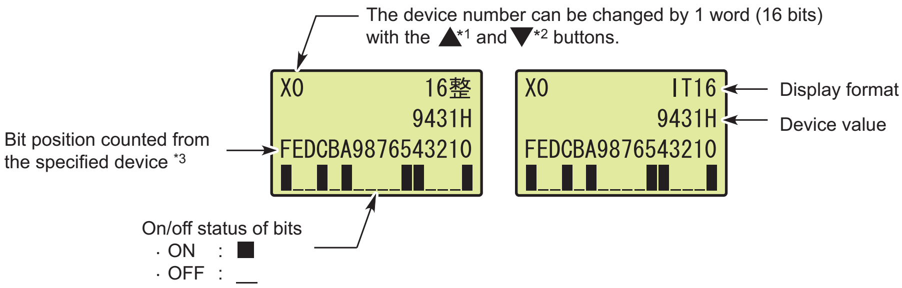
A device specified in (3) in this section can be monitored from the "device monitor" screen.

Point

Depending on update timing of a screen, the update of a value displayed in the "device monitor" screen may delay accordingly.

(a) Monitoring a bit device (display format: 16-bit integer)

The specified device, and display format, a value, and bit status of the specified device can be monitored.



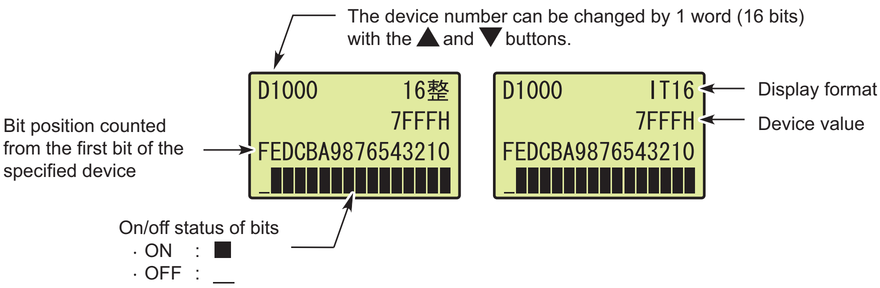
*1 When a bit device is specified within one word from the start number of the bit device, the specified device number will be changed to the start number.
Example: X1 to XF → X0

*2 When a bit device is specified within one word from the last number of the bit device, the specified device number will be changed to the last number.
Example: X1FF1 to X1FFE → X1FFF

*3 When X1 is specified as a monitor target, "F" indicates the on/off status of X10.

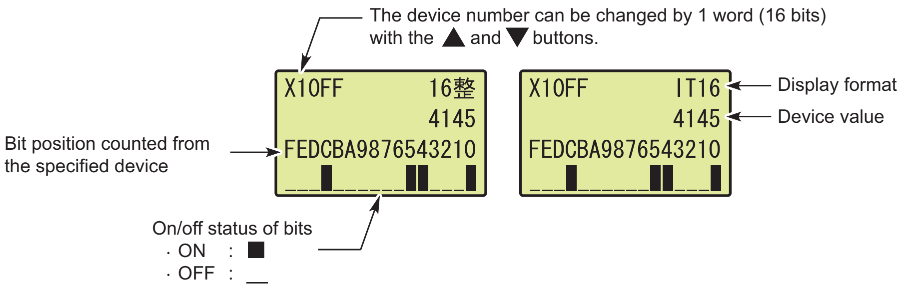
(b) Monitoring a word device (display format: 16-bit integer)

The specified device, and display format, a value, and bit status of the specified device can be monitored.



(c) Monitoring a bit device (display format: 32-bit integer)*1

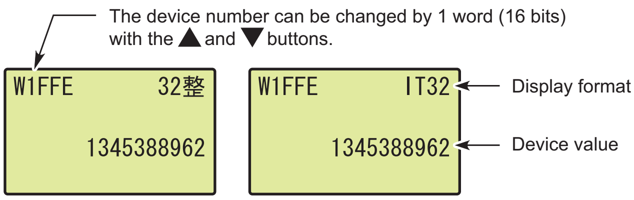
The specified device, and display format and a value of the specified device can be monitored.



*1 Even if the display format is set to 32-bit integer (decimal) or 32-bit integer (hexadecimal), a bit device value is displayed in 16-bit integer (decimal) or 16-bit integer (hexadecimal).

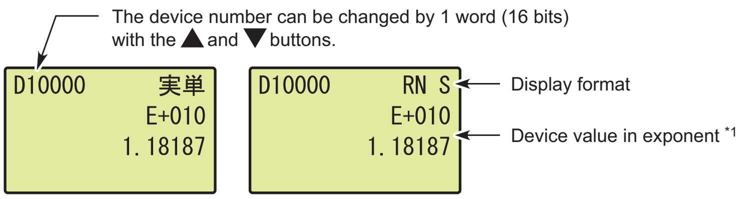
(d) Monitoring a word device (display format: 32-bit integer)

The specified device, and display format and a value of the specified device can be monitored.



(e) Monitoring a real number (single precision)*2

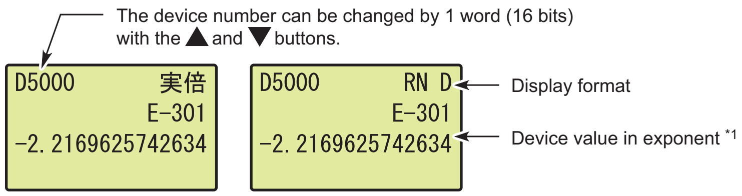
The specified device, and display format and a value of the specified device can be monitored.



*1 If the device value is "-0", unnormalized number, nonnumeric character, or ±∞, "*********" is displayed.
*2 When a bit device is monitored, the device value is displayed in 16-bit integer (decimal).

(f) Monitoring a real number (double precision)*2

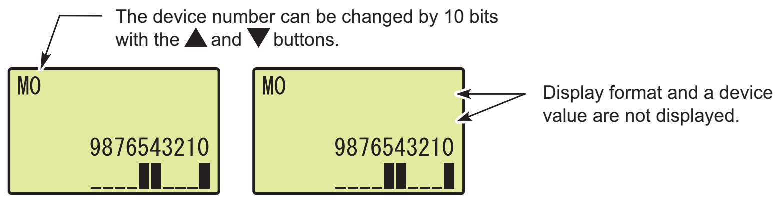
The specified device, and display format and a value of the specified device can be monitored.



*1 If the device value is "-0", unnormalized number, nonnumeric character, or ±∞, "*********" is displayed.
*2 When a bit device is monitored, the device value is displayed in 16-bit integer (decimal).

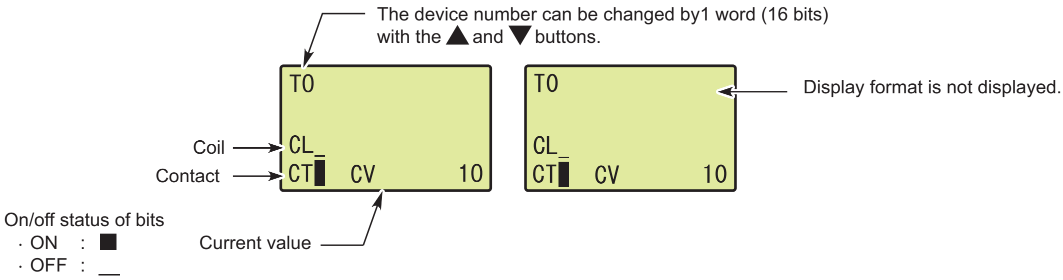
(g) Monitoring a bit device in decimal

The specified device and bit status of the specified device can be monitored. Regardless of display format, a screen with the following style appears.



(h) Monitoring timer (T), retentive timer (ST), and counter (C)

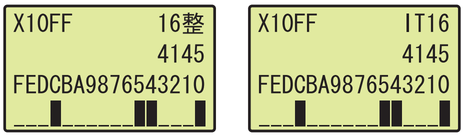
The specified device and a value and bit status of the specified device can be monitored. The current value is displayed in decimal or hexadecimal, the format specified in the "display format selection" screen. When "REAL" has been specified, the current value is displayed in decimal.



(6) Device test

On/off status of a bit device and a word device value can be changed.

"Device monitor" screen

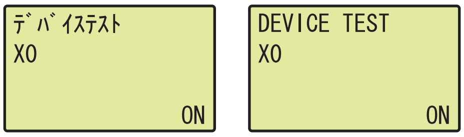


1. **Press the OK button.**

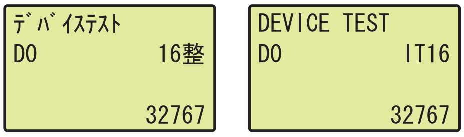
↓

"Device test check" screen

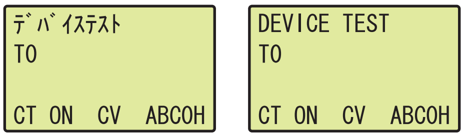
(Bit device)



(Word device)



(Timer (T), retentive timer (ST), and counter (C))

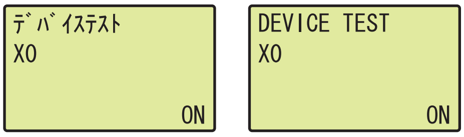


2. **Select a device using the ▲ and ▼ buttons. (a device value is changed by 1 bit for bit device and 1 word for word device) and press the OK button.**

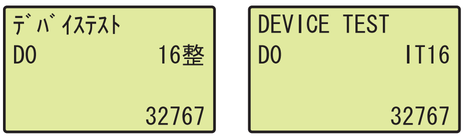
↓

"Device test" screen

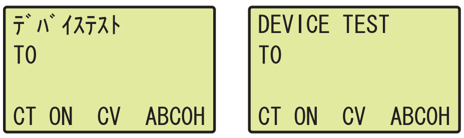
(Bit device)



(Word device)



(Timer (T), retentive timer (ST), and counter (C))



3. **Change device status. Change on/off status of a bit device using the ▲ and ▼ buttons. For a word device, the cursor can be moved forward or back using the ◀ and ▶ buttons. Increase or decrease the value at the cursor position using the ▲ and ▼ buttons.**

4. **Pressing the OK button will update the value to the one set in device test.**

4.2.2 Device clear

Device memory values and file register values can be cleared with the display unit.
The device clear includes the following functions.

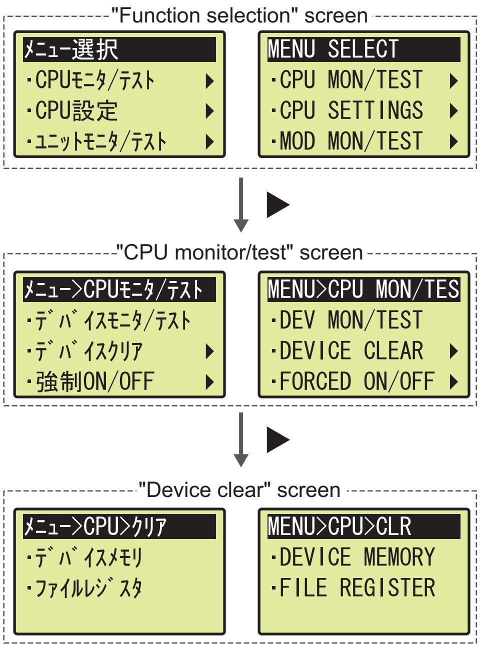
- Device memory clear
- File register clear

Point

Device clear can be executed while the CPU module is in the STOP or PAUSE status.

(1) Screen transition

The following shows screen transition of device clear.



(2) Devices whose value can be cleared

(a) Device memory clear

The following table shows devices whose value can be cleared by device memory clear. Whether to clear the values of devices within latch range can be selected.

Type	Device
Bit device	X (DX), Y (DY), M, L, B, F, SB, V, S, T (contact, coil), ST (contact, coil), C (contact, coil), FX, FY, FD
Word device	D, D (extended data register), W, W (extended link register), SW, Z, R*1, ZR*1, T (current value), ST (current value), C (current value)

*1 A value stored within latch range cannot be cleared.

(b) File register clear

Only fire register (R, ZR) values can be cleared by file register clear. (including extended data register (D) and extended link register (W))

(3) Operating procedure

The following is the procedure for clearing data.

"Device clear" screen

ﾒﾆｭｰ>CPU>ｸﾘｱ ・ﾃﾞﾊﾞｲｽﾒﾓﾘ ・ﾌｧｲﾙﾚｼﾞｽﾀ	MENU>CPU>CLR ・DEVICE MEMORY ・FILE REGISTER

1. **Select "DEVICE MEMORY" or "FILE REGISTER" using the ▲ and ▼ buttons and press the OK button.**

↓

"Latched device memory clear selection" screen

・ﾗｯﾁ含まない ・ﾗｯﾁ含む	・ALL BUT LATCHED ・INCLUDE LATCHED

2. **Select whether to clear the values of devices within latch range using the ▲ and ▼ buttons and press the OK button. (can be set for device memory clear only)**

↓

"Device memory clear" screen

ﾃﾞﾊﾞｲｽﾒﾓﾘを ｸﾘｱします ・はい ・いいえ	Clear device memory? ・YES ・NO

"File register clear" screen

ﾌｧｲﾙﾚｼﾞｽﾀを ｸﾘｱします ・はい ・いいえ	Clear file register? ・YES ・NO

3. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

↓

"Device memory clear confirmation" screen

ﾃﾞﾊﾞｲｽﾒﾓﾘｸﾘｱを 実行します ・はい ・いいえ	Execute clear device memory? ・YES ・NO

"File register clear confirmation" screen

ﾌｧｲﾙﾚｼﾞｽﾀｸﾘｱを 実行します ・はい ・いいえ	Execute clear file register? ・YES ・NO

4. **In the confirmation screen, select "YES" again using the ▲ and ▼ buttons and press the OK button.**

↓

"Device memory clear completion" screen

ﾃﾞﾊﾞｲｽﾒﾓﾘのｸﾘｱが 完了しました	Device memory cleared.

"File register clear completion" screen

ﾌｧｲﾙﾚｼﾞｽﾀのｸﾘｱが 完了しました	File register cleared.

5. **When the processing is completed successfully, either of the screens shown left appears.**
To return the display to the "Device clear" screen, press the OK button.

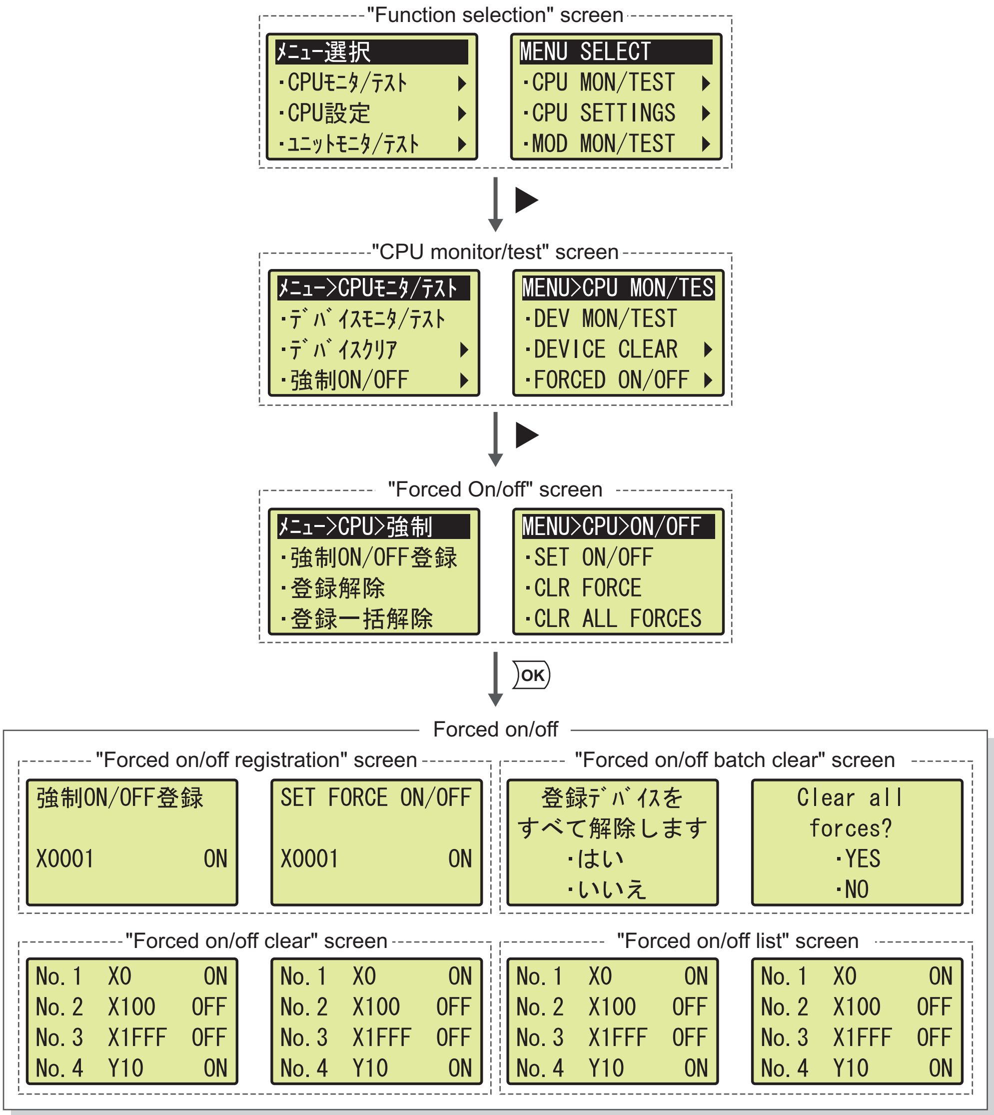
4.2.3 Forced on/off

X and Y devices can be forcibly turned on or off with the display unit.
The forced on/off includes the following functions.

- Forced on/off registration
- Forced on/off clear
- Forced on/off batch clear
- Forced on/off list

(1) Screen transition

The following shows screen transition of forced on/off.

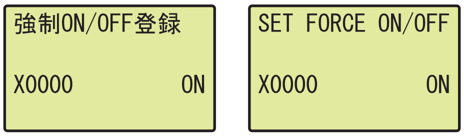


(2) Forced on/off registration

Specify a device (X or Y) and whether to turn on or off the specified device. The specified device and device status are registered in a list. Devices that have already been registered can be reset. Up to 32 devices can be registered.

Perform the following operation from the "forced on/off registration" screen.

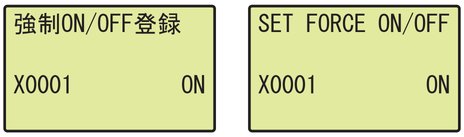
"Forced on/off registration" screen



1. **Switch a device using the ▲ and ▼ buttons.**

↓

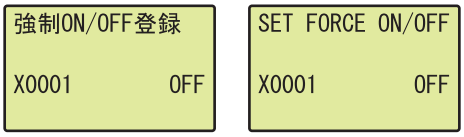
"Forced on/off registration" screen



2. **Move the cursor using the ◀ and ▶ buttons and increase or decrease a device number using the ▲ and ▼ buttons.**

↓

"Forced on/off registration" screen



3. **Move the cursor using the ◀ and ▶ buttons and switch on/off setting using the ▲ and ▼ buttons.**

4. **Pressing the OK button will register the setting and will return the display to the "forced on/off" screen.**

Point

When at least one device is registered, the MODE LED on the CPU module will be flashing in green.

(3) Forced on/off clear

Registered devices are cleared one by one. Perform the following operation from the "forced on/off clear" screen.

"Forced on/off clear" screen

```
No. 1  X0      ON     No. 1  X0      ON
No. 2  X100   OFF     No. 2  X100   OFF
No. 3  X1FFF  OFF     No. 3  X1FFF  OFF
No. 4  Y10     ON     No. 4  Y10     ON
```

1. Select a device to be cleared using the ▲ and ▼ buttons.

2. Pressing the OK button will clear the selected device.

(4) Forced on/off batch clear

Registered devices are batch-cleared. Perform the following operation from the "forced on/off batch clear" screen.

"Forced on/off batch clear" screen

```
 登録ﾃﾞﾊﾞｲｽを        Clear all
すべて解除します       forces?
   ･はい               ･YES
   ･いいえ             ･NO
```

↓

"Forced on/off batch clear completion" screen

```
登録の一括解除が      Forces cleared.
 完了しました
```

1. Select "YES" using the ▲ and ▼ buttons and press the OK button.

2. When the processing is completed successfully, either of the screens shown left appears.
To return the display to the "Device clear" screen, press the OK button.

(5) Forced on/off list

Registered devices are listed in the "forced on/off list" screen.

The screen is scrolled by one line by pressing the ▲ or ▼ button once.

```
No. 1  X0      ON     No. 1  X0      ON
No. 2  X100   OFF     No. 2  X100   OFF
No. 3  X1FFF  OFF     No. 3  X1FFF  OFF
No. 4  Y10     ON     No. 4  Y10     ON
```

4.2.4 Scan time monitor

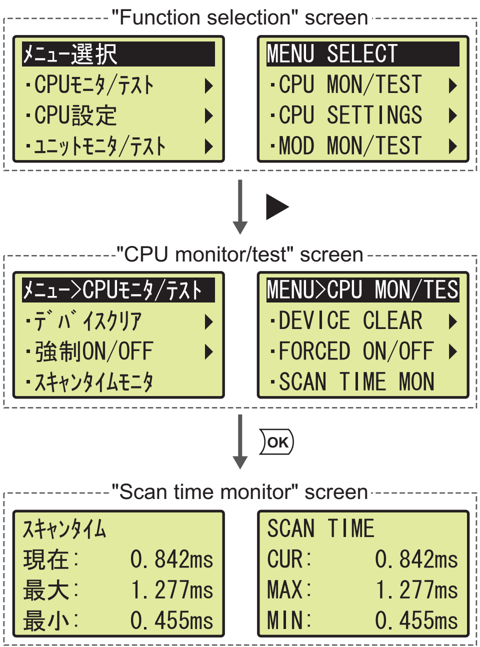
The current value, maximum value, and minimum value of scan time can be checked with the display unit. Values stored in the special register of the CPU module are displayed.

Displayed item	Special register	Description
Current value	SD520, SD521	Current scan time
Maximum value	SD526, SD527	Maximum scan time*1
Minimum value	SD524, SD525	Minimum scan time*1

*1 These values of an initial execution type program are not displayed.

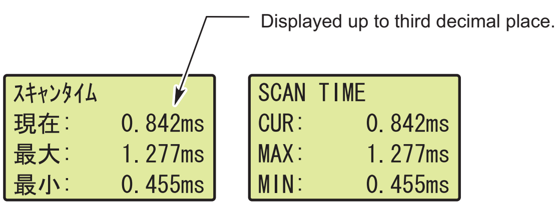
(1) Screen transition

The following shows screen transition of scan time monitor.



(2) Monitor data

The current value, maximum value, and minimum value of scan time are displayed.



4.2.5 Built-in I/O function monitor

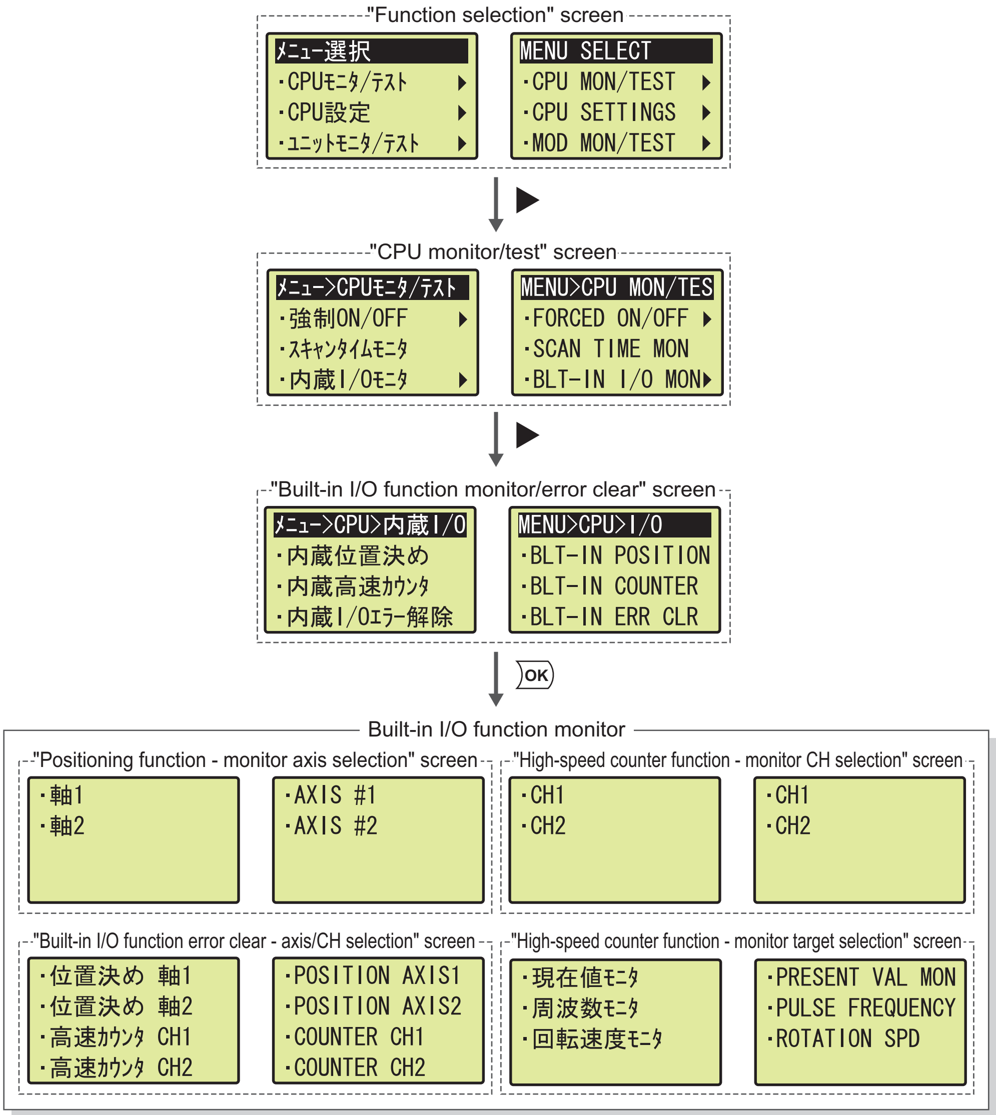
I/O status and setting values of the built-in I/O function can be checked with the display unit. Also, errors detected during execution of the built-in I/O function can be cleared.

The built-in I/O monitor includes the following functions.

- Built-in positioning function monitor
- Built-in high-speed counter function monitor
- Built-in I/O function error clear

(1) Screen transition

The following shows screen transition of built-in I/O function monitor.



(2) Built-in positioning function monitor

The setting values of the built-in positioning function can be monitored.

(a) Displayed items

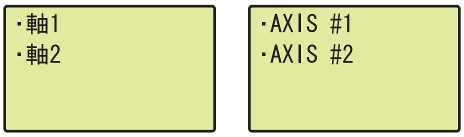
The following items are displayed.

Displayed item	Description	Setting range
Position (p)	Positioning range	-2147483648 to 2147483647 pls
Velocity (v)	Speed command (speed limit value)	0 to 200000 pps
Data No.	Data No. of positioning being executed	"ABSENT" or 1 to 10
Error code	Error code of the built-in positioning function	"ABSENT" or an error code is displayed.

(b) Operating procedure

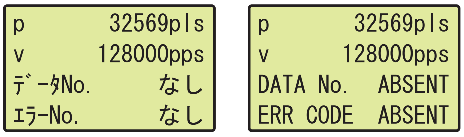
The following is the procedure for monitoring the values.

"Positioning function - monitor axis selection" screen



↓

"Positioning function monitor" screen



1. **Select "AXIS #1" or "AXIS #2" using the ▲ and ▼ buttons and press the OK button.**

2. **The displayed items are shown.**

(3) Built-in high-speed counter function monitor

The setting values of the built-in high-speed counter function can be monitored.

(a) Displayed items

The following items are displayed.

Displayed item	Description	Setting range
Current value (m)	Current counter value	-2147483648 to 2147483647
Frequency (f)	A frequency of the high-speed counter function	-100000 to 100000Hz *1
Measurement	Measurement status of the high-speed counter function	Operating/not operating
Rotation speed (r)	Rotation speed of the high-speed counter function	-12000000 to 12000000r/min*2*3
Error code	Error code of the high-speed counter function	"ABSENT" or an error code is displayed.

*1 This applies when "1-Phase Multiple of 1", "1-Phase Multiple of 1 (A Phase only)", or "CW/CCW" is set for "Pulse Input Mode".

*2 This applies when "200kpps" is set for "Counting Speed Setting".

*3 If pulses exceeding the number of pulses shown in the performance specifications are input, this range may be exceeded.

(b) Operating procedure

The following is the procedure for monitoring the values.

1. **Select "CH1" or "CH2" using the ▲ and ▼ buttons and press the OK button.**

"High-speed counter function - monitor CH selection" screen

```
・CH1
・CH2
```

```
・CH1
・CH2
```

↓

2. **Select "PRESENT VAL MON", "PULSE FREQUENCY", or "ROTATION SPD" using the ▲ and ▼ buttons and press the OK button.**

"High-speed counter function - monitor target selection" screen

```
・現在値ﾓﾆﾀ
・周波数ﾓﾆﾀ
・回転速度ﾓﾆﾀ
```

```
・PRESENT VAL MON
・PULSE FREQUENCY
・ROTATION SPD
```

↓

3. **The displayed items are shown.**

"High-speed counter function - current value monitor" screen

```
m       66358712
ｴﾗｰNo.       なし
```

```
m       66358712
ERR CODE  ABSENT
```

"High-speed counter function - frequency monitor" screen

```
f        10245Hz
測定       測定中
ｴﾗｰNo.       なし
```

```
f        10245Hz
MEASURE     EXEC
ERR CODE  ABSENT
```

"High-speed counter function - rotation speed monitor" screen

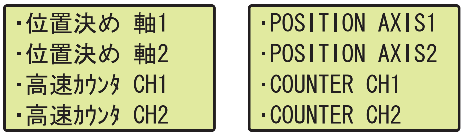
```
r      114256rpm
測定       測定中
ｴﾗｰNo.       なし
```

```
r      114256rpm
MEASURE     EXEC
ERR CODE  ABSENT
```

(4) Built-in I/O function error clear

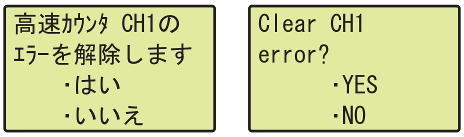
An error can be cleared by specifying an axis of the positioning function or CH (channel) of the high-speed counter function.

"Built-in I/O function error clear - axis/CH selection" screen



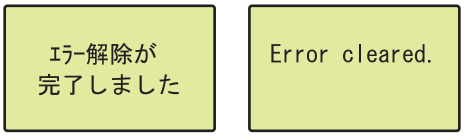
↓

"Built-in I/O function - error clear confirmation" screen



↓

"Built-in I/O function - error clear completion" screen



1. **Select the target item using the ▲ and ▼ buttons and press the OK button.**

2. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

3. **When the error is cleared successfully, either of the screens shown left appears.**

To return to the "Built-in monitor/error clear" screen, press the OK button.

4.2.6 Error display/clear

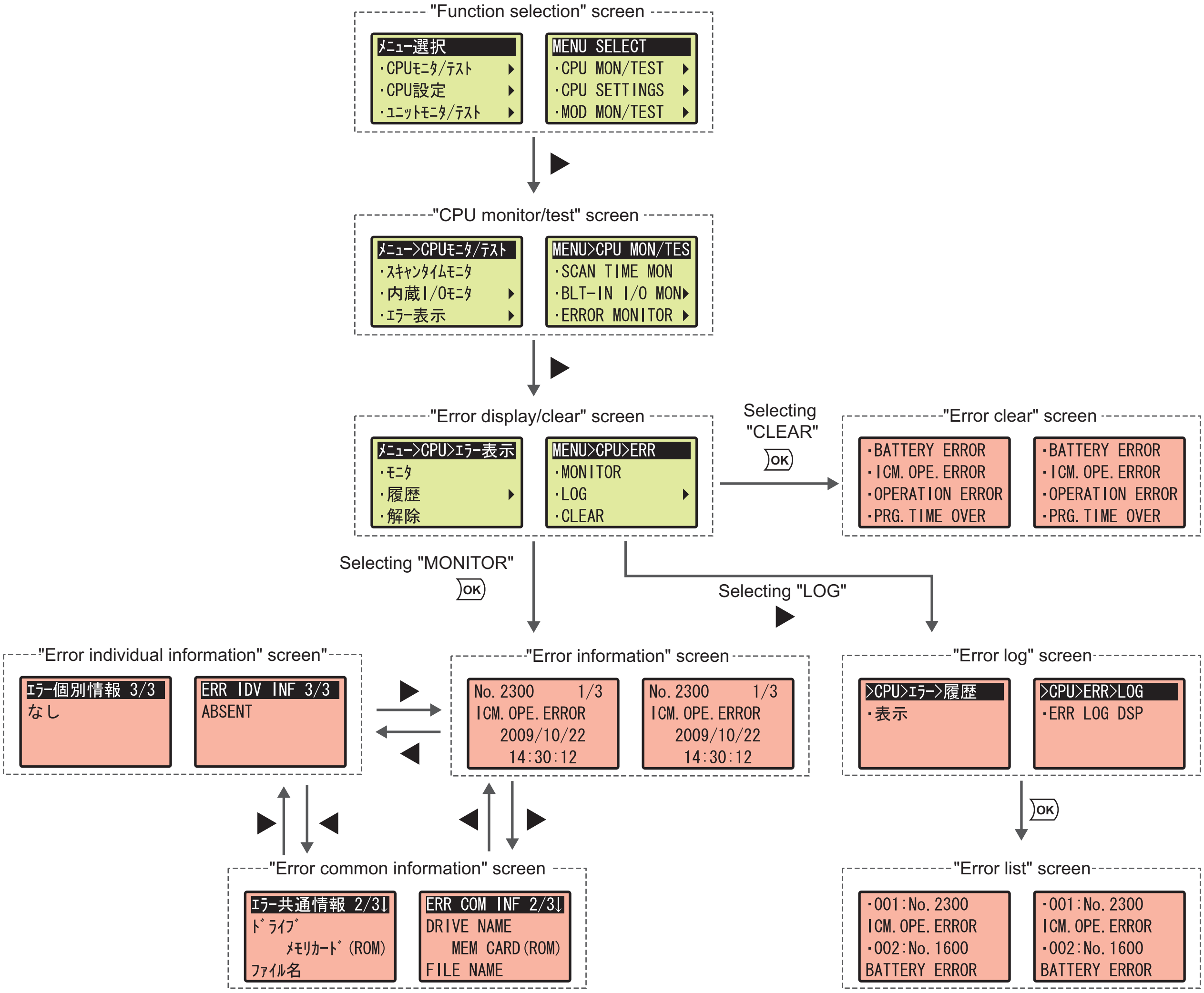
The current and past errors and their information can be checked with the display unit. Also, the current error can be cleared.

The error display/clear includes the following functions.

- Current error monitor
- Error log display
- Error clear

(1) Screen transition

The following shows screen transition of error display/clear.



(2) Current error monitor

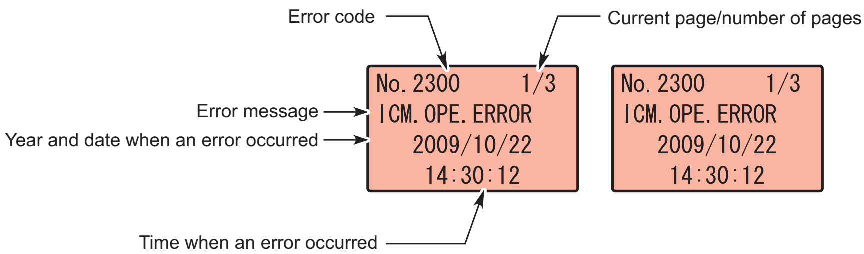
The information of latest and current CPU module error (including an annunciator) can be checked.

The current error monitor includes the following screens.

- "Error information" screen
- "Error common information" screen
- "Error individual information" screen

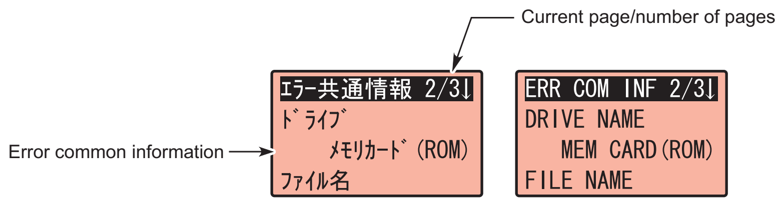
(a) Error information

If an error occurs, the following information is displayed.



(b) Error common information

If an error occurs, the following information is displayed.

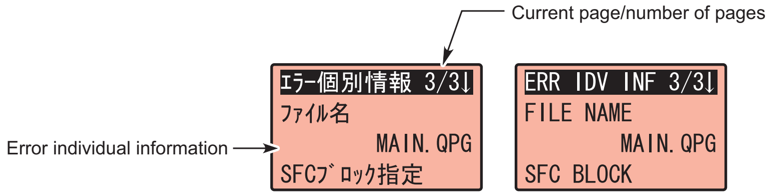


The following items are displayed in the "error common information" screen.

Error common information	Description	Error common information	Description
Module No.	Slot No./block No.	Program error location	File name
	I/O No.		SFC block specification
File name/drive name	Drive		SFC step specification
	File name		SFC transition specification
Time (setting value)	Time		Step No./transition condition
Not error	—		Sequence step No.

(c) Error individual information

If an error occurs, the following information is displayed.



The following items are displayed in the "error individual information" screen.

Error individual information	Description	Error individual information	Description
File name/drive name	Drive	Parameter No.	Parameter No.
	File name	Annunciator No.	F No.
Time (measured value)	Time	File diagnostic information	Drive No.
Program error location	File name		File name
	SFC block specification		Failure information 1
	SFC step specification		Failure information 2
	SFC transition specification		Failure information 3
	Step No./transition condition	Not error	—
	Sequence step No.		

(3) Error log display

Errors that have occurred in the CPU module are displayed in a list. The following table shows displayed items.

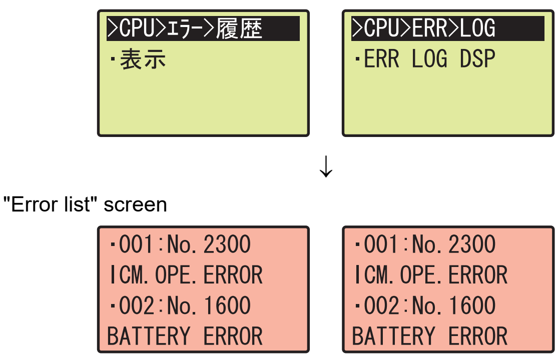
Displayed item	Description
Display number	Display order of error information (error code, error message)*1
Error code	Error code of the corresponding error
Error message	Error message of the corresponding error

*1 Time when errors have occurred is listed in reverse chronological order.

(a) Operating procedure

The following is the procedure for displaying the errors.

"Error log" screen



1. **Press the OK button.**

2. **To display error information, select the target error log using the ▲ and ▼ buttons and press the OK button.**

(4) Error clear

Continuation errors can be cleared with the display unit.

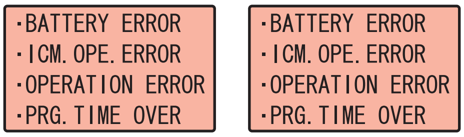
Point

Remove the error cause before clearing an error.
For how to remove error causes, refer to the following.
📖MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(a) Operating procedure

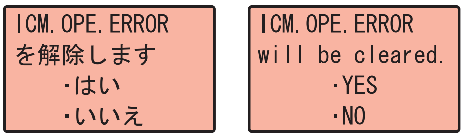
The following is the procedure for clearing the error.

"Error clear" screen



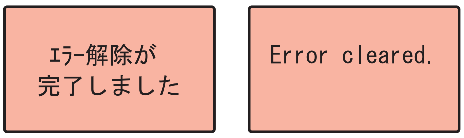
↓

"Error clear confirmation" screen



↓

"Error clear completion" screen



1. **Select an error to be cleared using the ▲ and ▼ buttons and press the OK button.**

2. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

3. **When the error is cleared successfully, either of the screens shown left appears.**

To return to the "Error clear" screen, press the OK button.

4.3 CPU Settings

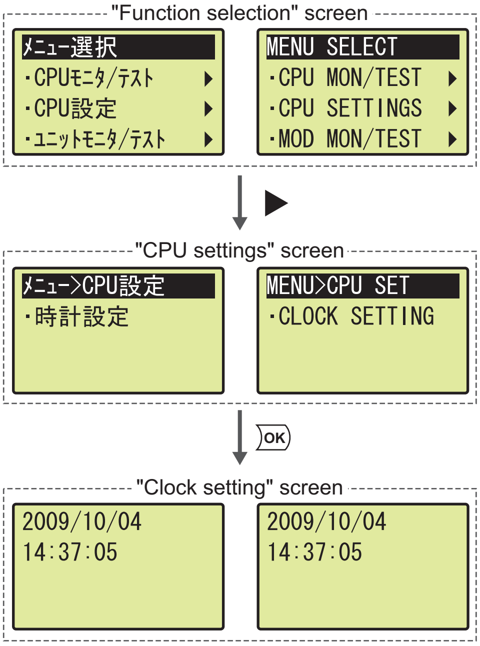
The internal time of the CPU module can be checked and changed with the display unit.

4.3.1 Clock setting

The internal time of the CPU module can be set.

(1) Screen transition

The following shows screen transition of clock setting.



(2) Setting item

The following table shows items that can be set by clock setting.

Item	Setting range	Item	Setting range
Year	1980 to 2079	Hour	00 to 23
Month	1 to 12	Minute	00 to 59
Day	1 to 31	Second	00 to 59

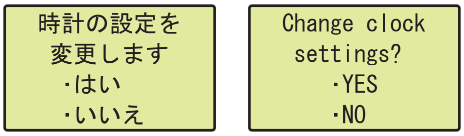
(3) Operating procedure

The following is the procedure for setting the clock data.

"Clock setting" screen

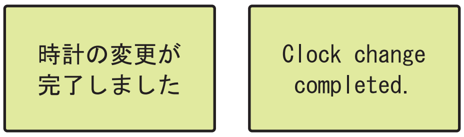
↓

"Clock setting confirmation" screen



↓

"Clock setting completion" screen



1. **Move the cursor using the ◀ and ▶ buttons, increase or decrease a time value using the ▲ and ▼ buttons, and press the OK button.**

2. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

3. **When the setting is completed, either of the screens shown left appears.**

To return to the "CPU settings" screen, press the OK button.

4.4 Module Monitor/Test

The buffer memory values of intelligent function modules (including the LCPU with the built-in CC-Link function) can be checked and changed with the display unit.[*1]

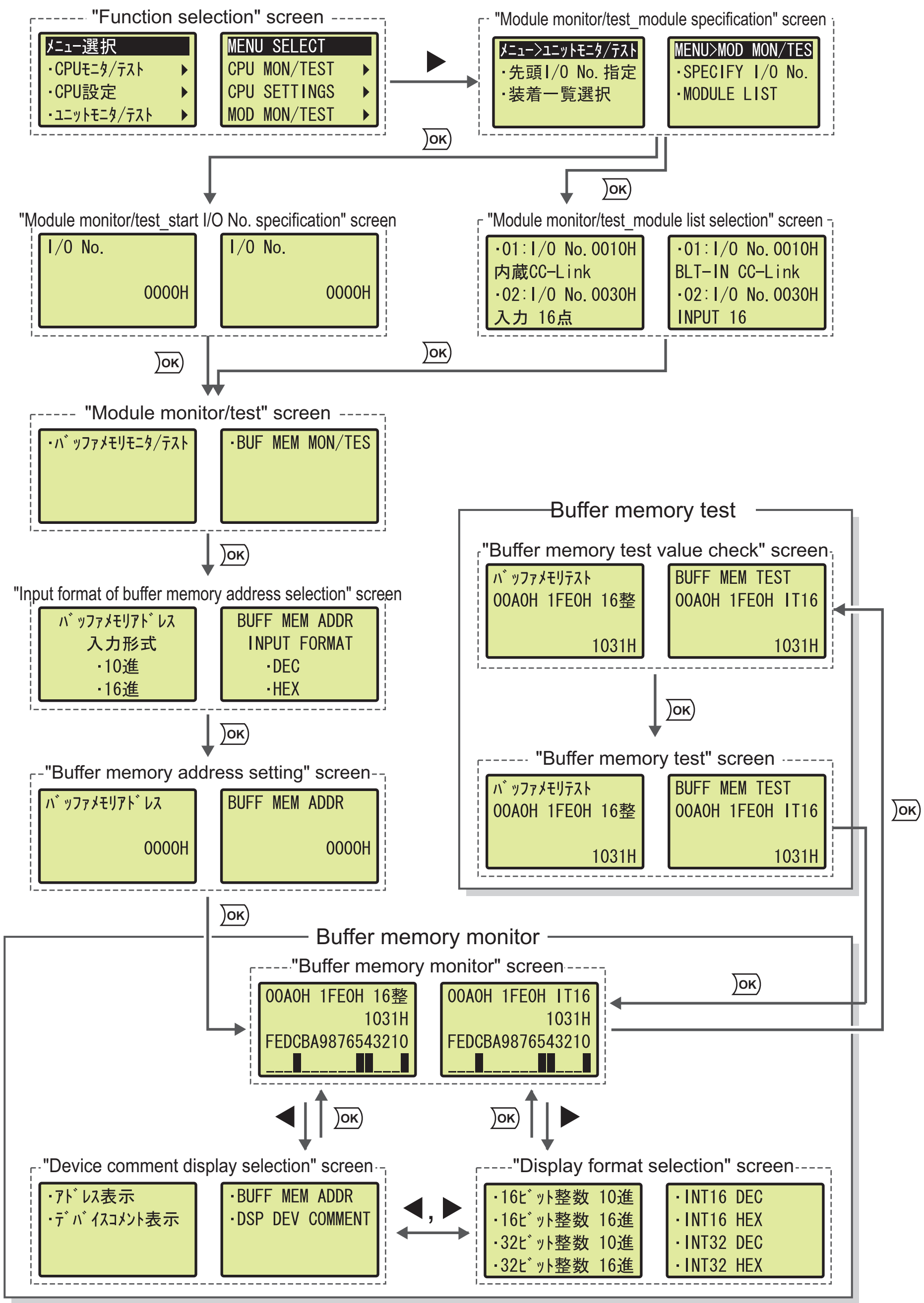
*1 The values of MELSEC-AnS/QnAS series modules mounted on the LA1S extension base unit cannot be checked.

4.4.1 Buffer memory monitor/test

Buffer memory values can be monitored and tested.

(1) Screen transition

The following shows screen transition of buffer memory monitor/test.



(2) Specifying a module

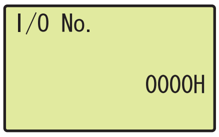
A module to be monitored or tested is specified by the following ways.

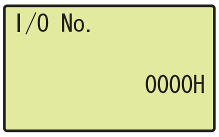
- Start I/O No. specification
- Module selection

(a) Start I/O No. specification

Perform the following operation.

"Module monitor/test_start I/O No. specification" screen



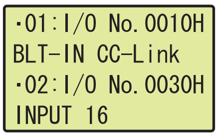


1. **Move the cursor using the ◀ and ▶ buttons, increase or decrease a start I/O No. using the ▲ and ▼ buttons, and press the OK button.**

(b) Module selection

Perform the following operation.

"Module monitor/test_module selection" screen



1. **Select a target item using the ▲ and ▼ buttons and press the OK button.**

(3) Specifying a buffer memory address

Specify the buffer memory address of the module selected in (2) in this section. Perform the following operation.

"Module monitor/test" screen

･ﾊﾞｯﾌｧﾒﾓﾘﾓﾆﾀ/ﾃｽﾄ	･BUF MEM MON/TES

↓

"Buffer memory address input format selection" screen

ﾊﾞｯﾌｧﾒﾓﾘｱﾄﾞﾚｽ 入力形式 ･10進 ･16進	BUFF MEM ADDR INPUT FORMAT ･DEC ･HEX

↓

"Buffer memory address specification" screen

ﾊﾞｯﾌｧﾒﾓﾘｱﾄﾞﾚｽ 00000	BUFF MEM ADDR 00000

1. **Press the OK button.**

2. **Select the input format of a buffer memory address using the ▲ and ▼ buttons and press the OK button.**

3. **Move the cursor using the ◀ and ▶ buttons, increase or decrease an address digit using the ▲ and ▼ buttons, and press the OK button.**

(4) Specifying display format

Perform the following operation.

"Buffer memory monitor" screen

00A0H 1FE0H 16整 1031H FEDCBA9876543210	00A0H 1FE0H IT16 1031H FEDCBA9876543210

↓

"Display format selection" screen

･16ﾋﾞｯﾄ整数 10進 ･16ﾋﾞｯﾄ整数 16進 ･32ﾋﾞｯﾄ整数 10進 ･32ﾋﾞｯﾄ整数 16進	･INT16 DEC ･INT16 HEX ･INT32 DEC ･INT32 HEX

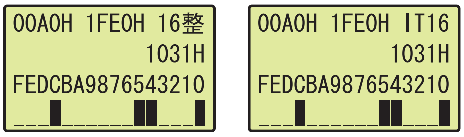
1. **Press the ▶ button.**

2. **Select display format using the ▲ and ▼ buttons and press the OK button.**

(5) Specifying a device comment

To display a device with device comment, perform the following operation.*1*2

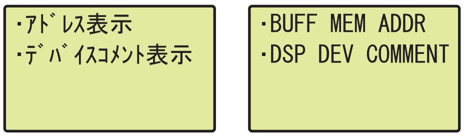
"Buffer memory monitor" screen



1. **Press the ◀ button.**

↓

"Device comment display selection" screen



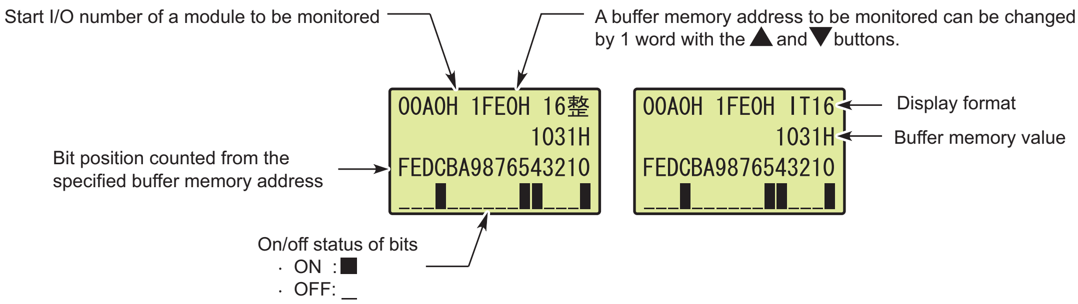
2. **Select "DSP DEV COMMENT" using the ▲ and ▼ buttons and press the OK button.**

*1 Device comments are displayed only in the following screens.
· "Buffer memory monitor" screen
· "Buffer memory test value check" screen
· "Buffer memory test" screen

*2 To display a device comment, preset "Comment File Used in a Command" in the PLC File tab of the PLC Parameter dialog box.

(6) Buffer memory monitor

The start I/O No., buffer memory address, display format, buffer memory value, and bit status of the buffer memory value of the specified module can be monitored.



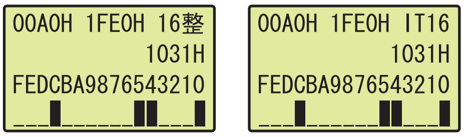
Point

The description of displayed items is the same as that of the device monitor. (☞ Page 245, Section 4.2.1 (5))

(7) Buffer memory test

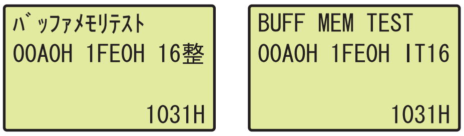
A buffer memory value can be changed.

"Buffer memory monitor" screen



↓

"Buffer memory test value check" screen



1. **Press the OK button.**

2. **Select the target address using the ▲ and ▼ buttons and press the OK button. The cursor moves to the buffer memory value. Increase or decrease the value using the ▲ and ▼ buttons.**

3. **Pressing the OK button will update the value set in buffer memory test.**

Point

Since the value displayed in the "buffer memory test value check" screen is used for testing, if the relevant buffer memory is being used in a program, the displayed value may differ from the one set in the "buffer memory test" screen.

4.5 Module Settings

Values set to intelligent function modules can be checked and changed with the display unit.

4.5.1 Initial setting change

The initial setting value of an intelligent function module can be changed.

The changed value is reflected to the CPU module after any of the following operations are performed.

- Powering off and then on the CPU module
- Resetting the CPU module
- Setting the CPU module from STOP to RUN*1

*1 Perform this operation twice. (STOP→RUN→STOP→RUN)

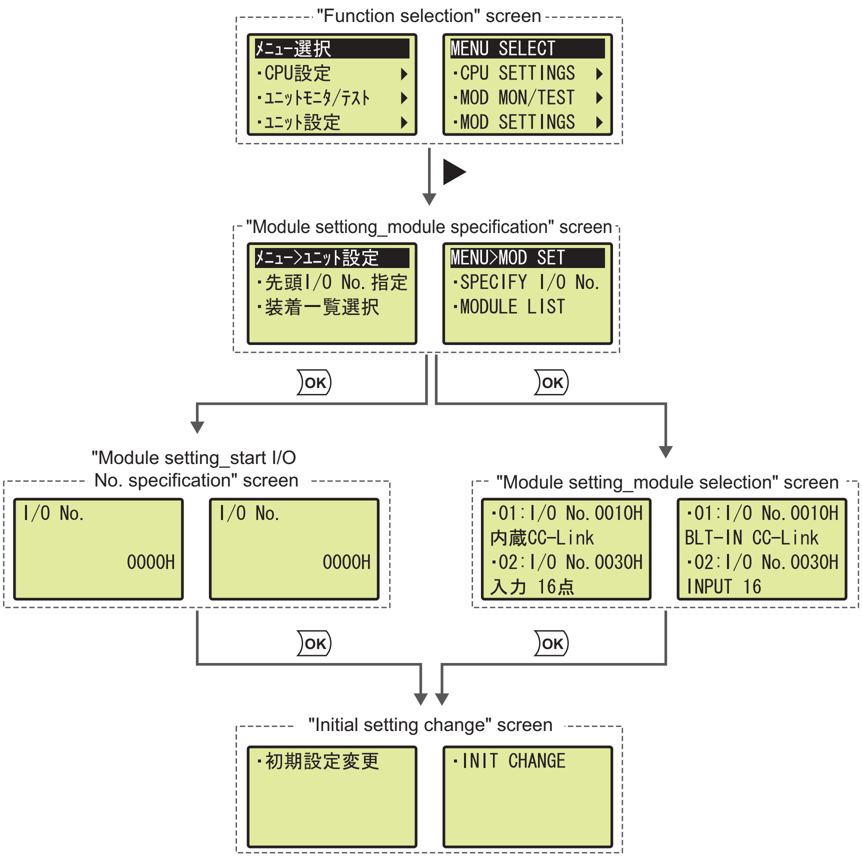
Point

Write the following items to the CPU module before changing the initial setting. If the setting is changed before the items are written, an error message will be displayed.

- Intelligent function module parameters (☞ Page 77, Section 2.11 (1))
- Menu definition file (☞ Page 272, Section 4.5.1 (2))

(1) Screen transition

The following shows screen transition of initial setting change.



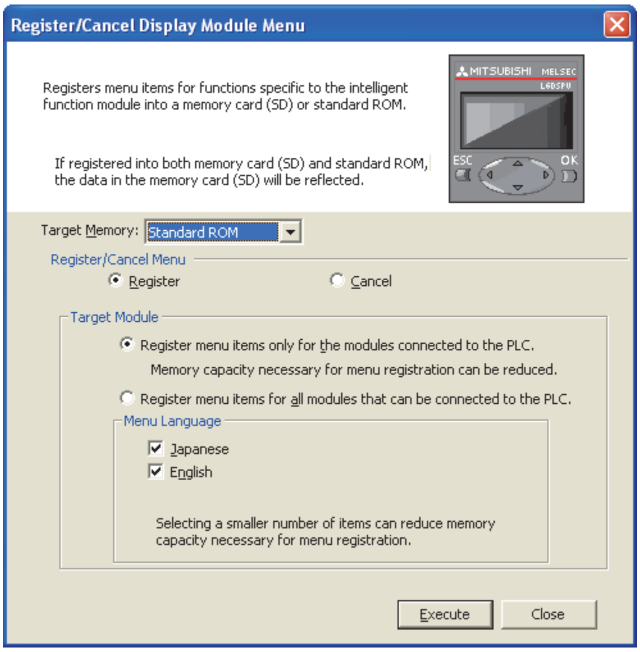
Levels lower than the "initial setting change" screen depend on intelligent function module used.

📖Manual for the intelligent function module used

(2) Registering/canceling display unit menu

To change an initial setting value with the display unit, write a menu definition file to the CPU module. Select the standard ROM or an SD memory card as a storage location.*1*2*3

[Online] ⇨ [Register/Cancel Display Module Menu...]



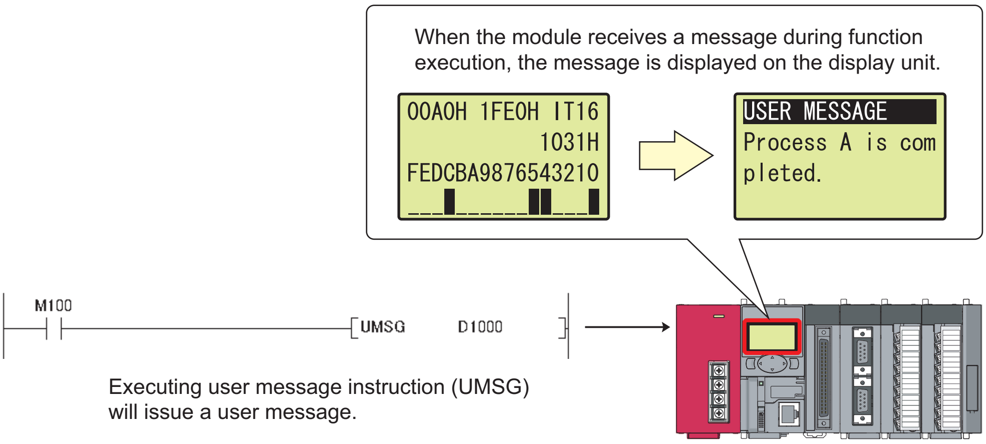
*1 GX Developer cannot write menu definition files.

*2 When a menu definition file has been stored in an SD memory card, the file in the SD memory card will be written even if a menu definition file is created and stored in the standard ROM.

*3 For the versions of GX Works2 that supports intelligent function modules, refer to the following.
GX Works2 Version 1 Operating Manual (Common)

4.6 User Message

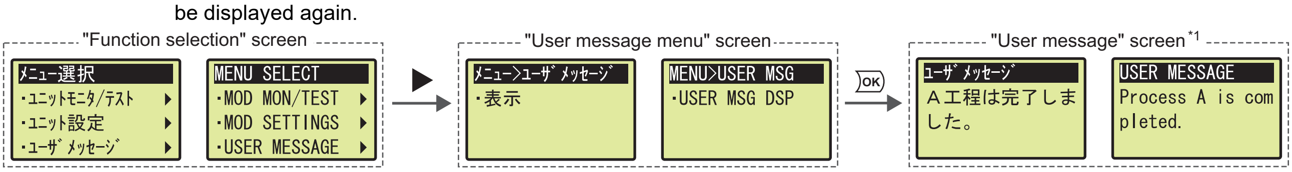
Executing User message instruction (UMSG) displays a user message on the display unit. User messages are user-defined messages for display unit.



For details on User message instruction, refer to the following.
MELSEC-Q/L Programming Manual (Common Instruction)

Point

- Under the "USER MESSAGE" menu selected from the "function selection" screen, the user message displayed last can be displayed again.



However, the user message will not be displayed if it is not held in the following cases.
  - A user message is attempted to be displayed after User message instruction where empty character strings have been specified as an argument is executed.
  - A user message is attempted to be displayed while User message instruction has not been executed after the CPU module is powered off and then on or is reset.
  - User message instruction where a null code (00_H) has been specified as an argument is executed while a user message is displayed.

*1 When a displayed language is switched by language setting, the language of a user message is not changed.

- A user message is displayed whenever the User message instruction is on the rising edge or there is a change in the message. If the intervals between the rising edges or changes in user message are too short, a user message will pop up continually and therefore, operation of the display unit will be frequently interrupted. In this case, holding down the ESC button on the display unit for a while will disable the pop-up display. For how to enable pop-up display, refer to ☞ Page 286, Section 4.8.4.

4.7 Memory Card Operation

File operations between the CPU module and an SD memory card can be performed with the display unit.
The memory card operation includes the following functions.

- File list/delete ("FILE LIST/DEL")
- Free space check ("FREE SPACE")
- Batch save ("BATCH SAVING")
- Batch load ("BATCH LOADING")

Point

To perform the memory card operation, an SD memory card must be inserted to the CPU module, and use of the card must be enabled.
If any one of the memory card operation is executed under the following conditions, an error message is displayed.

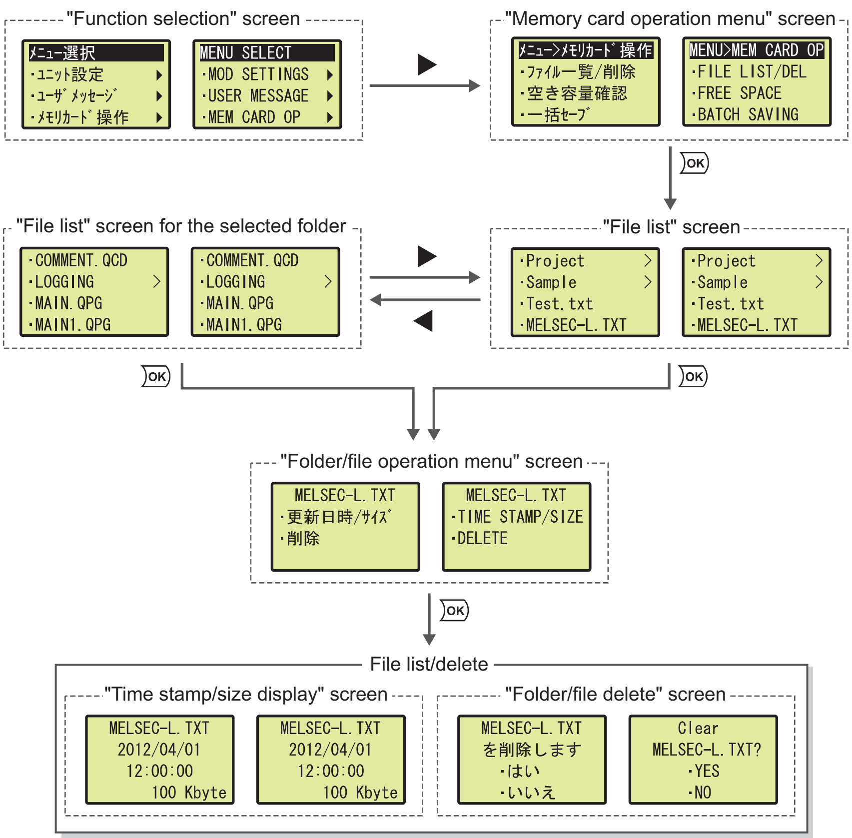
- An SD memory card is not inserted.
- Use of an SD memory card is disabled by SM606 (SD memory card forced disable instruction).

4.7.1 File list/delete

Folders and files stored in an SD memory card can be displayed in a list.
The time stamp and size information of a specified folder or file can be checked. Also, a specified folder or file can be deleted.

(1) Screen transition

The following shows screen transition of the file list/delete function.



(2) Displaying a list

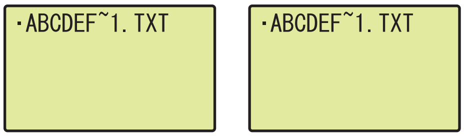
Folders and files stored in an SD memory card are displayed on the "File list" screen.

- For folder names, ">" is displayed at right of the name.
- A display unit displays information of 100 folders and files at a time. If there is information of more than 100 folders and files, "NEXT" is displayed on the last line of the screen. Select "NEXT" to display another set of 100 folders and files. This time, "PREV" is displayed on the first line of the screen. Select "PREV" to return the previous list.

Remark

- If the number of characters used for the name of a folder or file stored in the SD memory card exceeds the following ranges, the name cannot be displayed in one line. The name will be shortened and displayed.
  - File name: 15 one-byte characters (including an extension)
  - Folder name: 13 one-byte characters

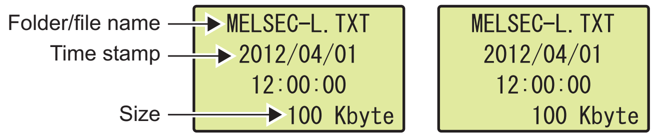
  Example: The following is a display example of the file name "abcdefghijkl.txt".



- Folders and files are displayed up to the sixth level of the hierarchy (including a root directory as the first level).

(3) Displaying the time stamp and size

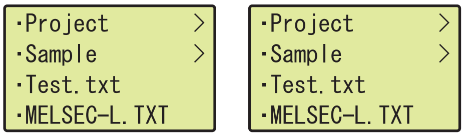
The time stamp and size of folders and files stored in an SD memory card are displayed.



(a) Operating procedure

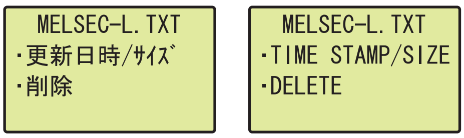
The following is the procedure for displaying the time stamp and size information.

"File list" screen



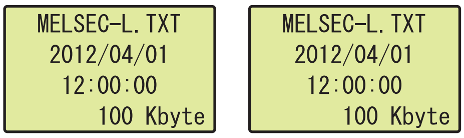
↓

"Folder/file operation menu" screen



↓

Time stamp/size" screen



1. **Select a folder or file using the ▲ and ▼ buttons in the "File list" screen, and press the OK button.**

2. **Select "TIME STAMP/SIZE" using the ▲ and ▼ buttons and press the OK button.**

3. **The time stamp and size of the specified folder or file are displayed.**

To return to the "Folder/file operation menu" screen, 1press the OK button.

(4) Deleting a folder or file

Folders and files stored in an SD memory card are deleted.

(a) Operating procedure

The following is the procedure for deleting a folder or file.

"File list" screen

･Project > ･Sample > ･Test.txt ･MELSEC-L.TXT	･Project > ･Sample > ･Test.txt ･MELSEC-L.TXT

1. **Select a folder or file using the ▲ and ▼ buttons in the "File list" screen, and press the OK button.**

↓

"Folder/file operation menu" screen

MELSEC-L.TXT ･更新日時/ｻｲｽﾞ ･削除	MELSEC-L.TXT ･TIME STAMP/SIZE ･DELETE

2. **Select "DELETE" using the ▲ and ▼ buttons and press the OK button.**

↓

"Folder/file delete" screen

MELSEC-L.TXT を削除します ･はい ･いいえ	Clear MELSEC-L.TXT? ･YES ･NO

3. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

↓

"Folder/file delete completion" screen

ﾌｫﾙﾀﾞ/ﾌｧｲﾙ削除が 完了しました	Folder/File cleared

4. **The specified folder or file is deleted.**

To return to the "File list" screen, press the OK button.

Point

When a folder is specified, all folders and files in the specified folder are deleted.
However, if the folder contains any read-only file, the read-only file will not be deleted and the folder remains. (Folders and files other than the read-only file will be deleted.)
To delete the folder that contains any read-only file, delete the read-only file first, and then delete the folder.

(b) Precautions

The following online functions cannot be executed while a folder or file is being deleted. If executed, an error response is returned to the request source

Category	Function
Drive operation	Format PLC memory
	Clear PLC memory (Clear all file registers)
	Write title
	Arrange PLC memory
File operation	Write to PLC
	Delete PLC data
	Write PLC user data
	Delete PLC user data
	Password registration
Trace	Sampling trace registration
Data logging	Data logging registration
FTP function	All operations and commands
Others	CPU module change function with SD memory card
	Register/cancel display unit menu

4.7.2 Free space display

The free space on the SD memory card that is inserted to the CPU module can be displayed with the display unit.

ﾒﾓﾘｶｰﾄﾞ空き容量: 3978784 Kbyte	Free space: 3978784 Kbyte

(1) Operating procedure

The following is the procedure for displaying the free memory space.

"Memory card operation menu" screen

ﾒﾆｭｰ>ﾒﾓﾘｶｰﾄﾞ操作 ・空き容量確認 ・一括ｾｰﾌﾞ ・一括ﾛｰﾄﾞ	MENU>MEM CARD OP ・FREE SPACE ・BATCH SAVING ・BATCH LOADING

↓

1. Select "FREE SPACE" using the ▲ and ▼ buttons and press the OK button.

"Free space display menu" screen

ﾒﾆｭｰ>ｶｰﾄﾞ>容量 ・表示	MENU>CARD>SPACE ・FREE SPACE DSP

↓

2. Press the OK button.

"Free space display" screen

ﾒﾓﾘｶｰﾄﾞ空き容量: 3978784 Kbyte	Free space: 3978784 Kbyte

3. Free space of the SD memory card is displayed.

Press the OK button to return to the "Memory card operation menu" screen.

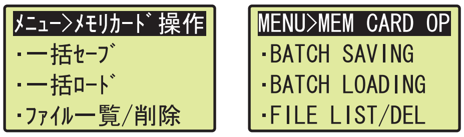
4.7.3 Batch save

Data (such as program file and parameter file) in the CPU module can be saved into an SD memory card.

(1) Operating procedure

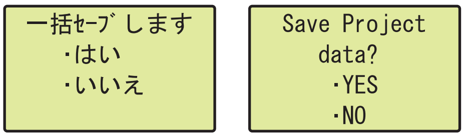
The following is the procedure for batch-saving data.

"Memory card operation menu" screen



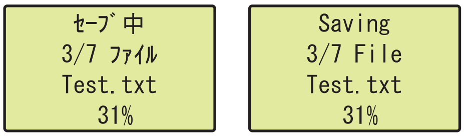
1. **Select "BATCH SAVING" using the ▲ and ▼ buttons and press the OK button.**

"Batch save execution" screen



2. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

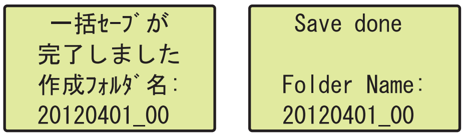
"Batch saving" screen



3. **The batch data processing starts.*1**

During the batch data processing, button operations on the display unit are disabled.

"Batch save completion" screen

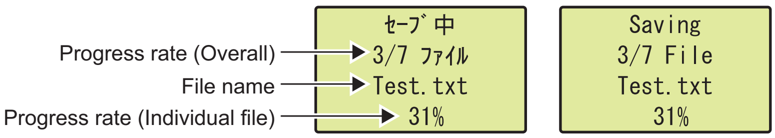


4. **When the processing is completed successfully, either of the screens shown left appears.**

To return to the "Memory card operation menu" screen, press the OK button.

A "SaveLoad" folder is created in the SD memory card.

*1 The following items are displayed on the "Batch saving" screen.



4.7.4 Batch load

Data saved in the SD memory card by the batch save function can be read to the CPU module.
This function can be executed only when the CPU module is in STOP status.

(1) Operating procedure

The following is the procedure for batch-loading data.

"Memory card operation menu" screen

ﾒﾆｭｰ>ﾒﾓﾘｶｰﾄﾞ操作 ・一括ﾛｰﾄﾞ ・ﾌｧｲﾙ一覧/削除 ・空き容量確認	MENU>MEM CARD OP ・BATCH LOADING ・FILE LIST/DEL ・FREE SPACE

1. **Select "BATCH LOADING" using the ▲ and ▼ buttons and press the OK button.**

↓

"Batch load target folder" screen

・20120401_00 ＞ ・20120401_01 ＞ ・20120401_02 ＞ ・20120401_03 ＞	・20120401_00 ＞ ・20120401_01 ＞ ・20120401_02 ＞ ・20120401_03 ＞

2. **Select a load-target folder using the ▲ and ▼ buttons and press the OK button.**
Folders in the "SaveLoad" folder can only be loaded.

↓

"Batch load execution" screen

20120401_00 を一括ﾛｰﾄﾞします ・はい ・いいえ	Load Project 20120401_00? ・YES ・NO

3. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**
To overwrite read-only files, the overwrite confirmation screen appears.

↓

"Drive format execution" screen

全ﾄﾞﾗｲﾌﾞをﾌｫｰﾏｯﾄ します(ｶｰﾄﾞ除く) ・はい ・いいえ	Format all built-in memory ・YES ・NO

4. **Select "YES" using the ▲ and ▼ buttons to format all drives in the CPU module and press the OK button.** *1
When "NO" is selected, the batch load function is executed without formatting the drives.

↓

"Drive formatting" screen

ﾌｫｰﾏｯﾄ中	Formatting

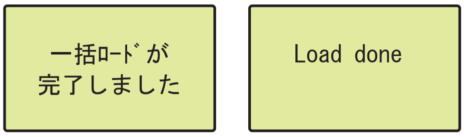
5. **When all drives are formatted, the "Drive formatting" screen is displayed and then the "Batch loading" screen is displayed.** *2
During the format processing or the batch load processing, button operations on the display unit are disabled.

"Batch loading" screen

ﾛｰﾄﾞ中 2/7 ﾌｧｲﾙ PARAM.QPA 27%	Loading 2/7 File PARAM.QPA 27%

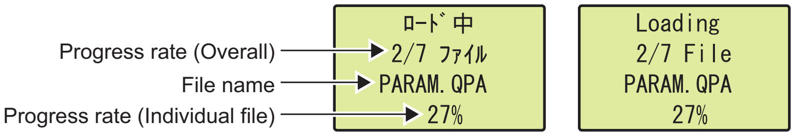
↓

"Batch load completion" screen



6. When the processing is completed successfully, either of the screens shown left appears.
To return to the "Memory card operation menu" screen, press the OK button.

*1 The program memory, standard RAM, and standard ROM are formatted. The SD memory card is not formatted.
*2 The following items are displayed on the "Batch loading" screen.



Point

- The batch load processing cannot be performed if the SD memory card is write-protected.
  Remove write protection from the SD memory card, and execute the function.
- If a folder or file (including read-only files) with the same name exists in the load-destination CPU module, data in the corresponding folder or file will be overwritten.
- When the batch load function is executed without formatting the drives, data in the folders that are not targeted for batch load in the load-destination CPU module will remain the same.

4.8 Options

The display format and basic settings of the display unit can be configured under the "OPTIONS" menu selected from the "function selection" screen. The option settings can be backed up to the CPU module without a battery.* 1*2

*1 The pop-up setting is not backed up.
*2 When the display unit is removed and attached to another CPU module, the display unit operates according to settings stored in the CPU module to which the display unit is currently attached.

The option settings include the following settings.

- Language setting
- Contrast adjustment
- Lighting period setting
- Pop-up display

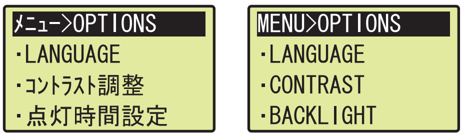
4.8.1 Language setting

A displayed language can be changed.

(1) Operating procedure

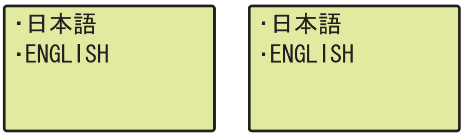
The following is the procedure for setting the language.

"Options" screen



↓

"Language setting" screen



1. **Select "LANGUAGE" using the ▲ and ▼ buttons and press the OK button.**

2. **Switch the language using the ▲ and ▼ buttons and press the OK button.**

Point

Information of language displayed on the display unit can be checked in SD581 (Displayed language information).
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

4.8.2 Contrast adjustment

Tone of displayed characters can be set.

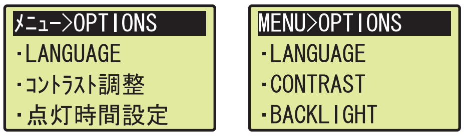
(1) Setting range

Tone can be selected from 0 (light) to 9 (dark). (default: 3)

(2) Operating procedure

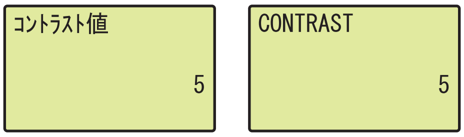
Perform the following operation.

"Options" screen



↓

"Contrast adjustment" screen



1. **Select "CONTRAST" using the ▲ and ▼ buttons and press the OK button.**

2. **Set a contrast value using the ▲ and ▼ buttons and press the OK button.**

4.8.3 Lighting period setting

The lighting period of the backlight on the display unit can be set.

(1) Setting range

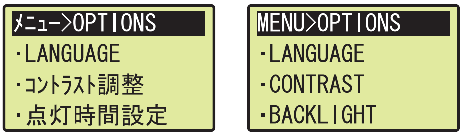
The following shows periods that can be set. (default: 5MIN)

• 1MIN • 3MIN • 5MIN • 10MIN • 15MIN • 30MIN • 60MIN • LIGHT ALWAYS ON

(2) Operating procedure

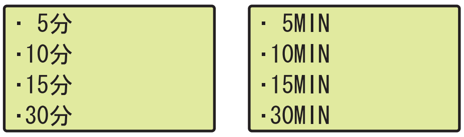
Perform the following operation.

"Options" screen



↓

"Lighting period setting" screen



1. **Select "BACKLIGHT" using the ▲ and ▼ buttons and press the OK button.**

2. **Set a lighting period using the ▲ and ▼ buttons and press the OK button.**

4.8.4 Pop-up display

A pop-up screen showing a user message (☞ Page 273, Section 4.6) can be enabled.

(1) Operating procedure

Enable pop-up display by the following operation.

"Options" screen

```
ﾒﾆｭｰ>OPTIONS          MENU>OPTIONS
･ｺﾝﾄﾗｽﾄ調整            ･CONTRAST
･点灯時間設定           ･BACKLIGHT
･ﾎﾟｯﾌﾟｱｯﾌﾟ     ▶       ･POPUP        ▶
```

↓

"Pop-up" screen

```
ﾒﾆｭｰ>OPT>ﾎﾟｯﾌﾟ         MENU>OPT>POPUP
･禁止状態解除           ･POP BLK DISABLE
```

↓

"Pop-up setting confirmation" screen

```
ﾎﾟｯﾌﾟｱｯﾌﾟ禁止          Enable popups?
状態を解除します
･はい                  ･YES
･いいえ                ･NO
```

↓

"Pop-up setting completion" screen

```
ﾎﾟｯﾌﾟｱｯﾌﾟ              Popups enabled.
禁止状態を
解除しました
```

1. **Select "POPUP" using the ▲ and ▼ buttons and press the ▶ button.**

2. **Press the OK button.**

3. **Select "YES" using the ▲ and ▼ buttons and press the OK button.**

4. **When the setting is completed, either of the screens shown left appears.**
Press the OK button to return to the "Pop-up" screen.

PART 3 DEVICES, CONSTANTS

In this part, the devices and constants used in the CPU module are described.

CHAPTER 5 DEVICES 288
CHAPTER 6 CONSTANTS 342
CHAPTER 7 CONVENIENT USAGE OF DEVICES 344

CHAPTER 5 DEVICES

This chapter describes the devices that can be used in the CPU module.

5.1 Device List

The following table shows the devices used in the CPU module and applicable ranges.

Classification	Type	Device name	Default			Parameter-set range	Reference
			Points	Range			
Internal user device	Bit device	Input	8192	X0 to X1FFF	Hexadecimal	Cannot be set	Page 293, Section 5.2.1
		Output	8192	Y0 to Y1FFF	Hexadecimal		Page 293, Section 5.2.2
		Internal relay	8192	M0 to M8191	Decimal	Can be set (up to 29K words for internal user devices in total)[*8*9]	Page 294, Section 5.2.3
		Latch relay	8192	L0 to L8191	Decimal		Page 294, Section 5.2.4
		Link relay	8192	B0 to B1FFF	Hexadecimal		Page 294, Section 5.2.5
		Annunciator	2048	F0 to F2047	Decimal		Page 295, Section 5.2.6
		Link special relay	2048	SB0 to SB7FF	Hexadecimal		Page 298, Section 5.2.7
		Edge relay	2048	V0 to V2047	Decimal		Page 298, Section 5.2.8
		Step relay	8192	S0 to S8191	Decimal		Page 299, Section 5.2.9
	• Bit device (contacts and coils) • Word device (current value)	Timer	2048	T0 to T2047	Decimal		Page 299, Section 5.2.10
		Retentive timer	0	(ST0 to ST2047)	Decimal		
		Counter	1024	C0 to C1023	Decimal		Page 308, Section 5.2.11
	Word device	Data register	12288	D0 to D12287	Decimal		Page 311, Section 5.2.12
		Link register	8192	W0 to W1FFF	Hexadecimal		Page 312, Section 5.2.13
		Link register	2048	SW0 to SW7FF	Hexadecimal		Page 313, Section 5.2.14
Internal system device	Bit device	Function input	16	FX0 to FXF	Hexadecimal	Cannot be set	Page 314, Section 5.3.1
		Function output	16	FY0 to FYF	Hexadecimal		
		Special relay	2048	SM0 to SM2047	Decimal		Page 316, Section 5.3.2
	Word device	Function register	5	FD0 to FD4	Decimal		Page 314, Section 5.3.1
		Special register	2048	SD0 to SD2047	Decimal		Page 316, Section 5.3.3

Classification	Type	Device name	Default Points	Default Range		Parameter-set range	Reference
Link direct device[*6]	Bit device	Link input	16384	Jn\X0 to Jn\X3FFF	Hexadecimal	Cannot be set	Page 317, Section 5.4
		Link output	16384	Jn\Y0 to Jn\Y3FFF	Hexadecimal		
		Link special relay	512	Jn\SB0 to Jn\SB1FF	Hexadecimal		
	Word device	Link register	16384	Jn\W0 to Jn\W3FFF[*7]	Hexadecimal		
		Link special register	512	Jn\SW0 to Jn\SW1FF	Hexadecimal		
Module access device	Word device	Intelligent function module device	65536	Un\G0 to Un\G65535[*2]	Decimal	Cannot be set	Page 320, Section 5.5
Index register or standard device register	Word device	Index register or standard device register	20	Z0 to Z19	Decimal	Cannot be set	Page 322, Section 5.6
File register	Word device	File register	0	—	Decimal	0 to 384K points in total[*3] (in increments of 1K)	Page 326, Section 5.7
Extended data register	Word device	Extended data register	128K[*1]	D12288 to D143359[*1]	Decimal		Page 330, Section 5.8
Extended link register	Word device	Extended link register	0	—	Hexadecimal		
Nesting	—	Nesting	15	N0 to N14	Decimal	Cannot be set	Page 335, Section 5.9
Pointer	—	Pointer	4096[*10]	P0 to P4095[*10]	Decimal	Cannot be set	Page 336, Section 5.10
		Interrupt pointer	256	I0 to I255	Decimal		Page 339, Section 5.11
Others	Bit device	SFC block device	320[*4]	BL0 to BL319[*4]	Decimal	Cannot be set	Page 340, Section 5.12.1
	—	I/O No. specification device	—	U0 to UFF[*5]	Hexadecimal		Page 340, Section 5.12.2
		Macro instruction argument device	10	VD0 to VD9	Decimal		Page 341, Section 5.12.3

*1 The number of points for the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P are 32K points (D12288 to D45055).
*2 The number of points that can be actually used varies depending on the intelligent function module.
Manual for the intelligent function module used
*3 The range for the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P is 0 to 64K in total.
*4 The number of points for the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P is 128 points (BL0 to BL127).
*5 The range for the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P is U0 to U3F.
*6 This device can be used in the CPU module whose serial number (first five digits) is "13012" or later.
*7 For CC-Link IE Field Network, J□\W0 to J□\W1FFF are for RWw (8192 points), and J□\W2000 to J□\W3FFF are for RWr (8192 points).
*8 When changing the number of device points, refer to the precautions on Page 293, Section 5.2 (1).
*9 The device point of the step relay for the CPU module whose serial number (first five digits) is "15101" or earlier is either 0K or 8K.
*10 For the L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT with a serial number (first five digits) of "16042" or later, the pointer for auto assignment device can be used up to 32768 points by using the GX Works2. (The PC parameters should be set. (Page 362, Appendix 1.2 (8)))

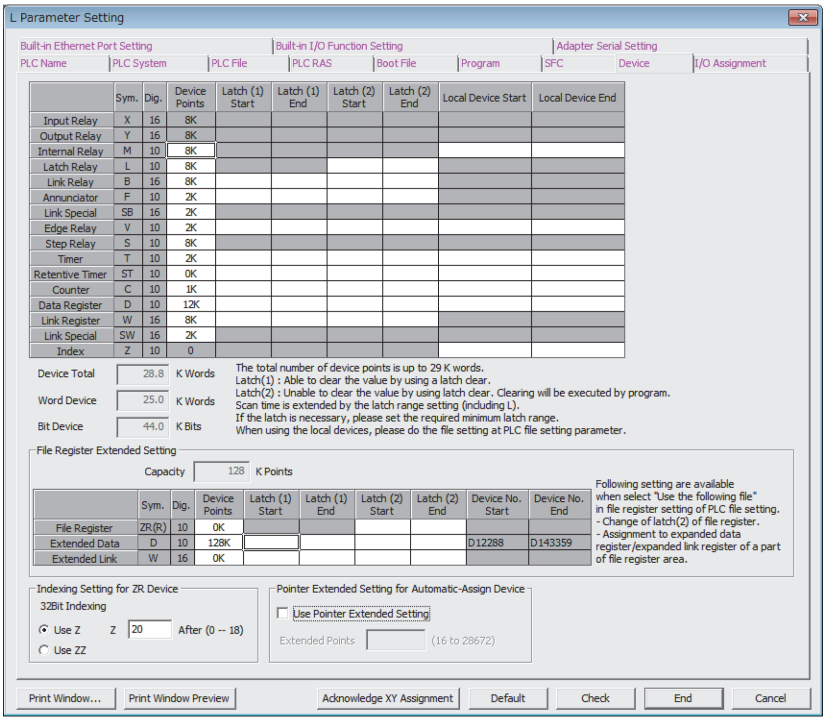
5.2 Internal User Devices

Internal user devices can be used for various user applications.

(1) Points for internal user devices

Set the number of device points to be used.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Device]



When changing device points, note the following.

- The points for the input (X), output (Y) cannot be changed.
- Set each device in increments of 16 points.
- Up to 29K words can be set for total internal user devices.
- The maximum number of points for a bit device is 32K. For an internal relay and link relay, the maximum number of points can be set up to 60K.
- One point of the timer, retentive timer, and counter is regarded as one word device point and two bit device points. (☞ Page 291, Section 5.2 (2))

Point

- When changing device points, the following refresh ranges must not exceed the corresponding device ranges.
    - Link refresh with CC-Link IE Field Network master/local module
    - Link refresh with CC-Link IE Field Network Basic
    - Auto refresh with CC-Link
    - Auto refresh with intelligent function module

  If device points are set exceeding the corresponding device range, data may be written to any other device or an error may occur.
- If the device points of the internal user devices are changed and the parameters are written from the "Write to PLC" screen, the device address may be shifted and does not correspond to the original stored value. Because the shifted value might be used for the operation, the following files, which are created by using the parameters before the device point change, cannot be used under existing condition.
    - Sequence program files
    - SFC program files
    - ST program files

  When change the device points of the internal user devices, perform the following operations from a programming tool.
  [Before changing the device points of the internal user devices]
  Read devices to be used and each program from the CPU module.
  [After the device points of the internal user devices are changed]
  Write the devices and each program, which were read before the device point change, to the CPU module.

  For the read/write of devices and programs, refer to the following.
  Operating manual for the programming tool used

(2) Memory size

Set the internal user devices so that the following condition is satisfied.

(Bit device size) + (Timer, retentive timer, and counter sizes) + (Word device size) ≤ 29K words

- For bit devices, 16 points are calculated as one word.

$$\text{(Bit device size)} = \frac{(X + Y + M + L + B + F + SB + V + S)}{16} \text{ (Words)}$$

- For the timer, retentive timer, and counter, 16 points are calculated as 18 words.

$$\text{(Timer, retentive timer, or counter size)} = \frac{(T + ST + C)}{16} \times 18 \text{ (Words)}$$

- For word devices, 16 points are calculated as 16 words.

$$\text{(Word device size)} = \frac{(D + W + SW)}{16} \times 16 \text{ (Words)}$$

(3) Device point assignment example

The following table shows device point assignment examples based on the device point assignment sheet in Appendix.6.

Device name	Symbol	Numeric notation	Number of device point[*2]		Restriction check			
			Points	Range	Size (words)[*3]		Points (bits)[*2]	
Input relay[*1]	X	16	8K (8192)	X0000 to X1FFF	÷ 16	512	×1	8192
Output relay[*1]	Y	16	8K (8192)	Y0000 to Y1FFF	÷ 16	512	×1	8192
Internal relay	M	10	16K (16384)	M0 to M16383	÷ 16	1024	×1	16384
Latch relay	L	10	4K (4096)	L0 to L4095	÷ 16	256	×1	4096
Link relay	B	16	4K (4096)	B0000 to B0FFF	÷ 16	256	×1	4096
Annunciator	F	10	1K (1024)	F0 to F1023	÷ 16	64	×1	1024
Link special relay	SB	16	2K (2048)	SB0000 to SB07FF	÷ 16	128	×1	2048
Edge relay	V	10	1K (1024)	V0 to V1023	÷ 16	64	×1	1024
Step relay[*1]	S	10	8K (8192)	S0 to S8191	÷ 16	512	×1	8192
Timer	T	10	2K (2048)	T0 to T2047	$\times\frac{18}{16}$	2304	×2	4096
Retentive timer	ST	10	2K (2048)	ST0 to ST2047	$\times\frac{18}{16}$	2304	×2	4096
Counter	C	10	1K (1024)	C0 to C1023	$\times\frac{18}{16}$	1152	×2	2048
Data register	D	10	14K (14336)	D0 to D14335	×1	14336	—	
Link register	W	16	4K (4096)	W0000 to W4095	×1	4096	—	
Link special register	SW	16	2K (2048)	SW0000 to SW07FF	×1	2048	—	
Total					29568 (29696 or less)		63488	

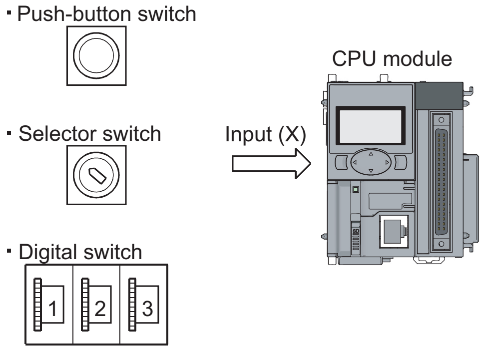
*1 The points are fixed by the system (cannot be changed) for the LCPU whose serial number (first five digits) is "15101" or earlier. However, the points for the step relay (S) can be changed to 0.

*2 Up to 32K points can be set for each device (60K for the internal relay and link relay).

*3 Enter the values multiplied or divided by the number shown in the Size (words) column.

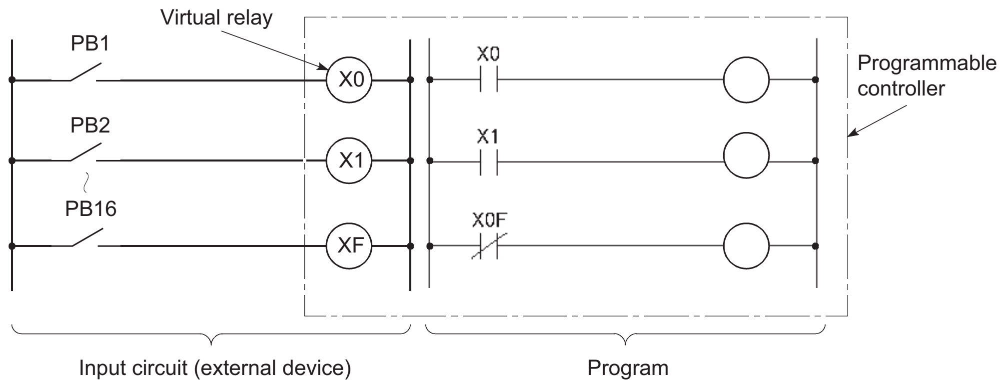
5.2.1 Input (X)

Inputs are the device used to obtain the on and off information from external devices to a CPU module.



(1) Concept of input

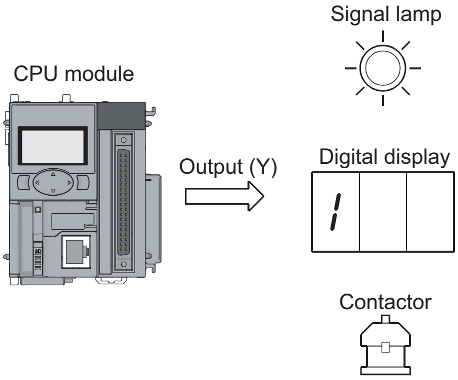
One input point is assumed to be a virtual relay Xn in the CPU module. Programs use the normally open or closed contact of Xn.



The input (X) can also be used for refresh target device (CPU module side) of remote input (RX) in CC-Link IE Field Network, CC-Link IE Field Network Basic, or CC-Link.

5.2.2 Output (Y)

The output (Y) is used to output control results in programs to external devices.



5.2.3 Internal relay (M)

The internal relay (M) is a device for auxiliary relays used in the CPU module. All of the internal relay are set to off when:

- the CPU module is powered off and then on,
- the CPU module is reset, or
- the latch clear is executed. (☞ Page 88, Section 3.4)

Point

The internal relay cannot be latched (data retention during power failure). Use the latch relay (L) when latch is required. (☞ Page 294, Section 5.2.4)

5.2.4 Latch relay (L)

The latch relay (L) is a device for auxiliary relays that can be latched inside the CPU module. During power failure, on/off information are retained by the battery in the CPU module. The previous on/off information are retained even after performing the following to the CPU module.

- Power OFF→ON
- Reset

The latch relay is turned off by the latch clear operation.

Point

Scan time is prolonged when the latch relay is used. (☞ Page 353, Appendix 1.2)

5.2.5 Link relay (B)

The link relay is an internal relay whose point is input in hexadecimal. In the range where the network parameters are not set, this relay can be used as an internal relay or a latch relay.

- The range where a link relay is not latched: Same as the range for an internal relay
- The range where a link relay is latched: Same as the range for a latch relay

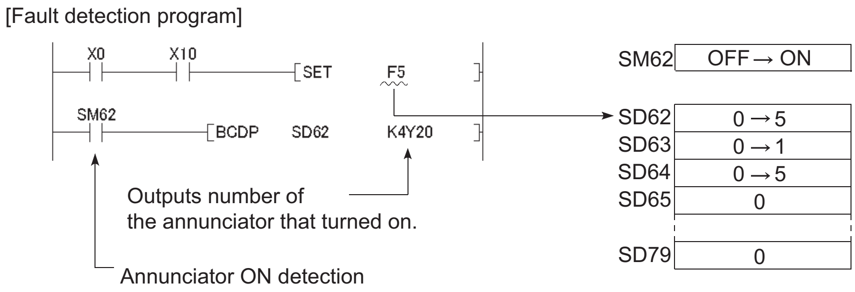
Point

When using a link device in the network module by using the points of the link register in the CPU module (default: 8192 points) or greater, change the points setting of the link register in the "Device" of the "PLC Parameter".

5.2.6 Annunciator (F)

The annunciator (F) is an internal relay which can be effectively used in fault detection programs for user-created system. Whenever an annunciator is turned on, SM62 turns on and the annunciator number is stored in SD62 to SD79. By monitoring SD62 to SD79, the system can be checked for error and failure.

Ex. When F5 is turned on, the corresponding annunciator number is output to the outside.



(1) Turning on the annunciator

Use either of the following instructions.

(a) SET F□ instruction

Annunciator turns on only on the leading edge of an input condition. Even if the input condition turns off, the annunciator is held on. Using many annunciator numbers can shorten scan time more than using the OUT F□ instruction.

(b) OUT F□ instruction

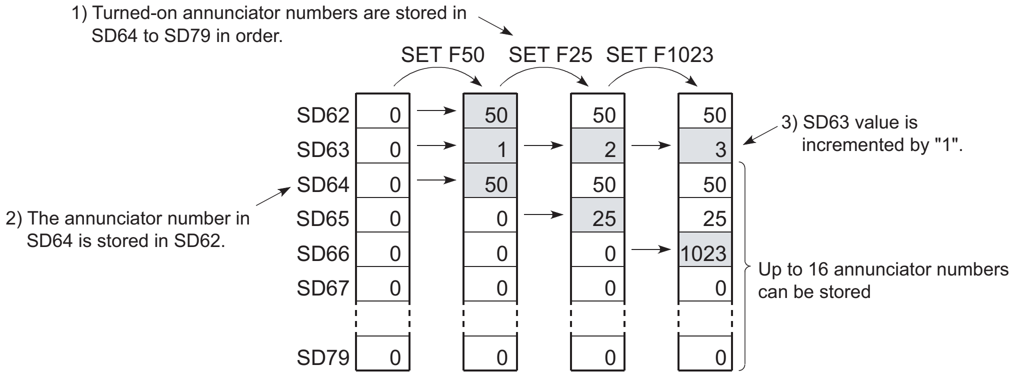
Since the processing is performed for every scan, the scan time is longer than the case of using the SET F□ instruction.

Point

If the annunciator is turned on with any instruction other than SET F□ and OUT F□ (for example, the MOV instruction), the same operation as the internal relay (M) is performed. The ON information is not stored in SM62, and annunciator numbers are not stored in SD62 and SD64 to SD79.

(2) Processing after annunciator on

Whenever an annunciator is turned on, SM62 turns on and the following data are stored in SD62 to SD79. The annunciator number in SD62 is registered to the memory for error history storage.



The USER LED on the front of the CPU module turns on (red).

Point

Whether to turn on the USER LED can be set using the LED control function (☞ Page 169, Section 3.26).

(3) Turning off the annunciator

Use any of the following instructions.

(a) RST F□ instruction

This is used to turn off the annunciator number that was turned on with the SET F□ instruction.

(b) LEDR instruction

This is used to turn off the annunciator number stored in SD62 and SD64.

(c) BKRST instruction

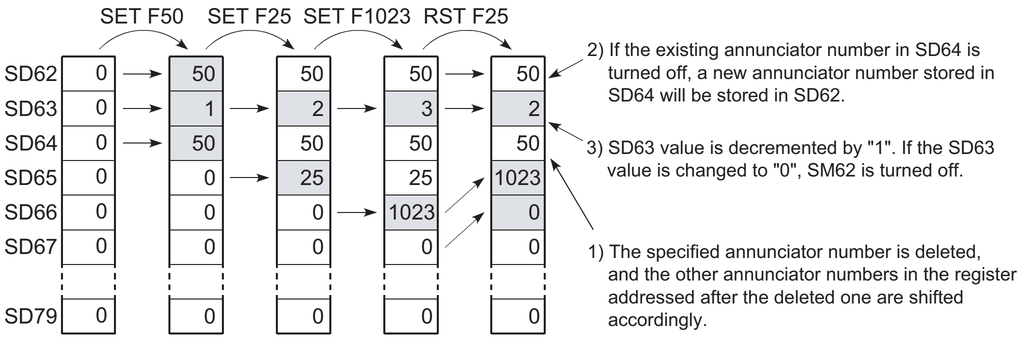
This is used to turn off all of the annunciator numbers within the specified range.

(d) OUT F□ instruction

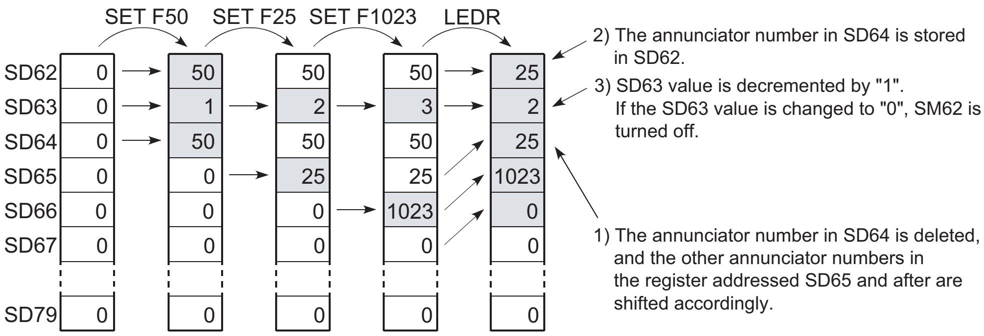
One annunciator number can be turned on or off with the same instruction. However, even if an annunciator number is turned off with the OUT F□ instruction, the off processing described in (4) in this section is not performed. If the annunciator is turned off with the OUT F□ instruction, execution of the RST F□ LEDR, or BKRST instruction is required.

(4) Processing after annunciator off

- Data stored in the special register (SD62 to SD79) when the annunciator is turned off with the RST F☐ or BKRST instruction



- Data stored in the special register (SD62 to SD79) after execution of the LEDR instruction



Point

If the LEDR instruction is executed while the annunciator is on and at the same time the operation continuation error that has higher priority than the annunciator has occurred, the LEDR instruction clears the higher priority error. To execute the LEDR instruction, remove the error whose priority is higher than that of the annunciator.

- LED indication

When all of the annunciator numbers in SD64 to SD79 turn off, the USER LED will turn off.

5.2.7 Link special relay (SB)

The link special relay (SB) is a relay that indicates communication status and error detection status of the CC-Link IE Field Network master/local module or the CC-Link system master/local module. The relay is turned on or off according to various factors that occur during data link. Status of the data link can be confirmed by monitoring the link special relay.

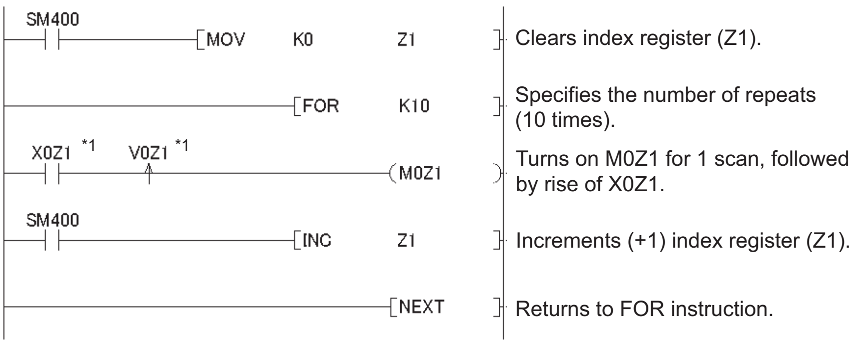
📖Manuals for each network module

5.2.8 Edge relay (V)

The edge relay (V) is a device in which the on/off information from the beginning of the ladder block. The EGP/EGF instructions only can be used. The edge relay of the same number cannot be set.

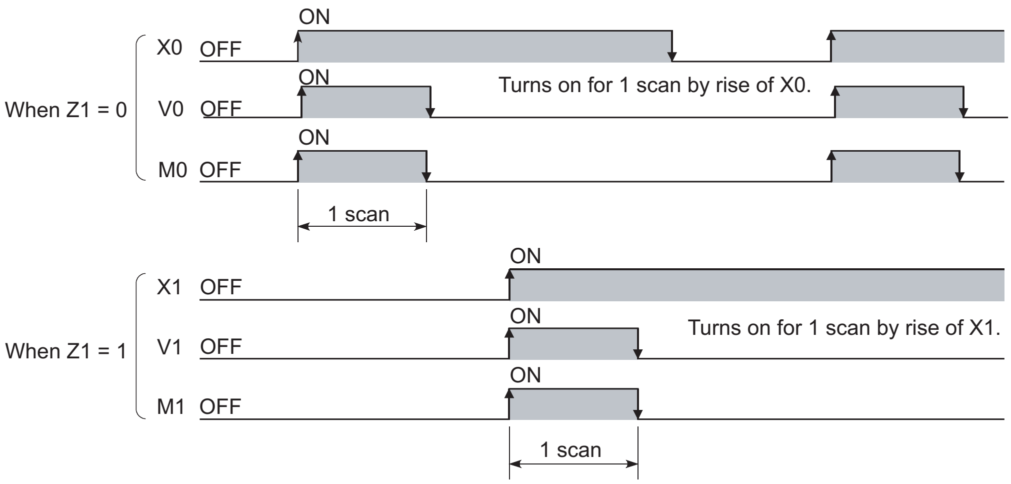
Ex. Detecting a leading edge and execute it in the program configured using index modification

[Program example]



*1 The on/off information for X0Z1 is stored in the V0Z1 edge relay. For example, the on/off information of X0 is stored in V0, and that of X1 is stored in V1.

[Timing chart]



5.2.9 Step relay (S)

This device is provided for SFC programs. (MELSEC-Q/L/QnA Programming Manual (SFC))

5.2.10 Timer (T, ST)

Time counting starts when a coil is turned on, and it times out and the contact turns on when the current value reaches the set value.

(1) Timer types

Timers are mainly classified into the following two types.

- Timer of which value is set to 0 and contact is turned off when a coil is turned off
- Retentive timer that holds both the current value and contact status even if the coil is turned off

Also, each timer has low-speed and high-speed timers.

- Timer
  - Timer
    - Low-speed timer
    - High-speed timer
  - Retentive timer
    - Low-speed retentive timer
    - High-speed retentive timer

(2) Specification of the timer

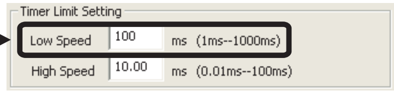
- The same device is used for the low- and high-speed timers, and the type is determined according to the instruction used. (Example: For the OUT T0 instruction, the low-speed timer is specified, and for the OUTH T0 instruction, the high-speed timer is specified.)
- The same device is used for the low- and high-speed retentive timers, and the type is determined according to the instruction used. (Example: For the OUT ST0 instruction, the low-speed retentive timer is specified, and for the OUTH ST0 instruction, the high-speed retentive timer is specified.)

(3) Low-speed timer

This type of timer measures time in increments of 1 to 1000ms. The default is 100ms (in increments of 1ms)

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Enter a value.

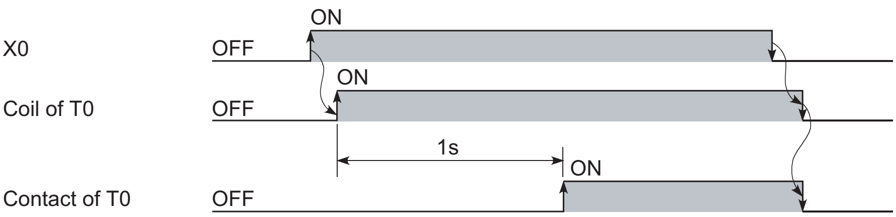


Timer counting starts when its coil is turned on, and the contact is turned on when the current value reaches the set value. The timer's coil is turned off, the current value is changed to "0" and the contact is turned off.

[Program example]

When X0 is turned on, coil of T0 turns on, and the contact turns on after 1s. (when unit of the timer is set to 100ms)

[Timing chart]

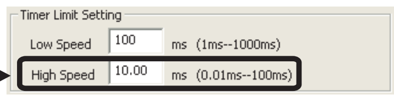


(4) High-speed timer

This type of timer measures time in increments of 0.01 to 100ms. The default is 10.0ms (in increments of 1ms)

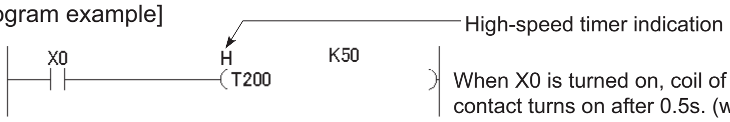
Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

Enter a value.



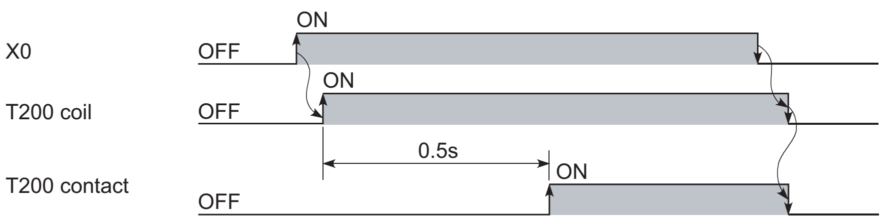
The timer starts time measurement when its coil is turned on, and when it times out, the contact is turned on. If the timer's coil is turned off, the current value is changed to "0" and the contact is turned off.

[Program example]



When X0 is turned on, coil of T200 turns on, and the contact turns on after 0.5s. (when unit of the timer is set to 10ms)

[Timing chart]



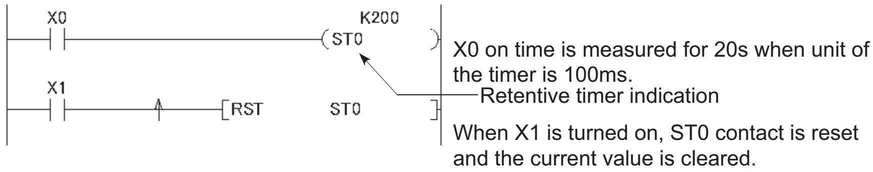
(5) Retentive timer

This timer measures the period of time during which the coil is on. The time increment is set in the same manner as the corresponding low- or high-speed timer.

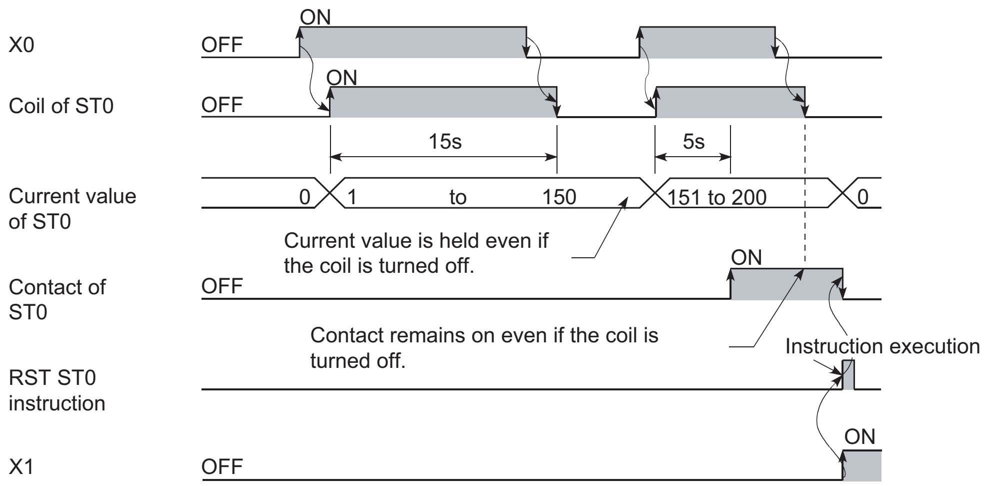
- Low-speed retentive timer: Low-speed timer
- High-speed retentive timer: High-speed timer

The timer starts time measurement when its coil is turned on, and when it times out, the contact is turned on. Even if the timer's coil is turned off, the current value and the on/off status of the contact are retained. When the coil is turned on again, the measurement restarts from the retained current value. The current value and the contact off status can be cleared with the RST ST□ instruction.

[Program example]



[Timing chart]



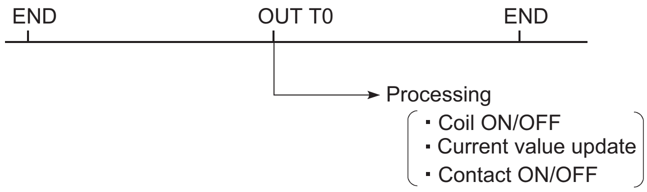
(6) Timer processing and accuracy

(a) Processing

When the OUT T□ or OUT ST□ instruction is executed, the on/off switching of the timer coil, current value update, and on/off switching of the contact are performed. In the END processing, the current timer value is not updated and the contact is not turned on/off.

[Program example]

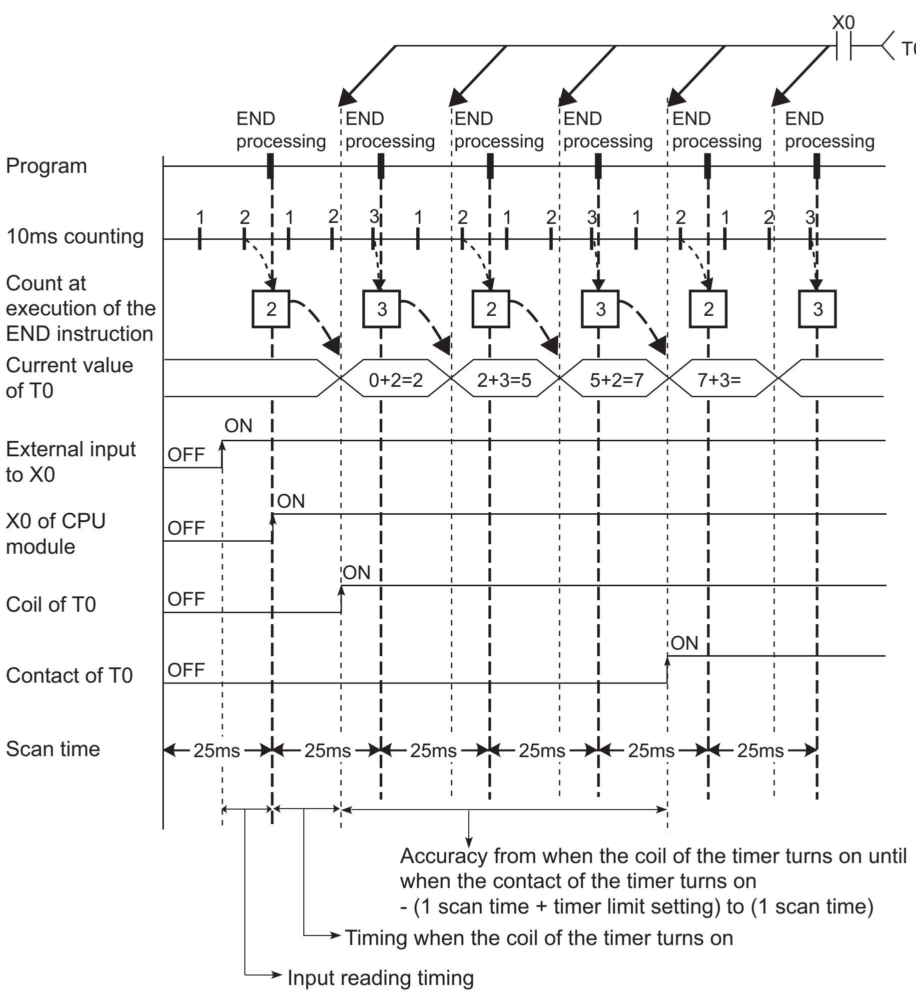
[Processing at execution of OUT T0 instruction]



(b) Accuracy

The value obtained by the END instruction is added to the current value when the OUT T□ or OUT ST□ instruction is executed. The current value is not updated while the timer coil is off even if the OUT T□ or OUT ST□ instruction is executed.

Timer limit setting=10ms, Setting value of T0=8 (10ms×8=80ms), Scan time=25ms



Accuracy of the timer response that is from reading input (X) to output the data are up to "2-scan time + timer limit setting".

(7) Precautions for using timers

(a) Use of the same timer

Do not use the OUT T☐ or OUT ST☐ instruction that describes the same timer more than once within one scan. If used, the current timer value will be updated by each OUT T☐ or OUT ST☐ instruction execution, resulting in incorrect time measurement.

(b) When the timer is not executed in every scan

Current values are not updated for every scan, resulting in incorrect time measurement. While a coil of a timer is on, do not make the OUT T☐ or OUT ST☐ instruction jumped to any other part with another instruction such as CJ.

For subroutine programs where a timer is provided and a coil is on, use the timer once in each scan.

(c) Programs that cannot use timers

Timers cannot be used in interrupt programs and fixed scan execution programs.

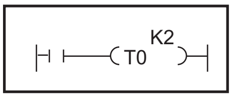
(d) When the set value is 0:

The contact turns on when the OUT T☐ or OUT ST☐ instruction is executed.

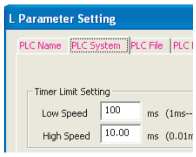
(e) Timer setting value and timer limit setting

Set the timer to meet the following condition:

Timer setting value ≧ Scan time + Timer limit setting



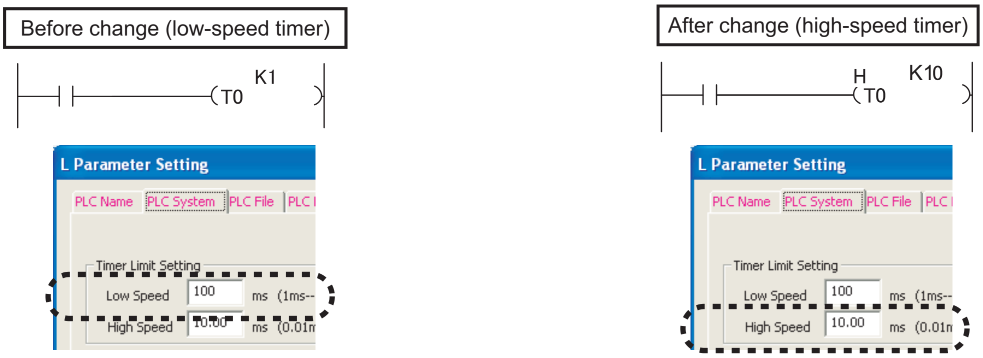
(SD526,SD527)



If the values are set to become "Timer setting value < Scan time + Timer Limit Setting", the coil and the contact might be simultaneously turned on depending on the timing on which the coil is turned on.

If the setting does not meet the above condition, make the value of the timer limit setting smaller to meet the condition.

Ex. Make the value of the timer limit setting smaller by changing from low speed timer to high speed timer. (Assume that the scan time is 20ms.)

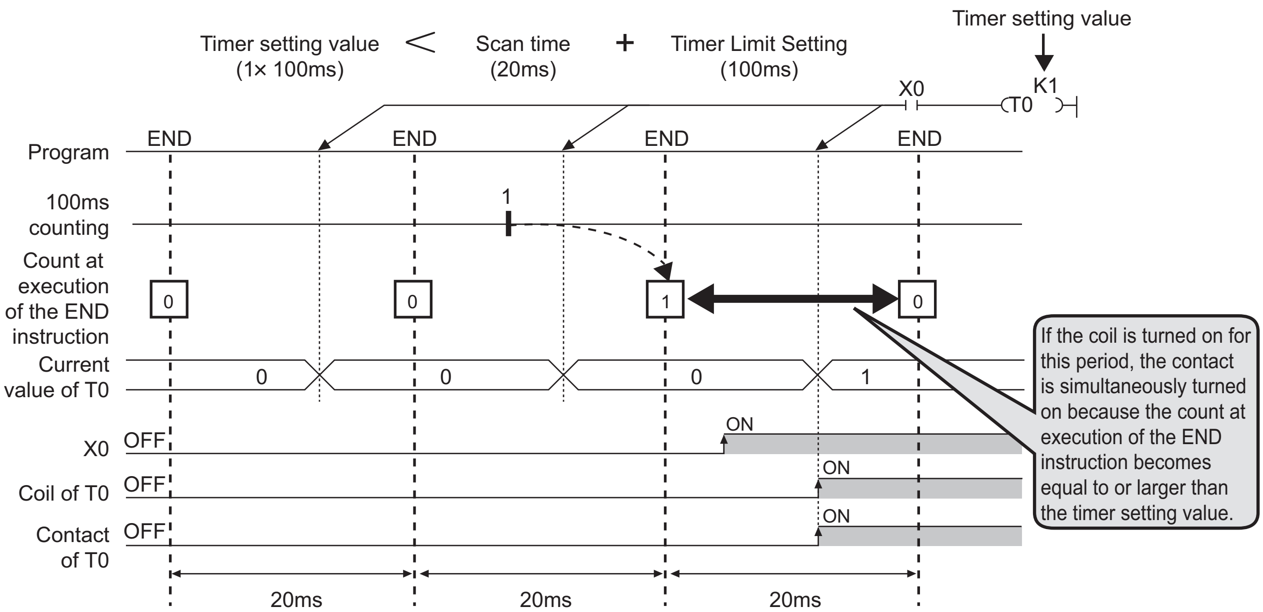


Timer setting value (100ms × 1=100ms) < Scan time (20ms) + Timer limit setting (100ms)

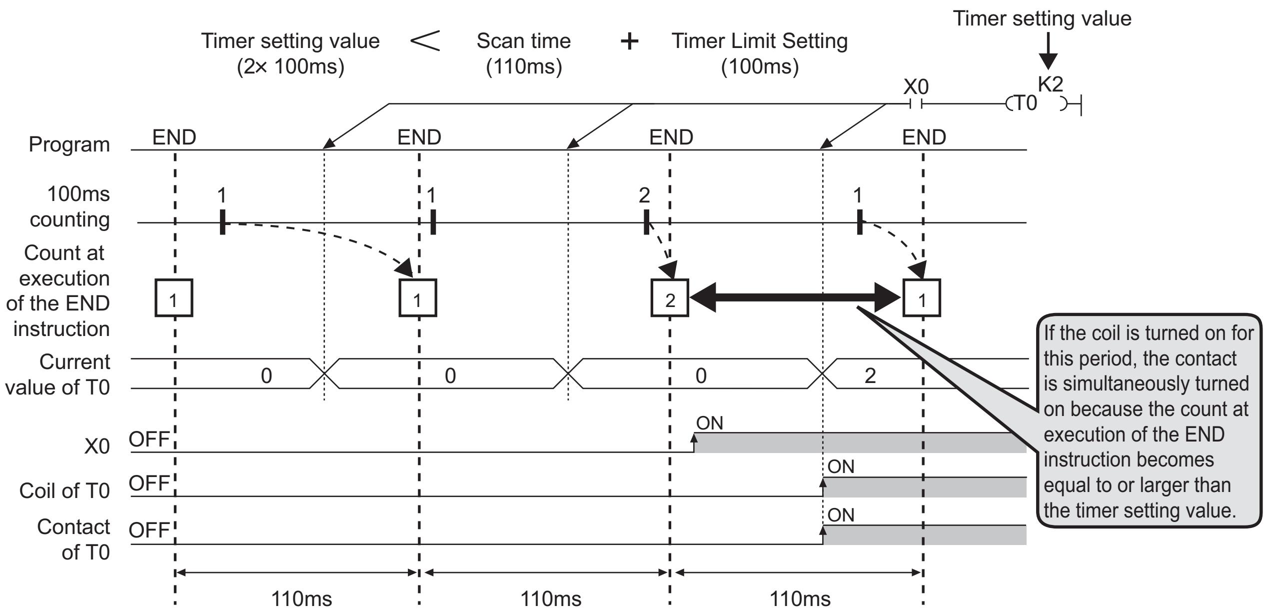
Timer setting value (10.00ms × 10=100ms) ≧ Scan time (20ms) + Timer limit setting (10ms)

The following show the examples of the coil and the contact being simultaneously turned on if the values are set to become "Timer setting value < Scan time + Timer Limit Setting":

Ex. When the timer setting value is 1 (1 × 100ms), the scan time is 20ms, and the timer limit setting is 100ms If the coil of the timer (T0) is turned on at the next scan after the values satisfy "Count at execution of the END instruction ≥ Timer setting value", the coil and the contact are simultaneously turned on because the values satisfy "Timer current value = Timer setting value" at the start of the timer.



Ex. When the timer setting value is 2 (2 × 100ms), the scan time is 110ms, and the timer limit setting is 100ms If the coil of the timer (T0) is turned on at the next scan after the values satisfy "Count at execution of the END instruction ≥ Timer setting value", the coil and the contact are simultaneously turned on because the values satisfy "Timer current value = Timer setting value" at the start of the timer.



(f) When the set value is changed after time-out

Even if the set value is changed to a larger value, the timer remains timed-out and does not start the operation.

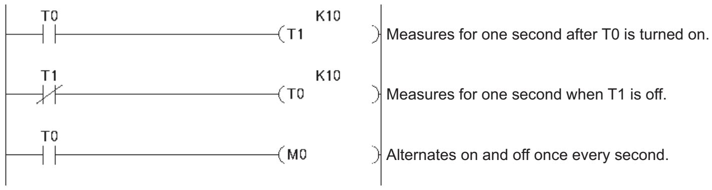
(g) When using multiple timers

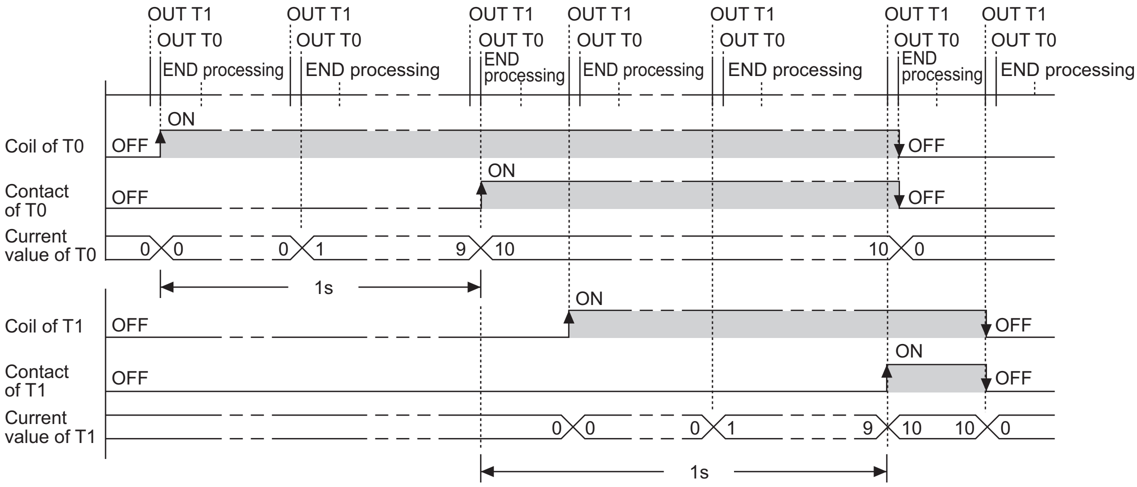
When using multiple timers, provide the timers from the last one to be measured because current value of the timers are updated upon execution of OUT T☐ or OUT ST☐ instruction.

Ex. Creating an on/off ladder using two timers

[Correct program example]

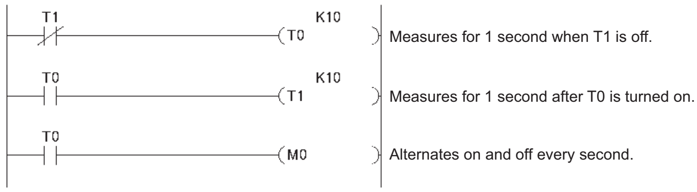
Coil of T1 is turned on for one scan after T0 is turned on.

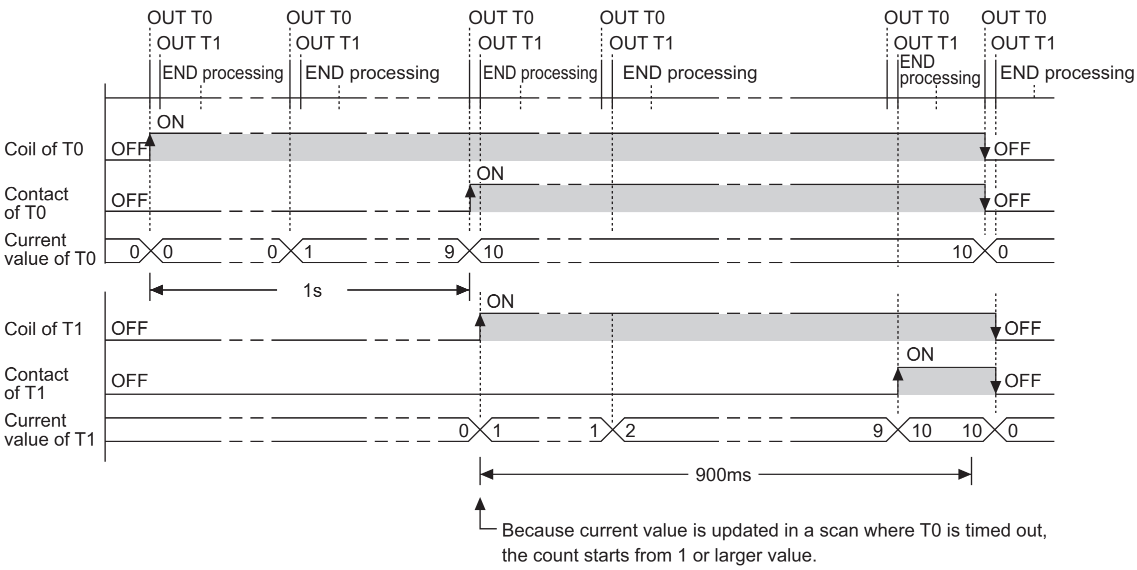




[Incorrect program example]

Coil of T1 is turned on for one scan after T0 is turned on.





5.2.11 Counter (C)

The counter (C) is a device that counts the number of rises for input conditions in programs. When the count value reaches the set value, its contact is turned on.

(1) Counter type

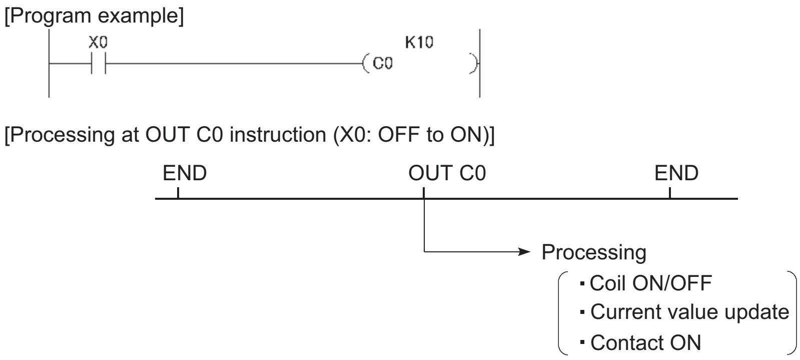
The following counter is available.

- Counter that counts the number of rises for input conditions in programs

(2) Counting

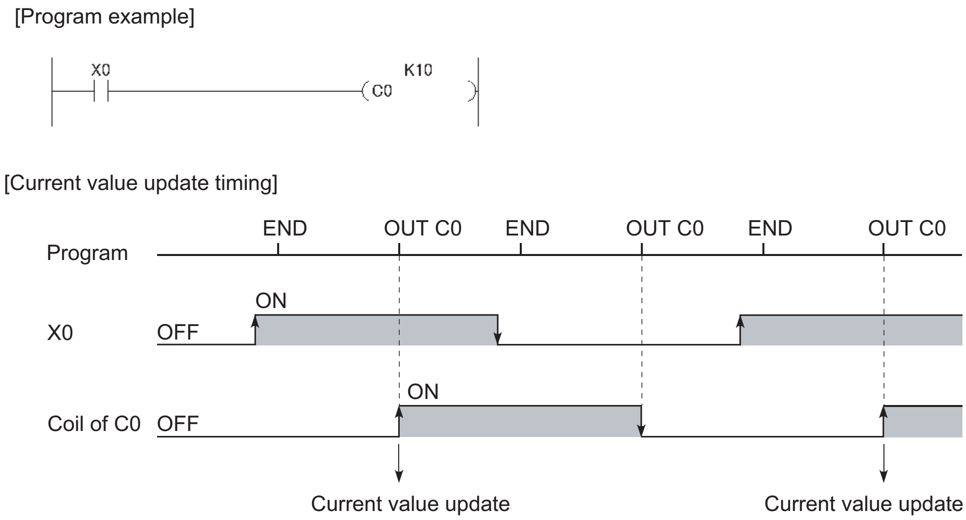
(a) When OUT C☐ instruction is executed

The coil of the counter is turned on/off, the current value is updated (the count value + 1), and the contact is turned on. In the END processing, the current timer value is not updated and the contact is not turned on.



(b) Current value update (count value + 1)

The current value is updated (count value + 1) at the leading edge (off → on) of the OUT C☐ instruction. The current value is not updated while the coil is off, or when it remains on or turns off from on by the OUT C☐ instruction.



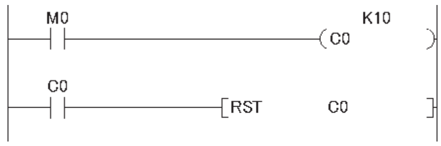
(3) Resetting the counter

To clear the current value and to turn off the contact of the counter, use the RST instruction. At the time of execution of the RST instruction, the counter value is cleared, and the contact is also turned off.

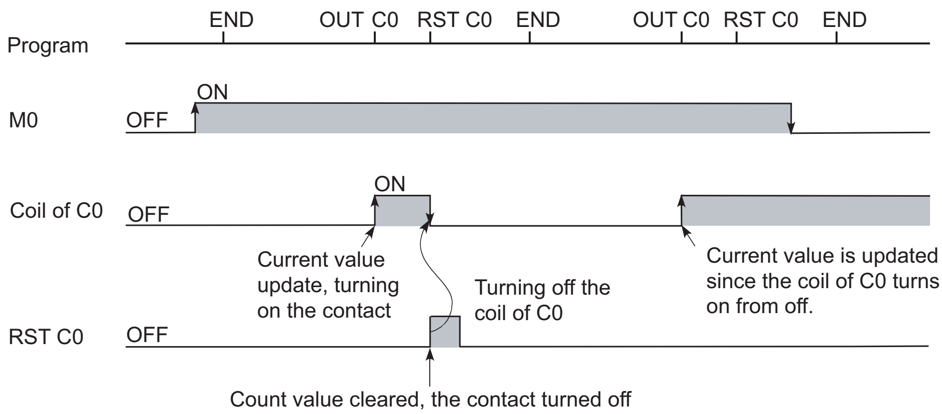
(a) Precautions for resetting the counter

Execution of the RST instruction also turns off the coil of counter. If the execution condition for the OUT instruction is still on after execution of the RST instruction, turn on the coil of counter at execution of the OUT instruction and update the current value (count value + 1).

[Program example]

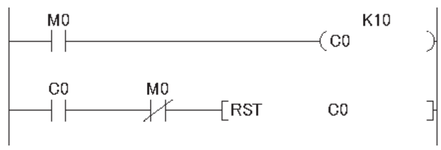


In the above program example, when M0 turns on from off, the coil of C0 turns on, updating the current value. When C0 reaches the preset value finally, the contact of C0 turns on, and execution of the RST instruction clears the execution value of C0. At this time, the coil of C0 also turns off. If M0 is still on in the next scan, the current value is updated since the coil of C0 turns on from off at execution of the OUT instruction. (The current value is changed to 1.)



To prevent the above, it is recommended to add a normally closed contact of the OUT instruction execution to the condition for the RST instruction execution so that the coil of C0 does not turn off while the execution condition (M0) of the OUT instruction is on.

[Modified ladder program]



(4) Maximum counting speed

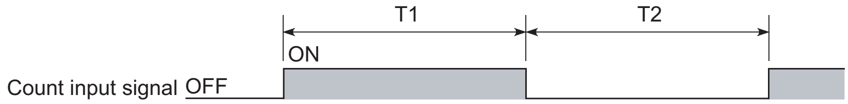
The counter can count only when the on/off time of the input condition is longer than the execution interval of the corresponding OUT instruction. The maximum counting speed is calculated by the following formula:

$$\text{Maximum counting speed (Cmax)} = \frac{n}{100} \times \frac{1}{T} \text{ [times/s]}$$

- n: Duty (%)*1
- T: Execution interval of the OUT instruction (sec)

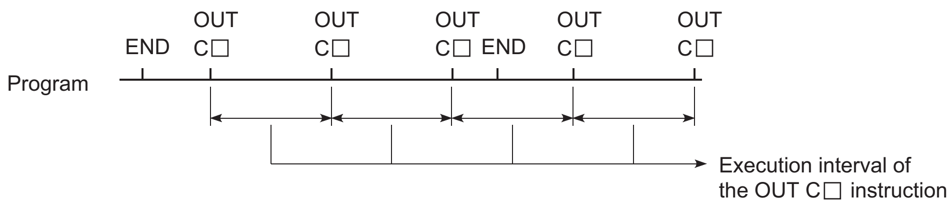
*1 Duty (n) is the ON-OFF time ratio of count input signal, and is expressed as a percentage value.

- When $T1 \geqq T2$, $n = \frac{T2}{T1 + T2} \times 100\%$
- When $T1 < T2$, $n = \frac{T1}{T1 + T2} \times 100\%$



Point

The maximum counting speed can be increased by placing multiple counters within one scan. At this time, use the direct access input (DX) for the counter input signal. (☞ Page 51, Section 2.6)



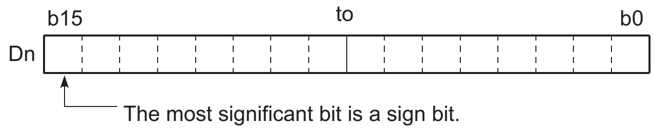
5.2.12 Data register (D)

The data register (D) is a memory in which numeric data (-32768 to 32767, or 0000_H to $FFFF_H$) can be stored.

(1) Bit structure of the file register

(a) Bit structure and read/write unit

One point of the data register consists of 16 bits, and data can be read or written in units of 16 bits.



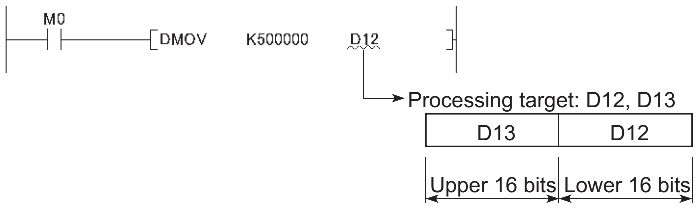
Point

Data register area handled as signed data. In the case of the hexadecimal notation, 0000_H to $FFFF_H$ can be stored.
However, because the most significant bit represents a sign bit, decimal values that can be specified are -32768 to 32767.

(b) Using in a 32-bit instruction

For a 32-bit instruction, two consecutive points of the data register (D_n and D_{n+1}) are the target of the processing. The lower 16 bits correspond to the data register number (D_n) specified in the program, and the higher 16 bits correspond to the specified data register number + 1.

Ex. When D12 is specified in the DMOV instruction, D12 represents the lower 16 bits and D13 represents the higher 16 bits.



Data of -2147483648 to 2147483647 or 00000000_H to $FFFFFFFF_H$ can be stored in a two-point area of the data register. (The most significant bit in a 32-bit structure is a sign bit.)

(2) Retention of stored data

The data stored in the data register are held until other different data are stored. Note that the stored data are initialized when the CPU module is powered off or reset.

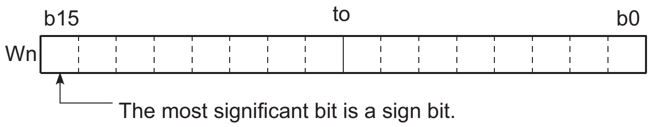
5.2.13 Link register (W)

The link register (W) is a data register with a device number represented in hexadecimal. The area where not used in network parameter can be used as a data register. In the link register, numeric data (-32768 to 32767, or 0000_H to $FFFF_H$) can be stored.

(1) Bit structure of the file register

(a) Bit structure and write/read unit

One point of the link register consists of 16 bits, and data can be written or read in units of 16 bits.



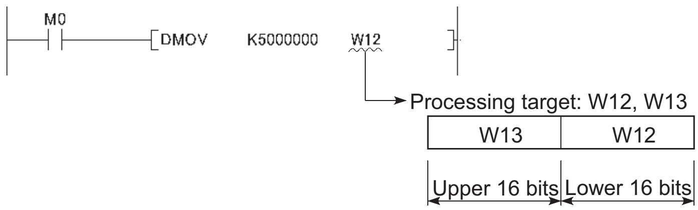
Point

Link register data are handled as signed data. In the case of the hexadecimal notation, 0000_H to $FFFF_H$ can be stored. However, because the most significant bit represents a sign bit, decimal values that can be specified are -32768 to 32767.

(b) Using in a 32-bit instruction

For a 32-bit instruction, two consecutive points of the data register (W_n and W_{n+1})are the target of the processing. The lower 16 bits correspond to the link register number (W_n) specified in the program, and the higher 16 bits correspond to the specified link register number + 1.

Ex. When W12 is specified in the DMOV instruction, W12 represents the lower 16 bits and D13 represents the higher 16 bits.



Data of -2147483648 to 2147483647 or 00000000_H to $FFFFFFFF_H$ can be stored in a two-point area of the link register. (The most significant bit in a 32-bit structure is a sign bit.)

(2) Retention of stored data

The data stored in the link register are held until other different data are stored. Note that the stored data are initialized when the CPU module is powered off or reset.

Point

When using a link device in the network module by using the points of the link register in the CPU module (default: 8192 points) or greater, change the points setting of the link register in the "Device" of the "PLC Parameter".

5.2.14 Link special register (SW)

The link special register (SW) is a register that stores communication status and error detection status of the CC-Link IE Field Network master/local module or the CC-Link system master/local module. Because the data link information is stored as numeric data, error locations and causes can be checked by monitoring the link special register.
Manuals for each network module

5.3 Internal System Devices

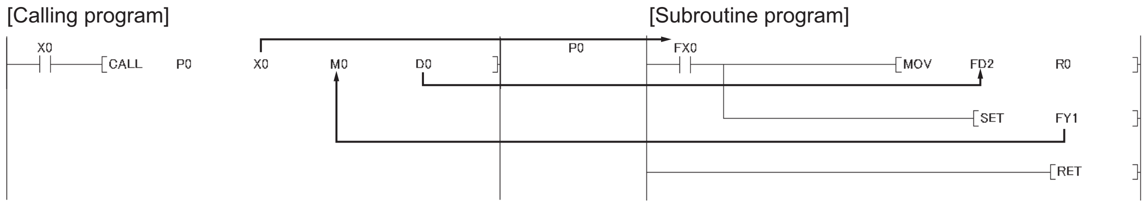
Internal system devices are provided for system operations. The allocations and sizes of internal system devices are fixed, and cannot be changed by the user.

5.3.1 Function devices (FX, FY, FD)

Function devices are used in subroutine programs with argument passing. Data are read or written between such subroutine programs and calling programs, using function devices.

Ex. When FX0, FY1, and FD2 are used in a subroutine program, and if X0, M0, and D0 are specified with a subroutine program call instruction, the following data passing are proceeded.

- On/Off data of X0 → FX0
- On/Off data of FY1 → M0
- Data of D0 → FD2



Because a device in each calling program can be determined by using a function device for subroutine programs, the same subroutine program can be used without considering other calling programs.

(1) Types of function devices

The following three types of function devices are available.

- Function input (FX)
- Function output (FY)
- Function register (FD)

(a) Function input (FX)

The function input is used to pass on/off data to a subroutine program. Bit data specified by a subroutine call instruction with argument passing are fetched into a subroutine program and they are used for operations. All bit devices for the CPU module can be used.

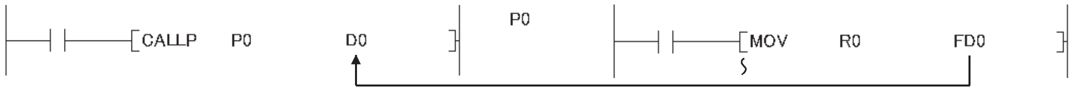
(b) Function output (FY)

The function output is used for passing an operation result (on/off data) in a subroutine program to a calling program. An operation result is stored in the device specified in the subroutine program with argument passing. All bit devices except for direct devices of the CPU module (DX) and annunciator (F) can be used.

(c) Function register (FD)

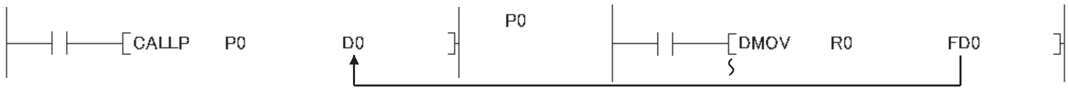
The function register is used for data writing or reading between a subroutine program and a calling program. The CPU module auto-detects the input or output conditions of the function register. Source data are input data of the subroutine program. Destination data are output data of the subroutine program. The function register of one point can occupy up to four words. Note that, however, the number of words used differs depending on the instruction in the subroutine program.

- A one-word instruction uses one word only.



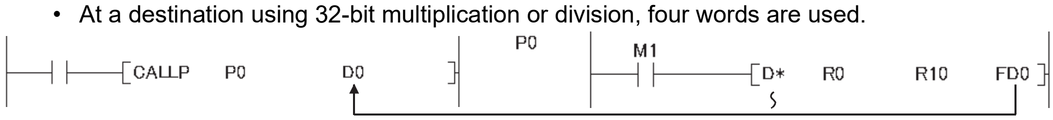
Data are stored in D0 (1 point).

- A two-word instruction uses two words.



Data are stored in D0 and D1 (2 points).

- At a destination using 32-bit multiplication or division, four words are used.

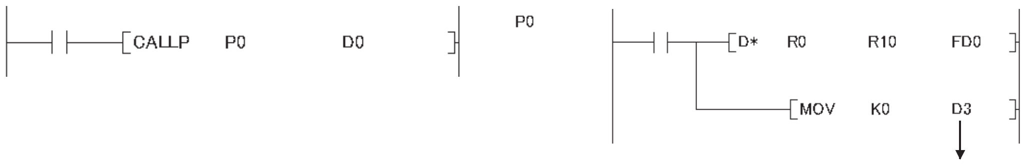


Data are stored in D0 to D3 (4 points).

- Word devices of the CPU module can be used.

Point

In subroutine programs with argument passing, do not use any devices that are used by the function register.
If this occurs, function register values will not be normally passed to the calling program.



Since D0 to D3 are used for FD0, D3 cannot be used in the subroutine program.

5.3.2 Special relay (SM)

The special relay (SM) is an internal relay of which details are specified inside the CPU module, and the CPU module status data are stored in this special relay.
📖MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

5.3.3 Special register (SD)

The special register (SD) is an internal register of which details are specified inside the CPU module, and the CPU module status data (such as failure diagnostics or system information) are stored in this special register.
📖MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

5.4 Link Direct Devices

The link direct device allows direct access to the link device of the CC-Link IE Field Network master/local module. Regardless of the link refresh of the CPU module, the direct reading/writing from/to the link device of the CC-Link IE Field Network master/local module can be done by using a program.

(1) Specification method

Specify by the following network number and device No.

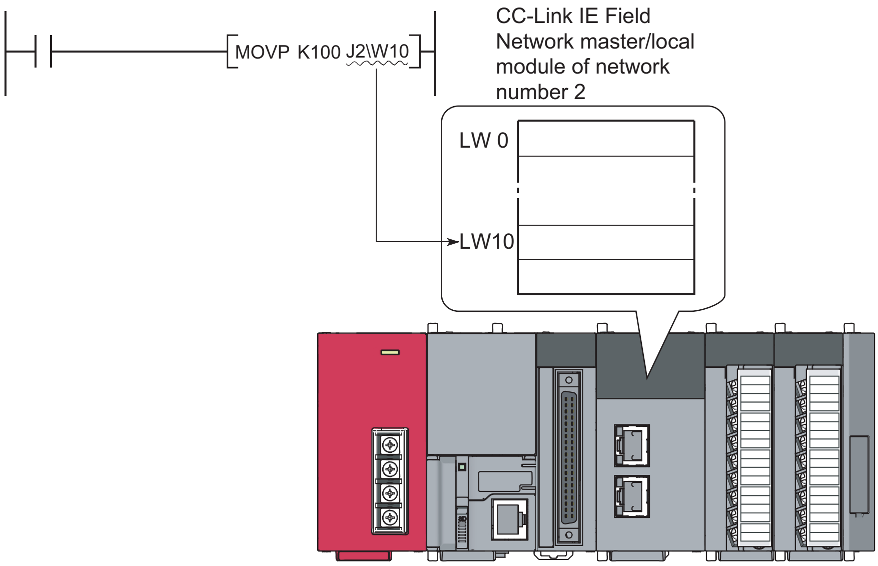
Specification method: J□\□

Device No.
- Link input (RX) ············ X0 to 3FFF
- Link output (RY) ··········· Y0 to 3FFF
- Link register (RWw) ········ W0 to 1FFF
- Link register (RWr) ········· W2000 to 3FFF
- Link special relay (SB) ····· SB0 to 1FF
- Link special register (SW) ·· SW0 to 1FF

Network number ···· 1 to 239

(2) Application example

"J2\W10" is for the network No.2 and the link register 10 (W10).



For the bit device (X, Y, B, SB), specify by the digit.

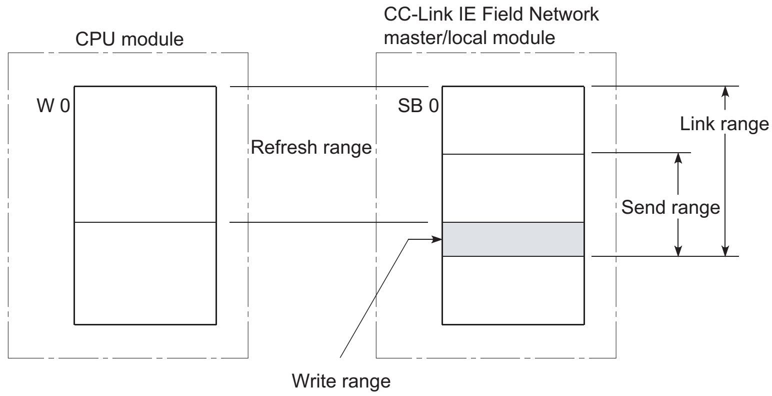
Ex. J1\K1X0, J10\K4B0

(3) Specification range

The link devices, which are out of the range specified by the refresh parameter, can be specified.

(a) Writing

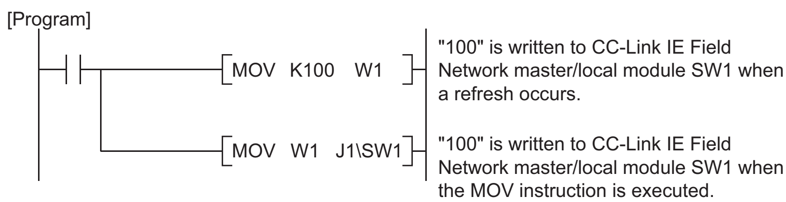
Write data within the link device range, which is set as the send range in the Common Parameter of the network parameter and out of the refresh range of the refresh parameter

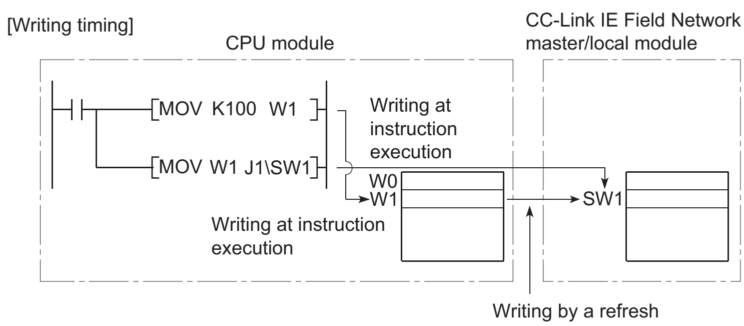


Although the writing within the link device range, which is set as the refresh range in the Refresh Parameter, can be done, the data of link device in the link module will be rewritten when a refresh occurs. When write to the link device by using the link direct device, write the same data to the corresponding device of the CPU module, which is set in the Refresh Parameter.

Ex. Writing to the link device which is set as the refresh range
[Settings of the refresh parameter]

- Network number: 1
- Device of the CPU module: W0 to 3F
- Device of the CC-Link IE Field Network master/local module: SW0 to 3F



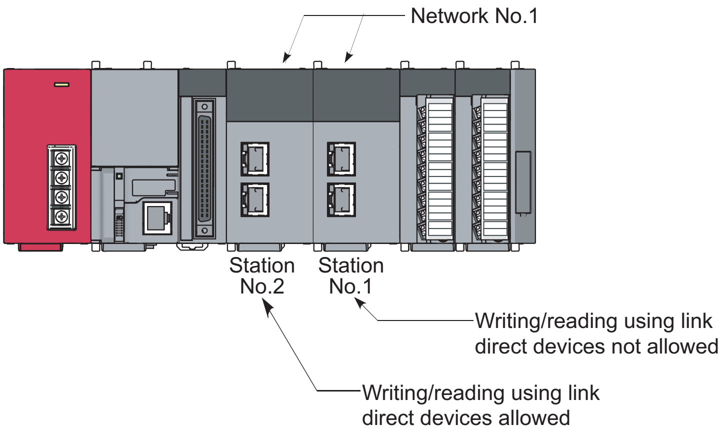


When the writing by the link direct device is done in the write range of the other station, the data will be replaced with the received data during the data reception from the other station.

(b) Reading

The reading of the link device range of the CC-Link IE Field Network master/local module can be done. The writing/reading by the link direct device can be done only for a single CC-Link IE Field Network master/local module per network number. For instance, when more than two CC-Link IE Field Network master/local modules are connected with the same network number, the writing/reading by the link direct device can be done only for the network module with the smaller slot number.

Ex. When the CC-Link IE Field Network master/local modules (Station No.1 and Station No.2) are connected as Network No.1 as shown below, the usage of the link direct device is allowed only for the network module of Station No.2.



(4) Difference between the link direct device and the link refresh

The following table shows the difference between the link direct device and the link refresh

Item		Link direct device	Link refresh
Notation in the program	Link register	J□\W0 to	W0 to
	Link special relay	J□\SB0 to	SB0 to
	Link special register	J□\SW0 to	SW0 to
Number of steps		2 steps	1 step
Access range of the CC-Link IE Field Network master/local module		J□\□0 to 3FFF	The range that is set in the Refresh Parameter
Guaranteed range of access data		2 words (32 bits) unit	

Remark

For the network parameters, common parameters, and refresh parameters, refer to the following.
MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual

5.5 Module Access Devices

5.5.1 Intelligent function module device

The intelligent function module device allows direct access from the CPU module to the buffer memories of the connected intelligent function modules.

(1) Specification method

Specify the I/O number and buffer memory address of the intelligent function module.

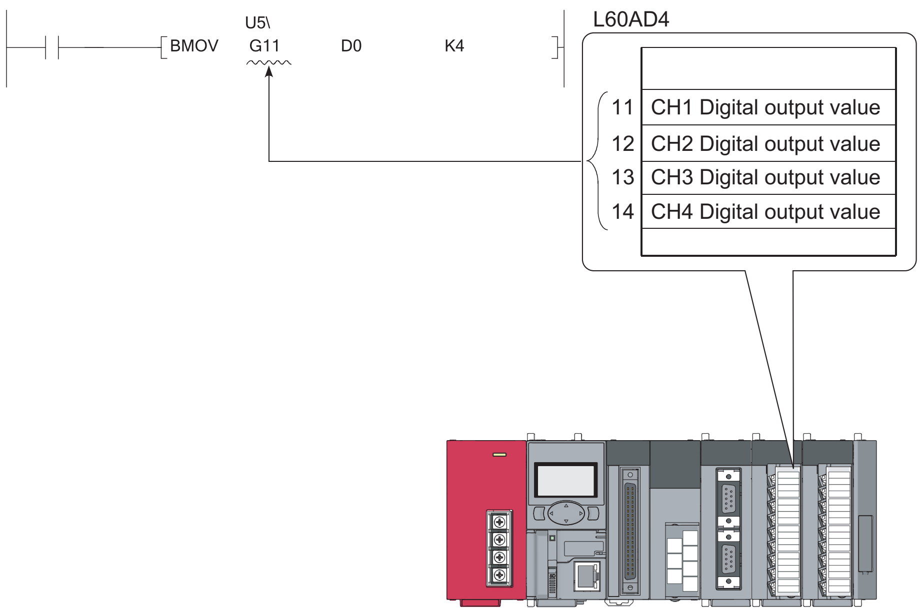
Specification method: U□ \G□

- G□ → Buffer memory address (setting range: 0 to 65535 in decimal)
- U□ → Start I/O number of intelligent function module
  - Setting: First 2 digits of start I/O number expressed with 3 digits

    X/Y1F0 ･･･ X/Y1F0 — Specification: 1F
  - Setting range: 00_H to FF_H

(2) Application example

When the L60AD4 high-speed analog-digital converter module is connected in the position of the start I/O number X/Y050, specify the device as shown below to store digital output values of CH.1 to CH.4 into D0 to D3 accordingly.



Point

If the intelligent function module device is used, device comments can be attached to the buffer memory.
Operating manual for the programming tool used

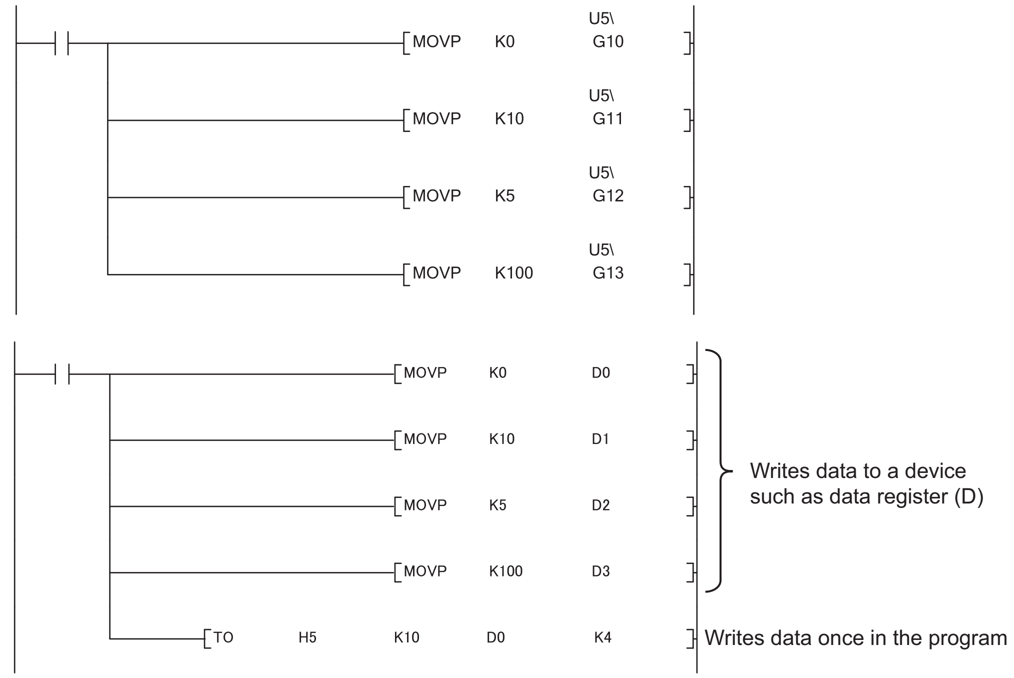
(3) Processing speed

The processing speed of the intelligent function module device is as follows:

- The processing speed of writing or reading using the intelligent function module device is slightly higher compared with the case of using the FROM or TO instruction. (Example: "MOV U5\G11 D0")
- When reading from the buffer memory of an intelligent function module and another processing with one instruction, totalize the processing speed of the FROM or TO instruction and the other instruction. (Example: "+ U5\G11 D0 D10")

Point

Instead of using the intelligent function module device in the program more than once to write or read buffer memory data, use the FROM or TO instruction once in one place so that the processing speed increases.



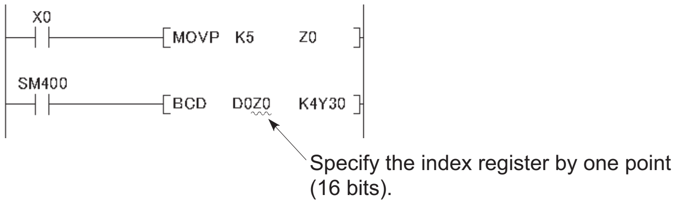
For the buffer memory address and application, refer to the manual for the intelligent function module used. For the FROM or TO instruction, refer to the following.

MELSEC-Q/L Programming Manual (Common Instruction)

5.6 Index Register/Standard Device Register (Z)

5.6.1 Index Register (Z)

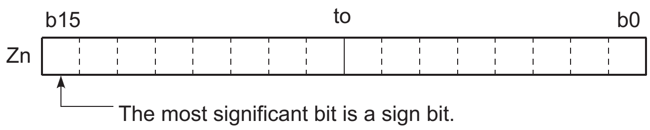
The index register is used for indirect specification (index modification) in programs. Index modification uses one point of the index register.



The index register has 20 points (Z0 to Z19).

(1) Bit structure of the index register

One point of the index register consists of 16 bits, and data can be read or written in units of 16 bits.



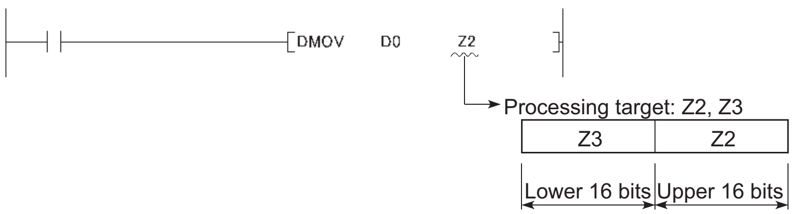
Point

Link register data are handled as signed data. In the case of the hexadecimal notation, 0000_H to $FFFF_H$ can be stored. However, because the most significant bit represents a sign bit, decimal values that can be specified are -32768 to 32767. (When using T, TS, or C device, specify the values within the range of -16384 to 16383.)

(2) Using in a 32-bit instruction

The processing target is Z_n and Z_{n+1}. The lower 16 bits correspond to the specified index register number (Z_n), and the higher 16 bits correspond to the specified index register number + 1.

Ex. When Z2 is specified in the DMOV instruction, Z2 represents the lower 16 bits and Z3 represents the higher 16 bits. (The most significant bit in a 32-bit structure is a sign bit.)



(3) When using 32-bit index modification

For the file register (ZR), extended data register (D), extended link register (W) using the serial number access method, 32-bit index modification using two points of the index register is available.
The following two kinds of methods can be used to specify the index register.

- Specify the range used for 32-bit index modification.
- Specify the 32-bit index modification using "ZZ".

For details on specification of the index register, refer to the following.
📖MELSEC-Q/L Programming Manual (Common Instruction)

5.6.2 Standard device register (Z)

By using the index register between register operations, operations can be executed at a higher speed. The index register used in this case is called the standard device resister.
📖MELSEC-Q/L Programming Manual (Common Instruction)

(1) Device number

Since the standard device register is the same device as the index register, pay attention not to use the same device number when using the index modification.

5.6.3 Saving and restoration of the index register

The CPU module performs the following when switching from the scan execution type program to the interrupt/fixed scan execution type program.

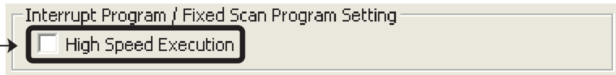
- Saving and restoring the index register data
- Saving and restoring block numbers of the file register

(1) Setting for saving and restoration

Configure the setting for saving and restoration using a programming tool.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]

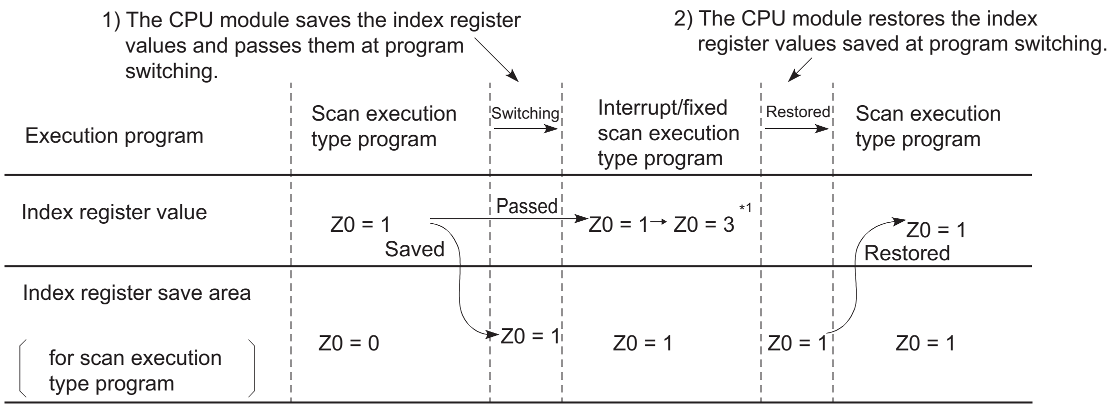
Selecting this saves or restores index register data.



When not writing data to the index register during interrupt/fixed scan execution type program, select "High Speed Execution" in the Interrupt Program/Fixed Scan Program Setting area. This setting enables faster program switching because saving and restoration are not performed.

(2) Processing of the index register

(a) When "High speed execution" is not selected

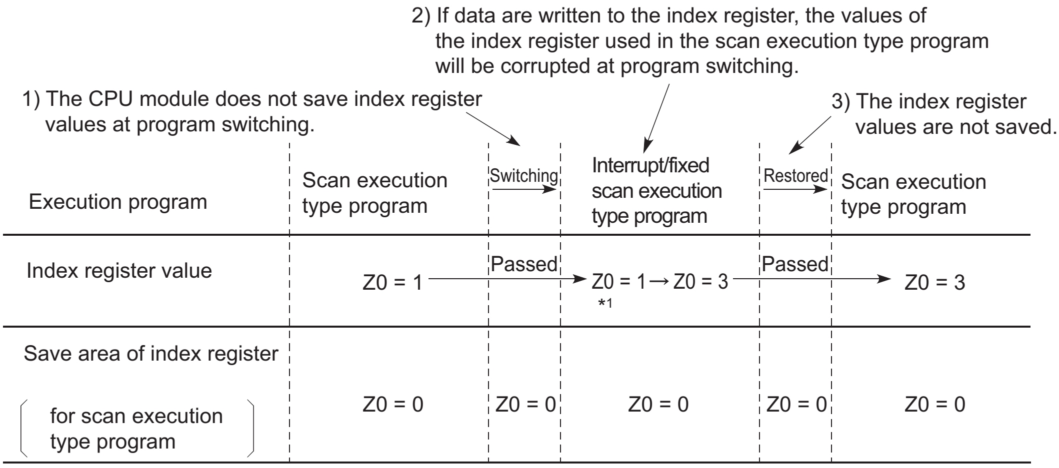


*1 The Z0 value is changed to 3 in the interrupt program.

Point

To pass index register values from the interrupt/fixed scan execution type program to the scan execution type program, use word devices.

(b) When "High Speed Execution" is selected

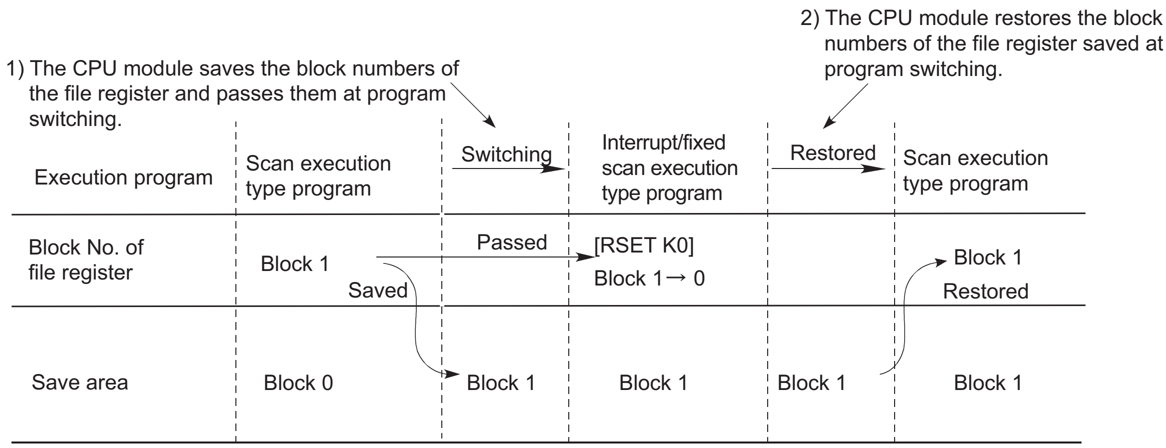


*1 The Z0 value is changed to 3 in the interrupt program.

Point

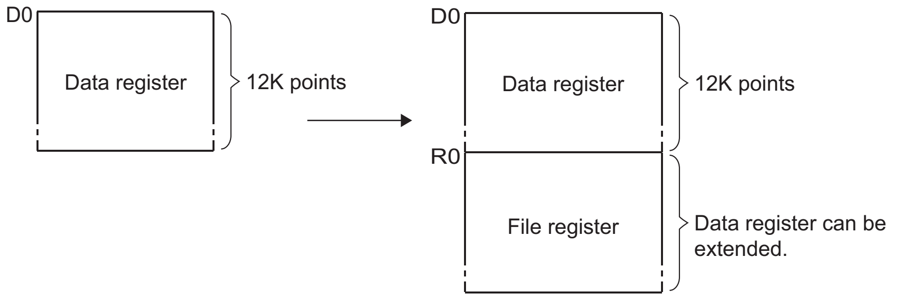
When writing data to the index register, use the ZPUSH or ZPOP instruction to save and restore the data. (MELSEC-Q/L Programming Manual (Common Instruction))

(3) Processing of file register block numbers



5.7 File Register (R, ZR)

These device are provided for extending the data register.



(1) Specification methods

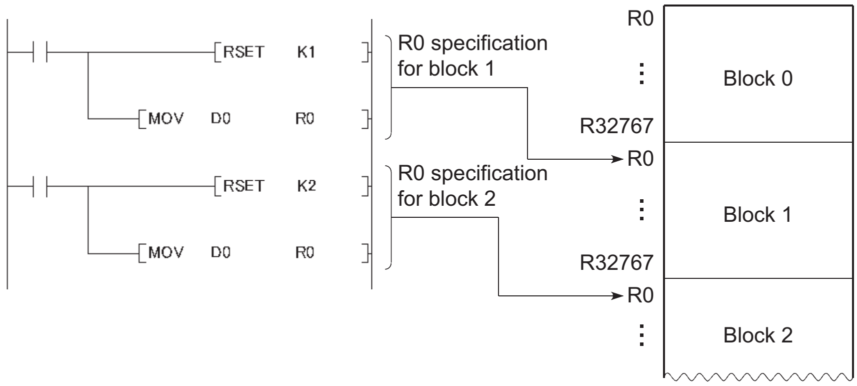
The following two methods are available.

- Block switching method (R)
- Serial number access method (ZR)

(a) Block switching method (R)

The file register points used are divided and specified in units of 32K points (R0 to R32767).

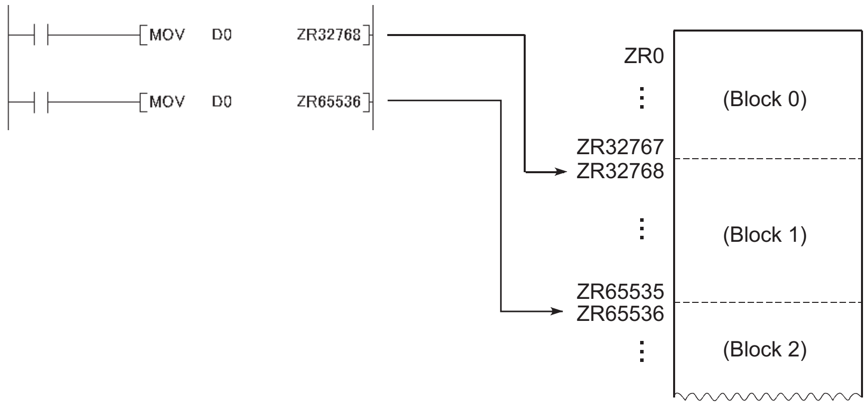
When multiple blocks are used, the desired block is specified with the block number by the RSET instruction.



(b) Serial number access method (ZR)

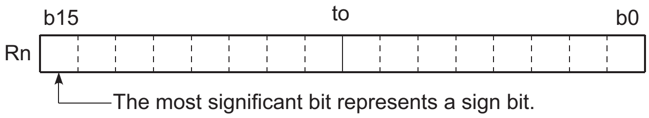
A file register exceeding 32K points in size can be specified using consecutive device numbers.

Multiple blocks of a file register can be used as a continuous file register.



(2) Bit structure of the file register

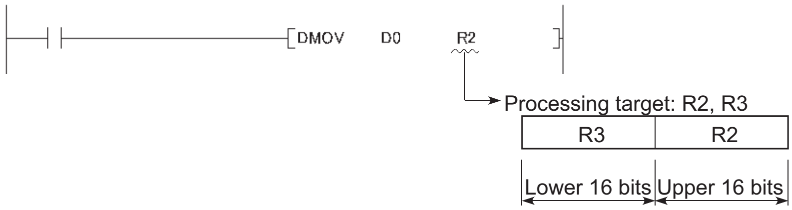
One point of the file register consists of 16 bits, and data can be read or written in units of 16 bits.



(3) When using a 32-bit instruction

The processing target is R_n and R_{n+1}. The lower 16 bits corresponds to the file register number (R_n) specified in the program, and the higher 16 bits corresponds to the specified file register number + 1.

Ex. When R2 is specified in the DMOV instruction, R2 represents the lower 16 bits and R3 represents the higher 16 bits.



Data of -2147483648 to 2147483647 or 00000000_H to $FFFFFFFF_H$ can be stored in a two-point area of the file register. (The most significant bit in a 32-bit structure is a sign bit.)

(4) File register data registration

The standard RAM is available as a memory media for storing file register data.

(5) File register size

The standard RAM can store the data up to the following size. Note that, however, if the standard RAM is used other than as the file register, available points are decreased. (☞ Page 31, Section 2.1)

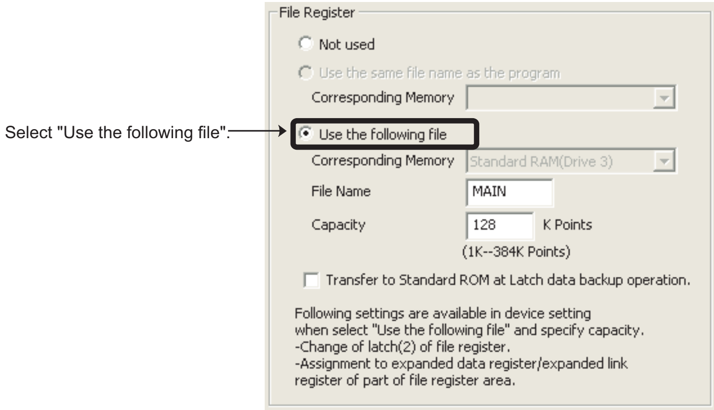
CPU module	Points
L02SCPU, L02SCPU-P, L02CPU, L02CPU-P	64K
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	384K

(6) Setting method

(a) File register setting

When using a file register, select "Use the following file". (Set in the same way when using an extended data register (D) or an extended link register (W).)

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC File]



CPU module	Corresponding memory	File name	Capacity*1
L02SCPU, L02SCPU-P, L02CPU, L02CPU-P	Standard RAM	Any name	1 to 64K points
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT			1 to 384K points

*1 The total points of the file register (ZR), extended data register (D), and extended link register (W).

Point

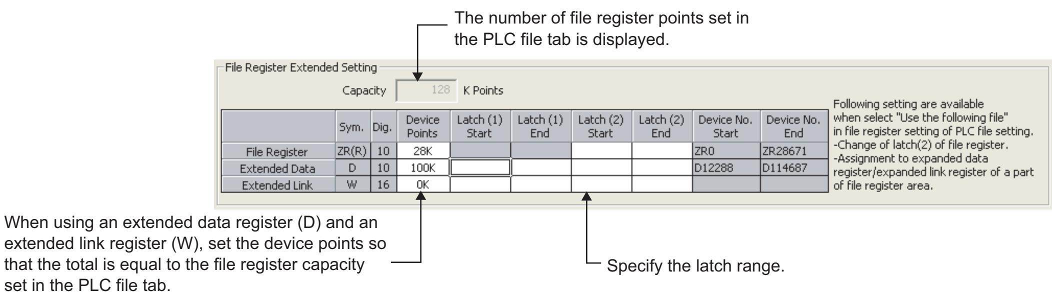
When using the file register, it is shared by all the programs to be executed. A file register file is created after the parameters specified in "Corresponding Memory" and "File Name" are written to the CPU module. When the capacity is not specified, note the following.

- When the specified file register file is stored in the specified drive, the file is used. (The capacity is the same as that of the stored file register file.)
- If the file register file with the specified file name is not found in the specified drive, "PARAMETER ERROR" (error code: 3002) occurs.

(b) Device setting

The points of the file register (ZR) can be set in the File Register Extended Setting.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Device]



1. The latch range can be changed if necessary.

- Latch (2) of the file register (ZR)

(7) Clearing the file register

If the Latch (2) is set in the Device tab of the PLC parameter dialog box, the data in the file register are not cleared even if the CPU module is powered off or reset. (Data cannot be cleared by performing a latch clear operation.[*1])

For how to clear the data, refer to the "Latch data clear" section. (☞ Page 89, Section 3.4 (4))

*1 The latch range of the file register can be set in the Device tab of the PLC Parameter dialog box. (☞ Page 328, Section 5.7 (6) (b))

(8) Deleting a file register file

Open the "Online Data Operation" dialog box.

[Online] ⇨ [Delete PLC Data...]

(9) Precautions

(a) If the file register number not registered is used

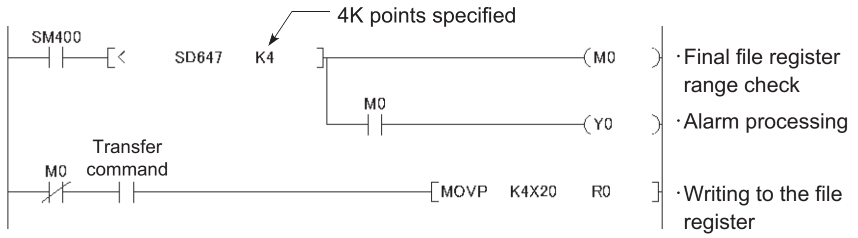
If a file register file is not registered and writing to or reading from the file register is performed, "OPERATION ERROR" (error code: 4101) occurs.

(b) If writing to or reading from the file register is performed exceeding the registered size of points

"OPERATION ERROR" (error code: 4101) occurs.

(c) File register size check

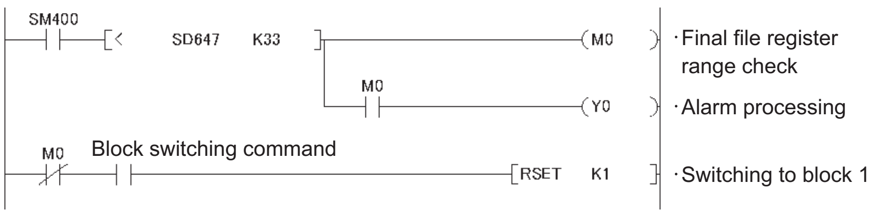
When writing to or reading from the file register, check the file register size so that data can be written or read within the size (points) set for the CPU module. Check the file register size at step 0 in the program if any file register is used.



The file register size can be checked in SD647.

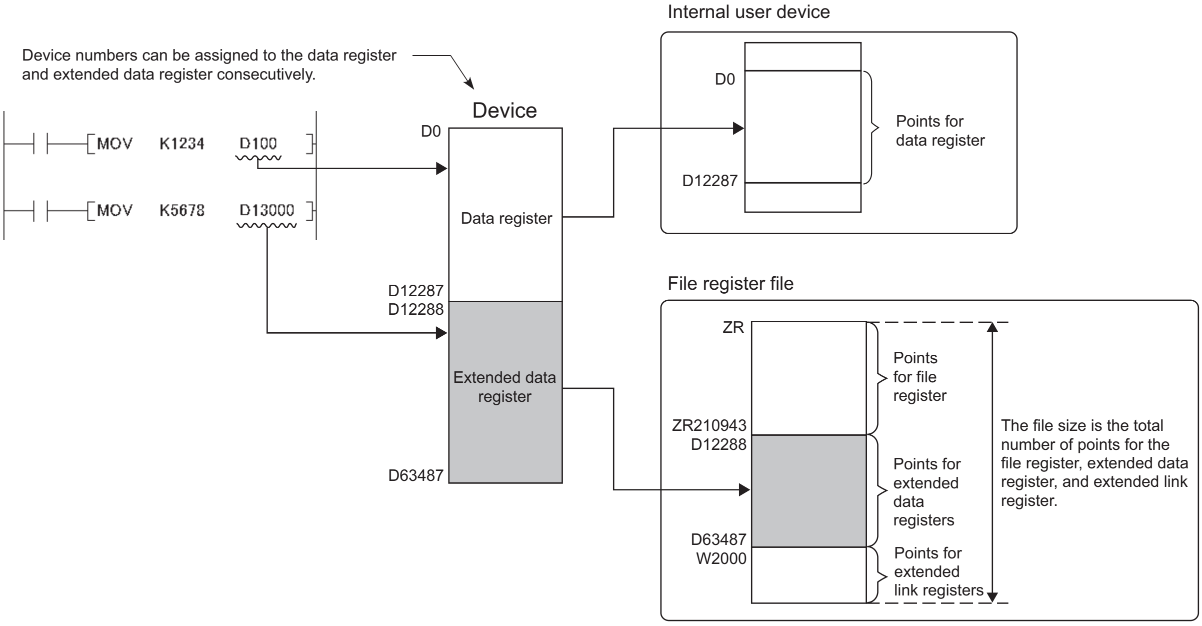
Before executing a file register block switching instruction (RSET), ensure that 1K points or more of space are secured after the switching.

(File register size) > [32K points × (Switching block No.) + 1K points]



5.8 Extended Data Register (D) and Extended Link Register (W)

The extended data register (D) and extended link register (W) are devices for utilizing the large-capacity file register (ZR) area as an extended area of the data register (D) and link register (W). These devices can be programmed as the data register (D) and link register (W) together with the file register (ZR) area.



Device numbers for the extended data register (D) and extended link register (W) can be consecutively assigned after those for the internal user devices, data register (D) and link register (W). Note that even though device numbers are consecutively assigned, there is no physical area contiguity between the data register (D) (internal user device) and the extended data register (D), and between the link register (W) (internal user device) and the extended link register (W). To use them as one contiguous area, set the points for the data register (D) and link register (W) (internal user device) to "0" in the Device tab of the PLC Parameter dialog box, and use only the extended data register (D) and extended link register (W).

Point

When using the file register (ZR), extended data register (D) and extended link register (W) with the auto refresh setting for the intelligent function module, the points set in the File register extended setting in the Device tab of the PLC Parameter dialog box must not be exceeded.

(1) Setting method

Since the extended data register (D) and extended link register (W) use the file register area, data must be set for both the file register setting and the device setting.

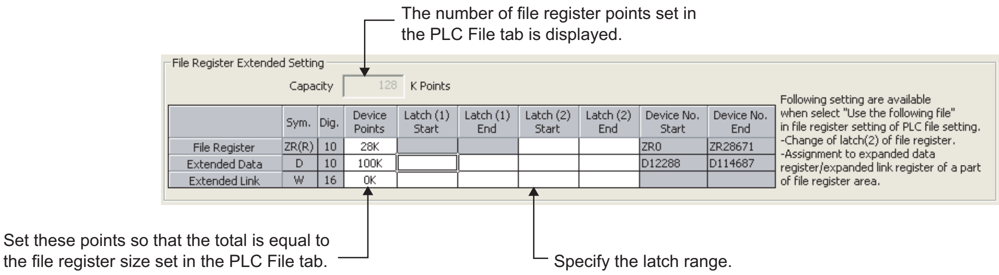
(a) File register setting

The setting method is the same as when using a file register. (☞ Page 328, Section 5.7 (6) (a))

(b) Device setting

Set points for the extended data register (D) and the extended link register (W) in the File Register Extended Setting.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Device]

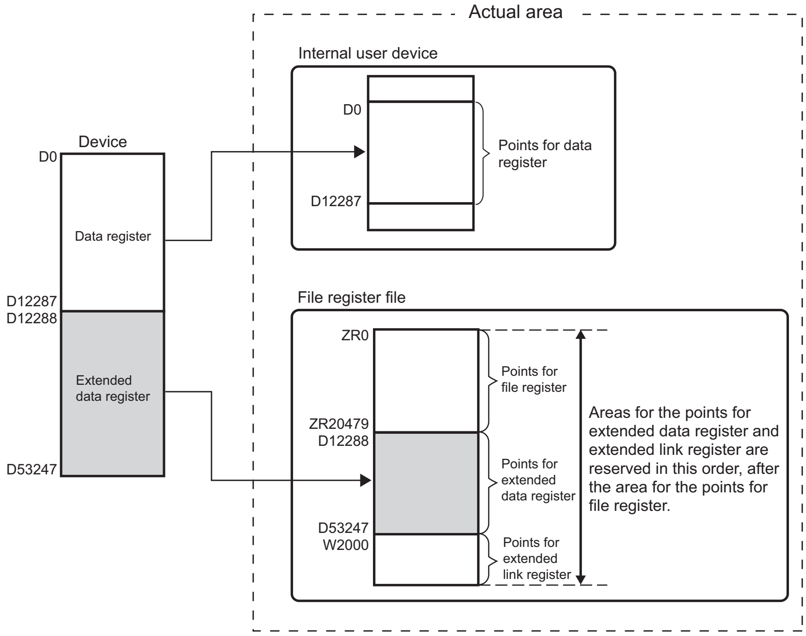


1. **Assign a part of the points set for the file register (ZR) in the PLC File tab to the extended data register (D) and extended link register (W).**

2. **The latch range can be changed if necessary.**

- Latch (1) and (2) of the extended data register (D)
- Latch (1) and (2) of the extended link register (W)

Once the points for the extended data register (D) and extended link register (W) is set, areas for these devices are reserved in the file register file.



The points for each of the file register (ZR), extended data register (D), and extended link register (W) can be checked in the following SD.

SD number	Corresponding points
SD306, SD307	File register (ZR)
SD308, SD309	Data register (D) + Extended data register (D)
SD310, SD311	Link register (W) + Extended link register

(2) Precautions

(a) Specifying the extended data register (D) and extended link register (W)

Since the file register (ZR) area is used, the values of the following items will be the same as those for the file register (ZR).

- Number of program steps
- Instruction processing time
- Link refresh time with the CC-Link IE Field Network master/local module
- Link refresh time with CC-Link IE Field Network Basic
- Auto refresh time with CC-Link
- Processing time of auto refresh with intelligent function modules

(b) Changing the file register size

The file register size cannot be changed while the CPU module is in the RUN status.

(c) Refresh range

Set the refresh ranges for the following auto refresh properly so that each refresh range does not cross over the boundary between the internal user device and the extended data register (D) or extended link register (W).

- Link refresh with CC-Link IE Field Network
- Link refresh with CC-Link IE Field Network Basic
- Auto refresh with CC-Link
- Refresh with intelligent function modules

(d) Specifying in a program

Set the following properly so that each specification does not cross over the boundary between the internal user device and the extended data register (D) or extended link register (W).

- Index modification
- Indirect specification
- Specification for instructions that use block data[*1]

*1 Block data means the following:
· Data used in instructions, such as FMOV, BMOV, and BK+, which treat more than one word for operation
· Control data, composed of two or more words, which are specified in instructions, such as SP.FWRITE and SP.FREAD.
· Data in a 32-bit or greater format (binary 32 bits, real number, indirect address of a device)

For details on the index modification and indirect specification with the extended data register (D) and extended link register (W), refer to the following.
MELSEC-Q/L Programming Manual (Common Instruction)

(e) Access from an inapplicable module

To access the extended data register (D) or extended link register (W) from a module that does not support the use of these devices, device numbers need to be specified with those of the file register (ZR). Calculation formulas for obtaining device numbers of the file register (ZR) to be specified and calculation examples are described below

Item	Calculation method
Device number of the file register (ZR) used to access the extended data register (D)	$ED_{ZN} = ZR_C + (ED_N - D_C)$
Device number of the file register (ZR) used to access the extended link register (W)	$EW_{ZN} = ZR_C + ED_C + (EW_N - W_C)$

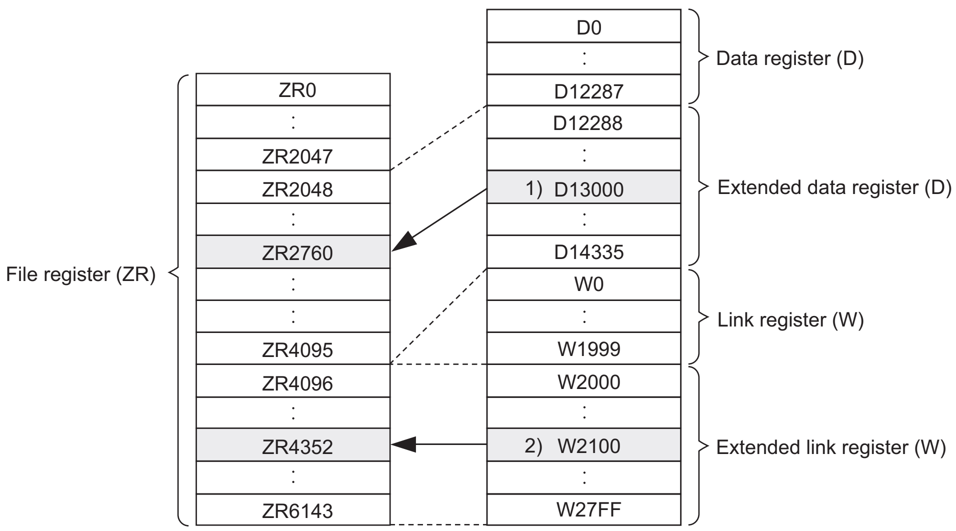
*1 Variables in the table indicate the following:
· ZR_C: Points of the file register (ZR)
· ED_{ZN}: Device number of the file register (ZR) used to access the extended data register (D)
· ED_N: Access target device number of the extended data register (D)
· ED_C: Points of the data register (D)
· ED_C: Points of the extended data register (D)
· EW_{ZN}: Device number of the file register (ZR) used to access the extended link register (W)
· EW_N: Access target device number of the extended link register (W) (hexadecimal)
· EW_C: Points of the link register (W)

[Calculation example]

- D_C: Points of the data register (D) ··· 12288 points
- W_C: Points of the link register (W) ··· 8192 points
- ZR_C: Points of the file register (ZR) ··· 2048 points
- ED_C: Points of the extended data register (D) ··· 2048 points

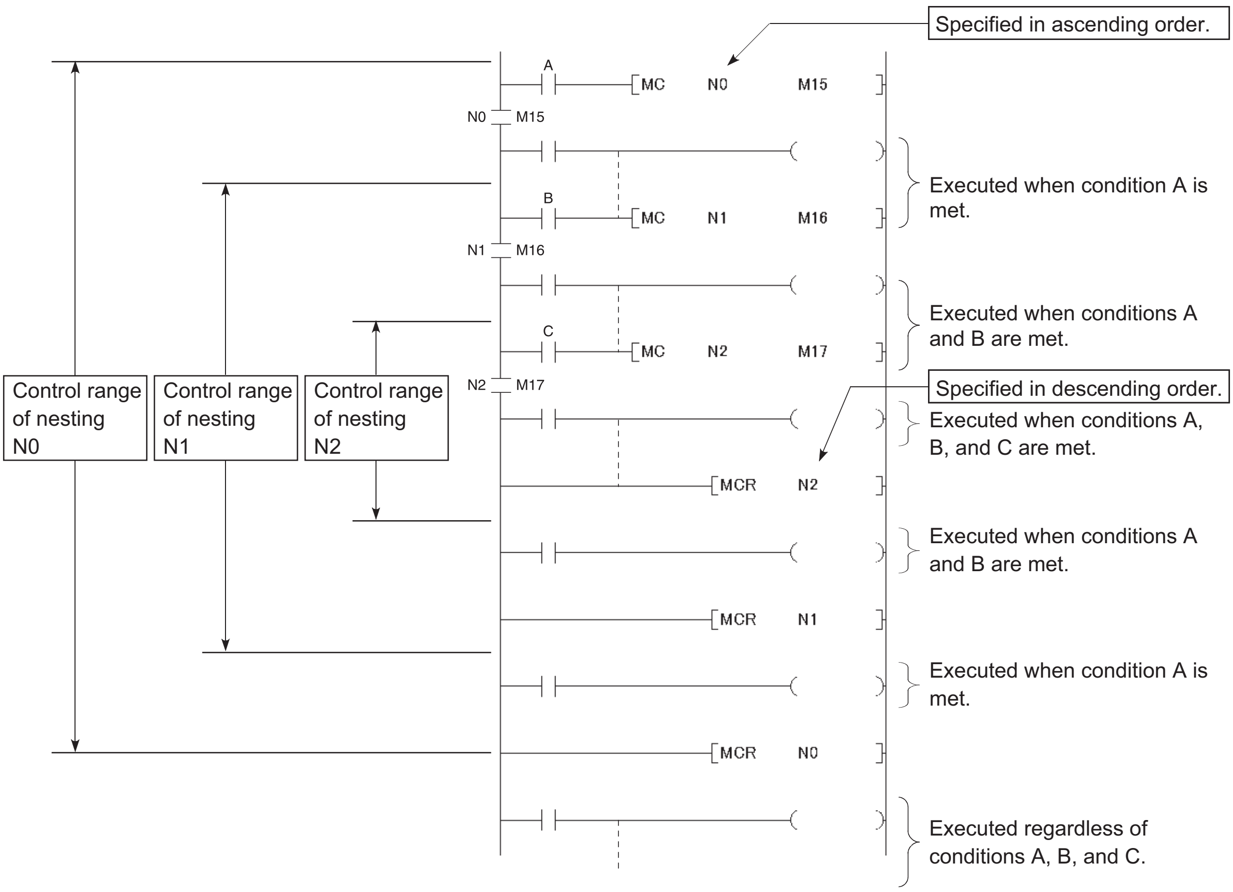
Ex. Device number of the file register (ZR) used to access D13000
ED_{ZN} = 2048 + (13000 - 12288) = 2760

Ex. Device number of the file register (ZR) used to access W2100
EW_{ZN} = 2048 + 2048 + (2100_H - 8192) = 2048 + 2048 + (8448 - 8192) = 4352



5.9 Nesting (N)

Nesting (N) is a device used in the master control instructions (MC and MCR instructions) to program operation conditions in a nesting structure. The master control instruction opens or closes a common ladder gate to efficiently switch the ladder of a program. Specify the nesting (N) in ascending order (in order of N0 to N14), starting from the outside of the nesting structure.



For use of the nesting, refer to the following.

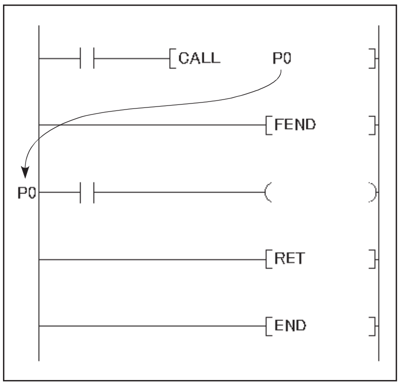
MELSEC-Q/L Programming Manual (Common Instruction)

5.10 Pointer (P)

The pointer (P) is a device used in jump instructions (CJ, SCJ, or JMP) or subroutine call instructions (such as CALL).

Pointers can be used in the following applications.

- Specification of the jump destination in a jump instruction (CJ, SCJ, or JMP) and a label (start address of the jump destination)
- Specification of the call destination of a subroutine call instruction (CALL or CALLP) and a label (start address of the subroutine program)



There are the following two different pointer types.

- Local pointer:

The pointer used independently in each program

- Common pointer:

The pointer that can be called in all running programs by the subroutine call instruction.

The number of points available for the pointer is 4096.

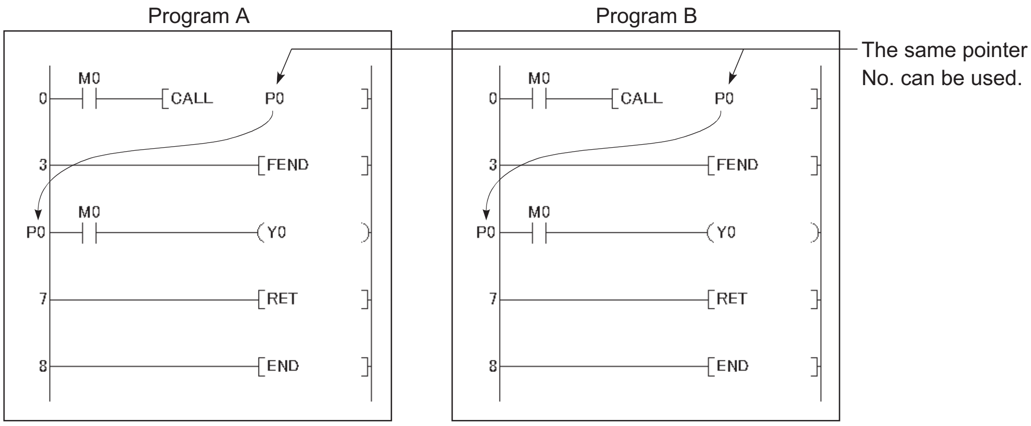
Point

For the L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT with a serial number of "16042" or later, the pointer for auto assignment device can be used up to 32768 points by using GX Works2. (The PC parameters should be set. (☞ Page 362, Appendix 1.2 (8)))

For the jump instructions and subroutine call instructions, refer to the following.
📖MELSEC-Q/L Programming Manual (Common Instruction)

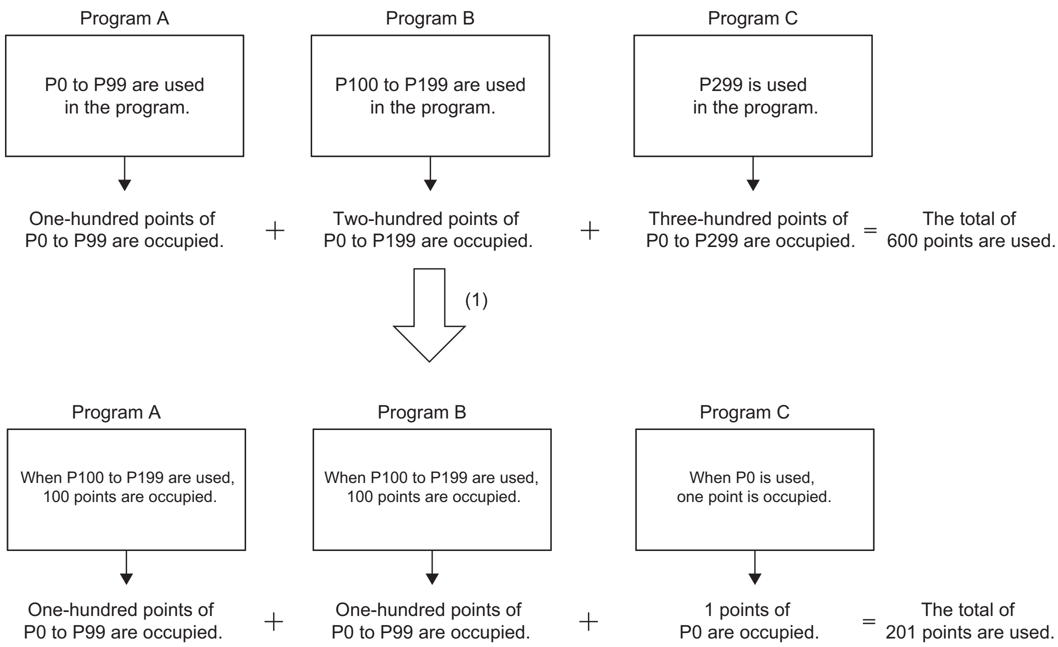
5.10.1 Local pointer

The local pointer is a pointer that can be used independently in jump instructions and subroutine call instructions in each program. The same pointer number can be used in respective programs.



Local pointer number of points are shared among all the programs. The range of the local pointer number of points used by each program is from #P0 to the maximum value of the local pointer being used in that program. For example, even when a program actually uses only #P99, 100 points (#P0 to #P99) are considered to be used. When using local pointers in multiple programs, they can be effectively used by using them in ascending order from #P0 in each program group.

Ex. The total is 600 points when the pointer is used as shown below.



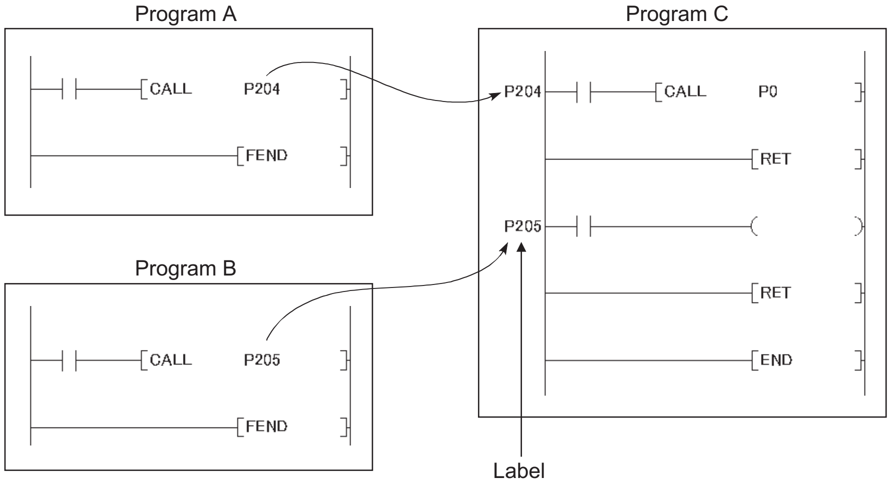
(1) By using pointers in ascending order from #P0 in each program group, a total of 201 points will be needed instead of a total of 600 points.

Point

- When the program where the local pointer is described, the program cannot jump in from another program is not allowed. Use the ECALL instruction from another program when calling a subroutine program in a program file that contains any local pointer.
- If the total number of pointers (in all programs) exceeds 4096 points, a "CAN'T SET (P)" (error code: 4020) occurs.

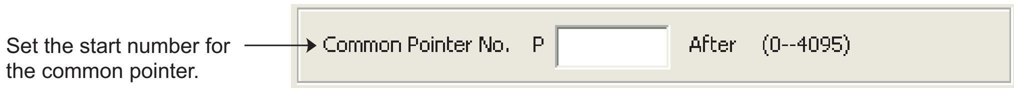
5.10.2 Common pointer

The common pointer is used to call subroutine programs from all programs that are being executed.



To set the common pointer range, enter the start number of the common pointer. The common pointer range is from the specified pointer number to P4095. However, the pointer number that can be entered here is a number greater than the total points used for the local pointer.

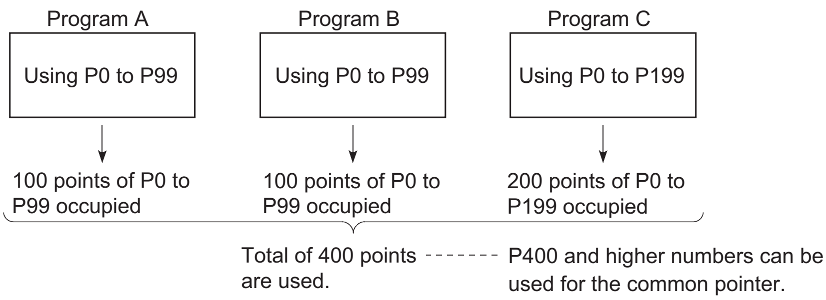
Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System]



Ex. If a total of 400 points are used in three programs (100 points in each of Program A and Program B, and 200 points in Program C), for example, P400 and higher numbers can be set for the common pointer.

Point

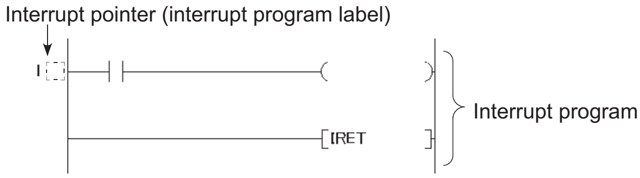
- The same pointer number cannot be used as a label. If used, "CAN'T SET(P)" (error code: 4021) occurs.
- If the total number of the local pointer points used in several programs exceeds the start number of the common pointer, a "CAN'T SET (P)" (error code: 4020) occurs.



- The jump instructions are not capable of executing a jump to the common pointer in other programs. Use the common pointer with subroutine call instructions only.

5.11 Interrupt Pointer (I)

The interrupt pointer (I) is used as a label at the start of an interrupt program, and can be used in any programs.



The number of points available for the interrupt pointer is 256 (I0 to I255). The following shows interrupt factors for the applicable interrupt pointers.

Interrupt factor	Interrupt pointer No.	Description
Interrupt by built-in I/O	I0 to I15*1	Interrupt from a built-in I/O
Interrupt by an internal timer	I28 to I31	Interrupt at fixed intervals by an internal timer of the CPU module
Intelligent function module interrupt	I50 to I255*2	Interrupt from the intelligent function module

*1 Interrupt pointer numbers can be changed in the PLC Parameter dialog box.
*2 They can be used for interruptions by built-in I/O by configuring parameter in the "PLC Parameter" dialog box.

Point

To use the intelligent function module interrupt, set values in the "Interrupt Pointer Setting" dialog box opened from the "PLC System" tab of the PLC Parameter dialog box. (☞ Page 354, Appendix 1.2 (2))

The list of interrupt pointer numbers and interrupt factors are shown below.

I No.	Interrupt factor		Priority	I No.	Interrupt factor		Priority
I0	Interrupt by built-in I/O	1st point	5	I16 to I27	—	Empty	—
I1		2nd point	6				
I2		3rd point	7				
I3		4th point	8				
I4		5th point	9				
I5		6th point	10				
I6		7th point	11	I28	Interrupt by internal timer*1*2	100ms	4
I7		8th point	12	I29		40ms	3
I8		9th point	13	I30		20ms	2
I9		10th point	14	I31		10ms	1
I10		11th point	15	I32 to I49	—	Empty	—
I11		12th point	16	I50 to I255	Intelligent function module interrupt	Empty	21 to 226*3
I12		13th point	17				
I13		14th point	18				
I14		15th point	19				
I15		16th point	20				

*1 The time-limit value of the internal timer is set by default. In the PLC System tab of the PLC Parameter dialog box, the value can be changed within the range of 2ms to 1000ms in increments of 1ms.
*2 When an interrupt occurs, even if no interrupt pointer exists on the program, "CAN'T EXECUTE(I)" (error code: 4220) does not occur.
*3 Among I50 to I255, I50 has the highest priority (priority 21), and I255 has the lowest priority (priority 226).

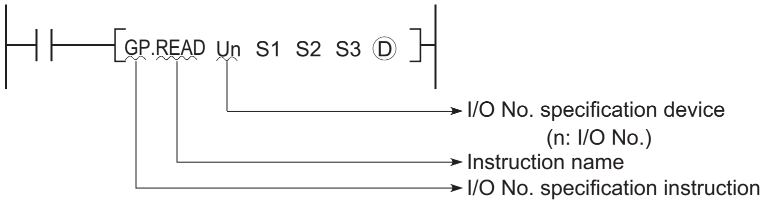
5.12 Other Devices

5.12.1 SFC block device (BL)

The SFC block is used to check that the specified block in the SFC program is activated.
📖MELSEC-Q/L/QnA Programming Manual (SFC)

5.12.2 I/O No. specification device (U)

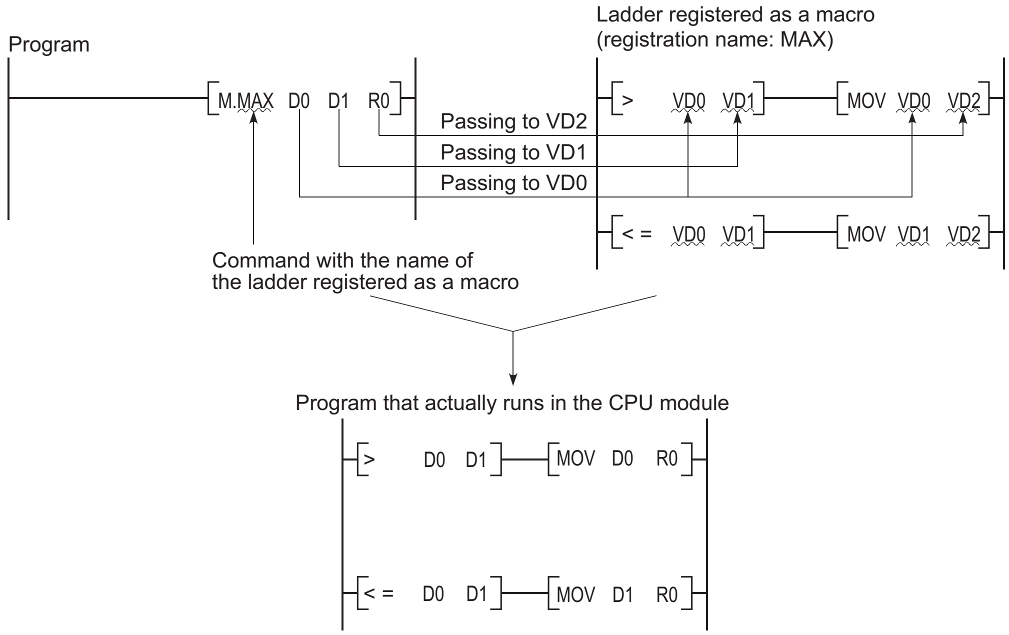
The I/O No. specification device is used to specify I/O numbers in the intelligent function module dedicated instructions. The following shows how to specify the device.



For details on the intelligent function module dedicated instructions, refer to the following.
📖Manual for the intelligent function module used

5.12.3 Macro instruction argument device (VD)

The macro instruction argument device (VD) is used with ladders registered as macros. When a VD□ setting is specified, the value is converted to the specified device when the macro instruction is executed. With the macro instruction argument device, VD0 to VD9 can be used in one macro registration ladder.
Among the devices used in the ladders registered as macros, specify a device used for VD. When using macro instructions in the program, specify devices that correspond to the macro instruction argument devices used in the macro registration ladders in ascending order.



CHAPTER 6 CONSTANTS

The following constants can be used in the CPU module.

- Decimal constant (K)
- Hexadecimal constant (H)
- Real number (E)
- Character string (" ")

6.1 Decimal Constant (K)

The decimal constant (K) is a device used to specify decimal data in programs. Specify it as K☐ (example: K1234) in programs. In the CPU module, data are stored in binary (BIN). (☞ Page 389, Appendix 4)

The specification ranges for decimal constants are as follows:
The most significant bit represents a sign bit.

- When using word data (16-bit data): K-32768 to K32767
- When using 2-word data (32-bit data): K-2147483648 to K2147483647

6.2 Hexadecimal Constant (H)

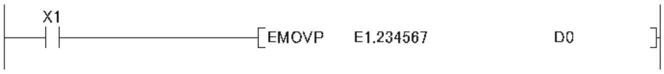
The hexadecimal constant (H) is a device used to specify hexadecimal or BCD data in programs. (For BCD data, each digit of a hexadecimal number is specified with 0 to 9.) In programs, specify it as H☐ (example: H1234). (☞ Page 389, Appendix 4)

The specification ranges for hexadecimal constants are as follows:

- When using word data (16-bit data): 0000_H to $FFFF_H$ (For BCD data, 0000_H to 9999_H)
- When using 2-word data (32-bit data): 00000000_H to $FFFFFFFF_H$ (For BCD data, 00000000_H to 99999999_H)

6.3 Real Number (E)

The real number (E) is a device used to specify real numbers in programs. In programs, specify it as E□ (example: E1.234).
(☞ Page 374, Appendix 1.3)



(1) Specification range

(a) Real number setting range

- For single-precision floating-point data
  $-2^{128} <$ Device $\leq -2^{-126}$, 0, $2^{-126} \leq$ Device $< 2^{128}$
- For double-precision floating-point data[*1]
  $-2^{1024} <$ Device $\leq -2^{-1022}$, 0, $2^{-1022} \leq$ Device $< 2^{1024}$

*1 Up to 15 digits can be entered in a programming tool.

(b) When an overflow or underflow has occurred

Overflow	Underflow
OPERATION ERROR (error code: 4141)	Turned to 0 without any error

(c) When a special value[*1] is input

If operation is performed with input data that contains a special value, "OPERATION ERROR" (error code: 4140) occurs.

*1 The special values are -0, unnormalized numbers, nonnumeric characters, and ±∞.

(2) Specification method

Real numbers can be specified in programs by the following expressions.

- Normal expression: A numeric value can be specified as it is. (Example: 10.2345 can be specified as E10.2345.)
- Exponential expression: A numeric value is specified by (Value) × 10^n. (Example: 1234 is specified as E1.234 + 3.[*1])

*1 + 3 represents 10^3 in E1.234 + 3.

6.4 Character String (" ")

The character string is a device used to specify a character string in program. Characters enclosed in quotation marks are specified. Alphabets are case-sensitive. A string from the specified character to the NUL code (00_H) is one unit. Note that, however, up to 32 characters can be specified for an instruction using a character string, such as $MOV.

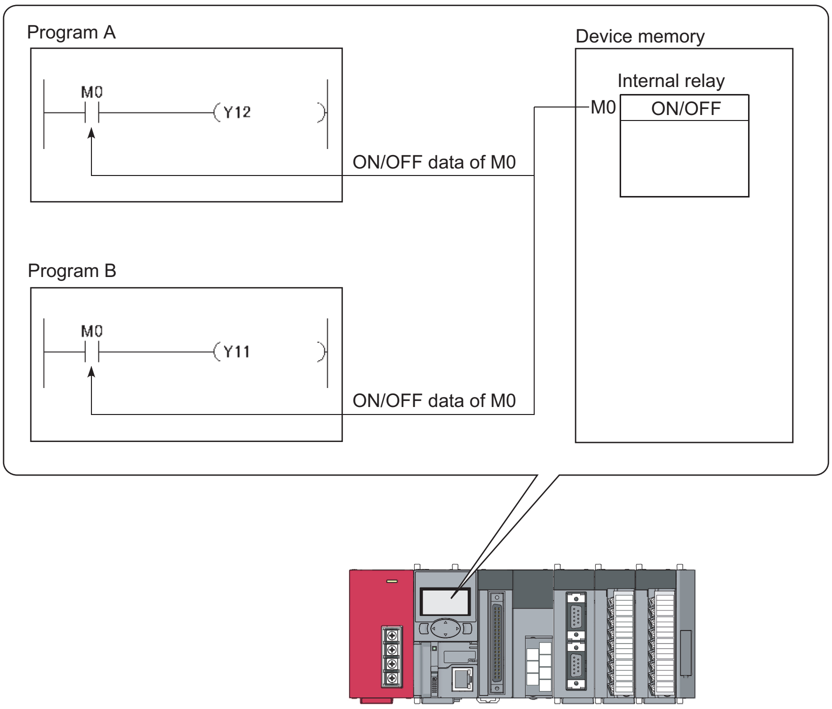
CHAPTER 7 CONVENIENT USAGE OF DEVICES

When multiple programs are executed in the CPU module, each program can be executed independently by specifying an internal user device as a local device. Devices of the CPU module are classified into the following two types:

- Global device that can be shared by multiple programs that are being executed.
- Local device that is used independently for each program.

7.1 Global Device

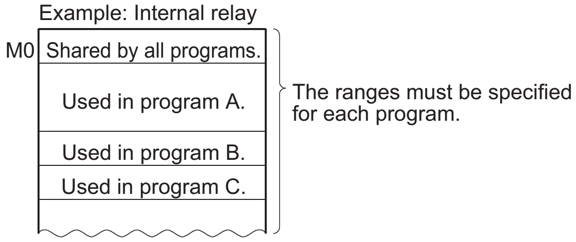
Programs being executed in the CPU module can share the global device. Global device data are stored in the device memory of the CPU module, and can be shared by all programs.



Point

- All of the devices that have not been set as local devices (☞ Page 345, Section 7.2) are global devices.
- For execution of multiple programs, the range to be shared by all programs and the range to be used independently by each program must be specified in advance. (☞ Page 345, Section 7.2)

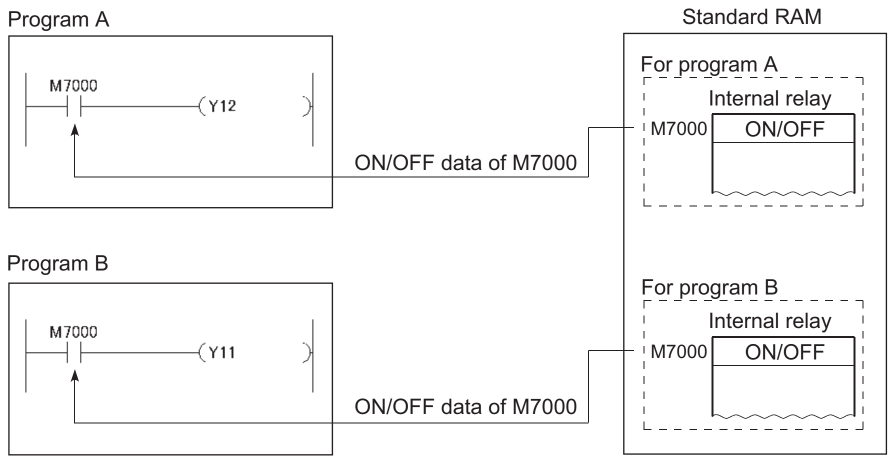
Example: Internal relay



7.2 Local Device

The local device is a device that can be used independently for each program. Using local devices allows programming of multiple independently-executed programs without considering other programs. Note that local device data can be stored in the standard RAM only.

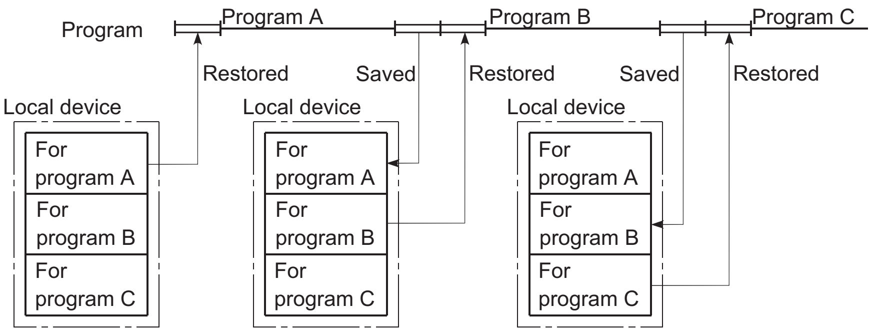
Ex. If M7000 and higher portion is set as a local device, it can be separately used for each program that is executing M7000 and higher portion.



The following devices can be used as local devices.

- Internal relay (M)
- Data register (D)
- Counter (C)
- Timer (T, ST)
- Edge relay (V)
- Index register (Z)

After program execution, data in the local device file of the standard RAM are exchanged with the data in the device memory of the CPU module. For this reason, the scan time increases by the time for data exchange.

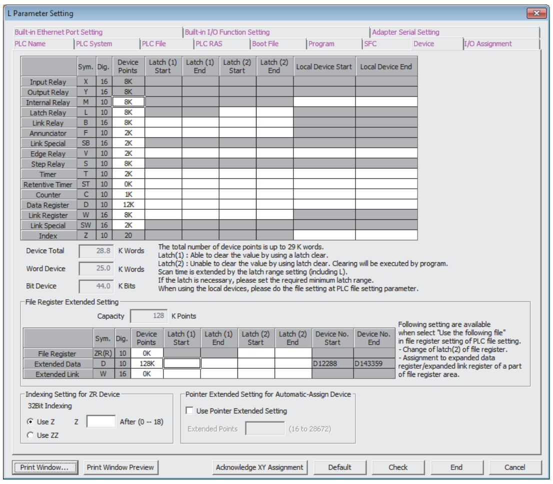


Point

- There are some instructions for which a local device cannot be specified. For details, refer to the pages describing devices available for each instruction in the following manual.
  MELSEC-Q/L Programming Manual (Common Instruction)
- For the concept of the number of words used for the local devices, refer to Page 290, Section 5.2.

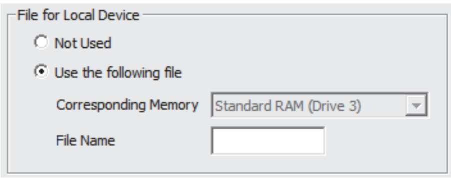
(1) Local device setting

To use a local device, perform the following procedure.



1. Set a local device range.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Device]



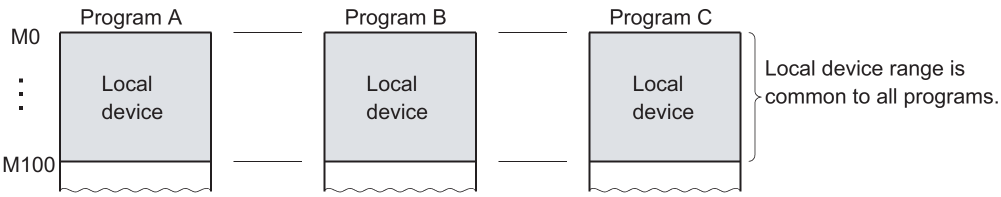
2. Set a memory for storing the local device file and a file name.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC File]

3. Write the parameters to the CPU module.

Note that the local device range is common to all programs, and cannot be changed for each program.

Ex. If a local device range is specified as MO to M100, this range setting applies to all programs that use the local device.

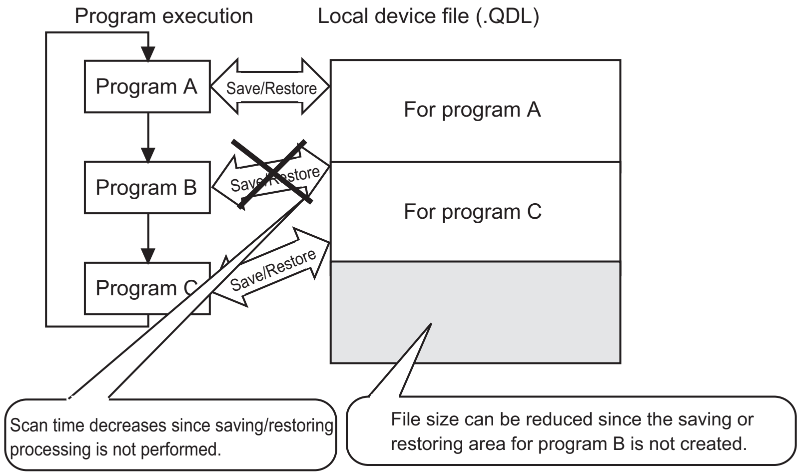


Point

- The 32-bit index modification range must not overlap with the local device setting range of the index register. If overlapped, 32-bit index modification values will be written over the local device values.
- If the size of the local device in the standard RAM is set to be changed while a sampling trace file is stored in the standard RAM, the sampling trace file will be cleared. To save the trace results, store them in a personal computer.
- All of the devices that have not been set as local devices are global devices.

(2) Setting a local device in units of program

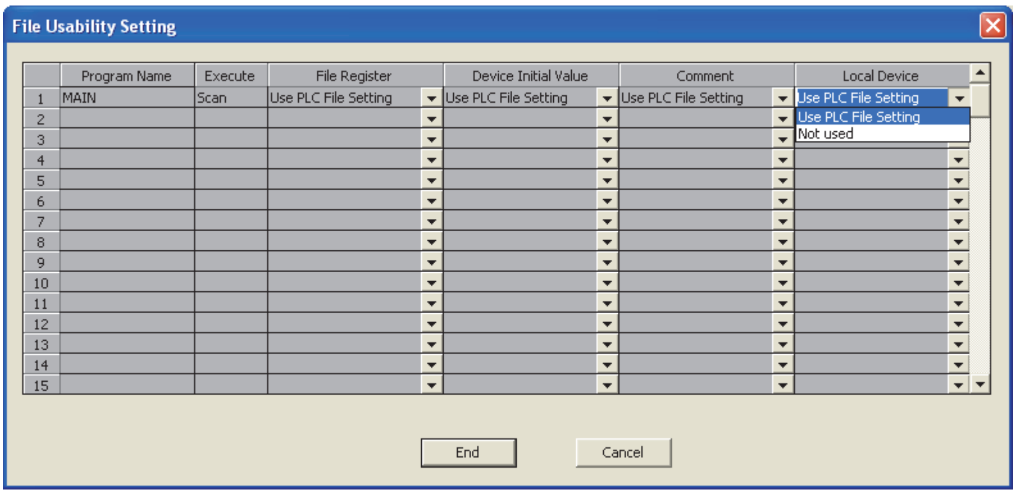
Use of the local device can be set for each program, and this function can reduce the scan time. Also, since the area for saving and restoring data are not required for the programs not using a local device, the local device file size can be reduced.



(a) Setting method

In addition to the setting in (1) in this section, click the "File Usability Setting" button, and specify the programs that use the local devices.

Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [Program] ⇨ [File Usability Setting]



(b) Precautions

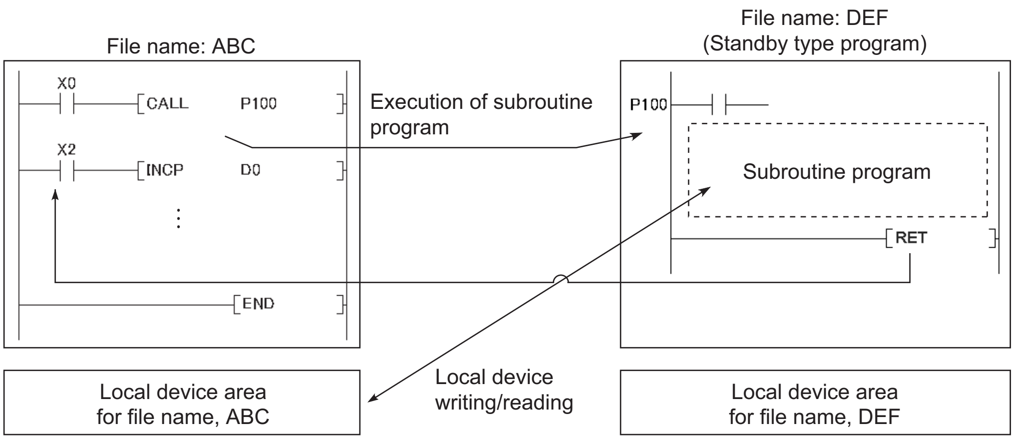
Do not change or refer to the local device in a program for which the local device is set to "Not used". The changed data are not held.

(3) Local devices when executing a subroutine program

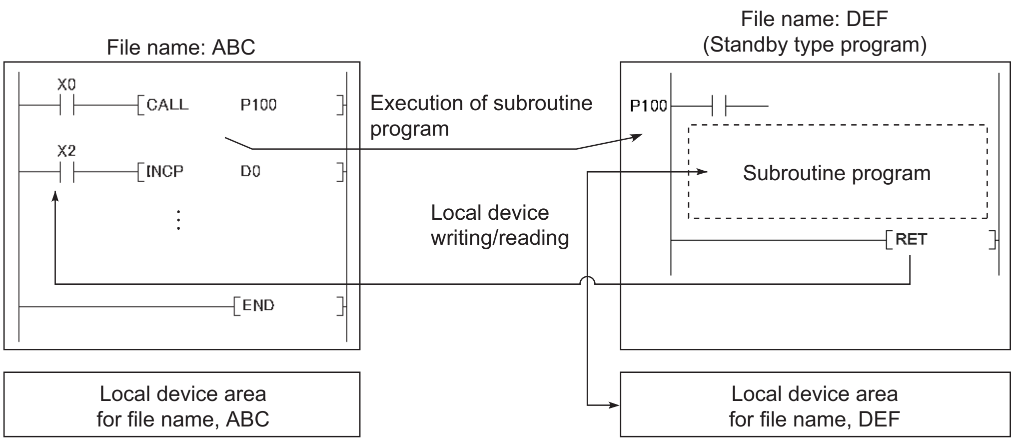
When executing a subroutine program, the local device in the file where the subroutine program is stored can be used by ON/OFF of SM776.

SM776	Operation
OFF	Perform operations with the local device that corresponds to the source file of the subroutine program.
ON	Perform operations with the local device that corresponds to the file where the subroutine program is stored.

(a) When SM776 is off



(b) When SM776 is on



(c) Precautions

- When SM776 is on, local device data area read out when a subroutine program is called, and the data are saved after execution of the RET instruction. Because of this, the scan time is increased if one subroutine program is executed with SM776 set to on.
- The on/off status of SM776 is set for each CPU module. It cannot be set for each file.
- If the on/off status of SM776 is changed during program execution, control is implemented according to the information after the change.

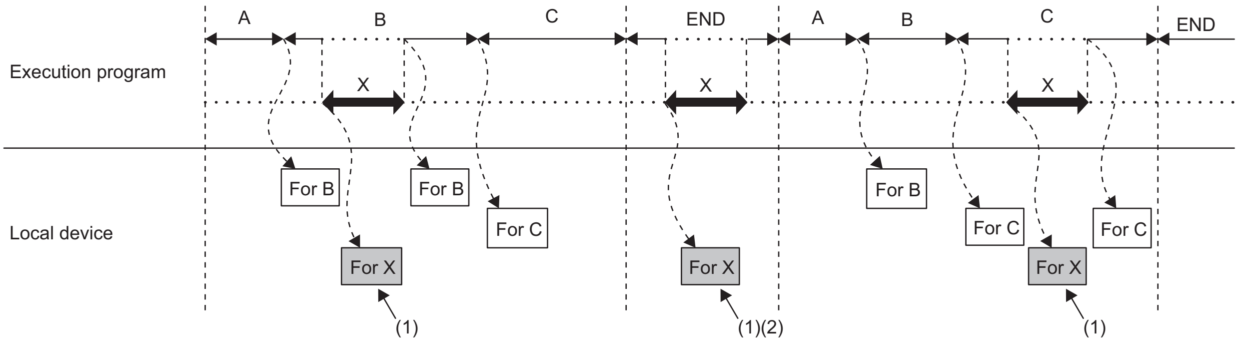
(4) When executing an interrupt/fixed scan execution type program

When the local device is used for an interrupt/fixed scan execution type program, turn on SM777 (Enable/disable local device in interrupt program). The programs will not function properly if SM777 is turned off.[*1]

*1 The index register set as the local device uses the local device area for the program executed before the interrupt/fixed scan execution type program, regardless of the on/off status of SM777.

Ex. Operation when SM777 is turned on with the following setting

Program name	Execute type	Local device
A	Scan	Not used
B	Scan	Used
C	Scan	Used
X	Fixed Scan	Used



(1)Uses the program X local device.
(2)When an interrupt/fixed scan execution type program is executed during the END processing, the local device for Program C, which was executed before the END processing, is read out and saved. Thus, the END processing time increases by the time required for the read and save.

(a) Precautions

- When SM777 is on, local device data are read out before execution of an interrupt/fixed scan execution type program, and the data are saved after execution of the IRET instruction. Because of this, the scan time is increased if one interrupt/fixed scan execution type program is executed with SM777 set to on.
- The on/off status of SM777 is set for each CPU module. It cannot be set for each file.
- For the local device monitoring, the target local device is monitored by switching the selection of corresponding program manually. When SM777 is off, when an interrupt occurred immediate after the switch processing, monitoring target local device is monitored. (The local device for the program executed prior to the interrupt (the program immediate before END) is not monitored.)

(5) Clearing local device data

Local device data are cleared by the either of the following.

- When the CPU module is powered off and then on or is reset.
- When the CPU module status is changed from STOP to RUN.

Local device data cannot be cleared from a programming tool.

APPENDICES

Appendix 1 Parameter Setting

Appendix 1.1 List of parameter numbers

Each parameter number is the value that is stored in the special register (SD16 to SD26) when an error occurs in the parameter settings. The following list indicates the parameter items and corresponding parameter numbers.

<table>
<tr><th>Parameter No.</th><th colspan="2">Parameter item</th><th>Set in:</th><th>Reference</th></tr>
<tr><td>0000$_H$</td><td colspan="2">Label</td><td rowspan="2">PLC Name</td><td>—</td></tr>
<tr><td>0001$_H$</td><td colspan="2">Comment</td><td>—</td></tr>
<tr><td>1000$_H$</td><td colspan="2">Timer Limit Setting</td><td rowspan="17">PLC System</td><td>Page 299, Section 5.2.10</td></tr>
<tr><td>1001$_H$</td><td colspan="2">RUN-PAUSE Contacts</td><td>Page 112, Section 3.12</td></tr>
<tr><td>1002$_H$</td><td colspan="2">Remote Reset</td><td>Page 112, Section 3.12</td></tr>
<tr><td>1003$_H$</td><td colspan="2">Output Mode at STOP to RUN</td><td>Page 101, Section 3.7</td></tr>
<tr><td>1004$_H$</td><td colspan="2">Floating-point operation</td><td>Page 389, Appendix 4</td></tr>
<tr><td>1005$_H$</td><td colspan="2">Common Pointer No.</td><td>Page 338, Section 5.10.2</td></tr>
<tr><td>1006$_H$</td><td colspan="2">Pointer Extended Setting for Automatic-Assign Device</td><td>Page 336, Section 5.10</td></tr>
<tr><td>1007$_H$</td><td colspan="2">Points Occupied by Empty Slot</td><td>Page 41, Section 2.2</td></tr>
<tr><td>1008$_H$</td><td colspan="2">System Interrupt Setting</td><td rowspan="2">Page 339, Section 5.11</td></tr>
<tr><td>100A$_H$</td><td colspan="2">Intelligent Function Module Setting (Interrupt Pointer Setting)</td></tr>
<tr><td>100C$_H$</td><td colspan="2">Module Synchronization</td><td>Page 91, Section 3.5</td></tr>
<tr><td>100D$_H$</td><td colspan="2">A-PLC Compatibility Setting</td><td>—</td></tr>
<tr><td>100E$_H$</td><td colspan="2">Serial communication function</td><td>Page 227, Section 3.38</td></tr>
<tr><td>1013$_H$</td><td colspan="2">Service Processing Setting</td><td>Page 94, Section 3.6</td></tr>
<tr><td>1014$_H$</td><td colspan="2">Latch Data Backup Operation Valid Contact</td><td>Page 175, Section 3.28</td></tr>
<tr><td>1016$_H$</td><td colspan="2">Built-in Ethernet Port Setting</td><td>Built-in Ethernet Port Setting</td><td>—</td></tr>
<tr><td>1017$_H$</td><td colspan="2">PLC Module Change Setting</td><td rowspan="2">PLC System</td><td>Page 182, Section 3.31</td></tr>
<tr><td>1018$_H$</td><td colspan="2">Built-in CC-Link Setting</td><td>—</td></tr>
<tr><td>1019$_H$</td><td colspan="2">Simple PLC Communication Setting</td><td>Built-in Ethernet Port Setting</td><td>—</td></tr>
<tr><td>101A$_H$</td><td colspan="2">Serial Communication Function</td><td rowspan="3">Adapter Serial Setting, Built-in Serial Setting</td><td>Page 227, Section 3.38</td></tr>
<tr><td>101B$_H$</td><td colspan="2">Predefined Protocol Support Function</td><td>Page 215, Section 3.37</td></tr>
<tr><td>101C$_H$</td><td colspan="2">GOT Connection</td><td>—</td></tr>
<tr><td>101D$_H$</td><td colspan="2">FTP Client Setting</td><td rowspan="4">Built-in Ethernet Port Setting</td><td>—</td></tr>
<tr><td>101E$_H$</td><td colspan="2">E-mail Setting</td><td>—</td></tr>
<tr><td>1030$_H$</td><td rowspan="2">CC-Link IEF Basic Setting</td><td>Network Configuration Setting</td><td>—</td></tr>
<tr><td>1031$_H$</td><td>Refresh Setting</td><td>—</td></tr>
</table>

Parameter No.	Parameter item		Set in:	Reference
1100$_H$	File Register		PLC File	Page 326, Section 5.7
1101$_H$	Comment File Used in a Command			Page 344, CHAPTER 7
1102$_H$	Initial Device Value			Page 91, Section 3.5
1103$_H$	File for Local Device			Page 345, Section 7.2
1104$_H$	Transfer to Standard ROM at Latch data backup operation			Page 175, Section 3.28
1105$_H$	Device data storage file			Page 180, Section 3.29
2000$_H$	Device Points		Device	Page 88, Section 3.4
2001$_H$	Latch range (latch clear is enabled)			
2002$_H$	Latch range (latch clear is disabled)			
2003$_H$	Local variable			
2004$_H$	Latch range extended devices (latch clear is enabled)			
2005$_H$	Latch range extended devices (latch clear is disabled)			
3000$_H$	WDT (Watchdog Timer) Setting		PLC RAS	Page 87, Section 3.3
3001$_H$	Error Check			Page 166, Section 3.25
3002$_H$	Operating Mode When there is an Error			
3003$_H$	Constant Scanning		PLC RAS	Page 85, Section 3.2
300A$_H$	Module Error History Collection			Page 171, Section 3.27
0400$_H$	Slot setting		I/O Assignment	Page 41, Section 2.2
0403$_H$	Error Time Output Mode			Page 103, Section 3.9
4004$_H$	Error Time Operation Mode			
0405$_H$	I/O Response Time			Page 102, Section 3.8
0409$_H$	Switch Setting			—
5001$_H$[*1]	Valid Module During Other Station Access		CC-Link IE Field Network setting	Page 374, Appendix 1.3
5003$_H$[*1]	Routing Parameter			
7000$_H$	Program Setting		Program	Page 61, Section 2.8
8002$_H$	SFC Program Start Mode		SFC	—
8003$_H$	Start Conditions			
8006$_H$	Output Mode When the Block is Stopped			
9000$_H$	Number of modules on Ethernet		Ethernet	Page 374, Appendix 1.3
9N00$_H$	Start I/O No.			
	Network No.			
	Group No.			
	Station No.			
	Operation Setting			
9N01$_H$	Initial Setting			
9N02$_H$	Open Setting			
9N03$_H$	Router relay parameter			
9N05$_H$	Station No. <-> IP information			
9N06$_H$	FTP Parameters			
9N07$_H$	E-mail Setting	—		
9N08$_H$		News Setting		
9N09$_H$	Interrupt Settings			
9N04$_H$	Routing Parameters			

Parameter No.	Parameter item	Set in:	Reference
$A080_H$ *1	Number of Modules Setting	CC-Link IE Field Network setting	Page 374, Appendix 1.3
$ANM0_H$ *1	Network Setting		
$ANM1_H$ *1	Refresh Parameter		
$ANM2_H$ *1	Common Parameter		
$ANM3_H$ *1	Intrinsic parameter		
$B000_H$	Positioning and High-speed counter functions settings	Built-in I/O Function Setting	MELSEC-L CPU Module User's Manual (Built-In I/O Function)
$B001_H$	Input Signal Function Selection		
$B002_H$	Output Signal Function Selection		
$B003_H$	Input Response Time		
$B004_H$	Interrupt Processing Condition		
$B005_H$	Error Time Output Mode		
$B006_H$, $B007_H$	Axis1: Positioning Parameter		
$B008_H$	Axis1: OPR Parameter		
$B009_H$	Axis1: Positioning Data		
$B00A_H$, $B00B_H$	Axis2: Positioning Parameter		
$B00C_H$	Axis2: OPR Parameter		
$B00D_H$	Axis2: Positioning Data		
$B00E_H$	CH1: Common setting		
$B00F_H$, $B010_H$	CH1: Operation Mode		
$B011_H$	CH1: Frequency Measurement Mode		
$B012_H$	CH1: Rotation Speed Measurement Mode		
$B013_H$	CH1: Pulse Measurement Mode		
$B014_H$	CH2: Common Setting		
$B015_H$, $B016_H$	CH2: Normal Mode Setting		
$B017_H$	CH2: Frequency Measurement Mode		
$B018_H$	CH2: Rotation Speed Measurement Mode		
$B019_H$	CH2: Pulse Measurement Mode		
$C000_H$	Number of Modules	I/O Assignment	Page 43, Section 2.2.2
$CNM1_H$ *1	Network Refresh Setting	CC-Link setting	—
$CNM2_H$ *1	Common Parameters		
$FE00_H$	Parameters for the programming tool	—	—
Start I/O number ÷ 10	Refresh Parameter	Intelligent function module parameter	Page 77, Section 2.11
—	Password Active Module Setting	Remote password	Page 375, Appendix 1.4

*1 "N" indicates the position of the module (counted from the first one) and "M" indicates the network type of the module.

Appendix 1.2 PLC parameters

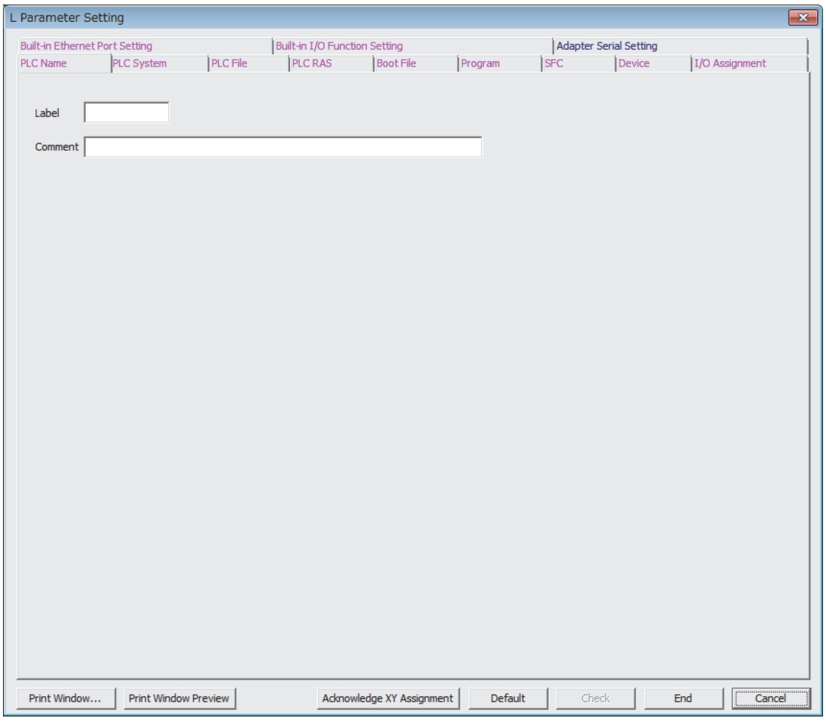
This section provides parameter setting screens and details of the setting items.

Point

Note that parameter setting is not available for the grayed out items.

(1) PLC Name setting

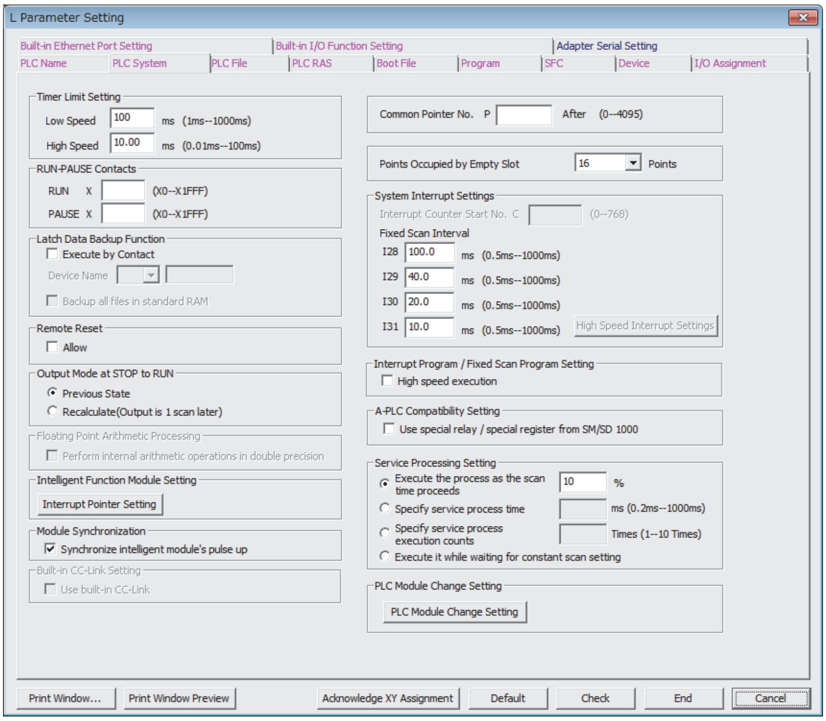
A label name and a comment for the CPU module are set. The settings are displayed in the list for the find CPU function. (MELSEC-L CPU Module User's Manual (Built-In Ethernet Function))



Item	Parameter No.	Description	Setting range	Default	Reference
Label	0000_H	Set a label (name, application) for the CPU module.	Up to 10 characters	—	—
Comment	0001_H	Set a comment for the CPU module label.	Up to 64 characters	—	—

(2) PLC System setting

Parameters of the system required for use of the CPU module are set.



Item		Parameter No.	Description	Setting range	Default	Reference
Timer Limit Setting	Low Speed	1000_H	Set the time limit for the low speed timer or high speed timer.	1ms to 1000ms (in increments of 1ms)	100ms	Page 299, Section 5.2.10
	High Speed			0.01ms to 100.0ms (in increments of 0.01ms)	10ms	
RUN-PAUSE Contacts	RUN	1001_H	Set the contacts that control RUN PAUSE of the CPU module.	X0 to X1FFF	—	Page 112, Section 3.12
	PAUSE [*1]					
Latch Data Backup Function		1014_H	Set the valid contact device No. used for backup of latch data to the standard ROM.	X, M, B	—	Page 175, Section 3.28
Remote Reset		1002_H	Select whether to allow the remote reset.	Selected/deselected	Deselected	Page 112, Section 3.12
Output Mode at STOP to RUN		1003_H	Set the status of the outputs (Y) when the operating status is switched from STOP to RUN.	Previous state, Recalculate (output is 1 scan later)	The previous state	Page 101, Section 3.7
Intelligent Function Module Setting (Interrupt Pointer Setting)		$100A_H$	Assign the interrupt pointers (I50 to I255) and set the start I/O number and start SI number of each intelligent function module.	[Start I/O No.] • L02SCPU, L02SCPU-P, L02CPU, L02CPU-P: 0_H to $3F0_H$ • L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT: 0_H to $FF0_H$ [Start SI No.] 50 to 255	—	Page 339, Section 5.11
Module Synchronization		$100C_H$	Select whether to synchronize CPU module startup with intelligent function module startup.	Selected/deselected	Selected	Page 91, Section 3.5

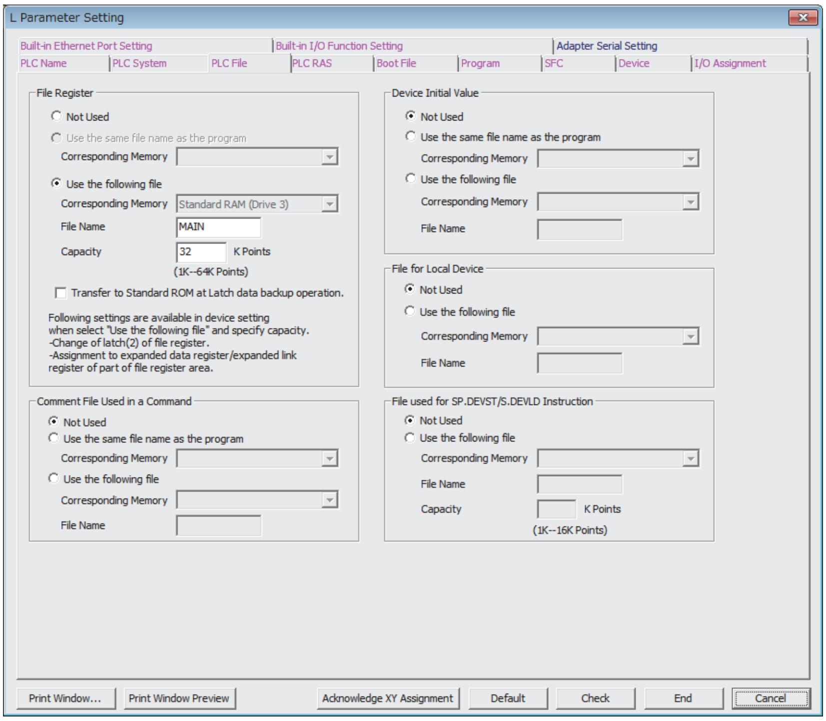
Item	Parameter No.	Description	Setting range	Default	Reference
Built-in CC-Link Setting *2	1018_H	Select whether to use the built-in CC-Link	Selected/deselected	Selected	MELSEC-L CC-Link System Master/Local Module User's Manual
Common Pointer No.	1005_H	Set the start number of common pointers.	P0 to P4095	—	Page 338, Section 5.10.2
System Interrupt Setting (Fixed Scan Interval)	1008_H	Set each execution interval for the interrupt pointers (I28 to I31).	0.5ms to 1000ms (In increments of 0.5ms)	• I28: 100.0ms • I29: 40.0ms • I30: 20.0ms • I31: 10.0ms	Page 339, Section 5.11
Interrupt Program/Fixed Scan Program Setting	1008_H	Select whether to enable or disable high speed execution of interrupt programs or fixed scan execution type programs.	Selected/deselected	Deselected	Page 324, Section 5.6.3
Service Processing Setting	1013_H	Select any of the following. • Execute the process as the scan time proceeds. • Specify service process time. • Specify service process execution counts. • Execute it while waiting for constant scan setting.	• 1 to 99% (in increments of 1%) • 1 to 10 (in increments of 1 time) • 0.2 to 1000ms (in increments of 0.1ms) • Blank	Execute the process as the scan time proceeds: 10%	Page 94, Section 3.6
CPU Module Change Setting	1017_H	Set items required when performing the CPU module change with SD memory card function.	• Backup Start Setup Contact • Backup Start Contact • Backup Target Data • Title Setting	—	Page 182, Section 3.31

*1 Setting of only a PAUSE contact is not allowed.

*2 This item is not available for the L02SCPU, L02SCPU-P, L02CPU, L02CPU-P, L06CPU, L06CPU-P, L26CPU, and L26CPU-P because these modules do not support the built-in CC-Link function.

(3) PLC File setting

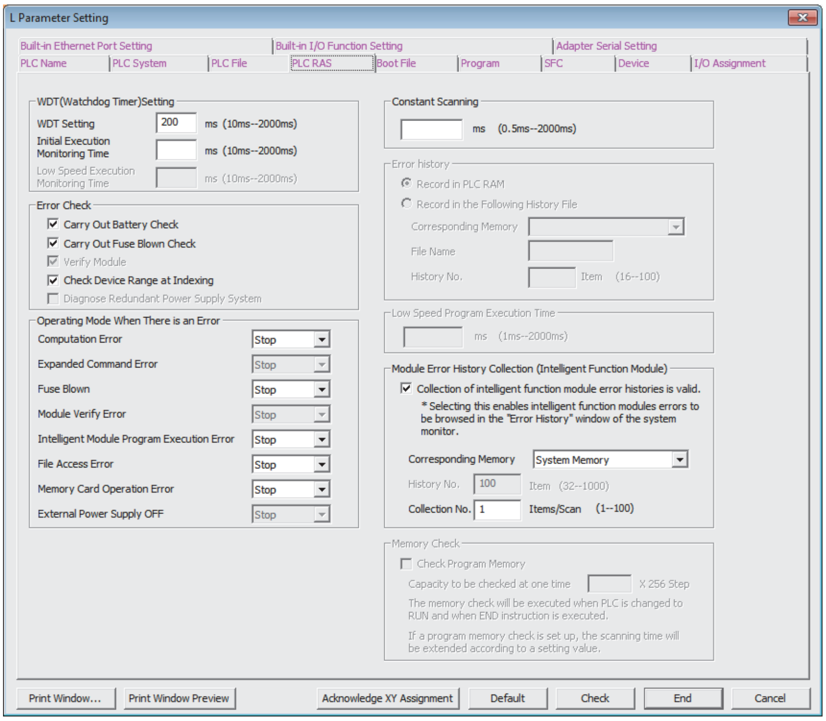
Parameters required for the files used in the CPU module are set.



Item		Parameter No.	Description	Setting range	Default	Reference
File Register	—	1100_H	Set a file for the file register used in the program.	• Not used • Use the following file	Use the following file	Page 326, Section 5.7
	Transfer to Standard ROM at Latch data backup operation	1104_H	Select whether to batch-transfer the data in the file register at the time of latch data backup to the standard ROM.	Selected/deselected	—	Page 175, Section 3.28
Comment File Used in a Command		1101_H	Set a file for device comments used in the program.	• Not used • Use the same file name as the program • Use the following file	Not used	—
Initial Device Value		1102_H	Set a file for initial values of the devices used for the CPU module.	• Not used • Use the same file name as the program • Use the following file	Not used	Page 91, Section 3.5
File for Local Device		1103_H	Set a file for local devices used in the program.	• Not used • Use the following file	Not used	Page 345, Section 7.2
File used for SP.DEVST/S.DEVLD Instruction		1105_H	Set a device data storage file used for writing to or reading from the standard ROM.	• Not used • Use the following file	Not used	Page 180, Section 3.29

(4) PLC RAS setting

Parameters required for performing the RAS functions are set.



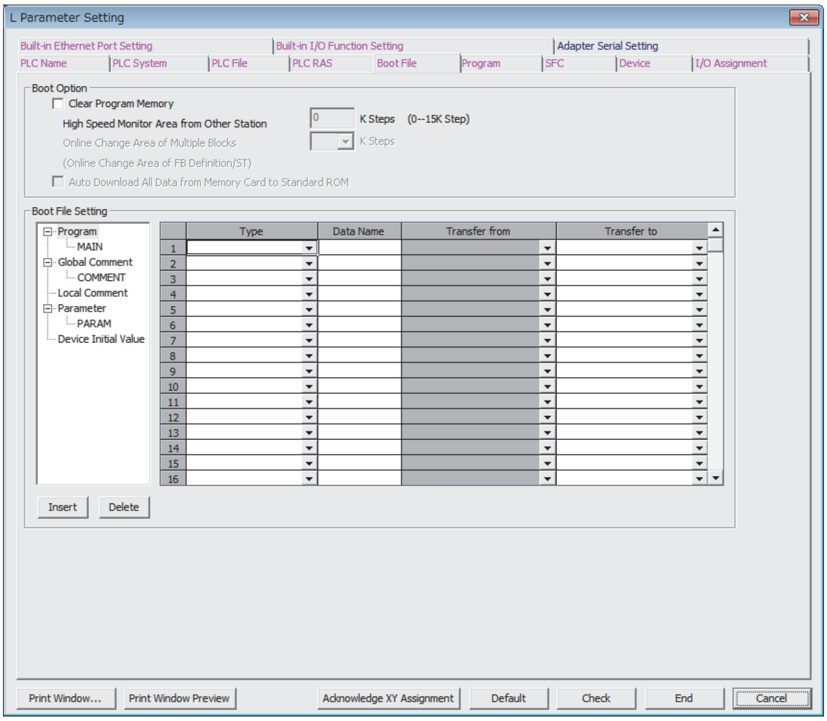
Item		Parameter No.	Description	Setting range	Default	Reference
WDT (Watchdog Timer) Setting	WDT Setting	3000_H	Set a watchdog timer value.	10ms to 2000ms (In increments of 10ms)	200ms	Page 87, Section 3.3
	Initial Execution Monitoring Time		Set a watchdog timer value for using an initial execution type program.	10ms to 2000ms (In increments of 10ms)	—	Page 63, Section 2.8.1
Error Check		3001_H	Whether to check the following errors can be selected. • Carry Out Battery Check • Carry Out Fuse Blown Check • Check Device Range at Indexing	Selected/deselected	Selected	Page 166, Section 3.25
Operating Mode When There is an Error		3002_H	Set the operation mode of the CPU module when the following errors are detected. • Computation Error • Fuse Blown • Intelligent Module Program Execution Error • File Access Error • Memory Card Operation Error *1	Stop/Continue	Stop	Page 166, Section 3.25
Constant Scanning		3003_H	Set a constant scan time value.	0.5ms to 2000ms (In increments of 0.5ms)	—	Page 85, Section 3.2

Item		Parameter No.	Description	Setting range	Default	Reference
Module Error History Collection (Intelligent Function Module)	Collection of intelligent function module error histories is valid.	$300A_H$	Set whether to collect module errors.	Selected/deselected	Selected	Page 171, Section 3.27
	Corresponding Memory		Select a storage location.	• System Memory • Standard RAM	System Memory	
	History No.		Enter the number of errors to collect only when they are stored in the standard RAM.	32 to 1000	100	
	Collection No.		Enter the number of errors to collect in one scan.	• Stored in system memory: 1 to 100 • Stored in standard RAM: 1 to 128	1	

*1 This item is not available for the L02SCPU and L02SCPU-P because these modules do not support the use of SD memory cards.

(5) Boot File setting

Parameters required for boot operations are set.



Remark

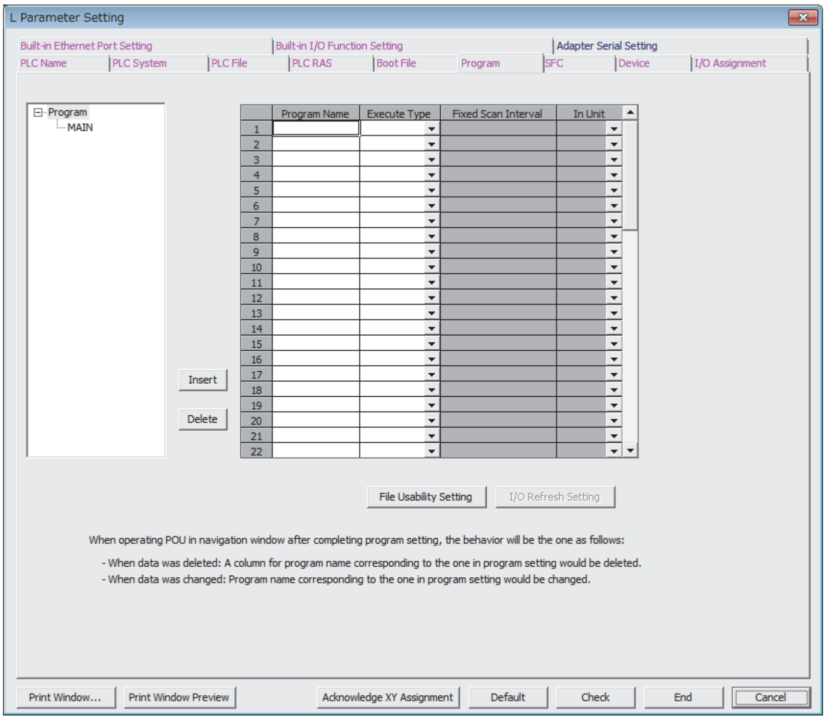
These parameters are not available for the L02SCPU and L02SCPU-P.

Item		Parameter No.	Description	Setting range	Default	Reference
Boot Option	Clear Program Memory	7000_H	Select whether to clear the program memory at the time of boot. Selecting this item enables the setting of "High Speed Monitor Area from Other Station". *1	Selected/deselected	Deselected	Page 74, Section 2.9
Boot File Setting			Set the type and data name of the boot file, and transfer source and destination drives for boot operation.	—	—	

*1 Selecting this area achieves faster monitoring from a programming tool connected to a device such as a serial communication module.

(6) Program setting

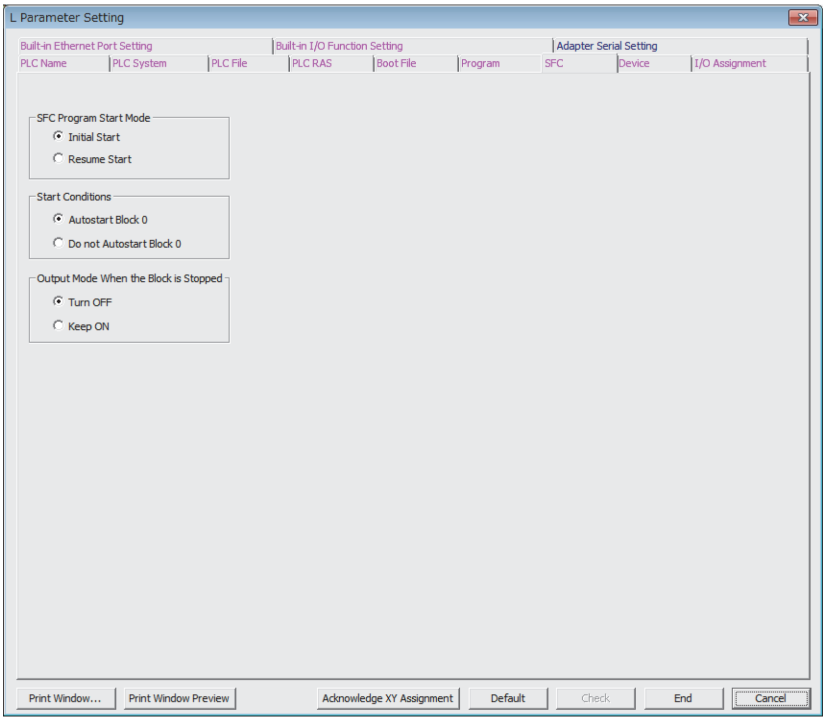
File names and execution types (execution conditions) are set for each program when more than one programs are written to the CPU module.



Item	Parameter No.	Description	Setting range	Default	Reference
Program setting	7000_H	Set the execution order and type of the program. For a fixed scan execution type program, set the fixed scan interval (execution interval).	• Wait • Scan • Initial • Fixed Scan	—	Page 61, Section 2.8
File Usability Setting		Set the use of local device files.	• Set (use the devices according to the PLC File settings) • Not used	—	

(7) SFC setting

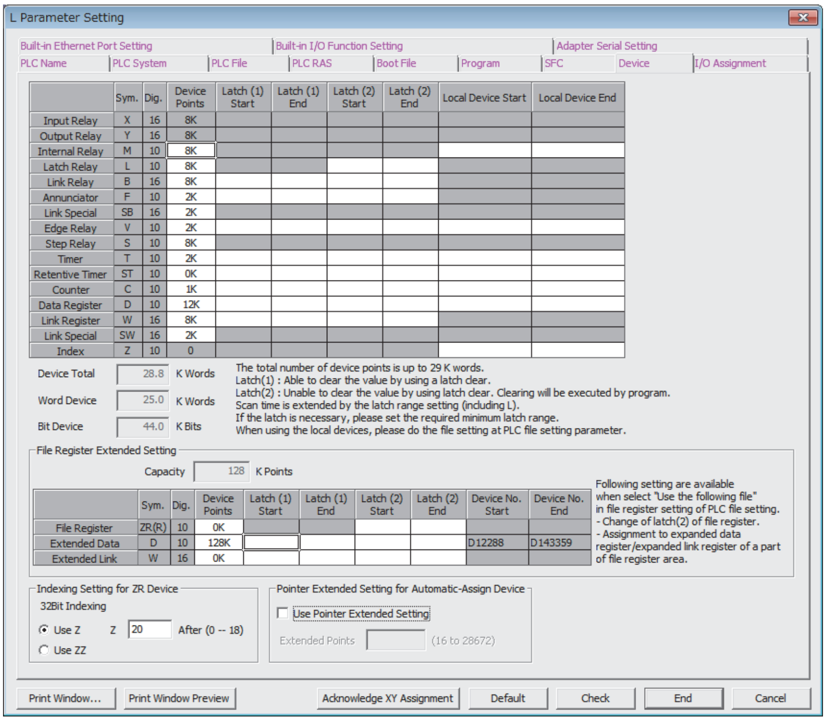
Parameters required for SFC programs are set.



Item	Parameter No.	Description	Setting range	Default	Reference
SFC Program Start Mode	8002_H	Set the mode for starting an SFC program.	• Initial Start • Resume start	Initial Start	MELSEC-Q/L/QnA Programming Manual (SFC)
Start Conditions	8003_H	Set the conditions for starting an SFC program.	• Autostart Block 0 • Do not Autostart Block 0	Autostart Block 0	
Output Mode When the Block is Stopped	8006_H	Set the output mode for the case of a block stop.	• Turn OFF • Keep ON	Turn OFF	

(8) Device setting

Number of points, latch range, and local device range are set for each device.



Item	Parameter No.	Description	Setting range	Default	Reference
Device Points *1	2000_H	Set the number of device points that is appropriate to the system.	X, Y, S are fixed to 8K points. Setting is available within the range of 29K words in total, including the above fixed points. One device: Up to 32K points*3	• X: 8K • Y: 8K • M: 8K • L: 8K *2 • B: 8K • F: 2K • SB: 2K • V: 2K • S: 8K • T: 2K • ST: 0K • C: 1K • D: 12K • W: 8K • SW: 2K	Page 88, Section 3.4
Latch (1) Start/Latch (1) End*2	2001_H	Set a latch range (start and end device numbers) to be cleared by a latch clear operation.	Setting is available for only one range for each of B, F, V, T, ST, C, D, and W devices.	—	
Latch (2) Start/End*2	2002_H	Set a latch range (start and end device numbers) not to be cleared by a latch clear operation.	Setting is available for only one range for each of L, B, F, V, T, ST, C, D, and W devices.	—	Page 88, Section 3.4
Local Device Start/End	2003_H	Set a range (start and end device numbers) to be used for a local device.	Setting is available for only one range for each of M, V, T, ST, C, D, and Z devices.	—	

Item		Parameter No.	Description	Setting range	Default	Reference
File Register Extended Setting	Device Points	2000_H	Set points for the file register (ZR), extended data register (D), and extended link register (W).	• Points of the file register (ZR) • Assign part of the file register points to the extended data register and extended link register.	—	Page 88, Section 3.4, Page 322, Section 5.6, Page 326, Section 5.7, Page 330, Section 5.8
	Latch (1) Start/Latch (1) End (latch clear enabled)	2004_H	Set a latch range (start and end device numbers) to be cleared by a latch clear operation.	Each latch range for the file register (ZR), extended data register (D), and extended link register (W).	—	
	Latch (2) Start/End (latch clear disabled)	2005_H	Set a latch range (start and end device numbers) not to be cleared by a latch clear operation.	Each latch range for the file register (ZR), extended data register (D), and extended link register (W).	—	
Indexing Setting for ZR Device	32Bit Indexing	2000_H	Select Z or ZZ device for 32-bit indexing.	Z0 to Z18 (when using device Z)	Use Z	
Pointer Extended Setting for Automatic-Assign Device	Pointer extended setting	1006_H	Set it when using a pointer number of 4096 or later in the Automatic-Assign Device setting.	• no setting • setting available	no setting	Page 336, Section 5.10
	Extension points		Set the extension points.	16 to 28672	28672	

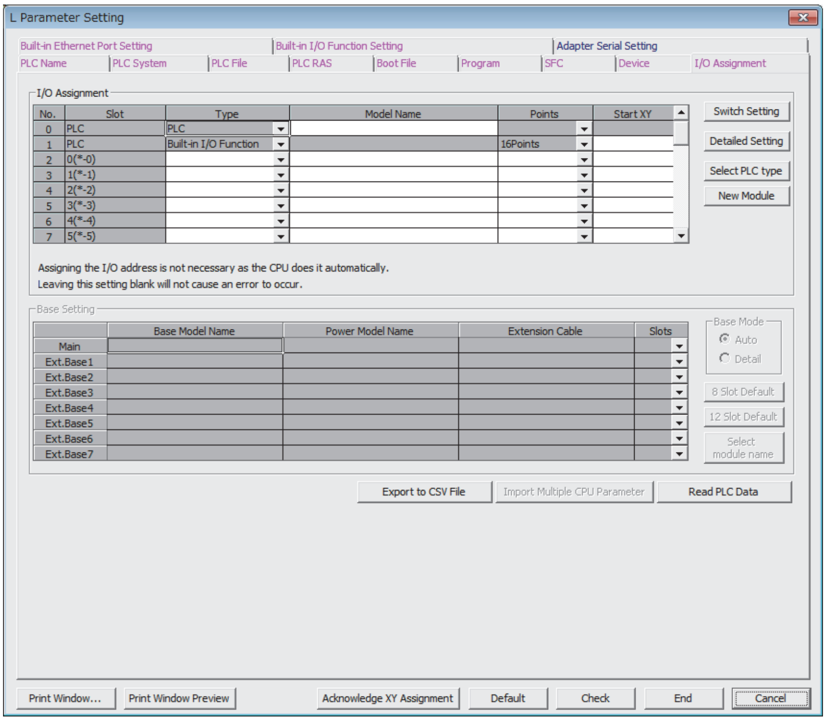
*1 When changing the device points, new setting must not exceed the refresh ranges of network modules or the auto refresh ranges of intelligent function modules. If a new device point setting exceeds the corresponding device range, the data may be written to another device or an error may occur.

*2 When latching a device, consider the increase in the scan time because the scan time increases. (☞ Page 378, Appendix 3.1)

*3 For an internal relay and link relay, the maximum number of points can be set up to 60K.

(9) I/O Assignment setting

The connecting status of each module in the system is set.



Item		Parameter No.	Description	Setting range	Default	Reference
I/O Assignment	Type	0400_H	Set the type of the connected module.	Select from the following. • Empty • Input • Output • I/O Mix • Intelligent • Branch*2 • Branch (for LA1S Extension)*2	Blank	Page 41, Section 2.2
	Model Name		Set the model name of the connected module. (Entered at user's discretion. CPU modules do not use this data.)	Up to 16 characters		
	Points		Set the number of points assigned to each slot.	Select a point of 0, 16, 32, 48, 64, 128, 256, 512, or 1024*1.		
	Start X/Y		Set the start I/O number of each slot.	• L02SCPU, L02SCPU-P, L02CPU, L02CPU-P: 0_H to $3F0_H$ • L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT: 0_H to $FF0_H$		

Item		Parameter No.	Description	Setting range	Default	Reference
Switch Setting		0409_H	Set the switches of the built-in I/O function or intelligent function modules.	—	—	Page 41, Section 2.2
Detailed Setting	Error Time Output Mode	0403_H	Set the output status for the case of a stop error in a CPU module.	Clear/Stop	Clear	
	PLC Operation Mode at H/W Error	4004_H	Sets an operation mode of the CPU module when a hardware error (CPU module detects SP.UNIT DOWN) occurred in an intelligent function module.	Stop/Continue	Stop	
	I/O Response Time	0405_H	Set a response time for the input module or I/O combined module.	Select 1ms, 5ms, 10ms, 20ms, or 70ms.	10ms	
Select PLC type		—	Sets the model of the connected CPU module automatically in the "I/O Assignment" area by selecting the module from the pull-down menu.	—	—	
Select module		—	Sets the model name, points, and start I/O number of the selected module automatically in the "I/O Assignment" area by selecting the module from the pull-down menu.	—	—	

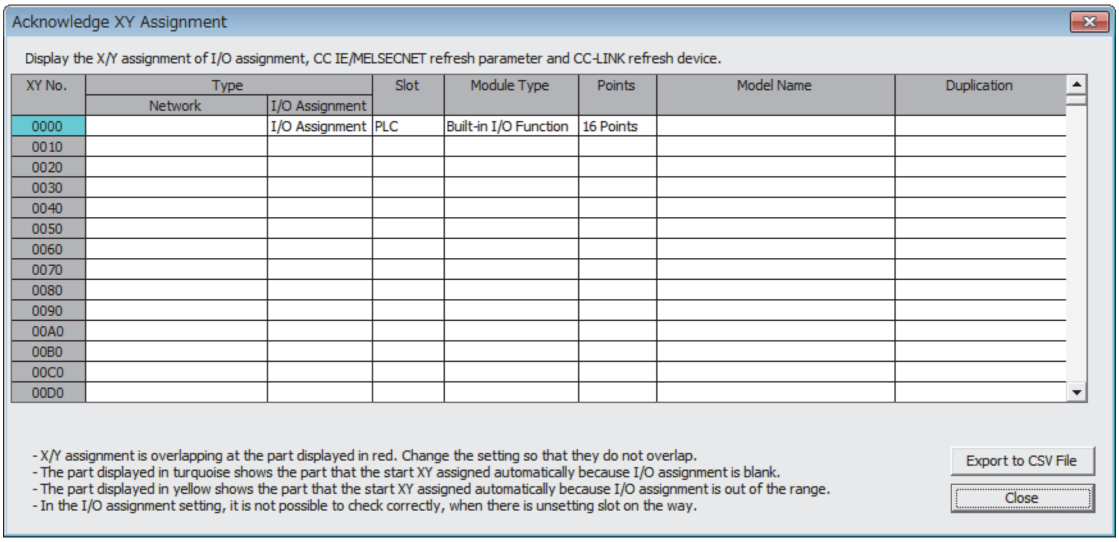
*1 For the L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P, 1024 points cannot be set.

*2 Since the branch module and the branch module (for LA1S Extension) have no I/O points, the number of I/O points and the start X/Y cannot be set.

A

(10) Acknowledge XY Assignment

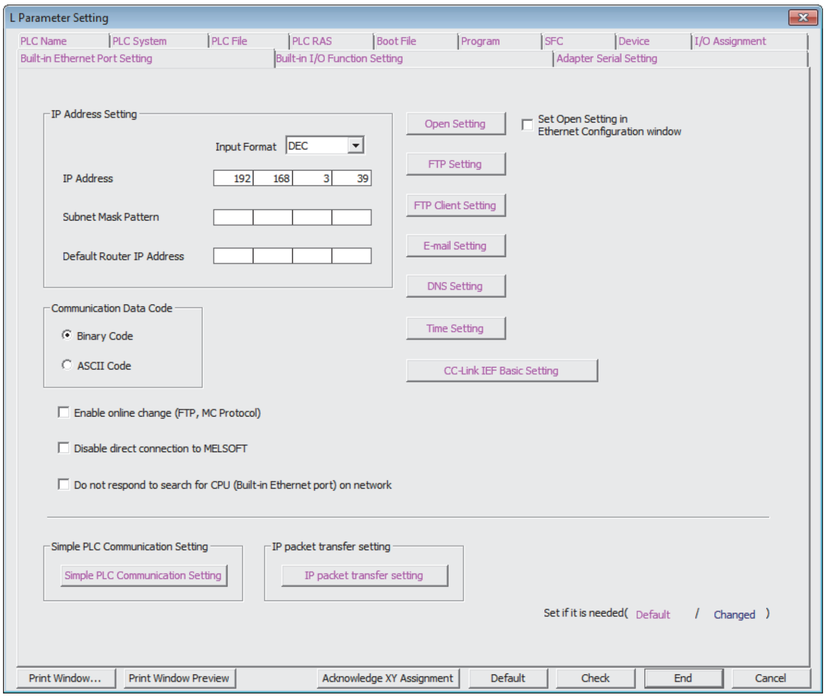
The parameters set in the I/O Assignment tab and CC-Link setting can be confirmed.



Item	Parameter No.	Description	Setting range	Default	Reference
Acknowledge XY Assignment	—	The parameters set in the I/O Assignment tab and CC-Link setting can be confirmed.	—	—	—
Export to CSV File	—	Writes parameters set in this screen to a CSV file.	—	—	—

(11) Built-in Ethernet Port setting

Parameters required for using the built-in Ethernet ports are set.



Remark

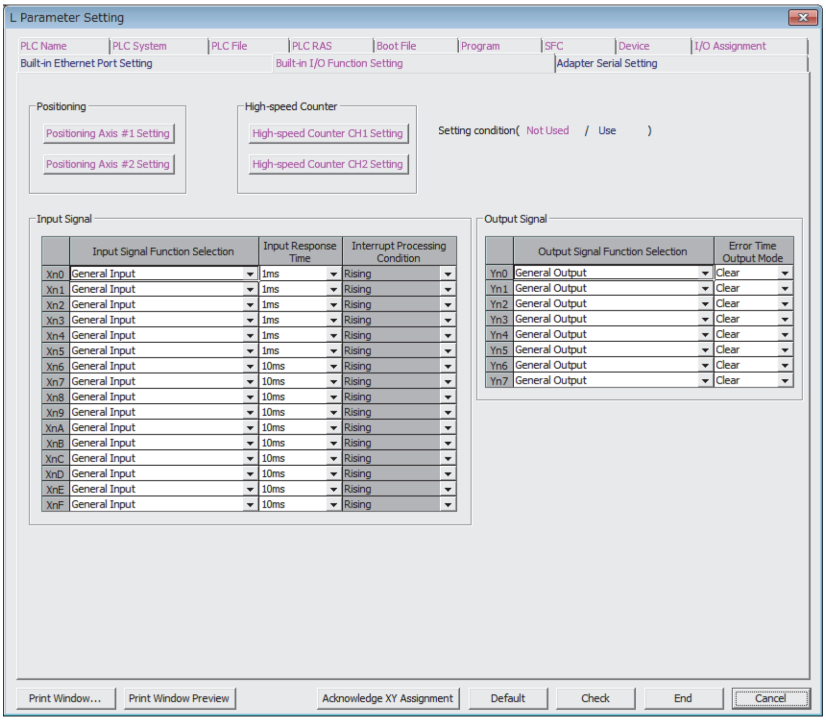
These parameters are not available for the L02SCPU and L02SCPU-P because these modules do not have built-in Ethernet ports.

Item	Parameter No.	Description	Setting range	Default	Reference
IP Address	1016_H	• IP Address: Enter the IP address of the CPU module. • Subnet Mask Pattern: Enter the subnet mask pattern when using a router. • Default Router IP Address: Enter the IP address of the router.	• IP Address: 0.0.0.1 to 223.255.255.254 (00000001_H to $DFFFFFFE_H$) • Subnet Mask Pattern: Blank or 192.0.0.0 to 255.255.255.252 ($C0000000_H$ to $FFFFFFFC_H$) • Default Router IP Address: Blank or 0.0.0.1 to 223.255.255.254 (00000001_H to $DFFFFFFE_H$)	• IP Address: 192.168.3.39 • Subnet Mask Pattern: Blank • Default Router IP Address: Blank	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)

Item		Parameter No.	Description	Setting range	Default	Reference
Communication Data Code		1016_H	Select the code for MC protocol communication.	Binary Code/ASCII Code	Binary Code	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Ethernet Conf. (Open Setting)			Set this parameter when using the following functions. • MC protocol • MELSOFT connection • Socket communications • Predefined protocol • FTP client • E-mail	—	—	
FTP Setting			Set this parameter when using the file transfer function (FTP server).	—	—	
FTP Client Setting		$101D_H$	Set this parameter when using the file transfer function (FTP client).	—	—	
E-mail Setting		$101E_H$	Set this parameter when using the e-mail send/receive function.	—	—	
DNS Setting		1016_H	Set a DNS server address when the destination server is specified by a name not an IP address in the E-mail Setting or FTP Client Setting parameter.	[DNS Server 1 Address] 0.0.0.1 to 223.255.255.254 (00000001_H to $DFFFFFFE_H$) [DNS Server 2 Address] 0.0.0.1 to 223.255.255.254 (00000001_H to $DFFFFFFE_H$)	• DNS Server 1 Address: Blank • DNS Server 2 Address: Blank	
Time setting			Set this parameter when using the time setting function.	—	—	
CC-Link IEF Basic Setting	Network Configuration Setting	1030_H	Set the network configuration when using CC-Link IE Field Network Basic.	—	—	CC-Link IE Field Network Basic Reference Manual
	Refresh Setting	1031_H	Set this parameter to refresh link device data to the internal device or file register automatically.	—	—	
Enable online change (FTP, MC Protocol)		1016_H	Enable or disable writing data in devices or files to the running CPU module when MC protocol or FTP is used.	Selected/deselected	Deselected	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Disable direct connection to MELSOFT			Enable or disable direct connection to MELSOFT. To enhance the security with the remote password setting, select this to disable.	Selected/deselected	Deselected	
Do not respond to search for CPU (Built-in Ethernet port) on network			Disables response to the find CPU function of the MELSOFT connection. To enhance the security, select this.	Selected/deselected	Deselected	
Simple PLC Communication Setting		1019_H	Set parameters when using the simple PLC communication function.	—	—	
IP packet transfer function		1016_H	Set parameters when using the IP packet transfer function.	—	—	

(12) Built-in I/O Function setting

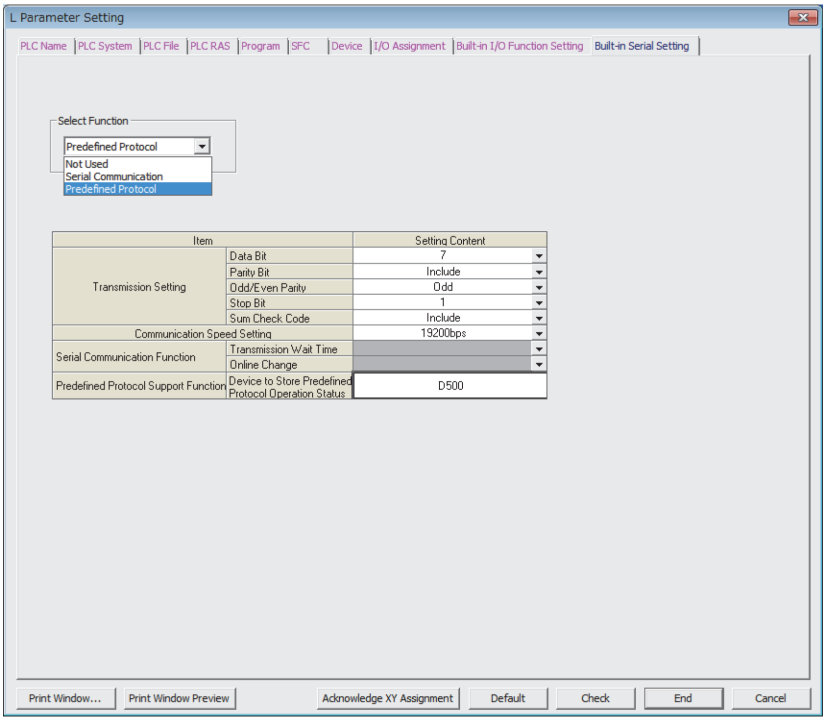
Parameters required for use of the built-in I/O function are set.



Item	Parameter No.	Description	Setting range	Default	Reference
Positioning	$B000_H$	Set whether to use the positioning function and high-speed counter function.	—	—	MELSEC-L CPU Module User's Manual (Built-In I/O Function)
High-speed Counter	$B000_H$	Set whether to use the positioning function and high-speed counter function.	—	—	MELSEC-L CPU Module User's Manual (Built-In I/O Function)
Input Signal	$B001_H$	Assign the functions to the input signals X0 to XF signals.	—	—	MELSEC-L CPU Module User's Manual (Built-In I/O Function)
Output Signal	$B002_H$	Assign the functions to the output signals Y0 to YF.	—	—	MELSEC-L CPU Module User's Manual (Built-In I/O Function)

(13) Built-in Serial Setting

Parameters required for data communications using the predefined protocol function or the serial communication function via the RS-232 interface of the CPU module are set.



Remark

The setting is available only for the L02SCPU and L02SCPU-P.

Item		Parameter No.	Description	Setting range	Default	Reference
Select Function		• For serial communication : $100E_H$ • For predefined protocol: $101B_H$	Select the function to be used.	• Not Used • Serial Communication • Predefined Protocol	Not Used	Page 215, Section 3.37 Page 227, Section 3.38
Not Used	Data Bit		Set the data bit.	• 7 • 8 For the serial communication function, this is fixed to "8".	8	
	Parity Bit		Set the parity bit.	• Not include • Include For the serial communication function, this is fixed to "Include".	Include	
	Odd/Even Parity		Select either odd parity or even parity. (Setting is available only when the parity bit is set to "Include".)	• Odd • Even For the serial communication function, this is fixed to "Odd".	Odd	
	Stop Bit		Set the stop bit.	• 1 • 2 For the serial communication function, this is fixed to "1".	1	

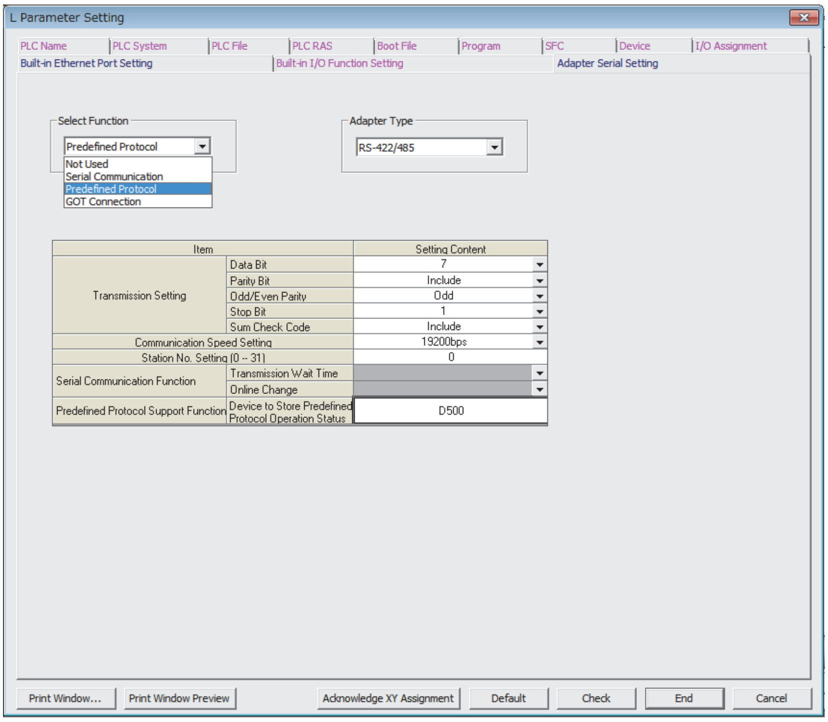
Item		Parameter No.	Description	Setting range	Default	Reference
Transmission Setting	Sum Check Code	• For serial communication : $100E_H$ • For predefined protocol: $101B_H$	Set whether to add sumcheck codes according to the specifications of the protocol to be executed.	• Include • Not include	Include	Page 215, Section 3.37 Page 227, Section 3.38
Communication Speed Setting			Set a communication speed.	9600bps, 19200bps, 38400bps, 57600bps, 115200bps	19200bps	
Transmission Wait Time			Set a transmission wait time when the serial communication function is selected.	• No wait time • 10 to 150ms (in increments of 10ms)	No wait time	
Online Change			Set whether to enable online change when the serial communication function is selected.	• Disable • Enable	Disable	
Device to Store Predefined Protocol Operation Status			Set the start device for storing the predefined protocol operating status when the predefined protocol function is selected. The operating status of the predefined protocol is stored in the area of three points starting from the set device.	• D (including the extended data register) • W (including the extended link register) • R, ZR	Blank	

Point

To communicate data using the predefined protocol function, set protocols for communicating with external devices using the predefined protocol support function of GX Works2.

(14) Adapter Serial Setting

Parameters required for data communications using the predefined protocol function or the serial communication function via the RS-232 adapter or RS-422/485 adapter are set.



Remark

The setting is available only for the LCPU where the RS-232 adapter or RS-422/485 adapter can be mounted.

Item	Parameter No.	Description	Setting range	Default	Reference
Select Function	• For serial communication: $101A_H$ • For predefined protocol: $101B_H$ • For GOT connection: $101C_H$	Select the function to be used.	RS-232 adapter: • Not Used • Serial Communication • Predefined Protocol RS-422/485 adapter: • Not Used • Serial Communication • Predefined Protocol • GOT Connection	Not Used	Page 215, Section 3.37 Page 227, Section 3.38
Adapter Type		Select the type of the adapter used.	• RS-232 • RS-422/485	RS-232	

Item		Parameter No.	Description	Setting range	Default	Reference
Transmission Setting	Data Bit	• For serial communication: $101A_H$ • For predefined protocol: $101B_H$ • For GOT connection: $101C_H$	Set the data bit.	• 7 • 8 For the serial communication function, this is fixed to "8".	8 (When "Predefined Protocol" is selected: "7")	Page 215, Section 3.37 Page 227, Section 3.38
	Parity Bit		Set the parity bit.	• Not include • Include For the serial communication function, this is fixed to "Include".	Include	
	Odd/Even Parity		Select either odd parity or even parity. (Setting is available only when the parity bit is set to "Include".)	• Odd • Even For the serial communication function, this is fixed to "Odd".	Odd	
	Stop Bit		Set the stop bit.	• 1 • 2 For the serial communication function, this is fixed to "1".	1	
	Sum Check Code		Set whether to add sumcheck codes according to the specifications of the protocol to be executed.	• Include • Not include	Include	
Communication Speed Setting			Set a communication speed.	RS-232 adapter: 9600bps, 19200bps, 38400bps, 57600bps, 115200bps RS-422/485 adapter: 1200bps, 2400bps, 4800bps, 9600bps, 19200bps, 38400bps, 57600bps, 115200bpss	19200bps	
Station No. Setting (0 - 31)			Set a station number for multidrop connection when the RS-422/485 adapter is used.	0 to 31	0	
Transmission Wait Time			Set a transmission wait time when the serial communication function is selected.	• No wait time • 10 to 150ms (in increments of 10ms)	No wait time	
Online Change			Set whether to enable online change using the MC protocol while the CPU module is in RUN state when the serial communication function is selected.	• Disable • Enable	Disable	
Device to Store Predefined Protocol Operation Status			Set the start device for storing the predefined protocol operating status when the predefined protocol function is selected. The operating status of the predefined protocol is stored in the area of three points starting from the set device.	• D (including the extended data register) • W (including the extended link register) • R, ZR	Blank	

Point

To communicate data using the predefined protocol function, set protocols for communicating with external devices using the predefined protocol support function of GX Works2.

Appendix 1.3 Network parameters

(1) CC-Link IE Field Network

For the network parameters of CC-Link IE Field Network, refer to the following.

MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual

(2) Ethernet

For the network parameters of Ethernet, refer to the following.

MELSEC-L Ethernet Interface Module User's Manual (Basic)

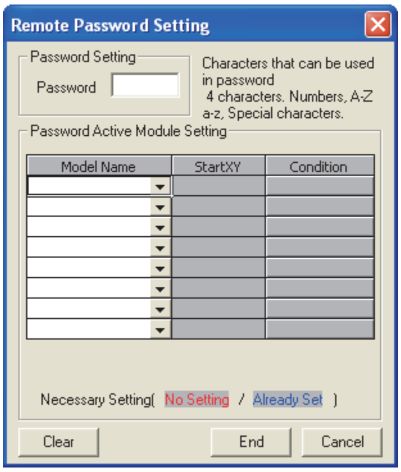
(3) CC-Link

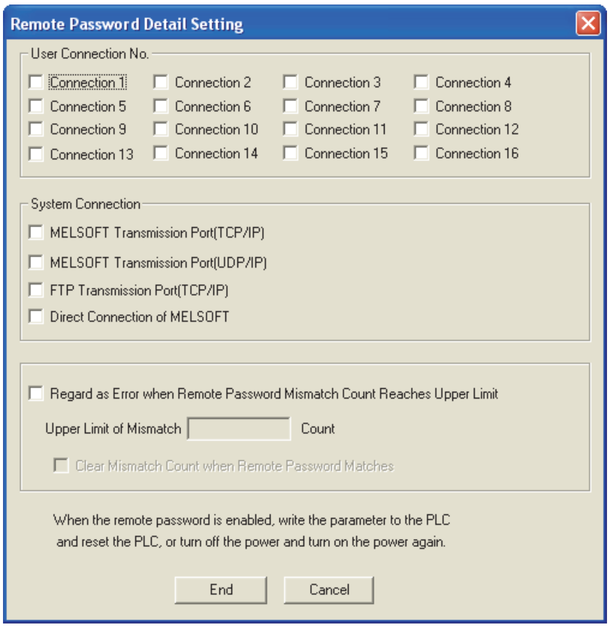
For the network parameters of CC-Link, refer to the following.

MELSEC-L CC-Link System Master/Local Module User's Manual

Appendix 1.4 Remote password setting

This section provides the remote password setting screens and details of the setting items.





<table>
<tr><th colspan="4">Item</th><th>Description</th><th>Setting range</th></tr>
<tr><td colspan="4">Password Setting</td><td>Enter a remote password.</td><td>Up to four characters (alphanumeric characters, special symbols)</td></tr>
<tr><td rowspan="7">Password Active Module Setting</td><td colspan="3">Model Name</td><td>Select a model name of the CPU module.</td><td>• Ethernet_Built-in_CPU
• LJ71C24
• LJ71E71-100</td></tr>
<tr><td colspan="3">Start X/Y</td><td>Set the start address of the module.</td><td>0000_H to $0FE0_H$</td></tr>
<tr><td rowspan="5">Condition (detailed setting)</td><td colspan="2">—</td><td>Configure the detailed setting when selecting the LCPU.</td><td>—</td></tr>
<tr><td colspan="2">User Connection No.*1</td><td>Select a user connection number(s).</td><td>Connection 1 to Connection 16</td></tr>
<tr><td colspan="2">System Connection*2*3</td><td>Select a port(s) where the remote password is enabled.</td><td>• MELSOFT Transmission Port (TCP/IP)
• MELSOFT Transmission Port (UDP/IP)
• FTP Transmission port (TCP/IP)
• Direct Connection of MELSOFT</td></tr>
<tr><td rowspan="2">Regard as Error when Remote Password Mismatch Count Reaches Upper Limit</td><td>—</td><td>Set whether to generate an error when the number of mismatches on remote password reaches to the upper limit.</td><td>Deselected</td></tr>
<tr><td>Upper Limit of Mismatch</td><td>Set the upper limit of the number of remote password mismatches.</td><td>1 to 65535 (default: 10 (Count))</td></tr>
</table>

*1 This is a connection used by users for communications using the MC protocol or fixed buffer.
*2 This is a connection used by a system for FTP or MELSOFT (TCP/IP, UDP/IP) communications.
*3 For details on the LJ71E71-100 settings, refer to the following.
MELSEC-L Ethernet Interface Module User's Manual (Basic)

Point

After setting a remote password, store the parameters in the parameter-valid drive.

Appendix 2 Added and Changed Functions

Some functions are added to the CPU module and GX Works2. The following table shows serial numbers of the CPU module and software versions of GX Works2 that support those added functions.

Added function	Serial No. (first 5 digits) of CPU module	GX Works2 version	Reference
Parameter-valid drive information	—	1.34L or later	Page 34, Section 2.1.2
Data logging file transfer function*1	"12112" or later	1.45X or later	QnUDVCPU/LCPU User's Manual (Data Logging Function)
CC-Link IE Field Network	"13012" or later	1.53F or later	Page 317, Section 5.4 MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual
Extension of available index register range (Z0 to Z19) when Jn and Un are used in the dedicated instruction		—	Manuals for the network module and the intelligent function module used
Simple PLC communication function*1	"13042" or later	1.62Q or later	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Memory check function (storage of device memory error information and program error location)	"14012" or later	—	MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)
Program cache memory auto recovery function			Page 200, Section 3.35
Monitor condition setting	—	1.73B or later	Page 123, Section 3.16
Project data batch save/load function*1	"14042" or later	—	Page 202, Section 3.36
SD memory card operation using a display unit*1			Page 274, Section 4.7
Own station number setting function (CC-Link IE Field Network module)		1.87R or later	MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual
Writing/reading data to/from refresh devices with the specified station number	"14072" or later	—	MELSEC-Q/L Programming Manual (Common Instruction)
Ethernet module support	"14112" or later	1.98C or later	MELSEC-L Ethernet Interface Module User's Manual (Basic)
IP packet transfer function*1*2			MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Serial communication function (RS-232 interface of the CPU module)*1			Page 227, Section 3.38
Support of the iQ Sensor Solution function (data backup/restoration only) for AnyWireASLINK and CC-Link*1		1.492N or later	Page 234, Section 3.39, iQ Sensor Solution Reference Manual
Latch clear by using the special relay and special register areas	"15042" or later	—	Page 89, Section 3.4 (4) (a)
Support of the iQ Sensor Solution function (automatic detection of connected device, system configuration check, communication setting reflection, sensor parameter read/write, and monitoring) for built-in Ethernet*1	"15043" or later	1.492N or later	Page 234, Section 3.39, iQ Sensor Solution Reference Manual
Support of the iQ Sensor Solution function (data backup/restoration only) for built-in Ethernet*1	"15072" or later	1.497T or later	Page 234, Section 3.39, iQ Sensor Solution Reference Manual

Added function	Serial No. (first 5 digits) of CPU module	GX Works2 version	Reference
Predefined protocol function	"15102" or later	1.501X or later	Page 215, Section 3.37
Serial communication function (RS-232 adapter, RS-422/485 adapter)*1			Page 227, Section 3.38
Operation mode setting at double block START (SFC)*1			MELSEC-Q/L/QnA Programming Manual (SFC)
Increase in the number of steps (SFC)*1			
1K point setting for the step relay (S)			
Step comment readout instruction (S(P).SFCSCOMR), transition condition comment readout instruction (S(P).SFCTCOMR)*1		—	
Online change (inactive blocks) (SFC)*1			
Storing MAC address in the special register, setting and storing IP address in the special register*1			MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)
RS-422/485 adapter*1		1.501X or later	
Simple PLC communication function (for MELSEC-F series FX3)*1	"16042" or later	1.513K or later	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Pointer points extension compatibility for the Auto assignment device*1			Page 362, Appendix 1.2 (8)
Support of the iQ Sensor Solution function (data backup/restoration only) for the CC-Link-AnyWireASLINK bridge module (NZ2AW1C2AL)*1			Page 234, Section 3.39, iQ Sensor Solution Reference Manual
Real-time monitor function*1	"16072" or later	*3	Page 150, Section 3.21, GX LogViewer Version 1 Operating Manual
Support of the iQ Sensor Solution function (data backup/restoration only) for the CC-Link IE Field Network module*1		—	Page 234, Section 3.39, iQ Sensor Solution Reference Manual
LA1S extension base unit*1	"16112" or later	1.525X or later	MELSEC-L LA1S Extension Base Unit User's Manual
File transfer function (FTP client)*1			MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
E-mail send/receive function*1			
CC-Link IE Field Network Basic function*1	"18112" or later	1.555D or later	CC-Link IE Field Network Basic Reference Manual
SLMP frame send instruction*1		—	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Simple PLC communication function (for MELSEC iQ-R series and SLMP-compatible devices (QnA-compatible 3E frame))*1	"20042" or later	1.575Z or later	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Simple PLC communication function (for MELSEC iQ-F series)*1	"20102" or later	1.580E or later	MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)

—: Function that is not related to serial No. or software version

*1 Some models do not support these functions. For the availability, refer to the corresponding section or manual of each function.

*2 For the versions of the intelligent function modules that support the function, refer to the manual for the intelligent function module used.

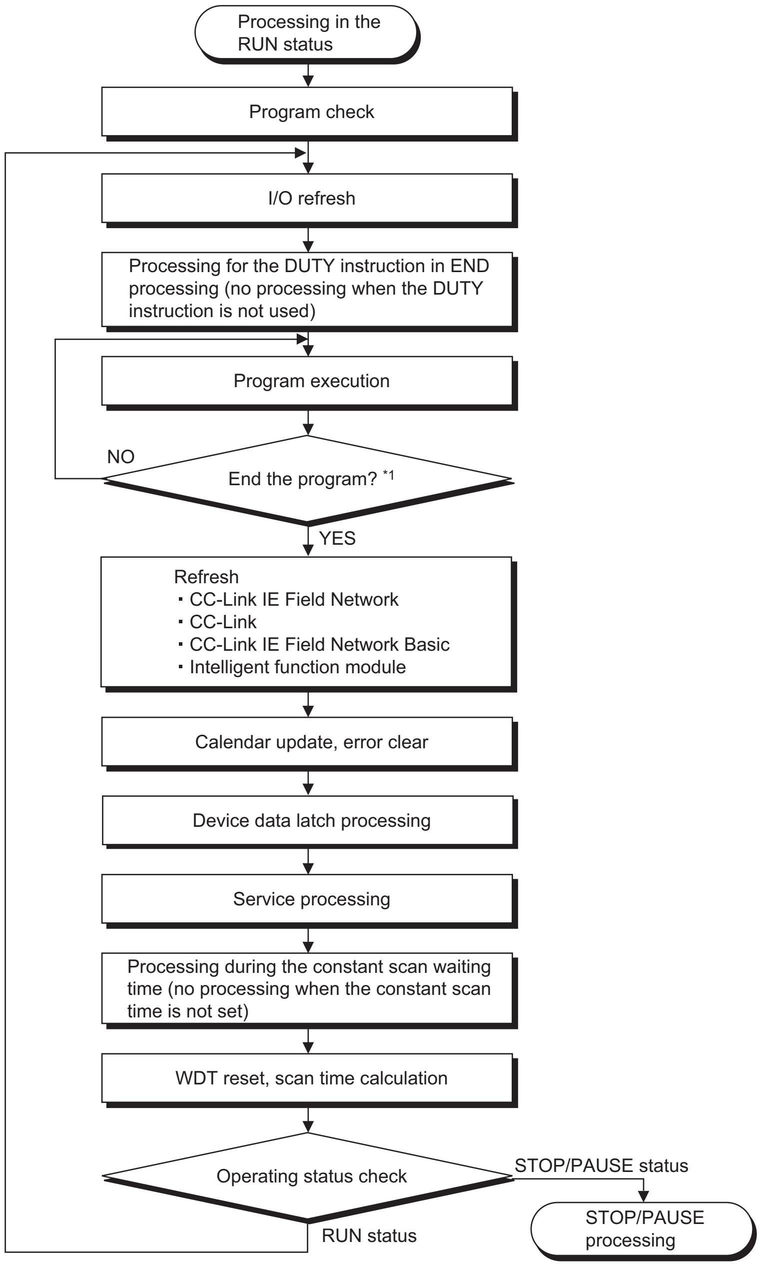
*3 GX LogViewer with a software version of 1.37P or later is required.

Appendix 3 CPU Module Processing Time

This section describes the CPU module processing time.

Appendix 3.1 Time required for each processing in a scan time

This section describes operations performed in a scan time and how to calculate the time required for each processing.[*2]



*1 End of a program indicates the timing when the END, GOEND, FEND, or STOP instruction is executed.

*2 For details, refer to ☞ Page 379, Appendix 3.1 (1) to ☞ Page 384, Appendix 3.1 (7).

The CPU module measures current, minimum, and maximum values of the scan time. The scan time can be checked by monitoring SD520, SD521, and SD524 to SD527, as the scan time is stored in the special register. (Accuracy: ± 0.1ms)

Current value	SD520	SD521
Minimum value	SD524	SD525
Maximum value	SD526	SD527

→ Stores the scan time of 1ms or less. (unit: μs)
→ Stores the scan time. (in increments of 1ms)

Ex. When the stored values in SD520 and SD521 are 3 and 400 respectively, the scan time is 3.4ms.

(1) I/O refresh time

Use the following expression to calculate the I/O refresh time. Note that the I/O refresh time for the built-in special functions is excluded.

[I/O refresh time]

(Number of input refresh points*1 × KM1) + (number of output refresh points*2 × KM2) [μs]

*1 The number of input refresh points is the value obtained by dividing the number of input points by 16/h.
*2 The number of output refresh points is the value obtained by dividing the number of output points by 16/h.

CPU module	When a CPU module is connected to a main block		When a CPU module is connected to an extension block		When a CPU module is connected to an extension base unit	
	KM1	KM2	KM1	KM2	KM1	KM2
L02SCPU, L02SCPU-P	1.8	1.1	2.6	1.9	4.9	4.0
L02CPU, L02CPU-P	1.5	1.0	2.3	1.8	4.3	3.9
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	1.5	1.0	2.3	1.8	4.3	3.9

(2) Processing time for the instruction (DUTY instruction) in END processing

Using the DUTY instruction increases the END processing time because the user timing clock (SM420 to SM424 and SM430 to SM434) specified with the instruction is turned on or off in END processing.

CPU module	Processing time in END processing	
	When set to 1	When set to 5
L02SCPU, L02SCPU-P	12.0 [μs]	14.0 [μs]
L02CPU, L02CPU-P	6.9 [μs]	7.4 [μs]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	5.7 [μs]	6.1 [μs]

(3) Instruction execution time

The instruction execution time is a total of the time required for all instructions used in the program to be executed and (a) and (B) described in the following page. For execution time of instructions, refer to the following.
MELSEC-Q/L Programming Manual (Common Instruction)

(a) Overhead time at execution of interrupt and fixed scan execution type programs

When using an interrupt or fixed scan execution type program, add the overhead time given in the following table. Two kinds of overhead time (pre-start and program-end) need to be added to interrupt programs.

CPU module	Pre-start overhead time for interrupt programs					
	Interrupt from the built-in I/O (I0 to I15)		Fixed scan interrupt (I28 to I31)		Interrupt from the intelligent function module (I50 to I127)	
	Without high-speed start	With high-speed start	Without high-speed start	With high-speed start	Without high-speed start	With high-speed start
L02SCPU, L02SCPU-P	65.0 [μs]	31.0 [μs]	55.0 [μs]	35.0 [μs]	76.0 [μs]	55.0 [μs]
L02CPU, L02CPU-P	57.0 [μs]	19.0 [μs]	51.0 [μs]	17.0 [μs]	66.0 [μs]	31.0 [μs]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	54.0 [μs]	18.0 [μs]	46.0 [μs]	16.0 [μs]	61.0 [μs]	26.0 [μs]

CPU module	Program-end overhead time for interrupt programs	
	Without high-speed start	With high-speed start
L02SCPU, L02SCPU-P	28.0 [μs]	15.0 [μs]
L02CPU, L02CPU-P	28.0 [μs]	9.0 [μs]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	26.0 [μs]	8.5 [μs]

CPU module	Overhead time for fixed scan execution type programs	
	Without high-speed start	With high-speed start
L02SCPU, L02SCPU-P	92.0 [μs]	60.0 [μs]
L02CPU, L02CPU-P	77.0 [μs]	25.0 [μs]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	72.0 [μs]	23.0 [μs]

(b) Overhead time when local devices in the interrupt program are enabled

Add the overhead time given in the following table when use of local devices is enabled in the interrupt program by turning on SM777 (Enable/disable local device in interrupt program).

[Pre-start overhead time for interrupt programs]
$(N1 \times KM1) + ((N2 + (N3 \div 16)) \times KM2) + KM3$ [μs]

[Overhead time after interrupt program activation]
$(N1 \times KM4) + ((N2 + (N3 \div 16)) \times KM5) + KM6$ [μs]

- N1: Number of devices that is specified as a local device (The index register is excluded.)
- N2: Number of word device points (The index register is excluded.)
- N3: Number of bit device points

CPU module	KM1	KM2	KM3	KM4	KM5	KM6
L02SCPU, L02SCPU-P	13.2	0.22	210.0	8.0	0.23	30.0
L02CPU, L02CPU-P	8.0	0.22	90.0	8.0	0.22	30.0
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	8.0	0.10	80.0	8.0	0.10	20.0

(4) Refresh time

Refresh time is the total time required for refresh of each network and auto refresh of the intelligent function module.

(a) Refresh time of CC-Link IE Field Network

This is the time required for refresh of CC-Link IE Field Network.

📖MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual

(b) Auto refresh time of CC-Link

This is the time required for auto refresh of CC-Link.

📖MELSEC-L CC-Link System Master/Local Module User's Manual

(c) Link refresh time of CC-Link IE Field Network Basic

This is the time required for refresh of CC-Link IE Field Network Basic.

[Refresh time of CC-Link IE Field Network Basic]

$\alpha T = KM1 + KM2 \times (((RX + RY) \div 16) + RWw + RWr) + \alpha E$ [μs]

$\alpha E = KM3 + KM4 \times (((RX + RY) \div 16) + RWw + RWr)$ [μs]

- αT: Link refresh time
- αE: Link refresh time when the file register (R, ZR) is used*1
- RX: Number of points of remote input (RX) refreshed by the master station*2
- RY: Number of points of remote output (RY) refreshed by the master station*2
- RWw: Number of points of remote register (RWw) refreshed by the master station*2
- RWr: Number of points of remote register (RWr) refreshed by the master station*2
- KM1 to KM4: Constant

CPU module	KM1	KM2	KM3	KM4
L02CPU, L02CPU-P	142.0	0.28	10.7	0.32
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	120.0	0.21	8.5	0.23

*1 This time is added when the file register (R, ZR) is used.

*2 This value is determined according to the number of slave stations to be connected and the number of occupied stations.

(d) Auto refresh time via intelligent function modules

This is the time required for refreshing data between the buffer memory of an intelligent function module and devices in the CPU module.

[Auto refresh time via intelligent function modules]

$M1 + KM2 \times$ (number of refresh points) [μs]

CPU module	When a CPU module is connected to a main block		When a CPU module is connected to an extension block	
	KM1	KM2	KM1	KM2
L02SCPU, L02SCPU-P	96.3	6.7	79.7	8.1
L02CPU, L02CPU-P	6.0	5.0	7.0	6.0
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	4.0	5.0	5.0	6.0

(5) Function execution time in END processing

This is the time required for updating calendar or clearing error in END processing.

(a) Calendar update processing time

When SM210 (Clock data set request) turned from off to on or when SM213 (Clock data read request) turned on, the processing time for changing or reading the clock data are required in END processing.

CPU module	Processing time	
	When the clock data set request is issued	When the clock data read request is issued
L02SCPU, L02SCPU-P	0.053 [ms]	0.017 [ms]
L02CPU, L02CPU-P	0.025 [ms]	0.006 [ms]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.018 [ms]	0.006 [ms]

(b) Error clear processing time

Upon the rising edge of SM50 (Error reset), the processing time for clearing the continuation error stored in SD50 is required

CPU module	Processing time	
	When the annunciator is cleared	When the error is cleared
L02SCPU, L02SCPU-P	0.185 [ms]	0.180 [ms]
L02CPU, L02CPU-P	0.101 [ms]	0.098 [ms]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.075 [ms]	0.072 [ms]

(c) Error clear processing time by error type

The processing time for clearing the continuation error by error time is required.

CPU module	Processing time	
	When the annunciator is cleared	When the error is cleared
L02SCPU, L02SCPU-P	0.205 [ms]	0.195 [ms]
L02CPU, L02CPU-P	0.117 [ms]	0.111 [ms]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.090 [ms]	0.087 [ms]

(d) Device data latch processing time

When the latch range is set, the processing time shown below is required. [*1]

[Device data latch processing time]
$(N1 \times KM1) + ((N2 \div 16 + N3) \times KM2)$ [μs]

- N1: Number of device types specified to be latched[*2]
- N2: Number of bit device points specified to be latched[*3]
- N3: Number of word device points specified to be latched[*3]

CPU module	KM1	KM2
L02SCPU, L02SCPU-P	4.4	0.12
L02CPU, L02CPU-P	3.0	0.12
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	3.0	0.05

*1 The scan time does not increase even if the latch range is set for the file register (R, ZR), extended data register (D), or extended link register (W).
*2 Count the latch range (1) and the latch range (2) as different device types.
*3 The scan time will not increase if the latch range is set for the file register (ZR), extended data register (D), or extended link register (W).

Point

To reduce the scan time increase due to latch[*1], minimize the number of latch points (latch (1) setting, latch (2) setting, and latch relay) as much as possible by performing the following.

- Move data to be latched to the file register.
- Store device data that is less frequently updated in the standard ROM with the SP.DEVST instruction. (The device data stored in the standard ROM can be read with the S(P).DEVLD instruction. (☞ Page 180, Section 3.29)

*1 For file registers (including an extended data register (D) and an extended link register (W)), the scan time is not increased due to latch.

(6) Service processing time

The following shows the service processing time required for each communication function when the number of processing is set to one.

CPU module	Processing time				
	Program read (when a 10K-step program file is read)	Online change in the ladder mode (when adding 100 steps to the start of an 8K-step program)	Online change in the ladder mode (when adding 100 steps to the start of a 20K-step program)	Online change (file) (when a 10K-step program file is read)	Device data Monitor (Data register: 32 points)
L02SCPU, L02SCPU-P	3.70 [ms]	2.45 [ms]	—	7.00 [ms]	2.65 [ms]
L02CPU, L02CPU-P	0.95 [ms]	1.20 [ms]	—	4.40 [ms]	0.50 [ms]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.95 [ms]	—	1.00 [ms]	3.70 [ms]	0.40 [ms]

(7) Common processing time

The CPU module performs common processing by the system. The common processing time shown below is required.

CPU module	Processing time
L02SCPU, L02SCPU-P	0.28 [ms]
L02CPU, L02CPU-P	0.22 [ms]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.18 [ms]*1

*1 Processing time when the setting of the built-in CC-Link is disabled

Appendix 3.2 Factors that increase the scan time

When any of the function or operation described below is executed, the scan time is increased by the amount of its processing time.

(1) Batch transfer to the program memory

Use the following expression to calculate the increased scan time when data are batch transferred to the program memory.

[Time required for batch-transferring data to the program memory]
Scan time $\times$ KM1 + KM2 [s]

CPU module	KM1	KM2
L02SCPU, L02SCPU-P	120.0	1.2
L02CPU, L02CPU-P	170.0	1.0
L06CPU, L06CPU-P	260.0	4.7
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	1100.0	15.0

(2) Use of local devices

Use the following expression to calculate the increased scan time when local devices are used.

[Increase in the scan time]
$((N1 \times KM1) + ((N2 + N3 \div 16)) \times KM2) + (N4 \times KM3) + KM4) \times n + KM5$ [μs]

- N1: Number of devices that is specified as a local device
- N2: Number of word device points
- N3: Number of bit device points
- N4: Number of index register points that were specified as a local device

CPU module	KM1	KM2	KM3	KM4	KM5
L02SCPU, L02SCPU-P	16.0	0.23	1.49	98.3	92.0
L02CPU, L02CPU-P	8.0	0.22	0.65	58.0	14.2
L06CPU, L06CPU-P	8.0	0.10	0.47	35.5	12.7
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	8.0	0.10	0.68	35.5	17.3

Use the following expression to calculate the increase in scan time when local devices are used in a subroutine program (only when the subroutine program is called from another file).

[Increase in the scan time]
$(N1 \times KM1) + ((N2 + (N3 \div 16)) \times KM2) + (N4 \times KM3) + KM4$ [μs]

- N1: Number of devices that is specified as a local device
- N2: Number of word device points
- N3: Number of bit device points
- N4: Number of index register points that were specified as a local device

CPU module	KM1	KM2	KM3	KM4
L02SCPU, L02SCPU-P	20.3	0.76	4.47	257
L02CPU, L02CPU-P	16.0	0.44	1.30	80.0
L06CPU, L06CPU-P	16.0	0.20	0.94	100.0
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	16.0	0.20	1.36	100.0

(3) Execution of multiple programs

Use the following formula to calculate the increase in scan time when executing multiple programs.

[Increase in the scan time]

Number of program files to be executed × KM1 [ms]

CPU module	KM1
L02SCPU, L02SCPU-P	0.053
L02CPU, L02CPU-P	0.024
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.02

(4) Removal and insertion of an SD memory card

The following shows the increase in scan time when an SD memory card is inserted or removed.

CPU module	Increase in scan time	
	When an SD memory card is inserted	When an SD memory card is removed
L02CPU, L02CPU-P	0.89 [ms]	0.49 [ms]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	0.82 [ms]	0.42 [ms]

(5) Scan time measurement

Use the following expression to calculate the increase in scan time when the scan time is set to be measured.

[Increase in the scan time]

KM1 + number of branch instructions*1 × KM2 [μs]

CPU module	KM1	KM2
L02SCPU, L02SCPU-P	179.5	5.8
L02CPU, L02CPU-P	40.0	1.7
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	40.0	1.5

*1 The number of branch instructions is a total of the following pointer branch instructions and structure creation instructions, which are executed between a start point and an end point of the scan time measurement.
· Pointer branch instructions: CJ, SCJ, JMP, GOEND
· Structure creation instruction: CALL(P), FCALL(P), ECALL(P), EFCALL(P), XCALL(P), BREAK, NEXT, RET

(6) A series CPU module compatibility setting

The following shows the increase in scan time when an A series CPU module compatibility setting is enabled.

CPU module	Increase in scan time
L02SCPU, L02SCPU-P	95 [μs]
L02CPU, L02CPU-P	40 [μs]
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	34 [μs]

(7) Time taken to collect module errors

Use the following formula to calculate the increase in scan time when the module error collection function is used.

[Increase in the scan time]
KM1 + (N1 × KM2) + (N2 × KM3) [μs]

- N1: Number of errors collected per scan by the built-in CC-Link or external modules
- N2: Number of errors collected by the built-in I/O

CPU module	When a CPU module is connected to a main block			When a CPU module is connected to an extension block		
	KM1	KM2	KM3	KM1	KM2	KM3
L02SCPU, L02SCPU-P	145	120	80	190	140	70
L02CPU, L02CPU-P	28	80	60	28	105	50
L06CPU, L06CPU-P, L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	21	70	50	21	100	40

(8) Data logging function

For the increase in scan time when using the data logging function, refer to the following.
QnUDVCPU/LCPU User's Manual (Data Logging Function)

(9) Realtime monitor function

Use the following formula to calculate the increase in scan time on the realtime monitor function.

[Increase in the scan time]
KM1 + (KM2 × N1) + (KM3 × N2) [μs]

- N1: Number of word devices
- N2: Number of bit devices
- KM1: Constant (common processing time)
- KM2: Constant (word device processing time)
- KM3: Constant (bit device processing time)

Internal user device (X, Y, M, L, B, F, SB, V, T, ST, C, D, W, SW, FX, FY, FD, SM, SD, BL□\S)

CPU module	KM1	KM2	KM3
L02CPU, L02CPU-P	79.00	0.68	0.95
L06CPU, L06CPU-P	74.00	0.60	0.87
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	64.00	0.56	0.75

File register (R, ZR)

CPU module	KM1	KM2	KM3
L02CPU, L02CPU-P	79.00	0.78	1.11
L06CPU, L06CPU-P	74.00	0.70	1.03
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	64.00	0.66	0.91

Module access device (U□\G),
Link direct device(J□\W, J□\X, J□\SW, J□\Y, J□\SB)

CPU module	KM1	KM2	KM3
L02CPU, L02CPU-P	128.10	14.50	15.06
L06CPU, L06CPU-P	125.12	14.20	15.00
L26CPU, L26CPU-P, L26CPU-BT, L26CPU-PBT	112.00	12.00	13.50

Appendix 3.3 Realtime monitor function processing time

This is the minimum interval for monitoring data without any loss under the following conditions during the execution of the realtime monitor function.

- Scan time = 1.5ms (up to 3ms)*1
- Timing = Select "Step No. Specification" as timing condition and set conditions.*2
- Buffer capacity = 32K bytes
- Fixed scan interrupt timing = Start step number of the program set as a condition under "Step No. Specification"

*1 The average scan time shall be 1.0ms when the realtime monitor function is not executed.
*2 Program name: Any, Step No.: Start step of the program, Execution condition: "Always"

Cable type	Number of word device points	Monitor interval
USB	1 or 8	0.5ms
	16	1.0ms
Ethernet	1	0.5ms
	8 or 16	1.0ms
RS-232	1	1.50ms
	8	3.50ms
	16	5.50ms

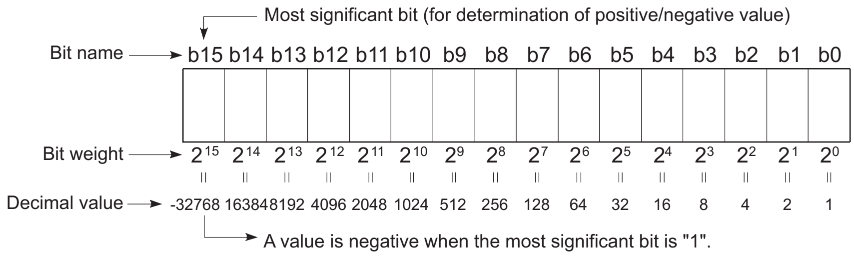
Appendix 4 Data Used in the CPU Module

A

In the CPU module, data such as numeric values and alphabets are represented as a series of bits, "0" and "1". This notation is called BIN (binary).
In addition, DEC (decimal), HEX (hexadecimal), BCD (binary-coded decimal), and floating-point data are available.

(1) BIN (binary code)

BIN is a data notation where a value is represented as a series of bits "0" and "1". Each place digit is shifted to the higher digit when the value exceeds 1.
The value ranged from -32768 to 32767 can be stored in the CPU module.



(2) DEC (decimal)

DEC represents binary data of the CPU module in decimal notation. (☞ Page 342, Section 6.2)

(3) HEX (hexadecimal)

HEX represents binary data of the CPU module in hexadecimal notation. (☞ Page 343, Section 6.3)
In hexadecimal notation, four bits are handled as one digit. Because the place value in a digit is from 0 to 15 in decimal notation, a value larger than nine is represented by A, B, C, D, E, or F. When the value is larger than F, the digit is shifted to the higher digit.
This notation is useful when using 16-bit devices such as the data register and link register.

(4) BCD (binary-coded decimal)

BCD is a data notation that represents a value in the ones place of a decimal number in a binary number. As with hexadecimal notation, data are represented in four bits but the alphabets A to F are not used. This notation is useful when handling data from external devices, such as digital switches.

(5) Floating-point data

In floating point data, real number data are handled as a numeric value with the decimal point. The single-precision floating-point data and double-precision floating-point data are available. (☞ Page 343, Section 6.4)

(6) Numeric representation list

The following table shows the numeric representation of BIN (binary), DEC (decimal), HEX (hexadecimal), and BCD (binary-coded decimal).

BIN (binary)				DEC (decimal)	HEX (hexadecimal)	BCD (binary-coded decimal)			
			0	0	0				0
			1	1	1				1
			10	2	2				10
			11	3	3				11
			.	4	4				.
			.	5	5				.
			.	6	6				.
			.	7	7				.
			.	8	8				.
			1001	9	9				1001
			1010	10	A			1	0000
			1011	11	B			1	0001
			1100	12	C			1	0010
			1101	13	D			1	0011
			1110	14	E			1	0100
			1111	15	F			1	0101
		1	0000	16	10			1	0110
		1	0001	17	11			1	0111
			. . . . .	. . . . .	. . . . .				. . . . .
		10	1111	47	2F			100	0111
				. . . .	. . . .				
0111	1111	1111	1110	32766	7FFE	—			
0111	1111	1111	1111	32767	7FFF	—			
1000	0000	0000	0000	-32768	8000	1000	0000	0000	0000
1000	0000	0000	0001	-32767	8001	1000	0000	0000	0001
. . .	. . .	. . .	. . .	. . .	. . .				
1111	1111	1111	1110	-2	FFFE	—			
1111	1111	1111	1111	-1	FFFF	—			

Appendix 5 Character Codes Available in the Display Unit

A

The following chart shows character codes available in the display unit.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000$_H$	NULL															
0010$_H$																
0020$_H$		!	″	#	$	%	&	’	(	)	*	+	,	−	.	/
0030$_H$	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040$_H$	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050$_H$	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
0060$_H$	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070$_H$	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
0080$_H$*1															‶	
00A0$_H$		｡	｢	｣	､	･	ｦ	ｧ	ｨ	ｩ	ｪ	ｫ	ｬ	ｭ	ｮ	ｯ
00B0$_H$	ｰ	ｱ	ｲ	ｳ	ｴ	ｵ	ｶ	ｷ	ｸ	ｹ	ｺ	ｻ	ｼ	ｽ	ｾ	ｿ
00C0$_H$	ﾀ	ﾁ	ﾂ	ﾃ	ﾄ	ﾅ	ﾆ	ﾇ	ﾈ	ﾉ	ﾊ	ﾋ	ﾌ	ﾍ	ﾎ	ﾏ
00D0$_H$	ﾐ	ﾑ	ﾒ	ﾓ	ﾔ	ﾕ	ﾖ	ﾗ	ﾘ	ﾙ	ﾚ	ﾛ	ﾜ	ﾝ	ﾞ	ﾟ
00F0$_H$*1																

*1 "0081$_H$ to 009F$_H$" and "00E0$_H$ to 00FC$_H$" are recognized as upper bytes of the Shift-JIS code.

The [shaded box] parts in the tables on the following pages are out of the Shift-JIS code range. The Shift-JIS code range is "81_H to $9F_H$" and "$E0_H$ to FC_H" for upper bytes, and "40_H to $7E_H$" and "80_H to FC_H" for lower bytes.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
8140:		、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨	＾
8150:	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／	＼
8160:	～	∥	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］	｛
8170:	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×	
8180:	÷	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥
8190:	＄	￠	￡	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇	◆
81A0:	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	〓			
81B0:									∈	∋	⊆	⊇	⊂	⊃	∪	∩
81C0:									∧	∨	￢	⇒	⇔	∀	∃	
81D0:											∠	⊥	⌒	∂	∇	≡
81E0:	≒	≪	≫	√	∽	∝	∵	∫	∬							
81F0:	Å	‰	♯	♭	♪	†	‡	¶					◯			
8240:																０
8250:	１	２	３	４	５	６	７	８	９							
8260:	Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ	Ｐ
8270:	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ						
8280:		ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ
8290:	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ					ぁ
82A0:	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く	ぐ	け
82B0:	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た	だ	ち
82C0:	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は	ば	ぱ
82D0:	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み	む	め
82E0:	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ	ゐ	ゑ
82F0:	を	ん														
8340:	ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク	グ
8350:	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ	ダ
8360:	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ
8370:	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ	
8380:	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
8390:	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ									Α
83A0:	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ
83B0:	Σ	Τ	Υ	Φ	Χ	Ψ	Ω									α
83C0:	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ
83D0:	σ	τ	υ	φ	χ	ψ	ω									
83E0:																
83F0:																
8440:	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
8450:	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю
8460:	Я															
8470:	а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н	
8480:	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
8490:	ю	я														─
84A0:	│	┌	┐	┘	└	├	┬	┤	┴	┼	━	┃	┏	┓	┛	┗
84B0:	┣	┳	┫	┻	╋	┠	┯	┨	┷	┿	┝	┰	┥	┸	╂	
84C0:																
84D0:																
84E0:																
84F0:																
8540:																
8550:																
8560:																
8570:																
8580:																
8590:																
85A0:																
85B0:																
85C0:																
85D0:																
85E0:																

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
85F0:																
8640:																
8650:																
8660:																
8670:																
8680:																
8690:																
86A0:																
86B0:																
86C0:																
86D0:																
86E0:																
86F0:																
8740:	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯
8750:	⑰	⑱	⑲	⑳	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ		㍉
8760:	㌔	㌢	㍍	㌘	㌧	㌃	㌶	㍑	㍗	㌍	㌦	㌣	㌫	㍊	㌻	㎜
8770:	㎝	㎞	㎎	㎏	㏄	㎡									㍻	
8780:	〝	〟	№	㏍	℡	㊤	㊥	㊦	㊧	㊨	㈱	㈲	㈹	㍾	㍽	㍼
8790:	≒	≡	∫	∮	∑	√	⊥	∠	∟	⊿	∵	∩	∪			
87A0:																
87B0:																
87C0:																
87D0:																
87E0:																
87F0:																
8840:																
8850:																
8860:																
8870:																
8880:																
8890:																亜
88A0:	唖	娃	阿	哀	愛	挨	姶	逢	葵	茜	穐	悪	握	渥	旭	葦
88B0:	芦	鯵	梓	圧	斡	扱	宛	姐	虻	飴	絢	綾	鮎	或	粟	袷
88C0:	安	庵	按	暗	案	闇	鞍	杏	以	伊	位	依	偉	囲	夷	委
88D0:	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃	萎	衣
88F0:	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭			
8940:	院	陰	隠	韻	吋	右	宇	烏	羽	迂	雨	卯	鵜	窺	丑	碓
8950:	臼	渦	嘘	唄	欝	蔚	鰻	姥	厩	浦	瓜	閏	噂	云	運	雲
8960:	荏	餌	叡	営	嬰	影	映	曳	栄	永	泳	洩	瑛	盈	穎	頴
8970:	英	衛	詠	鋭	液	疫	益	駅	悦	謁	越	閲	榎	厭	円	
8980:	園	堰	奄	宴	延	怨	掩	援	沿	演	炎	焔	煙	燕	猿	縁
8990:	艶	苑	薗	遠	鉛	鴛	塩	於	汚	甥	凹	央	奥	往	応	押
89A0:	旺	横	欧	殴	王	翁	襖	鴬	鴎	黄	岡	沖	荻	億	屋	憶
89B0:	臆	桶	牡	乙	俺	卸	恩	温	穏	音	下	化	仮	何	伽	価
89C0:	佳	加	可	嘉	夏	嫁	家	寡	科	暇	果	架	歌	河	火	珂
89D0:	禍	禾	稼	箇	花	苛	茄	荷	華	菓	蝦	課	嘩	貨	迦	過
89E0:	霞	蚊	俄	峨	我	牙	画	臥	芽	蛾	賀	雅	餓	駕	介	会
89F0:	解	回	塊	壊	廻	快	怪	悔	恢	懐	戒	拐	改			
8A40:	魁	晦	械	海	灰	界	皆	絵	芥	蟹	開	階	貝	凱	劾	外
8A50:	咳	害	崖	慨	概	涯	碍	蓋	街	該	鎧	骸	浬	馨	蛙	垣
8A60:	柿	蛎	鈎	劃	嚇	各	廓	拡	撹	格	核	殻	獲	確	穫	覚
8A70:	角	赫	較	郭	閣	隔	革	学	岳	楽	額	顎	掛	笠	樫	
8A80:	橿	梶	鰍	潟	割	喝	恰	括	活	渇	滑	葛	褐	轄	且	鰹
8A90:	叶	椛	樺	鞄	株	兜	竃	蒲	釜	鎌	噛	鴨	栢	茅	萱	粥
8AA0:	刈	苅	瓦	乾	侃	冠	寒	刊	勘	勧	巻	喚	堪	姦	完	官
8AB0:	寛	干	幹	患	感	慣	憾	換	敢	柑	桓	棺	款	歓	汗	漢
8AC0:	澗	潅	環	甘	監	看	竿	管	簡	緩	缶	翰	肝	艦	莞	観
8AD0:	諌	貫	還	鑑	間	閑	関	陥	韓	館	舘	丸	含	岸	巌	玩
8AE0:	癌	眼	岩	翫	贋	雁	頑	顔	願	企	伎	危	喜	器	基	奇
8AF0:	嬉	寄	岐	希	幾	忌	揮	机	旗	既	期	棋	棄			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
8B40:	機	帰	毅	気	汽	畿	祈	季	稀	紀	徽	規	記	貴	起	軌
8B50:	輝	飢	騎	鬼	亀	偽	儀	妓	宜	戯	技	擬	欺	犠	疑	祇
8B60:	義	蟻	誼	議	掬	菊	鞠	吉	吃	喫	桔	橘	詰	砧	杵	黍
8B70:	却	客	脚	虐	逆	丘	久	仇	休	及	吸	宮	弓	急	救	
8B80:	朽	求	汲	泣	灸	球	究	窮	笈	級	糾	給	旧	牛	去	居
8B90:	巨	拒	拠	挙	渠	虚	許	距	鋸	漁	禦	魚	亨	享	京	供
8BA0:	侠	僑	兇	競	共	凶	協	匡	卿	叫	喬	境	峡	強	彊	怯
8BB0:	恐	恭	挟	教	橋	況	狂	狭	矯	胸	脅	興	蕎	郷	鏡	響
8BC0:	饗	驚	仰	凝	尭	暁	業	局	曲	極	玉	桐	粁	僅	勤	均
8BD0:	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊	芹	菌	衿	襟	謹	近
8BE0:	金	吟	銀	九	倶	句	区	狗	玖	矩	苦	躯	駆	駈	駒	具
8BF0:	愚	虞	喰	空	偶	寓	遇	隅	串	櫛	釧	屑	屈			
8C40:	掘	窟	沓	靴	轡	窪	熊	隈	粂	栗	繰	桑	鍬	勲	君	薫
8C50:	訓	群	軍	郡	卦	袈	祁	係	傾	刑	兄	啓	圭	珪	型	契
8C60:	形	径	恵	慶	慧	憩	掲	携	敬	景	桂	渓	畦	稽	系	経
8C70:	継	繋	罫	茎	荊	蛍	計	詣	警	軽	頚	鶏	芸	迎	鯨	
8C80:	劇	戟	撃	激	隙	桁	傑	欠	決	潔	穴	結	血	訣	月	件
8C90:	倹	倦	健	兼	券	剣	喧	圏	堅	嫌	建	憲	懸	拳	捲	検
8CA0:	権	牽	犬	献	研	硯	絹	県	肩	見	謙	賢	軒	遣	鍵	険
8CB0:	顕	験	鹸	元	原	厳	幻	弦	減	源	玄	現	絃	舷	言	諺
8CC0:	限	乎	個	古	呼	固	姑	孤	己	庫	弧	戸	故	枯	湖	狐
8CD0:	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇	顧	鼓	五	互	伍	午
8CE0:	呉	吾	娯	後	御	悟	梧	檎	瑚	碁	語	誤	護	醐	乞	鯉
8CF0:	交	佼	侯	候	倖	光	公	功	効	勾	厚	口	向			
8D40:	后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	広	庚	康	弘
8D50:	恒	慌	抗	拘	控	攻	昂	晃	更	杭	校	梗	構	江	洪	浩
8D60:	港	溝	甲	皇	硬	稿	糠	紅	紘	絞	綱	耕	考	肯	肱	腔
8D70:	膏	航	荒	行	衡	講	貢	購	郊	酵	鉱	砿	鋼	閤	降	
8D80:	項	香	高	鴻	剛	劫	号	合	壕	拷	濠	豪	轟	麹	克	刻
8D90:	告	国	穀	酷	鵠	黒	獄	漉	腰	甑	忽	惚	骨	狛	込	此
8DA0:	頃	今	困	坤	墾	婚	恨	懇	昏	昆	根	梱	混	痕	紺	艮
8DB0:	魂	些	佐	叉	唆	嵯	左	差	査	沙	瑳	砂	詐	鎖	裟	坐
8DC0:	座	挫	債	催	再	最	哉	塞	妻	宰	彩	才	採	栽	歳	済
8DD0:	災	采	犀	砕	砦	祭	斎	細	菜	裁	載	際	剤	在	材	罪
8DE0:	財	冴	坂	阪	堺	榊	肴	咲	崎	埼	碕	鷺	作	削	咋	搾
8DF0:	昨	朔	柵	窄	策	索	錯	桜	鮭	笹	匙	冊	刷			
8E40:	察	拶	撮	擦	札	殺	薩	雑	皐	鯖	捌	錆	鮫	皿	晒	三
8E50:	傘	参	山	惨	撒	散	桟	燦	珊	産	算	纂	蚕	讃	賛	酸
8E60:	餐	斬	暫	残	仕	仔	伺	使	刺	司	史	嗣	四	士	始	姉
8E70:	姿	子	屍	市	師	志	思	指	支	孜	斯	施	旨	枝	止	
8E80:	死	氏	獅	祉	私	糸	紙	紫	肢	脂	至	視	詞	詩	試	誌
8E90:	諮	資	賜	雌	飼	歯	事	似	侍	児	字	寺	慈	持	時	次
8EA0:	滋	治	爾	璽	痔	磁	示	而	耳	自	蒔	辞	汐	鹿	式	識
8EB0:	鴫	竺	軸	宍	雫	七	叱	執	失	嫉	室	悉	湿	漆	疾	質
8EC0:	実	蔀	篠	偲	柴	芝	屡	蕊	縞	舎	写	射	捨	赦	斜	煮
8ED0:	社	紗	者	謝	車	遮	蛇	邪	借	勺	尺	杓	灼	爵	酌	釈
8EE0:	錫	若	寂	弱	惹	主	取	守	手	朱	殊	狩	珠	種	腫	趣
8EF0:	酒	首	儒	受	呪	寿	授	樹	綬	需	囚	収	周			
8F40:	宗	就	州	修	愁	拾	洲	秀	秋	終	繍	習	臭	舟	蒐	衆
8F50:	襲	讐	蹴	輯	週	酋	酬	集	醜	什	住	充	十	従	戎	柔
8F60:	汁	渋	獣	縦	重	銃	叔	夙	宿	淑	祝	縮	粛	塾	熟	出
8F70:	術	述	俊	峻	春	瞬	竣	舜	駿	准	循	旬	楯	殉	淳	
8F80:	準	潤	盾	純	巡	遵	醇	順	処	初	所	暑	曙	渚	庶	緒
8F90:	署	書	薯	藷	諸	助	叙	女	序	徐	恕	鋤	除	傷	償	勝
8FA0:	匠	升	召	哨	商	唱	嘗	奨	妾	娼	宵	将	小	少	尚	庄
8FB0:	床	廠	彰	承	抄	招	掌	捷	昇	昌	昭	晶	松	梢	樟	樵
8FC0:	沼	消	渉	湘	焼	焦	照	症	省	硝	礁	祥	称	章	笑	粧
8FD0:	紹	肖	菖	蒋	蕉	衝	裳	訟	証	詔	詳	象	賞	醤	鉦	鍾
8FE0:	鐘	障	鞘	上	丈	丞	乗	冗	剰	城	場	壌	嬢	常	情	擾
8FF0:	条	杖	浄	状	畳	穣	蒸	譲	醸	錠	嘱	埴	飾			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
9040:	拭	植	殖	燭	織	職	色	触	食	蝕	辱	尻	伸	信	侵	唇
9050:	娠	寝	審	心	慎	振	新	晋	森	榛	浸	深	申	疹	真	神
9060:	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	刃	塵
9070:	壬	尋	甚	尽	腎	訊	迅	陣	靭	笥	諏	須	酢	図	厨	
9080:	逗	吹	垂	帥	推	水	炊	睡	粋	翠	衰	遂	酔	錐	錘	随
9090:	瑞	髄	崇	嵩	数	枢	趨	雛	据	杉	椙	菅	頗	雀	裾	澄
90A0:	摺	寸	世	瀬	畝	是	凄	制	勢	姓	征	性	成	政	整	星
90B0:	晴	棲	栖	正	清	牲	生	盛	精	聖	声	製	西	誠	誓	請
90C0:	逝	醒	青	静	斉	税	脆	隻	席	惜	戚	斥	昔	析	石	積
90D0:	籍	績	脊	責	赤	跡	蹟	碩	切	拙	接	摂	折	設	窃	節
90E0:	説	雪	絶	舌	蝉	仙	先	千	占	宣	専	尖	川	戦	扇	撰
90F0:	栓	栴	泉	浅	洗	染	潜	煎	煽	旋	穿	箭	線			
9140:	繊	羨	腺	舛	船	薦	詮	賎	践	選	遷	銭	銑	閃	鮮	前
9150:	善	漸	然	全	禅	繕	膳	糎	噌	塑	岨	措	曾	曽	楚	狙
9160:	疏	疎	礎	祖	租	粗	素	組	蘇	訴	阻	遡	鼠	僧	創	双
9170:	叢	倉	喪	壮	奏	爽	宋	層	匝	惣	想	捜	掃	挿	掻	
9180:	操	早	曹	巣	槍	槽	漕	燥	争	痩	相	窓	糟	総	綜	聡
9190:	草	荘	葬	蒼	藻	装	走	送	遭	鎗	霜	騒	像	増	憎	臓
91A0:	蔵	贈	造	促	側	則	即	息	捉	束	測	足	速	俗	属	賊
91B0:	族	続	卒	袖	其	揃	存	孫	尊	損	村	遜	他	多	太	汰
91C0:	詑	唾	堕	妥	惰	打	柁	舵	楕	陀	駄	騨	体	堆	対	耐
91D0:	岱	帯	待	怠	態	戴	替	泰	滞	胎	腿	苔	袋	貸	退	逮
91E0:	隊	黛	鯛	代	台	大	第	醍	題	鷹	滝	瀧	卓	啄	宅	托
91F0:	択	拓	沢	濯	琢	託	鐸	濁	諾	茸	凧	蛸	只			
9240:	叩	但	達	辰	奪	脱	巽	竪	辿	棚	谷	狸	鱈	樽	誰	丹
9250:	単	嘆	坦	担	探	旦	歎	淡	湛	炭	短	端	箪	綻	耽	胆
9260:	蛋	誕	鍛	団	壇	弾	断	暖	檀	段	男	談	値	知	地	弛
9270:	恥	智	池	痴	稚	置	致	蜘	遅	馳	築	畜	竹	筑	蓄	
9280:	逐	秩	窒	茶	嫡	着	中	仲	宙	忠	抽	昼	柱	注	虫	衷
9290:	註	酎	鋳	駐	樗	瀦	猪	苧	著	貯	丁	兆	凋	喋	寵	帖
92A0:	帳	庁	弔	張	彫	徴	懲	挑	暢	朝	潮	牒	町	眺	聴	脹
92B0:	腸	蝶	調	諜	超	跳	銚	長	頂	鳥	勅	捗	直	朕	沈	珍
92C0:	賃	鎮	陳	津	墜	椎	槌	追	鎚	痛	通	塚	栂	掴	槻	佃
92D0:	漬	柘	辻	蔦	綴	鍔	椿	潰	坪	壷	嬬	紬	爪	吊	釣	鶴
92E0:	亭	低	停	偵	剃	貞	呈	堤	定	帝	底	庭	廷	弟	悌	抵
92F0:	挺	提	梯	汀	碇	禎	程	締	艇	訂	諦	蹄	逓			
9340:	邸	鄭	釘	鼎	泥	摘	擢	敵	滴	的	笛	適	鏑	溺	哲	徹
9350:	撤	轍	迭	鉄	典	填	天	展	店	添	纏	甜	貼	転	顛	点
9360:	伝	殿	澱	田	電	兎	吐	堵	塗	妬	屠	徒	斗	杜	渡	登
9370:	菟	賭	途	都	鍍	砥	砺	努	度	土	奴	怒	倒	党	冬	
9380:	凍	刀	唐	塔	塘	套	宕	島	嶋	悼	投	搭	東	桃	梼	棟
9390:	盗	淘	湯	涛	灯	燈	当	痘	祷	等	答	筒	糖	統	到	董
93A0:	蕩	藤	討	謄	豆	踏	逃	透	鐙	陶	頭	騰	闘	働	動	同
93B0:	堂	導	憧	撞	洞	瞳	童	胴	萄	道	銅	峠	鴇	匿	得	徳
93C0:	涜	特	督	禿	篤	毒	独	読	栃	橡	凸	突	椴	届	鳶	苫
93D0:	寅	酉	瀞	噸	屯	惇	敦	沌	豚	遁	頓	呑	曇	鈍	奈	那
93E0:	内	乍	凪	薙	謎	灘	捺	鍋	楢	馴	縄	畷	南	楠	軟	難
93F0:	汝	二	尼	弐	迩	匂	賑	肉	虹	廿	日	乳	入			
9440:	如	尿	韮	任	妊	忍	認	濡	禰	祢	寧	葱	猫	熱	年	念
9450:	捻	撚	燃	粘	乃	廼	之	埜	嚢	悩	濃	納	能	脳	膿	農
9460:	覗	蚤	巴	把	播	覇	杷	波	派	琶	破	婆	罵	芭	馬	俳
9470:	廃	拝	排	敗	杯	盃	牌	背	肺	輩	配	倍	培	媒	梅	
9480:	楳	煤	狽	買	売	賠	陪	這	蝿	秤	矧	萩	伯	剥	博	拍
9490:	柏	泊	白	箔	粕	舶	薄	迫	曝	漠	爆	縛	莫	駁	麦	函
94A0:	箱	硲	箸	肇	筈	櫨	幡	肌	畑	畠	八	鉢	溌	発	醗	髪
94B0:	伐	罰	抜	筏	閥	鳩	噺	塙	蛤	隼	伴	判	半	反	叛	帆
94C0:	搬	斑	板	氾	汎	版	犯	班	畔	繁	般	藩	販	範	釆	煩
94D0:	頒	飯	挽	晩	番	盤	磐	蕃	蛮	匪	卑	否	妃	庇	彼	悲
94E0:	扉	批	披	斐	比	泌	疲	皮	碑	秘	緋	罷	肥	被	誹	費
94F0:	避	非	飛	樋	簸	備	尾	微	枇	毘	琵	眉	美			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
9540:	鼻	柊	稗	匹	疋	髭	彦	膝	菱	肘	弼	必	畢	筆	逼	桧
9550:	姫	媛	紐	百	謬	俵	彪	標	氷	漂	瓢	票	表	評	豹	廟
9560:	描	病	秒	苗	錨	鋲	蒜	蛭	鰭	品	彬	斌	浜	瀕	貧	賓
9570:	頻	敏	瓶	不	付	埠	夫	婦	富	冨	布	府	怖	扶	敷	
9580:	斧	普	浮	父	符	腐	膚	芙	譜	負	賦	赴	阜	附	侮	撫
9590:	武	舞	葡	蕪	部	封	楓	風	葺	蕗	伏	副	復	幅	服	福
95A0:	腹	複	覆	淵	弗	払	沸	仏	物	鮒	分	吻	噴	墳	憤	扮
95B0:	焚	奮	粉	糞	紛	雰	文	聞	丙	併	兵	塀	幣	平	弊	柄
95C0:	並	蔽	閉	陛	米	頁	僻	壁	癖	碧	別	瞥	蔑	箆	偏	変
95D0:	片	篇	編	辺	返	遍	便	勉	娩	弁	鞭	保	舗	鋪	圃	捕
95E0:	歩	甫	補	輔	穂	募	墓	慕	戊	暮	母	簿	菩	倣	俸	包
95F0:	呆	報	奉	宝	峰	峯	崩	庖	抱	捧	放	方	朋			
9640:	法	泡	烹	砲	縫	胞	芳	萌	蓬	蜂	褒	訪	豊	邦	鋒	飽
9650:	鳳	鵬	乏	亡	傍	剖	坊	妨	帽	忘	忙	房	暴	望	某	棒
9660:	冒	紡	肪	膨	謀	貌	貿	鉾	防	吠	頬	北	僕	卜	墨	撲
9670:	朴	牧	睦	穆	釦	勃	没	殆	堀	幌	奔	本	翻	凡	盆	
9680:	摩	磨	魔	麻	埋	妹	昧	枚	毎	哩	槙	幕	膜	枕	鮪	柾
9690:	鱒	桝	亦	俣	又	抹	末	沫	迄	侭	繭	麿	万	慢	満	漫
96A0:	蔓	味	未	魅	巳	箕	岬	密	蜜	湊	蓑	稔	脈	妙	粍	民
96B0:	眠	務	夢	無	牟	矛	霧	鵡	椋	婿	娘	冥	名	命	明	盟
96C0:	迷	銘	鳴	姪	牝	滅	免	棉	綿	緬	面	麺	摸	模	茂	妄
96D0:	孟	毛	猛	盲	網	耗	蒙	儲	木	黙	目	杢	勿	餅	尤	戻
96E0:	籾	貰	問	悶	紋	門	匁	也	冶	夜	爺	耶	野	弥	矢	厄
96F0:	役	約	薬	訳	躍	靖	柳	薮	鑓	愉	愈	油	癒			
9740:	諭	輸	唯	佑	優	勇	友	宥	幽	悠	憂	揖	有	柚	湧	涌
9750:	猶	猷	由	祐	裕	誘	遊	邑	郵	雄	融	夕	予	余	与	誉
9760:	輿	預	傭	幼	妖	容	庸	揚	揺	擁	曜	楊	様	洋	溶	熔
9770:	用	窯	羊	耀	葉	蓉	要	謡	踊	遥	陽	養	慾	抑	欲	
9780:	沃	浴	翌	翼	淀	羅	螺	裸	来	莱	頼	雷	洛	絡	落	酪
9790:	乱	卵	嵐	欄	濫	藍	蘭	覧	利	吏	履	李	梨	理	璃	痢
97A0:	裏	裡	里	離	陸	律	率	立	葎	掠	略	劉	流	溜	琉	留
97B0:	硫	粒	隆	竜	龍	侶	慮	旅	虜	了	亮	僚	両	凌	寮	料
97C0:	梁	涼	猟	療	瞭	稜	糧	良	諒	遼	量	陵	領	力	緑	倫
97D0:	厘	林	淋	燐	琳	臨	輪	隣	鱗	麟	瑠	塁	涙	累	類	令
97E0:	伶	例	冷	励	嶺	怜	玲	礼	苓	鈴	隷	零	霊	麗	齢	暦
97F0:	歴	列	劣	烈	裂	廉	恋	憐	漣	煉	簾	練	聯			
9840:	蓮	連	錬	呂	魯	櫓	炉	賂	路	露	労	婁	廊	弄	朗	楼
9850:	榔	浪	漏	牢	狼	篭	老	聾	蝋	郎	六	麓	禄	肋	録	論
9860:	倭	和	話	歪	賄	脇	惑	枠	鷲	亙	亘	鰐	詫	藁	蕨	椀
9870:	湾	碗	腕													
9880:																
9890:																弌
98A0:	丐	丕	个	丱	丶	丼	丿	乂	乖	乘	亂	亅	豫	亊	舒	弍
98B0:	于	亞	亟	亠	亢	亰	亳	亶	从	仍	仄	仆	仂	仗	仞	仭
98C0:	仟	价	伉	佚	估	佛	佝	佗	佇	佶	侈	侏	侘	佻	佩	佰
98D0:	侑	佯	來	侖	儘	俔	俟	俎	俘	俛	俑	俚	俐	俤	俥	倚
98E0:	倨	倔	倪	倥	倅	伜	俶	倡	倩	倬	俾	俯	們	倆	偃	假
98F0:	會	偕	偐	偈	做	偖	偬	偸	傀	傚	傅	傴	傲			
9940:	僉	僊	傳	僂	僖	僞	僥	僭	僣	僮	價	僵	儉	儁	儂	儖
9950:	儕	儔	儚	儡	儺	儷	儼	儻	儿	兀	兒	兌	兔	兢	竸	兩
9960:	兪	兮	冀	冂	囘	册	冉	冏	冑	冓	冕	冖	冤	冦	冢	冩
9970:	冪	冫	决	冱	冲	冰	况	冽	凅	凉	凛	几	處	凩	凭	
9980:	凰	凵	凾	刄	刋	刔	刎	刧	刪	刮	刳	刹	剏	剄	剋	剌
9990:	剞	剔	剪	剴	剩	剳	剿	剽	劍	劔	劒	剱	劈	劑	辨	辧
99A0:	劬	劭	劼	劵	勁	勍	勗	勞	勣	勦	飭	勠	勳	勵	勸	勹
99B0:	匆	匈	甸	匍	匐	匏	匕	匚	匣	匯	匱	匳	匸	區	卆	卅
99C0:	丗	卉	卍	凖	卞	卩	卮	夘	卻	卷	厂	厖	厠	厦	厥	厮
99D0:	厰	厶	參	簒	雙	叟	曼	燮	叮	叨	叭	叺	吁	吽	呀	听
99E0:	吭	吼	吮	吶	吩	吝	呎	咏	呵	咎	呟	呱	呷	呰	咒	呻
99F0:	咀	呶	咄	咐	咆	哇	咢	咸	咥	咬	哄	哈	咨			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
9A40:	咫	哂	咤	咾	咼	哘	哥	哦	唏	唔	哽	哮	哭	哺	哢	唹
9A50:	啀	啣	啌	售	啜	啅	啖	啗	唸	唳	啝	喙	喀	咯	喊	喟
9A60:	啻	啾	喘	喞	單	啼	喃	喩	喇	喨	嗚	嗅	嗟	嗄	嗜	嗤
9A70:	嗔	嘔	嗷	嘖	嗾	嗽	嘛	嗹	噎	噐	營	嘴	嘶	嘲	嘸	
9A80:	噫	噤	嘯	噬	噪	嚆	嚀	嚊	嚠	嚔	嚏	嚥	嚮	嚶	嚴	囂
9A90:	嚼	囁	囃	囀	囈	囎	囑	囓	囗	囮	囹	圀	囿	圄	圉	圈
9AA0:	國	圍	圓	團	圖	嗇	圜	圦	圷	圸	坎	圻	址	坏	坩	埀
9AB0:	垈	坡	坿	垉	垓	垠	垳	垤	垪	垰	埃	埆	埔	埒	埓	堊
9AC0:	埖	埣	堋	堙	堝	塲	堡	塢	塋	塰	毀	塒	堽	塹	墅	墹
9AD0:	墟	墫	墺	壞	墻	墸	墮	壅	壓	壑	壗	壙	壘	壥	壜	壤
9AE0:	壟	壯	壺	壹	壻	壼	壽	夂	夊	夐	夛	梦	夥	夬	夭	夲
9AF0:	夸	夾	竒	奕	奐	奎	奚	奘	奢	奠	奧	奬	奩			
9B40:	奸	妁	妝	佞	侫	妣	妲	姆	姨	姜	妍	姙	姚	娥	娟	娑
9B50:	娜	娉	娚	婀	婬	婉	娵	娶	婢	婪	媚	媼	媾	嫋	嫂	媽
9B60:	嫣	嫗	嫦	嫩	嫖	嫺	嫻	嬌	嬋	嬖	嬲	嫐	嬪	嬶	嬾	孃
9B70:	孅	孀	孑	孕	孚	孛	孥	孩	孰	孳	孵	學	斈	孺	宀	
9B80:	它	宦	宸	寃	寇	寉	寔	寐	寤	實	寢	寞	寥	寫	寰	寶
9B90:	寳	尅	將	專	對	尓	尠	尢	尨	尸	尹	屁	屆	屎	屓	屐
9BA0:	屏	孱	屬	屮	乢	屶	屹	岌	岑	岔	妛	岫	岻	岶	岼	岷
9BB0:	峅	岾	峇	峙	峩	峽	峺	峭	嶌	峪	崋	崕	崗	嵜	崟	崛
9BC0:	崑	崔	崢	崚	崙	崘	嵌	嵒	嵎	嵋	嵬	嵳	嵶	嶇	嶄	嶂
9BD0:	嶢	嶝	嶬	嶮	嶽	嶐	嶷	嶼	巉	巍	巓	巒	巖	巛	巫	已
9BE0:	巵	帋	帚	帙	帑	帛	帶	帷	幄	幃	幀	幎	幗	幔	幟	幢
9BF0:	幤	幇	幵	并	幺	麼	广	庠	廁	廂	廈	廐	廏			
9C40:	廖	廣	廝	廚	廛	廢	廡	廨	廩	廬	廱	廳	廰	廴	廸	廾
9C50:	弃	弉	彝	彜	弋	弑	弖	弩	弭	弸	彁	彈	彌	彎	弯	彑
9C60:	彖	彗	彙	彡	彭	彳	彷	徃	徂	彿	徊	很	徑	徇	從	徙
9C70:	徘	徠	徨	徭	徼	忖	忻	忤	忸	忱	忝	悳	忿	怡	恠	
9C80:	怙	怐	怩	怎	怱	怛	怕	怫	怦	怏	怺	恚	恁	恪	恷	恟
9C90:	恊	恆	恍	恣	恃	恤	恂	恬	恫	恙	悁	悍	惧	悃	悚	悄
9CA0:	悛	悖	悗	悒	悧	悋	惡	悸	惠	惓	悴	忰	悽	惆	悵	惘
9CB0:	慍	愕	愆	惶	惷	愀	惴	惺	愃	愡	惻	惱	愍	愎	慇	愾
9CC0:	愨	愧	慊	愿	愼	愬	愴	愽	慂	慄	慳	慷	慘	慙	慚	慫
9CD0:	慴	慯	慥	慱	慟	慝	慓	慵	憙	憖	憇	憬	憔	憚	憊	憑
9CE0:	憫	憮	懌	懊	應	懷	懈	懃	懆	憺	懋	罹	懍	懦	懣	懶
9CF0:	懺	懴	懿	懽	懼	懾	戀	戈	戉	戍	戌	戔	戛			
9D40:	戞	戡	截	戮	戰	戲	戳	扁	扎	扞	扣	扛	扠	扨	扼	抂
9D50:	抉	找	抒	抓	抖	拔	抃	抔	拗	拑	抻	拏	拿	拆	擔	拈
9D60:	拜	拌	拊	拂	拇	抛	拉	挌	拮	拱	挧	挂	挈	拯	拵	捐
9D70:	挾	捍	搜	捏	掖	掎	掀	掫	捶	掣	掏	掉	掟	掵	捫	
9D80:	捩	掾	揩	揀	揆	揣	揉	插	揶	揄	搖	搴	搆	搓	搦	搶
9D90:	攝	搗	搨	搏	摧	摯	摶	摎	攪	撕	撓	撥	撩	撈	撼	據
9DA0:	擒	擅	擇	撻	擘	擂	擱	擧	舉	擠	擡	抬	擣	擯	攬	擶
9DB0:	擴	擲	擺	攀	擽	攘	攜	攅	攤	攣	攫	攴	攵	攷	收	攸
9DC0:	畋	效	敖	敕	敍	敘	敞	敝	敲	數	斂	斃	變	斛	斟	斫
9DD0:	斷	旃	旆	旁	旄	旌	旒	旛	旙	无	旡	旱	杲	昊	昃	旻
9DE0:	杳	昵	昶	昴	昜	晏	晄	晉	晁	晞	晝	晤	晧	晨	晟	晢
9DF0:	晰	暃	暈	暎	暉	暄	暘	暝	曁	暹	曉	暾	暼			
9E40:	曄	暸	曖	曚	曠	昿	曦	曩	曰	曵	曷	朏	朖	朞	朦	朧
9E50:	霸	朮	朿	朶	杁	朸	朷	杆	杞	杠	杙	杣	杤	枉	杰	枩
9E60:	杼	杪	枌	枋	枦	枡	枅	枷	柯	枴	柬	枳	柩	枸	柤	柞
9E70:	柝	柢	柮	枹	柎	柆	柧	檜	栞	框	栩	桀	桍	栲	桎	
9E80:	梳	栫	桙	档	桷	桿	梟	梏	梭	梔	條	梛	梃	檮	梹	桴
9E90:	梵	梠	梺	椏	梍	桾	椁	棊	椈	棘	椢	椦	棡	椌	棍	棔
9EA0:	棧	棕	椶	椒	椄	棗	棣	椥	棹	棠	棯	椨	椪	椚	椣	椡
9EB0:	棆	楹	楷	楜	楸	楫	楔	楾	楮	椹	楴	椽	楙	椰	楡	楞
9EC0:	楝	榁	楪	榲	榮	槐	榿	槁	槓	榾	槎	寨	槊	槝	榻	槃
9ED0:	榧	樮	榑	榠	榜	榕	榴	槞	槨	樂	樛	槿	權	槹	槲	槧
9EE0:	樅	榱	樞	槭	樔	槫	樊	樒	櫁	樣	樓	橄	樌	橲	樶	橸
9EF0:	橇	橢	橙	橦	橈	樸	樢	檐	檍	檠	檄	檢	檣			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
9F40:	檗	蘗	檻	櫃	櫂	檸	檳	檬	櫞	櫑	櫟	檪	櫚	櫪	櫻	欅
9F50:	蘖	櫺	欒	欖	鬱	欟	欸	欷	盜	欹	飮	歇	歃	歉	歐	歙
9F60:	歔	歛	歟	歡	歸	歹	歿	殀	殄	殃	殍	殘	殕	殞	殤	殪
9F70:	殫	殯	殲	殱	殳	殷	殼	毆	毋	毓	毟	毬	毫	毳	毯	
9F80:	麾	氈	氓	气	氛	氤	氣	汞	汕	汢	汪	沂	沍	沚	沁	沛
9F90:	汾	汨	汳	沒	沐	泄	泱	泓	沽	泗	泅	泝	沮	沱	沾	沺
9FA0:	泛	泯	泙	泪	洟	衍	洶	洫	洽	洸	洙	洵	洳	洒	洌	浣
9FB0:	涓	浤	浚	浹	浙	涎	涕	濤	涅	淹	渕	渊	涵	淇	淦	涸
9FC0:	淆	淬	淞	淌	淨	淒	淅	淺	淙	淤	淕	淪	淮	渭	湮	渮
9FD0:	渙	湲	湟	渾	渣	湫	渫	湶	湍	渟	湃	渺	湎	渤	滿	渝
9FE0:	游	溂	溪	溘	滉	溷	滓	溽	溯	滄	溲	滔	滕	溏	溥	滂
9FF0:	溟	潁	漑	灌	滬	滸	滾	漿	滲	漱	滯	漲	滌			
E040:	漾	漓	滷	澆	潺	潸	澁	澀	潯	潛	濳	潭	澂	潼	潘	澎
E050:	澑	濂	潦	澳	澣	澡	澤	澹	濆	澪	濟	濕	濬	濔	濘	濱
E060:	濮	濛	瀉	瀋	濺	瀑	瀁	瀏	濾	瀛	瀚	潴	瀝	瀘	瀟	瀰
E070:	瀾	瀲	灑	灣	炙	炒	炯	烱	炬	炸	炳	炮	烟	烋	烝	
E080:	烙	焉	烽	焜	焙	煥	煕	熈	煦	煢	煌	煖	煬	熏	燻	熄
E090:	熕	熨	熬	燗	熹	熾	燒	燉	燔	燎	燠	燬	燧	燵	燼	燹
E0A0:	燿	爍	爐	爛	爨	爭	爬	爰	爲	爻	爼	爿	牀	牆	牋	牘
E0B0:	牴	牾	犂	犁	犇	犒	犖	犢	犧	犹	犲	狃	狆	狄	狎	狒
E0C0:	狢	狠	狡	狹	狷	倏	猗	猊	猜	猖	猝	猴	猯	猩	猥	猾
E0D0:	獎	獏	默	獗	獪	獨	獰	獸	獵	獻	獺	珈	玳	珎	玻	珀
E0E0:	珥	珮	珞	璢	琅	瑯	琥	珸	琲	琺	瑕	琿	瑟	瑙	瑁	瑜
E0F0:	瑩	瑰	瑣	瑪	瑶	瑾	璋	璞	璧	瓊	瓏	瓔	珱			
E140:	瓠	瓣	瓧	瓩	瓮	瓲	瓰	瓱	瓸	瓷	甄	甃	甅	甌	甎	甍
E150:	甕	甓	甞	甦	甬	甼	畄	畍	畊	畉	畛	畆	畚	畩	畤	畧
E160:	畫	畭	畸	當	疆	疇	畴	疊	疉	疂	疔	疚	疝	疥	疣	痂
E170:	疳	痃	疵	疽	疸	疼	疱	痍	痊	痒	痙	痣	痞	痾	痿	
E180:	痼	瘁	痰	痺	痲	痳	瘋	瘍	瘉	瘟	瘧	瘠	瘡	瘢	瘤	瘴
E190:	瘰	瘻	癇	癈	癆	癜	癘	癡	癢	癨	癩	癪	癧	癬	癰	癲
E1A0:	癶	癸	發	皀	皃	皈	皋	皎	皖	皓	皙	皚	皰	皴	皸	皹
E1B0:	皺	盂	盍	盖	盒	盞	盡	盥	盧	盪	蘯	盻	眈	眇	眄	眩
E1C0:	眤	眞	眥	眦	眛	眷	眸	睇	睚	睨	睫	睛	睥	睿	睾	睹
E1D0:	瞎	瞋	瞑	瞠	瞞	瞰	瞶	瞹	瞿	瞼	瞽	瞻	矇	矍	矗	矚
E1E0:	矜	矣	矮	矼	砌	砒	礦	砠	礪	硅	碎	硴	碆	硼	碚	碌
E1F0:	碣	碵	碪	碯	磑	磆	磋	磔	碾	碼	磅	磊	磬			
E240:	磧	磚	磽	磴	礇	礒	礑	礙	礬	礫	祀	祠	祗	祟	祚	祕
E250:	祓	祺	祿	禊	禝	禧	齋	禪	禮	禳	禹	禺	秉	秕	秧	秬
E260:	秡	秣	稈	稍	稘	稙	稠	稟	禀	稱	稻	稾	稷	穃	穗	穉
E270:	穡	穢	穩	龝	穰	穹	穽	窈	窗	窕	窘	窖	窩	竈	窰	
E280:	窶	竅	竄	窿	邃	竇	竊	竍	竏	竕	竓	站	竚	竝	竡	竢
E290:	竦	竭	竰	笂	笏	笊	笆	笳	笘	笙	笞	笵	笨	笶	筐	筺
E2A0:	笄	筍	笋	筌	筅	筵	筥	筴	筧	筰	筱	筬	筮	箝	箘	箟
E2B0:	箍	箜	箚	箋	箒	箏	筝	箙	篋	篁	篌	篏	箴	篆	篝	篩
E2C0:	簑	簔	篦	篥	籠	簀	簇	簓	篳	篷	簗	簍	篶	簣	簧	簪
E2D0:	簟	簷	簫	簽	籌	籃	籔	籏	籀	籐	籘	籟	籤	籖	籥	籬
E2E0:	籵	粃	粐	粤	粭	粢	粫	粡	粨	粳	粲	粱	粮	粹	粽	糀
E2F0:	糅	糂	糘	糒	糜	糢	鬻	糯	糲	糴	糶	糺	紆			
E340:	紂	紜	紕	紊	絅	絋	紮	紲	紿	紵	絆	絳	絖	絎	絲	絨
E350:	絮	絏	絣	經	綉	絛	綏	絽	綛	綺	綮	綣	綵	緇	綽	綫
E360:	總	綢	綯	緜	綸	綟	綰	緘	緝	緤	緞	緻	緲	緡	縅	縊
E370:	縣	縡	縒	縱	縟	縉	縋	縢	繆	繦	縻	縵	縹	繃	縷	
E380:	縲	縺	繧	繝	繖	繞	繙	繚	繹	繪	繩	繼	繻	纃	緕	繽
E390:	辮	繿	纈	纉	續	纒	纐	纓	纔	纖	纎	纛	纜	缸	缺	罅
E3A0:	罌	罍	罎	罐	网	罕	罔	罘	罟	罠	罨	罩	罧	罸	羂	羆
E3B0:	羃	羈	羇	羌	羔	羞	羝	羚	羣	羯	羲	羹	羮	羶	羸	譱
E3C0:	翅	翆	翊	翕	翔	翡	翦	翩	翳	翹	飜	耆	耄	耋	耒	耘
E3D0:	耙	耜	耡	耨	耿	耻	聊	聆	聒	聘	聚	聟	聢	聨	聳	聲
E3E0:	聰	聶	聹	聽	聿	肄	肆	肅	肛	肓	肚	肭	冐	肬	胛	胥
E3F0:	胙	胝	胄	胚	胖	脉	胯	胱	脛	脩	脣	脯	腋			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
E440:	隋	腆	脾	腓	腑	胼	腱	腮	腥	腦	腴	膃	膈	膊	膀	膂
E450:	膠	膕	膤	膣	腟	膓	膩	膰	膵	膾	膸	膽	臀	臂	膺	臉
E460:	臍	臑	臙	臘	臈	臚	臟	臠	臧	臺	臻	臾	舁	舂	舅	與
E470:	舊	舍	舐	舖	舩	舫	舸	舳	艀	艙	艘	艝	艚	艟	艤	
E480:	艢	艨	艪	艫	舮	艱	艷	艸	艾	芍	芒	芫	芟	芻	芬	苡
E490:	苣	苟	苒	苴	苳	苺	莓	范	苻	苹	苞	茆	苜	茉	苙	茵
E4A0:	茴	茖	茲	茱	荀	茹	荐	荅	茯	茫	茗	茘	莅	莚	莪	莟
E4B0:	莢	莖	茣	莎	莇	莊	荼	莵	荳	荵	莠	莉	莨	菴	萓	菫
E4C0:	菎	菽	萃	菘	萋	菁	菷	萇	菠	菲	萍	萢	萠	莽	萸	蔆
E4D0:	菻	葭	萪	萼	蕚	蒄	葷	葫	蒭	葮	蒂	葩	葆	萬	葯	葹
E4E0:	萵	蓊	葢	蒹	蒿	蒟	蓙	蓍	蒻	蓚	蓐	蓁	蓆	蓖	蒡	蔡
E4F0:	蓿	蓴	蔗	蔘	蔬	蔟	蔕	蔔	蓼	蕀	蕣	蕘	蕈			
E540:	蕁	蘂	蕋	蕕	薀	薤	薈	薑	薊	薨	蕭	薔	薛	藪	薇	薜
E550:	蕷	蕾	薐	藉	薺	藏	薹	藐	藕	藝	藥	藜	藹	蘊	蘓	蘋
E560:	藾	藺	蘆	蘢	蘚	蘰	蘿	虍	乕	虔	號	虧	虱	蚓	蚣	蚩
E570:	蚪	蚋	蚌	蚶	蚯	蛄	蛆	蚰	蛉	蠣	蚫	蛔	蛞	蛩	蛬	
E580:	蛟	蛛	蛯	蜒	蜆	蜈	蜀	蜃	蛻	蜑	蜉	蜍	蛹	蜊	蜴	蜿
E590:	蜷	蜻	蜥	蜩	蜚	蝠	蝟	蝸	蝌	蝎	蝴	蝗	蝨	蝮	蝙	蝓
E5A0:	蝣	蝪	蠅	螢	螟	螂	螯	蟋	螽	蟀	蟐	雖	螫	蟄	螳	蟇
E5B0:	蟆	螻	蟯	蟲	蟠	蠏	蠍	蟾	蟶	蟷	蠎	蟒	蠑	蠖	蠕	蠢
E5C0:	蠡	蠱	蠶	蠹	蠧	蠻	衄	衂	衒	衙	衞	衢	衫	袁	衾	袞
E5D0:	衵	衽	袵	衲	袂	袗	袒	袮	袙	袢	袍	袤	袰	袿	袱	裃
E5E0:	裄	裔	裘	裙	裝	裹	褂	裼	裴	裨	裲	褄	褌	褊	褓	襃
E5F0:	褞	褥	褪	褫	襁	襄	褻	褶	褸	襌	褝	襠	襞			
E640:	襦	襤	襭	襪	襯	襴	襷	襾	覃	覈	覊	覓	覘	覡	覩	覦
E650:	覬	覯	覲	覺	覽	覿	觀	觚	觜	觝	觧	觴	觸	訃	訖	訐
E660:	訌	訛	訝	訥	訶	詁	詛	詒	詆	詈	詼	詭	詬	詢	誅	誂
E670:	誄	誨	誡	誑	誥	誦	誚	誣	諄	諍	諂	諚	諫	諳	諧	
E680:	諤	諱	謔	諠	諢	諷	諞	諛	謌	謇	謚	諡	謖	謐	謗	謠
E690:	謳	鞫	謦	謫	謾	謨	譁	譌	譏	譎	證	譖	譛	譚	譫	譟
E6A0:	譬	譯	譴	譽	讀	讌	讎	讒	讓	讖	讙	讚	谺	豁	谿	豈
E6B0:	豌	豎	豐	豕	豢	豬	豸	豺	貂	貉	貅	貊	貍	貎	貔	豼
E6C0:	貘	戝	貭	貪	貽	貲	貳	貮	貶	賈	賁	賤	賣	賚	賽	賺
E6D0:	賻	贄	贅	贊	贇	贏	贍	贐	齎	贓	賍	贔	贖	赧	赭	赱
E6E0:	赳	趁	趙	跂	趾	趺	跏	跚	跖	跌	跛	跋	跪	跫	跟	跣
E6F0:	跼	踈	踉	跿	踝	踞	踐	踟	蹂	踵	踰	踴	蹊			
E740:	蹇	蹉	蹌	蹐	蹈	蹙	蹤	蹠	踪	蹣	蹕	蹶	蹲	蹼	躁	躇
E750:	躅	躄	躋	躊	躓	躑	躔	躙	躪	躡	躬	躰	軆	躱	躾	軅
E760:	軈	軋	軛	軣	軼	軻	軫	軾	輊	輅	輕	輒	輙	輓	輜	輟
E770:	輛	輌	輦	輳	輻	輹	轅	轂	輾	轌	轉	轆	轎	轗	轜	
E780:	轢	轣	轤	辜	辟	辣	辭	辯	辷	迚	迥	迢	迪	迯	邇	迴
E790:	逅	迹	迺	逑	逕	逡	逍	逞	逖	逋	逧	逶	逵	逹	迸	遏
E7A0:	遐	遑	遒	逎	遉	逾	遖	遘	遞	遨	遯	遶	隨	遲	邂	遽
E7B0:	邁	邀	邊	邉	邏	邨	邯	邱	邵	郢	郤	扈	郛	鄂	鄒	鄙
E7C0:	鄲	鄰	酊	酖	酘	酣	酥	酩	酳	酲	醋	醉	醂	醢	醫	醯
E7D0:	醪	醵	醴	醺	釀	釁	釉	釋	釐	釖	釟	釡	釛	釼	釵	釶
E7E0:	鈞	釿	鈔	鈬	鈕	鈑	鉞	鉗	鉅	鉉	鉤	鉈	銕	鈿	鉋	鉐
E7F0:	銜	銖	銓	銛	鉚	鋏	銹	銷	鋩	錏	鋺	鍄	錮			
E840:	錙	錢	錚	錣	錺	錵	錻	鍜	鍠	鍼	鍮	鍖	鎰	鎬	鎭	鎔
E850:	鎹	鏖	鏗	鏨	鏥	鏘	鏃	鏝	鏐	鏈	鏤	鐚	鐔	鐓	鐃	鐇
E860:	鐐	鐶	鐫	鐵	鐡	鐺	鑁	鑒	鑄	鑛	鑠	鑢	鑞	鑪	鈩	鑰
E870:	鑵	鑷	鑽	鑚	鑼	鑾	钁	鑿	閂	閇	閊	閔	閖	閘	閙	
E880:	閠	閨	閧	閭	閼	閻	閹	閾	闊	濶	闃	闍	闌	闕	闔	闖
E890:	關	闡	闥	闢	阡	阨	阮	阯	陂	陌	陏	陋	陷	陜	陞	陝
E8A0:	陟	陦	陲	陬	隍	隘	隕	隗	險	隧	隱	隲	隰	隴	隶	隸
E8B0:	隹	雎	雋	雉	雍	襍	雜	霍	雕	雹	霄	霆	霈	霓	霎	霑
E8C0:	霏	霖	霙	霤	霪	霰	霹	霽	霾	靄	靆	靈	靂	靉	靜	靠
E8D0:	靤	靦	靨	勒	靫	靱	靹	鞅	靼	鞁	靺	鞆	鞋	鞏	鞐	鞜
E8E0:	鞨	鞦	鞣	鞳	鞴	韃	韆	韈	韋	韜	韭	齏	韲	竟	韶	韵
E8F0:	頏	頌	頸	頤	頡	頷	頽	顆	顏	顋	顫	顯	顰			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
E940:	顱	顳	顴	颪	颯	颱	颶	飄	飃	飆	飩	飫	餃	餉	餒	餔
E950:	餘	餡	餝	餞	餤	餠	餬	餮	餽	餾	饂	饉	饅	饐	饋	饑
E960:	饒	饌	饕	馗	馘	馥	馭	馮	馼	駟	駛	駝	駘	駑	駭	駮
E970:	駱	駲	駻	駸	騁	騏	騅	駢	騙	騫	騷	驅	驂	驀	驃	
E980:	騾	驕	驍	驛	驗	驟	驢	驥	驤	驩	驫	驪	骭	骰	骼	髀
E990:	髏	髑	髓	體	髞	髟	髢	髣	髦	髯	髫	髮	髴	髱	髷	髻
E9A0:	鬆	鬘	鬚	鬟	鬢	鬣	鬥	鬧	鬨	鬩	鬪	鬮	鬯	鬲	魄	魃
E9B0:	魏	魍	魎	魑	魘	魴	鮓	鮃	鮑	鮖	鮗	鮟	鮠	鮨	鮴	鯀
E9C0:	鯊	鮹	鯆	鯏	鯑	鯒	鯣	鯢	鯤	鯔	鯡	鰺	鯲	鯱	鯰	鰕
E9D0:	鰔	鰉	鰓	鰌	鰆	鰈	鰒	鰊	鰄	鰮	鰛	鰥	鰤	鰡	鰰	鱇
E9E0:	鰲	鱆	鰾	鱚	鱠	鱧	鱶	鱸	鳧	鳬	鳰	鴉	鴈	鳫	鴃	鴆
E9F0:	鴪	鴦	鶯	鴣	鴟	鵄	鴕	鴒	鵁	鴿	鴾	鵆	鵈			
EA40:	鵝	鵞	鵤	鵑	鵐	鵙	鵲	鶉	鶇	鶫	鵯	鵺	鶚	鶤	鶩	鶲
EA50:	鷄	鷁	鶻	鶸	鶺	鷆	鷏	鷂	鷙	鷓	鷸	鷦	鷭	鷯	鷽	鸚
EA60:	鸛	鸞	鹵	鹹	鹽	麁	麈	麋	麌	麒	麕	麑	麝	麥	麩	麸
EA70:	麪	麭	靡	黌	黎	黏	黐	黔	黜	點	黝	黠	黥	黨	黯	
EA80:	黴	黶	黷	黹	黻	黼	黽	鼇	鼈	皷	鼕	鼡	鼬	鼾	齊	齒
EA90:	齔	齣	齟	齠	齡	齦	齧	齬	齪	齷	齲	齶	龕	龜	龠	堯
EAA0:	槇	遙	瑤	凜	熙											
EAB0:																
EAC0:																
EAD0:																
EAE0:																
EAF0:																
EB40:																
EB50:																
EB60:																
EB70:																
EB80:																
EB90:																
EBA0:																
EBB0:																
EBC0:																
EBD0:																
EBE0:																
EBF0:																
EC40:																
EC50:																
EC60:																
EC70:																
EC80:																
EC90:																
ECA0:																
ECB0:																
ECC0:																
ECD0:																
ECE0:																
ECF0:																
ED40:	纊	褜	鍈	銈	蓜	俉	炻	昱	棈	鋹	曻	彅	丨	仡	仼	伀
ED50:	伃	伹	佖	侒	侊	侚	侔	俍	偀	倢	俿	倞	偆	偰	偂	傔
ED60:	僴	僘	兊	兤	冝	冾	凬	刕	劜	劦	勀	勛	匀	匇	匤	卲
ED70:	厓	厲	叝	﨎	咜	咊	咩	哿	喆	坙	坥	垬	埈	埇	﨏	
ED80:	塚	增	墲	夋	奓	奛	奝	奣	妤	妺	孖	寀	甯	寘	寬	尞
ED90:	岦	岺	峵	崧	嵓	﨑	嵂	嵭	嶸	嶹	巐	弡	弴	彧	德	忞
EDA0:	恝	悅	悊	惞	惕	愠	惲	愑	愷	愰	憘	戓	抦	揵	摠	撝
EDB0:	擎	敎	昀	昕	昻	昉	昮	昞	昤	晥	晗	晙	晴	晳	暙	暠
EDC0:	暲	暿	曺	朎	朗	杦	枻	桒	柀	栁	桄	棏	﨓	楨	﨔	榘
EDD0:	槢	樰	橫	橆	橳	橾	櫢	櫤	毖	氿	汜	沆	汯	泚	洄	涇
EDE0:	浯	涖	涬	淏	淸	淲	淼	渹	湜	渧	渼	溿	澈	澵	濵	瀅
EDF0:	瀇	瀨	炅	炫	焏	焄	煜	煆	煇	凞	燁	燾	犱			

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
EE40:	犾	猤	猪	獷	玽	珉	珖	珣	珒	琇	珵	琦	琪	琩	琮	瑢
EE50:	璉	璟	甁	畯	皂	皜	皞	皛	皦	益	睆	劯	砡	硎	硤	硺
EE60:	礰	礼	神	祥	禔	福	禛	竑	竧	靖	竫	箞	精	絈	絜	綷
EE70:	綠	緖	繒	罇	羡	羽	茁	荢	荿	菇	菶	葈	蒴	蕓	蕙	
EE80:	蕫	﨟	薰	蘒	﨡	蠇	裵	訒	訷	詹	誧	誾	諟	諸	諶	譓
EE90:	譿	賰	賴	贒	赶	﨣	軏	﨤	逸	遧	郞	都	鄕	鄧	釚	釗
EEA0:	釞	釭	釮	釤	釥	鈆	鈐	鈊	鈺	鉀	鈼	鉎	鉙	鉑	鈹	鉧
EEB0:	銧	鉷	鉸	鋧	鋗	鋙	鋐	﨧	鋕	鋠	鋓	錥	錡	鋻	﨨	錞
EEC0:	鋿	錝	錂	鍰	鍗	鎤	鏆	鏞	鏸	鐱	鑅	鑈	閒	隆	﨩	隝
EED0:	隯	霳	霻	靃	靍	靏	靑	靕	顗	顥	飯	飼	餧	館	馞	驎
EEE0:	髙	髜	魵	魲	鮏	鮱	鮻	鰀	鵰	鵫	鶴	鸙	黑			ⅰ
EEF0:	ⅱ	ⅲ	ⅳ	ⅴ	ⅵ	ⅶ	ⅷ	ⅸ	ⅹ	￢	￤	＇	＂			
EF40:																
EF50:																
EF60:																
EF70:																
EF80:																
EF90:																
EFA0:																
EFB0:																
EFC0:																
EFD0:																
EFE0:																
EFF0:																

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
FA40:	ⅰ	ⅱ	ⅲ	ⅳ	ⅴ	ⅵ	ⅶ	ⅷ	ⅸ	ⅹ	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ
FA50:	Ⅶ	Ⅷ	Ⅸ	Ⅹ	￢	￤	＇	＂	㈱	№	℡	∵	纊	褜	鍈	銈
FA60:	蓜	俉	炻	昱	棈	鋹	曻	彅	丨	仡	仼	伀	伃	伹	佖	侒
FA70:	侊	侚	侔	俍	偀	倢	俿	倞	偆	偰	偂	傔	僴	僘	兊	
FA80:	兤	冝	冾	凬	刕	劜	劦	勀	勛	匀	匇	匤	卲	厓	厲	叝
FA90:	﨎	咜	咊	咩	哿	喆	坙	坥	垬	埈	埇	﨏	塚	增	墲	夋
FAA0:	奓	奛	奝	奣	妤	妺	孖	寀	甯	寘	寬	尞	岦	岺	峵	崧
FAB0:	嵓	﨑	嵂	嵭	嶸	嶹	巐	弡	弴	彧	德	忞	恝	悅	悊	惞
FAC0:	惕	愠	惲	愑	愷	愰	憘	戓	抦	揵	摠	撝	擎	敎	昀	昕
FAD0:	昻	昉	昮	昞	昤	晥	晗	晙	晴	晳	暙	暠	暲	暿	曺	朎
FAE0:	朗	杦	枻	桒	柀	栁	桄	棏	﨓	楨	﨔	榘	槢	樰	橫	橆
FAF0:	橳	橾	櫢	櫤	毖	氿	汜	沆	汯	泚	洄	涇	浯			
FB40:	涖	涬	淏	淸	淲	淼	渹	湜	渧	渼	溿	澈	澵	濵	瀅	瀇
FB50:	瀨	炅	炫	焏	焄	煜	煆	煇	凞	燁	燾	犱	犾	猤	猪	獷
FB60:	玽	珉	珖	珣	珒	琇	珵	琦	琪	琩	琮	瑢	璉	璟	甁	畯
FB70:	皂	皜	皞	皛	皦	益	睆	劯	砡	硎	硤	硺	礰	礼	神	
FB80:	祥	禔	福	禛	竑	竧	靖	竫	箞	精	絈	絜	綷	綠	緖	繒
FB90:	罇	羡	羽	茁	荢	荿	菇	菶	葈	蒴	蕓	蕙	蕫	﨟	薰	蘒
FBA0:	﨡	蠇	裵	訒	訷	詹	誧	誾	諟	諸	諶	譓	譿	賰	賴	贒
FBB0:	赶	﨣	軏	﨤	逸	遧	郞	都	鄕	鄧	釚	釗	釞	釭	釮	釤
FBC0:	釥	鈆	鈐	鈊	鈺	鉀	鈼	鉎	鉙	鉑	鈹	鉧	銧	鉷	鉸	鋧
FBD0:	鋗	鋙	鋐	﨧	鋕	鋠	鋓	錥	錡	鋻	﨨	錞	鋿	錝	錂	鍰
FBE0:	鍗	鎤	鏆	鏞	鏸	鐱	鑅	鑈	閒	隆	﨩	隝	隯	霳	霻	靃
FBF0:	靍	靏	靑	靕	顗	顥	飯	飼	餧	館	馞	驎	髙			
FC40:	髜	魵	魲	鮏	鮱	鮻	鰀	鵰	鵫	鶴	鸙	黑				
FC50:																
FC60:																
FC70:																
FC80:																
FC90:																
FCA0:																
FCB0:																
FCC0:																
FCD0:																
FCE0:																
FCF0:																

Appendix 6 Differences Between LCPU and QnUCPU

This section describes the specifications comparison between the LCPU and QnCPU and the precautions for using the existing system.

Appendix 6.1 Specifications comparison

The following table shows a comparison of the specifications between LCPU and QnCPU.

Item	Description	Difference	
		LCPU	QnUCPU
System configuration	Basic configuration	• Power supply module • CPU module • I/O modules and intelligent function modules • END cover	• Power supply module • CPU module • I/O modules and intelligent function modules • Base unit
	Module installation	Modules are connected in order.	Modules are mounted on the base unit.
	Number of modules	• Module whose serial number (first five digit) is "13071" or earlier: Up to 10 modules • Module whose serial number (first five digits) is "13072" or later: Up to 40 modules (L02SCPU, L02SCPU-P, L02CPU, and L02CPU-P: Up to 30 modules) Note that a power supply module, CPU module, display unit, branch module, extension module, RS-232 adapter, and END cover are not included.	Up to 64 modules (Q00UJCPU: 16 modules, Q00U/Q01UCPU: 24 modules, Q02UCPU: 32 modules)
	System extension	• Module whose serial number (first five digit) is "13071" or earlier: Not allowed • Module whose serial number (first five digit) is "13072" or later: Allowed	Allowed
	Power supply module	ERR contact not provided	ERR contact provided
	Multiple CPU system configuration	Not available	Available
	Redundant power supply system configuration	Not available	Available
	Memory card type	SD memory card	• SRAM memory card • Flash memory card • ATA memory card • SD memory card (only QnUDVCPU and QnUDPVCPU)
Built-in function	Serial communication function	Available • L02SCPU • L02SCPU-P • CPU module where the RS-232 adapter or RS-422/485 adapter can be mounted (the one whose serial number (first five digits) is "15102" or later)	Available • Q00U(J)/Q01UCPU • Q02UCPU whose serial number (first five digits) is "10102" or later • Q03UD/Q04UDH/Q06UDH/Q10UDH/Q13UDH/Q20UDH/Q26UDHCPU whose serial number (first five digits) is "13062" or later only

Appendix 6.2 Precautions for utilizing programs

This section describes precautions for applying a QnCPU program to the LCPU.

(1) I/O assignment

Since the LCPU is equipped with built-in functions, the start I/O number assigned by default is different from that of the QnCPU. Therefore, to use the programs of the QnCPU, I/O assignments must be changed even if the configuration of the module has been the same. Change the start I/O number of the connected module in the I/O Assignment tab of the PLC Parameter dialog box according to the program. (☞ Page 364, Appendix 1.2 (9))

(2) Instructions

Some instructions among those applicable to the QnCPU cannot be used in the LCPU. The following table shows the instructions not applicable to the LCPU.

<table>
<tr><th colspan="3">Classification</th><th>Instruction</th></tr>
<tr><td rowspan="4">Application instruction</td><td>File register switching instruction</td><td>File set</td><td>QDRSET(P)</td></tr>
<tr><td rowspan="3">Other instructions</td><td>Reading routing information</td><td>S(P).RTREAD</td></tr>
<tr><td>Registering routing information</td><td>S(P).RTWRITE</td></tr>
<tr><td>File register high-speed block transfer</td><td>RBMOV(P)</td></tr>
<tr><td rowspan="23">Multiple CPU dedicated instruction</td><td rowspan="5">CPU module shared memory access instruction</td><td rowspan="3">Write to host CPU shared memory</td><td>S(P).TO</td></tr>
<tr><td>TO(P)</td></tr>
<tr><td>DTO(P)</td></tr>
<tr><td rowspan="2">Reading from the CPU shared memory of another CPU</td><td>FROM(P)</td></tr>
<tr><td>DFRO(P)</td></tr>
<tr><td rowspan="7">Motion CPU dedicated instruction [*1]</td><td>Motion SFC program startup request</td><td>S(P).SFCS</td></tr>
<tr><td>Servo program startup request</td><td>S(P).SVST</td></tr>
<tr><td>Axis speed change during positioning and JOG operation</td><td>S(P).CHGV</td></tr>
<tr><td>Torque control value change during operation and suspension in real mode</td><td>S(P).CHGT</td></tr>
<tr><td>Current value change of halted axis/synchronized encoder/cam axis</td><td>S(P).CHGA</td></tr>
<tr><td>Write device data to motion controller</td><td>S(P).DDWR</td></tr>
<tr><td>Read device data from motion controller</td><td>S(P).DDRD</td></tr>
<tr><td rowspan="5">Multiple CPU high speed bus compatible Motion CPU dedicated instruction[*2]</td><td>Motion SFC program startup request</td><td>D(P).SFCS</td></tr>
<tr><td>Servo program startup request</td><td>D(P).SVST</td></tr>
<tr><td>Axis speed change during positioning and JOG operation</td><td>D(P).CHGV</td></tr>
<tr><td>Torque control value change during operation and suspension in real mode</td><td>D(P).CHGT</td></tr>
<tr><td>Current value change of halted axis/synchronized encoder/cam axis</td><td>D(P).CHGA</td></tr>
<tr><td rowspan="2">Multiple CPU high speed bus compatible NC dedicated instruction[*2]</td><td>ATC dedicated instruction</td><td>D(P).ATC</td></tr>
<tr><td>Rotation control instruction</td><td>D(P).ROT</td></tr>
<tr><td>Another CPU access instruction[*1]</td><td>Other CPU interrupt program startup</td><td>S(P).GINT</td></tr>
<tr><td rowspan="3">Multiple CPU high speed bus compatible other CPU access instruction[*2]</td><td>Writing devices to another CPU</td><td>D(P).DDWR</td></tr>
<tr><td>Reading devices to another CPU</td><td>D(P).DDRD</td></tr>
<tr><td>Other CPU (motion controller) interrupt program startup</td><td>D(P).GINT</td></tr>
</table>

*1 Only the Q00UCPU, Q01UCPU, and Q02UCPU support these instructions.
*2 The Universal model QCPUs (except the Q00UJCPU, Q00UCPU, Q01UCPU, and Q02UCPU) support these instructions.

Appendix 7 Precautions for Using GX Works2 and Differences with GX Developer

For the precautions for using GX Works2 and differences with GX Developer, refer to the following.

📖 GX Works2 Version 1 Operating Manual (Common)

Appendix 8 Device Point Assignment Sheet

Device name	Symbol	Numeric notation	Number of device points[*1]		Restriction check			
			Points	Range	Size (words)[*2]		Points (bits)[*1]	
Input relay	X	16	8K (8192)	X0000 to X1FFF	÷16	512	×1	8192
Output relay	Y	16	8K (8192)	Y0000 to Y1FFF	÷16	512	×1	8192
Internal relay	M	10	K (　　)	M0 to	÷16		×1	
Latch relay	L	10	K (　　)	L0 to	÷16		×1	
Link relay	B	16	K (　　)	B0000 to	÷16		×1	
Annunciator	F	10	K (　　)	F0 to	÷16		×1	
Link special relay	SB	16	K (　　)	SB0000 to	÷16		×1	
Edge relay	V	10	K (　　)	V0 to	÷16		×1	
Step relay	S	10	8K (8192)	S0 to S8191	÷16	512	×1	
Timer	T	10	K (　　)	T0 to	$\times\frac{18}{16}$		×2	
Retentive timer	ST	10	K (　　)	ST0 to	$\times\frac{18}{16}$		×2	
Counter	C	10	K (　　)	C0 to	$\times\frac{18}{16}$		×2	
Data register	D	10	K (　　)	D0 to	×1		—	
Link register	W	16	K (　　)	W0000 to	×1		—	
Link special register	SW	16	K (　　)	SW0000 to	×1		—	
Total					(29696 or less)			

*1 Up to 32K points can be set for each device. (60K for the internal relay and link relay.)

*2 Enter the values multiplied (or divided) by the number shown in the Size (words) column.

INDEX

A

Acknowledge XY Assignment 366
Allowable momentary power failure time 50
Annunciator (F) 295
Automatic data transfer to the program memory . . . 157

B

Backlight 238
Backup data restoration 192
Backup to SD memory card 184
Batch-disabling executional conditioned device test settings 140
Battery 18
Battery life-prolonging function 198
BCD (binary-coded decimal) 389
BIN (binary code) 389
Block switching method (R) 326
Boot file setting 359
Boot operation 74
Buffer memory monitor 268
Buffer memory monitor/test 265
Buffer memory test 269
Built-in Ethernet port setting 367
Built-in high-speed counter function monitor 256
Built-in I/O function error clear 258
Built-in I/O function monitor 255
Built-in I/O Function Setting 369
Built-in positioning function monitor 256
Built-in serial setting 370

C

Changing a remote password 111
Changing clock data 196
Changing the program execution type 72
Character codes available in the display unit 391
Character string (" ") 343
Checking the executional conditioned device test . . 139
Clearing the file register 329
Clock data read 197
Clock function 196
Clock setting 263
Common pointer 338
Common processing time 384
Compilation 25
Constant scan 85
Constants 342
Contrast adjustment 284
Converting a program 25
Counter (C) 308
CPU module change function with SD memory card 182
CPU monitor/test 241
CPU settings 263
Creating a program 23
Current value changing 29

D

Data acquisition timing setting 146
Data logging file 33
Data logging setting file 33
Data register (D) 311
Debug from multiple programming tools 158
DEC (decimal) 389
Decimal constant (K) 342
Deleting a file register file 329
Deleting a remote password 111
Device clear 249
Device comment 33
Device memory clear 249
Device monitor 245
Device monitor/test 241
Device point assignment sheet 292,406
Device setting 362
Device test 130,248
Differences between LCPU and QnUCPU 403
Direct processing 54
Disable the pop-up display 273
Disabling the executional conditioned device test . . 139
Display unit functions 236
Drive heading 33

E

Edge relay (V) 298
Enable pop-up display 286
END cover 18
END processing 47
Error clear 166
Error display/clear 259
Error history 161
Error time output mode setting 103
Executing a program 28
Executing a sampling trace 147
Executing multiple programs 61
Executional conditioned device test 135
Extended data register (D) 330
Extended link register (W) 330
Extension 33
External input/output forced on/off 131

F

Factors that increase the scan time 385
File name 33
File register (R, ZR) 326
File register clear 249
Files 35
Fixed scan execution type program 69
Floating-point data 389
Forced on/off 251
Format 26
Format PLC memory 26
Formatting a memory 26
Function devices (FX, FY, FD) 314
Function execution time in END processing 382

I

Function input (FX). 314
Function output (FY). 314
Function register (FD). 315

G

Global device. 344

H

HEX (hexadecimal) 389
Hexadecimal constant (H). 342
High-speed timer 300

I

I/O assignment setting 364
I/O No. specification device (U) 340
I/O number assignment 42,43
I/O numbers 23
I/O refresh time 379
Index register (Z) 322
Initial device value 91
Initial execution monitoring time. 64
Initial execution type program 63
Initial processing 45
Initial scan time 64
Initial setting change. 270
Input (X) 293
Input response time setting 102
Instruction execution time 379
Instructions do not operate normally during
online change 153
Intelligent function module device 320
Intelligent function module interrupt 339
Intelligent function module parameters 33,77
Internal relay (M) 294
Internal system devices 314
Internal user devices 290
Interrupt by an internal timer 339
Interrupt by built-in I/O 339
Interrupt pointer (I) 339
Interrupt program 56
Interrupt program list monitor. 122

L

Ladder 76
Language setting 283
Latch data backup 176
Latch data backup to standard ROM 175
Latch function 88
Latch range setting. 88
Latch relay (L) 294
LED control function. 169
LED indication priority. 170
Level navigation. 239
Lighting period setting 285
Link register (W) 312
Link relay (B). 294
Link special register (SW) 313
Link special relay (SB) 298
Local device 345
Local pointer 337
Low-speed timer 300

M

Macro instruction argument device (VD). 341
Main routine program. 46
Memories 31
Menu definition file. 272
Methods for turning off the LEDs 169
Module access devices 320
Module error collection function. 171
Module errors 171
Module model name read 181
Module monitor/test 265
Module settings 270
Monitor. 158
Monitoring local devices 128
Multiple timers. 306

N

Nesting (N) 335
Network parameter 374

O

Online change (files) 152
Online change (ladder mode) 151
Online change based on relative step No. 160
Online change from multiple programming tools . . 160
Options. 283
Other devices 340
Output (Y). 293
Output mode at operating status change
(STOP to RUN) 101
Overflow. 343

P

Parameter. 33
Parameter numbers 350
Parameter setting 350
Parameter-valid drive 34
PAUSE status 48
PLC file setting 356
PLC name setting 353
PLC operation mode at H/W error setting 104
PLC parameter 353
PLC RAS setting 357
PLC system setting 354
PLC user data 33
Pointer (P) 336
Pop up 273
Pop-up display 286
Predefined protocol function 215
Processing time for the instruction (DUTY instruction)
in END processing 379
Program 23,33
Program cache memory 31
Program file 39
Program list monitor. 121
Program memory. 31

Program operation 46
Program setting 360
Programming language 76
Programming tool 18
Project 22

Q

QnUCPU 18

R

Real number (E) 343
Refresh processing 51
Refresh processing of input/output modules or
intelligent function modules 46
Refresh time. 381
Registering executional conditioned device test
settings 136
Registering/canceling display unit menu 272
Remote latch clear 117
Remote operation 112
Remote password 375
Remote PAUSE 114
Remote RESET 116
Remote RUN 112
Remote STOP 112
Restoring backup data. .179
Retentive timer 301
RUN status. 48

S

Sampling trace 143
Sampling trace file 143
Saving a project 30
Saving and restoration of the index register 324
Scan execution type program. 65
Scan time measurement 118
Scan time monitor. 254
Scan time structure 45
Scrolling. 239
SD memory card. 18,32
Self-diagnostic function 161,238
Serial communication function 227
Serial number access method (ZR). 326
Service processing setting 94
Service processing time. 384
SFC. 76
SFC block device (BL). .340
SFC setting 361
Simultaneous monitoring from multiple
programming tools 159
Special register (SD) 316
Special relay (SM). 316
Special value 343
Specification of the timer 299
ST. 76
Standard device register (Z). 323
Standard RAM 32
Standard ROM 32
Standby screen. 238
Standby type program 66
Step relay (S) 299
STOP status 48
Structured ladder 76
Subroutine programs. 46
Symbolic information. 33

T

Timer (T, ST) 299
Timer accuracy. 303
Transfer confirmation to the program memory 32
Trigger condition setting 146

U

Underflow 343
User message 273
User setting system file 159

W

Watchdog timer (WDT) 87
WDT 87
Writing a project 26
Writing Programs in RUN Status 151
Writing to the CPU module 27
Writing to the program memory 31
Writing/reading device data to/from standard ROM
. . . . 180

REVISIONS

*The manual number is given on the bottom left of the back cover.

Print date	*Manual number	Revision
January 2010	SH(NA)-080889ENG-A	First edition
April 2010	SH(NA)-080889ENG-B	Partial correction Section 2.3.1, 3.11.1, 3.21.2, 3.30.1, 3.30.2
June 2010	SH(NA)-080889ENG-C	Partial correction Section 2.10, 3.1, 3.6, 3.17, 3.30, 3.30.2, 5.2.11
January 2011	SH(NA)-080889ENG-D	Partial correction TERMS, Section 2.1.2, 2.1.3, 2.9, 3.1, 3.5, 3.6, 4.7.1 Addition Appendix 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, 1.2.9, 1.2.10, 1.2.11, 1.2.12, Appendix 2
April 2011	SH(NA)-080889ENG-E	Partial correction RELEVANT MANUALS, Section 2.1.3, 2.3.1, 2.3.4, 2.4, 2.6.1, 2.6.2, 2.7, 3.6, 3.13.1, 3.18, 3.20, 3.26, 5.1, 5.2, 5.2.1, 5.2.5, 5.2.7, 5.2.13, 5.2.14, 5.8, Appendix 1.1, 1.3, 2, 3.1 Addition Section 5.4
July 2011	SH(NA)-080889ENG-F	Addition of LCPU models Model Addition L02CPU-P, L26CPU-PBT Partial correction INTRODUCTION, RELEVANT MANUALS, Section 1.2, 2.1.1, 2.1.3, 2.2.1, 2.6.1, 2.11, 3.1, 3.21.3, 3.23, 3.27, 3.30, 3.30.1, 3.30.2, 5.1, 5.7, 5.8, Appendix 1.2.2, Appendix 1.2.9, Appendix 1.2.11, Appendix1.4, Appendix 2, Appendix 3.1, Appendix 3.2
September 2011	SH(NA)-080889ENG-G	Partial correction SAFETY PRECAUTIONS, RELEVANT MANUALS, TERMS, Section 2.2.1, 2.2.2, 4.2.6, 5.2.10, Appendix 1.2.9, 3.1, 3.2, 6.1
October 2011	SH(NA)-080889ENG-H	Partial correction SAFETY PRECAUTIONS, Section 1.7, 2.6.1, 3.1, 3.4, 3.27, 4.2.1, 4.5.1, 5.2, 5.5.1, 5.6.1, 5.7, 5.8, Appendix 2, 3.1 Addition Section 3.33, 3.34

Print date	*Manual number	Revision
May 2012	SH(NA)-080889ENG-I	Partial correction Section 1.5.1, 1.5.2, 1.6, 2.1.1, 2.1.3, 2.2.2, 2.8.1, 2.9, 3.1, 3.3, 3.4, 3.5, 3.6, 3.8, 3.9, 3.10, 3.13.1, 3.13.2, 3.13.3, 3.13.4, 3.21.3, 3.22.1, 3.22.2, 3.23, 3.30.1, 3.30.2, 3.31, 3.32, 3.34, Chapter 4, Section 4.1, 4.5.1, 5.2.13, 5.8, Appendix 1.2.4, 1.2.6, 1.2.9, 1.2.10, 1.2.11, 2, 3.1, 3.2 Addition Section 3.35, 4.7
August 2012	SH(NA)-080889ENG-J	Partial correction Section 3.1, 3.35, Appendix 1.1, 2
February 2013	SH(NA)-080889ENG-K	Revision due to the addition of LCPU models Model Addition L02SCPU, L06CPU, L26CPU Partial correction INTRODUCTION, TERMS, Section 2.1.1, 2.1.3, 2.2.1, 3.21.3, 3.27, 3.28, 3.30, 5.7, Appendix 1.2.2, 1.2.9, 3.1, 3.2
May 2013	SH(NA)-080889ENG-L	Revision due to the addition of LCPU models Model Addition L02SCPU-P, L06CPU-P, L26CPU-P Partial correction INTRODUCTION, TERMS, Section 2.1.1, 2.1.3, 2.2.1, 2.9, 3.21.3, 3.27, 3.28, 3.30, 3.30.2, 3.35, 3.35.2, 3.37, Chapter 4, Section 5.1, 5.7, Appendix 1.2, 3.1, 3.2, 6.1, 6.2
October 2013	SH(NA)-080889ENG-M	Partial correction Section 2.1.1, 3.21.3, 5.11, Appendix 1.2, 2

Print date	*Manual number	Revision
January 2014	SH(NA)-080889ENG-N	Partial correction Section 2.1.1, 2.1.2, 2.1.3, 2.2.1, 3.1, 3.35, 3.37, Appendix 1.1, 1.2, 2, 3.2, 6, 7.1 Addition Section 3.36
June 2014	SH(NA)-080889ENG-O	Partial correction Section 2.1.3, 5.1, 5.2, 5.10, Appendix 1.1, 1.2
October 2014	SH(NA)-080889ENG-P	Partial correction Section 3.1, 3.20, Appendix 2, 3.2 Addition Section 3.21
December 2014	SH(NA)-080889ENG-Q	Partial correction Section 3.1, 3.25, Appendix 1.1, 1.2, 2, 3.2
March 2015	SH(NA)-080889ENG-R	Model Addition NZ1MEM-2GBSD, NZ1MEM-4GBSD, NZ1MEM-8GBSD, NZ1MEM-16GBSD Partial correction TERMS, Section 2.1.1, 3.28, 3.31, 3.36, Chapter 4, Appendix 2 Addition Appendix 3.3
October 2016	SH(NA)-080889ENG-S	Partial correction Section 2.4, 3.13, 3.28, 3.31.1, 3.31.2, 3.36.2, 3.39, 7.2, Appendix 1.2, 2
January 2017	SH(NA)-080889ENG-T	Partial correction RELEVANT MANUALS, 2.3.4, 2.4, 2.6.1, 3.1, 3.6, 3.11, 3.18, 3.31.1, 3.31.2, 3.36.1, 3.36.2, 5.2, 5.2.1, Appendix 1.1, 1.2, 2, 3.1
December 2017	SH(NA)-080889ENG-U	Partial correction Section 3.6, 3.38
June 2018	SH(NA)-080889ENG-V	Partial correction Appendix 2
September 2018	SH(NA)-080889ENG-W	Partial correction Appendix 6
December 2018	SH(NA)-080889ENG-X	Partial correction Appendix 2
April 2019	SH(NA)-080889ENG-Y	Partial correction TERMS Addition GENERIC TERMS AND ABBREVIATIONS
June 2024	SH(NA)-080889ENG-Z	Partial correction Section 5.10.1

Japanese manual version SH-080873-AB

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

(1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.

(2) Even within the gratis warranty term, repairs shall be charged for in the following cases.

1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
2. Failure caused by unapproved modifications, etc., to the product by the user.
3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

(1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.

(2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

TRADEMARKS

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
The SD and SDHC logos are trademarks of SD-3C, LLC.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

SH(NA)-080889ENG-Z(2406)MEE
MODEL: LCPU-U-KP-E
MODEL CODE: 13JZ35

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN