Panasonic®

Motion Controller

GM1 Controller EtherCAT User’s Manual

Operation

WUME-GM1ETCOP-08

2024.4

(MEMO)

Introduction

Thank you for purchasing a Panasonic product. Before you use the product, please carefully read through the user’s manual, and understand it in detail to use the product properly.

Types of Manual

- There are different types of manuals for the GM1 series, as listed below. Refer to the appropriate manual according to your need.

  These manuals can be downloaded from our website: https://industrial.panasonic.com/ac/e/motor/motion-controller/mc/gm1/index.jsp

Manuals for GM1 series

Manual name	Manual code	Description
GM1 Controller EtherCAT User's Manual (Setup Edition)	WUME-GM1ETCSU	Explains wiring between the GM1 and its peripheral devices, installation method, and operation check method.
GM1 Controller EtherCAT User's Manual (Operation Edition)	WUME-GM1ETCOP	Explains how to use GM Programmer and PANATERM Lite for GM, set up each function, create projects, and perform other operations.
GM1 Series Reference Manual (Hardware Edition)	WUME-GM1H	Explains the functions and performance of each GM1 unit.
GM1 Series Reference Manual (Instruction Edition)	WUME-GM1PGR	Explains the specifications of each instruction that can be used with the GM1 Series.
GM1 Series Reference Manual (Analog I/O Unit)	WUME-GM1AIO	Explains the functions and performance of each GM1 Analog Expansion Unit.
GM1 Series Reference Manual (Pulse Output Unit)	WUME-GM1PG	Explains the functions and performance of each GM1 Pulse Output unit.

Copyright / Trademarks

Compliance with IEC 61131

International standard IEC 61131 is the international common standards in which the International Electrotechnical Commission (IEC) defines programming languages for PLC.

IEC defines the following five programming languages:

- Ladder Diagram (LD)
- Structured Text (ST)

- Sequential Function chart (SFC)
- Function Block Diagram (FBD)
- Instruction List (IL)

Glossary

■ GM Programmer

A configuration tool for the GM1 controller. Using GM Programmer makes it possible to set positioning data and various positioning parameters, and perform various monitoring. As this tool is equipped with tool operation mode that starts a motor independently without using user programs, it is convenient especially to verify operations at the time of initial startup.

■ PANATERM Lite for GM

A setup support tool for the MINAS series servo amplifiers manufactured by Panasonic Corporation. When GM Programmer is installed, "PANATERM Lite for GM" is also installed at the same time. By using this tool, parameter setup within servo amplifiers, control status monitoring, setup support, machine analysis, and other operations can be executed on the PC screen.

■ P-point control

Refers to control passing through a "Pass Point". In this manual, this control is referred to as "P-point control" for the sake of convenience.

This method is used when target multi-stage velocities are specified in a sequence of motions.

■ C-point control

Refers to control passing through a "Continuance Point". In this manual, this control is referred to as "C-point control" for the sake of convenience. This method is used to execute consecutive E-point controls by one-time startup.

■ E-point control

Refers to movement up to an "End Point". In this manual, this control is referred to as "E-point control" for the sake of convenience. This method is used for single-speed acceleration / deceleration control. It is also called "trapezoidal control".

■ Automatic operation

An operation that is automatically performed. It means position control.

■ Manual operation

An operation that is performed at initial startup or during adjustment. Home return, JOG operation, and pulser operation are manual operations.

■ Position control

A generic term for E-point control, P-point control, and C-point control. For each control, control for single axes and interpolation control for multiple axes can be performed. Interpolation control can be selected from 2-axis linear interpolation, 2-axis circular interpolation, and 3-axis linear interpolation.

■ Home return

The reference position for positioning is called a home position and an operation to travel to a home position is called home return. Each axis is moved to the preset home position and the coordinates of the home position are defined as absolute position zero. The motor rotation is reversed automatically when the limit input (+) or the limit input (-) is input and the home position or near home position is searched to return to the home position automatically.

■ JOG operation

Refers to an operation in which the motor is rotated only while operation commands are being input. This is used to forcibly rotate the motor using inputs from external switches during startup or adjustment, for example. This can also be applied to unlimited feed.

■ Limit input (+), limit input (-)

A limit switch input that is used to limit the motor movement. Limit input (+) is the limit point on the side where the elapsed value increases and limit input (-) is the limit point on the side where the elapsed value decreases.

■ Dwell time

For E-point control, the time from the completion of a position command until the operation done contact turns ON can be specified as a dwell time. For C-point control, similarly, the time from deceleration stop until execution of the next positioning table can be specified as a dwell time.

■ Soft limit

It is possible to set software limits on the absolute coordinates managed by the GM1 controller. Decelerates and stops according to the deceleration of the dynamic limit so as not to exceed the range of the soft limit. The deceleration time can be set individually.

■ Torque control

The output torque of the servo amplifier can be limited arbitrarily.

■ Servo ON / Servo OFF

The operation that changes the servo free state to a servo lock state is called "servo ON", and the operation that changes the servo lock state to a servo free state is called "servo OFF".

■ Linear interpolation

Interpolation control that controls straight lines as loci for the operations of 2-axis motors with grouped X-axis and Y-axis or 3-axis motors with grouped X-axis, Y-axis, and Z-axis. There are two setting methods, which are a composite speed specification and long axis speed specification.

■ Circular interpolation

Interpolation control that controls arcs as loci for the operation of 2-axis motors with grouped X-axis and Y-axis. There are two setting methods, which are a center point specification and pass point specification.

■ Edge detection

One of the methods for detecting the request signals allocated to this unit. It executes each requested process by detecting a trigger that is the rising edge when the request signal turns ON.

Therefore, the next request cannot be accepted until the current request signal turns OFF.

(MEMO)

Table of Contents

1 Before Using This Product

1.1 Safety Precautions

This section explains important rules that must be observed to prevent personal injury and property damage.

- Injuries and damages that may occur as a result of incorrect use are classified into the following levels and safety precautions are explained according to the level.

⚠ **WARNING**	Indicates that there is a risk of death or serious injury
⚠ **CAUTION**	Indicates that there is a risk of minor injury or property damage

⊘	Indicates an action that is prohibited
❗	Indicates an action that must be taken

	⚠ **WARNING**
❗	● Take safety measures outside this product to ensure the safety of the entire system even if this product fails or an error occurs due to external factors.
⊘	● Do not use this product in atmospheres that contain flammable gases. Doing so may result in explosion.
⊘	● Do not throw this product into the fire. Doing so may cause the batteries or other electronic parts to explode.

	⚠ **CAUTION**
❗	● To prevent abnormal heat generation or smoke generation, use this product with some leeway from the guaranteed characteristics and performance values of the product.
⊘	● Do not disassemble or modify this product. Doing so may result in abnormal heat generation or smoke generation.
⊘	● Do not touch any terminals while the power is on. Doing so may result in electrical shock.
❗	● Configure emergency stop and interlock circuits outside this product.
❗	● Connect wires and connectors properly. Failure to do so may result in abnormal heat generation or smoke generation.
⊘	● Do not perform work (such as connection or removal) with the power turned on. Doing so may result in electrical shock.
❗	● If this product is used in any way that is not specified by Panasonic, its protection function may be impaired.
❗	● This product has been developed and manufactured for industrial use only.

1.2 Security precautions

When using this product, the following damages may occur.

(1) Information leakage or outflow through this product

(2) Fraudulent operation of this product by a malicious third party

(3) Obstructing or stopping this product by a malicious third party

Sufficient network security measures, including the following measures, should be taken at your own risk to prevent such damages.

- Use this product on a network where safety is secured by using a firewall.
- When using this product on a system where a PC is connected, make sure that checking and cleaning of infection by computer virus or malicious program is performed periodically.
- In order to prevent malicious attacks, set user name and password to limit users who can log in.
- To prevent leakage of authentication information (user name, password), FTP server information, etc. on the network, take measures such as restricting access by user authentication.
- Be sure to close all browsers immediately after accessing this product as an administrator.
- Periodically change the administrator's password.
- Do not install this product in a location where the product or cables can be easily damaged.
- Furthermore, it is recommended that the product be used in an environment that has VPN (Virtual Private Network) or leased line network.

In addition, if this product is discarded, transferred, repaired, etc., important information recorded on this product or the SD memory card used may remain. At your own risk, please be careful when handling it, such as erasing it.

If you use this product, important information such as confidential information may remain on the terminal. When this product is no longer needed, take measures such as uninstalling this product from the terminal and deleting related files under the customer's responsibility.

1.3 Handling Precautions

- **In this manual, the following symbols are used to indicate safety information that must be observed.**

Stop	Indicates an action that is prohibited or a matter that requires caution.
!	Indicates an action that must be taken.
Info.	Indicates supplemental information.
Note	Indicates details about the subject in question or information useful to remember.
1 2 Procedure	Indicates operation procedures.

1.4 Software License Agreement

Panasonic Industry Co., Ltd. (“PANASONIC”) grants to you a license to use this Software on condition that you accept this Agreement. You must read this Software License Agreement (this “Agreement”) carefully before using this Software. Only in case that you accept this Agreement, you may start your use of this Software.

Your unsealing the package of this Software, or your downloading, installing or launching this Software or the like shall be deemed as your acceptance of this Agreement.

The Software includes not only proprietary computer programs owned by or licensed by PANASONIC but also open source software programs. As for the open source software programs, refer to the detailed terms and conditions thereof shown in the installation package of the Software. Should a discrepancy arise between any of the terms of this Agreement and any open source software program license statement, the open source software program license statement shall take precedence over this Agreement.

Article 1 Grant of License

PANASONIC hereby grants to you a non-exclusive license to use this Software only in combination with PANASONIC product(s) specified in the manual of this Software (the “Product”) in accordance with the terms of this Agreement. You may not use this Software in connection with products of any third party other than PANASONIC.

Article 2 Restrictions

You may NOT:

1. Modify, reverse engineer, decompile, or disassemble this Software, except where the terms and conditions of open source software program license statement (including, but not limited to, GPL and LGPL) apply.

   Should a defect of the Software arise owing to your modification, reverse engineering, decompiling, or disassembling, to the extent permitted under the law,PANASONIC shall not assume any responsibility for such defect,
2. Use this Software by methods or for purposes other than those specified in the manual of this Software provided by PANASONIC, nor
3. Distribute, rent, lease or otherwise transfer this Software to any third party; provided,

however, that you may assign the rights to use this Software under this Agreement along with the Product on the condition that the assignee agrees to be bound by all the terms of this Agreement. In the case of such assignment, you must deliver any and all the copies of this Software and all the accompanying materials to the assignee and you may not retain any copies of this Software including backups.

Article 3 Disclaimer

1. PANASONIC HEREBY DISCLAIMS ALL OTHER WARRANTIES ON THIS SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MARCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.
2. UNDER NO CIRCUMSTANCES SHALL PANASONIC BE LIABLE FOR ANY DAMAGES (INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR SPECIAL OR WHATSOEVER) ARISING OUT OF THE USE OF THIS SOFTWARE, INABILITY TO USE THIS SOFTWARE, DEFECTS IN THIS SOFTWARE (e.g., BUGS, SECURITY HOLES, AND MALFUNCTION), OR OTHERWISE IN CONNECTION WITH THIS SOFTWARE.

Article 4 Term

1. This Agreement shall come into effect upon your unsealing the package of this Software, or your downloading, installing or launching this Software or the like.

2. PANASONIC may terminate this Agreement immediately, if you breach any of the provisions of this Agreement.
3. You shall, at your own costs, return, delete or destroy this Software and any of its copies within four (4) weeks after termination of this Agreement.

Article 5 Export Control

You shall comply with all laws and regulations regarding export control under any competent jurisdiction, including but not limited to the Japanese Foreign Exchange & Foreign Trade Control Law, the export control regulations based on resolutions of the United Nations Security Council, etc. If any license or appropriate approval from a governmental authority is required under the applicable laws, you may not export this Software without such approval to any countries either directly or indirectly. Furthermore, you shall neither use nor sell this Software for military purposes either directly or indirectly.

Article 6 Intellectual Property Rights

Except the open source software program (including, but not limited to, GPL and LGPL), all intellectual property rights in this Software, including the copyright, belong to PANASONIC and/or the licensors of PANASONIC.

Article 7 Upgrade of this Software

1. Release of future upgrades or updates of this Software is not guaranteed and left to the sole discretion of PANASONIC. Furthermore, PANASONIC may charge fees for upgrading or updating of this Software.
2. If any upgrades or updates are provided to you either for fees or for free, such upgrades or updates shall be deemed as a part of this Software and shall be governed by this Agreement, unless PANASONIC designates otherwise at the time of provision of such upgrades or updates.

Article 8 Limitation on Liability

AGGREGATE LIABILITIES OF PANASONIC IN CONNECTION WITH THIS AGREEMENT OR THIS SOFTWARE SHALL IN NO EVENT EXCEED TEN THOUSAND (10,000) YEN.

Article 9 Governing Law and Jurisdiction

1. This Agreement shall be governed by the laws of Japan.
2. Should any dispute arise from or in connection with this Agreement, Osaka District Court, Japan shall exclusively have the jurisdiction over such dispute.

This Software consists of the following types of software.

1. Software developed independently by PANASONIC
2. Software owned by and licensed by the third party
3. Software licensed under GNU General Public License Version 2.0 (GPL V2.0)
4. Software licensed under GNU Lesser General Public License Version 2.0 (LGPL V2.0) or Version 2.1 (LGPL V2.1)
5. Open source software licensed on conditions other than those of GPL V2.0, LGPL V2.0, or LGPL V2.1

Software in categories 3. - 5. above is distributed with the expectation of effectiveness as a single piece of software, but there is no guarantee provided, including implied guarantees regarding viability as a product and/or suitability for specific purposes. For details, Please refer to the detailed terms and conditions thereof shown in the installation package of the Software.

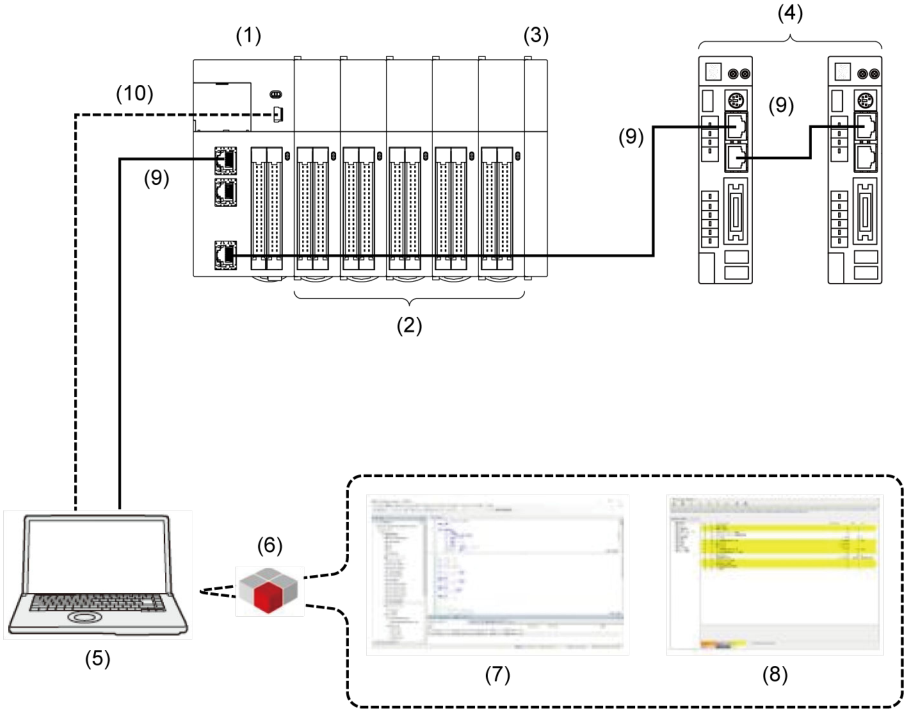
For at least three years following the release of the Product, PID will provide, at customer's expense, complete machine-readable source code for software licensed under GPL V2.0, LGPL V2.0, LGPL V2.1, or a license based on other conditions that meet source code disclosure requirements, along with information on the respective copyright holders, to customers who contact us at the following e-mail address.

【Contact e-mail address: oss-cd-request@gg.jp.panasonic.com】

2 Operation Flow

2.1 System Configuration Diagram

The figure below shows the configuration of the GM1 series motion controller (controller and expansion units), servo amplifiers, and PC. GM Programmer and PANATERM Lite for GM communicate with the GM1 Controller via Gateway.



No.	Name
(1)	GM1 Controller
(2)	Expansion unit
(3)	End unit
(4)	Servo amplifier
(5)	PC (on which GM Programmer and PANATERM Lite for GM are installed)
(6)	Gateway, CodeMeter
(7)	GM Programmer
(8)	PANATERM Lite for GM
(9)	Ethernet cable(Note 1)
(10)	USB cable(Note 1)

(Note 1) Use either Ethernet cables or USB cables.

Info.

- To operate the system, you must install GM Programmer and PANATERM Lite for GM on the PC.
- When GM Programmer is installed, MINAS setup support software "PANATERM Lite for GM", Gateway (the application that connects GM Programmer and the GM1 Controller), and CodeMeter are installed at the same time.

2.2 Work Flowchart

The following table explains the workflow from installation of the GM1 controller through to its operation.

Step	Description		Reference
1	Install GM Programmer and PANATERM Lite for GM.		"GM1 Controller EtherCAT User's Manual (Setup Edition)"
2	Make preparations for the servo amplifiers.		
	2-1	Connect the servo amplifiers and the PC.	
	2-2	Install the USB driver on the PC.	
	2-3	Configure initial settings for the servo amplifiers.	
	2-4	Disconnect the servo amplifiers from the PC.	
3	Connect the GM1 controller and EtherCAT slaves.		
4	Connect the GM1 controller and the GM Programmer.		
	4-1	Connect the GM1 controller and the PC with a cable.	
	4-2	Create a new project.	
	4-3	Make communication settings.	
	4-4	Add and set up EtherCAT slaves.	
	4-5	Make basic settings of the EtherCAT axis	
	4-6	Connect the GM1 controller, PC, and EtherCAT slaves and perform an operation check.	
	4-7	Log in to the GM1 controller.	
	4-8	Log out from the GM1 controller.	
5	Connect the GM1 controller and PANATERM Lite for GM.		
	5-1	Set up the servo amplifier connected to the GM1 controller.	
	5-2	Write parameters to the servo amplifier.	
	5-3	Write objects to the servo amplifier.	
6	Prepare for operation.		
	6-1	Check if safety circuit design is implemented.	
	6-2	Check wiring for each device.	
	6-3	Perform an operation check.	
7	Using the GM Programmer, make settings for GM1 parameters, motion control, unit control, and communication function.		
	7-1	Make settings for the GM1 controller.	"P.5-3"
	7-2	Make settings for the motion control.	"P.5-4"
	7-3	Make settings for the unit control.	"P.5-13"
	7-4	Make settings for the communication function	"P.5-20"
8	Create programs with GM Programmer.		"P.6-1"
	8-1	Create objects (POU objects) for a program.	"P.6-15"
	8-2	Select a programming language (LD, ST, SFC, FBD, IL, or CFC program) and enter a program.	"P.6-16"

Step	Description		Reference
	8-3	Set variables.	"P.7-45"
9	Set up the GM1 controller in GM Programmer.		
	9-1	Make time setting.	"P.8-12"
	9-2	Log in to the GM1 controller.	"P.8-21"
	9-3	Log out from the GM1 controller.	"P.8-23"
	9-4	Upload the source.	"P.8-29"
10	Configure security settings with GM Programmer.		"P.10-3"
	10-1	Configure user management settings.	"P.10-3"
	10-2	Configure encryption and signature settings.	"P.10-16"
	10-3	Configure write-protection settings.	"P.10-24"

(MEMO)

3 Overview of the GM Programmer

3.1 System Requirements

3.1.1 Usage Environment of the GM Programmer

Programming software

Product name	Applicable language
GM Programmer	Japanese / English / Chinese

(Note 1) When GM Programmer is installed, MINAS setup support software"PANATERM Lite for GM"is installed at the same time.

Software operating environment

Item	Description
OS	Microsoft(R) Windows(R) 10：32bit/64bit Microsoft(R) Windows(R) 11：64bit
PC	PC with the following installed: ● Microsoft.NET Framework 4.6.1 or higher ● Microsoft Visual C++ 2010 SP1 Redistributable Package (x86) ● Microsoft Visual C++ 2010 SP1 Redistributable Package (x64) ● Microsoft Visual C++ 2013 Redistributable Package (x86) ● Microsoft Visual C++ 2013 Redistributable Package (x64) ● Microsoft Visual C++ 2015 Update 3 Redistributable Package (x86) ● Microsoft Visual C++ 2015 Update 3 Redistributable Package (x64)
HDD	At least 4 GB of free space
Memory	At least 8 GB
Communication port	LAN port (for Ethernet connection) USB 2.0 port (for USB connection)

3.2 Installation and Uninstallation

3.2.1 Installing GM Programmer

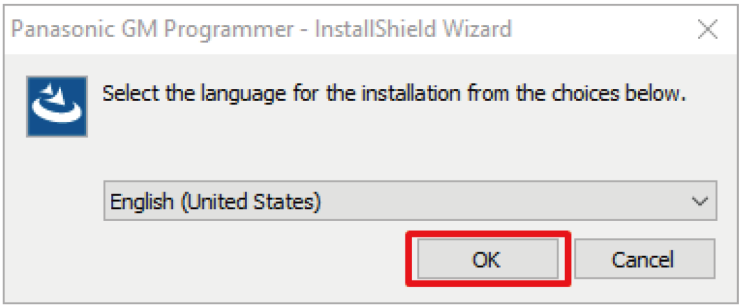
Before installing the GM Programmer on a PC, log on to the PC as an account with Administrator privileges.

If other applications are running, be sure to close all the applications before installing GM Programmer.

1 2 Procedure

1. Double-click "setup.exe".

The following window will be displayed. Click [OK].



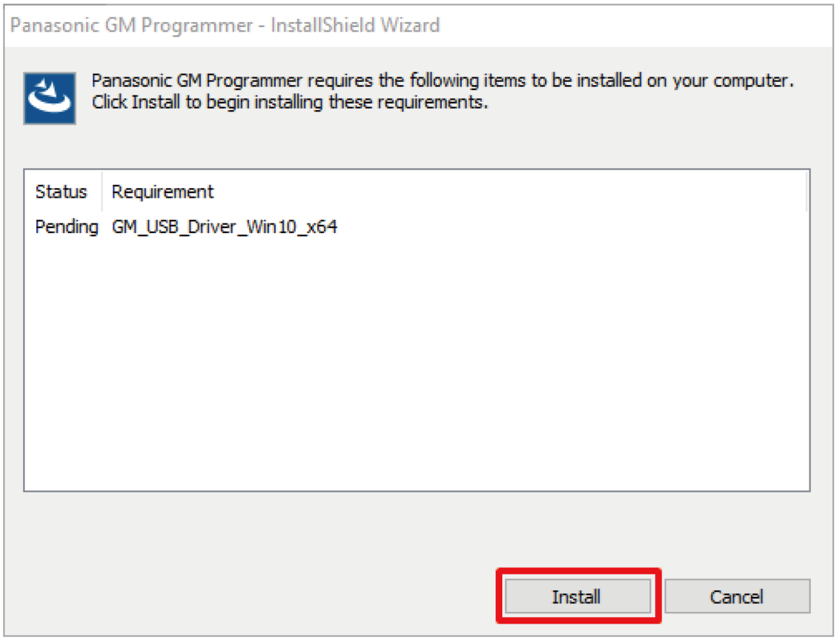
Panasonic GM Programmer - InstallShield Wizard

Select the language for the installation from the choices below.

English (United States)

2. The following window will be displayed. Click [Install].

The display content differs according to the PC environment that you use. (This window may not be displayed at all, depending on the situation.)

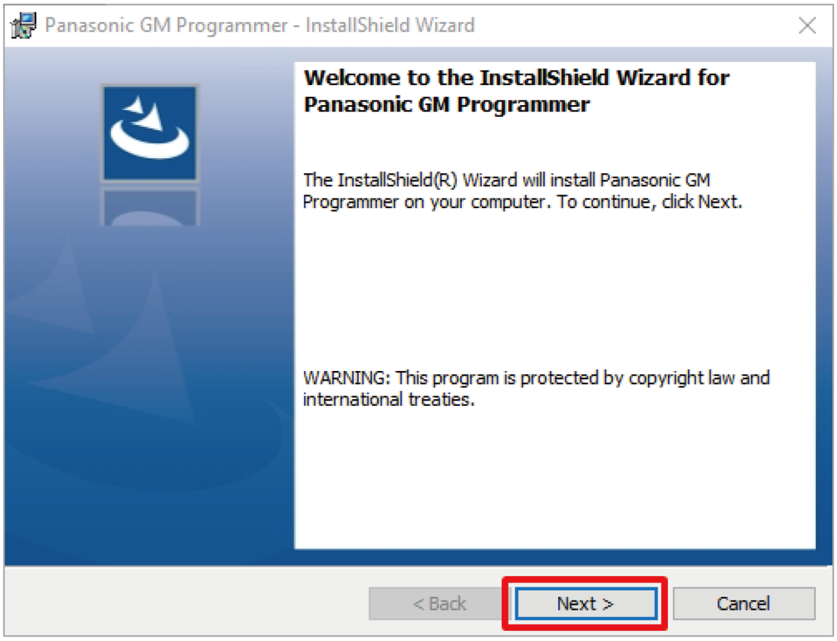


Panasonic GM Programmer - InstallShield Wizard

Panasonic GM Programmer requires the following items to be installed on your computer. Click Install to begin installing these requirements.

Status	Requirement
Pending	GM_USB_Driver_Win10_x64

3. The following window will be displayed. Click [Next].



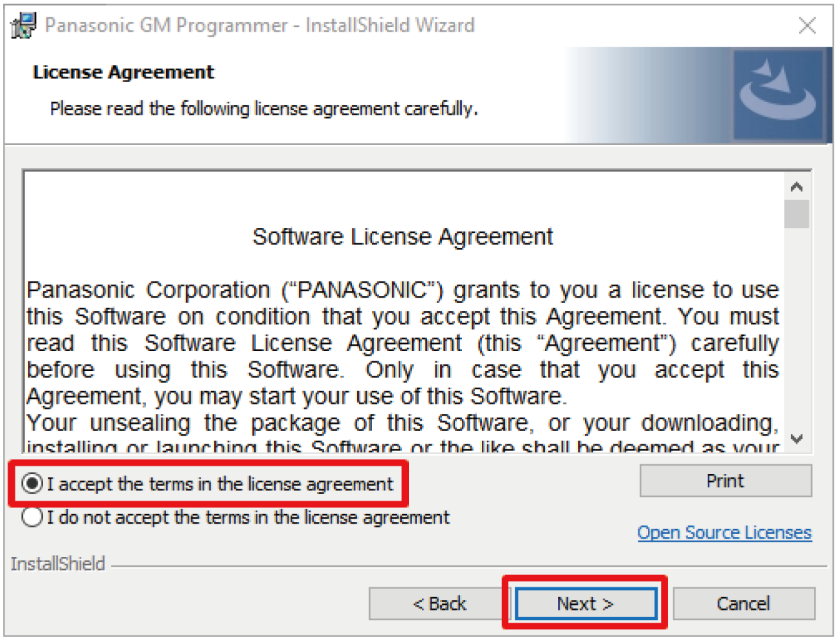
Panasonic GM Programmer - InstallShield Wizard

Welcome to the InstallShield Wizard for Panasonic GM Programmer

The InstallShield(R) Wizard will install Panasonic GM Programmer on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

4. The following window will be displayed. Select [I accept the terms in the license agreement] and click [Next].



Panasonic GM Programmer - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

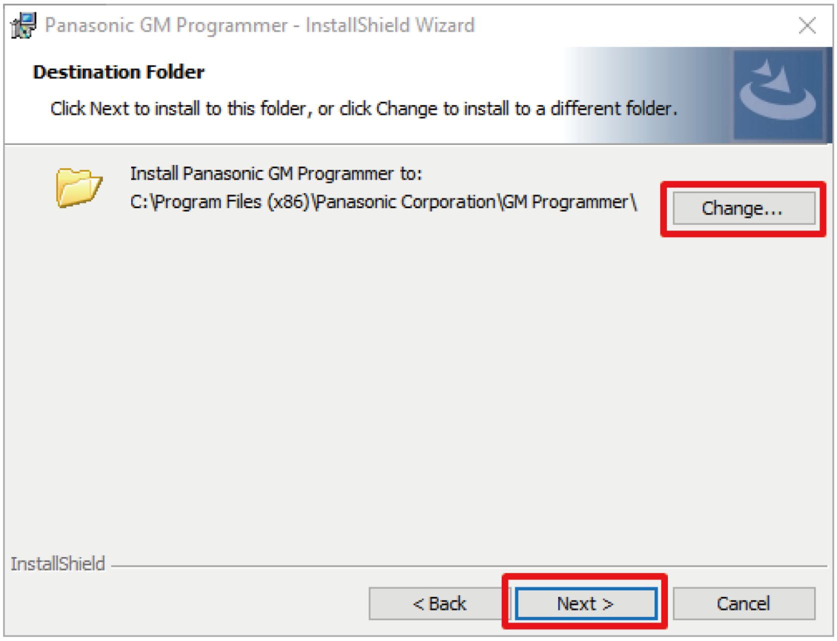
Software License Agreement

Panasonic Corporation ("PANASONIC") grants to you a license to use this Software on condition that you accept this Agreement. You must read this Software License Agreement (this "Agreement") carefully before using this Software. Only in case that you accept this Agreement, you may start your use of this Software.
Your unsealing the package of this Software, or your downloading, installing or launching this Software or the like shall be deemed as your

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

5. The following window will be displayed. If you change the installation destination folder, click [Change] and specify a desired installation destination. If you do not change the installation destination folder, click [Next].



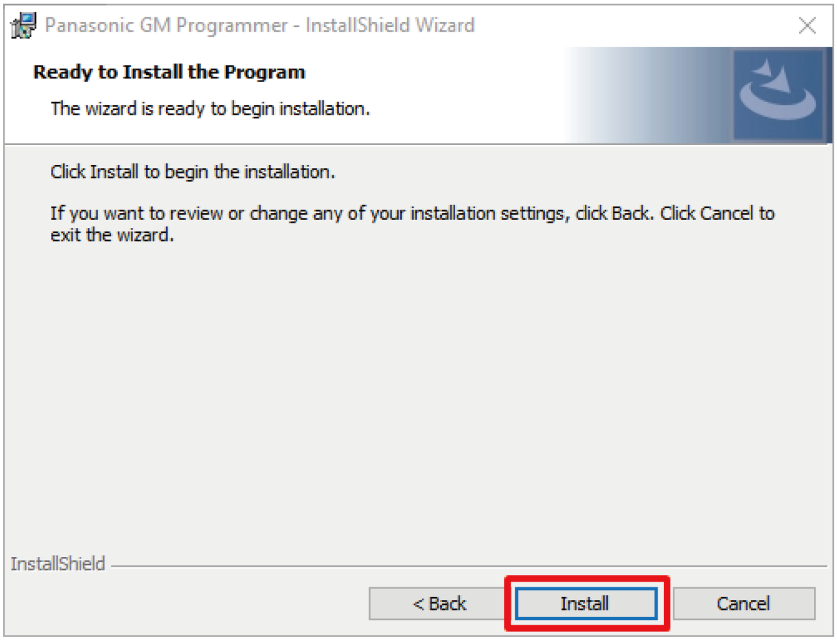
Panasonic GM Programmer - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install Panasonic GM Programmer to:
C:\Program Files (x86)\Panasonic Corporation\GM Programmer\

6. The window below will be displayed. Click [Install] to start the installation.



Panasonic GM Programmer - InstallShield Wizard

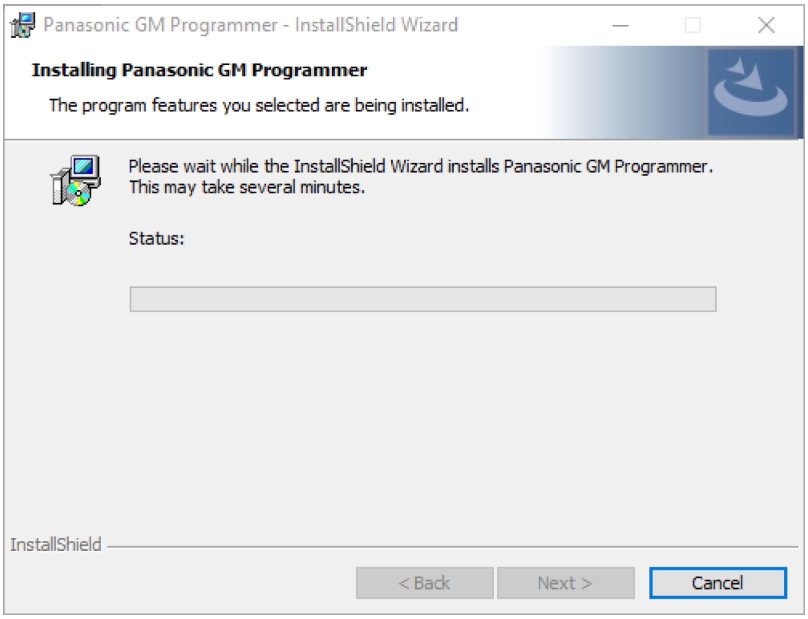
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

7. The following window will be displayed while the installation is in progress.


Panasonic GM Programmer - InstallShield Wizard
Installing Panasonic GM Programmer
The program features you selected are being installed.
InstallShield
< Back
Next >
Cancel

Please wait while the InstallShield Wizard installs Panasonic GM Programmer. This may take several minutes.

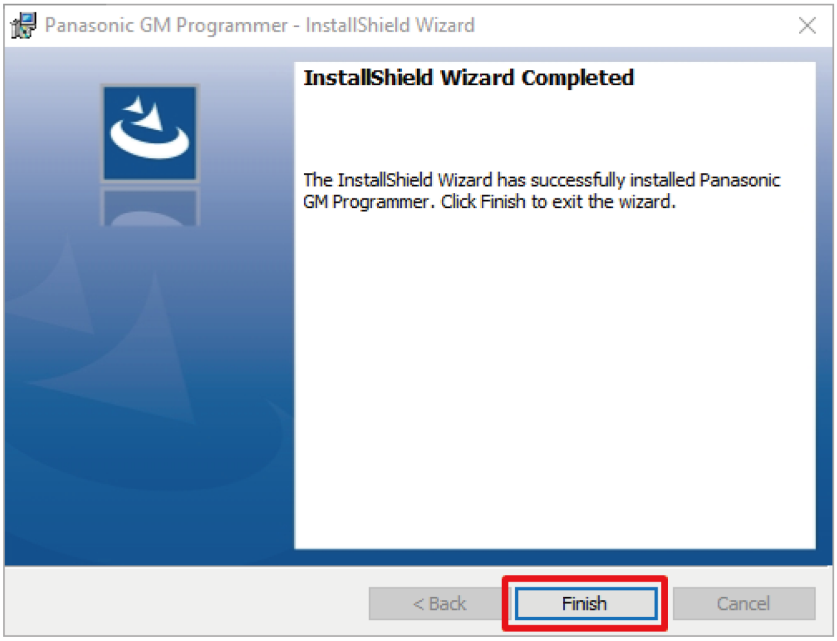
Status:

Following this installation, the three packages below will be installed. (The segments indicated by * differ according to the version of the software.)

- CODESYS SoftMotion*.*.*.*_P
- GMPLibrary (*.*.*.*)
- PANATERM-Lite for GM V*.*

These packages take a long time to install. Take care not to click [Cancel] while the installation is in progress.

8. When the installation of all the packages is completed, the following window will be displayed. Click [Finish].


Panasonic GM Programmer - InstallShield Wizard
InstallShield Wizard Completed
< Back
Finish
Cancel

The InstallShield Wizard has successfully installed Panasonic GM Programmer. Click Finish to exit the wizard.

This completes the installation procedure.

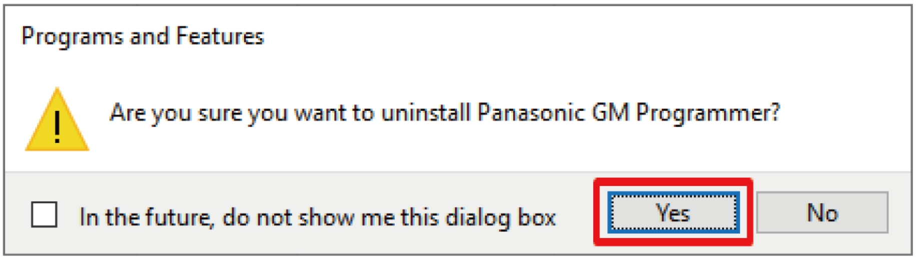
Info.

- When the GM Programmer is installed, PANATERM Lite for GM, Gateway (CODESYS Gateway), and CodeMeter applications are installed at the same time.

3.2.2 Uninstalling GM Programmer

Procedure

1. From the Start menu, select **Windows System**>**Control Panel**, and then click "Uninstall a program".
   A list of installed programs will be displayed.
2. Double-click "Panasonic GM Programmer".
   The following window will be displayed. [Yes]



Programs and Features

Are you sure you want to uninstall Panasonic GM Programmer?

☐ In the future, do not show me this dialog box

3. Click the [Yes] button.
   The GM Programmer will be uninstalled.

Info.

- When the GM Programmer is uninstalled, PANATERM Lite for GM and Gateway are also uninstalled at the same time.
- CodeMeter will not be uninstalled at this time. Uninstall it separately.

3.3 Version System

This chapter explains versions used for GM Programmer and the GM1 controller.

3.3.1 Version Relationship

This chapter explains how GM Programmer versions, project versions, and firmware versions are related with one another.

■ Objects subject to version management

GM Programmer, project files, and firmware for the GM1 controller are each managed with four-digit version numbers.

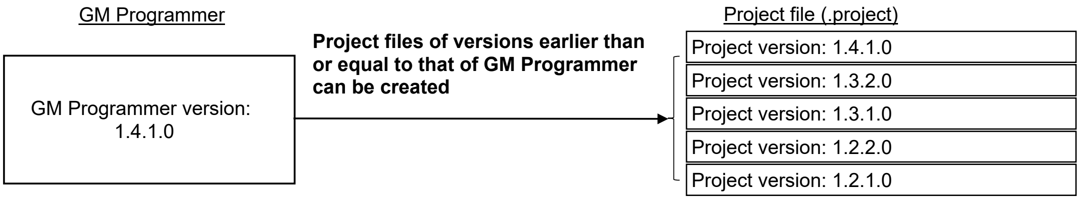
Each version number is made up of four digits (A . B . C . D).

Version type	Description and way of checking version
GM Programmer version	The version of the GM Programmer installer can be checked by following "3.8.3 Version Display Function".
Project version	The version of project files created by GM Programmer. You can check it on the “Information” tab of the Device object. (described later)
Firmware version of GM1 controller	Please refer to "15.6.1 Checking Unit Configuration" for the firmware version installed on the GM1 controller.

■ Relationship between GM Programmer versions and project versions

The version of GM Programmer indicates versions of projects that can be created. (The GM Programmer version provides only backward compatibility)

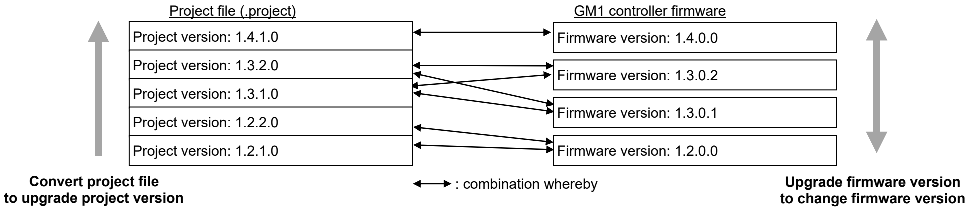
For example, with GM Programmer version 1.4.1.0, project version 1.4.1.0 and earlier of project files can be created.



■ Relationship between project versions and GM1 controller firmware versions

When connecting to the GM1 controller, the first two digits (A . B) of the project version and the firmware version are required to match.

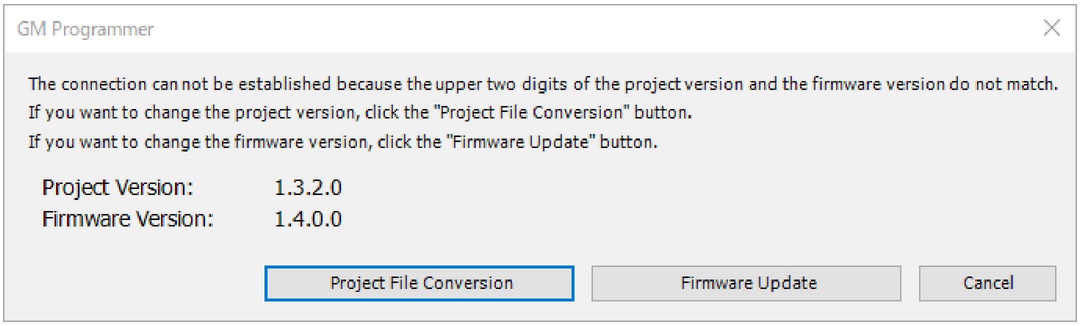
Combinations whereby connection is allowed are shown below.


Project file (.project)
Project version: 1.4.1.0
Project version: 1.3.2.0
Project version: 1.3.1.0
Project version: 1.2.2.0
Project version: 1.2.1.0
GM1 controller firmware
Firmware version: 1.4.0.0
Firmware version: 1.3.0.2
Firmware version: 1.3.0.1
Firmware version: 1.2.0.0
Convert project file to upgrade project version
: combination whereby
Upgrade firmware version to change firmware version

For example, if your PC with a project of project version 1.3.2.0 is connecting to the GM1 controller whose firmware version is 1.4.0.0, a message indicating connection failure appears because the first two digits of the versions do not match 1.3 ≠ 1.4.

To upgrade the project version and thereby make them match, perform "4.15.1 Converting Project File (.project) Version".

To downgrade the firmware version and thereby make them match, perform "15.6.2 Unit Version Upgrade" so that the firmware version is downgraded.



GM Programmer

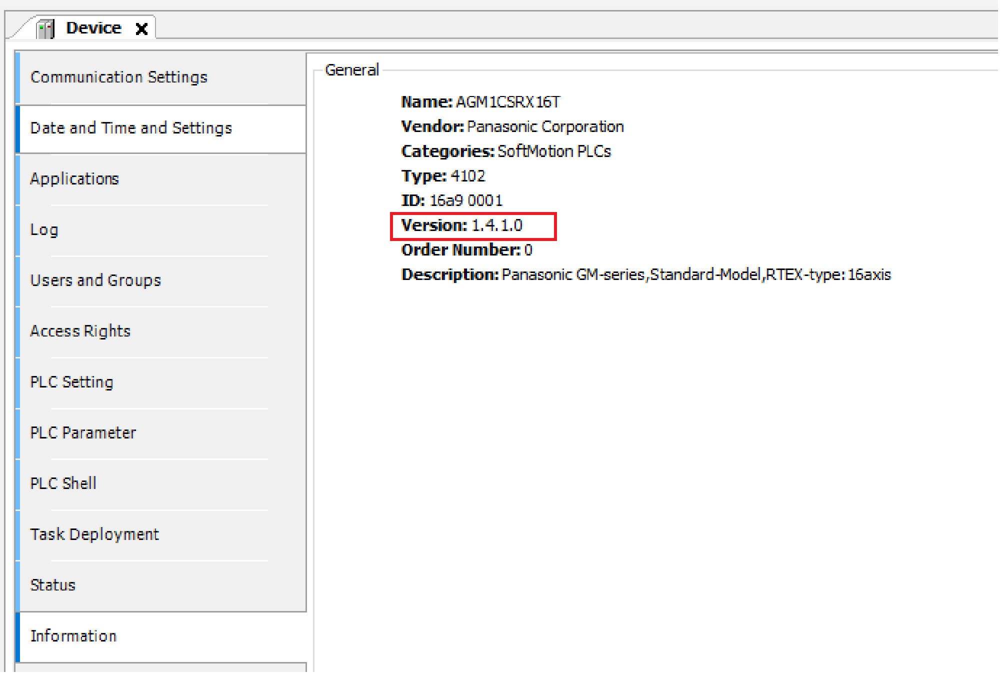
The connection can not be established because the upper two digits of the project version and the firmware version do not match.
If you want to change the project version, click the "Project File Conversion" button.
If you want to change the firmware version, click the "Firmware Update" button.

Project Version: 1.3.2.0
Firmware Version: 1.4.0.0

Note

- To create a library for users, select a template whose version is the same as the project version.

  Example: When a library is used for projects of project version 1.4.1.0, select template version 1.4.1.0.
- The project version of a project you are editing can be confirmed at “Version” on the “Information” tab of the Device object.



General

Name: AGM1CSRX16T
Vendor: Panasonic Corporation
Categories: SoftMotion PLCs
Type: 4102
ID: 16a9 0001
Version: 1.4.1.0
Order Number: 0
Description: Panasonic GM-series,Standard-Model,RTEX-type:16axis

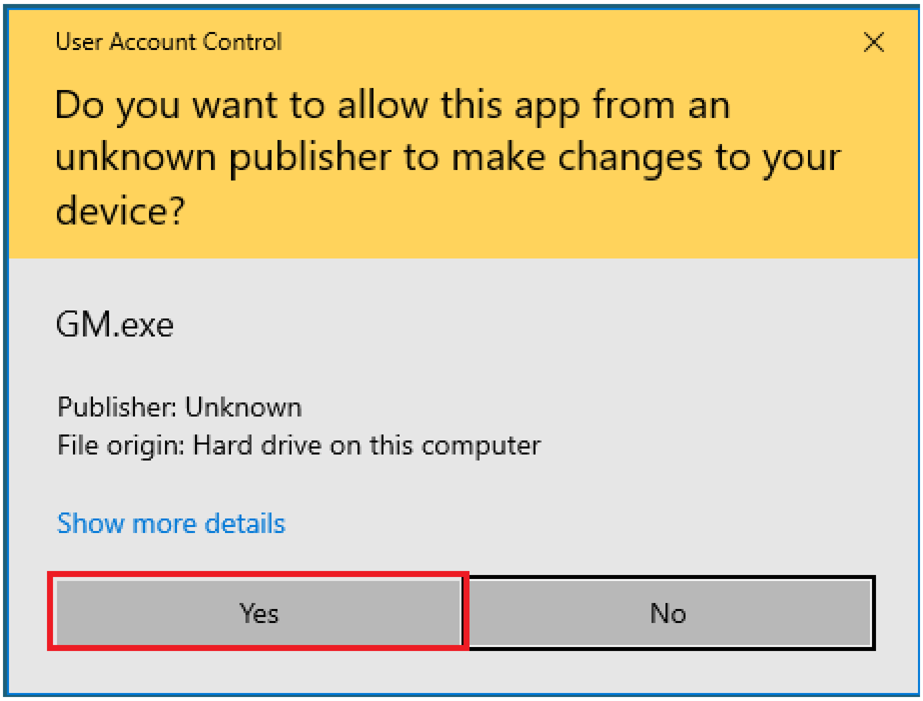
3.4 Basic Operations

This section explains how to start and quit GM Programmer.

3.4.1 How to start

1 2 Procedure

1. Click the [Start] button and select **Panasonic Corpration**>**GM Programmer**.
The "User Account Control" dialog box will be displayed. Click [Yes].



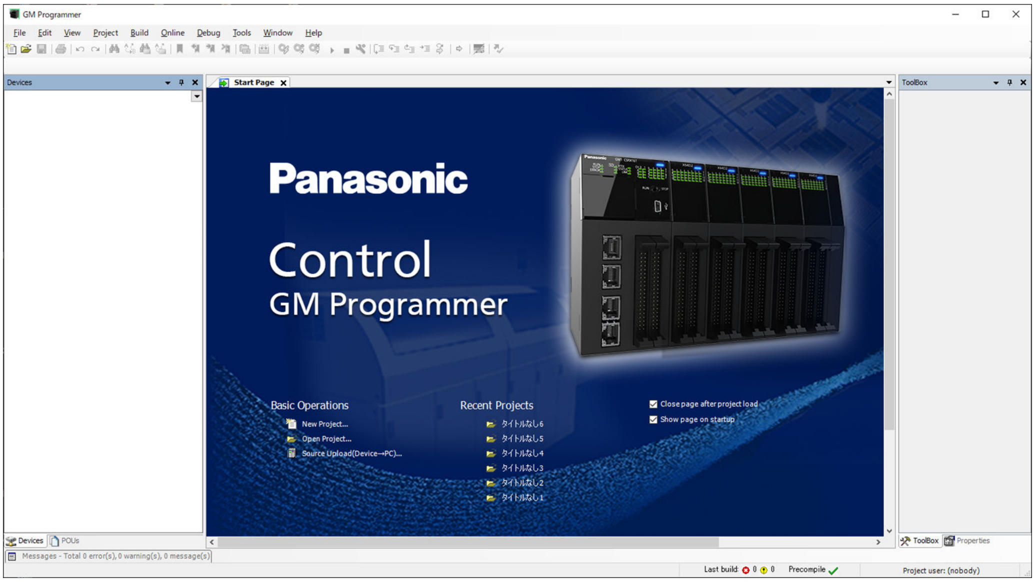
User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

GM.exe

Publisher: Unknown
File origin: Hard drive on this computer

GM Programmer will be started.

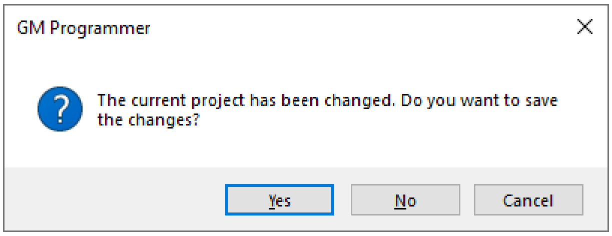


3.4.2 How to quit

- Before closing GM Programmer, be sure to save any project files that you are editing and must save.

Procedure

1. From the menu bar, select **File**>**Exit.**
   If changes have not been saved, the following window will be displayed.
   If exiting without saving, select [No].
   If changes need to be saved, select [Yes] to perform the save process.


GM Programmer

The current project has been changed. Do you want to save the changes?

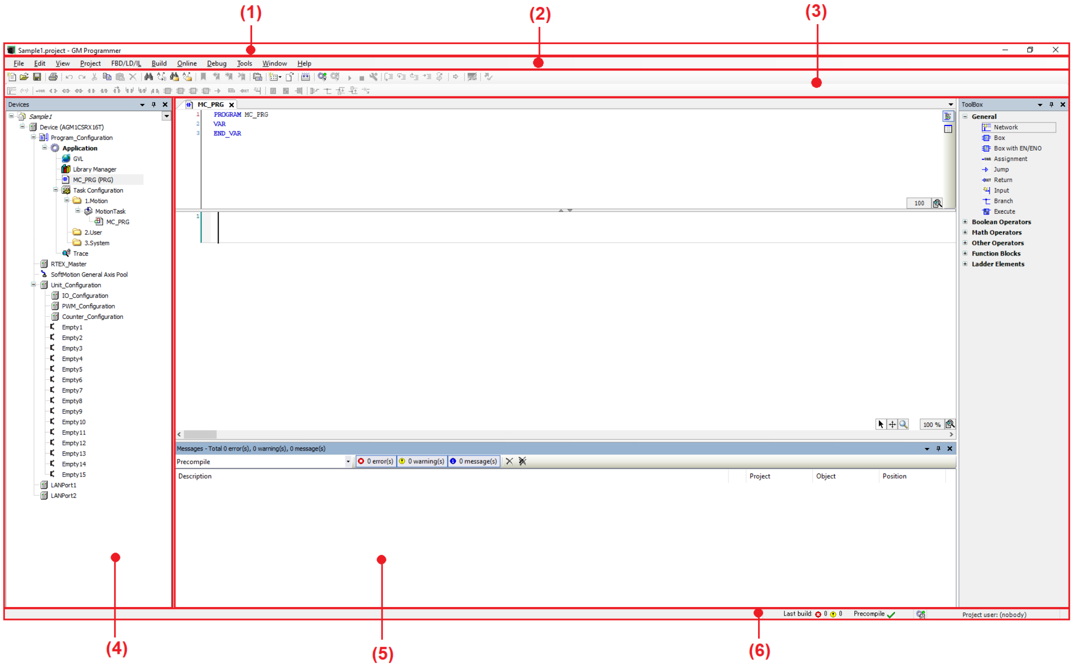
2. Click the [Yes] button.
   GM Programmer will be closed.

Info.

- You can also close GM Programmer by clicking the [×] button on the title bar.

3.5 Component Names

This section presents the name and display content of each component of GM Programmer.

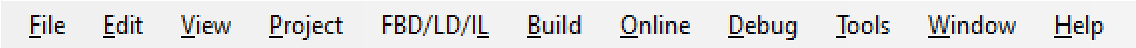


```
PROGRAM MC_PRG
VAR
END_VAR
```

No.	Name	Description
(1)	Title bar	The title bar displays the project file name, [minimize] button, [maximize] button, and [close] button.
(2)	Menu bar	The menu bar displays the menu commands for each purpose in list format.
(3)	Toolbar	The toolbar displays each command as an icon.
(4)	Navigator pane	The navigator pane displays the objects (such as devices, applications, and programs) added to the project in a tree structure.
(5)	Main pane	The main pane displays a program, function settings, messages, and other data. The window can be switched by selecting a desired tab.
(6)	Status field	The status bar displays the build status, logged-in users, and other information.

3.5.1 Menu Bar

The menu bar displays the following menus:



File

Item	Function
New Project	Creates a new project.
Open Project	Opens a project that is stored.
Close Project	Close the project that is currently viewed.
Save Project	Saves the project that is currently viewed, in overwrite mode.
Save Project As	Saves the project that is currently viewed, as a different file name.
Projetct Archive	Unzip the archive or save the archive.
Source Upload(Device→PC)	Loads the project source code as a project archive.
Print	Prints the active editor screen.
Print Preview	Displays the active editor screen in print preview mode.
Page Setup	Opens the Page Setup dialog box to configure a print layout.
Recent Projects	Displays the recently used projects.
Exit	Closes GM Programmer.

Edit

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Undo</td><td>Reverses the results of a previous editing action.</td></tr>
<tr><td>Redo</td><td>Allows the user to redo the last editing action after Undo.</td></tr>
<tr><td>Cut</td><td>Cuts data.</td></tr>
<tr><td>Copy</td><td>Copies data.</td></tr>
<tr><td>Paste</td><td>Pastes data.</td></tr>
<tr><td>Delete</td><td>Deletes data.</td></tr>
<tr><td>Select All</td><td>Selects all text.</td></tr>
<tr><td>Find Replace</td><td>Used to find and replace a string.
<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Find</td><td>Opens the "Find" dialog box.</td></tr>
<tr><td>Replace</td><td>Opens the "Replace" dialog box.</td></tr>
<tr><td>Find in Project</td><td>Opens the "Find" dialog box to find the target within the entire project.</td></tr>
<tr><td>Replace in Project</td><td>Opens the "Replace" dialog box to replace the target within the entire project.</td></tr>
<tr><td>Find Next</td><td>Finds the next match from the selected cursor position within the project.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td></td><td><table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Find Next (Selected)</td><td>Finds the next match from the selected cursor position within the editor.</td></tr>
<tr><td>Find Previous</td><td>Finds the previous match from the selected cursor position within the project.</td></tr>
<tr><td>Find Previous (Selected)</td><td>Finds the previous match from the selected cursor position within the editor.</td></tr>
<tr><td>Toggle Field for Incremental Search</td><td>Searches for the character string within the POU editor each time a single character is entered.</td></tr>
</table></td></tr>
<tr><td>Browse</td><td>Used to browse the positions where the declaration part of a defined variable is referenced or used.
<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Go To Definition</td><td>Allows the cursor to move to the position where the variable or function specified by the cursor is defined within the editor.</td></tr>
<tr><td>Browse Cross References</td><td>Allows the positions where the variable specified by the cursor is used to be displayed in the "Cross reference List" view.</td></tr>
<tr><td>Browse Call Tree</td><td>Allows the callee and caller of the variable specified by the cursor to be displayed in the "Call Tree" view.</td></tr>
<tr><td>Go To Reference</td><td>Displays the declaration position of the variable to which the pointer variable specified by the cursor refers.</td></tr>
<tr><td>Go To Instance</td><td>Displays the instance of the function block specified by the cursor in the new editor.</td></tr>
</table></td></tr>
<tr><td>Insert File as Text</td><td>Inserts the contents of the specified text file in the cursor position.</td></tr>
<tr><td>Advanced</td><td>Executes functions related to the text editor.
<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Overwrite Mode</td><td>Switches the text input mode from insert mode to overwrite mode.</td></tr>
<tr><td>View Whitespace</td><td>Displays the control characters of spaces and tabs.</td></tr>
<tr><td>View Indentation Guides</td><td>Inserts a broken line between indents when an indent is inserted in the program code.</td></tr>
<tr><td>Go To Line</td><td>Displays a line number dialog box and moves the cursor to the specified line.</td></tr>
<tr><td>Make Uppercase</td><td>Converts the selected character string in the text editor to uppercase letters.</td></tr>
<tr><td>Make Lowercase</td><td>Converts the selected character string in the text editor to lowercase letters.</td></tr>
<tr><td>Go to Matching Bracket</td><td>Moves the cursor to the corresponding bracket when the cursor is positioned in a bracket in the code.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Select to Matching Bracket</td><td>Selects the entire code in brackets when the cursor is positioned in either of the brackets in the code.</td></tr>
<tr><td>Expand All Folds</td><td>Unfolds the indented code segment.</td></tr>
<tr><td>Collapse All Folds</td><td>Folds the indented code segment.</td></tr>
<tr><td>Comment out selected lines</td><td>Comments out the selected line.</td></tr>
<tr><td>Uncomment selected lines</td><td>Uncomments the selected line.</td></tr>
<tr><td>Enable inline monitoring</td><td>Sets whether to enable or disable the function that displays the value of each variable on the code during online mode.</td></tr>
</table>
</td></tr>
<tr><td>Bookmarks</td><td>
Allows the cursor to move to bookmarked locations.
Used to browse the positions where the declaration part of a defined variable is referenced or used.
<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Toggle Bookmark</td><td>Saves the position selected in the active editor as a bookmark.</td></tr>
<tr><td>Next Bookmark (active editor)</td><td>Moves to the previous bookmark in the active editor.</td></tr>
<tr><td>Previous Bookmark (active editor)</td><td>Moves to the next bookmark in the active editor.</td></tr>
<tr><td>Clear All Bookmarks (active editor)</td><td>Removes all bookmarks in the active editor.</td></tr>
</table>
</td></tr>
<tr><td>Input Assistant</td><td>Allows the user to select variables, function blocks, operators, types, or other data that can be inserted in the cursor position from a category and insert them in the cursor position.</td></tr>
<tr><td>Function Block Guidance</td><td>Invokes the Function Block Guidance.</td></tr>
<tr><td>Auto Declare</td><td>Opens the Auto Declare dialog box to support variable declaration.</td></tr>
<tr><td>Next Message</td><td>Selects the next message in the message view.</td></tr>
<tr><td>Previous Message</td><td>Selects the previous message in the message view.</td></tr>
<tr><td>Go To Source Position</td><td>Moves to the position of the source code applicable to the message selected in the message view.</td></tr>
<tr><td>Refactoring</td><td>Displays the positions where the changed variable name is used and allows changes to be made collectively.</td></tr>
</table>

View

Item	Function
Devices	Displays the device view.
POUs	Displays the POU view.
Messages	Displays the message window.

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Element properties</td><td>Displays element properties.</td></tr>
<tr><td>ToolBox</td><td>Displays the toolbox.</td></tr>
<tr><td>Watch</td><td>Displays the watch window.<table><tr><th>Item</th><th>Function</th></tr><tr><td>Watch 1 to Watch 4</td><td>Displays a list of user-defined variables for the purpose of value monitoring.</td></tr><tr><td>Watch all Forces</td><td>Displays a list of value-forced variables.</td></tr></table></td></tr>
<tr><td>Cross Reference List</td><td>Displays the cross reference list window.</td></tr>
<tr><td>Call Tree</td><td>Displays the call tree window.</td></tr>
<tr><td>Bookmarks</td><td>Displays the bookmark window.</td></tr>
<tr><td>Breakpoints</td><td>Displays the breakpoint window.</td></tr>
<tr><td>Call Stack</td><td>Displays the call stack window.</td></tr>
<tr><td>Start Page</td><td>Displays the start page.</td></tr>
<tr><td>Security Screen</td><td>Displays the certificate creation and registration screen..</td></tr>
<tr><td>Full Screen</td><td>Displays the window in full-screen mode.</td></tr>
<tr><td>Properties</td><td>Displays the properties dialog box.</td></tr>
</table>

Project

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Add Object</td><td>Adds an object.</td></tr>
<tr><td>Add Folder</td><td>Adds a folder.</td></tr>
<tr><td>Edit Object</td><td>Allows the user to edit an object.</td></tr>
<tr><td>Online Config Mode</td><td>Removes the applications downloaded to the GM1 controller and allows connection to the GM1 controller.</td></tr>
<tr><td>Project Information</td><td>Allows the user to set project author information or check project file information.</td></tr>
<tr><td>Project Settings</td><td>Allows the user to configure project-related settings.</td></tr>
<tr><td>Project File Conversion</td><td>Allows the user to convert project files from the old project version to the new version.</td></tr>
<tr><td>Localization</td><td>Allows the user to translate and register comments, titles, and other information in the program to display the translated content in the program window.<table><tr><th>Item</th><th>Function</th></tr><tr><td>Create Localization Template</td><td>Creates and saves a localization template.</td></tr><tr><td>Manage Localizations</td><td>Imports a localization template that has been created.</td></tr><tr><td>Toggle Localization</td><td>Switches the language in the project.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Document</td><td>Allows the user to print the entire project.</td></tr>
<tr><td>Compare</td><td>Compares the displayed project with the stored project.</td></tr>
<tr><td>Commit accepted changes</td><td>Commits the difference between the objects compared by selecting <b>Project</b>&gt;<b>Compare</b> from the menu bar.</td></tr>
<tr><td>Export</td><td>Outputs an object from the displayed project as an XML file.</td></tr>
<tr><td>Import</td><td>Imports an object into the displayed project.</td></tr>
<tr><td>User Management</td><td>Allows execution permissions for operations (such as executing menu commands and adding, editing, and deleting objects) to be assigned to each group in which users are registered.
<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>User Logon</td><td>Logs in to the displayed project.</td></tr>
<tr><td>User Logoff</td><td>Logs off from the displayed project.</td></tr>
<tr><td>Permissions</td><td>Logs off from the displayed project.</td></tr>
</table></td></tr>
</table>

Build

Item	Function
Build	Verifies the syntax of objects.
Rebuild	Verifies the syntax of all objects again.
Generate code	Generates application codes.
Clean	Deletes application build information.
Clean all	Deletes all application build information in the same way as "Clean".

Online

Item	Function
Scan Network	Connect to the GM1 controller.
Add USB Port	Adds a USB port as a communication interface.
Login	Downloads the applications generated by code generation to the GM1 controller at the time of login.
Logout	Logs out from the device to which the user logged in.
Download	Downloads a program while the user is logged in.
Online Change	Allows the user to change applications without having to stop the GM1 controller during operation.
Status	Allows the user to check any errors that are currently occurring in the GM1 controller.
System Data History	Allows the user to check any errors that occurred in the GM1 controller.
Reset Warm	Initializes variables other than the RETAIN and PERSISTENT variables.
Reset Cold	Initializes variables other than the PERSISTENT variable.
Reset Origin	Initializes all variables. Removes active applications from the GM1 controller.

<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Simulation</td><td>Allows the user to perform a login operation without connecting to the GM1 controller and check behaviors in the same way as if the user logged in.</td></tr>
<tr><td>Security</td><td>Allows the user to configure user management, project encryption, and other settings.<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Logoff Current Device User</td><td>Logs off the users who are logged in to the device.</td></tr>
<tr><td>Add Device User</td><td>Adds users who can log in to the device.</td></tr>
<tr><td>Change Password Device User</td><td>Changes the passwords of users who are logged in to the device.</td></tr>
<tr><td>Remove Device User</td><td>Removes users who can log in to the device.</td></tr>
</table></td></tr>
<tr><td>SSL / TLS Certificate Management</td><td>Allows the user to register or delete the SSL / TLS certificate of the FTP server function.</td></tr>
<tr><td>Operation Mode</td><td>Allows the user to prevent some debug operations from being executed.<table>
<tr><th>Item</th><th>Function</th></tr>
<tr><td>Debug</td><td>Allows all debug operations to be executed.</td></tr>
<tr><td>Locked</td><td>Prohibits some operations such as adding new breakpoints or forcing variable values.</td></tr>
<tr><td>Operational</td><td>Prohibits any changes other than writing variables.</td></tr>
</table></td></tr>
<tr><td>Unit Composition　Confirmation</td><td>Allows the users to confirm the configuration of the main unit being connected</td></tr>
<tr><td>Unit Version Upgrade</td><td>Allows the users to update the firmware version of the unit.</td></tr>
</table>

Debug

Item	Function
Start	Starts the application.
Stop	Stops the application.
Single Cycle	Executes the application in every single cycle.
New Breakpoint	Creates a new breakpoint.
Edit Breakpoint	Allows the user to edit breakpoints.
Toggle Breakpoint	Allows the user to set or delete breakpoints.
Disable Breakpoint	Disables invalid breakpoints.
Enable Breakpoint	Enables valid breakpoints.
Step Over	Executes the program line by line. When a block (function or function block) is executed in the block invocation location, the cursor moves to the next line.

Item	Function
Step Into	Executes the program line by line. When a block (function or function block) is executed in the block invocation location, the cursor moves to the first line of the called block.
Step Out	When the program is executed within the called block, the execution continues until control returns to the calling block. When the program is executed outside the called block, the execution continues until control returns to the beginning of the program.
Run to Cursor	Executes the program up to the line specified by the cursor.
Set next Statement	Regards the line specified by the cursor as the next statement to be executed and skips processes over to that line.
Show next Statement	Jumps the cursor to the program line to be executed as the next step.
Write Values	Sets a value (to be changed later) only once. This value can then be changed by the program.
Force Values	Sets a value to be changed in every cycle and maintains the value.
Unforce Values	Cancels forced value change
Toggle Flow Control Mode	Performs monitoring by using different colors in positions where the program is executed and in positions where the program is not executed.
Display Mode	Allows the user to select binary, decimal, or hexadecimal as the display format of the variable value to be displayed.

Tools

Item	Function
PANATERM Lite for GM	Allows the user to select a device to which PANATERM Lite for GM is to connect.
Library Repository	Allows the user to install a created library in the library repository in order to use the functions or function blocks in the library.
Customize	Allows the user to change the layout of the menu bar and toolbar, and the shortcut assignment.
Options	Allows the user to set up each function of GM Programmer.

Window

Item	Function
Next Editor	Displays the next window.
Previous Editor	Displays the previous window.
Close All Editors	Closed all windows.
Reset Window Layout	Resets the layout of the window to its initial state.
New Horizontal Tab Group	Moves the selected window downward.
New Vertical Tab Group	Moves the selected window to the right.
Float	Sets the selected window in a floating state.
Dock	Sets the selected window in a docking state.
Auto Hide	Minimizes the window.

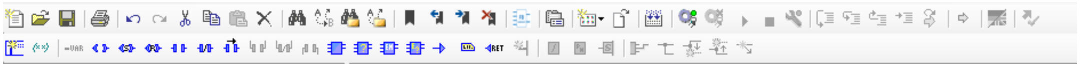
Item	Function
Next Pane	Switches the pane between the declaration section (first pane) and the implementation section (second pane).
Previous Pane	Switches the pane between the declaration section (first pane) and the implementation section (second pane).
Window	Displays a list of open windows.

Help

Item	Function
About	Displays version information.

3.5.2 Toolbar

The toolbar displays the following icons:



Name	Icon	Function
New Project	📄	Creates a new project.
Open Project	📂	Opens a project that is stored.
Save Project	💾	Saves the project that is currently viewed, in overwrite mode.
Print	🖶	Prints the active editor screen.
Undo	↶	Reverses the results of a previous editing action.
Redo	↷	Allows the user to redo the last editing action after Undo.
Cut	✂	Cuts data.
Copy	⧉	Copies data.
Paste	📋	Pastes data.
Delete	✕	Deletes data.
Find	🔍	Searches for a particular character string that appears in the active editor.
Find Replace	A→B	Searches for a particular character string that appears in the active editor and replaces it with another character string.
Find in Project	🔍	Searches for a specified character string within the current project.
Replace in Project	A→	Searches for a specified character string within the current project and replaces it with another character string.
Toggle Bookmark	🔖	Saves the position selected in the active editor as a bookmark.

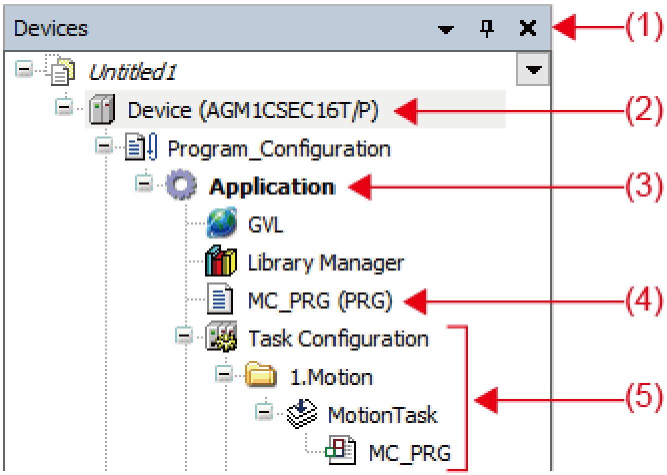
Name	Icon	Function
Previous Bookmark (active editor)		Moves to the previous bookmark in the active editor.
Next Bookmark (active editor)		Moves to the next bookmark in the active editor.
Clear All Bookmarks (active editor)		Removes all bookmarks in the active editor.
Function Block Guidance		Displays the Function Block Guidance.
Properties		Displays the properties.
Add Object		Adds an object.
Edit Object		Opens an object.
Build		Compiles an object in the application.
Login		Downloads the applications generated by code generation to the GM1 controller at the time of login.
Logout		Logs out from the device to which the user logged in.
Start	▶	Starts the application.
Stop	■	Stops the application.
Online Config Mode		Removes the applications downloaded to the GM1 controller and allows connection to the GM1 controller.
Step Over		Executes the program line by line. When a block (function or function block) is executed in the block invocation location, the cursor moves to the next line.
Step Into		Executes the program line by line. When a block (function or function block) is executed in the block invocation location, the cursor moves to the first line of the called block.
Step Out		When the program is executed within the called block, the execution continues until control returns to the calling block. When the program is executed outside the called block, the execution continues until control returns to the beginning of the program.

Name	Icon	Function
Run to Cursor	→≡	Executes the program up to the line specified by the cursor.
Set next Statement	⇨	Regards the line specified by the cursor as the next statement to be executed and skips processes over to that line.
Show next Statement	⇨	Jumps the cursor to the program line to be executed as the next step.
Toggle Localization		Switches the language to the one enabled in [Default Localization] in the window displayed by selecting [Project Localization] and then [Manage Localizations] from the [Project] menu.
Commit accepted changes	✓	Commits the difference between the objects compared by selecting the [Project] > [Compare] from the menu bar.
Insert Network		Inserts an empty network.
Toggle network comment state	(* *)	Changes the comment status of the selected network.
Insert Assignment	-VAR	Inserts a new assignment in the specified position.
Insert Coil	( )	Inserts a coil in the specified position.
Insert Set Coil	(S)	Inserts a set coil in the specified position.
Insert Reset Coil	(R)	Inserts a reset coil in the specified position.
Insert Contact	-| |-	Inserts a normally open contact in the specified position.
Insert Negated Contact	-|/|-	Inserts a normally closed contact in the specified position.
Insert Contact (right)	→ -| |-	Inserts a normally open contact on the right side of the specified position.
Insert Contact Parallel (below)		Inserts a normally open contact below and in parallel with the contact at the specified position.
Insert Negated Contact Parallel (below)		Inserts a normally closed contact below and in parallel with the contact at the specified position.
Insert Contact Parallel (above)		Inserts a normally open contact above and in parallel with the contact at the specified position.

Name	Icon	Function
Insert Box	▭	Opens the Input Assistant to insert a box in the specified position.
Insert Empty Box	?	Inserts an empty box in the specified position.
Insert Box with EN/ENO	EN	Opens the Input Assistant to insert a box with EN/ENO in the specified position.
Insert Empty Box with EN/ENO	?	Inserts a box with EN/ENO in the specified position.
Insert Jump	→	Inserts a jump in the specified position.
Insert label	LBL	Inserts a label in the selected network.
Insert Return	◀RET	Inserts a return value in the specified position.
Insert Input	⊣	Adds an input to the specified box.
Negation	/	Adds a negation to the selected element.
Edge Detection	P	Adds an edge detection (rising edge detection) to the selected element.
Set/Reset	S	Converts the selected coil to a set coil or reset coil.
Set output connection	⊢	Converts box output to forwarding box output.
Insert Branch	┬	Inserts a branch on the right side of the selected contact.
Insert Branch below	↓	Inserts a new branch below the selected branch.
Insert Branch above	↑	Inserts a new branch above the selected branch.
Set Branch Start/End Point	↘	Sets the selected line as the branch starting point.

3.5.3 Navigator Pane

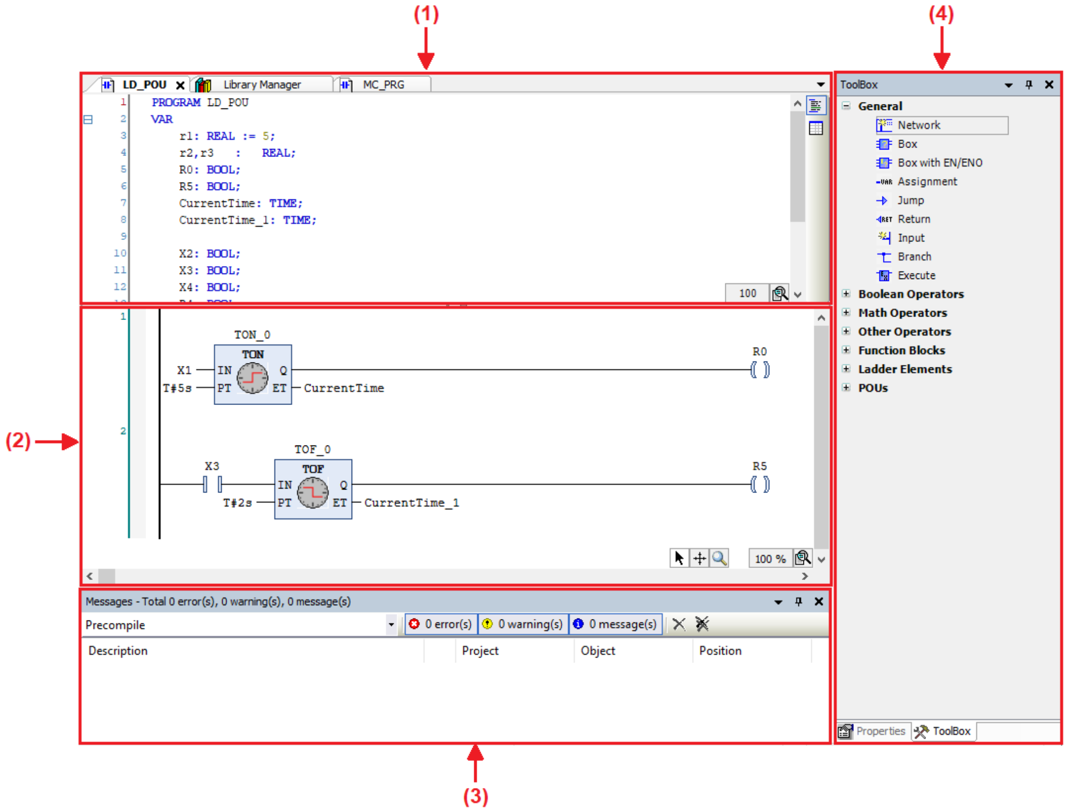
The navigator pane displays the following tree:



No.	Name	Icon	Function
(1)	Window Position	▼	● New Horizontal Tag Group Moves the selected window to the right. ● New Vertical Tag Group Moves the selected window downward. ● Float Sets the selected window in a floating state. ● Dock Sets the selected window in a docking state. ● Auto Hide Minimizes and hides the navigator pane.
	Auto Hide	📌	Always shows the navigator pane.
		⊣	Minimizes and hides the navigator pane.
	Close	✕	Closes the navigator pane.
(2)	Device object		Sets up device objects.
(3)	Application object		Sets up application objects.
(4)	Program object (POU object)		Sets up program objects (POU objects).
(5)	Task object		Sets up task objects.

3.5.4 Main Pane

The main pane displays the following sub-panes:



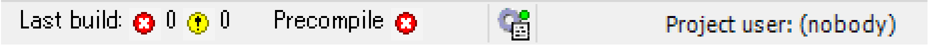
```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;
    R0: BOOL;
    R5: BOOL;
    CurrentTime: TIME;
    CurrentTime_1: TIME;

    X2: BOOL;
    X3: BOOL;
    X4: BOOL;
```

Messages - Total 0 error(s), 0 warning(s), 0 message(s)

No.	Name	Function
(1)	Declaration section (first pane)	Allows the user to declare variables.
(2)	Implement section (second pane)	Allows the user to enter a program.
(3)	Message view	Displays any error or warning messages.
(4)	ToolBox	Allows the user to place elements in the implementation section by selecting them and then dragging and dropping them in the implementation section.

3.5.5 Status Bar

The status bar displays the following icons:



Last build: ⊗ 0 ⚠ 0 Precompile ⊗ ⚙ Project user: (nobody)

Name	Icon	Function
Last Build	⊗ 0	Displays the number of errors in the results of the build process.
	⚠ 0	Displays the number of warnings in the results of the build process.
Precompile	—	Displays the results of the precompile process.
Application Information	⚙	Compares the application information of the displayed project with the application information downloaded to the GM1 controller.
Project user	—	Displays the users who are logged in to the displayed project.

3.6 Window Operations

This section explains operations related to common windows for GM Programmer.

3.6.1 Moving the Pane Location

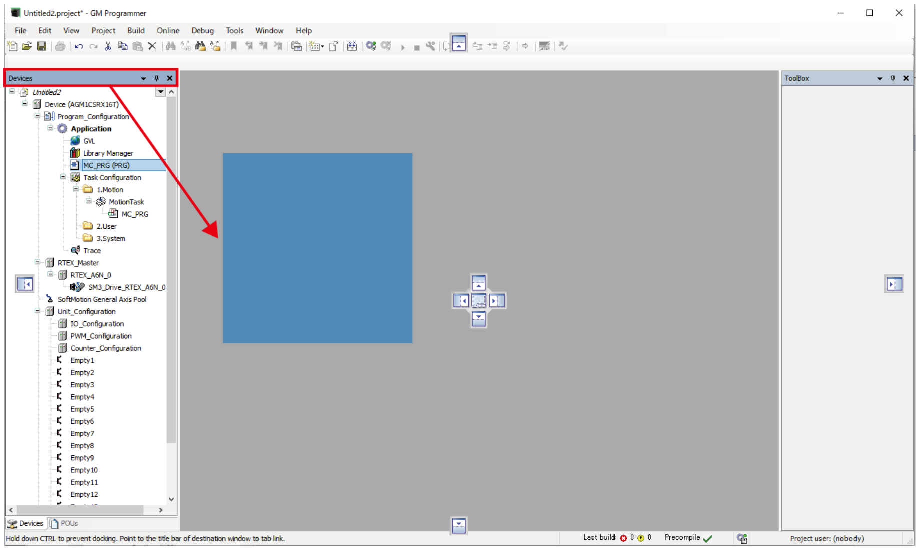
You can freely change the layout of each window for GM Programmer.

For example, use the following procedure to move the navigator pane from the left edge to the right edge of the window.

1 2 Procedure

1. Click the title bar of the navigator pane and then drag it to the main pane.

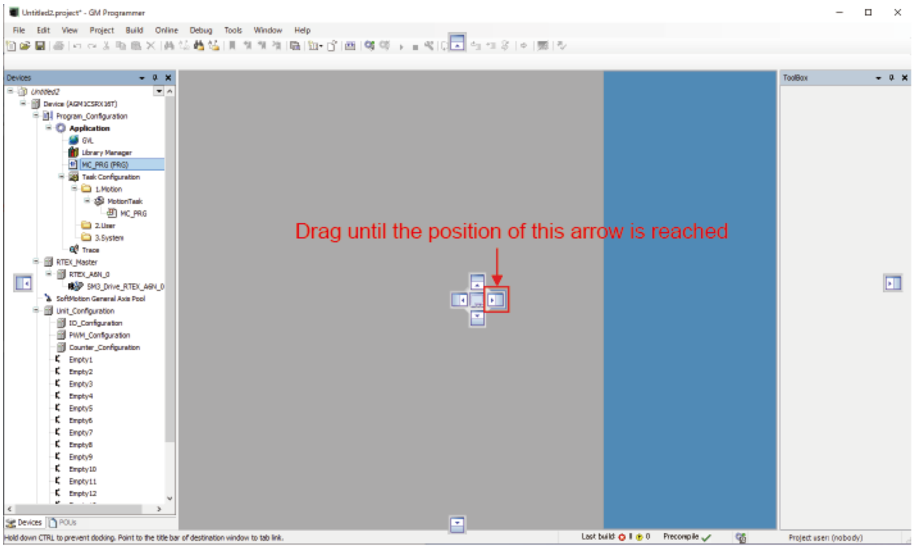
   The navigator pane will stay in a floating state and arrows indicating movable directions will be displayed.



Hold down CTRL to prevent docking. Point to the title bar of destination window to tab link.

2. Drag the navigator pane in the direction in which you want to move it.

   The relocation destination will be displayed in light blue.

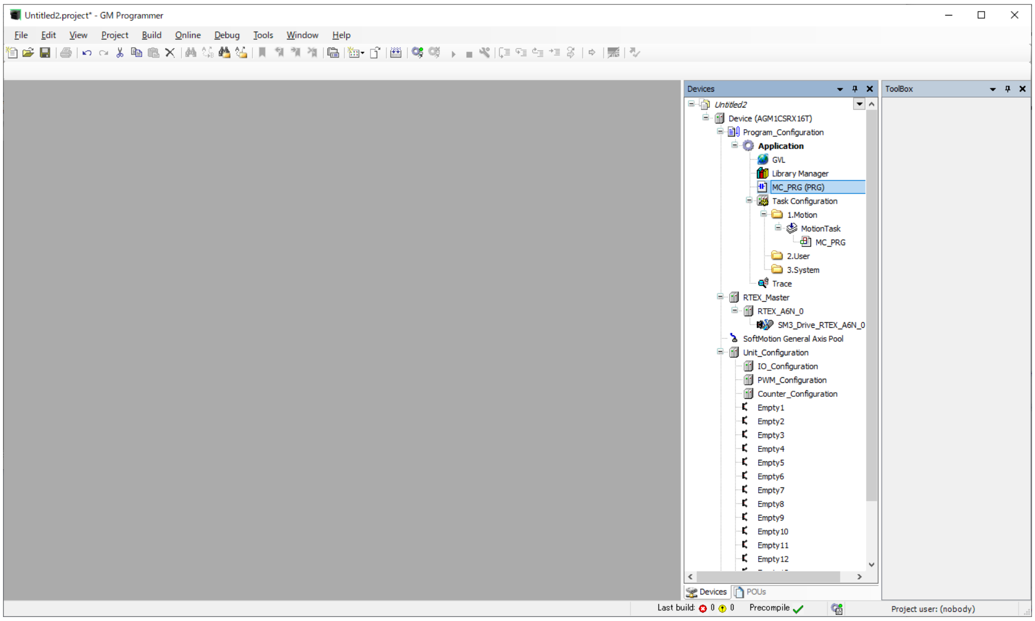


Drag until the position of this arrow is reached

Hold down CTRL to prevent docking. Point to the title bar of destination window to tab link.

3. Release the left mouse button.

The navigator pane will be docked into the existing pane and the relocation will be completed.



Info.

- You can return the changed layout of the window to its initial state.

  From the menu bar, select **Window**>**Reset Window Layout**.

- You can put a pane in a floating or docking state. To put a pane in a floating state, select **Window**>**Float** from the menu bar. To put a pane in a docking state, right-click the title bar in the window in a floating state and then select Dock from the context-sensitive menu that is displayed.

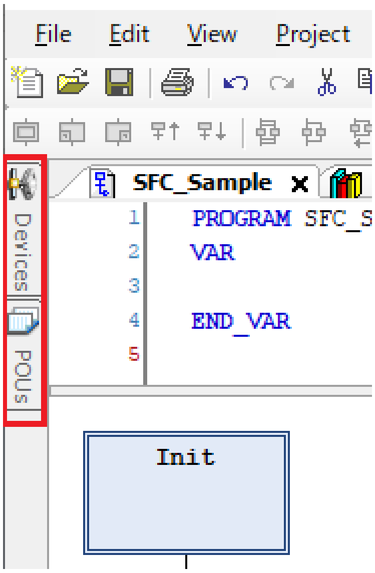
3.6.2 Showing / Hiding Panes

You can normally hide the navigator pane and some sub-panes in the main pane and show them only when you use them.

■ Showing / hiding the navigator pane

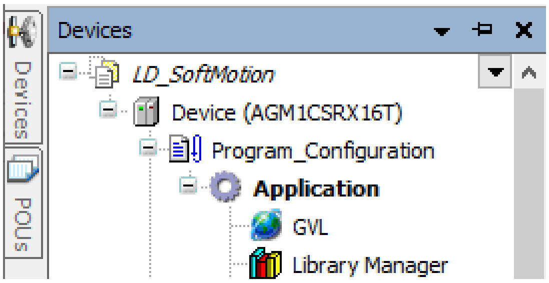
1 2 Procedure

1. Click on the title bar of the navigator pane.
The navigator pane will be minimized and hidden.



```
PROGRAM SFC_S
VAR

END_VAR

```

2. Click the minimized pane.
The navigator pane will be displayed. Clicking in another pane automatically hides the navigator pane again.



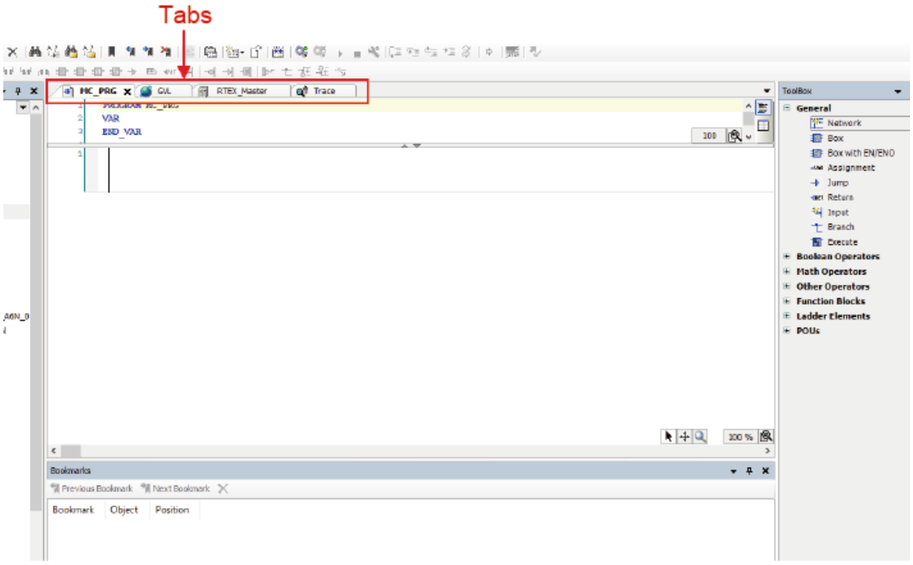
3. Click on the title bar of the navigator pane.
The navigator pane will always be displayed.

Info.

- You can also hide the navigator pane from the menu bar. From the menu bar, select **Window**>**Auto Hide**. To always display the navigator pane again, select **Window**>**Auto Hide** again from the menu bar.

3.6.3 Switching the Tab of the Main Pane

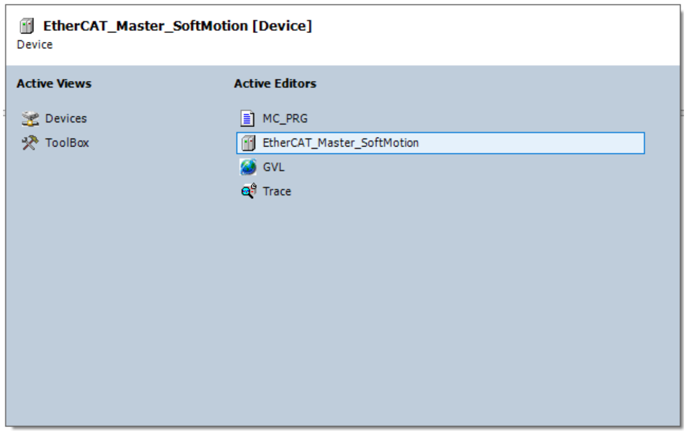
You can switch the tab of the main pane.



```
VAR
END_VAR
```

Procedure

1. Press the shortcut keys "Ctrl+Tab" simultaneously.
   The window for switching the tab of the main pane will be displayed.



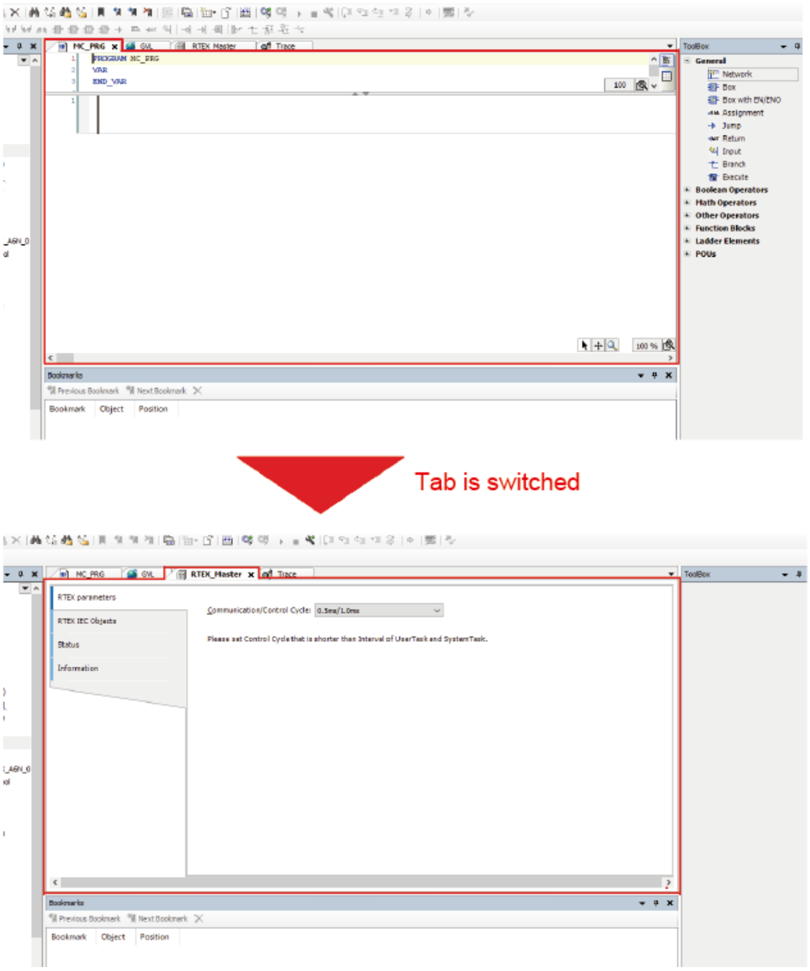
EtherCAT_Master_SoftMotion [Device]
Device

Active Views
- Devices
- ToolBox

Active Editors
- MC_PRG
- EtherCAT_Master_SoftMotion
- GVL
- Trace

2. While holding down the "Ctrl" key, press the "Tab" key until the desired tab is selected.

3. Release the "Ctrl" key.
   The current tab will be switched to the selected tab.



```
PROGRAM MC_PRG
VAR
END_VAR
```

Communication/Control Cycle: 0.5ms/1.0ms

Please set Control Cycle that is shorter than Interval of UserTask and SystemTask.

3.6.4 Full-screen Display

You can display each window of GM Programmer in full-screen mode.

1 2 Procedure

1. From the menu bar, select **View**>**Full Screen**.
   Then GM Programmer window will be displayed in full-screen mode.

2. From the menu bar, select **View**>**Full Screen** again.
   Then GM Programmer window will return from full-screen mode to normal display mode.

Info.

- You can also switch to full-screen mode by pressing shortcut keys "Ctrl+Shift+F12" simultaneously.

3.7 Switching the Object Window

Double-clicking an object added to the navigator pane displays its window in the main pane.

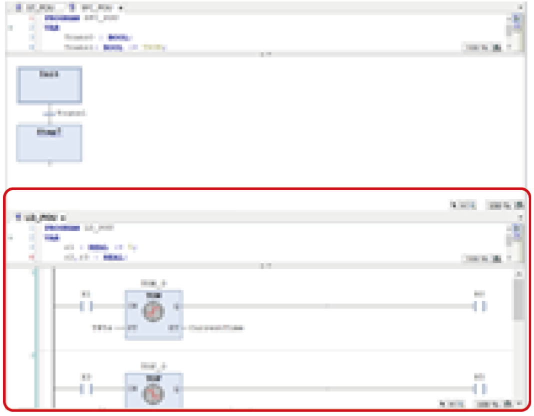
You can open multiple objects in the main pane and switch to each of their windows using the tab.

3.7.1 Operating the Object Window

The following operations can be performed on the object window displayed in the main pane.

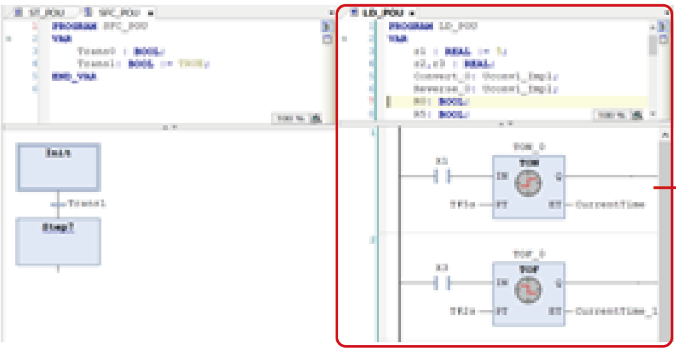
Operation	Menu	Shortcut keys
Displaying the next window	**Window>Next Editor**	<Ctrl> + <F6>
Displaying the previous window	**Window>Previous Editor**	<Ctrl> + <Shift> + <F6>
Closing all windows	**Window>Close All Editors**	None
Moving the selected window downward	**Window>New Horizontal Tab Group**	None
Moving the selected window to the right	**Window>New Vertical Tab Group**	None

<Moving the selected window downward (New Vertical Tab Group)>



Move the selected pane downward until both panes are arranged vertically

<Moving the selected window to the right (New Horizontal Tab Group)>



Move the selected pane to the right until both panes are arranged horizontally

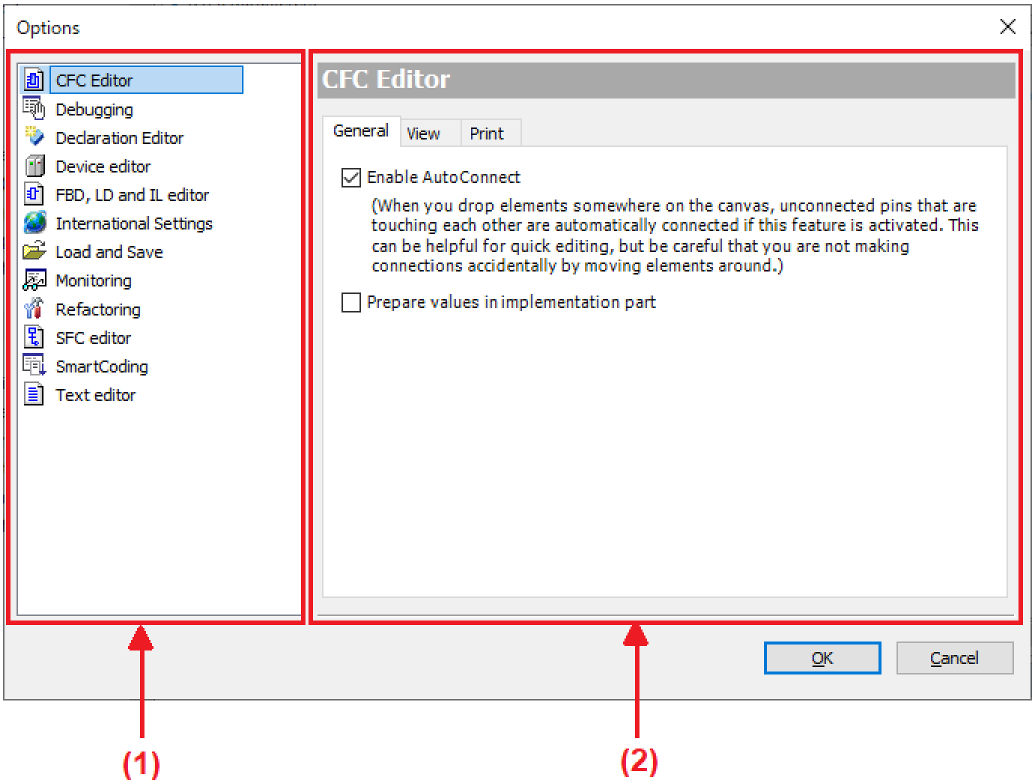
3.8 Other Functions

3.8.1 Option Setting Function

GM Programmer allows the user to set up each function from the "Options" dialog box. The settings will be applied to all projects created with GM Programmer.

Procedure

1. From the menu bar, select **Tools**>**Options**.
   The "Options" dialog box will be displayed.



☑ Enable AutoConnect
(When you drop elements somewhere on the canvas, unconnected pins that are touching each other are automatically connected if this feature is activated. This can be helpful for quick editing, but be careful that you are not making connections accidentally by moving elements around.)

□ Prepare values in implementation part

No.	Name	Function
(1)	Categories pane	Displays option categories.
(2)	Setting pane	Displays the settings of the selected category and allows the user to configure settings.

Option categories

Category name	Function	Reference page
CFC Editor	Allows the user to configure settings related to editing and printing CFC programs.	-

Category name	Function	Reference page
FBD, LD and IL editor	Allows the user to configure settings related to editing, commenting, and printing FBD, LD, and IL programs.	"P.7-12" "P.7-34"
SFC editor	Allows the user to configure settings related to the sizes and fonts of SFC editor elements, the behavior at the time of action element insertion, the display of embedded objects in the navigator pane, the display of properties, stepwise execution time during online operation, and other items.	"P.7-24"
SmartCoding	Allows the user to configure settings related to the functions for supporting program creation, such as Input Assistant.	"P.7-15" "P.6-7" "P.6-48"
Text editor	Allows the user to configure settings related to program editing and inline monitoring.	"P.9-16"
Device editor	Allows the user to configure settings related to displays for the device editor.	-
Debugging	Allows the user to configure settings regarding whether to restore breakpoints after resetting.	-
Monitoring	Allows the user to configure settings related to displays for monitoring.	-
Refactoring	Allows the user to configure settings for the valid range of refactoring.	"P.6-49"
Load and save	Allows the user to configure settings regarding whether to enable backup and auto saving of project files.	"P.4-23" "P.4-25"
International Settings	Allows the user to set a display language for GM Programmer and PANATERM Lite for GM, as well as a display language for the manual.	"P.3-39"
Declaration Editor	Allows the user to configure settings related to the display format (text format or table format) for the declaration section.	"P.6-4"

2. Select a desired category from the Categories pane.
   The setting items for the selected category will be displayed in the setting pane.

3. Change the setting items as appropriate and click the [OK] button.
   The setting items will be applied.

3.8.2 Display Language Setting Function

This function allows the user to change the display language setting for GM Programmer.

The default setting is the same language as the one used in the operating system. If you want to use a different language from the one used in the operating system, change the display language setting. After you change the language setting, you must restart GM Programmer.

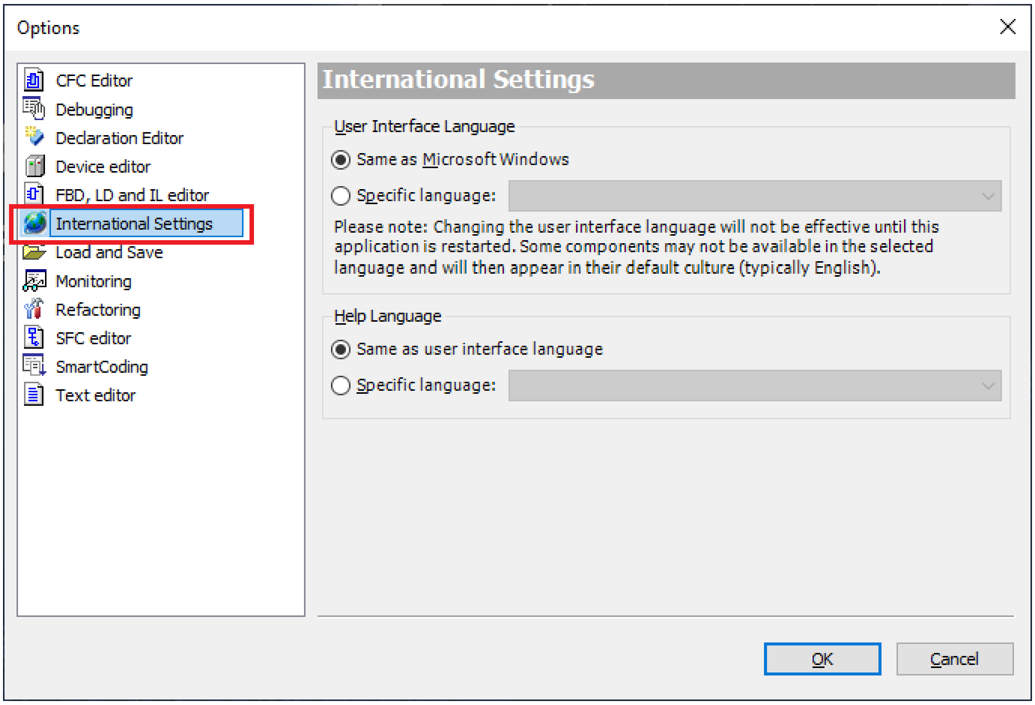
1 2 Procedure

1. From the menu bar, select **Tools**>**Options**.

   The "Options" dialog box will be displayed.

2. Select "International Settings" from the Categories pane.

   The "International Settings" pane will be displayed.



Options

- CFC Editor
- Debugging
- Declaration Editor
- Device editor
- FBD, LD and IL editor
- International Settings
- Load and Save
- Monitoring
- Refactoring
- SFC editor
- SmartCoding
- Text editor

International Settings

User Interface Language

● Same as Microsoft Windows

○ Specific language:

Please note: Changing the user interface language will not be effective until this application is restarted. Some components may not be available in the selected language and will then appear in their default culture (typically English).

Help Language

● Same as user interface language

○ Specific language:

3. Select **User Interface Language**>**Specific language** option and specify a desired language in the field.

4. Click [OK].

   The "Options" dialog box will be closed.

   At this stage, the language has not been changed yet.

5. Close GM Programmer and then start GM Programmer again.

   After GM Programmer is started, the selected language takes effect.

Info.

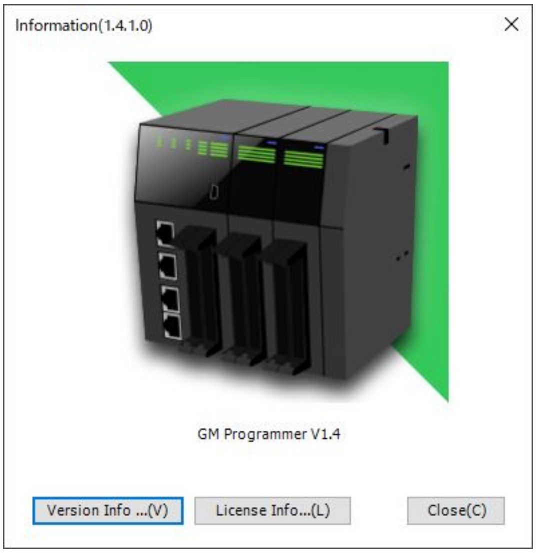
- The display language setting of GM Programmer is linked with that of PANATERM Lite for GM. Therefore, if the display language setting of PANATERM Lite for GM is changed, the display language setting of GM Programmer will also be changed automatically.

3.8.3 Version Display Function

This function allows the user to check the version, license, and other information for GM Programmer.

1 2 Procedure

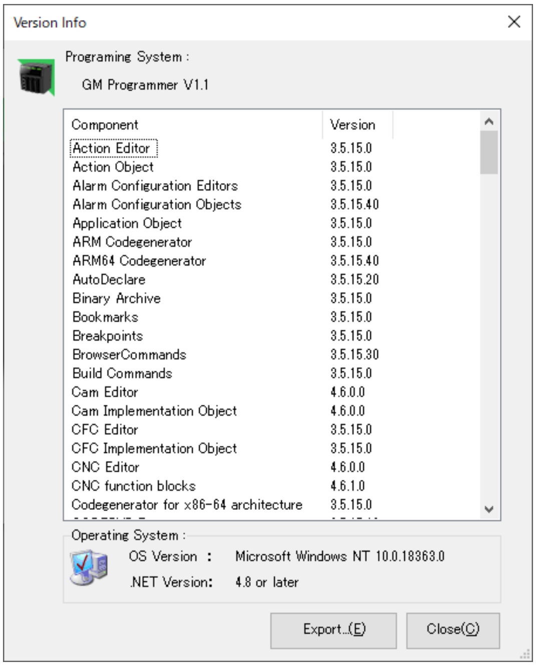
1. From the menu bar, select **Help**>**Version Info**.
   The GM Programmer version is displayed in parentheses on the title bar.



2. Click a desired button at the bottom of the window.

Button	Description
Version Info	Displays information about the plug-ins that have been applied and the operating system of the PC that is used.
License Info	Displays license information for the software used by GM Programmer.

Clicking the[Version Info]button displays the"Version Info"dialog box.



Programing System :

GM Programmer V1.1

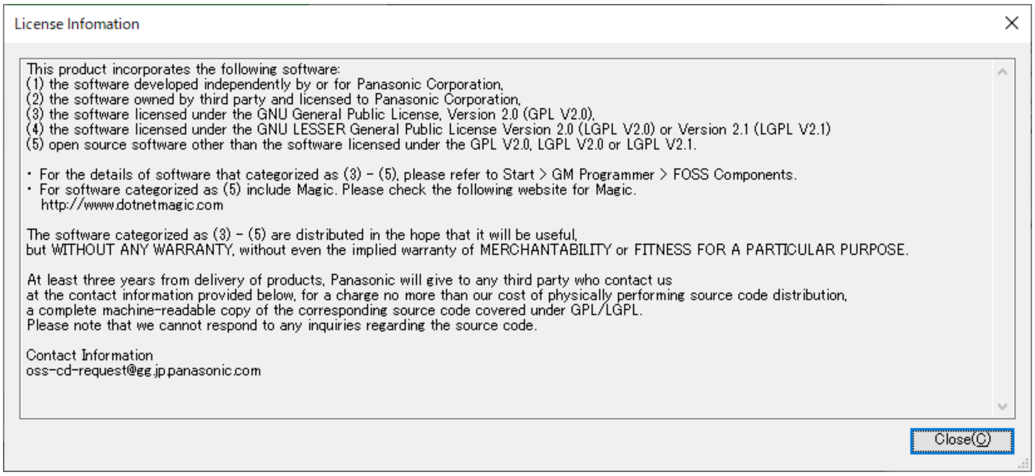
Component	Version
Action Editor	3.5.15.0
Action Object	3.5.15.0
Alarm Configuration Editors	3.5.15.0
Alarm Configuration Objects	3.5.15.40
Application Object	3.5.15.0
ARM Codegenerator	3.5.15.0
ARM64 Codegenerator	3.5.15.40
AutoDeclare	3.5.15.20
Binary Archive	3.5.15.0
Bookmarks	3.5.15.0
Breakpoints	3.5.15.0
BrowserCommands	3.5.15.30
Build Commands	3.5.15.0
Cam Editor	4.6.0.0
Cam Implementation Object	4.6.0.0
CFC Editor	3.5.15.0
CFC Implementation Object	3.5.15.0
CNC Editor	4.6.0.0
CNC function blocks	4.6.1.0
Codegenerator for x86-64 architecture	3.5.15.0

Operating System :

OS Version : Microsoft Windows NT 10.0.18363.0

.NET Version: 4.8 or later

Clicking the[License Info]button displays the"License Information"dialog box.



This product incorporates the following software:
(1) the software developed independently by or for Panasonic Corporation,
(2) the software owned by third party and licensed to Panasonic Corporation,
(3) the software licensed under the GNU General Public License, Version 2.0 (GPL V2.0),
(4) the software licensed under the GNU LESSER General Public License Version 2.0 (LGPL V2.0) or Version 2.1 (LGPL V2.1)
(5) open source software other than the software licensed under the GPL V2.0, LGPL V2.0 or LGPL V2.1.

- For the details of software that categorized as (3) - (5), please refer to Start > GM Programmer > FOSS Components.
- For software categorized as (5) include Magic. Please check the following website for Magic.
  http://www.dotnetmagic.com

The software categorized as (3) - (5) are distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

At least three years from delivery of products, Panasonic will give to any third party who contact us
at the contact information provided below, for a charge no more than our cost of physically performing source code distribution,
a complete machine-readable copy of the corresponding source code covered under GPL/LGPL.
Please note that we cannot respond to any inquiries regarding the source code.

Contact Information
oss-cd-request@gg.jp.panasonic.com

3.8.4 Customize function

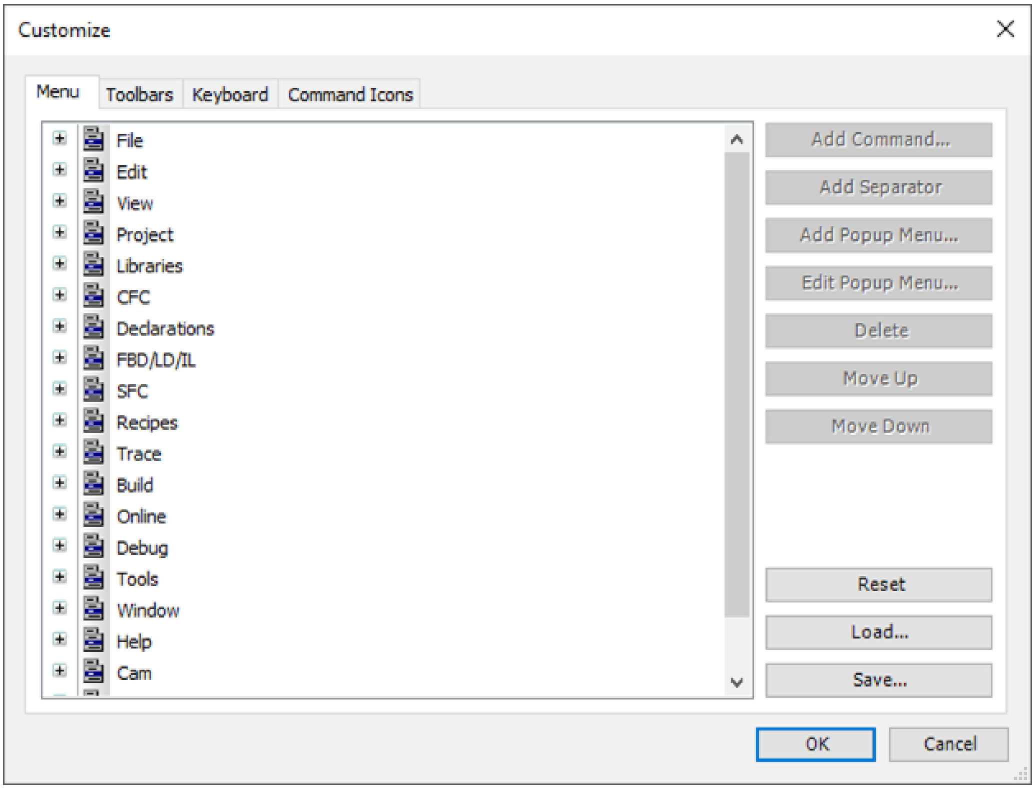
You can use custom functions to change the configuration of the menu bar and toolbar.
You can also change the assignment of shortcut keys and menu command icons.

Changing the menu bar

You can add menu commands to the category.

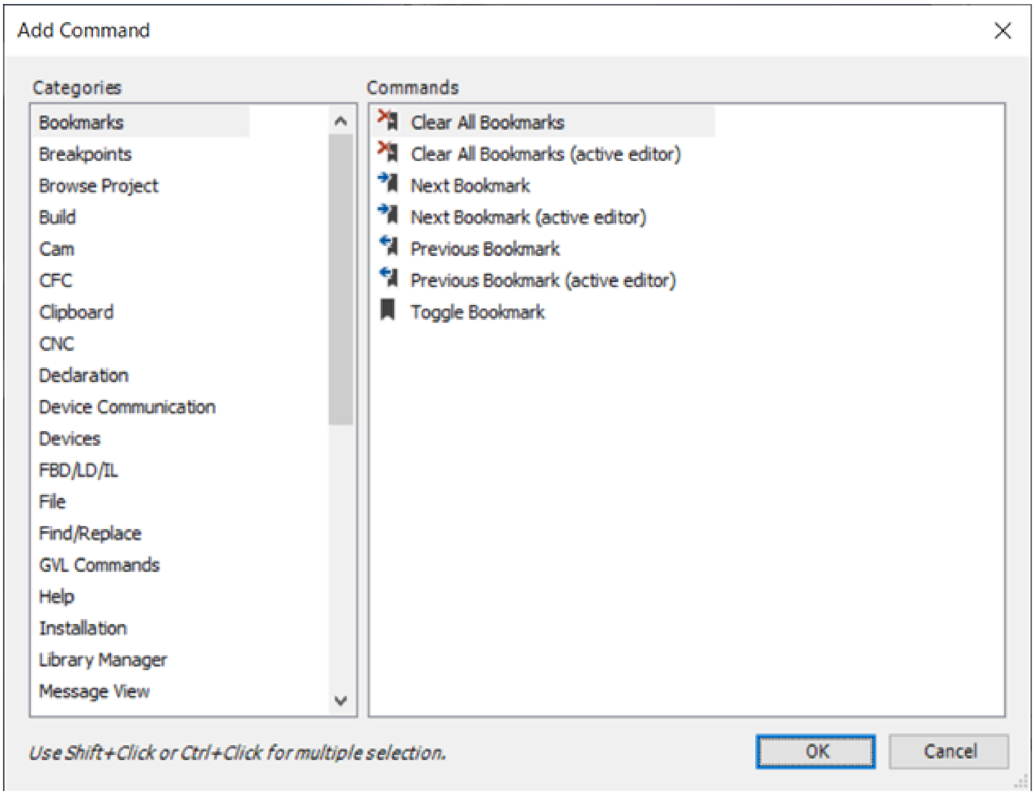
Procedure

1. Select [Tools]➔ [Customize] in the menu bar.
   The Customize dialog box appears.



2. Select the category you want to add a new command to and expand the menu.
3. Select the menu below where you want to add the menu.
   Select the whitespace symbol (□) if you want to add it to the bottom of the category.
4. Click [Add Command].
   The [Add Command] dialog box appears.

A list of commands that can be added by category appears.



Use Shift+Click or Ctrl+Click for multiple selection.

5. Select the command you want to add and click the [OK] button.
The command is added to the menu of the [Customize] dialog box.
You can change the order of the menus by clicking the [Move Up] and [Move Down] buttons.
You can add a border between the menus by clicking the[Add Separator] button.

6. Click the [OK] button.
The [Customize] dialog box closes and commands are added to the menu bar.

Info.

- When the version of GM Programmer is updated, it will return to the default settings, so save the necessary settings before performing.
- You can delete the menu command by clicking the [Delete] button.
- When the [Reset] button is clicked, the menu configuration returns to the default state.
- You can save the menu configuration by clicking the [Save] button. The file extension of the saved menu configuration is “.opt.menu”.
- You can load the saved menu configuration file by clicking the[Load] button.

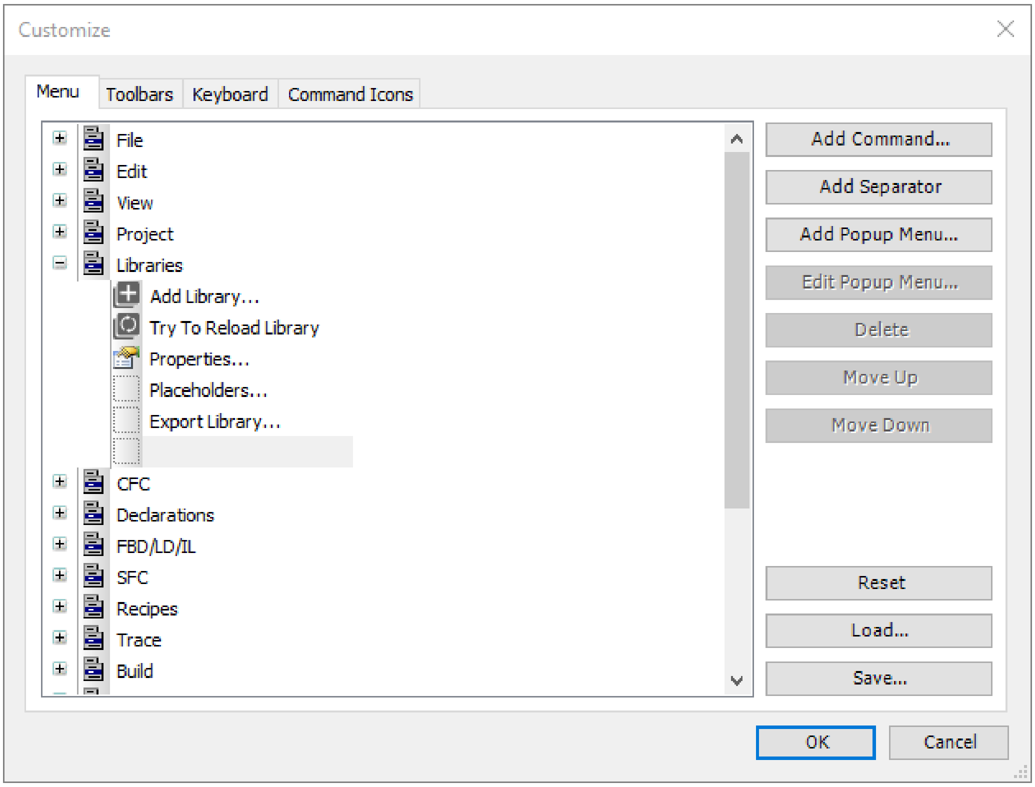
Adding a Popup Menu

You can add a popup menu to the menu bar. A menu command can be added to the popup menu.

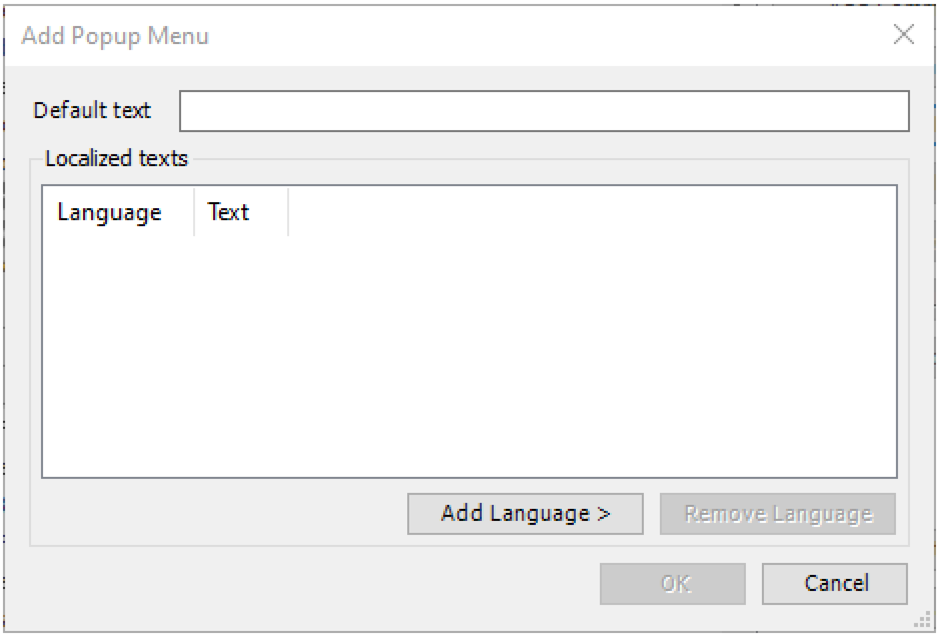
1 2 Procedure

1. Select [Tools] -> [Customize] in the menu bar.
The [Customize] dialog box appears.

2. Select the whitespace symbol (⬚) at the bottom of the hierarchy where you want to add a menu to the menu tree.



3. Click the [Add Popup Menu] button.
The [Add Popup Menu] dialog box appears.



4. Enter a name for the new popup menu in the default text field.

   You can use the [Add Language] button to add the name of the language you want to change. After selecting the language, click in the text field and enter text in the selected language.

5. Click the [OK] button.

   A popup menu is added to the menu of the Customize dialog box.

   You can change the order of the popup menus by clicking the [Move Up] and [Move Down] buttons.

6. Click the [OK] button.

   The [Customize] dialog box closes and a popup menu is added to the menu bar.

i Info.

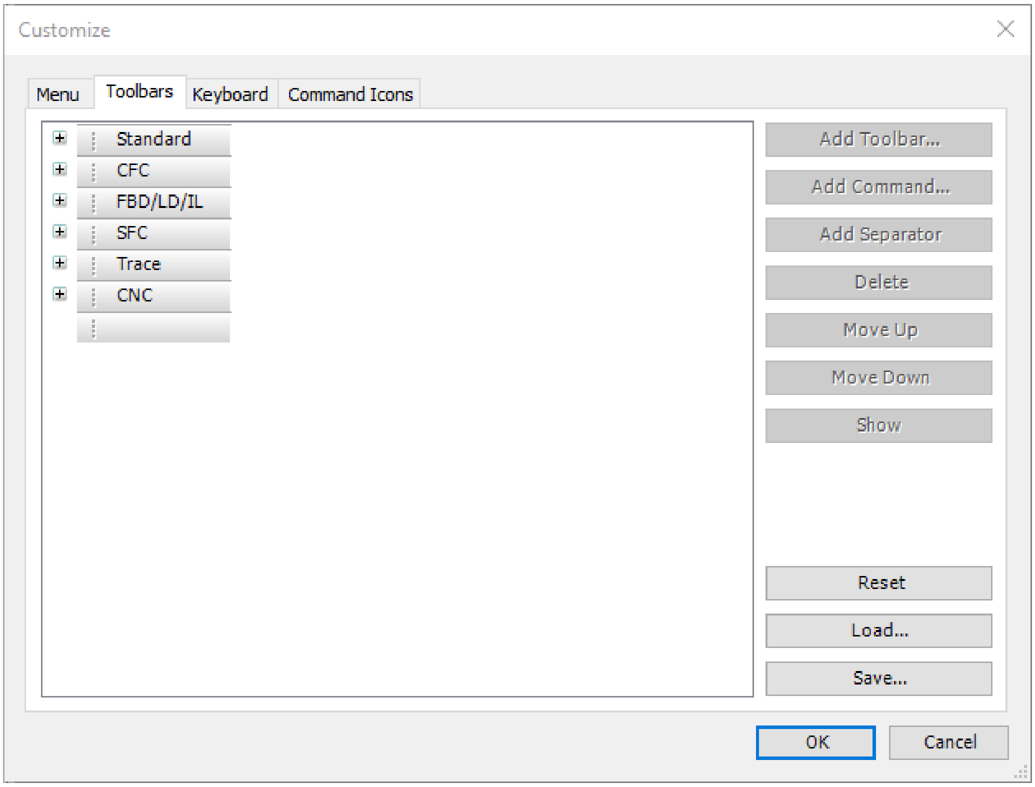
- When upgrading the version of GM Programmer, it will return to the default settings. Therefore, when upgrading the version of GM Programmer, please save the necessary settings in advance.
- You can edit the display of the added popup menu by clicking the "Edit Popup Menu" button.
- You can delete the added popup menu by clicking the [Delete] button.
- When the [Reset] button is clicked, the menu configuration returns to the default state.
- You can save the menu configuration by clicking the [Save] button. The extension of the saved file is ***.opt.menu.
- You can load the saved menu configuration file by clicking the [Load] button.

Adding a toolbar

1 2 Procedure

1. Select [Tools] -> [Customize] in the menu bar.
   The [Customize] dialog box appears.

2. Select the [Toolbar] tab.
   The list of toolbars appears.



3. Select the toolbar and click the [Add Toolbar] button.
   A new toolbar is added above the selected toolbar.

4. Enter a name for the toolbar.
   The toolbar is added to the [Customize] dialog box.
   You can change the order of the toolbars by clicking the [Move Up] and [Move Down] buttons.
   This completes the addition of the toolbar.

5. This section explains how to add commands to the toolbar you've added. Expand the toolbar tree where you want to add new commands.

6. Select the whitespace symbol (⬚).

7. Click [Add Command].
   The [Add Command] dialog box appears.
   A list of the commands you can add by category appears.

8. Select the command you want to add and click the [OK] button.
   The command is added to the toolbar.
   You can change the order of the commands by clicking the [Move Up] and [Move Down] buttons.
   You can add a border between the commands by clicking the [Add Separator]button.

9. Click the [OK] button.
   The Customize dialog box closes and the toolbar and commands are added.

i Info.

- When upgrading the version of GM Programmer, it will return to the default settings. Therefore, when upgrading the version of GM Programmer, please save the necessary settings in advance.
- You can delete added toolbars or commands by clicking the "Delete" button.
- Select the toolbar and click the [Hide] button to hide the toolbar. Click the [Show] button to make the toolbar visible again.
- Click the [Reset] button to reset the toolbar to the default state.
- You can save the toolbar by clicking the [Save] button. The extension of the saved file is ***.opt.tbar.
- You can load a saved toolbar by clicking the [Load] button.

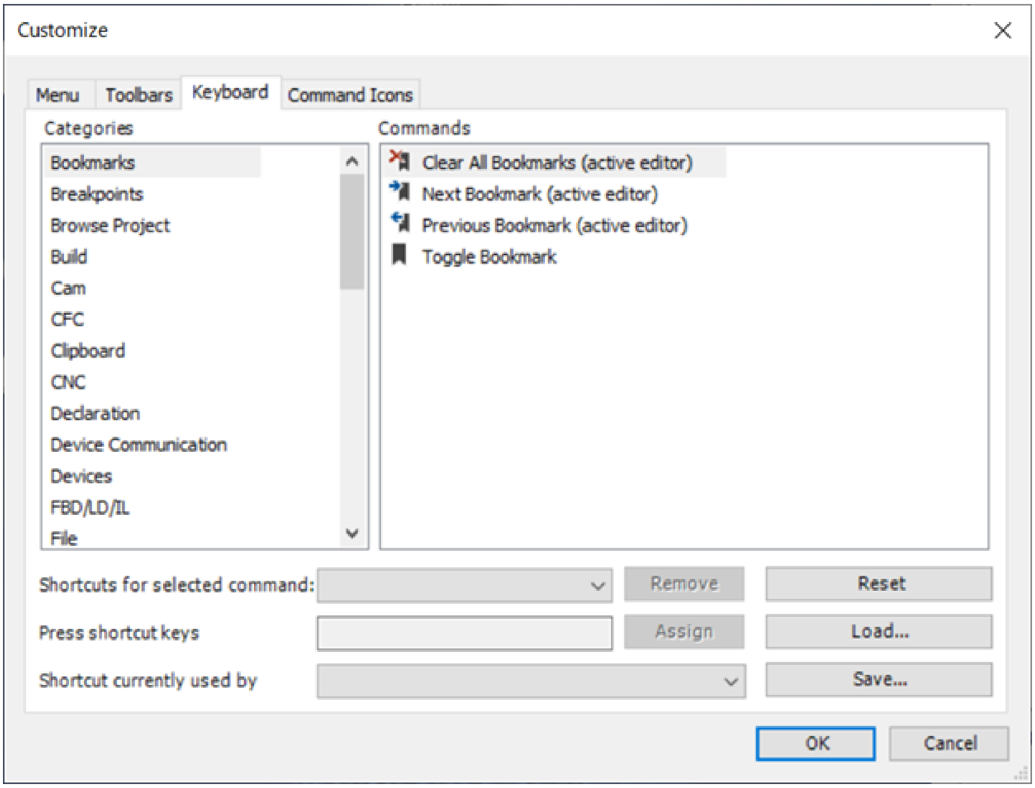
Changing command shortcut keys

You can change the shortcut keys assigned to commands.

1 2 Procedure

1. Select [Tools] -> [Customize] in the menu bar.
   The [Customize] dialog box appears.

2. Select the [Keyboard] tab.
   A list of commands by category appears.



Customize

Menu Toolbars Keyboard Command Icons

Categories: Bookmarks, Breakpoints, Browse Project, Build, Cam, CFC, Clipboard, CNC, Declaration, Device Communication, Devices, FBD/LD/IL, File

Commands: Clear All Bookmarks (active editor), Next Bookmark (active editor), Previous Bookmark (active editor), Toggle Bookmark

3. Select a category from the list in the Categories column.
   The list of commands is displayed in the Commands field.

4. Select the command for which you want to set the shortcut key.
   If a shortcut key is assigned to the selected command, it is displayed in the Shortcut field of the selected command.

5. In the shortcut key field, press the shortcut key you want to assign.
   If the key you pressed appears in the field to press the shortcut key, it can be assigned.
   - <Ctrl>key + alphanumeric key or <F2>key to <F12>key
   - <Alt>key + alphanumeric key or <F2>key to <F12>key

- <Ctrl>key + <Shift>key + alphanumeric key or <F2>key to <F12>key
- <Shift>key + <F2>key to <F12>key
- <Del>key
- <Ins>key

6. Click the [Assign] button.
   The shortcut that is assigned to the shortcut field for the selected command is shown.

7. Click the [OK] button.
   The [Customize] dialog box closes and shortcut key is assigned to the command.

Info.

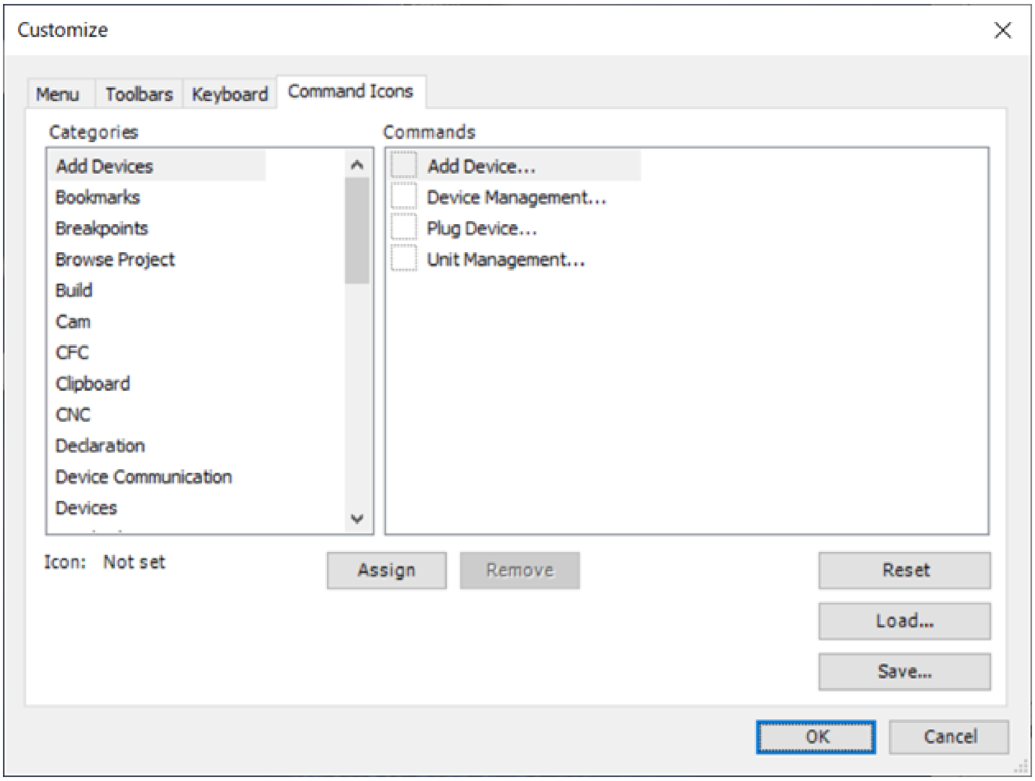
- When upgrading the version of GM Programmer, it will return to the default settings. Therefore, when upgrading the version of GM Programmer, please save the necessary settings in advance.
- If you want to delete the shortcut key assigned to the command, select the command and click the "Delete" button.Click the "OK" button to confirm the command shortcut key deletion.
- Select the toolbar and click the [Hide] button to hide the toolbar. Click the [Show] button to make the toolbar visible again.
- Click the [Reset] button to return the shortcut key assignment to the default state.
- You can save the shortcut key by clicking the [Save] button. The saved file has the extension ***.opt.keyb.
- You can load the saved shortcut key file by clicking the [Load] button.

Changing command icons

You can change the icon assigned to a command.

Procedure

1. Select [Tools] -> [Customize] in the menu bar.
   The [Customize] dialog box appears.
2. Select the [Command Icon] tab.
   A list of commands by category appears.



Icon: Not set

3. Select a category from the list in the Categories column.
   The list of commands is displayed in the Commands field.
4. Select the command for which you want to set the icon.
5. Click the [Assign] button on the small icon and select an icon file (***.ico).
   The selected icons are displayed.
6. Click the [OK] button.
   The Customize dialog box closes and an icon is assigned to the command.

Info.

- When upgrading the version of GM Programmer, it will return to the default settings. Therefore, when upgrading the version of GM Programmer, please save the necessary settings in advance.
- You can delete the icon assignment by clicking the "Delete" button.
- Click the [Reset] button to return the icon assignment to the default state
- You can save the icon assignment state by clicking the [Save] button.
- You can load the saved file by clicking the [Load] button.

4 Project Operations

4.1 Creating a New Project

When creating a program using GM Programmer for the first time, create a new project. For the new project, set a device and a programming language to be used.

This section describes how to create a new project.

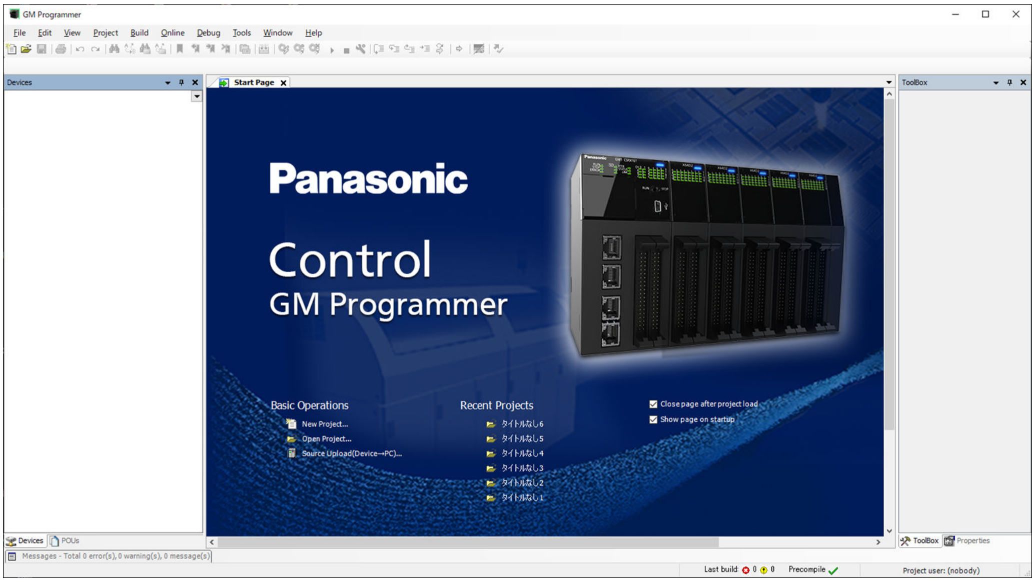
Given below is an example that explains the procedure to create a project for the GM1 Controller (product number: AGM1CSEC16T/P) in Structured Text (ST) format.

1 2 Procedure

1. Start GM Programmer.

   For details on how to start GM Programmer, refer to "3.4.1 How to start".

   When GM Programmer is started, the Start Page will be displayed.



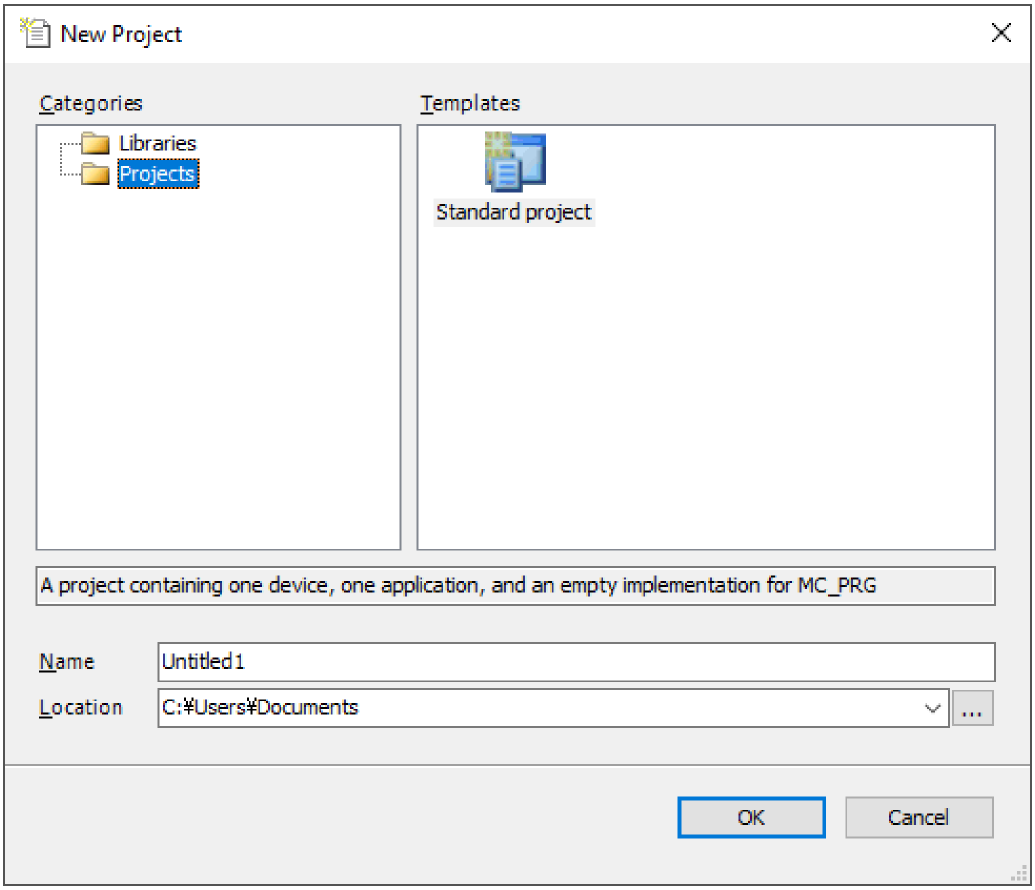
☑ Close page after project load
☑ Show page on startup

Messages - Total 0 error(s), 0 warning(s), 0 message(s)

Last build: 0 0 Precompile ✓ Project user: (nobody)

2. Select "New Project" under "Basic Operations".

   The "New Project" dialog box will be displayed.



New Project

Categories: Libraries, Projects

Templates: Standard project

A project containing one device, one application, and an empty implementation for MC_PRG

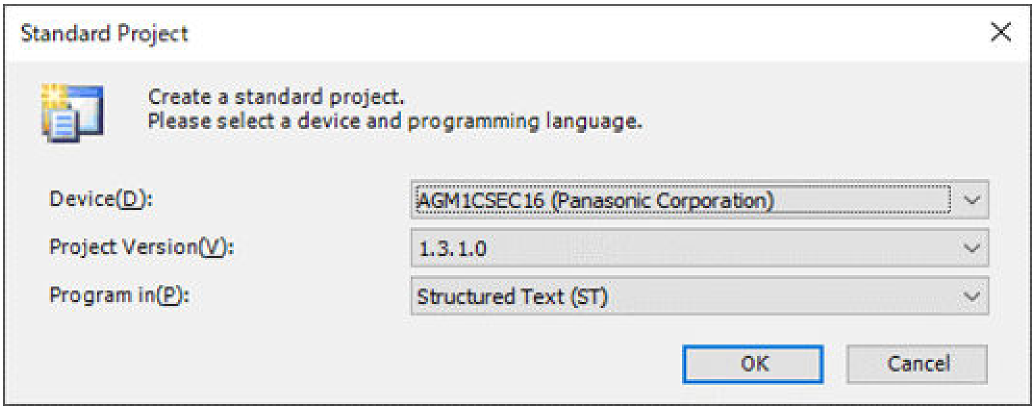
Name: Untitled1

Location: C:¥Users¥Documents

3. Select **Projects**>**Standard project**, and specify a project file name in the "Name" field and a project storage location in the "Location" field.

4. Click the [OK] button.

The "Standard project" dialog box will be displayed.



Standard Project

Create a standard project.
Please select a device and programming language.

Device(D): AGM1CSEC16 (Panasonic Corporation)

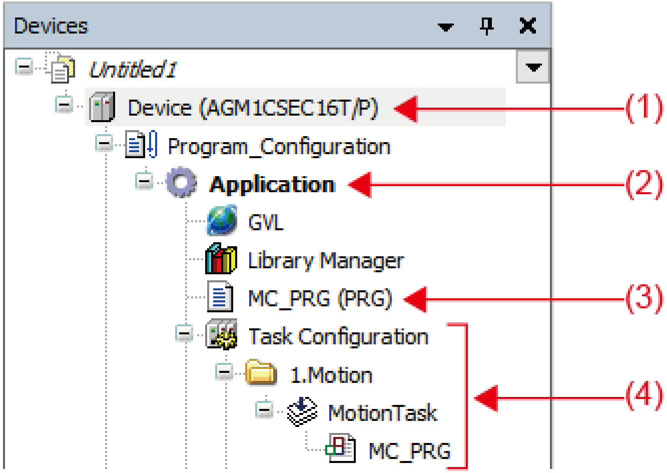
Project Version(V): 1.3.1.0

Program in(P): Structured Text (ST)

5. Select "AGM1CSEC16T/P(Panasonic Corporation)" in the "Device" field and "Structured" in the "Program in" field, and click the [OK] button.

A new project will be created. Device and other objects including objects for ST programs are arranged in the navigator pane.

<Uses of objects arranged in the navigator pane>



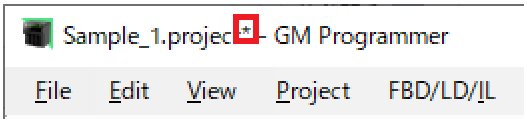
No.	Name	Function
(1)	Device object	Sets up device objects.
(2)	Application object	Sets up application objects.
(3)	Program object (POU object)	Sets up program objects (POU objects).
(4)	Task object	Sets up task objects.

Info.

- A new project can also be created from the menu bar by selecting **File**>**New Project**.

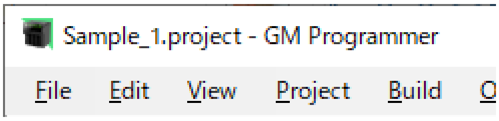
4.2 Saving a Project

Save a project that is created. The project will be saved as a file with extension "".project"". Unsaved projects are indicated by "*" on the right side of their project file names on the title bar.



1 2 Procedure

1. From the menu bar, select **File**>**Save Project**, or press the shortcut keys "Ctrl+s".
   A project that has been created will be saved.
   ""*"" displayed on the right side of the project file name will disappear.



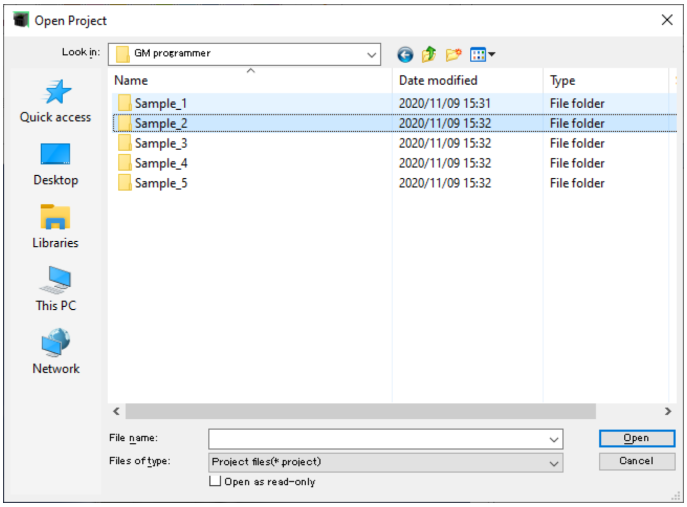
Info.

- Before saving a project, you can change its project name. From the menu bar, select **File**>**Save Project As**.
- Project files can be saved automatically. For details, refer to "4.11 Automatically Saving Project Files".
- Before updating a file, you can save it as a backup file. For details, refer to "4.10 Creating a Backup when a Project Is Saved".

4.3 Opening a Project

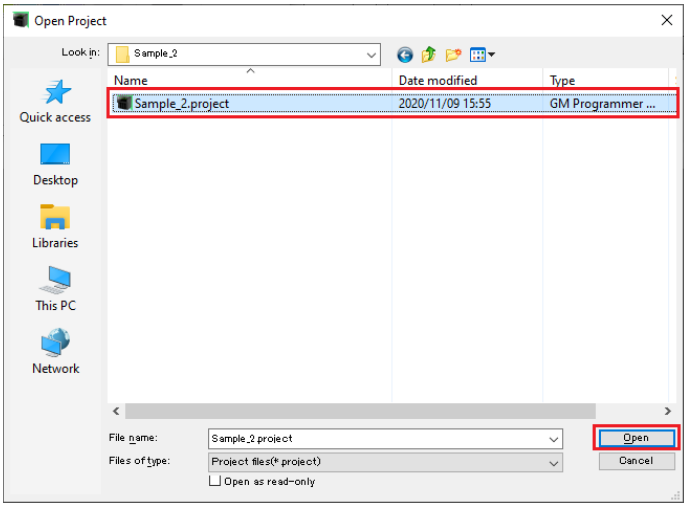
1 2 Procedure

1. From the menu bar, select **File**>**Open Project**.
   The "Open Project" dialog box will be displayed.



Name	Date modified	Type
Sample_1	2020/11/09 15:31	File folder
Sample_2	2020/11/09 15:32	File folder
Sample_3	2020/11/09 15:32	File folder
Sample_4	2020/11/09 15:32	File folder
Sample_5	2020/11/09 15:32	File folder

2. Select a project file and click the [Open] button.
   The selected project file will be opened.



Name	Date modified	Type
Sample_2.project	2020/11/09 15:55	GM Programmer ...

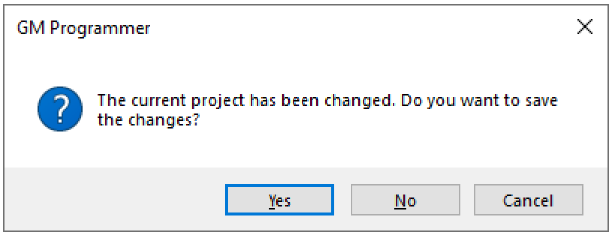
4.4 Closing a Project

1 2 Procedure

1. From the menu bar, select **File**>**Close Project**.
   The project that has been created will be closed.

Info.

- If you select "Close Project" without saving a project file that has been updated, a confirmation dialog box will be displayed, asking whether to save the project. Click the [Yes] button to save the project.

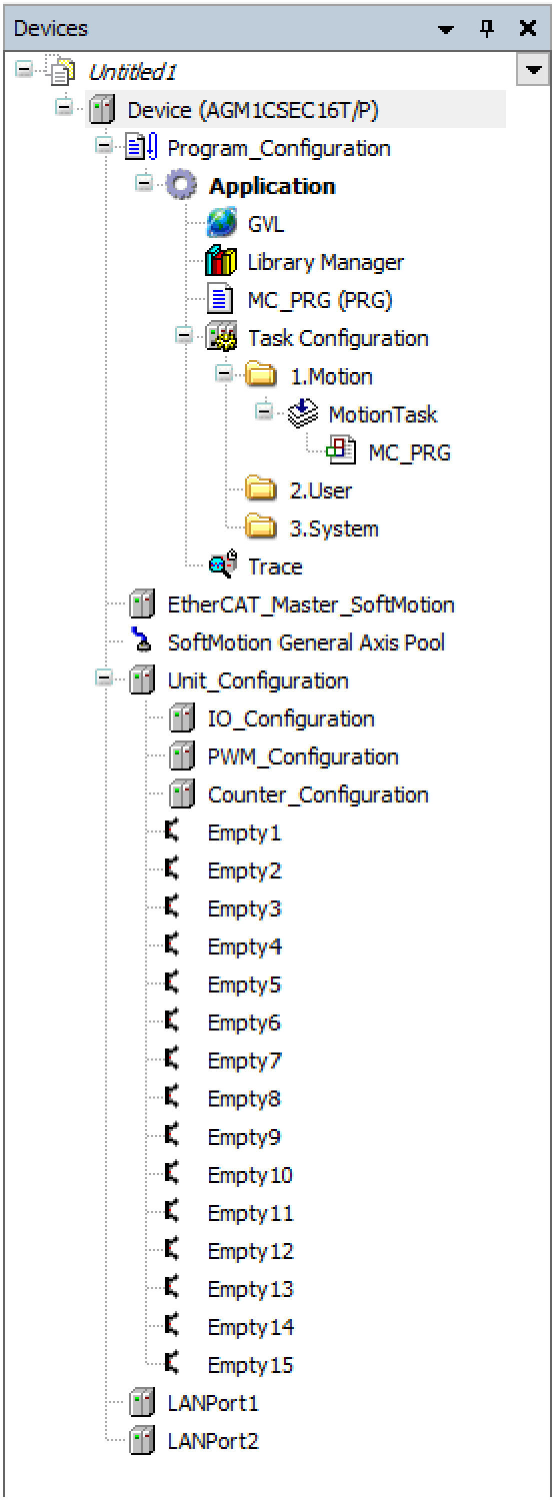


GM Programmer

The current project has been changed. Do you want to save the changes?

4.5 Device Tree Configuration

When a new project is created, it is started in the device tree configuration shown below.



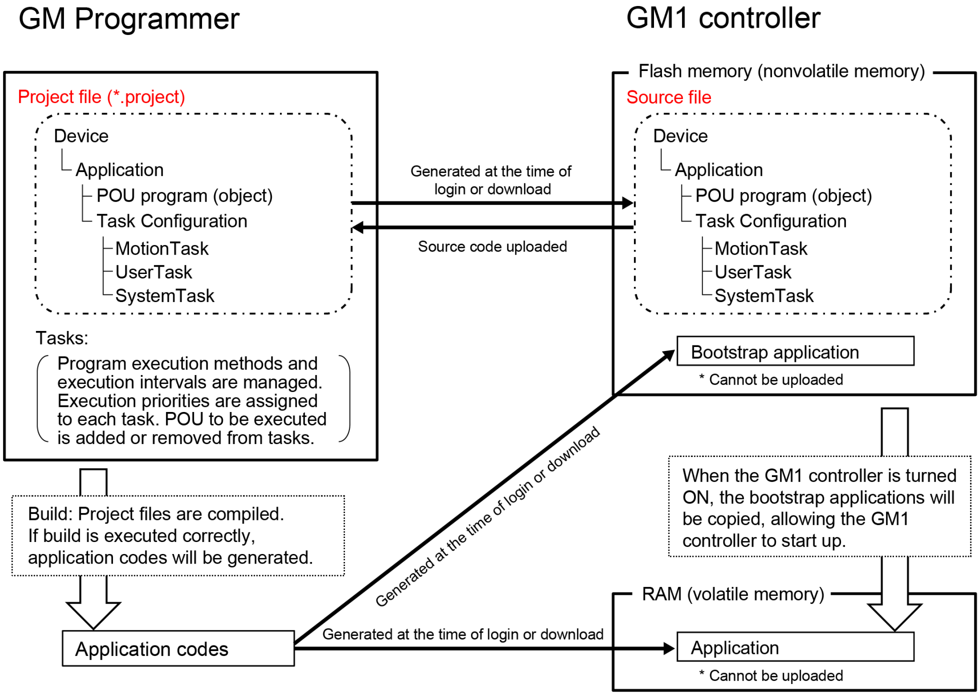
Devices

- *Untitled1*
  - Device (AGM1CSEC16T/P)
    - Program_Configuration
      - **Application**
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Task Configuration
          - 1.Motion
            - MotionTask
              - MC_PRG
          - 2.User
          - 3.System
        - Trace
    - EtherCAT_Master_SoftMotion
    - SoftMotion General Axis Pool
    - Unit_Configuration
      - IO_Configuration
      - PWM_Configuration
      - Counter_Configuration
      - Empty1
      - Empty2
      - Empty3
      - Empty4
      - Empty5
      - Empty6
      - Empty7
      - Empty8
      - Empty9
      - Empty10
      - Empty11
      - Empty12
      - Empty13
      - Empty14
      - Empty15
    - LANPort1
    - LANPort2

Sample1	Project name

Device	GM1 controller device
Program Configuration	Object that defines applications including source codes
Application	Application object
GVL	Global variable object
Library_Manager	List of all libraries linked to the project
MC_PRG(PRG)	Main program
Task Configuration	Configuration of tasks invoking application programs
1.Motion	User program tasks for motion control
MotionTask	
MC_PRG	
2.User	User program task for any control other than motion control
3.System	Task used by the system
Trace	Object that monitors variable data graphically
EtherCAT_Master_SoftMotion	Object that is the parent node of servo amplifiers for EtherCAT
SoftMotion General Axis Pool	Object that is an interface for inserting a free drive unit
Unit_Configuration	Object that is the parent node of I/O related devices
IO_Configuration	General-purpose I/O incorporated in GM1 controller
PWM_Configuration	PWM output incorporated in GM1 controller
Counter_Configuration	High-speed counter incorporated in GM1 controller
Empty1	Objects for adding I/O for expansion unit
Empty2	
Empty3	
Empty4	
Empty5	
Empty6	
Empty7	
Empty8	
Empty9	
Empty10	
Empty11	
Empty12	
Empty13	
Empty14	
Empty15	
LAN Port1	Objects that are the parent node of devices that use the Ethernet protocol
LAN Port2	

4.6 Project Configuration


GM Programmer
GM1 controller
Project file (*.project)
Device
Application
POU program (object)
Task Configuration
MotionTask
UserTask
SystemTask
Tasks:
Program execution methods and execution intervals are managed. Execution priorities are assigned to each task. POU to be executed is added or removed from tasks.
Generated at the time of login or download
Source code uploaded
Flash memory (nonvolatile memory)
Source file
Device
Application
POU program (object)
Task Configuration
MotionTask
UserTask
SystemTask
Bootstrap application
* Cannot be uploaded
Build: Project files are compiled. If build is executed correctly, application codes will be generated.
Generated at the time of login or download
When the GM1 controller is turned ON, the bootstrap applications will be copied, allowing the GM1 controller to start up.
RAM (volatile memory)
Application codes
Generated at the time of login or download
Application
* Cannot be uploaded

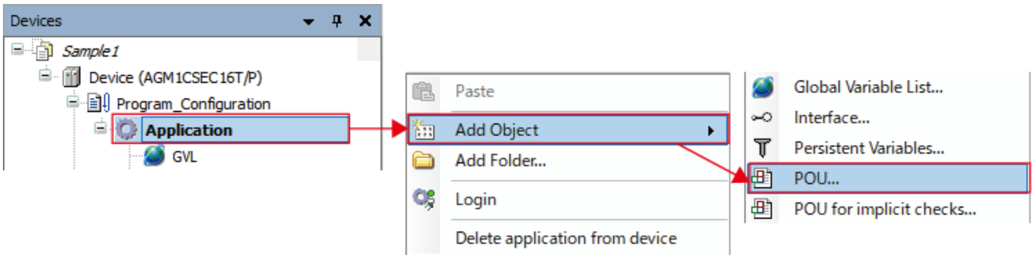
4.7 Adding Objects

Program creation objects (POU objects) and objects with various functions can be added to applications within a project.

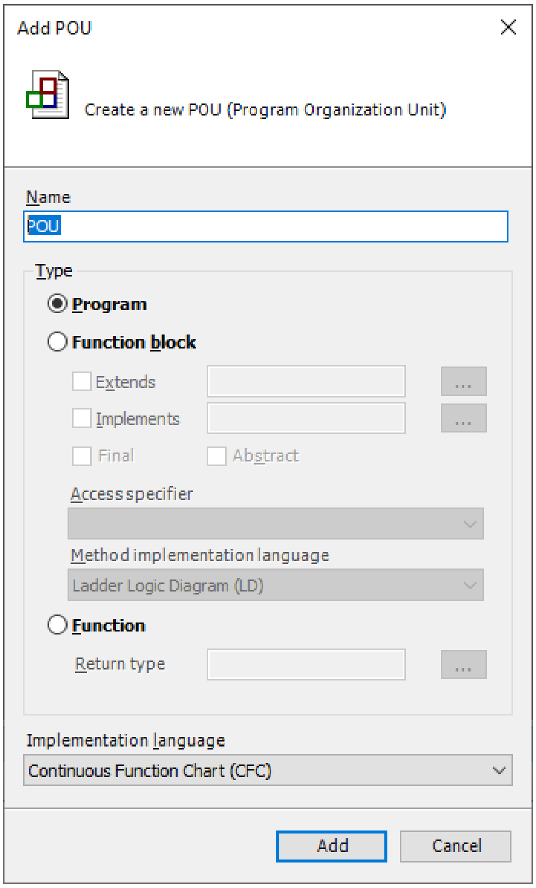
For example, use the following procedure to add POU objects for ST programs.

Procedure

1. Right-click the [Application] object in the navigator pane and then select **Add Object**>**POU** from the context-sensitive menu that is displayed.



The "Add POU" dialog box will be displayed.



Add POU

Create a new POU (Program Organization Unit)

Name: POU

Type

● Program

○ Function block

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

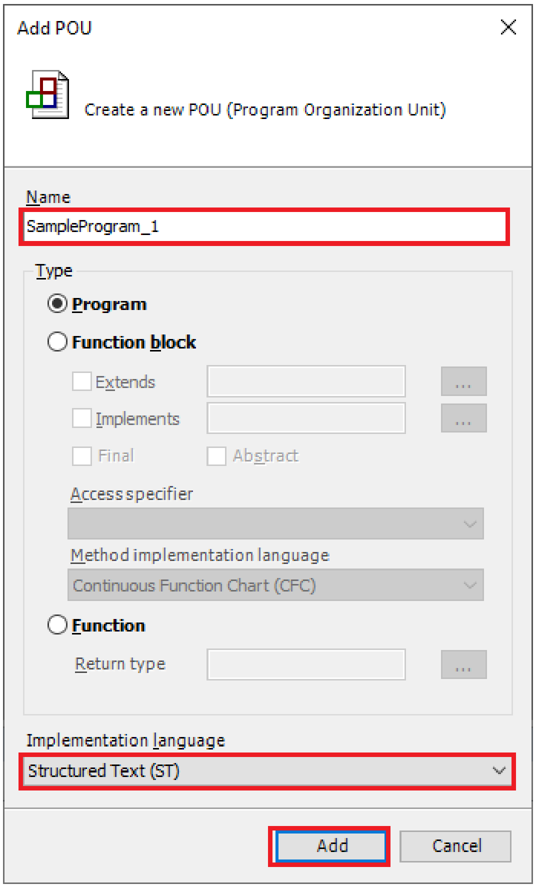
Method implementation language: Ladder Logic Diagram (LD)

○ Function

Return type

Implementation language: Continuous Function Chart (CFC)

2. Enter a program name in the "Name" field, select a programming language from the "Implementation Language" drop-down list, and click the [Add] button.



Add POU

Create a new POU (Program Organization Unit)

Name: SamplePrograrn_1

Type

● Program

○ Function block

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

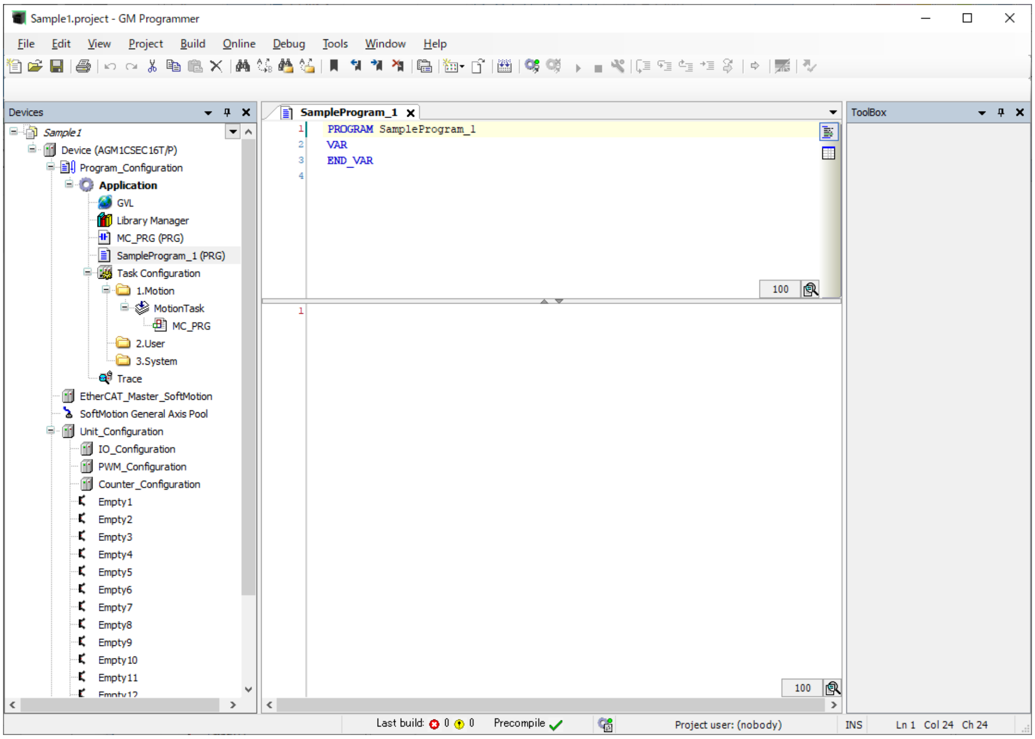
Method implementation language: Continuous Function Chart (CFC)

○ Function

Return type

Implementation language: Structured Text (ST)

A [POU] object for the programming language selected in the "Implementation Language" drop-down list will be added to the navigator pane.



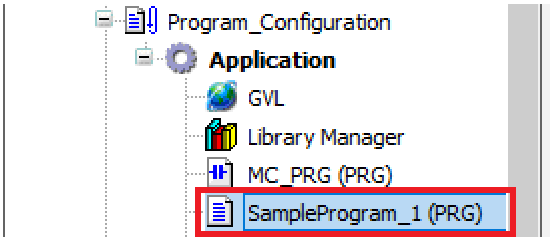
```
PROGRAM SampleProgram_1
VAR
END_VAR
```

Info.

- You can also add objects by selecting the [Application] object in the navigator pane and then selecting **Project**>**Add Object** from the menu bar.
- To remove an object, select the object in the navigation pane and then press the "Delete" key or right-click the object and then select "Delete" from the context-sensitive menu that is displayed.
- You can add folders under the [Application] object in the navigation pane. By adding objects under each folder, you can create a hierarchical structure.

  Right-click the [Application] object in the navigator pane and then select "Add Folder" from the context-sensitive menu that is displayed. The "Add Folder" dialog box will be displayed. Enter a folder name and click the [OK] button.

Example: When a [POU] object is added under an added folder (program for project A)



- You can also add an object for creating functions and function blocks. For details, refer to "6.5 Function and Function Block".

4.7.1 List of objects

Check the following list for objects that can be added.

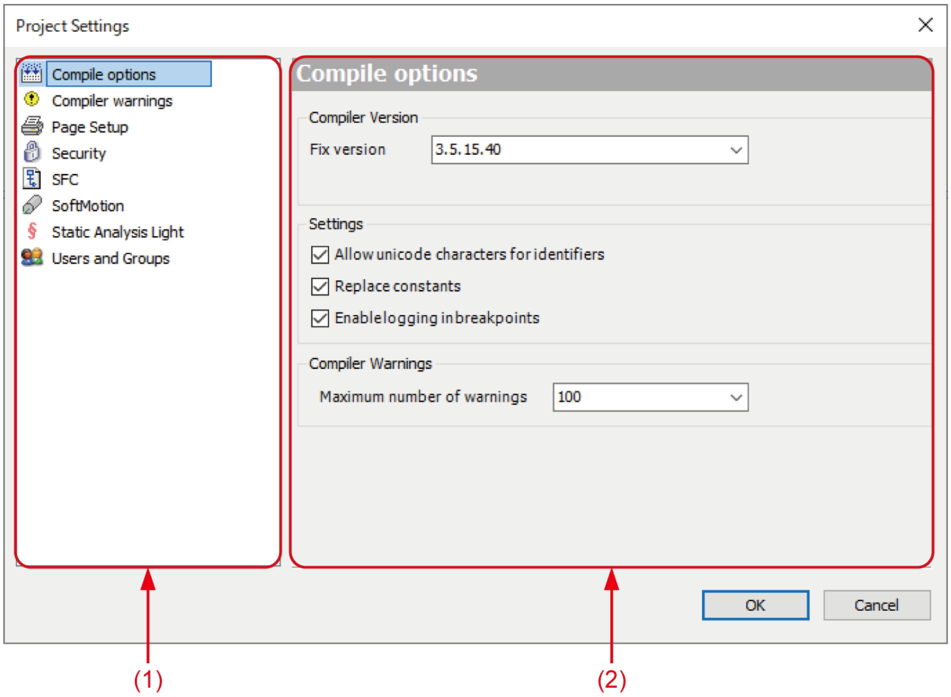
Addition source	Added object
Device	DeviceTrace
Application	CNC program CNC setting DUT POU Interface Cam table Global variable list Symbol Configuration Trace Visualization Recipe manager External file Extended Cam table Persistent variable POU for implicit checks
POU	Action Transition Property Methods
Interface	Interface property Interface method
Global Variables	Property
Persistent variable	Property
MotionTask	Program call
"2.User" folder	UserTask

4.8 Setting up A Project

Projects can be set up using the "Project Settings" dialog box. The settings will be applied to only the project that is set up.

Procedure

1. From the menu bar, select **Project**>**Project Settings**.
   The "Project Settings" dialog box will be displayed.



Compile options

Compiler Version

Fix version: 3.5.15.40

Settings

☑ Allow unicode characters for identifiers

☑ Replace constants

☑ Enablelogging inbreakpoints

Compiler Warnings

Maximum number of warnings: 100

No.	Name	Function
(1)	Categories pane	Displays project setting categories.
(2)	Setting pane	Displays the settings of the selected category and allows the user to configure settings.

Project setting categories

Category name	Function	Reference page
SFC	Allows the user to configure settings related to variables used in SFC and code generation.	"P.7-24"
SoftMotion	Displays the version of the SoftMotion package.	"P.5-11"
Sraric Analysis Light	Allows the user to configure settings regarding whether to enable code analysis during code generation.	-
Compiler warnings	Allows the user to configure settings regarding whether to enable warnings output during build.	"P.10-42"

Category name	Function	Reference page
Compile options	Allows the user to configure build-related settings such as whether to enable Unicode for the program objects (POU objects) and the number of compiler warnings to be output.	"P.9-7"
Security	Allows the user to configure settings related to project file encryption using passwords.	"P.10-16"
Page Setup	Allows the user to configure printing-related settings.	-
Users and groups	Allows the user to add, edit, and remove users and groups for project user management.	"P.10-4"

2. Select a desired category from the category pane.
   The setting items for the selected category will be displayed in the setting pane.

3. Change the setting items as appropriate and click the [OK] button.
   The setting items will be applied.

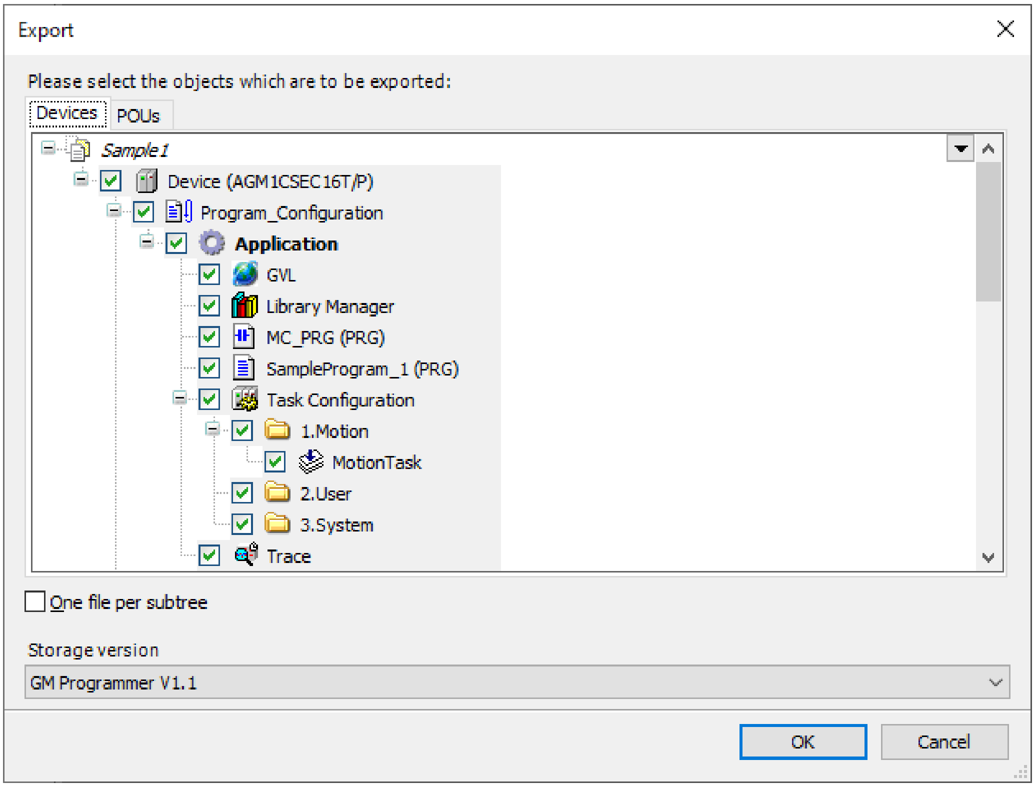
4.9 Exporting and Importing Objects

Objects in a project can be exported as XML files. The extension of exported files is ".export". Files that are being exported can be imported to GM Programmer.

4.9.1 Exporting Objects

1 2 Procedure

1. From the menu bar, select **Project**>**Export**.
The "Export" dialog box will be displayed.



Please select the objects which are to be exported:

- Sample1
  - ☑ Device (AGM1CSEC16T/P)
    - ☑ Program_Configuration
      - ☑ **Application**
        - ☑ GVL
        - ☑ Library Manager
        - ☑ MC_PRG (PRG)
        - ☑ SampleProgram_1 (PRG)
        - ☑ Task Configuration
          - ☑ 1.Motion
            - ☑ MotionTask
          - ☑ 2.User
          - ☑ 3.System
        - ☑ Trace

☐ One file per subtree

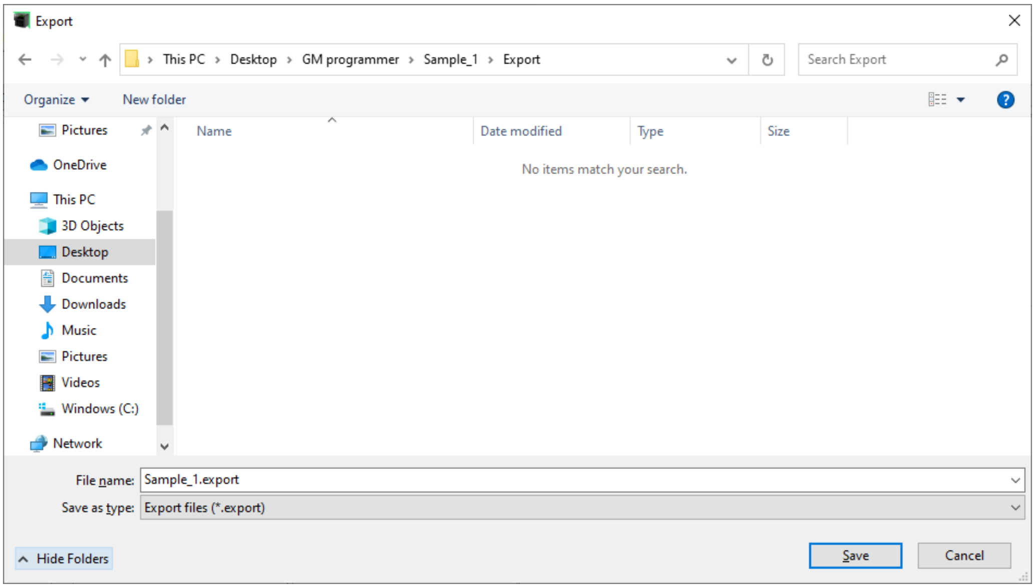
Storage version
GM Programmer V1.1

Info.

- To import an object into GM Programmer, select only one object below the [Application] object and export it.

2. Select objects to be exported.
Normally, there is no need to make changes.

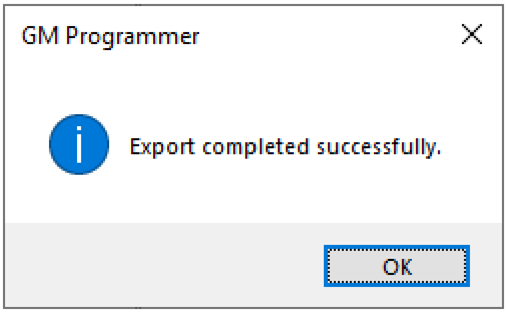
3. Click the [OK] button.
The "Export" dialog box will be displayed.
If necessary, change the file name and save destination.



File name: Sample_1.export

Save as type: Export files (*.export)

4. Click the [Save] button.
   Export will be executed.



Export completed successfully.

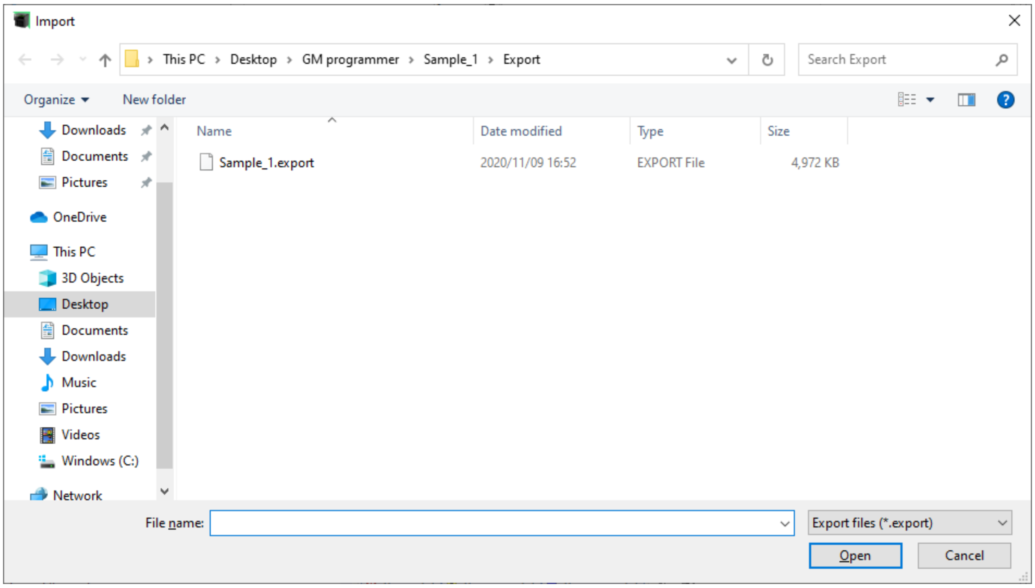
4.9.2 Importing Objects

This section explains the procedure for importing objects exported to a project into GM Programmer.

For example, use the following procedure to import objects below the "Application" object.

Procedure

1. Select the [Application] object in the navigator pane and then select **Project>Import** from the menu bar.
   The "Import" dialog box will be displayed.

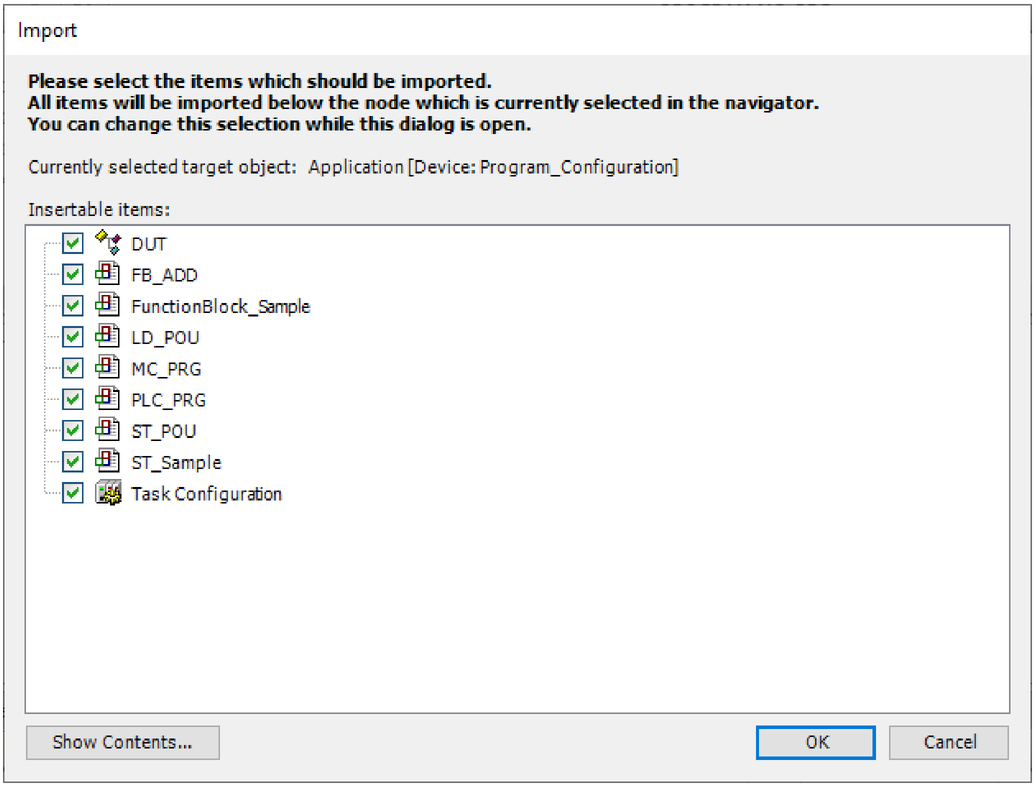


Name	Date modified	Type	Size
Sample_1.export	2020/11/09 16:52	EXPORT File	4,972 KB

File name:

Export files (*.export)

2. Select a file with extension ".export" and click the [Open] button.
The "Import" dialog box will be displayed.
Objects that can be imported will be displayed in the "Insertable items" area.



Please select the items which should be imported.
All items will be imported below the node which is currently selected in the navigator.
You can change this selection while this dialog is open.

Currently selected target object: Application [Device: Program_Configuration]

Insertable items:

- ☑ DUT
- ☑ FB_ADD
- ☑ FunctionBlock_Sample
- ☑ LD_POU
- ☑ MC_PRG
- ☑ PLC_PRG
- ☑ ST_POU
- ☑ ST_Sample
- ☑ Task Configuration

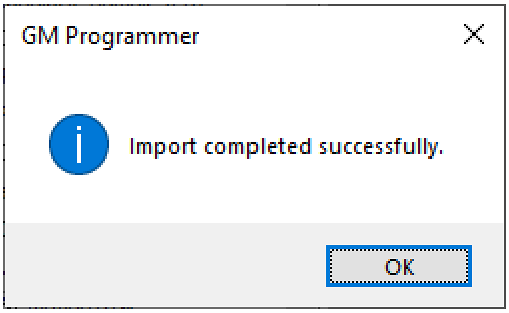
Info.

- To import an object into GM Programmer, select an export file where only one object below the [Application] object is selected.

3. Clear the check boxes of the objects that do not need to be imported and click the [OK] button.

Import will be executed.

The objects that have been imported will be displayed below the [Application] object in the navigator pane.


GM Programmer

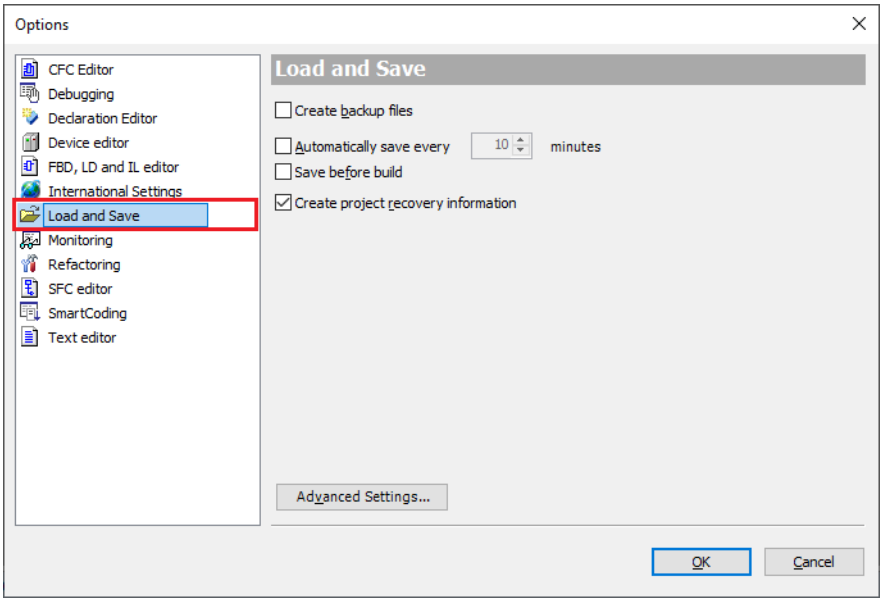
Import completed successfully.

4.10 Creating a Backup when a Project Is Saved

When saving a project, you can create a backup file of the project file to be updated. The extension of backup files is ".backup".

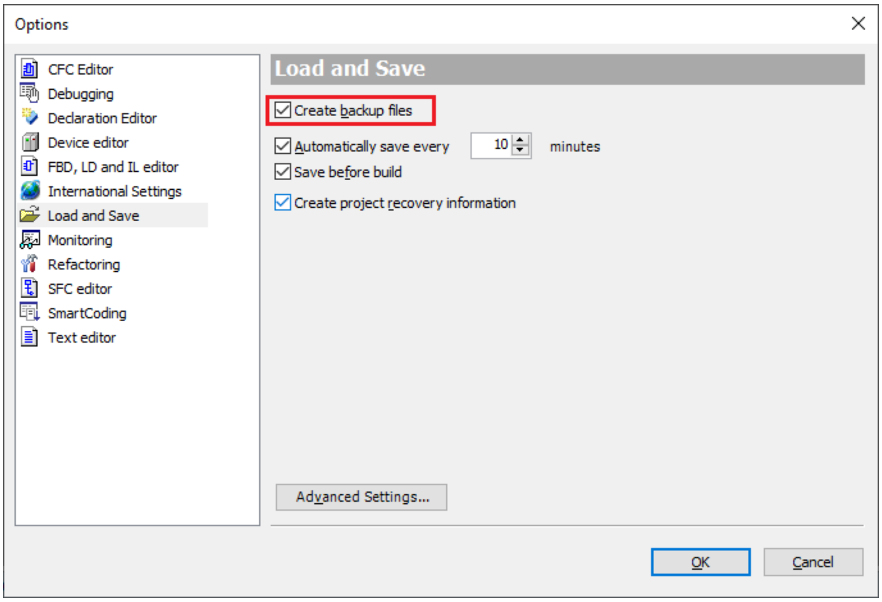
1 2 Procedure

1. From the menu bar, select **Tools**>**Options**.
   The "Options" dialog box will be displayed.
2. Select the "Load and Save" category.
   The "Load and Save" pane will be displayed.



□ Create backup files
□ Automatically save every 10 minutes
□ Save before build
☑ Create project recovery information

3. Select the "Create backup files" check box and click the [OK] button.
   Then, whenever a project is saved, the project file to be updated will be automatically saved as a backup file (with extension ".backup").



Options

Load and Save

☑ Create backup files

☑ Automatically save every 10 minutes

☑ Save before build

☑ Create project recovery information

<Restoring backup files>

To restore a project file that has been backed up, manually change the extension of the file from ".backup" to ".project" and then open the project file in GM Programmer.

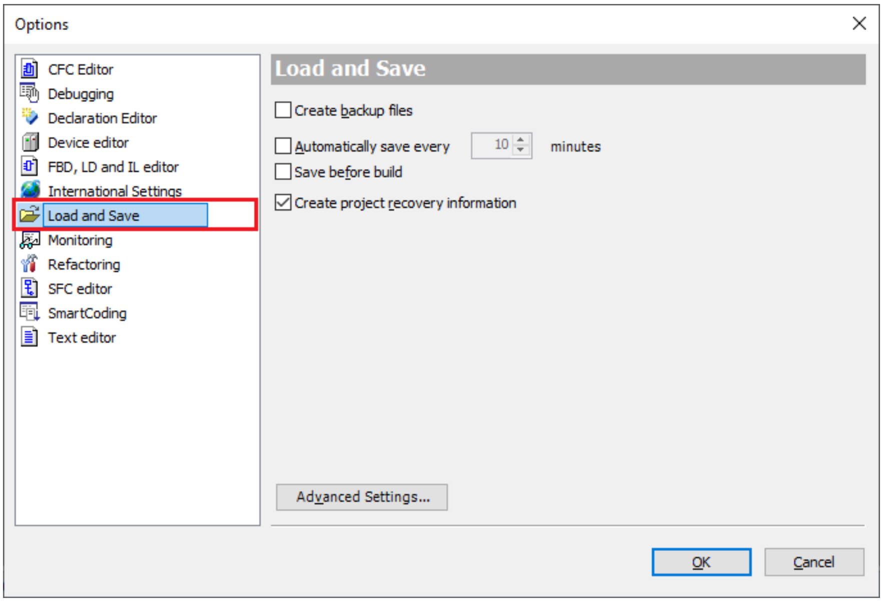
4.11 Automatically Saving Project Files

Project files can be saved automatically during editing. Even if data disappears when GM Programmer terminates abnormally, a file up to the point in time when it was saved automatically can be restored. The extension of backup files is ".autosave".

Procedure

1. From the menu bar, select **Tools**>**Options**.
   The "Options" dialog box will be displayed.

2. Select the "Load and Save" category.
   The "Load and Save" pane will be displayed.



Load and Save

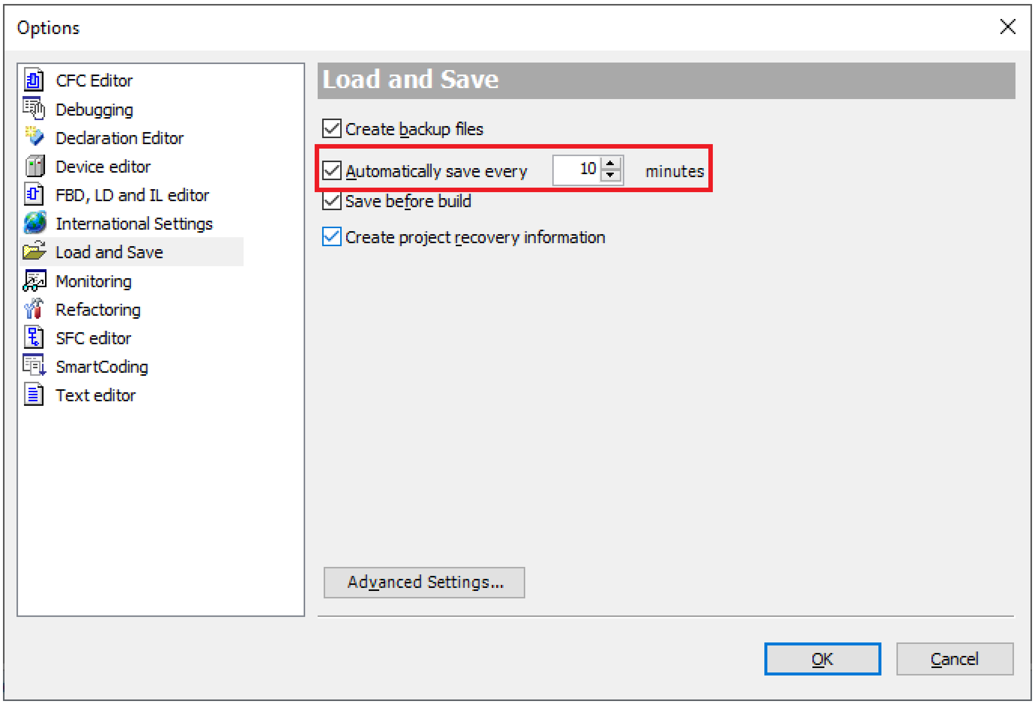
□ Create backup files

□ Automatically save every 10 minutes

□ Save before build

☑ Create project recovery information

3. Select the "Automatically save evey" check box and click the [OK] button.
   The automatic save interval can be changed in minutes (default value: 10 minutes).



Options

Load and Save

☑ Create backup files

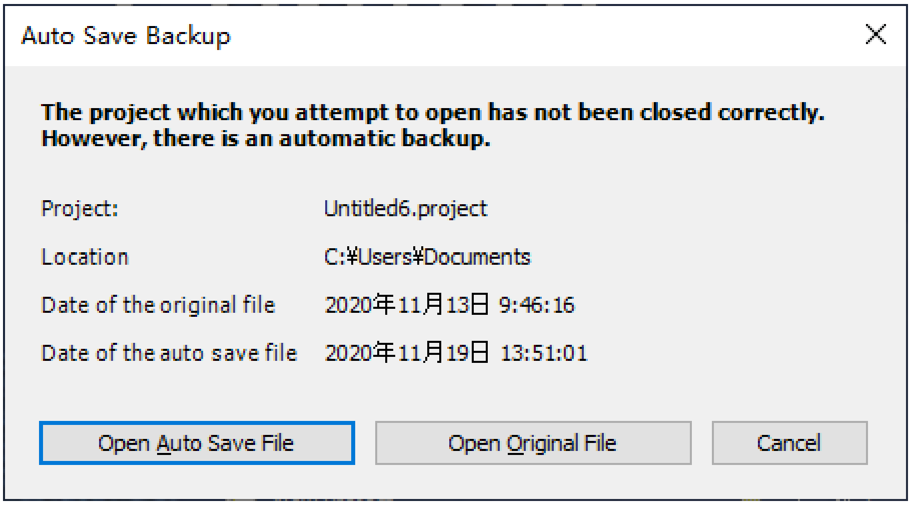
☑ Automatically save every 10 minutes

☑ Save before build

☑ Create project recovery information

After auto save is set, project files will be automatically saved as files with extension ".autosave" at the specified interval during editing.

After a project file is closed due to abnormal termination of GM Programmer, when you open the project file again, you can select either the original project file with extension ".project" or the automatically saved project file with extension ".autosave". To open the automatically saved project file, click the [Open Auto Save File] button.



Auto Save Backup

The project which you attempt to open has not been closed correctly. However, there is an automatic backup.

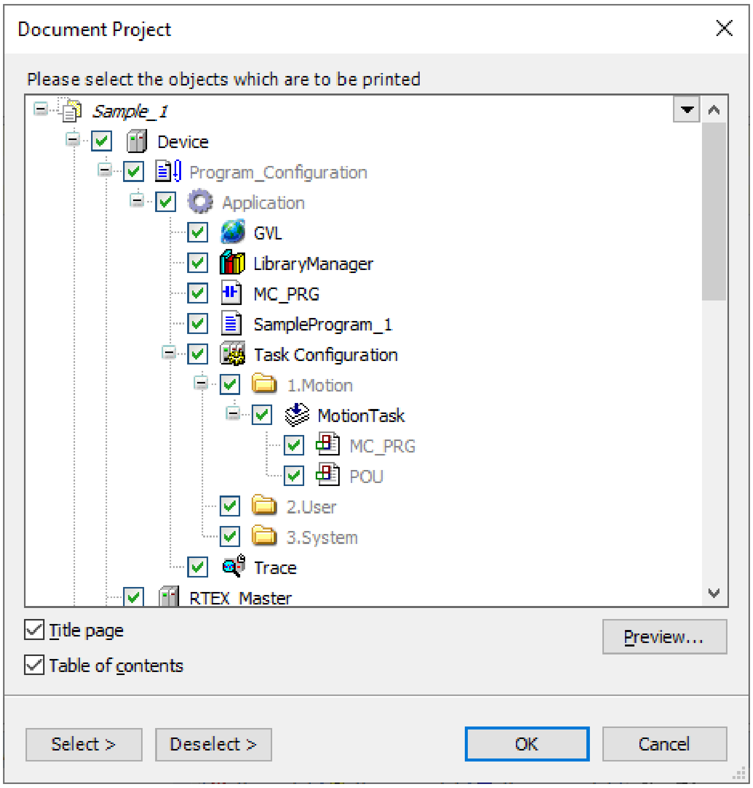
Project:	Untitled6.project
Location	C:¥Users¥Documents
Date of the original file	2020年11月13日 9:46:16
Date of the auto save file	2020年11月19日 13:51:01

4.12 Printing a Project

You can print the entire project.

Procedure

1. From the menu bar, select **Project**>**Document**.
   The "Document Project" dialog box will be displayed.



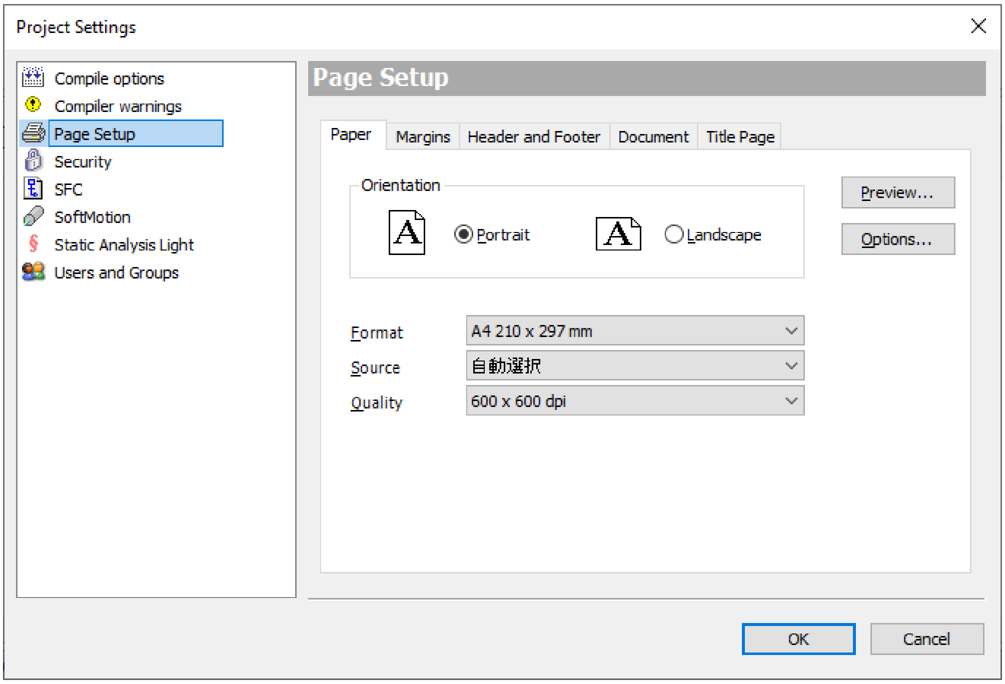
2. Select the check boxes of the devices and objects for which documents are to be printed, and click the [OK] button.
   The "Print" dialog box will be displayed. Select a printer to be used and click the [OK] button.
   Documents will be printed.

Info.

- By clicking the [Preview] button in the "Document Project" dialog box, a print preview window is displayed, so that you can check what a hard copy would look like when printed.
- You can edit page orientation, margin, header, footer, table of contents, and title page details.

  From the menu bar, select **File**>**Page Setup**. The "Project Settings" dialog box will appear with the Page Setup pane displayed.

  Select the tab of the item to be changed and change the settings.



Project Settings

- Compile options
- Compiler warnings
- Page Setup
- Security
- SFC
- SoftMotion
- Static Analysis Light
- Users and Groups

Page Setup

Paper | Margins | Header and Footer | Document | Title Page

Orientation: ● Portrait ○ Landscape

Format: A4 210 x 297 mm

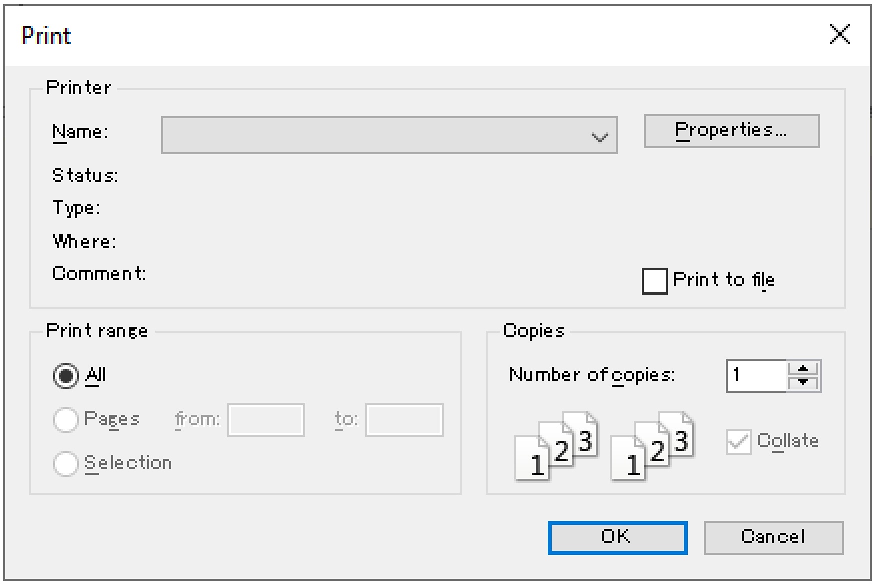
Source: 自動選択

Quality: 600 x 600 dpi

4.13 Printing an Object within a Project

Procedure

1. With the object editor opened, from the menu bar, select **File**>**Print**.
   The "Print" dialog box will be displayed. Execute printing.



Print

Printer

Name:

Properties...

Status:

Type:

Where:

Comment:

□ Print to file

Print range

● All

○ Pages from: to:

○ Selection

Copies

Number of copies: 1

☑ Collate

OK

Cancel

4.14 Comparing Projects

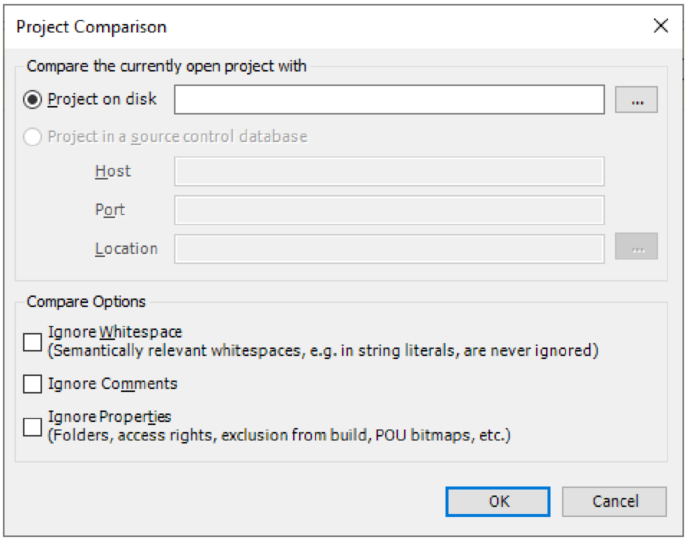
You can compare the opened project file with another project file to display and merge the differences between them.

4.14.1 Project Comparison Method

Compare the opened project file with another project file.

Procedure

1. From the menu bar, select **Project**>**Compare**.
   The "Project Comparison" dialog box will be displayed.



Project Comparison

Compare the currently open project with

● Project on disk

○ Project in a source control database

Host

Port

Location

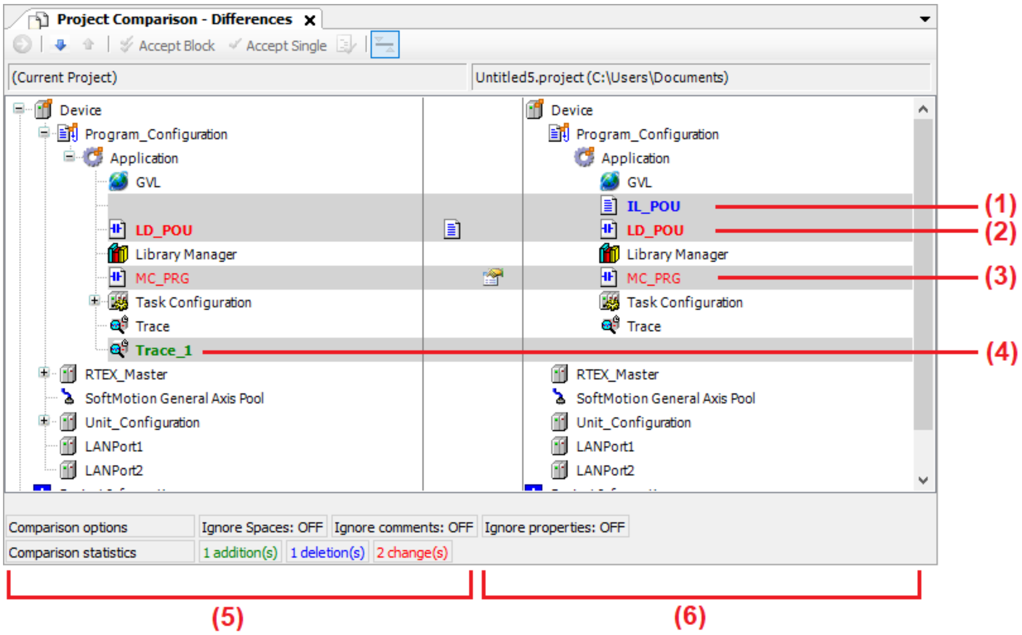
Compare Options

□ Ignore Whitespace
(Semantically relevant whitespaces, e.g. in string literals, are never ignored)

□ Ignore Comments

□ Ignore Properties
(Folders, access rights, exclusion from build, POU bitmaps, etc.)

2. In the "Project on disk" field, specify a project file to be compared with and click the [OK] button.
   The comparison results will be displayed in the main pane.
   The backgrounds of object lines with differences are displayed in gray. The text color of each object indicates the type of difference.



(Current Project)

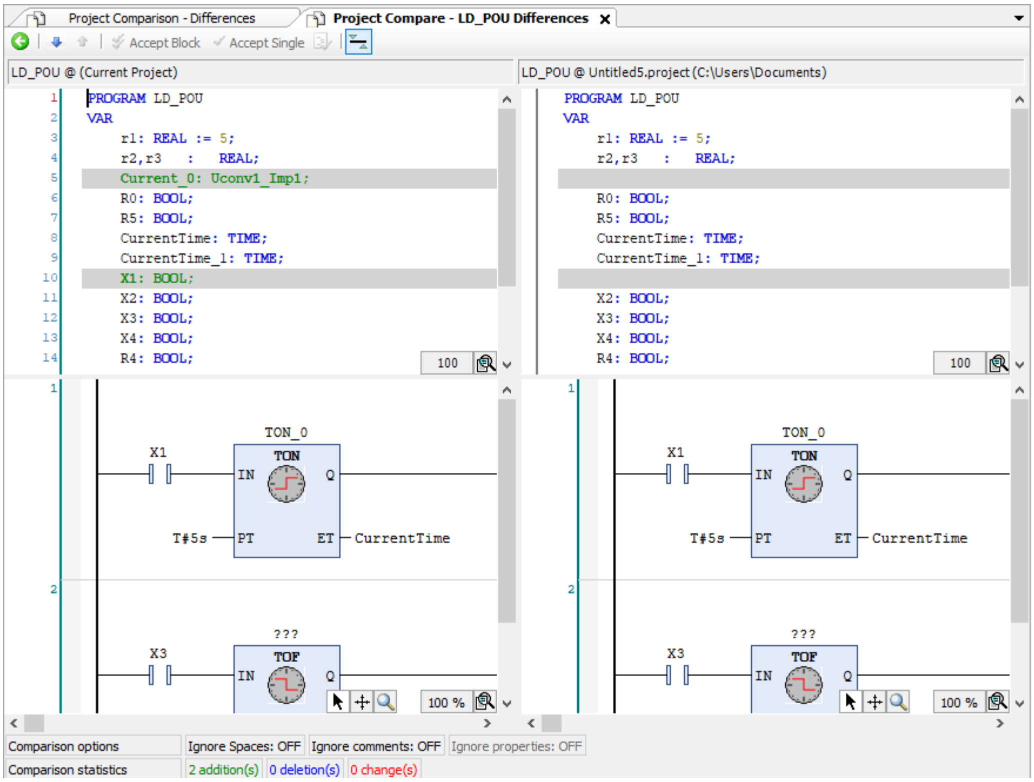
Untitled5.project (C:\Users\Documents)

Comparison options: Ignore Spaces: OFF | Ignore comments: OFF | Ignore properties: OFF

Comparison statistics: 1 addition(s) | 1 deletion(s) | 2 change(s)

No.	Description
(1)	Removed object
(2)	Object with a difference in contents
(3)	Object with a difference in properties
(4)	Added object
(5)	Opened project
(6)	Specified project

Double-clicking an object line with a difference in contents displays the object comparison results.



LD_POU @ (Current Project)

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;
    Current_0: Uconv1_Imp1;
    R0: BOOL;
    R5: BOOL;
    CurrentTime: TIME;
    CurrentTime_1: TIME;
    X1: BOOL;
    X2: BOOL;
    X3: BOOL;
    X4: BOOL;
    R4: BOOL;
```

LD_POU @ Untitled5.project (C:\Users\Documents)

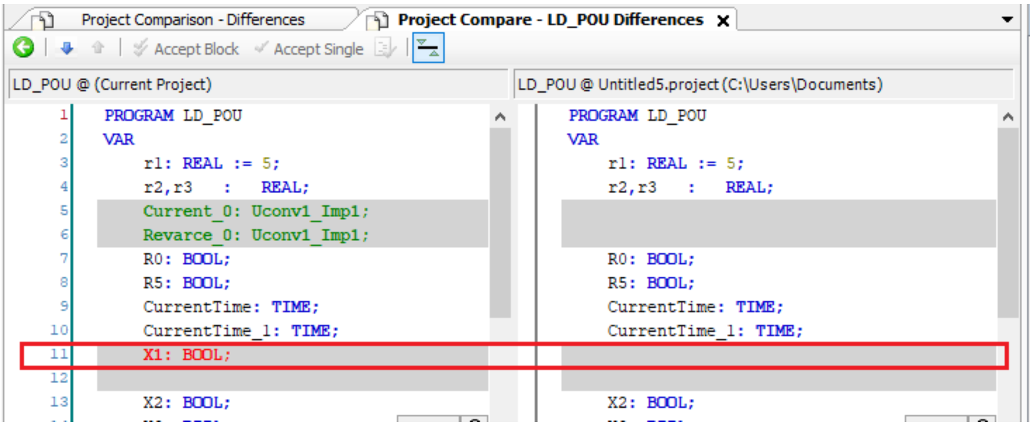
```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;

    R0: BOOL;
    R5: BOOL;
    CurrentTime: TIME;
    CurrentTime_1: TIME;

    X2: BOOL;
    X3: BOOL;
    X4: BOOL;
    R4: BOOL;
```

Comparison options: Ignore Spaces: OFF | Ignore comments: OFF | Ignore properties: OFF

Comparison statistics: 2 addition(s) | 0 deletion(s) | 0 change(s)

Clicking the icon returns the display to the comparison view in the navigator pane.

Clicking the icon switches to the mode in which comparison results including removed object lines are displayed. The above window in which comparison results are displayed will be switched to the window shown below. If the differences displayed as variable bVar0 and empty lines that have been added differ from the specifications of variable bVar0, differences will be displayed as changes in variable bVar0.


Project Comparison - Differences
Project Compare - LD_POU Differences
Accept Block
Accept Single

LD_POU @ (Current Project)

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;
    Current_0: Uconv1_Imp1;
    Revarce_0: Uconv1_Imp1;
    R0: BOOL;
    R5: BOOL;
    CurrentTime: TIME;
    CurrentTime_1: TIME;
    X1: BOOL;

    X2: BOOL;
```

LD_POU @ Untitled5.project (C:\Users\Documents)

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;


    R0: BOOL;
    R5: BOOL;
    CurrentTime: TIME;
    CurrentTime_1: TIME;


    X2: BOOL;
```

4.14.2 Merging Differences

Differences displayed by comparing projects can be merged into the opened project.

When there are differences in the contents of objects, the differences can be merged by using the [Accept Block] button or [Accept Single] button.

For example, use the following procedure to merge the differences in the contents of programs.

1 2 Procedure

1. Perform project comparison.

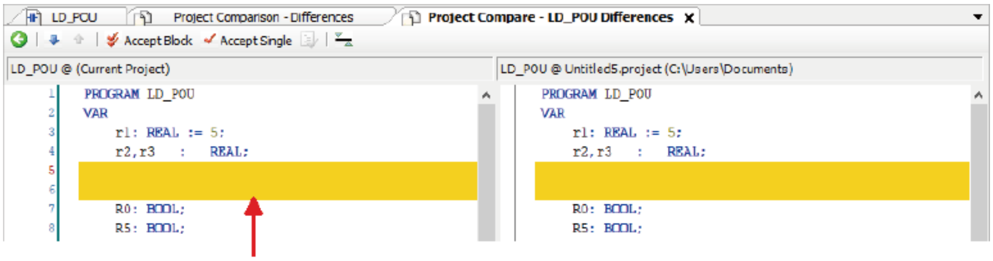
Project comparison results will be displayed.

For details on how to perform project comparison, refer to "4.14.1 Project Comparison Method".

2. Move the cursor to the difference location and click the [Accept Block] button or [Accept Single] button.

The difference will be merged into the opened project. The background of the merged section will be displayed in yellow. At this stage, the merged content has not been reflected in the project yet. To reflect the merged content, approval operation is required as below.

[Accept Block] button: Merges the entire block containing the difference at the cursor position

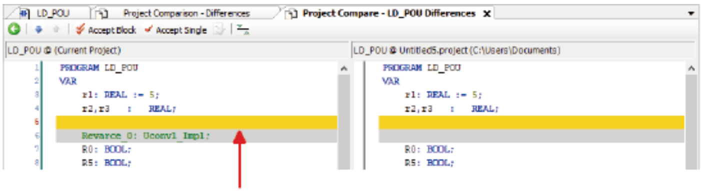

LD_POU
Project Comparison - Differences
Project Compare - LD_POU Differences
Accept Block
Accept Single
LD_POU @ (Current Project)
LD_POU @ Untitled5.project (C:\Users\Documents)

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;


    R0: BOOL;
    R5: BOOL;
```

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;


    R0: BOOL;
    R5: BOOL;
```

Merge the entire block containing the difference at the cursor position (delete two lines)

[Accept Single] button: Merges the difference at the cursor position


LD_POU
Project Comparison - Differences
Project Compare - LD_POU Differences
Accept Block
Accept Single
LD_POU @ (Current Project)
LD_POU @ Untitled5.project (C:\Users\Documents)

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;

    Revarce_0: Uconv1_Impl;
    R0: BOOL;
    R5: BOOL;
```

```
PROGRAM LD_POU
VAR
    r1: REAL := 5;
    r2,r3   :   REAL;


    R0: BOOL;
    R5: BOOL;
```

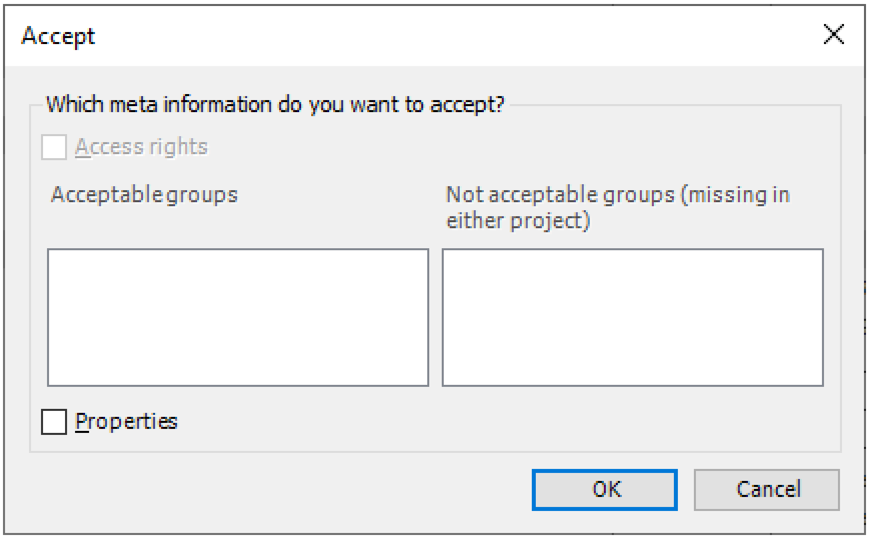
Merge the difference at the cursor position (delete one line)

3. From the menu bar, select **Project>Commit accepted changes**.

The merged content will be approved and reflected in the opened project. Because the reflected content has not been saved yet, save the project as necessary.

Info.

- To merge between objects whose properties are different, select an object with differences and click the icon. The "Accept" dialog box will be displayed. Select the "Properties" check box and click the [OK] button.



Accept

Which meta information do you want to accept?

□ Access rights

Acceptable groups

Not acceptable groups (missing in either project)

□ Properties

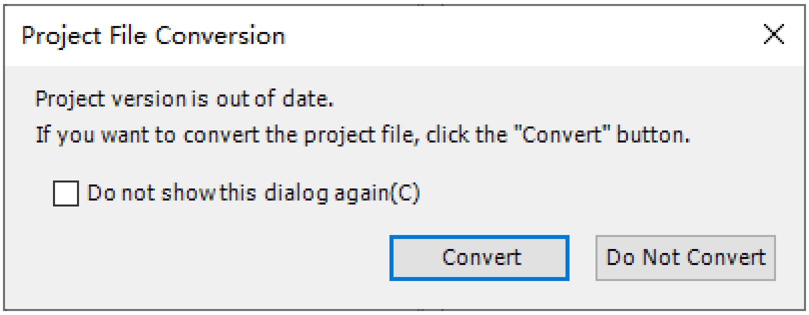
4.15 Project File Conversion

4.15.1 Converting Project File (.project) Version

Example: Converting the project version of a RTEX project file from 1.2.2.0 to 1.4.1.0

1 2 Procedure

1. Open a project file.
   The “Project File Conversion” dialog box will be displayed.

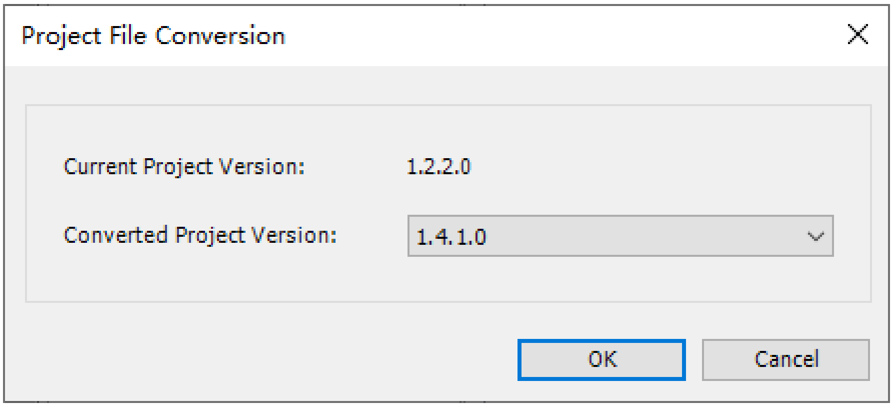


Project File Conversion

Project version is out of date.
If you want to convert the project file, click the "Convert" button.

□ Do not show this dialog again(C)

2. Click the [Convert] button.
   The Project File Conversion dialog box will be displayed.
   Confirm that the converted project version is shown at “Project version after conversion”.

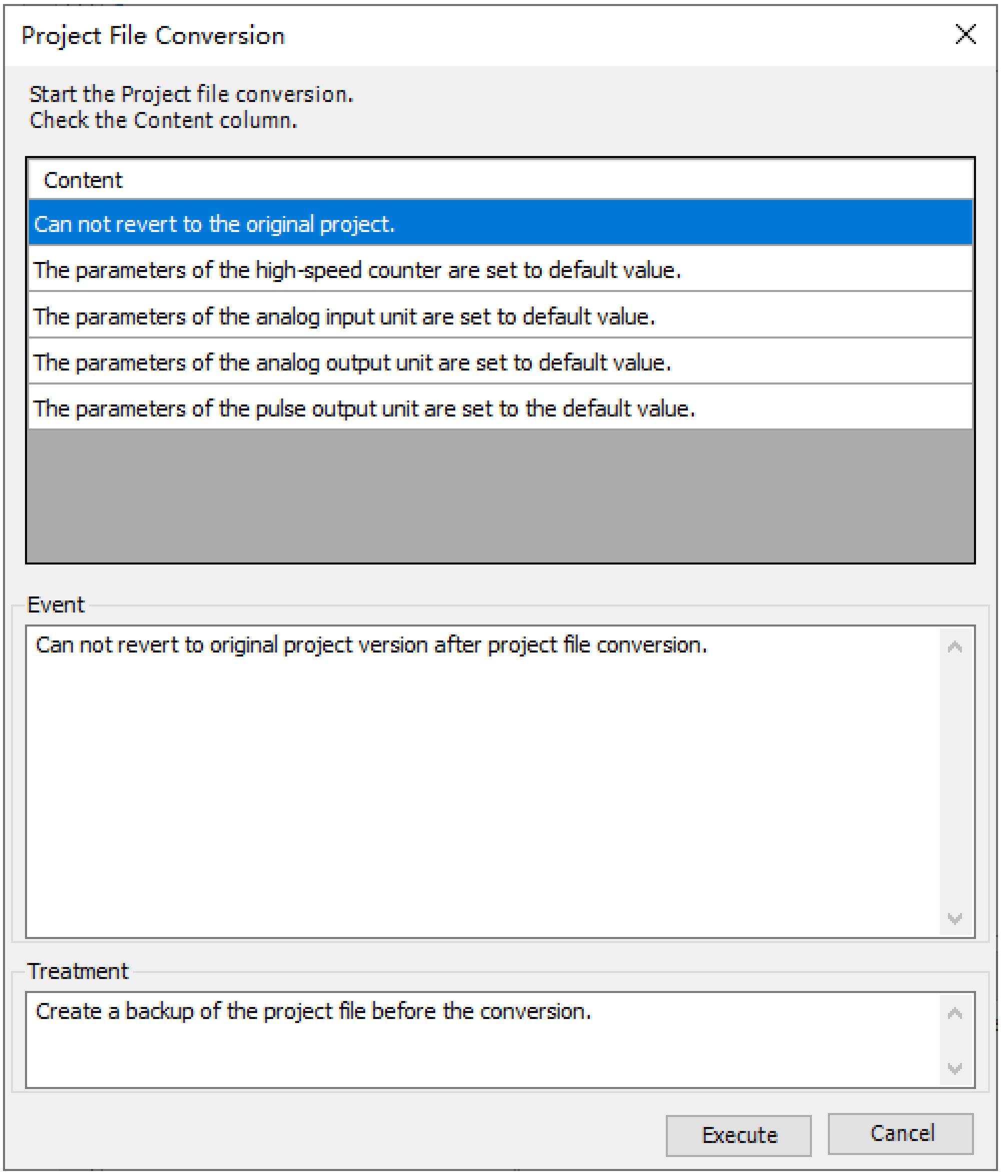


Project File Conversion

Current Project Version: 1.2.2.0

Converted Project Version: 1.4.1.0

3. Click the [OK] button.
   A confirmation dialog box will be displayed. Check the confirmation items about project file conversion.



Start the Project file conversion.
Check the Content column.

Content
Can not revert to the original project.
The parameters of the high-speed counter are set to default value.
The parameters of the analog input unit are set to default value.
The parameters of the analog output unit are set to default value.
The parameters of the pulse output unit are set to the default value.

Event

Can not revert to original project version after project file conversion.

Treatment

Create a backup of the project file before the conversion.

4. Click the [Execute] button.
Conversion of the project file starts.

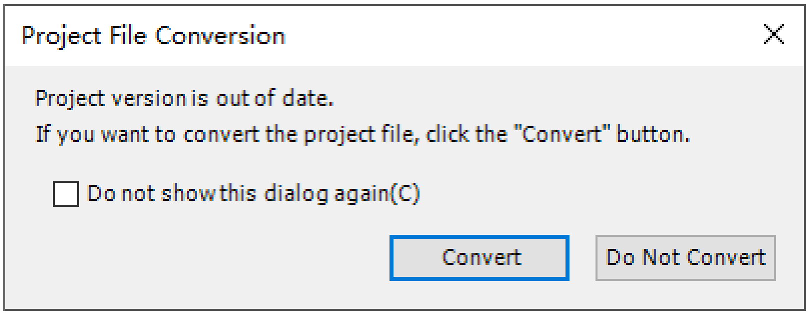
Info.

- The project version of project files cannot be converted to any project version older than the current project version before conversion.
- You can also convert the version of a project file by selecting Project > Project File Conversion from the menu bar.

4.15.2 Converting Library Project File (.library)

1 2 Procedure

1. Open a library project file.
The "Project File Conversion" dialog box will be displayed.

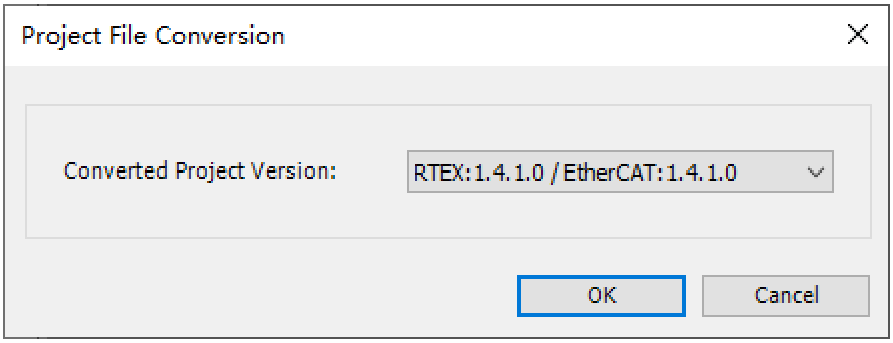


Project File Conversion

Project version is out of date.
If you want to convert the project file, click the "Convert" button.

□ Do not show this dialog again(C)

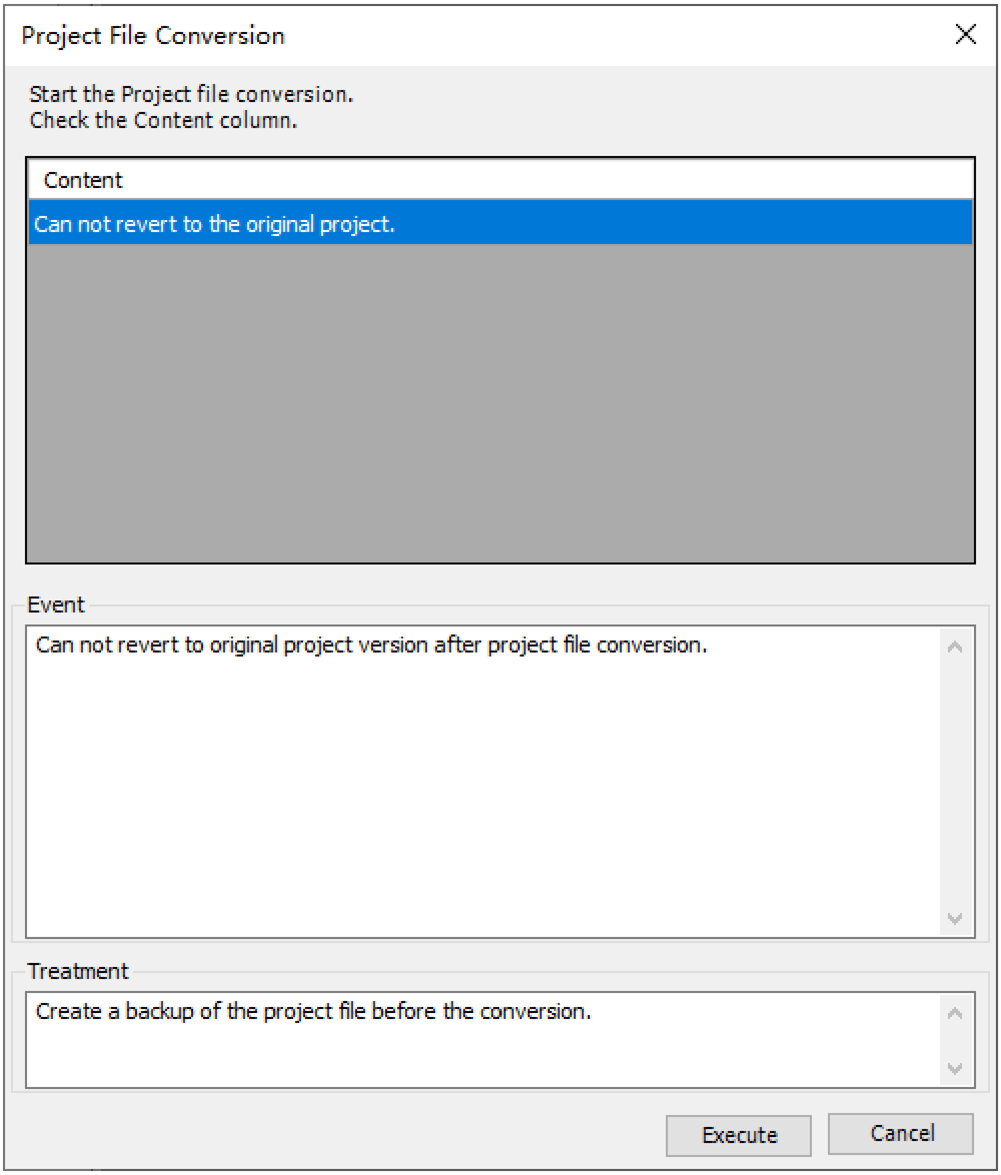
2. Click the [Convert] button.
The Project File Conversion dialog box will be displayed.
Confirm that the converted project version is shown at "Project version after conversion".



Project File Conversion

Converted Project Version: RTEX:1.4.1.0 / EtherCAT:1.4.1.0

3. Click the [OK] button.
A confirmation dialog box will be displayed. Check the confirmation items about project file conversion.



Project File Conversion

Start the Project file conversion.
Check the Content column.

Content
Can not revert to the original project.

Event

Can not revert to original project version after project file conversion.

Treatment

Create a backup of the project file before the conversion.

4. Click the [Execute] button.
Conversion of the project file starts.

Info.

- The project version of project files cannot be converted to any project version older than the current project version before conversion.
- You can also convert the version of a project file by selecting Project > Project File Conversion from the menu bar.

4.16 Saving and unzipping the project archive

You can save project files and user libraries that have been added to the project together as a project archive file. You can extract the saved file in GM Programmer on another PC and edit the project file.

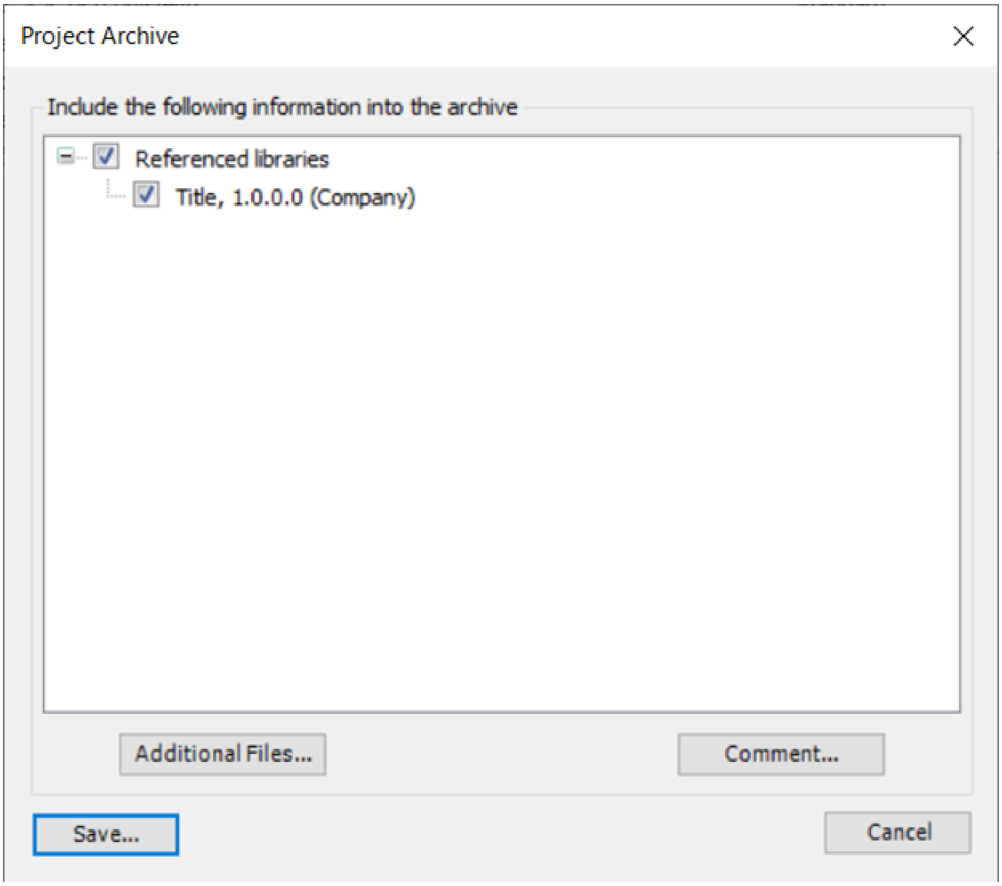
Project archive files have the extension “.projectarchive.”

4.16.1 Saving the project archive

1 2 Procedure

1. Select File ➔ Project Archive ➔ Save Archive from the menu bar.
The “Project Archive” dialog box appears.

2. Specify a file to save to the project archive.
Select the user libraries to add to the project archive, specify the files to add, and provide any comments that will be displayed during decompression.



Project Archive

Include the following information into the archive

☑ Referenced libraries

☑ Title, 1.0.0.0 (Company)

3. Click the [Save] button.

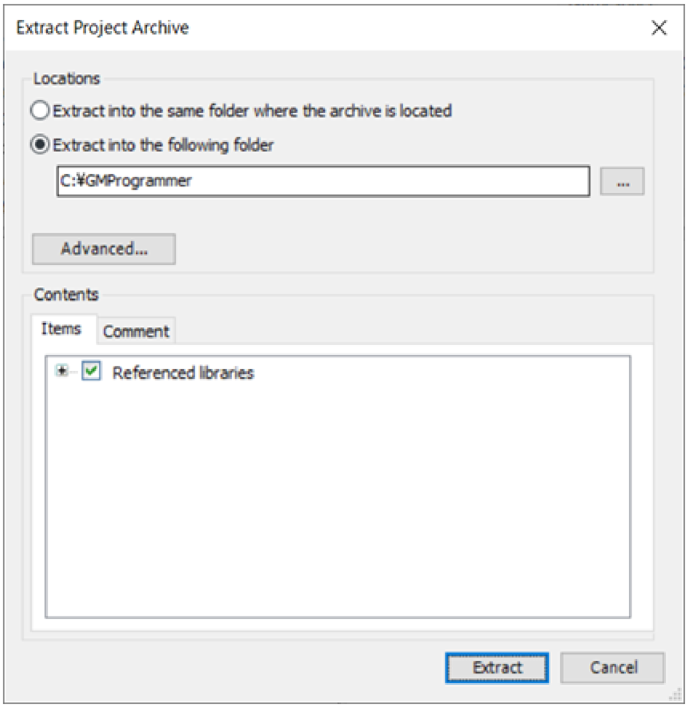
The “Save Project Archive” dialog box appears.

Specify a file name and save location, and then save.

4.16.2 Unzip Project Archive

1 2 Procedure

1. Select File ➔ Project Archive ➔ Extract Archive from the menu bar.
   The “Extract Archive” dialog box appears.
2. Select the project archive file to unzip.
   Once selected, the “Extract Project Archive” dialog box is displayed.
   Specify the location to extract the project archive, select the user libraries to extract, and review the comments.
   Click the [Advanced] button to display the “Advanced” dialog box where you can specify where to extract the added files.



Extract Project Archive

Locations

○ Extract into the same folder where the archive is located

● Extract into the following folder

C:¥GMProgrammer

3. Click the [Extract] button.
   The project files are extracted to the specified location.

5 Project Creation

5.1 Project Creation Flow

To create a program, you must first create a project.

This section explains operations for projects, operations for adding objects to projects, and other related operations.

First, this section explains the flow of project creation.

1. Setting up the GM1 controller

Set up parameters for the GM1 controller.
Refer to "5.2 Setting up the GM1 Controller".

↓

2. Setting up motion control

- Add device objects for servo amplifiers to a project and set them up.
- Add device objects for free encoders and virtual drives to a project and set them up.

Refer to "5.3 Setting up Motion Control".

↓

3. Setting up unit control

- Set up general-purpose I/O, PWM output, and high-speed counter for the GM1 controller.
- Add device objects for expansion units to a project and set them up.

Refer to "5.4 Setting up Unit Control".

↓

4. Setting up the communication function

- Add an object of the protocol to be used for the LAN port to a project and set it up.
- Add an object of the protocol to be used for the COM port to a project and set it up.

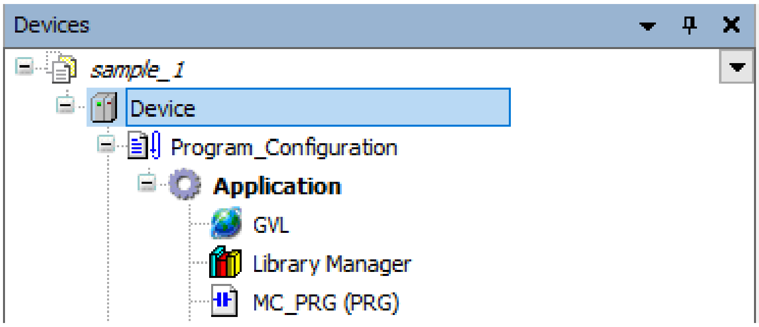
Refer to "5.5 Setting up the Communication Function".

5.2 Setting up the GM1 Controller

This section explains how to set up parameters for the GM1 controller.

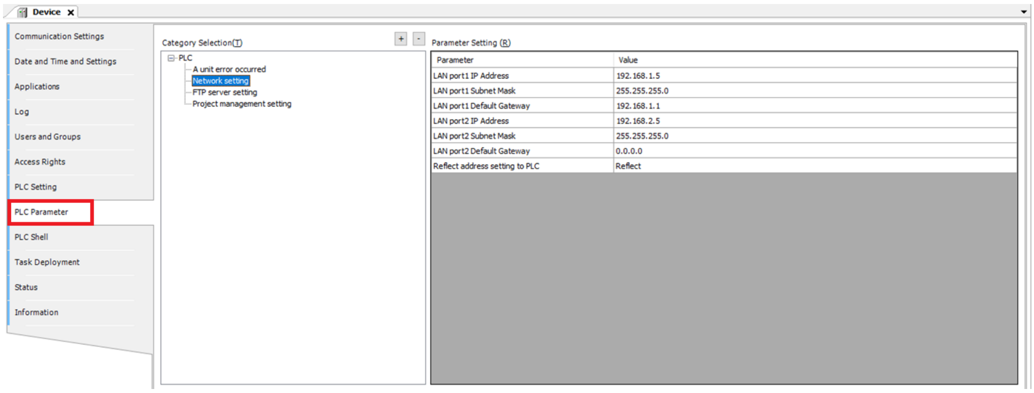
1 2 Procedure

1. Double-click the [Device] object in the navigator pane.In the Device pane, the Communication Settings sub-pane will be displayed.



2. Click the "PLC parameters" tab.

   The parameter settings sub-pane will be displayed.



Parameter	Value
LAN port1 IP Address	192.168.1.5
LAN port1 Subnet Mask	255.255.255.0
LAN port1 Default Gateway	192.168.1.1
LAN port2 IP Address	192.168.2.5
LAN port2 Subnet Mask	255.255.255.0
LAN port2 Default Gateway	0.0.0.0
Reflect address setting to PLC	Reflect

3. Change the values of the parameters that you want to update.

   A unit error occurred ：Select whether to stop or continue operation when an error occurs(For details, refer to "13.6.10 Unit Error" of high-speed counter function.)

   Network setting ： IP address, subunet mask, and defalt gateway of LAN port 1 and LAN port2, Reflect address setting to PLC
   FTP server setting ：Set parameters related to FTP server (For details, refer to "14.7.3 List of Setting Items" .)

   Project management setting ：Set parameters related to project management (For details, refer to "15.7 Project Management Function")

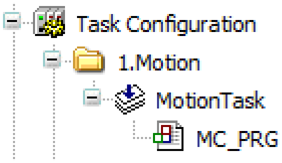
5.3 Setting up Motion Control

5.3.1 Setting up the EtherCAT master

Set a control cycle for the GM1 Controller.

1 2 Procedure

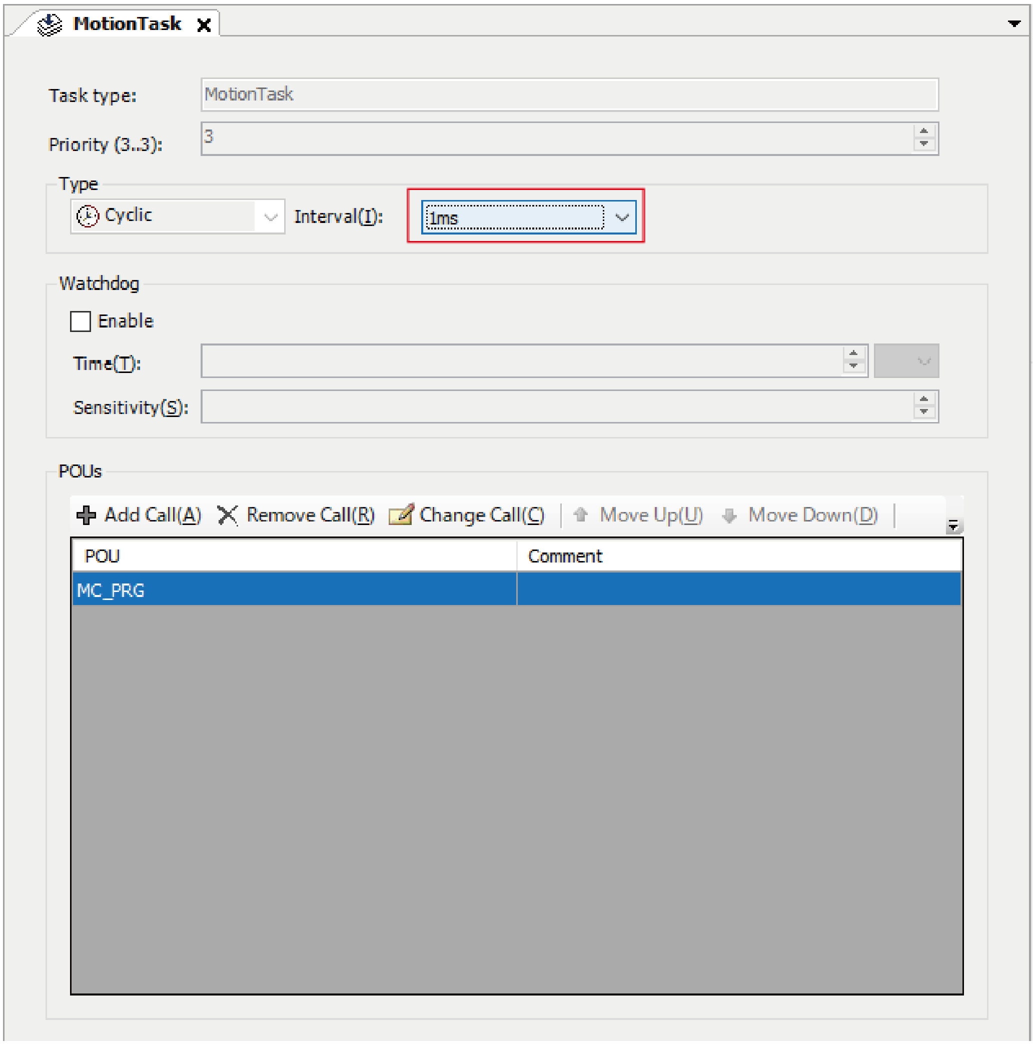
1. Double-click the **MotionTask** object in the navigator pane.



Task Configuration
1.Motion
MotionTask
MC_PRG

The "MotionTask" editor will open in the main pane.

2. Set a control cycle.



Task type: MotionTask

Priority (3..3): 3

Type

Cyclic Interval(I): 1ms

Watchdog

□ Enable

Time(T):

Sensitivity(S):

POUs

POU	Comment
MC_PRG	

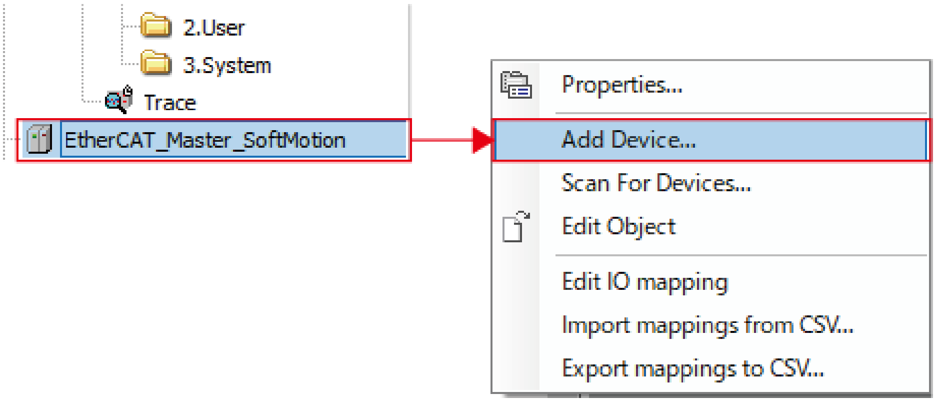
5.3.2 Adding and Setting up Servo Amplifiers

This section explains how to add device objects for servo amplifiers to a project and set them up.

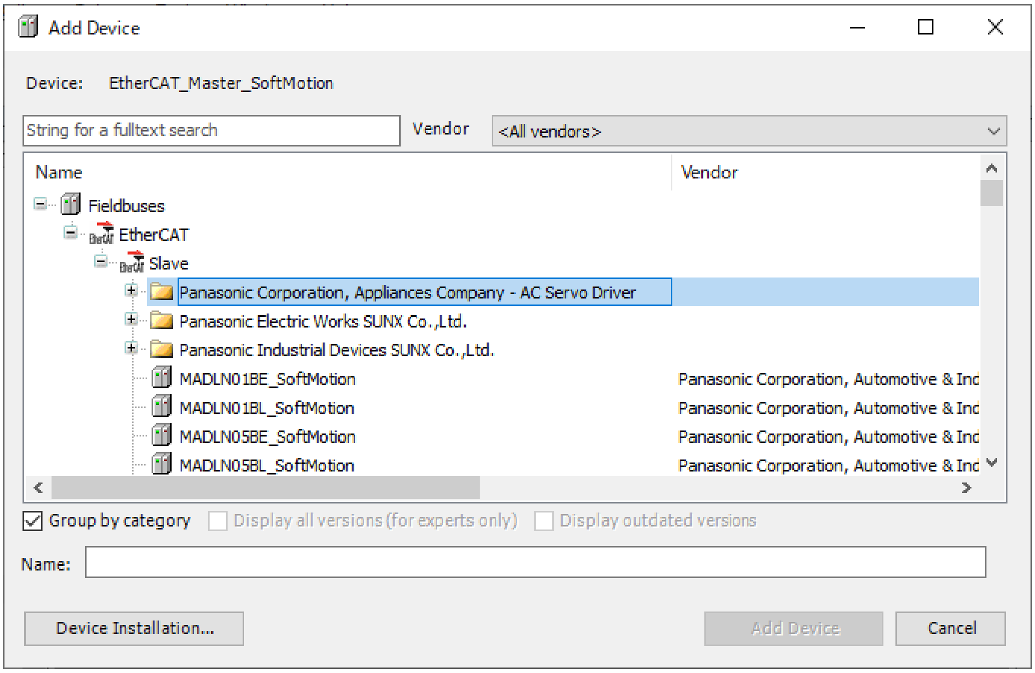
The description below explains how to add device objects for A5B servo amplifiers to a project and how to set them up.

Procedure

1. Right-click the**EtherCAT_Master_SoftMotion**object in the navigator pane and then select**Device addition**from the context-sensitive menu that is displayed.



The"Add Device"dialog box will be displayed.



Add Device

Device: EtherCAT_Master_SoftMotion

String for a fulltext search Vendor <All vendors>

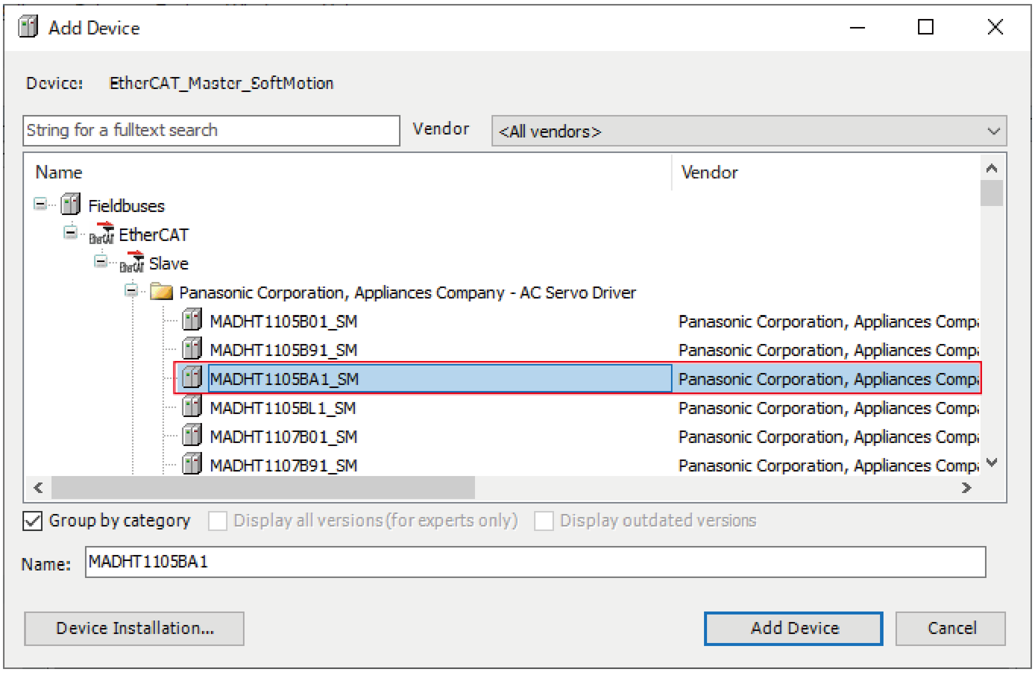
Name	Vendor
Fieldbuses	
EtherCAT	
Slave	
Panasonic Corporation, Appliances Company - AC Servo Driver	
Panasonic Electric Works SUNX Co.,Ltd.	
Panasonic Industrial Devices SUNX Co.,Ltd.	
MADLN01BE_SoftMotion	Panasonic Corporation, Automotive & Inc
MADLN01BL_SoftMotion	Panasonic Corporation, Automotive & Inc
MADLN05BE_SoftMotion	Panasonic Corporation, Automotive & Inc
MADLN05BL_SoftMotion	Panasonic Corporation, Automotive & Inc

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name:

2. Select a device object for the servo amplifier.

The selected device object of the servo amplifier will be added.



Add Device

Device: EtherCAT_Master_SoftMotion

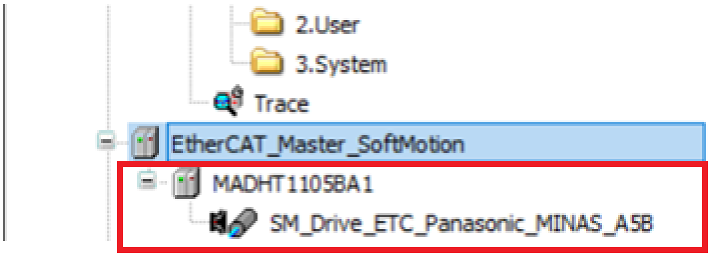
String for a fulltext search   Vendor <All vendors>

Name	Vendor
Fieldbuses	
EtherCAT	
Slave	
Panasonic Corporation, Appliances Company - AC Servo Driver	
MADHT1105B01_SM	Panasonic Corporation, Appliances Comp
MADHT1105B91_SM	Panasonic Corporation, Appliances Comp
MADHT1105BA1_SM	Panasonic Corporation, Appliances Comp
MADHT1105BL1_SM	Panasonic Corporation, Appliances Comp
MADHT1107B01_SM	Panasonic Corporation, Appliances Comp
MADHT1107B91_SM	Panasonic Corporation, Appliances Comp

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

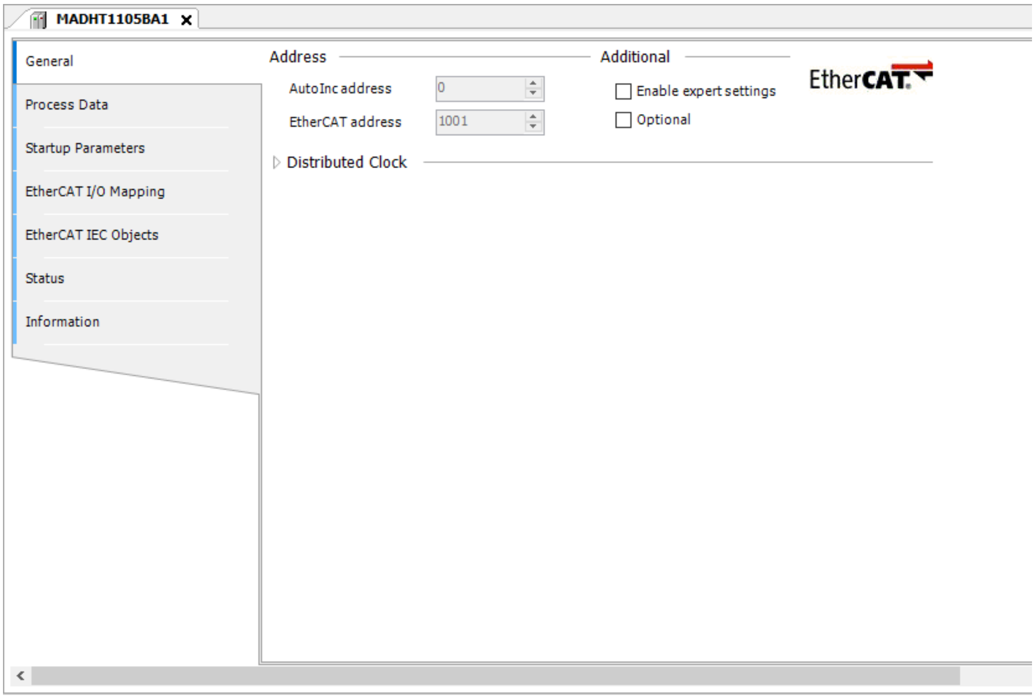
Name: MADHT1105BA1

3. Click the[Add Device]button.
The selected device object of the servo amplifier will be added to the navigator pane.



Click the[Cancel]button to close the “Select Device” dialog box.

4. Double-click the added object.
The setting pane will be displayed in the main pane.



Address

AutoInc address: 0

EtherCAT address: 1001

Additional

□ Enable expert settings

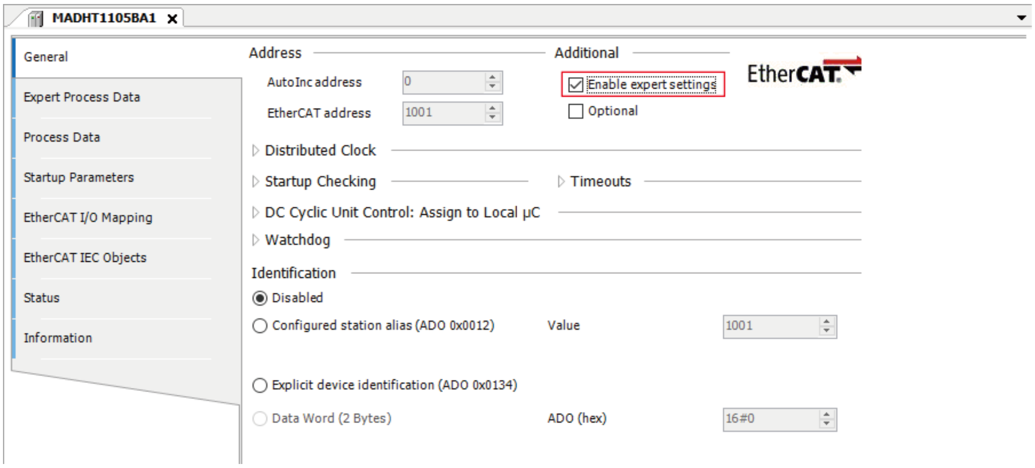
□ Optional

▷ Distributed Clock

Info.

- To remove a device object that has been added, select the device object in the navigation pane and press the<Delete>key.

5. Enable the “Enable expert setting” item.



Address

AutoInc address: 0

EtherCAT address: 1001

Additional

☑ Enable expert settings

□ Optional

▷ Distributed Clock

▷ Startup Checking

▷ Timeouts

▷ DC Cyclic Unit Control: Assign to Local µC

▷ Watchdog

Identification

● Disabled

○ Configured station alias (ADO 0x0012) Value: 1001

○ Explicit device identification (ADO 0x0134)

○ Data Word (2 Bytes) ADO (hex): 16#0

6. Set “Station alias”.

Depending on the setting of Pr7.41 as the initial setup for the servo amplifier, the method for setting “Station alias” is different.

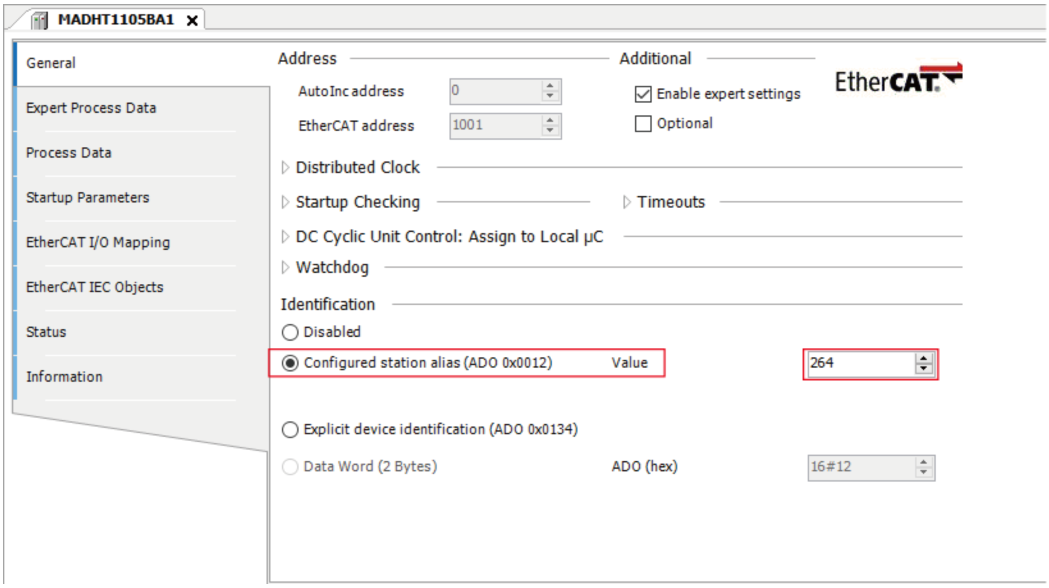
- When Pr.7.41 is set to 0

Select the "Configured station alias” option and enter the setting of the rotary switch on the front panel of the servo amplifier and the setting of Pr7.40 into the input field.

Example) Set value of the rotary switch on the front panel of the servo amplifier: 8, set value of Pr7.40: 1

The higher 8 bits are 1 and lower 8 bits are 8. Therefore, input 264 in the input field.

Station alias	
Higher 8 bits	**Lower 8 bits**
Set value of 3740h	Set value of the rotary switch



MADHT1105BA1

General / Expert Process Data / Process Data / Startup Parameters / EtherCAT I/O Mapping / EtherCAT IEC Objects / Status / Information

Address

AutoInc address: 0

EtherCAT address: 1001

Additional

☑ Enable expert settings

□ Optional

▷ Distributed Clock

▷ Startup Checking

▷ Timeouts

▷ DC Cyclic Unit Control: Assign to Local µC

▷ Watchdog

Identification

○ Disabled

● Configured station alias (ADO 0x0012) Value: 264

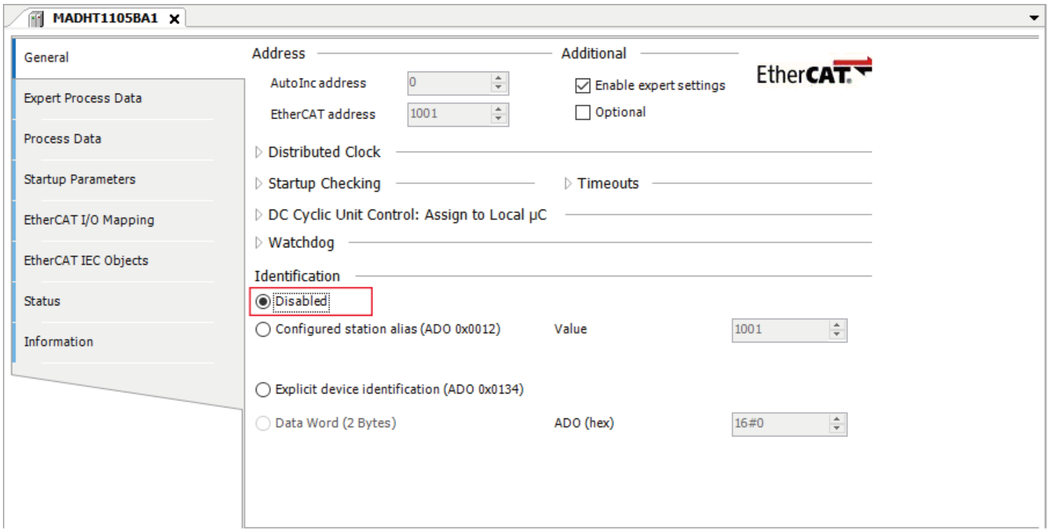
○ Explicit device identification (ADO 0x0134)

○ Data Word (2 Bytes) ADO (hex): 16#12

- When Pr.7.41 is set to 1

Select the "Disable” option.

The numerical value displayed in "EtherCAT address” is set as the station alias value.



MADHT1105BA1

General / Expert Process Data / Process Data / Startup Parameters / EtherCAT I/O Mapping / EtherCAT IEC Objects / Status / Information

Address

AutoInc address: 0

EtherCAT address: 1001

Additional

☑ Enable expert settings

□ Optional

▷ Distributed Clock

▷ Startup Checking

▷ Timeouts

▷ DC Cyclic Unit Control: Assign to Local µC

▷ Watchdog

Identification

● Disabled

○ Configured station alias (ADO 0x0012) Value: 1001

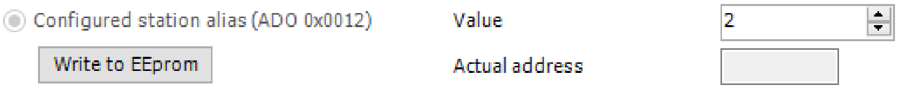
○ Explicit device identification (ADO 0x0134)

○ Data Word (2 Bytes) ADO (hex): 16#0

Info.

- You can also set any desired value after logging in by selecting the configured station alias value.

  Enter any desired value in the input field and click the [Write to EEPROM] button.

  ○ Configured station alias (ADO 0x0012)

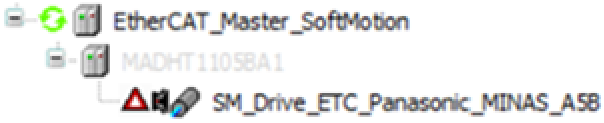
  Write to EEprom

  Value: 2

  Actual address:

  When you log in again after restarting the servo amplifier, the value entered in the input field is set as the station alias value.

  Select the configured station alias value and log in for the first time, and the device name of the servo amplifier in the navigation pane will be grayed out due to a mismatch of the station alias value.

  EtherCAT_Master_SoftMotion

  MADHT1105BA1

  SM_Drive_ETC_Panasonic_MINAS_A5B

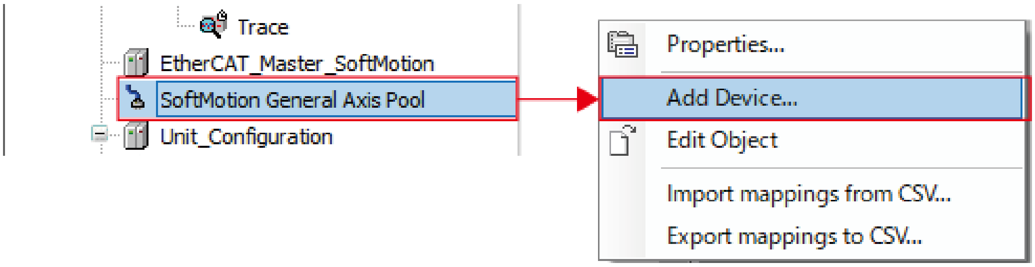
5.3.3 Adding and Setting up Free Encoder and Virtual Drive

This section explains how to add device objects for free encoders and virtual drives to a project and set them up.

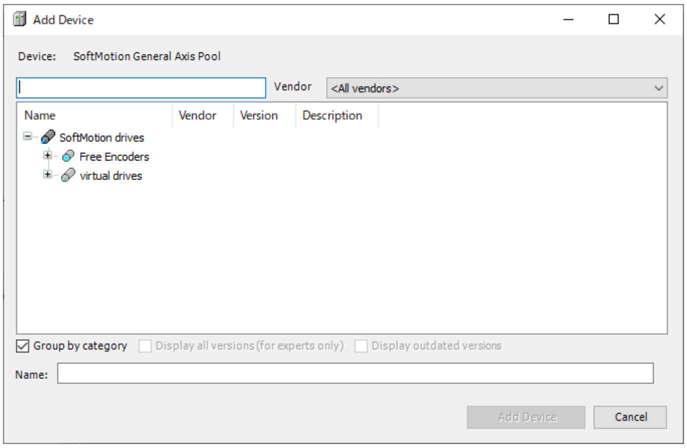
For example, use the following procedure to add a device object for a virtual drive to a project and set it up.

1 2 Procedure

1. Right-click the [SoftMotion General Axis Pool] object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.


Trace
EtherCAT_Master_SoftMotion
SoftMotion General Axis Pool
Unit_Configuration
Properties...
Add Device...
Edit Object
Import mappings from CSV...
Export mappings to CSV...

The "Device Management" dialog box will be displayed.



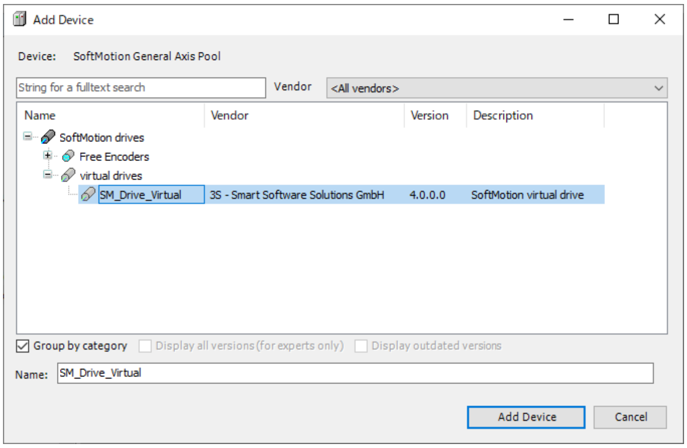
Device: SoftMotion General Axis Pool

Vendor <All vendors>

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name:

2. Select a device object for the virtual drive.



Add Device

Device: SoftMotion General Axis Pool

String for a fulltext search

Vendor <All vendors>

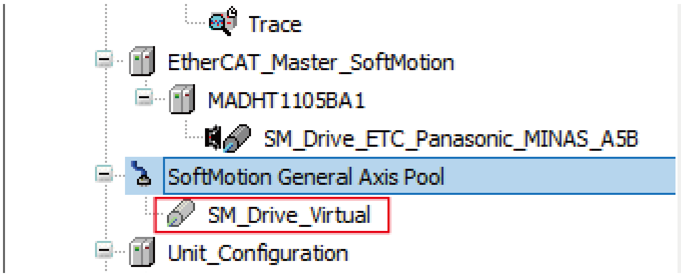
Name	Vendor	Version	Description
SoftMotion drives			
Free Encoders			
virtual drives			
SM_Drive_Virtual	3S - Smart Software Solutions GmbH	4.0.0.0	SoftMotion virtual drive

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name: SM_Drive_Virtual

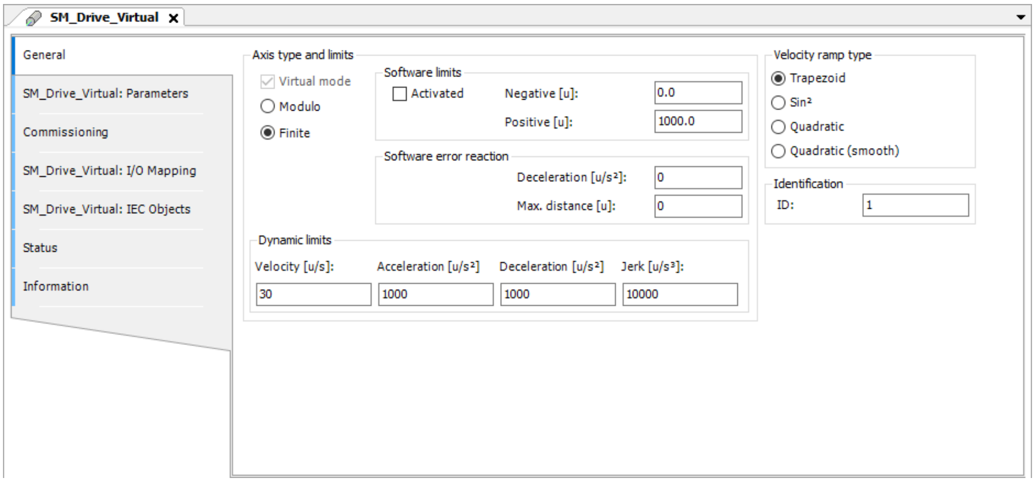
3. Click the [Add Device] button.

The selected device object of the virtual drive will be added to the navigator pane.


Trace
EtherCAT_Master_SoftMotion
MADHT1105BA1
SM_Drive_ETC_Panasonic_MINAS_A5B
SoftMotion General Axis Pool
SM_Drive_Virtual
Unit_Configuration

4. Double-click the added object.

The setting pane will be displayed in the main pane. Specify settings related to the virtual drive.



SM_Drive_Virtual

General
SM_Drive_Virtual: Parameters
Commissioning
SM_Drive_Virtual: I/O Mapping
SM_Drive_Virtual: IEC Objects
Status
Information

Axis type and limits

☑ Virtual mode
○ Modulo
● Finite

Software limits

□ Activated Negative [u]: 0.0 Positive [u]: 1000.0

Software error reaction

Deceleration [u/s²]: 0
Max. distance [u]: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
30	1000	1000	10000

Velocity ramp type

● Trapezoid
○ Sin²
○ Quadratic
○ Quadratic (smooth)

Identification

ID: 1

Info.

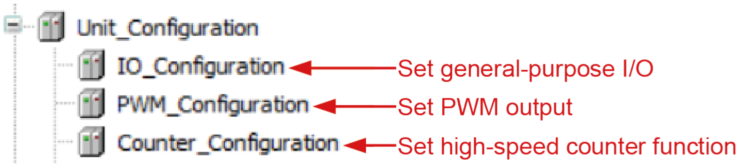
- To remove a device object that has been added, select the device object in the navigation pane and press the "Delete" key.

5.4 Setting up Unit Control

5.4.1 Setting up the GM1 units

This section explains how to set up general-purpose I/O, PWM output, and high-speed counter for the GM1 Controller.

The object of each unit is shown below.



Set general-purpose I/O

Set PWM output

Set high-speed counter function

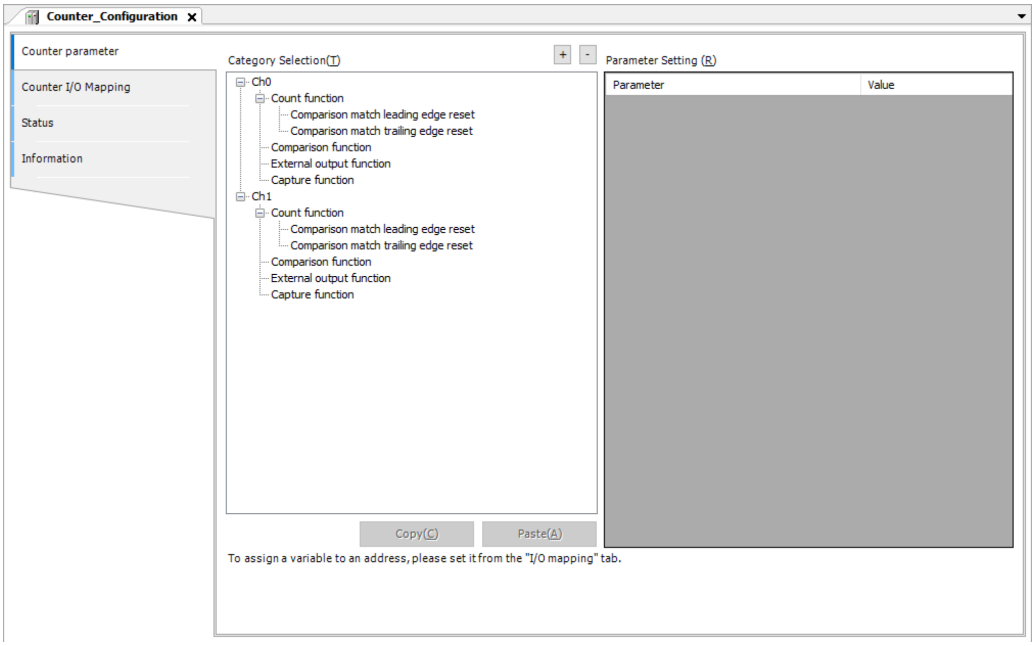
Info.

- For details on general-purpose I/O, refer to "13.4 General-purpose I/O".
- For details on PMW output, refer to "13.5 PWM Output".
- For details on the high-speed counter function, refer to "13.6 High-speed Counter Function".

For example, use the following procedure to set up a high-speed counter.

Procedure

1. In the navigator pane, double-click the [Counter_Configuration] object.
   The high-speed counter setting window will be displayed.



Ch0
- Count function
  - Comparison match leading edge reset
  - Comparison match trailing edge reset
- Comparison function
- External output function
- Capture function

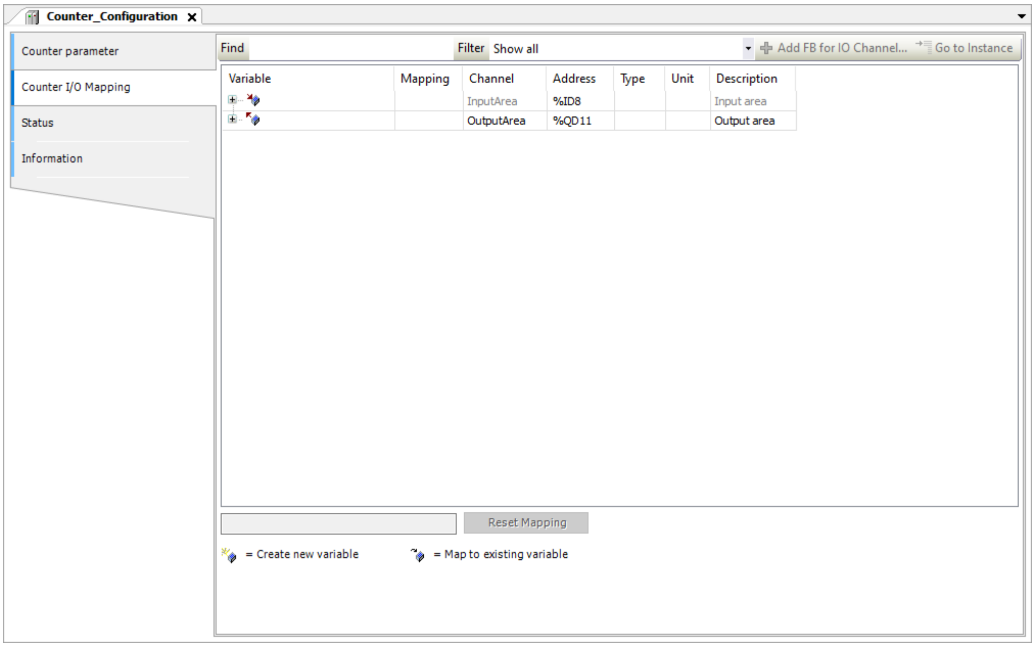
Ch1
- Count function
  - Comparison match leading edge reset
  - Comparison match trailing edge reset
- Comparison function
- External output function
- Capture function

Parameter	Value

To assign a variable to an address, please set it from the "I/O mapping" tab.

2. Set up parameters for the high-speed counter.

Select the category of a channel to be set up in the "Category Selection" section and enter values in the "Parameter Settings" section.

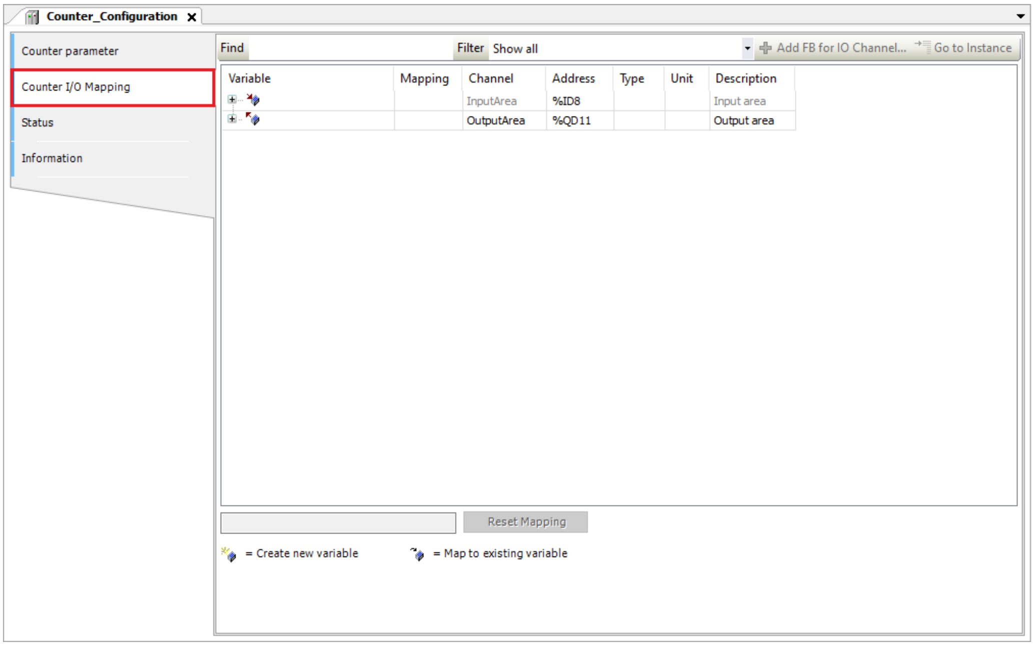


Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		OutputArea	%QD11			Output area

3. Select the "Counter I/O Mapping" tab and set the correspondence (mapping) between the channel and variable in the mapping setting pane.

Click the "Variable" column corresponding to the channel to be used by the program and enter a variable name.

Clicking the mark in the "Mapping" column allows you to change the type of mapping.



Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		OutputArea	%QD11			Output area

Info.

- You can copy the parameter set in a channel. To do so, select a channel (CH0 or CH1) in the "Category Selection" column and click the [Copy] button. Next, select another channel and click the [Paste] button.

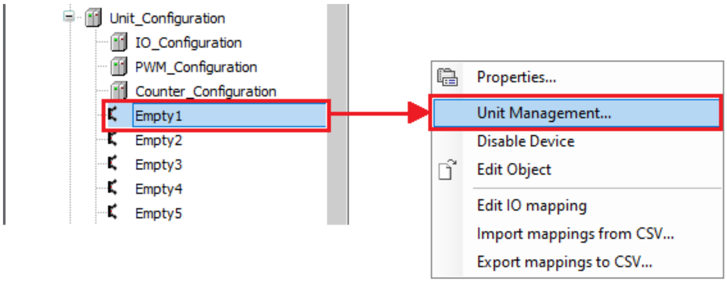
5.4.2 Adding Expansion Units

This section explains how to add device objects for expansion units to a project and set them up.

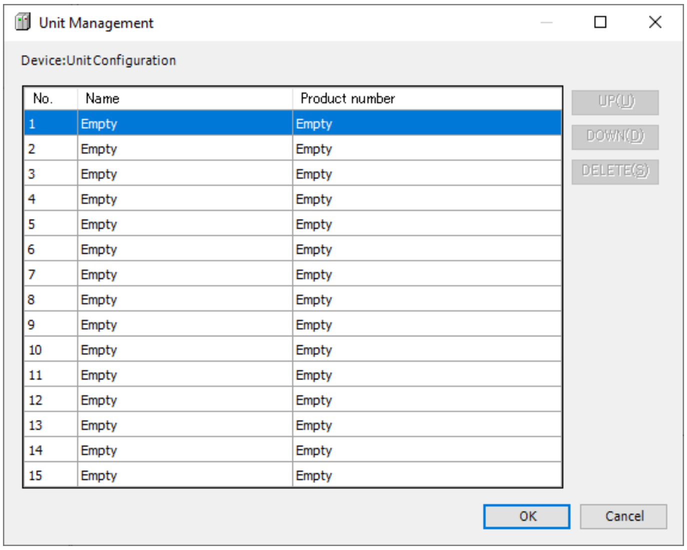
For example, use the following procedure to add a digital input unit (product number: AGM1X64D2) to Empty1.

1 2 Procedure

1. Right-click the [Empty1] object in the navigator pane and then select "Unit Management" from the context-sensitive menu that is displayed.



The "Unit Management" dialog box will be displayed.

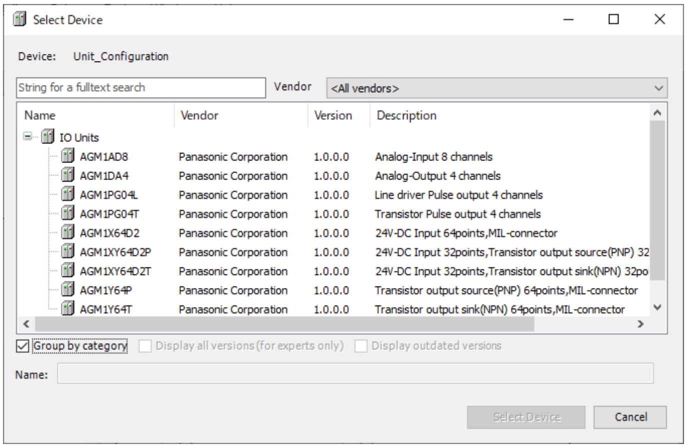


Device:UnitConfiguration

No.	Name	Product number
1	Empty	Empty
2	Empty	Empty
3	Empty	Empty
4	Empty	Empty
5	Empty	Empty
6	Empty	Empty
7	Empty	Empty
8	Empty	Empty
9	Empty	Empty
10	Empty	Empty
11	Empty	Empty
12	Empty	Empty
13	Empty	Empty
14	Empty	Empty
15	Empty	Empty

2. Double-click the first row in the "Device: Unit_Configuration" table.

The "Select Device" dialog box will be displayed.



Select Device

Device: Unit_Configuration

String for a fulltext search

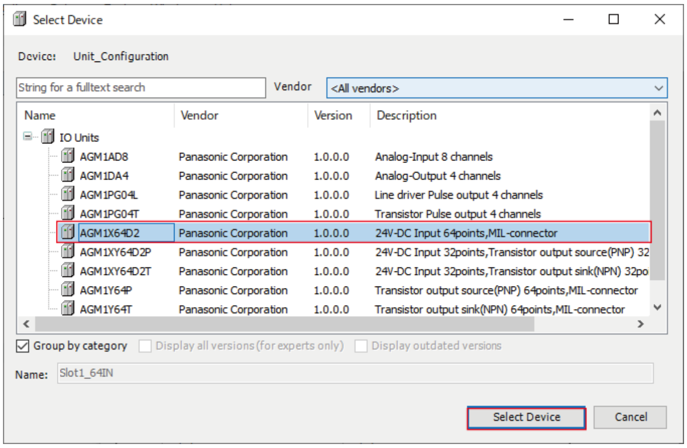
Vendor <All vendors>

Name	Vendor	Version	Description
IO Units			
AGM1AD8	Panasonic Corporation	1.0.0.0	Analog-Input 8 channels
AGM1DA4	Panasonic Corporation	1.0.0.0	Analog-Output 4 channels
AGM1PG04L	Panasonic Corporation	1.0.0.0	Line driver Pulse output 4 channels
AGM1PG04T	Panasonic Corporation	1.0.0.0	Transistor Pulse output 4 channels
AGM1X64D2	Panasonic Corporation	1.0.0.0	24V-DC Input 64points,MIL-connector
AGM1XY64D2P	Panasonic Corporation	1.0.0.0	24V-DC Input 32points,Transistor output source(PNP) 32
AGM1XY64D2T	Panasonic Corporation	1.0.0.0	24V-DC Input 32points,Transistor output sink(NPN) 32po
AGM1Y64P	Panasonic Corporation	1.0.0.0	Transistor output source(PNP) 64points,MIL-connector
AGM1Y64T	Panasonic Corporation	1.0.0.0	Transistor output sink(NPN) 64points,MIL-connector

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name:

3. Select a device object for the expansion unit to be added.



Select Device

Device: Unit_Configuration

String for a fulltext search

Vendor <All vendors>

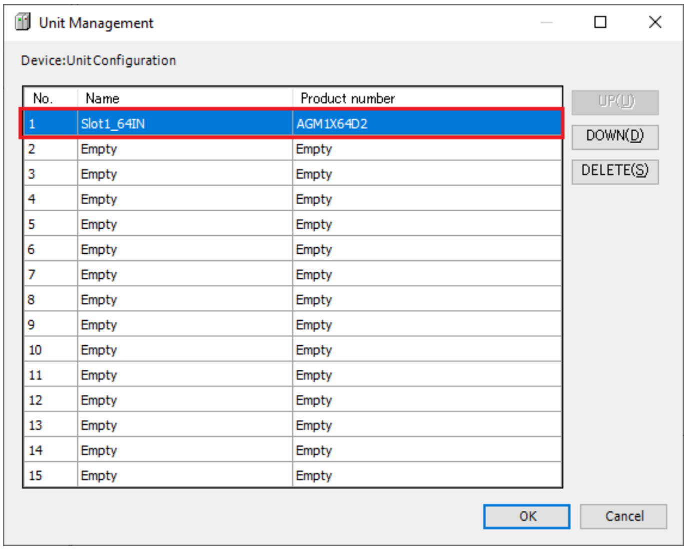
Name	Vendor	Version	Description
IO Units			
AGM1AD8	Panasonic Corporation	1.0.0.0	Analog-Input 8 channels
AGM1DA4	Panasonic Corporation	1.0.0.0	Analog-Output 4 channels
AGM1PG04L	Panasonic Corporation	1.0.0.0	Line driver Pulse output 4 channels
AGM1PG04T	Panasonic Corporation	1.0.0.0	Transistor Pulse output 4 channels
AGM1X64D2	Panasonic Corporation	1.0.0.0	24V-DC Input 64points,MIL-connector
AGM1XY64D2P	Panasonic Corporation	1.0.0.0	24V-DC Input 32points,Transistor output source(PNP) 32
AGM1XY64D2T	Panasonic Corporation	1.0.0.0	24V-DC Input 32points,Transistor output sink(NPN) 32po
AGM1Y64P	Panasonic Corporation	1.0.0.0	Transistor output source(PNP) 64points,MIL-connector
AGM1Y64T	Panasonic Corporation	1.0.0.0	Transistor output sink(NPN) 64points,MIL-connector

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name: Slot1_64IN

4. Click the [Select Device] button.

The selected device object of the expansion unit will be added.



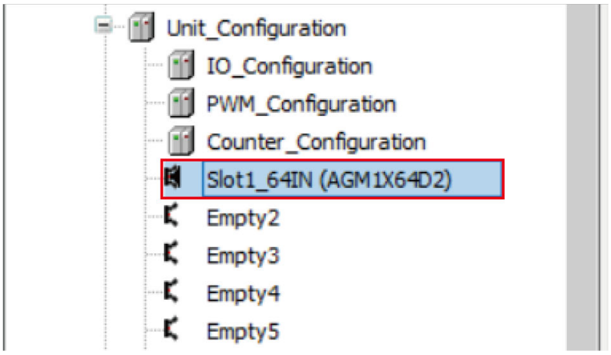
Unit Management

Device:UnitConfiguration

No.	Name	Product number
1	Slot1_64IN	AGM1X64D2
2	Empty	Empty
3	Empty	Empty
4	Empty	Empty
5	Empty	Empty
6	Empty	Empty
7	Empty	Empty
8	Empty	Empty
9	Empty	Empty
10	Empty	Empty
11	Empty	Empty
12	Empty	Empty
13	Empty	Empty
14	Empty	Empty
15	Empty	Empty

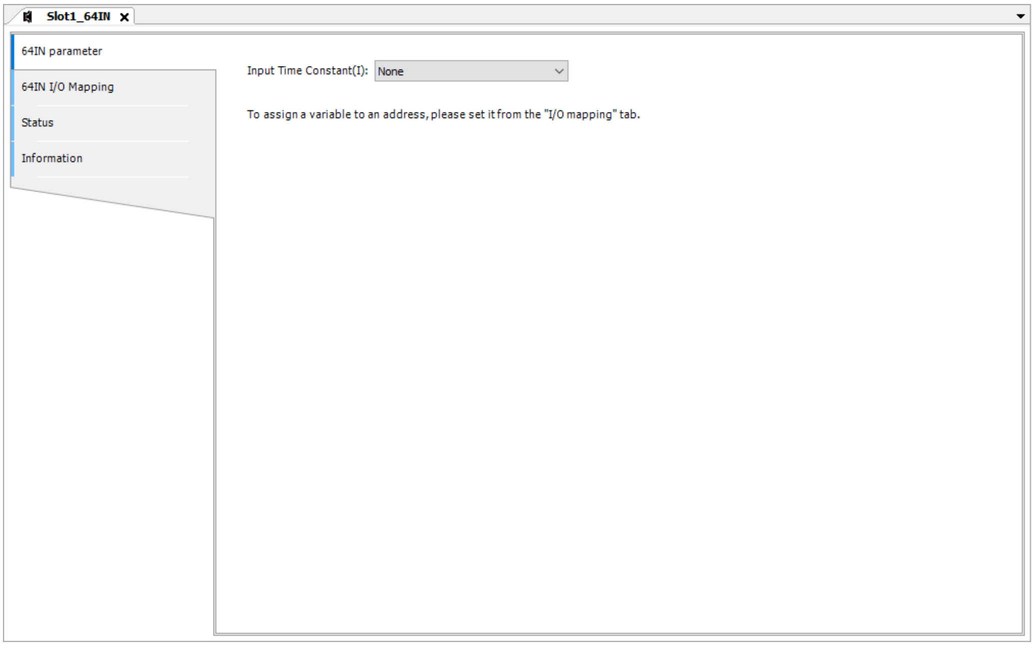
5. Click the [OK] button.

The selected device object of the expansion unit will be added to the navigator pane.



6. Double-click the added object.

The setting pane will be displayed in the main pane. Specify settings related to the expansion unit.



Input Time Constant(I): None

To assign a variable to an address, please set it from the "I/O mapping" tab.

Info.

- To remove the device object of an expansion unit that has been added, select the expansion unit to be removed in the "Unit Management" dialog box and press the "Delete" key or click the [Delete] button.

5.5 Setting up the Communication Function

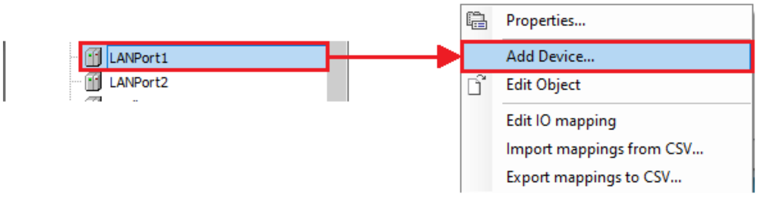
5.5.1 Adding a Protocol to Be Used for the LAN Port

This section explains how to add an object of the protocol to be used for the LAN port to a project and set it up.

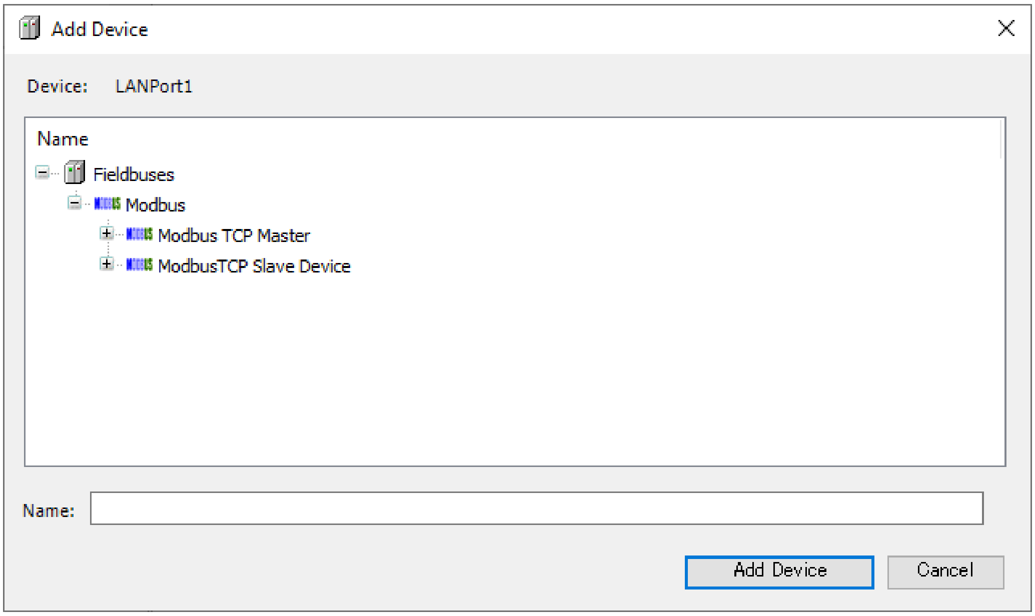
For example, use the following procedure to add an object of Modbus TCP to LANPort1 and set it up.

1 2 Procedure

1. Right-click the [LANPort1] object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.



Add Device

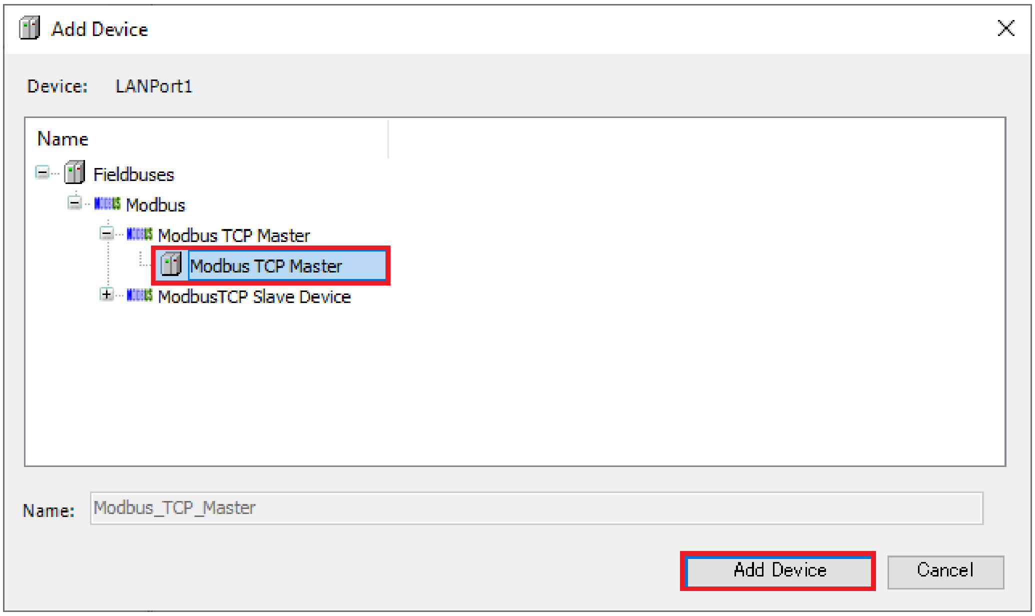
Device: LANPort1

Name

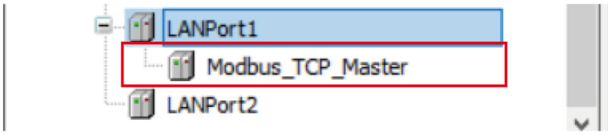
- Fieldbuses
  - Modbus
    - Modbus TCP Master
    - ModbusTCP Slave Device

Name:

2. Select device "Modbus TCP Master".

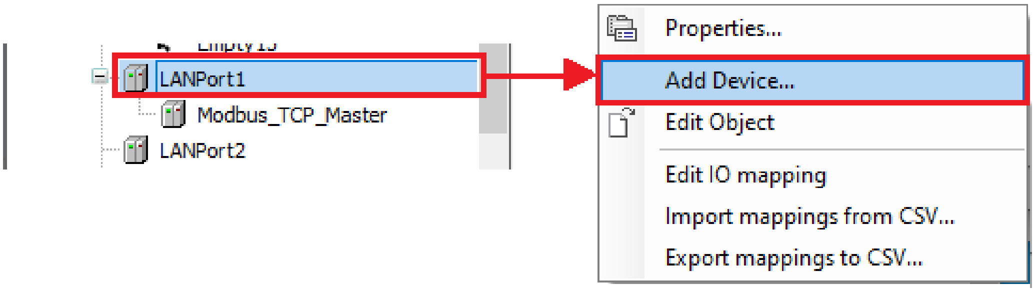


3. Click the [Add Device] button.
Object [Modbus_TCP_Master] will be added to the navigator pane.

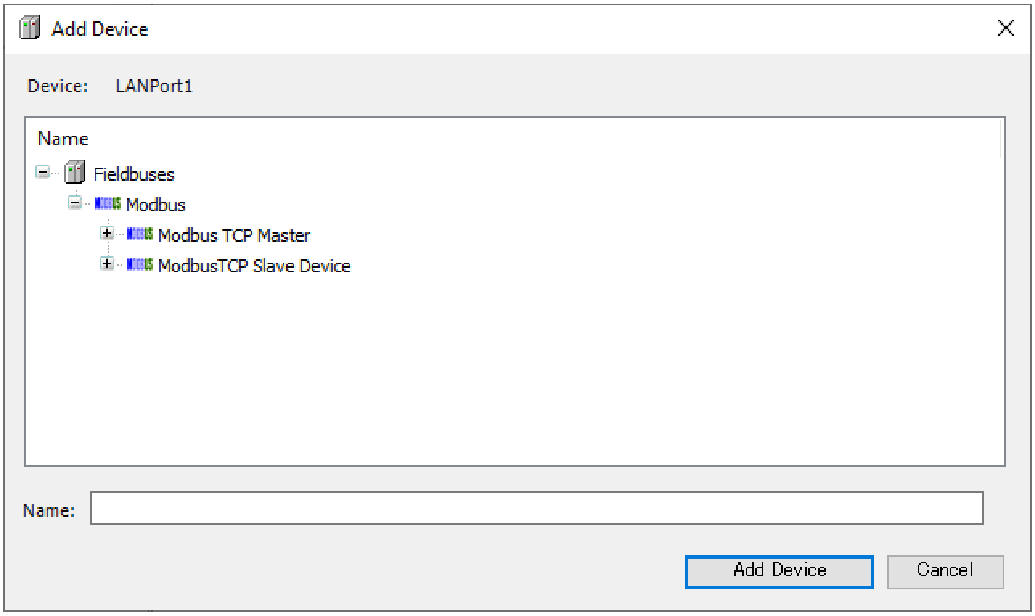


Then, add object [ModbusTCP_Slave_Device] below object [Modbus_TCP_Master].

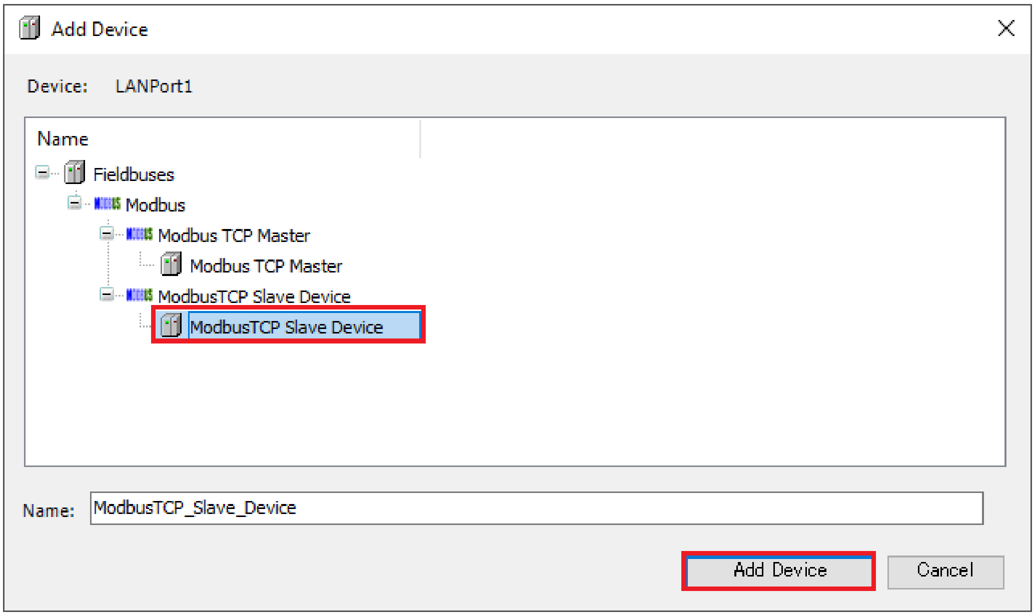
4. Right-click the [LANPort1] object and then select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.

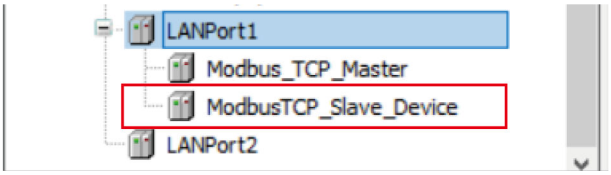


5. Select device "Modbus TCP Slave Device".



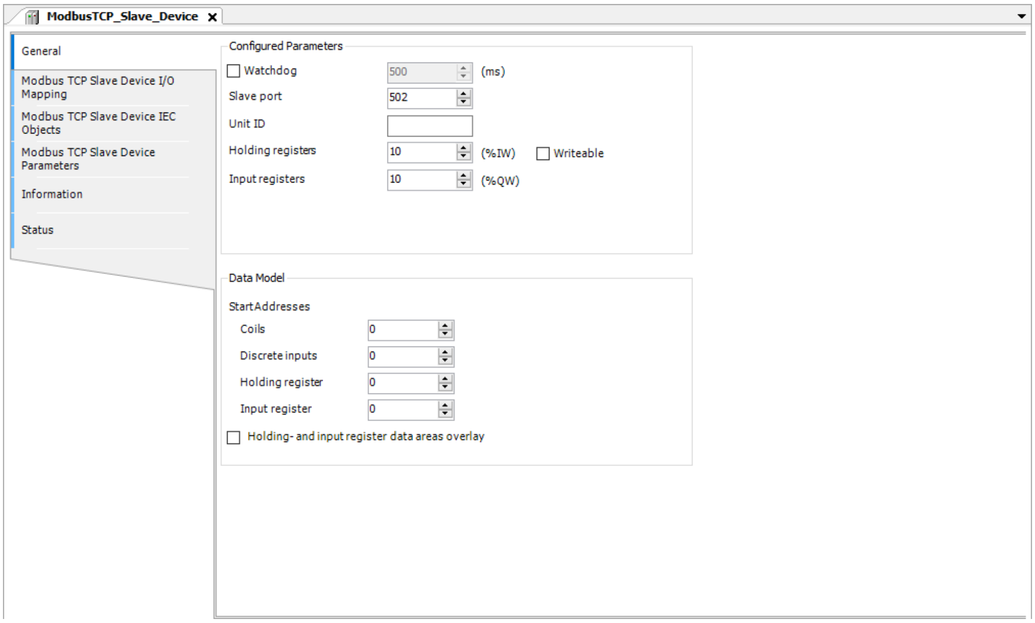
Name: ModbusTCP_Slave_Device

6. Click the [Add Device] button.
The [ModbusTCP_Slave_Device] object will be added.



7. Double-click the added object.

   The setting pane will be displayed in the main pane. Specify settings related to Modbus TCP.



ModbusTCP_Slave_Device
General
Modbus TCP Slave Device I/O Mapping
Modbus TCP Slave Device IEC Objects
Modbus TCP Slave Device Parameters
Information
Status

Configured Parameters

□ Watchdog 500 (ms)

Slave port 502

Unit ID

Holding registers 10 (%IW) □ Writeable

Input registers 10 (%QW)

Data Model

StartAddresses

Coils 0

Discrete inputs 0

Holding register 0

Input register 0

□ Holding- and input register data areas overlay

Info.

- To remove a device that has been added, select the device in the navigator pane and press the "Delete" key.

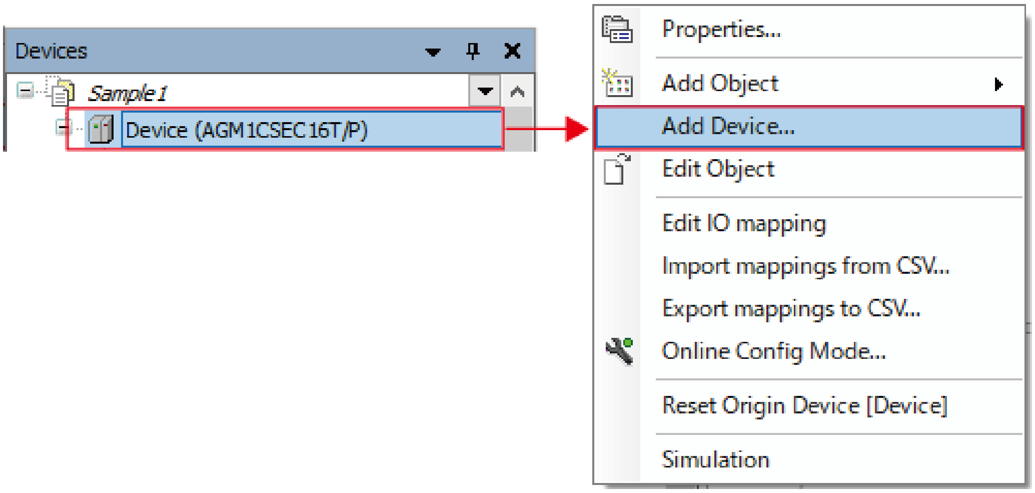
5.5.2 Adding a Protocol to Be Used for the COM Port

This section explains how to add an object of the protocol to be used for the COM port to a project and set it up.

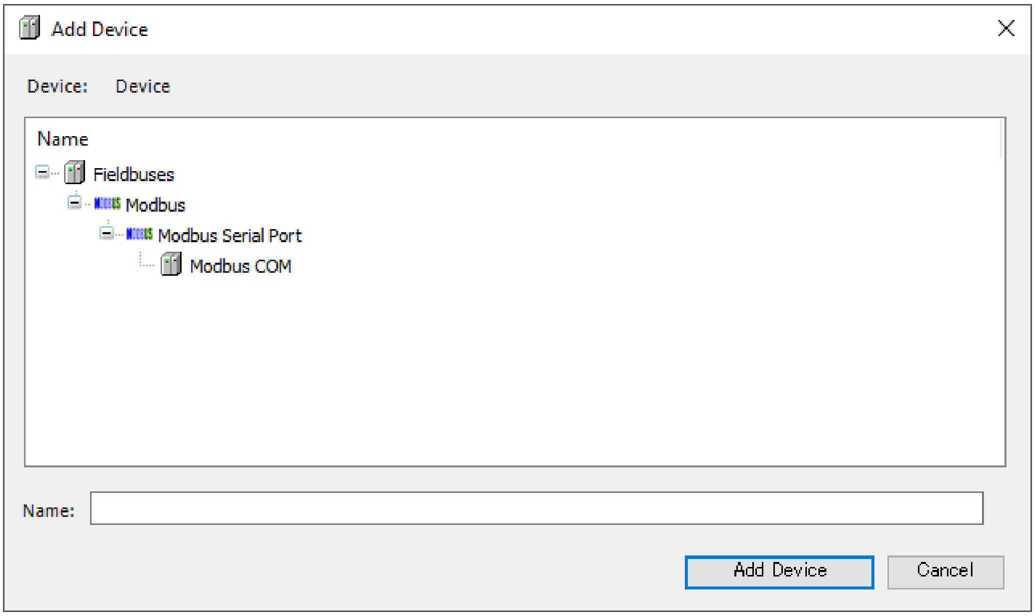
For example, use the following procedure to add an object of Modbus RTU to the COM port and set it up.

1 2 Procedure

1. Right-click the [Device] object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.



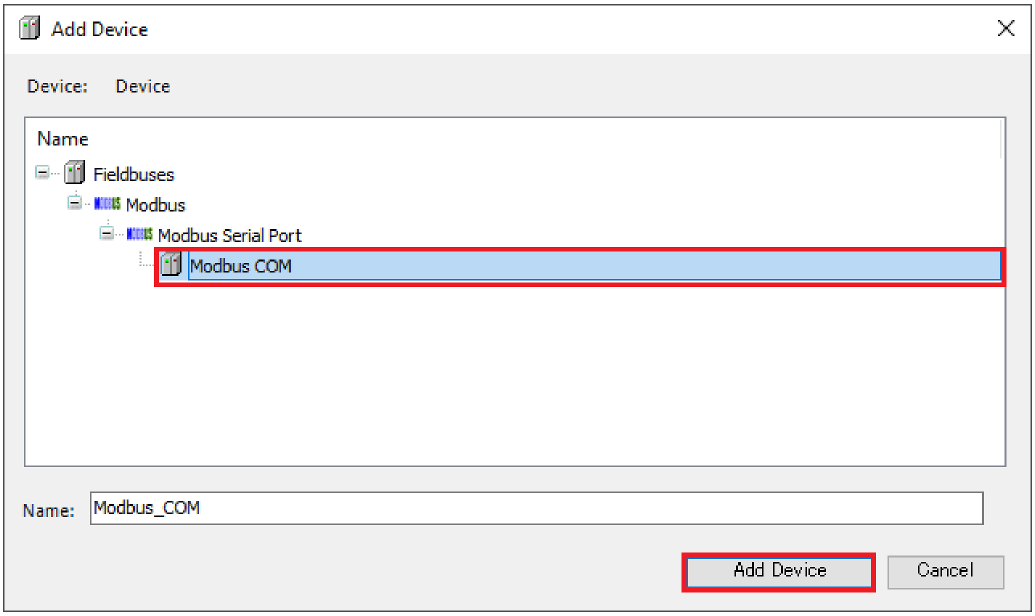
The "Add Device" dialog box will be displayed.



Device: Device

Name:

2. Select device "Modbus COM".

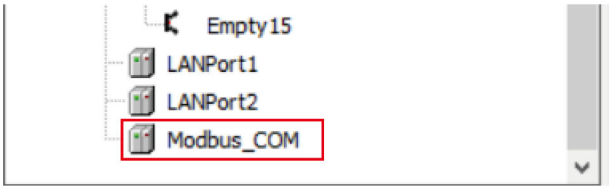


Add Device

Device: Device

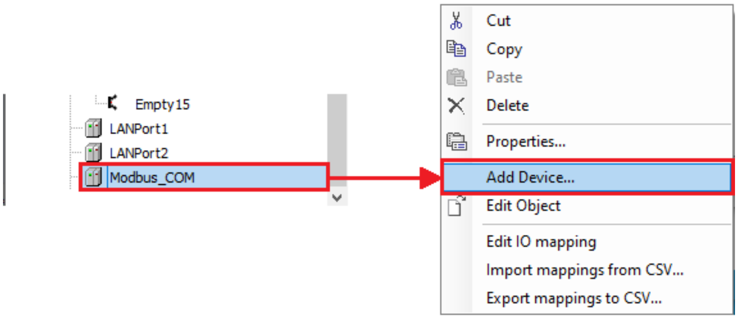
Name: Modbus_COM

3. Click the [Add Device] button.
Object [Modbus_COM] will be added to the navigator pane.

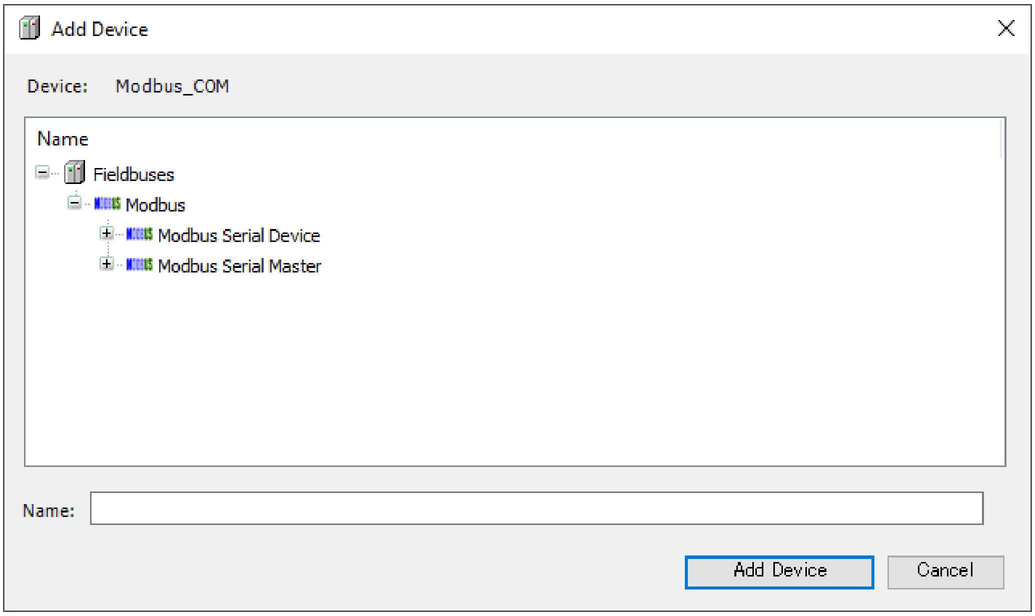


Then, add object [Modbus_COM_Master] below object [Modbus_COM].

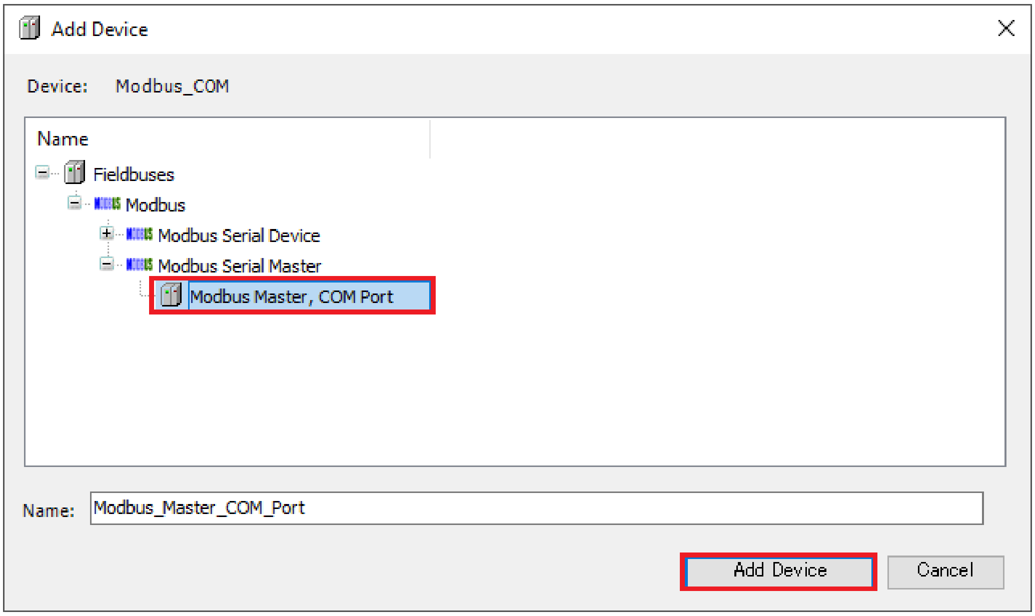
4. Right-click the [Modbus_COM] object in the navigation pane and then select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.

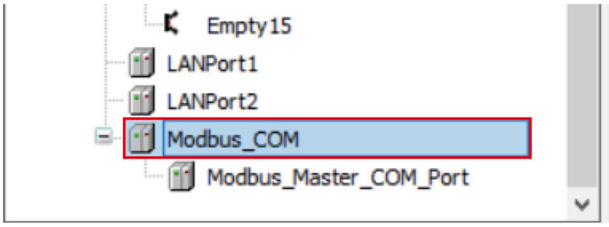


5. Select device "Modbus Master, COM Port".



Name: Modbus_Master_COM_Port

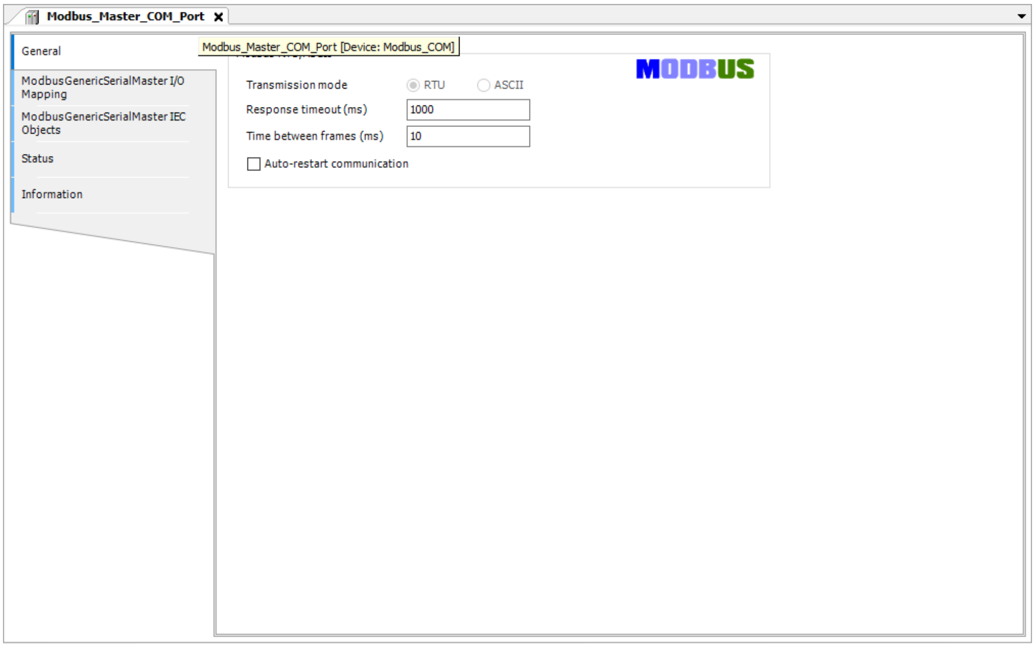
6. Click the [Add Device] button.
The [Modbus_Master_COM_Port] object will be added.



Empty15
LANPort1
LANPort2
Modbus_COM
Modbus_Master_COM_Port

7. Double-click the added [Modbus_Master_COM_Port] object.

The setting pane will be displayed in the main pane. Specify settings related to Modbus RTU.



Modbus_Master_COM_Port

General
ModbusGenericSerialMaster I/O Mapping
ModbusGenericSerialMaster IEC Objects
Status
Information
MODBUS

Modbus_Master_COM_Port [Device: Modbus_COM]

Transmission mode ● RTU ○ ASCII

Response timeout (ms) 1000

Time between frames (ms) 10

□ Auto-restart communication

(MEMO)

6 Program Creation

6.1 Flow of Program Creation

1. Creating a POU object

Create an object (POU object) for the program.

⬇

2. Entering program data (refer to "7.1 Programming in Ladder Diagram (LD)" through to "7.7 Variables")

Open the POU object. Enter program data, declare variables, and perform other necessary work.

⬇

3. Executing build (refer to "6.7 Build")

- Execute build and check the program.
- If there are any errors, return to "Step 2" and correct the program.

⬇

4. Registering for a task (refer to "6.6 Tasks")

Register the POU objects (to be executed on the GM1 controller) for a task.

Info.

- GM Programmer provides support functions that can be used to create programs efficiently. For details on support functions, refer to "6.8 Program Creation Support Functions".
- Programs consisting of functions and function blocks can be created. For details on functions and function blocks, refer to "6.5 Function and Function Block".

6.2 Program Creation Window

This section explains the window for creating programs in GM Programmer.

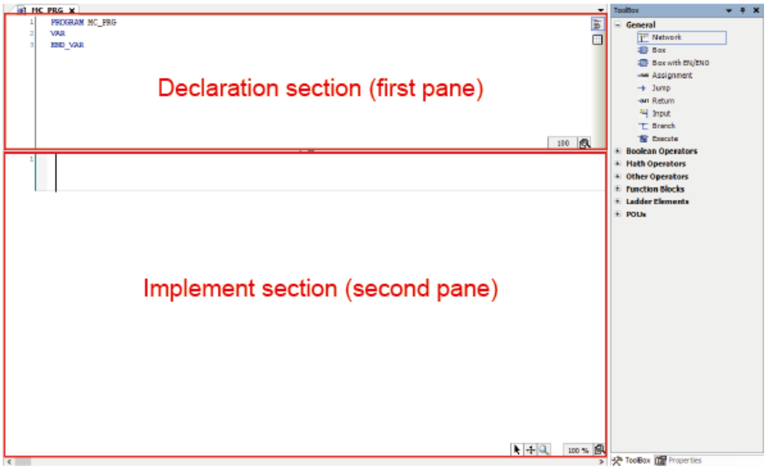
6.2.1 Main Pane

The upper section of the main pane is the declaration section where variables are declared.

The lower section of the main pane is the implementation section where program processes are coded. The declaration section and implementation section may be called the first and second panes, respectively.

The editing method in the implementation section differs according to the program. Refer to the section related to each program creation.

Example: Main pane for LD programs



```
PROGRAM MC_PRG
VAR
END_VAR
```

Declaration section (first pane)

Implement section (second pane)

Info.

- The selected pane can be switched between the declaration section (first pane) and the implementation section (second pane). To switch the selected pane, from the menu bar, select **Window**>**Next Pane** or "Previous Pane".
- You can also hide the declaration section (first pane) or the implementation section (second pane).

  To hide the declaration section, from the menu bar, select **Window**>**Switch First Pane**.

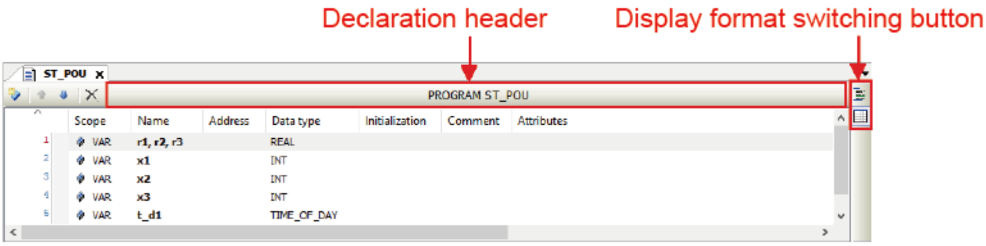
  To hide the implementation section, from the menu bar, select **Window**>**Switch Second Pane**.
- When the cursor stays in a variable position in the implementation section, the cursor can be moved to the declaration position of the variable by selecting **Edit**>**Browse**>**Go To Definition**.
- You can also declare variables of user-defined types such as structure. User-defined types must be defined in DUT objects beforehand. For details, refer to "7.7.6 Structure, Enumeration, Alias, and Union Data Types".

6.2.2 Declaration Editor

The declaration editor is used to declare variables.

The display format of declarations is divided into table format and text format. The table format and text format can be switched using the switching button on the top right of the declaration editor.

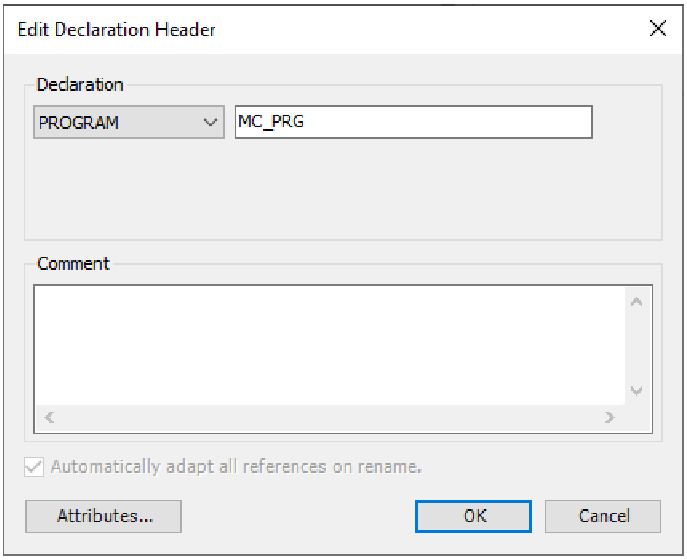
■ Table format



	Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR	r1, r2, r3		REAL			
2	VAR	x1		INT			
3	VAR	x2		INT			
4	VAR	x3		INT			
5	VAR	t_d1		TIME_OF_DAY			

- To add a new declaration, click the [ ] icon ("Insert") to add a new row. Enter a variable name in the "Name" column. For other items, double-click each cell to set the cell in an input-enabled state, and enter values as necessary.
- To sort variables, use the [ ] icon ("Move up" or "Move down").
- To delete variables, use the [✕] icon ("Delete").
- When adding a program name or program name comment, click the declaration header section.

  The "Edit Declaration Header" dialog box will be displayed.

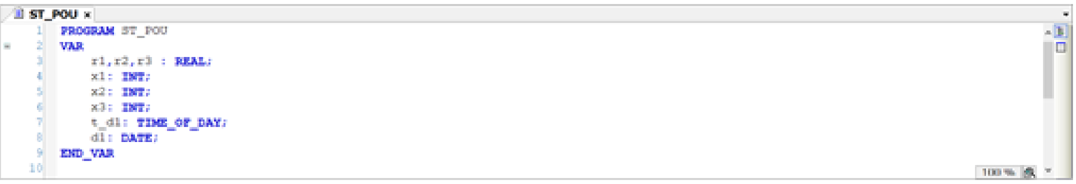


Declaration: PROGRAM MC_PRG

Comment:

☑ Automatically adapt all references on rename.

■ Text format



```
PROGRAM ST_POU
VAR
    r1,r2,r3 : REAL;
    x1: INT;
    x2: INT;
    x3: INT;
    t_d1: TIME_OF_DAY;
    d1: DATE;
END_VAR
```

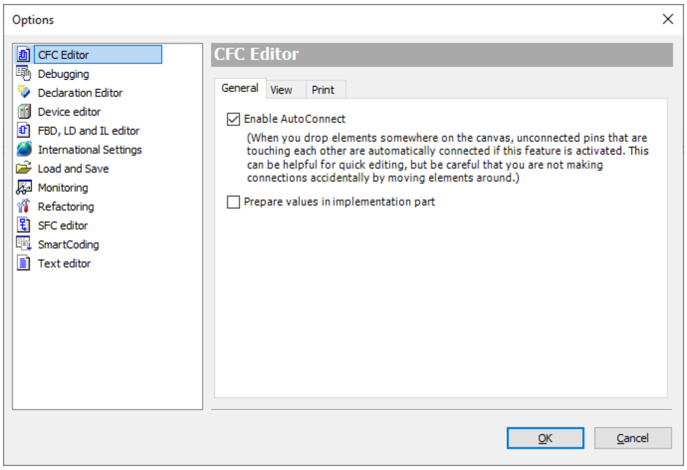
- As is the case with the text editor, enter variables to be declared.
- You can use single-line comments (prefixed with //) and multiple-line comments (enclosed with *).
- Pressing the "F2" key starts Input Assistant, which allows the user to enter variable types and other items by selecting them. For details on Input Assistant, refer to "6.8.5 Input Assistant Function".

Info.

- You can set the display format to be used, as below.

1. From the menu bar, select **Tools**>**Options**.

   The "Options" dialog box will be displayed.

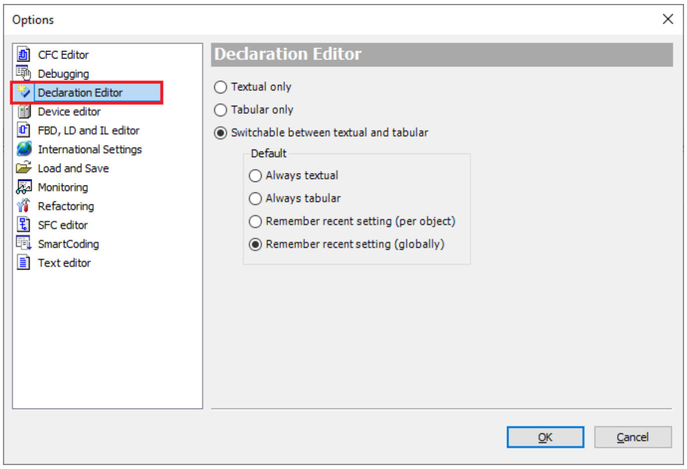


☑ Enable AutoConnect

(When you drop elements somewhere on the canvas, unconnected pins that are touching each other are automatically connected if this feature is activated. This can be helpful for quick editing, but be careful that you are not making connections accidentally by moving elements around.)

□ Prepare values in implementation part

2. In the Options dialog box, select the "Declaration Editor" category.



○ Textual only

○ Tabular only

● Switchable between textual and tabular

Default

○ Always textual

○ Always tabular

○ Remember recent setting (per object)

● Remember recent setting (globally)

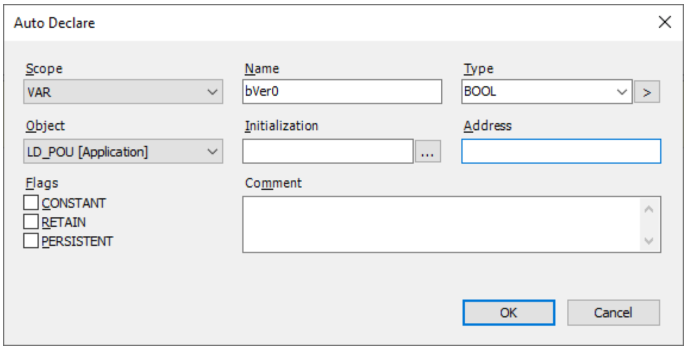
3. Select a desired display format.

6.2.3 Auto Declaration

If a variable that has not been declared in the declaration section is entered in the implementation section, the "Auto Declare" dialog box will be displayed.

If you change any necessary items and click the [OK] button, the variable will be declared in the declaration section.

Example: When the variable name of a contact is entered as bVar0 in an LD program



Auto Declare

Scope: VAR

Name: bVer0

Type: BOOL

Object: LD_POU [Application]

Initialization:

Address:

Flags:
□ CONSTANT
□ RETAIN
□ PERSISTENT

Comment:

■ Address

In the Address field, you can specify the address of input data or output data for the GM1 controller or expansion unit. In such a case, the variable assigned to the input data or output data corresponding to the entered address is declared.

■ Flag

If you select the CONSTANT, RETAIN, and PERSISTENT check boxes in the Flag section, you can set variable attributes.

CONSTANT

Declares the variable as a constant. Enter a default value.

RETAIN

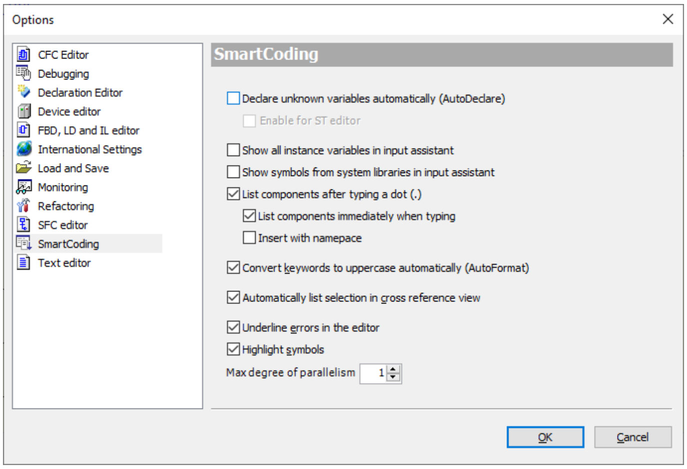
Declares the variable as a retain variable. For retain variables, values are not reset even if warm reset is performed. For details on warm reset, refer to "9.5.1 Reset Warm, Reset Cold, and Reset Origin".

PERSISTENT

Declares the variable as a persistent variable. To declare a persistent variable, you must also select the RETAIN check box. For persistent variables, values are not reset even if cold reset or warm reset is performed.For details on cold reset or warm reset, refer to "9.5.1 Reset Warm, Reset Cold, and Reset Origin".

Info.

- You can also prevent the "Auto Declare" dialog box from being displayed when a variable that has not been declared is entered in the implementation section. From the menu bar, select **Tools**>**Options**>**SmartCoding** category and clear the "Delcare unknown variables automatically (AutoDeclare)" check box.



SmartCoding

□ Declare unknown variables automatically (AutoDeclare)

□ Enable for ST editor

□ Show all instance variables in input assistant

□ Show symbols from system libraries in input assistant

☑ List components after typing a dot (.)

☑ List components immediately when typing

□ Insert with namepace

☑ Convert keywords to uppercase automatically (AutoFormat)

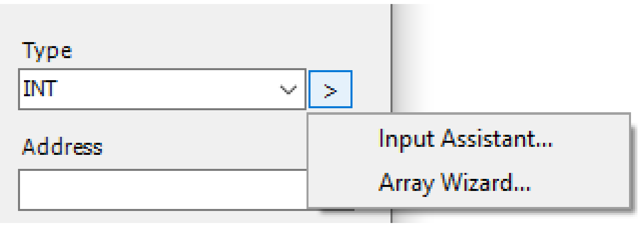
☑ Automatically list selection in cross reference view

☑ Underline errors in the editor

☑ Highlight symbols

Max degree of parallelism 1

- Using the Array Wizard, you can declare an array only by entering an index and base type. Click > on the right side of the Type field and select "Array Wizard".

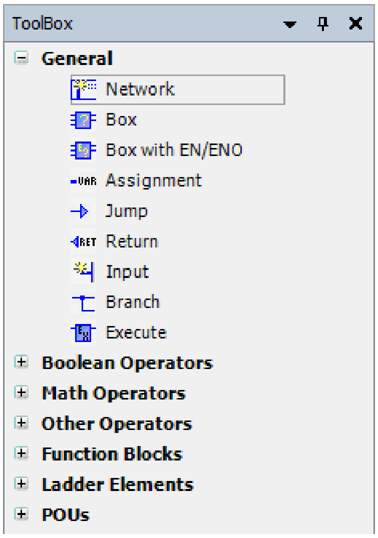


6.2.4 Toolbox

Programs can be created by dragging the programming elements displayed in the toolbox. For programs other than ST programs, programming elements are displayed in the toolbox.

For details on how to create programs, refer to the section related to each program creation.

Example: Toolbox for LD programs



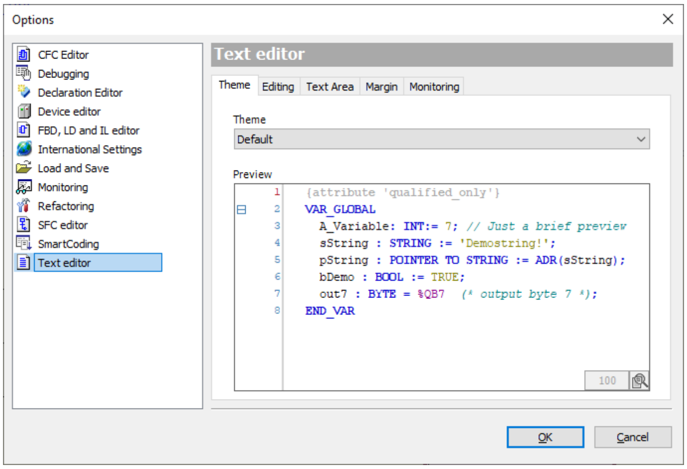
6.2.5 Setting up the Program Input Window

You can change settings related to the text editor.

From the menu bar, select **Tools**>**Options** to open the "Options" dialog box.

In the "Options" dialog box, select the "Text editor" category and change the settings.

Theme



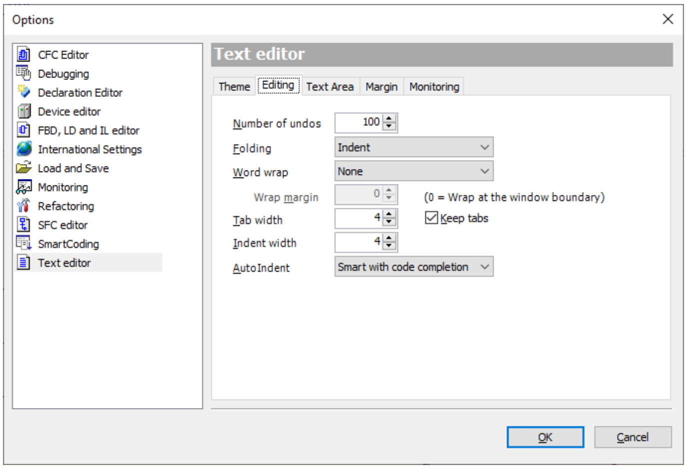
Theme: Default

Preview

```
{attribute 'qualified_only'}
VAR_GLOBAL
  A_Variable: INT:= 7; // Just a brief preview
  sString : STRING := 'Demostring!';
  pString : POINTER TO STRING := ADR(sString);
  bDemo : BOOL := TRUE;
  out7 : BYTE = %QB7  (* output byte 7 *);
END_VAR
```

Item name	Default value	Settings
Theme	Default	Sets a color scheme theme for the text editor Default / Dark

Edit



Number of undos: 100

Folding: Indent

Word wrap: None

Wrap margin: 0 (0 = Wrap at the window boundary)

Tab width: 4 ☑ Keep tabs

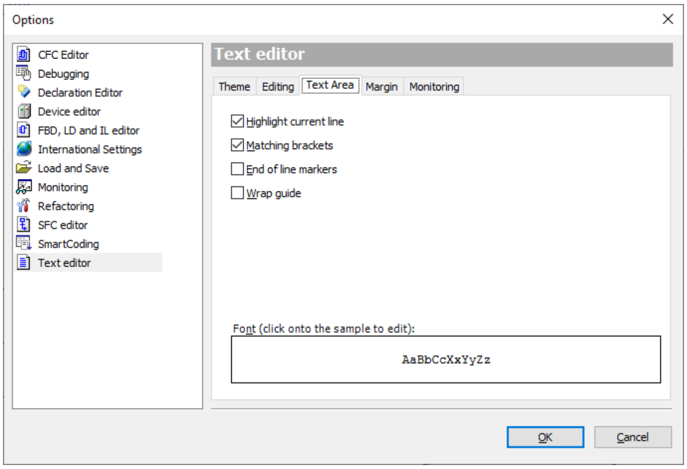
Indent width: 4

AutoIndent: Smart with code completion

Item name	Default value	Settings
Number of undos	100	Sets the maximum number of times [Edit]-[Undo] can be executed from the menu bar Setting range: 1 to 1000 (times)
Folding	Indent	Specifies the method for defining the code structure **None:** Does not collapse code **Indent:** Combines all lines indented from the previous line as a unit **Explicit:** Explicitly marks a unit of code sections as a comment
Word wrap	None	Sets a rule for wrapping entered text automatically **None:** Does not collapse code **Soft:** If the number of characters entered in a single line exceeds the value specified in “Wrap margin”, a code continuation mark (:) will be added and a line break will be inserted automatically. If "0" is selected in the "Wrap margin" drop-down list, a line break will be inserted at the right edge of the editor window. **Hard:** If the number of characters entered in a single line exceeds the value specified in "Wrap margin", a line break will be inserted automatically. However, a code continuation mark (":") will not be added. If the number of initially entered word characters exceeds the value specified in "Wrap margin", a line break will not be inserted.
Wrap margin	0	Specifies the number of characters per line that triggers a line feed

Item name	Default value	Settings
		Setting range: 0 to 240
Tab width	4	Specifies the number of space characters equivalent to the code to be inserted when the Tab key is pressed Setting range: 1 to 16
Keep tabs	Selected	Specifies whether to insert space characters or a tab character when the Tab key is pressed Selected: Inserts a tab character when the Tab key is pressed Cleared: Inserts space characters when the Tab key is pressed
Indent width	4	Inserts tab spaces with the specified width when "Auto" or "Auto coding" is selected from the "Auto indent" drop-down list. However, if the "Keep tabs" check box is cleared, space characters will be inserted. Setting range: 1 to 16
Auto Indent	Smart with code completion	Specifies the behavior to be performed when auto indentation is performed **None:** Does not insert indentation automatically **Block:** Inserts indentation with the same width as that of the previous line at the time of line feed **Auto:** Inserts indentation automatically for lines following a line containing keywords (such as VAR) according to the setting of "Indent width" **Smart with code completion:** Inserts applicable keywords such as "END_IF" and "END_VAR" automatically, in addition to the behavior performed by "Auto"

Text Area



Text editor

☑ Highlight current line

☑ Matching brackets

☐ End of line markers

☐ Wrap guide

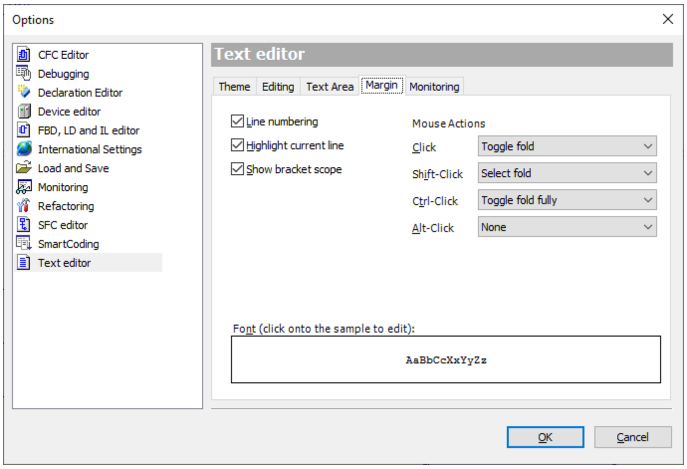
Font (click onto the sample to edit):

AaBbCcXxYyZz

Item name	Default value	Settings
Highlight current line	Selected	Highlights the line selected by the cursor Selected / Cleared

Item name	Default value	Settings
Matching brackets	Selected	Highlights the corresponding bracket when the cursor is positioned at a bracket within code Selected / Cleared
End of line markers	Cleared	Indicates the end of a line as a small dash mark (".") with the color specified for the theme Selected / Cleared
Wrap guide	Cleared	Displays a guide as the vertical line specified for the theme, in the column used as the base of wrapping. If any value other than "0" is specified in "Wrap margin", a guide will be displayed. Selected / Cleared
Font	-	Displays a font dialog box for setting fonts.

Margin



☑ Line numbering

☑ Highlight current line

☑ Show bracket scope

Mouse Actions

Click	Toggle fold
Shift-Click	Select fold
Ctrl-Click	Toggle fold fully
Alt-Click	None

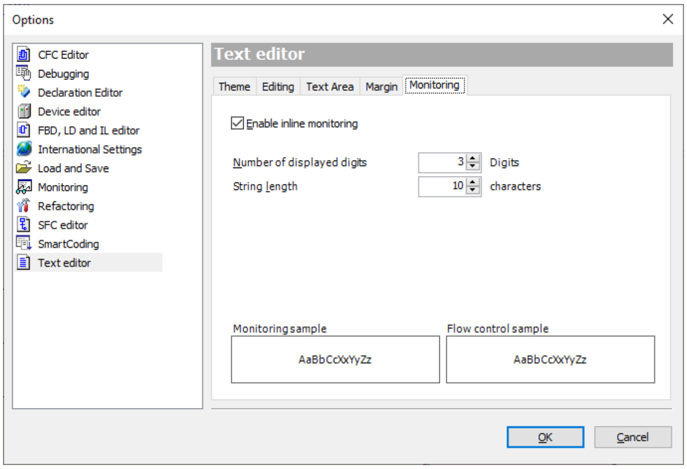
Font (click onto the sample to edit):

AaBbCcXxYyZz

Item name	Default value	Settings
Line numbering	Selected	Displays line numbers in the variable declaration section and program implementation section
Highlight current line	Selected	Selected: Displays line numbers Cleared: Does not display line numbers Highlights the line selected by the cursor by changing the color of the line number. If the "Line numbering" check box is cleared, the current line will not be highlighted. Selected: Highlights the current line by changing the color of the line number Cleared: Does not change the color of the line number
Show bracket scope	Selected	Displays a scope in the space on the left side of the line number to indicate the beginning to the end of a keyword (such as "IF" to "END_IF") Selected: Displays a scope Cleared: Does not display a scope

Item name	Default value	Settings
Mouse Actions	-	Assigns a mouse action to be performed when "+" or "-" in a space is clicked None: Assigns no mouse action Select fold: Selects all lines within the area enclosed in brackets Toggle fold: Expands or folds the area enclosed in brackets Toggle fold fully: Expands or folds all nested areas when areas are nested
Font	-	Displays a "Font" dialog box for setting fonts

Monitoring



☑ Enable inline monitoring

Number of displayed digits 3 Digits

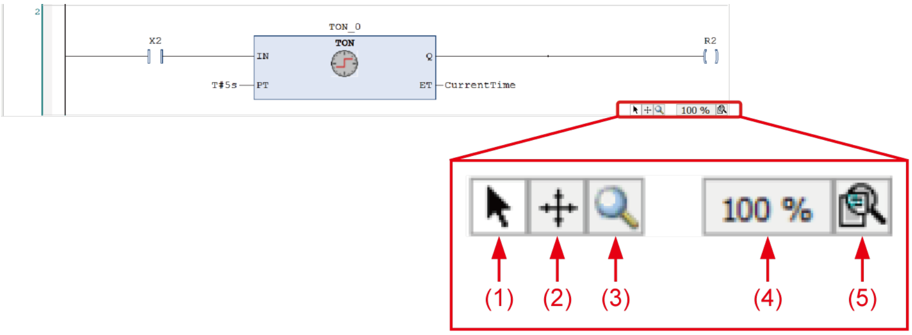
String length 10 characters

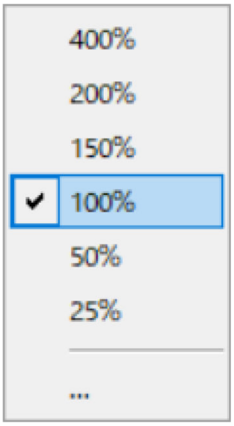
Item name	Default value	Settings
Enable inline monitoring	Selected	Displays a monitoring field in the program implementation section in online mode Selected: Displays a monitoring field Cleared: Does not display a monitoring field
Number of displayed digits	3	Sets the number of digits after the decimal point that are displayed in the monitoring field Setting range: 1 to 20
String length	10	Sets the maximum length of string variables in the monitoring field Setting range: 1 to 80

6.2.6 Window Operations for the Program Input Window

You can perform window operations such as increasing the display size in the program input window.

The window operation icons are displayed in the bottom right corner of the window.



Number	Item	Description
(1)	Normal mode	This mode allows the user to select a component by clicking it.
(2)	Move mode	This mode allows the user to moves the window by clicking in the window and then dragging the mouse.
(3)	Enlarged view tool	Clicking the icon opens the enlarged view tool window. The display at the cursor position is enlarged in the window.
(4)	Current display size	This section indicates the current display size of the program input window
(5)	Change display size	This icon is used to change the display size. Clicking the icon displays a menu. Select a desired size.  Selecting ... displays the "Enlarge" dialog box where you can enter a magnification rate.

6.3 Creating a Program Object (POU Object)

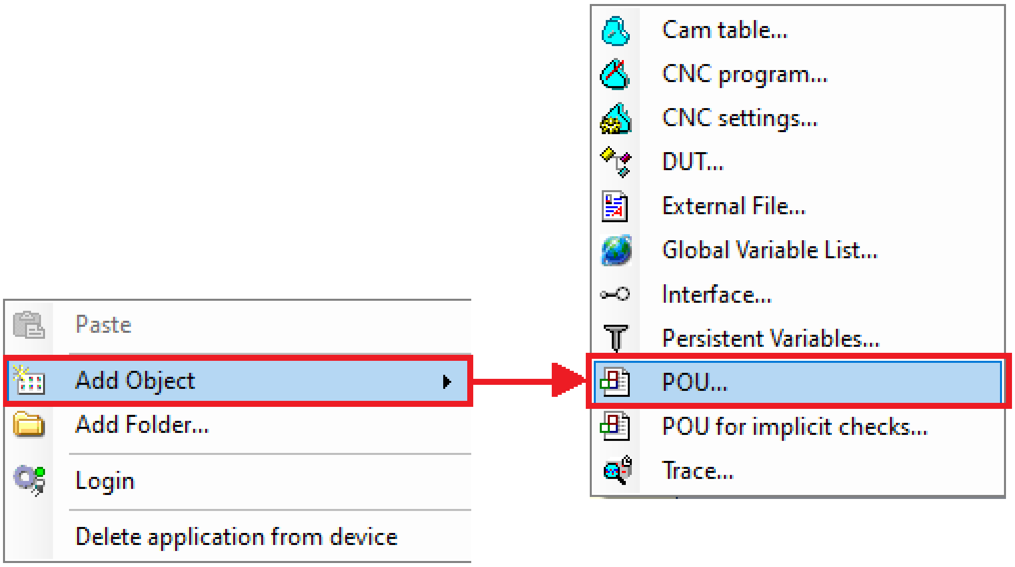
■ Creating programs

Programs are created within POU objects. A single POU object allows the use of only one program. If you want to use different programs within a project, you must add POU objects.

■ Adding POU objects

To add POU objects, right-click the [Application] object in the navigator pane and then select **Add Object**>**POU** from the context-sensitive menu that is displayed.

For details, refer to "4.7 Adding Objects".



6.4 Types of Programming Language

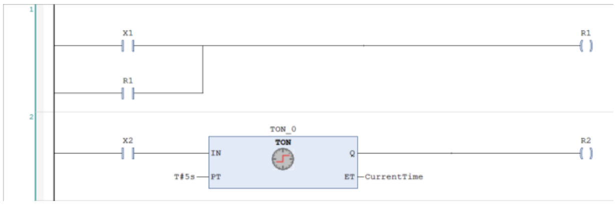
GM Programmer supports five programming languages that comply with IEC 61131-3, the international standard for PLC programming languages.

Info.

- Continuous Function Chart (CFC) and Page-Oriented CFC are not included in the five programming languages compliant with IEC 61131-3. However, the third edition of IEC 61131-3 defines them as object-oriented programming languages.

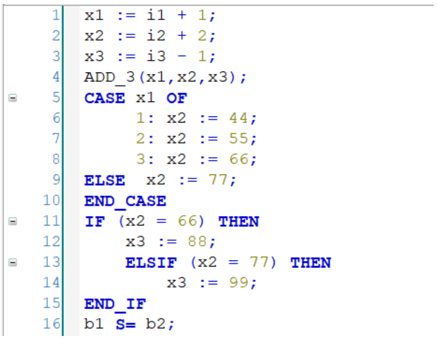
■ Ladder Diagram program (LD program)

Ladder Diagram is a graphical programming language used to create a program by arranging ladder logic elements such as contacts and coils on a network (circuit). It also allows the use of functions and function blocks with various functions.



■ Structured Text program (ST program)

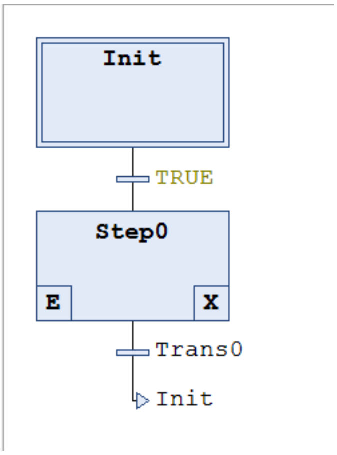
Structured Text is a programming language that creates expressions, conditional statements, and other program elements in text format. It is based on the Pascal programming language and suitable for numerical calculation, data processing, and processing such as conditional branch and repetitive processing.



```
x1 := i1 + 1;
x2 := i2 + 2;
x3 := i3 - 1;
ADD_3(x1,x2,x3);
CASE x1 OF
     1: x2 := 44;
     2: x2 := 55;
     3: x2 := 66;
ELSE  x2 := 77;
END_CASE
IF (x2 = 66) THEN
    x3 := 88;
    ELSIF (x2 = 77) THEN
        x3 := 99;
END_IF
b1 S= b2;
```

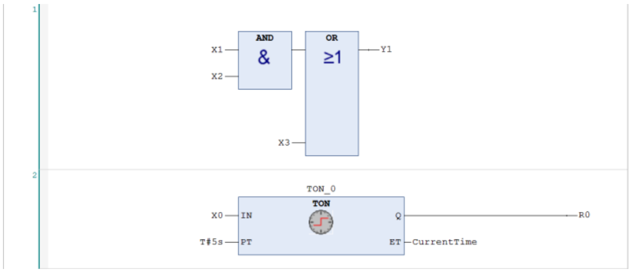
■ Sequential Function Chart program (SFC program)

Sequential Function Chart is a graphical programming language used to create a program by arranging steps, transitions, actions, and other elements sequentially from top to bottom. It is suitable for processing that describes state transitions.



■ Function Block Diagram program (FBD program)

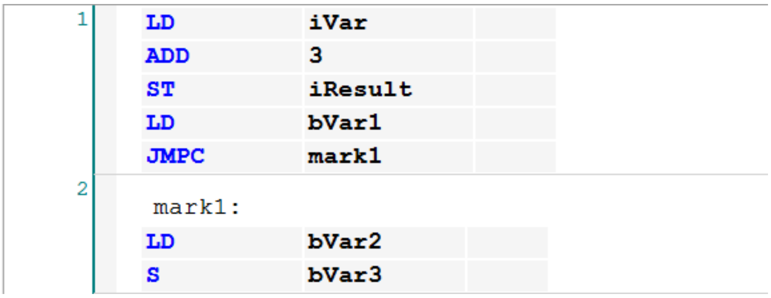
Function Block Diagram is a graphical programming language used to create a program by arranging functions and function blocks on a network (circuit). Unlike Ladder Diagram programs, contacts, coils and other ladder logic elements cannot be arranged in Function Block Diagram programs.



In addition to the five programming languages compliant with IEC 61131-3, Continuous Function Chart (CFC) and Page-Oriented CFC can also be used.

■ Instruction List program (IL program)

Instruction List is a programming language that creates assembler-like instructions sequentially in text format. This language is suitable when you want to perform high-speed processing, restrict the memory usage, and perform other similar operations.



```
LD      iVar
ADD     3
ST      iResult
LD      bVar1
JMPC    mark1

 mark1:
LD      bVar2
S       bVar3
```

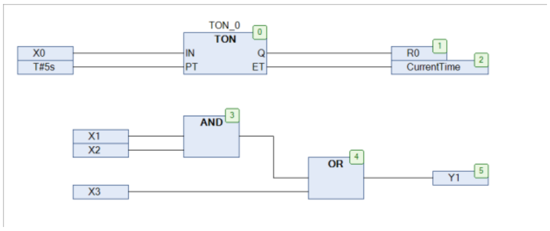
In addition to the five programming languages compliant with IEC 61131-3, Continuous Function Chart (CFC) and Page-Oriented CFC can also be used.

■ Continuous Function Chart (CFC) and Page-Oriented CFC programs

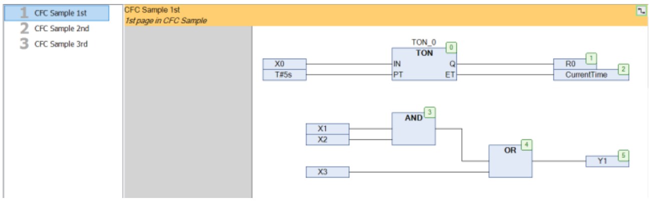
Continuous Function Chart (CFC) and Page-Oriented CFC are graphical programming languages used to create a program by arranging function blocks and other elements on the screen. Elements can be freely arranged on the screen and the order of execution can be specified.

CFC allows the user to create a program on a single screen, while Page-Oriented CFC allows the user to create a program while switching the screen called a page.

<CFC program>



<Page-Oriented CFC program>



6.5 Function and Function Block

Functions and function blocks can be invoked from programs. Functions and function blocks can be created with POU objects.

The differences between functions and function blocks are as below.

■ Function (FUN)

- Functions can be used without being declared in the declaration section.
- Only one output is generated. However, additional outputs can be defined.
- Output variable and internal variable values are not saved.

■ Function block (FB)

- Function blocks can be used by declaring instances in the declaration section.
- Multiple outputs can be generated.
- Output variable and internal variable values are saved.
- Object-oriented definitions can be made by using EXTENDS (inheritance), IMPLEMENTS (interface implementation), or access qualifiers.

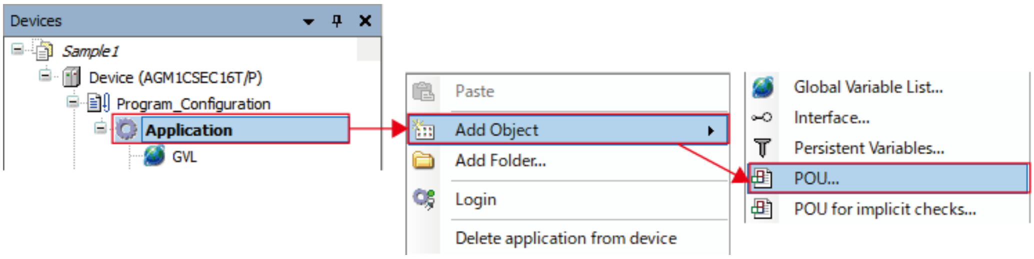
6.5.1 Function

A function generates one output in response to one or more inputs. Functions can be used without declaring variables.

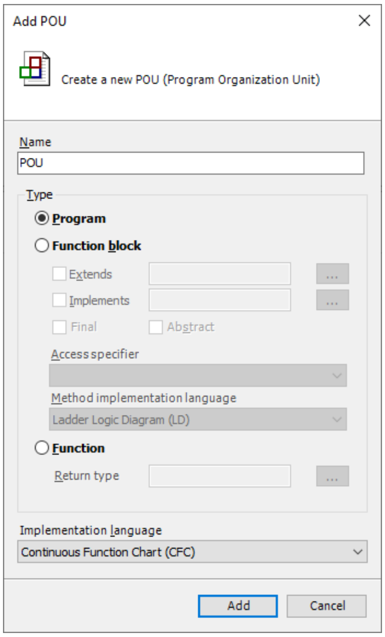
For example, use the following procedure to create and invoke function "ADD_SUB", which uses three INT type arguments as inputs, calculates (first argument) + (second argument) - (third argument), and output the result.

1 2 Procedure

1. Right-click the [Application] object in the navigator pane and then select **Add Object>POU** from the context-sensitive menu that is displayed.



The "Add POU" dialog box will be displayed.


Add POU

Create a new POU (Program Organization Unit)

Name: POU

Type

● Program

○ Function block

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

Method implementation language: Ladder Logic Diagram (LD)

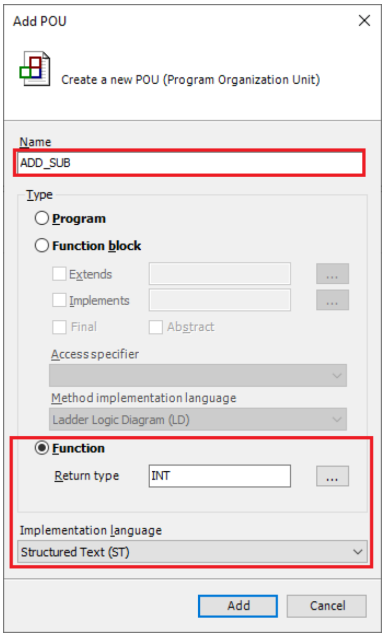
○ Function

Return type

Implementation language: Continuous Function Chart (CFC)

2. Select the "Function" radio button, enter appropriate values in the Name and Return type fields, and select an appropriate programming language from the Implementation Language drop-down list.

In the Name field, specify a function name. In the Return type field, select a return value to be output when the function is executed. From the Implementation Language drop-down list, select a programming language that is used to code function processing.


Add POU

Create a new POU (Program Organization Unit)

Name: ADD_SUB

Type

○ Program

○ Function block

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

Method implementation language: Ladder Logic Diagram (LD)

● Function

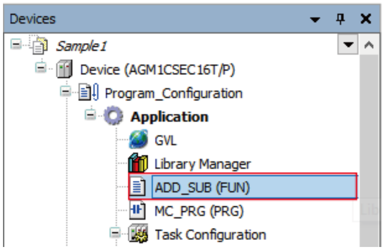
Return type: INT

Implementation language: Structured Text (ST)

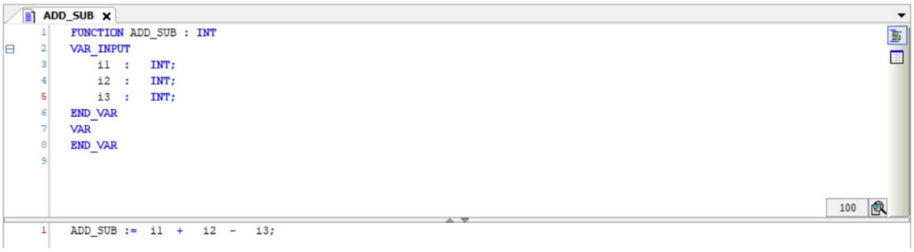
3. Click the [Add] button.

A POU object of the function will be added.

The POU object is displayed as "function-name (FUN)" in the navigator pane.



4. Enter function processing.
Open the POU object of the function and create a function.
In "VAR_INPUT", declare input variables for the function.
Substitute function output for the variable of the function name.



```
FUNCTION ADD_SUB : INT
VAR_INPUT
    i1  :   INT;
    i2  :   INT;
    i3  :   INT;
END_VAR
VAR
END_VAR
```

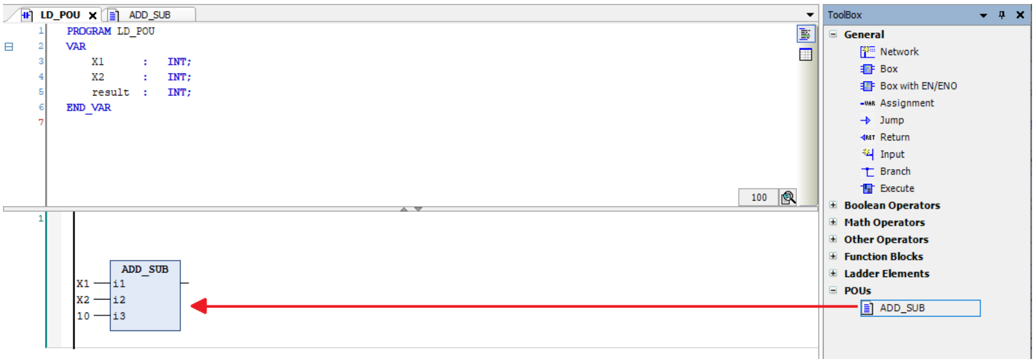
```
ADD_SUB := i1  +  i2  -  i3;
```

This completes the function creation procedure.

Next, the procedure for invoking the created function is explained below.

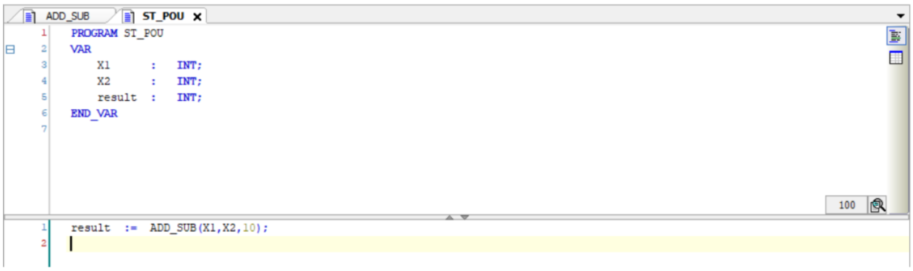
5. Open the POU object from which the function is to be invoked, and invoke the function.
The function can be invoked by using its name. To invoke the function, there is no need to declare variables.

Example: Invoking the function from LD program



```
PROGRAM LD_POU
VAR
    X1      :   INT;
    X2      :   INT;
    result  :   INT;
END_VAR
```

Example: Invoking the function from ST program



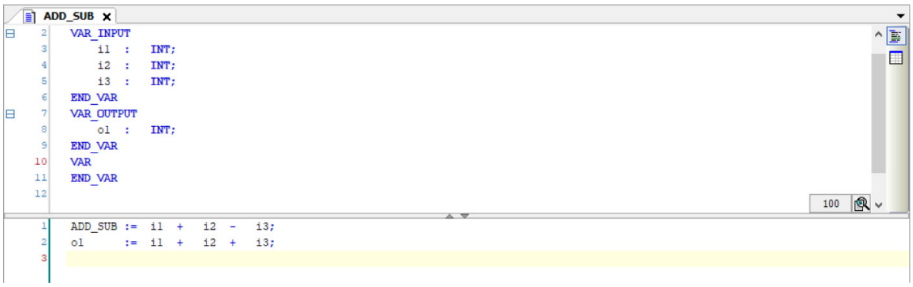
```
PROGRAM ST_POU
VAR
    X1      :   INT;
    X2      :   INT;
    result  :   INT;
END_VAR
```

```
result  :=  ADD_SUB(X1,X2,10);
```

ⓘ Info.

- Additional outputs can be defined for the function. Declare an additional output as variable "VAR_OUTPUT" in the declaration section of the POU object that defines the function.

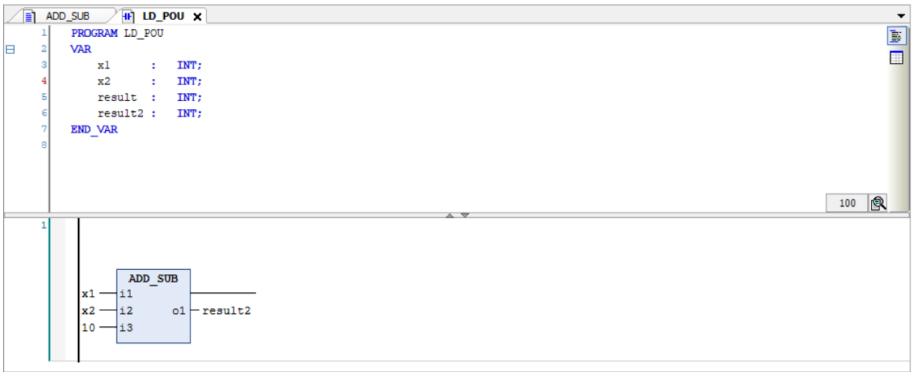
Example: Definition of function "ADD_SUB" to which variable iOut that outputs the sum of three input variables is added



```
VAR_INPUT
    i1  :   INT;
    i2  :   INT;
    i3  :   INT;
END_VAR
VAR_OUTPUT
    o1  :   INT;
END_VAR
VAR
END_VAR
```

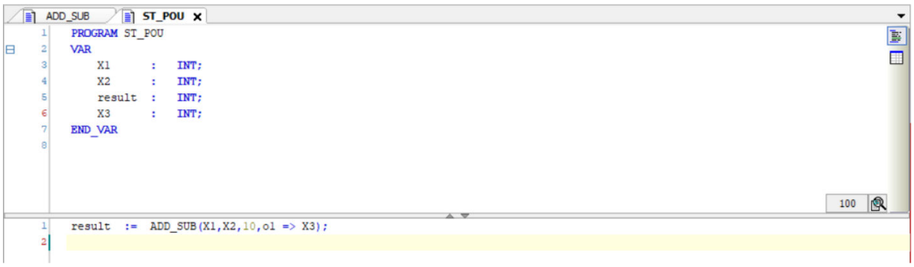
```
ADD_SUB :=  i1  +   i2  -   i3;
o1      :=  i1  +   i2  +   i3;
```

Invoking function "ADD_SUB" from LD program



```
PROGRAM LD_POU
VAR
    x1      :   INT;
    x2      :   INT;
    result  :   INT;
    result2 :   INT;
END_VAR
```

Invoking function "ADD_SUB" from ST program



```
PROGRAM ST_POU
VAR
    X1      :   INT;
    X2      :   INT;
    result  :   INT;
    X3      :   INT;
END_VAR
```

```
result  :=  ADD_SUB(X1,X2,10,o1 => X3);
```

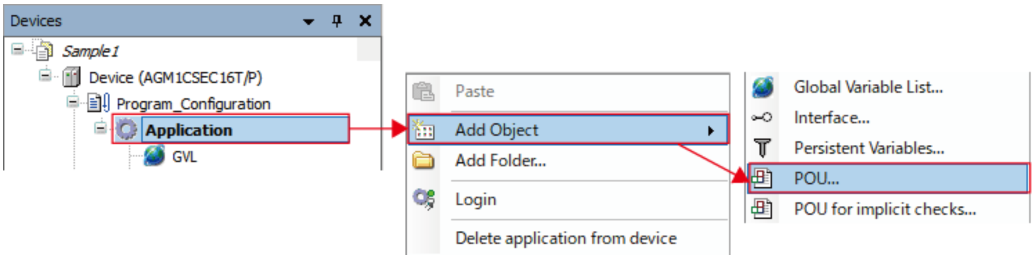
6.5.2 Function Block

A function block generates one or more outputs in response to one or more inputs. To use function blocks, variables (instances) must be declared.

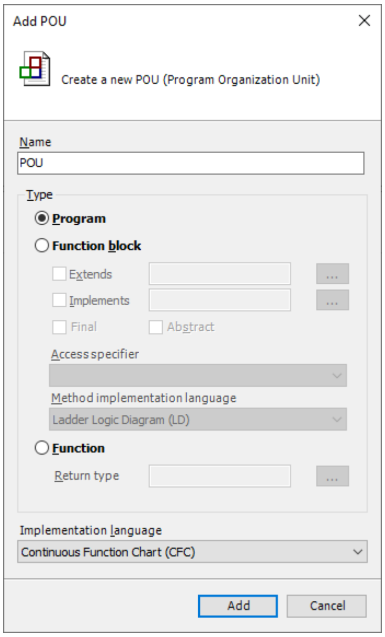
For example, use the following procedure to create function block "FB_ADD", which uses three INT type variables as inputs and outputs the sum of three arguments, and to invoke an instance.

1 2 Procedure

1. Right-click the [Application] object in the navigator pane and then select **Add Object**>**POU** from the context-sensitive menu that is displayed.



The "Add POU" dialog box will be displayed.



Add POU

Create a new POU (Program Organization Unit)

Name
POU

Type

● Program

○ Function block

□ Extends ...

□ Implements ...

□ Final □ Abstract

Access specifier

Method implementation language
Ladder Logic Diagram (LD)

○ Function

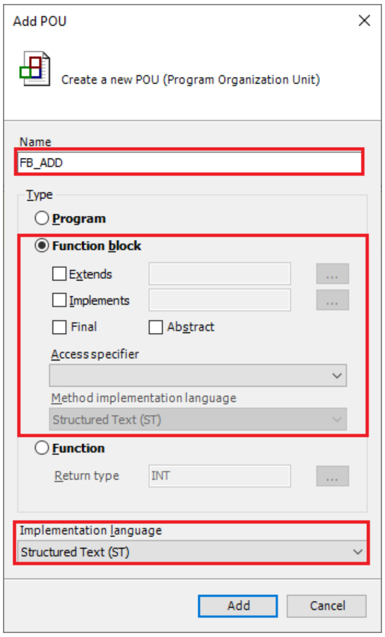
Return type ...

Implementation language
Continuous Function Chart (CFC)

Add Cancel

2. Select the "Function block" radio button, enter a name in the Name field, and select a programming language from the Implementation Language drop-down list.

In the Name field, specify a function block name. In the Implementation Language drop-down list, select a programming language that is used to code function block processing.



Add POU

Create a new POU (Program Organization Unit)

Name

FB_ADD

Type

○ Program

● Function block

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

Method implementation language

Structured Text (ST)

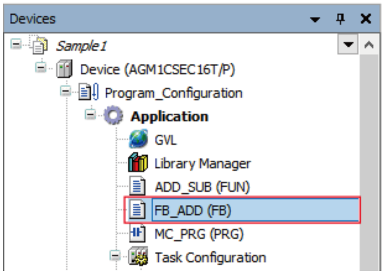
○ Function

Return type INT

Implementation language

Structured Text (ST)

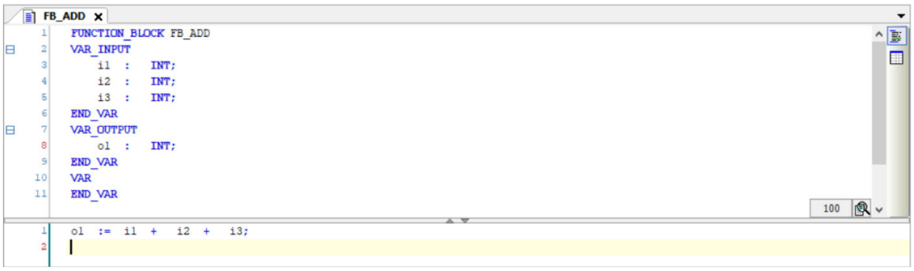
3. Click the [Add] button.
A POU object of the function block will be added.
The POU object is displayed as "function-block-name (FB)" in the navigator pane.



Devices

- Sample1
  - Device (AGM1CSEC16T/P)
    - Program_Configuration
      - Application
        - GVL
        - Library Manager
        - ADD_SUB (FUN)
        - FB_ADD (FB)
        - MC_PRG (PRG)
        - Task Configuration

4. Enter function block processing.
Open the POU object of the function block and create a function block.
In "VAR_INPUT", declare input variables to the function block.
In "VAR_OUTPUT", declare output variables from the function block.



FB_ADD

```
FUNCTION_BLOCK FB_ADD
VAR_INPUT
    i1  :   INT;
    i2  :   INT;
    i3  :   INT;
END_VAR
VAR_OUTPUT
    o1  :   INT;
END_VAR
VAR
END_VAR
```

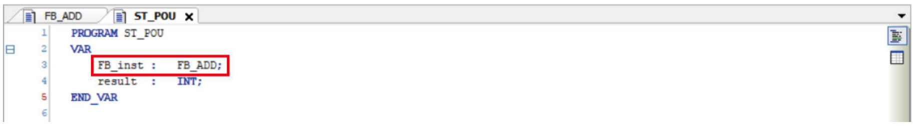
```
o1  :=  i1  +   i2  +   i3;
```

This completes the function block creation procedure.
Next, the procedure for invoking the created function block is explained below.

5. Open the POU object from which the function block is to be invoked, and declare an instance of the function block in the declaration section.

   Declare an instance that is a copy of the function block.

   Declare an instance name in the form of "instance-name: function-block-name".

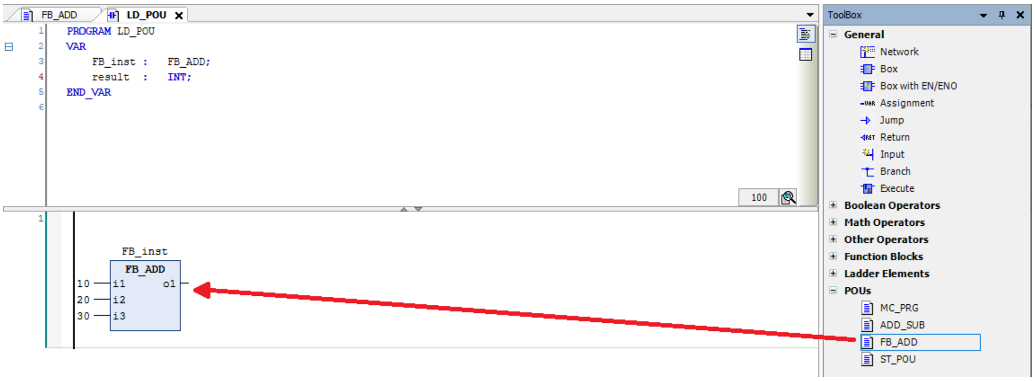


```
PROGRAM ST_POU
VAR
    FB_inst :   FB_ADD;
    result  :   INT;
END_VAR
```

6. Invoke the instance of the function block.

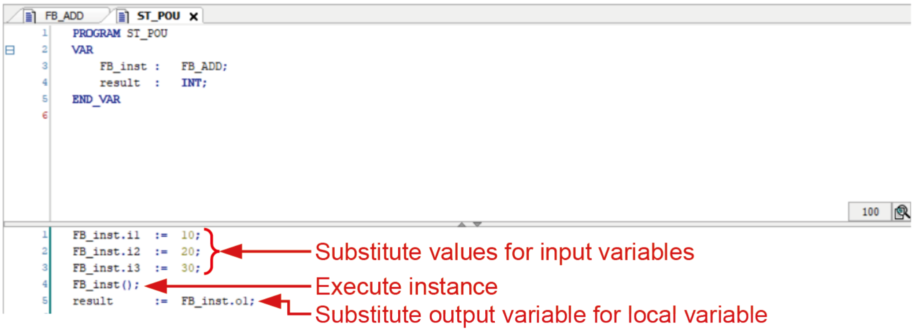
   Invoking the instance of the function block executes the processing defined with the function block. Input variables and output variables can be accessed by specifying "instance.variable-name".

Example: Invoking the instance from LD program



```
PROGRAM LD_POU
VAR
    FB_inst :   FB_ADD;
    result  :   INT;
END_VAR
```

Example: Invoking the instance from ST program



```
PROGRAM ST_POU
VAR
    FB_inst :   FB_ADD;
    result  :   INT;
END_VAR
```

```
FB_inst.i1  :=  10;
FB_inst.i2  :=  20;
FB_inst.i3  :=  30;
FB_inst();
result      :=  FB_inst.o1;
```

Substitute values for input variables

Execute instance

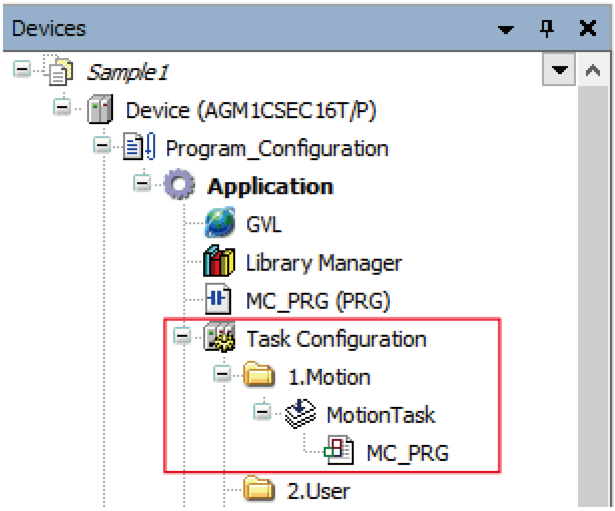
Substitute output variable for local variable

6.6 Tasks

The GM1 series motion controller executes the following three tasks.

Task	Description
MotionTask	This is a user program task to perform motion control. It is given the highest priority. Only one MotionTask is allowed for each project.
UserTask	This is a user program task to perform control other than motion control. The user can set the level of priority. Up to 50 tasks can be registered in a single project.
SystemTask	This is a task that is used by the system and cannot be added by user programs. It is processed while other tasks are inactive.

To execute a user program, the program (POU object) must be added to tasks.
Add motion control programs to MotionTask. Add other control programs to UserTask.

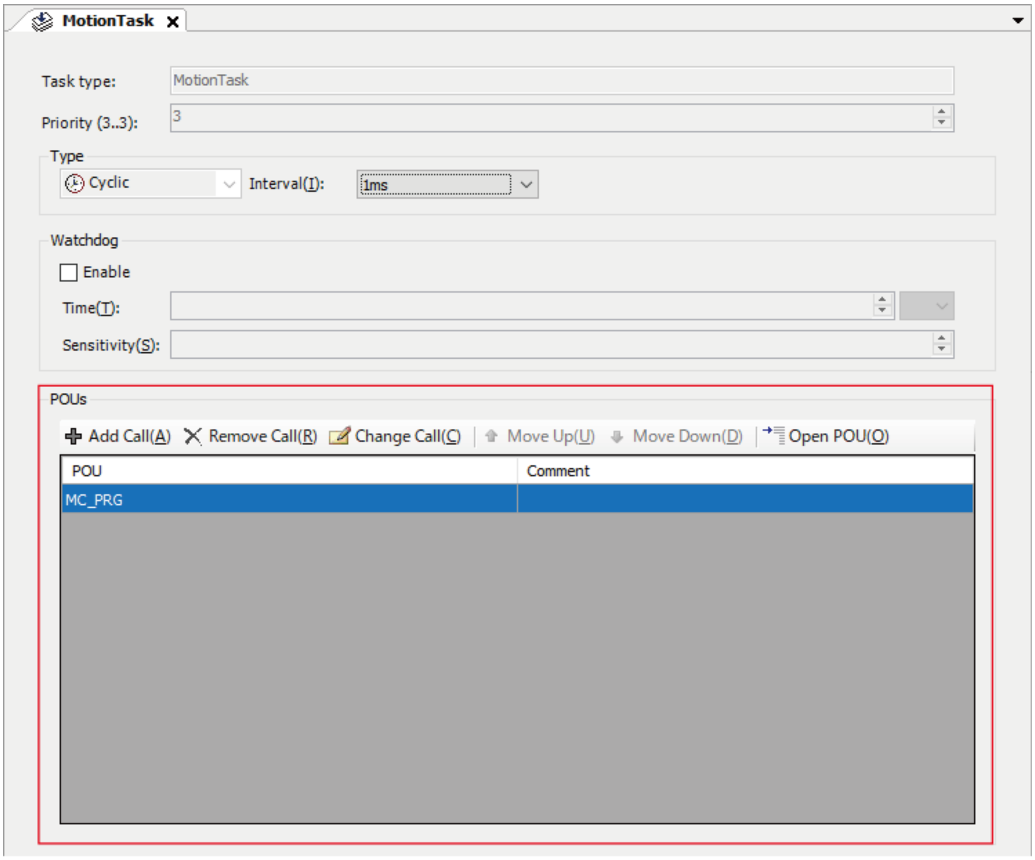


6.6.1 Adding Programs

POU objects of programs can be added to tasks.
For example, use the following procedure to add program "LD_POU" to task "MotionTask".

Procedure

1. Double-click the "MotionTask" object in the navigator pane.
The "MotionTask" task configuration window will be displayed in the main pane.



Task type: MotionTask

Priority (3..3): 3

Type

Cyclic Interval(I): 1ms

Watchdog

□ Enable

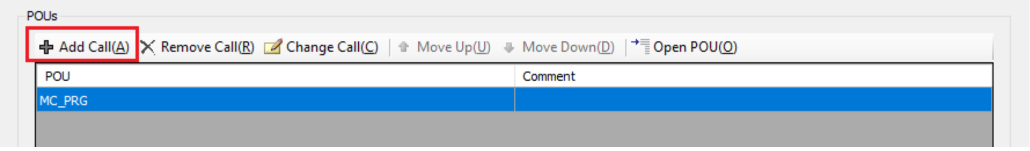
Time(T):

Sensitivity(S):

POUs

POU	Comment
MC_PRG	

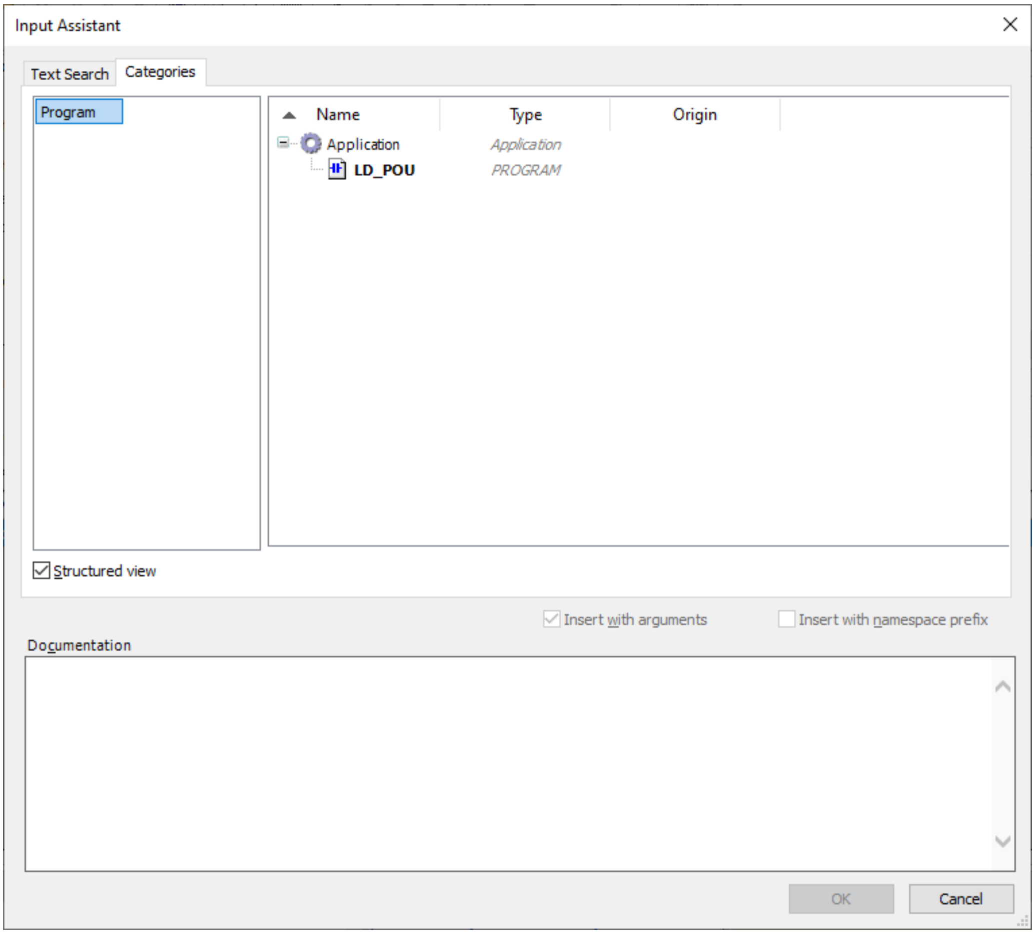
2. In the "MotionTask" window, click the [Add Call] button.



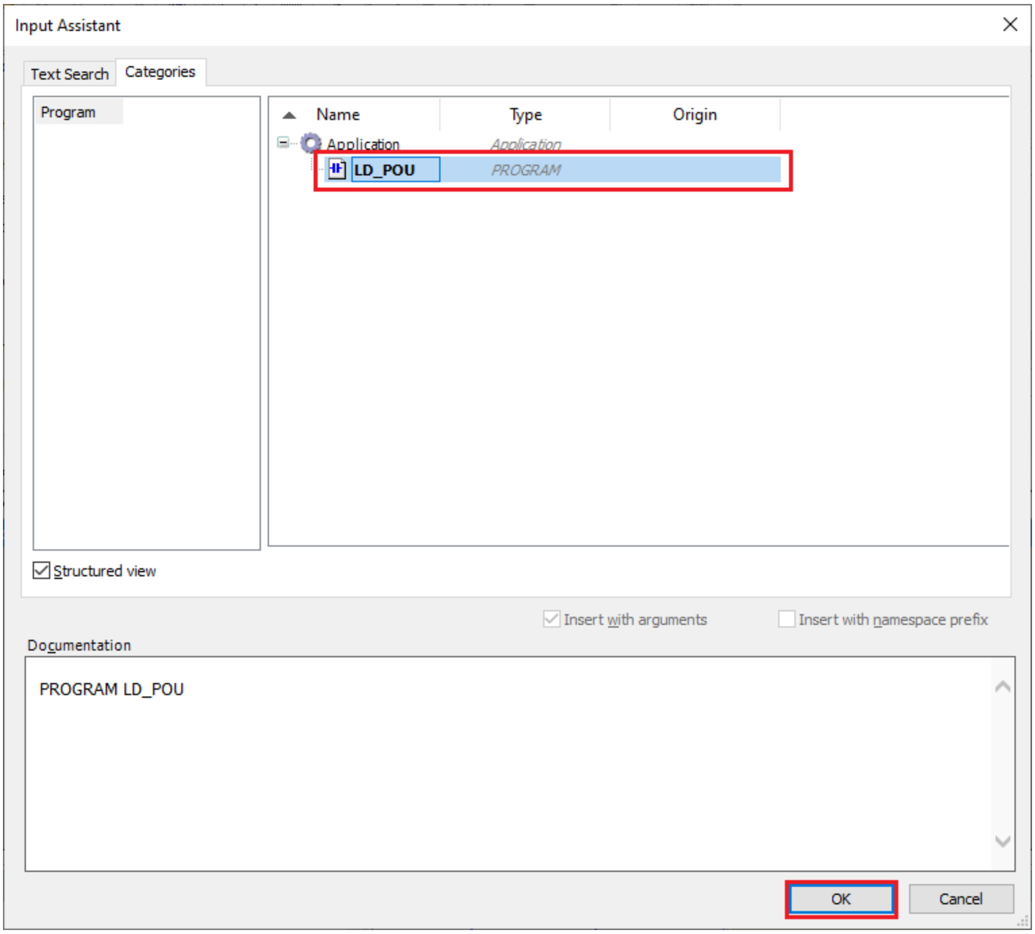
POUs

POU	Comment
MC_PRG	

The "Input Assistant" dialog box will be displayed.

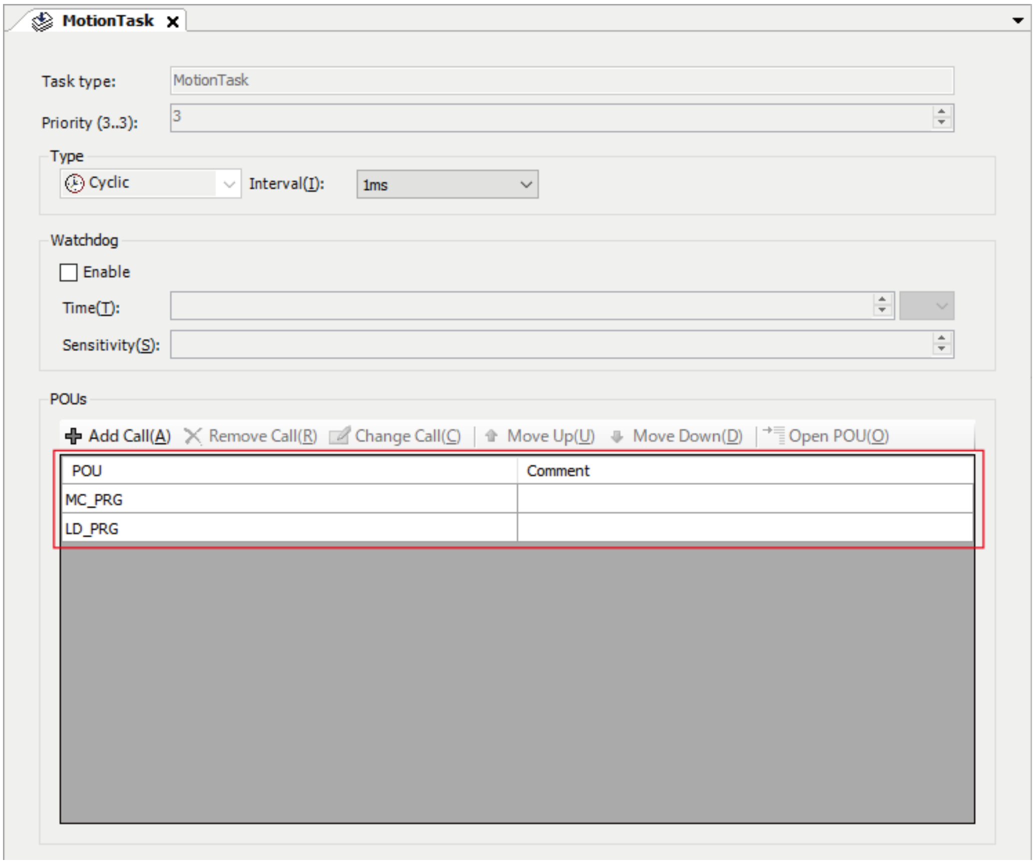


3. Select the POU object (LD_POU) of the program to be added to the task and click the [OK] button.



PROGRAM LD_POU

The POU object of the program will be added to the task.



Task type: MotionTask

Priority (3..3): 3

Type: Cyclic

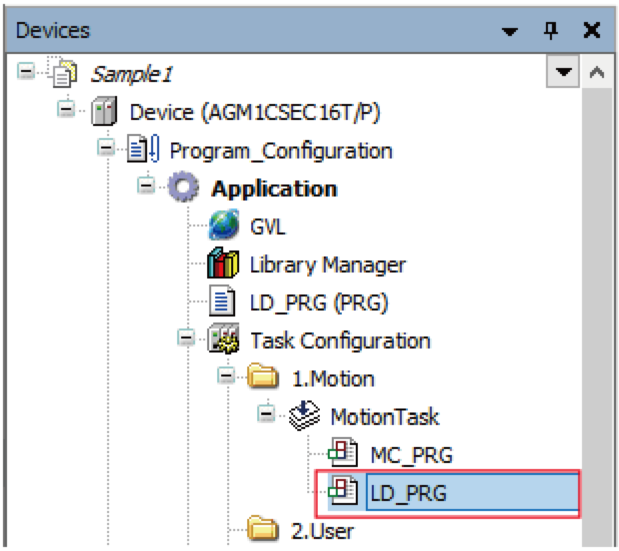
Interval(I): 1ms

Watchdog

□ Enable

POU	Comment
MC_PRG	
LD_PRG	

Selecting a cell in the "Comment" column allows you to edit the comment.
The task will also be added to the navigator pane.



Info.

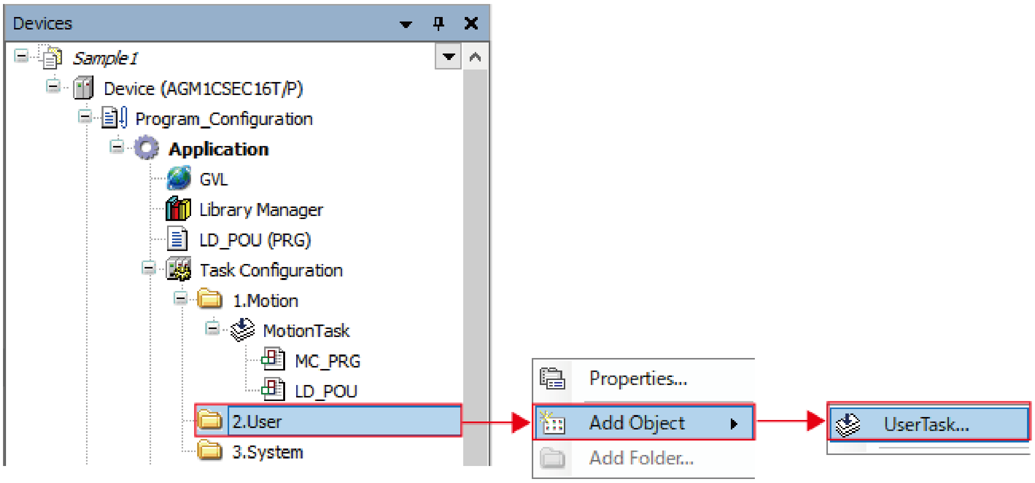
- You can also add a POU object to a task by dragging the POU object in the navigator pane and dropping it onto the task object.

6.6.2 Adding a UserTask

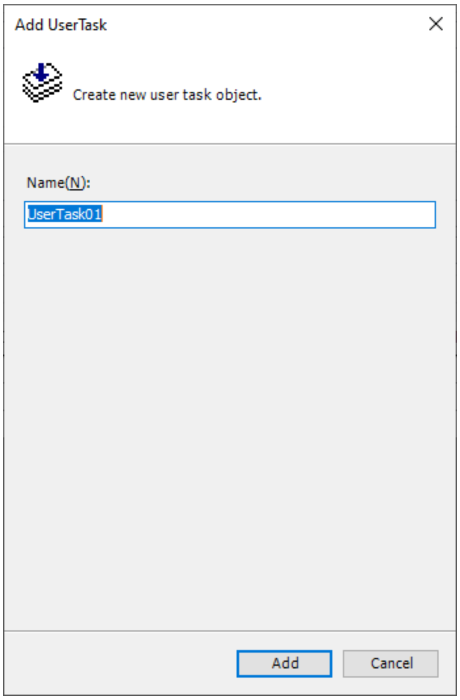
UserTask can be added to a project.

1 2 Procedure

1. Right-click "2.User" in the navigator pane and then select **Add Object**>**UserTask** from the context-sensitive menu that is displayed.



The "Add UserTask" dialog box will be displayed.

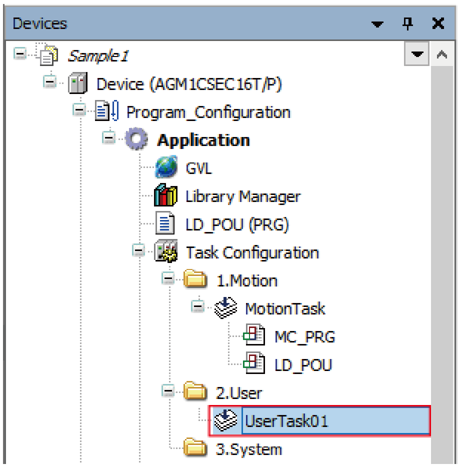

Add UserTask

Create new user task object.

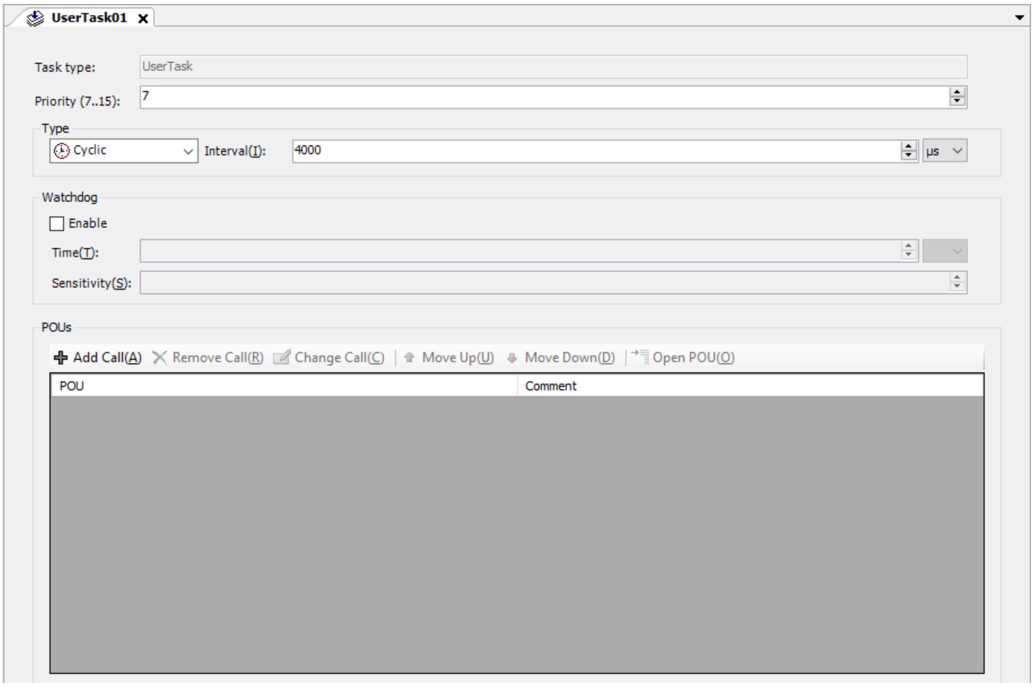
Name(N): UserTask01

2. Enter a task name in the Name field and click the [Add] button.

The task object will be added.



3. Add a POU object to the UserTask.
The added POU object will be executed as a UserTask.
Double-clicking the object of the added UserTask displays a task configuration window where task settings can be configured.


UserTask01

Task type: UserTask

Priority (7..15): 7

Type

Cyclic Interval(I): 4000 µs

Watchdog

□ Enable

Time(T):

Sensitivity(S):

POUs

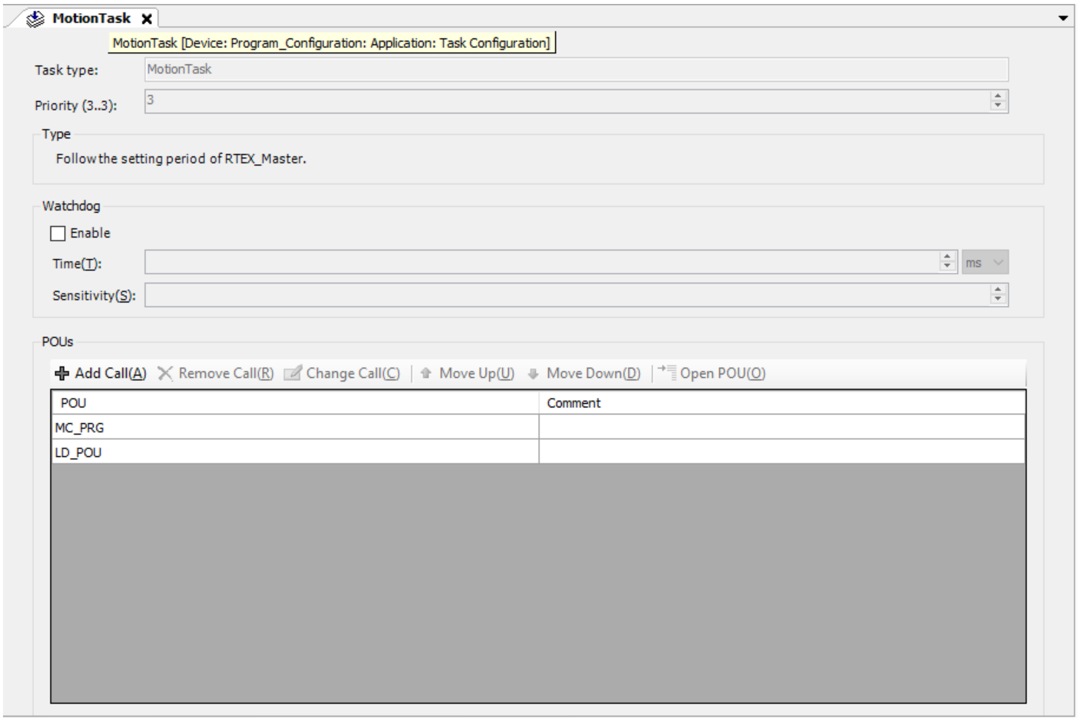
POU	Comment

6.6.3 Task Configuration Window

The task configuration window allows the user to configure settings related to task execution, such as execution priorities, execution methods, execution intervals, and watchdog timer.

For the procedure for displaying the task configuration window, refer to "6.6.1 Adding Programs".

Example: Task configuration window for task "MotionTask"



MotionTask [Device: Program_Configuration: Application: Task Configuration]

Task type: MotionTask

Priority (3..3): 3

Type

Follow the setting period of RTEX_Master.

Watchdog

□ Enable

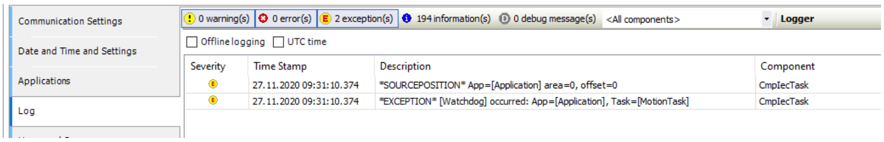
Time(T): ms

Sensitivity(S):

POUs

POU	Comment
MC_PRG	
LD_POU	

Item	Description
Task type	This field displays the type of the task. It displays MotionTask, UserTask, or SystemTask.
Priority	This field displays the priority of the task. The smaller the value, the higher the priority. For UserTask, the priority can be set between 7 and 15 inclusive.
Type	This field specifies the processing method for the task. For UserTask, one of the following two types can be selected. ● Cyclic: Processes the task at intervals. Specify a task interval in the "Period" field. ● Event: Starts task processing as soon as a rising edge of the global variable specified in the "Event" field is detected
Watchdog	If the "Enable" check box is selected, when the program execution time exceeds the preset time, the task will enter an error state and comes to a halt. The stop conditions are divided into the following two cases: ● Case where the program execution time exceeds the number of times specified in "Sensitivity" or the time specified in "Time"

Item	Description
	● Case where the program execution time exceeds "Sensitivity"×"Time" during a single cycle (Example: If "Sensitivity" is set to "3" and "Time" is set to "t#20ms", when the execution time exceeds 60 ms during a single cycle, the task will stop.) If the watchdog timer causes the task to stop, the event will be recorded in the "Log" tab of the device editor.  Communication Settings Date and Time and Settings Applications Log 0 warning(s) 0 error(s) 2 exception(s) 194 information(s) 0 debug message(s) <All components> Logger □ Offline logging □ UTC time Severity | Time Stamp | Description | Component E | 27.11.2020 09:31:10.374 | *SOURCEPOSITION* App=[Application] area=0, offset=0 | CmpIecTask E | 27.11.2020 09:31:10.374 | *EXCEPTION* [Watchdog] occurred: App=[Application], Task=[MotionTask] | CmpIecTask

6.7 Build

When a created program is subject to a build process, objects in the application are compiled. If code generation is executed after the build process is executed, an application to be downloaded to the GM1 controller will be generated.

6.7.1 Build

The syntax of all objects is verified when the build process is executed for the first time.

The syntax of only differences is verified when the build process is executed a second time and thereafter. No application code will be generated.

6.7.2 Rebuild

Verifies the syntax of all objects again.

As is the case with build, no application code will be generated.

1 2 Procedure

1. From the menu bar, select **Build**>**Rebuild**.

   The syntax of all objects will be verified. If an error or warning occurs, an error or warning message will be displayed in the message view.

   Check the message displayed in the message view and correct the program as necessary.

   After correcting the program, execute rebuild again.

6.7.3 Code Generation

The GM1 controller generates codes (application codes) to be executed when the application starts.

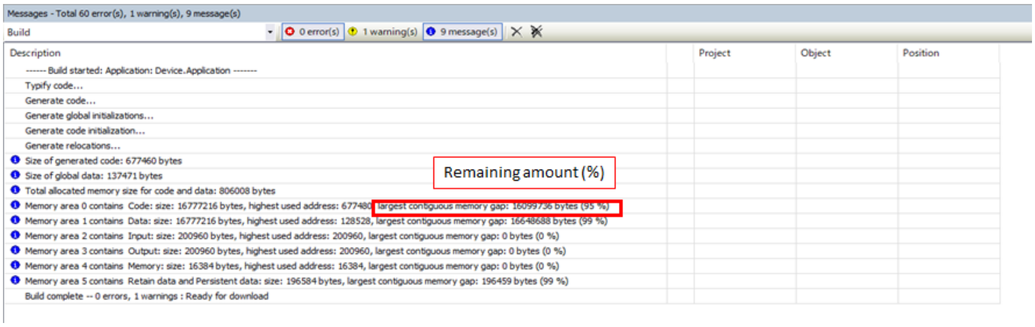
Displays the remaining program capacity and variable capacity during code generation.

1 2 Procedure

1. From the menu bar, select **Build**>**Code Generation**.

Tests will be executed to check memory allocations, data types, and library availability and code size (in bytes), data size (in bytes), allocated memory contents, and most frequently used address (in bytes) will be displayed in the Messages view.

Example: Messages view displayed when code generation is completed correctly



```
------ Build started: Application: Device.Application -------
Typify code...
Generate code...
Generate global initializations...
Generate code initialization...
Generate relocations...
Size of generated code: 677460 bytes
Size of global data: 137471 bytes
Total allocated memory size for code and data: 806008 bytes
Memory area 0 contains  Code: size: 16777216 bytes, highest used address: 677480 largest contiguous memory gap: 16099736 bytes (95 %)
Memory area 1 contains  Data: size: 16777216 bytes, highest used address: 128528, largest contiguous memory gap: 16648688 bytes (99 %)
Memory area 2 contains  Input: size: 200960 bytes, highest used address: 200960, largest contiguous memory gap: 0 bytes (0 %)
Memory area 3 contains  Output: size: 200960 bytes, highest used address: 200960, largest contiguous memory gap: 0 bytes (0 %)
Memory area 4 contains  Memory: size: 16384 bytes, highest used address: 16384, largest contiguous memory gap: 0 bytes (0 %)
Memory area 5 contains  Retain data and Persistent data: size: 196584 bytes, largest contiguous memory gap: 196459 bytes (99 %)
Build complete -- 0 errors, 1 warnings : Ready for download
```

Remaining amount (%)

Memory area 0	Program capacity	For the maximum capacity, refer to the GM1 Series Reference Manual (Hardware).
Memory area 1	Variable capacity (non-hold)	For the maximum capacity, refer to the GM1 Series Reference Manual (Hardware).
Memory area 2	Input variable	Automatic assignment during code generation
Memory area 3	Output variable	Automatic assignment during code generation
Memory area 4	Internal memory	Automatic assignment during code generation
Memory area 5	Variable capacity (hold)	For the maximum capacity, refer to the GM1 Series Reference Manual (Hardware).

Info.

- After application codes are generated, if some codes are changed and code generation is executed again, differential compilation will reallocate memory to only newly added and changed blocks and variables. If memory fragmentation occurs as a result of memory reallocation, the amount of memory that can be actually used will be reduced. To eliminate memory fragmentation, you must perform the procedure described in "6.7.4 Clean".

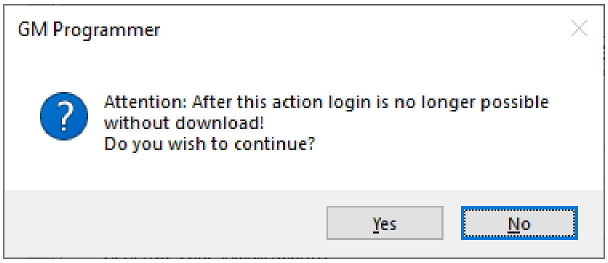
6.7.4 Clean

Deletes application build information.

1 2 Procedure

1. From the menu bar, select **Build**>**Clean**.

   If "Clean" is executed, the following confirmation message will be displayed.



GM Programmer

Attention: After this action login is no longer possible without download!
Do you wish to continue?

Clicking the [Yes] button executes "Clean".

Info.

- If "Clean" is executed, online change can no longer be performed. Therefore, to log in to the GM1 controller again, you must download the applications.
- If you copy a program object (POU object), execute “Clean” for the copied POU object.

Note

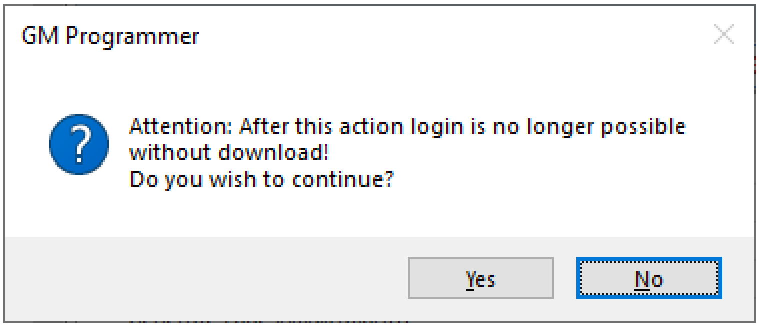
- Even if "Clean" is executed, the variables registered in the global persistent variable list will not be initialized. Other variables and persistent variables will be initialized.

6.7.5 Clean All

Deletes all application build information in the same way as "Clean".

1 2 Procedure

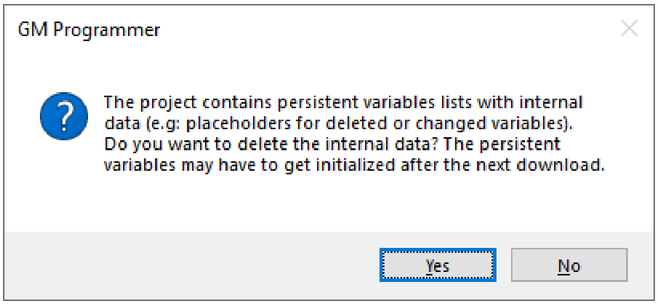
1. From the menu bar, select **Build**>**Clean All**.
   If "Clean All" is executed, the following confirmation message will be displayed.



GM Programmer

Attention: After this action login is no longer possible without download!
Do you wish to continue?

2. Clicking the [Yes] button executes "Clean All".
   If variables have been registered in the persistent variable list, the following confirmation message will be displayed.



GM Programmer

The project contains persistent variables lists with internal data (e.g: placeholders for deleted or changed variables).
Do you want to delete the internal data? The persistent variables may have to get initialized after the next download.

3. If you delete internal data, click the [Yes] button.
   If you do not delete it, click the [No] button.

Note

- If you have made changes such as "Add new variables to the top of persistent variable list", note that actual variable values could become different values because the memory areas for retaining variables have become misaligned.

6.8 Program Creation Support Functions

This section explains the program creation support functions used to create programs.

6.8.1 Bookmark

The bookmark function allows the cursor to move to bookmarked locations.

The bookmark function can be used in all programs other than SFC programs.

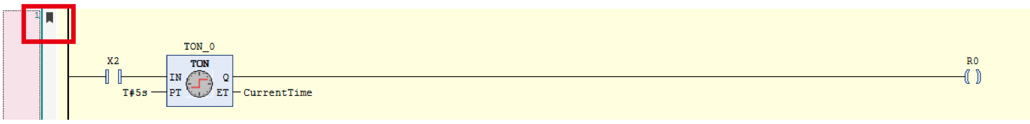
This section explains the procedures for setting bookmarks and moving the cursor between bookmarks, using an LD program as an example.

1 2 Procedure

1. Select the network (circuit) where you want to set a bookmark. From the menu bar, select **Edit**>**Bookmarks**>**Toggle Bookmark**, or click the <Ctrl> key + <F12> key simultaneously.

   A bookmark (🔖) will be set in the selected network (circuit).

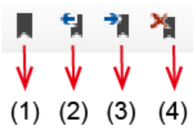
   Performing the above operation again clears (deletes) the set bookmark.



2. To move to the next bookmark, press the <F12> key. To move to the previous bookmark, press the <Shift> key + <F12> key simultaneously.

Info.

- To clear (delete) all the set bookmarks, from the menu bar, select **Edit**>**Bookmarks**>**Clear All Bookmarks**.
- You can also perform bookmark operations by clicking appropriate icons on the toolbar.



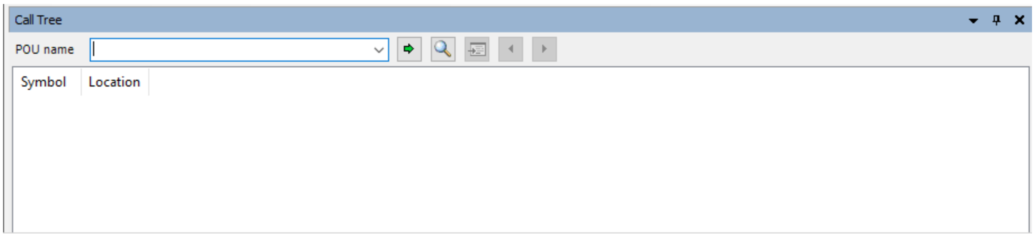
No.	Item
(1)	Toggle Bookmark
(2)	Previous Bookmark
(3)	Next Bookmark
(4)	Clear All Bookmarks

6.8.2 Call Tree View

Opening the Call Tree view enables the user to search the callers and callees of blocks such as functions, function blocks, or POU.

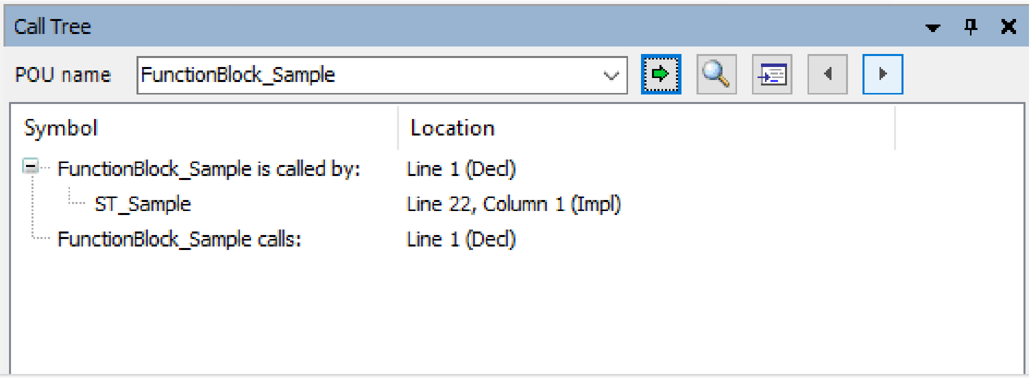
1 2 Procedure

1. From the menu bar, select **View**>**Call Tree**.
   The Call Tree view will be displayed.



2. Enter a block to be searched in the POU Name field and press the <Enter> key.
   The caller and callee of the block will be displayed in tree structure.
   Double-clicking in any search result line displays the corresponding window in the main pane.

Example: Searching the caller and callee of FunctionBlock_Sample



POU name: FunctionBlock_Sample

Symbol	Location
FunctionBlock_Sample is called by:	Line 1 (Decl)
ST_Sample	Line 22, Column 1 (Impl)
FunctionBlock_Sample calls:	Line 1 (Decl)

Info.

- Pressing the <F4> key moves the cursor to the next search result line. Pressing the <Shift> key + <F4> key simultaneously moves the cursor to the previous search result line.
- Position the cursor on a block in the implementation section. From the menu bar, select **Edit**>**Browse**>**Browse Call Tree**. The Call Tree view will be displayed with a search conducted for the block at the cursor position.

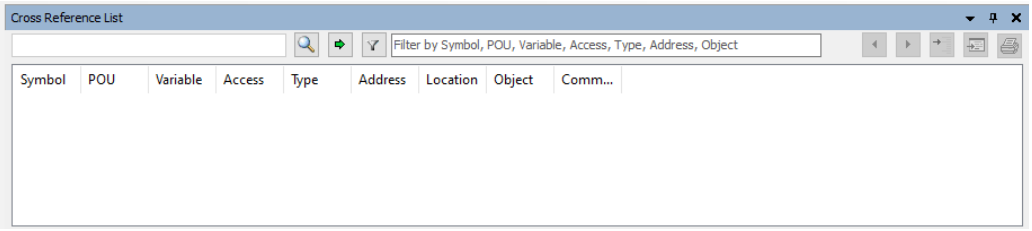
6.8.3 Cross reference List View

Opening the Cross reference List view allows the user to search the locations of variables and other elements used within the entire project.

1 2 Procedure

1. From the menu bar, select **View**>**Cross Reference List**.

The Cross reference List view will be displayed.


Cross Reference List
Filter by Symbol, POU, Variable, Access, Type, Address, Object

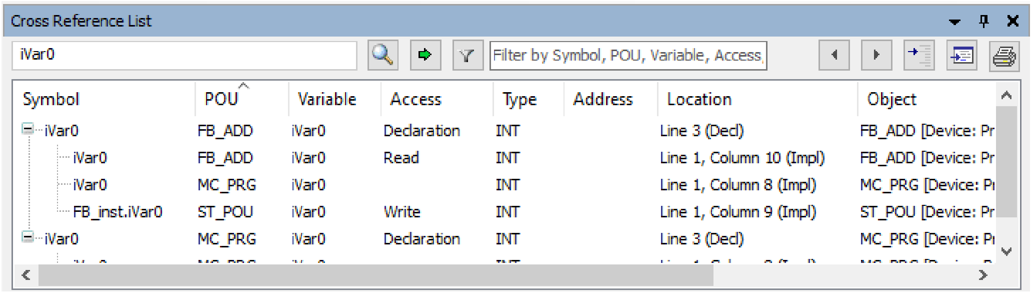
Symbol	POU	Variable	Access	Type	Address	Location	Object	Comm...

2. Enter a variable name (or another element name) to be searched in the search field and press the <Enter> key.

The locations of the variable (or element) used will be displayed in list form.

Double-clicking in any search result line displays the corresponding section in the main pane.

Example: When variable iVar0 is entered



Symbol	POU	Variable	Access	Type	Address	Location	Object
iVar0	FB_ADD	iVar0	Declaration	INT		Line 3 (Decl)	FB_ADD [Device: Pr
iVar0	FB_ADD	iVar0	Read	INT		Line 1, Column 10 (Impl)	FB_ADD [Device: Pr
iVar0	MC_PRG	iVar0		INT		Line 1, Column 8 (Impl)	MC_PRG [Device: Pr
FB_inst.iVar0	ST_POU	iVar0	Write	INT		Line 1, Column 9 (Impl)	ST_POU [Device: Pr
iVar0	MC_PRG	iVar0	Declaration	INT		Line 3 (Decl)	MC_PRG [Device: Pr

Info.

- Pressing the <F4> key moves the cursor to the next search result line. Pressing the <Shift> key + <F4> key simultaneously moves the cursor to the previous search result line.
- For searches, you can use an asterisk (*) that represents any character string or a question mark (?) that represents a single character.
- Position the cursor on a variable in the implementation section. From the menu bar, select **Edit**>**Browse**>**Browse Cross Reference**. The Cross-reference view will be displayed with a search conducted for the variable at the cursor position.

6.8.4 Function Block Guidance

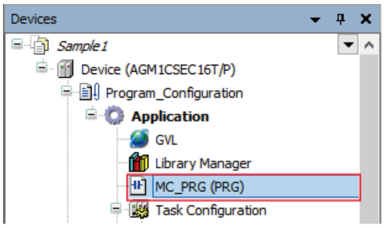
The Function Block Guidance allows the user to enter motion function blocks into a program.

The Function Block Guidance can be used in LD programs, ST programs, FBD programs, and CFC programs.

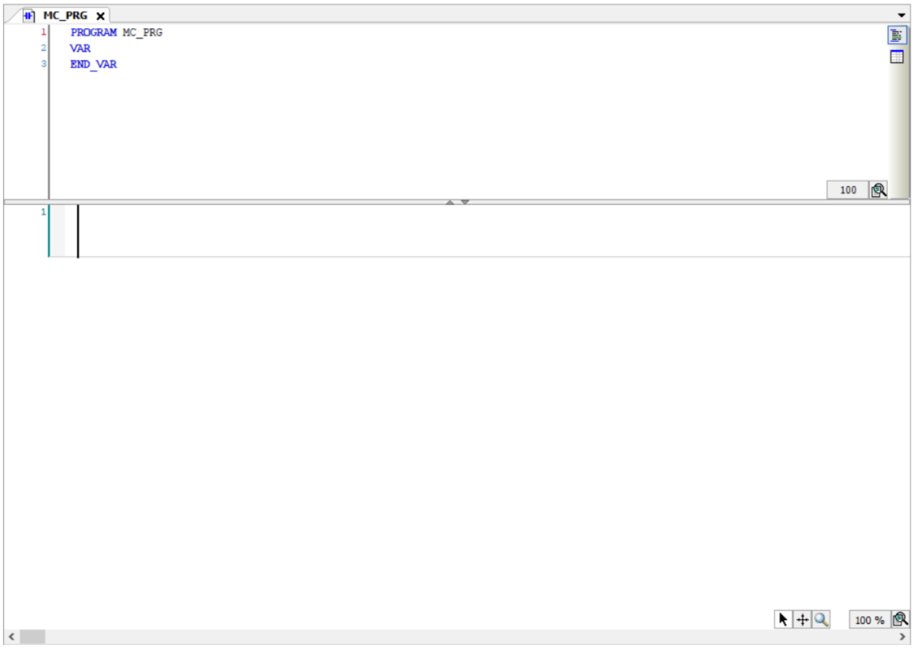
The following procedure is explained, using an example in which "Power" is searched and function block "MC_Power" is inserted into an LD program.

1 2 Procedure

1. Double-click the [MC_PRG(PRG)] object in the navigator pane.

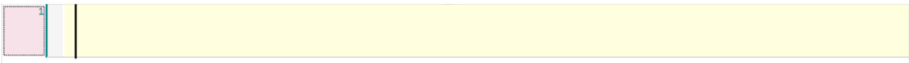


The "MC_PRG" window will be displayed.

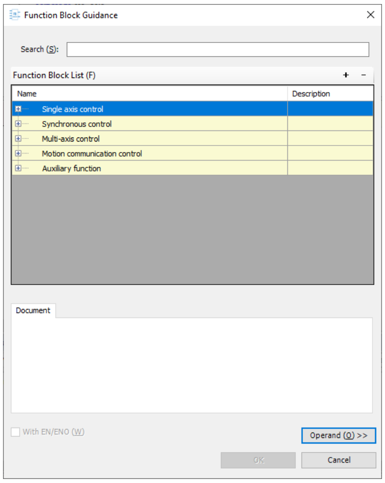


```
PROGRAM MC_PRG
VAR
END_VAR
```

2. Select the network in the implementation section and, from the menu bar, select **Edit**>**Function Block Guidance**.



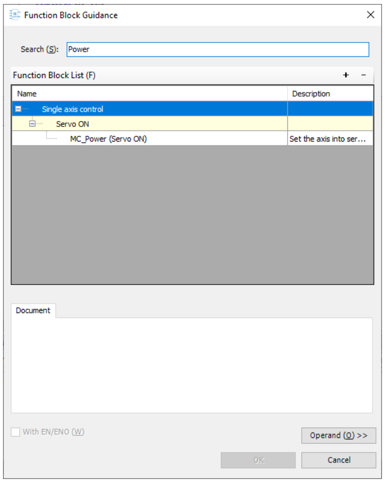
The "Function Block Guidance" dialog box will be displayed.



Name	Description
⊞ Single axis control	
⊞ Synchronous control	
⊞ Multi-axis control	
⊞ Motion communication control	
⊞ Auxiliary function	

3. Enter a character string in the Search field.

Function blocks related to the entered character string will be displayed in the Function Block List table.

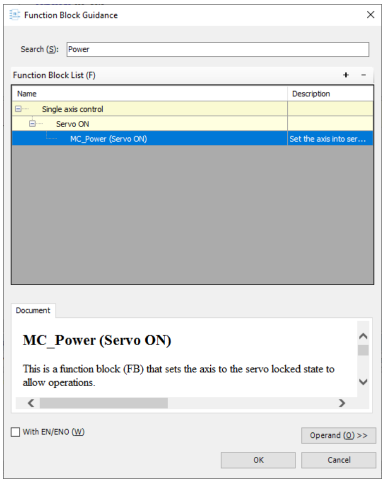


Search (S): Power

Name	Description
⊟ Single axis control	
⊟ Servo ON	
MC_Power (Servo ON)	Set the axis into ser…

4. Select a function block that you want to insert into the program.

A description of the selected instruction will be displayed in the Document tab pane.



Search (S): Power

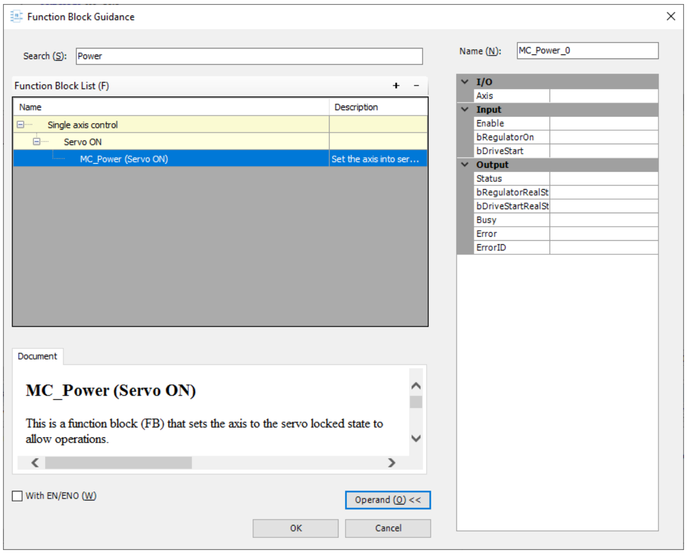
Name	Description
Single axis control	
Servo ON	
MC_Power (Servo ON)	Set the axis into ser…

MC_Power (Servo ON)

This is a function block (FB) that sets the axis to the servo locked state to allow operations.

5. Click the [Operand] button.

The instance and operand input fields will be displayed.



Search (S): Power

Name (N): MC_Power_0

Name	Description
Single axis control	
Servo ON	
MC_Power (Servo ON)	Set the axis into ser…

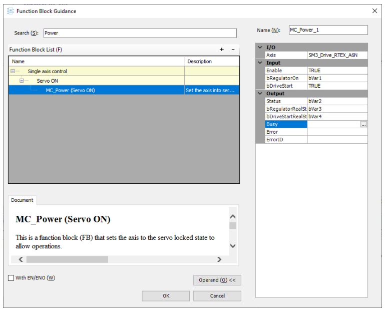
I/O	
Axis	
Input	
Enable	
bRegulatorOn	
bDriveStart	
Output	
Status	
bRegulatorRealSt	
bDriveStartRealSt	
Busy	
Error	
ErrorID	

MC_Power (Servo ON)

This is a function block (FB) that sets the axis to the servo locked state to allow operations.

6. Enter an instance name in the Name field and values in each operand field.

If the operand for which a value has been entered is a variable that has not been declared, the "Auto Declare" dialog box will be displayed, so that the variable can be declared.



Search (S): Power

Name (N): MC_Power_1

Name	Description
Single axis control	
Servo ON	
MC_Power (Servo ON)	Set the axis into ser…

I/O	
Axis	SM3_Drive_RTEX_A6N
Input	
Enable	TRUE
bRegulatorOn	bVar1
bDriveStart	TRUE
Output	
Status	bVar2
bRegulatorRealSt	bVar3
bDriveStartRealSt	bVar4
Busy	…
Error	
ErrorID	

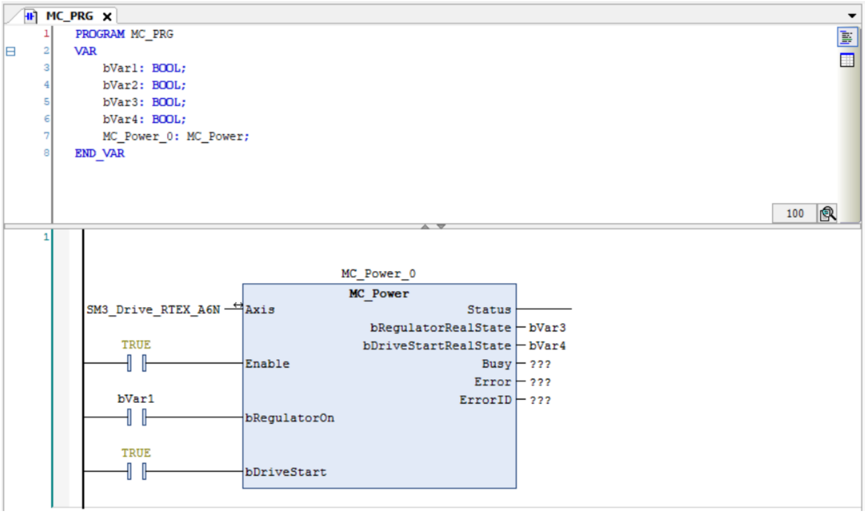
MC_Power (Servo ON)

This is a function block (FB) that sets the axis to the servo locked state to allow operations.

7. Click the [OK] button.

If an instance name has not been declared, the "Auto Declare" dialog box will be displayed, so that an instance can be declared.

The function block will be inserted into the program.



```
PROGRAM MC_PRG
VAR
    bVar1: BOOL;
    bVar2: BOOL;
    bVar3: BOOL;
    bVar4: BOOL;
    MC_Power_0: MC_Power;
END_VAR
```

Info.

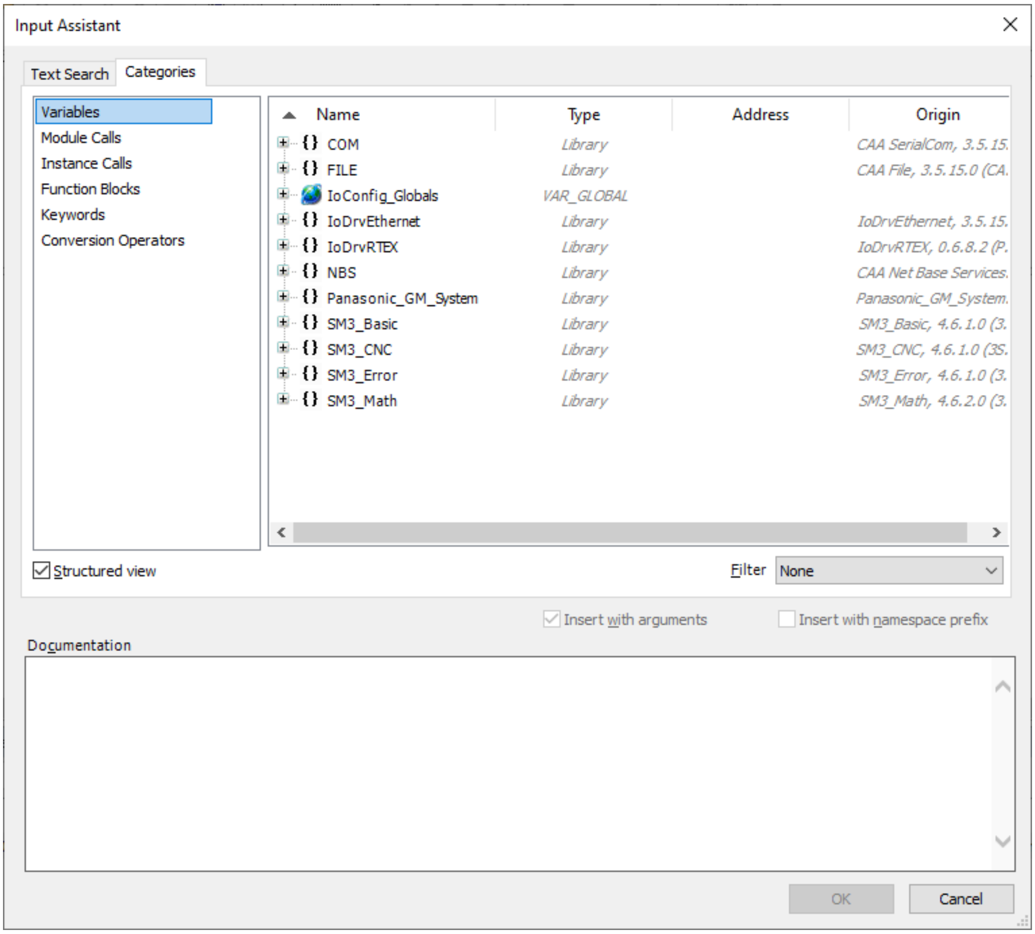
- To insert a function block into a CFC program, insert a box first and, with the box selected, start the Function Block Guidance.
- If the "With EN / ENO" check box is selected, a function block with EN input and ENO output will be inserted. When the value of EN input is TRUE, the function block is executed. Similarly, when the value of EN input is FALSE, the function block is not executed. The same value as EN input is output to ENO output.
- The Function Block Guidance can also be started using the following operations:
  - Click the "Function Block Guidance" icon on the toolbar.
  - Press the <Alt> key + <F2> key simultaneously.
  - Select the network in the implementation section, and then right-click and select "Function Block Guidance" from the context-sensitive menu that is displayed.

6.8.5 Input Assistant Function

By using the Input Assistant Function, variables, function blocks, operators, types, and other elements that can be inserted in the cursor position can be selected from categories.

1 2 Procedure

1. Move the cursor to the position where you want to insert a desired element and press the <F2> key.
   The "Input Assistant" dialog box will be displayed.
   Select a desired element from the "Categories" tab pane or the "Text Search" tab pane.



Name	Type	Address	Origin
COM	Library		CAA SerialCom, 3.5.15.
FILE	Library		CAA File, 3.5.15.0 (CA.
IoConfig_Globals	VAR_GLOBAL		
IoDrvEthernet	Library		IoDrvEthernet, 3.5.15.
IoDrvRTEX	Library		IoDrvRTEX, 0.6.8.2 (P.
NBS	Library		CAA Net Base Services.
Panasonic_GM_System	Library		Panasonic_GM_System.
SM3_Basic	Library		SM3_Basic, 4.6.1.0 (3.
SM3_CNC	Library		SM3_CNC, 4.6.1.0 (3S.
SM3_Error	Library		SM3_Error, 4.6.1.0 (3.
SM3_Math	Library		SM3_Math, 4.6.2.0 (3.

☑ Structured view

Filter: None

☑ Insert with arguments

□ Insert with namespace prefix

2. Click the [OK] button.
   The selected element will be inserted.

Info.

- You can also open the "Input Assistant" dialog box by selecting **Edit**>**Input Assistant** from the menu bar.

6.8.6 Argument / Variable Input Support (Component List)

This section explains the functions (component list) that support input of arguments and other data for function blocks during program creation.

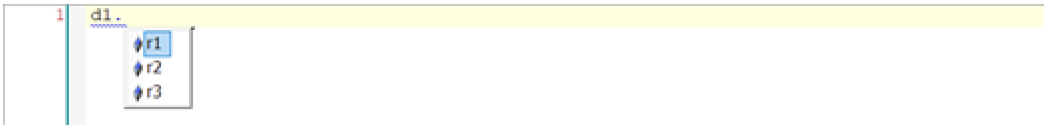
The following three input support functions are available.

■ Displaying candidates for arguments that can be entered

Entering a dot (.) after a name such as a function block name or structure name displays a list of candidates for arguments that can be entered.

To enable this function, in the Options window, select **SmartCoding**>**List components after typing a dot(.)**.

Example: Displaying a list of members of structure variable stVar after structure variable stVar and a dot are entered

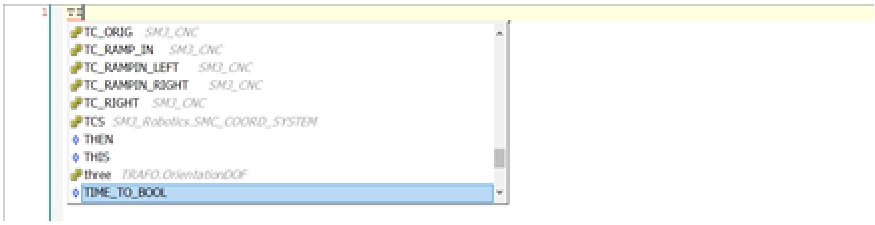


```
d1.
```

■ Displaying candidates for variables or other components starting with the entered character string

Entering any character string and then pressing the <Ctrl> key + <Space> key simultaneously displays a list of elements that can be inserted.

Example: Entering "TI" and pressing the <Ctrl> key + <Space> key simultaneously selects and displays the positions of variables or other components starting with "TI"

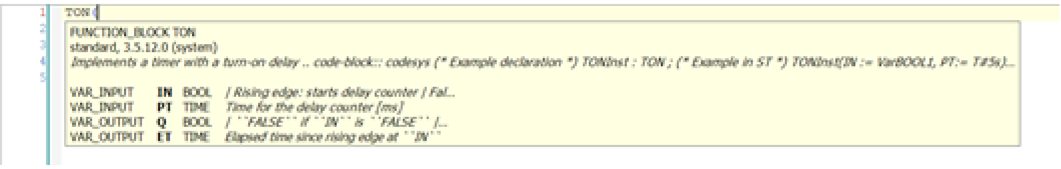


```
TI
```

■ Displaying a description of function block or function

Entering a function block name (or some other name) followed by a left parenthesis displays a description of the function block in a pop-up window.

To open the closed pop-up window again, press the <Ctrl> key + <Shift> key + <Space> key simultaneously.



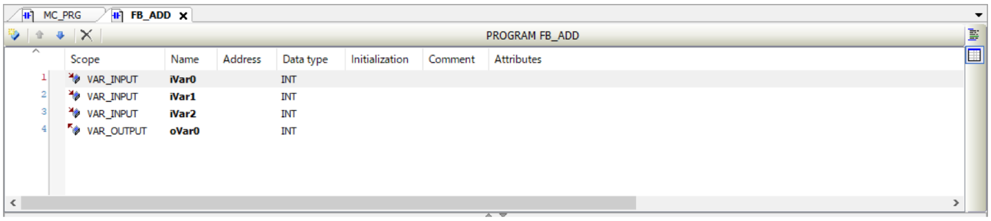
```
TON(
```

```
FUNCTION_BLOCK TON
standard, 3.5.12.0 (system)
Implements a timer with a turn-on delay .. code-block:: codesys (* Example declaration *) TONInst : TON ; (* Example in ST *) TONInst(IN := VarBOOL1, PT:= T#5s)...

VAR_INPUT    IN  BOOL  | Rising edge: starts delay counter | Fal...
VAR_INPUT    PT  TIME  Time for the delay counter [ms]
VAR_OUTPUT   Q   BOOL  | ``FALSE`` if ``IN`` is ``FALSE`` |...
VAR_OUTPUT   ET  TIME  Elapsed time since rising edge at ``IN``
```

6.8.7 Global Renaming (Refactoring)

When a POU object name in the navigator pane or a variable name in the declaration section is changed, the sections where the changed name is used are displayed, so that the name can be changed collectively (this function is called "refactoring").

This section explains the procedure for changing the variable name of input variable " iVar2" in function block "FB_ADD" and using the refactoring function to change the variable name in the sections where the variable is called.



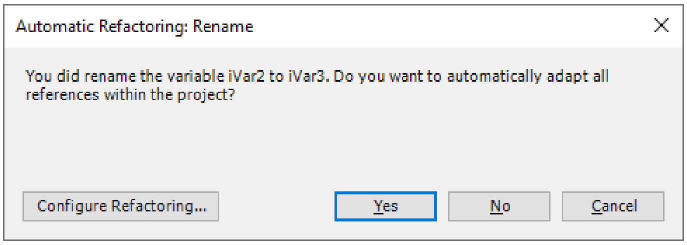
PROGRAM FB_ADD

	Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR_INPUT	iVar0		INT			
2	VAR_INPUT	iVar1		INT			
3	VAR_INPUT	iVar2		INT			
4	VAR_OUTPUT	oVar0		INT			

1 2 Procedure

1. Change the name of variable "iVar2" in the declaration section of the function block to "iVar3".

   The "Automatic Refactoring" dialog box will be displayed.

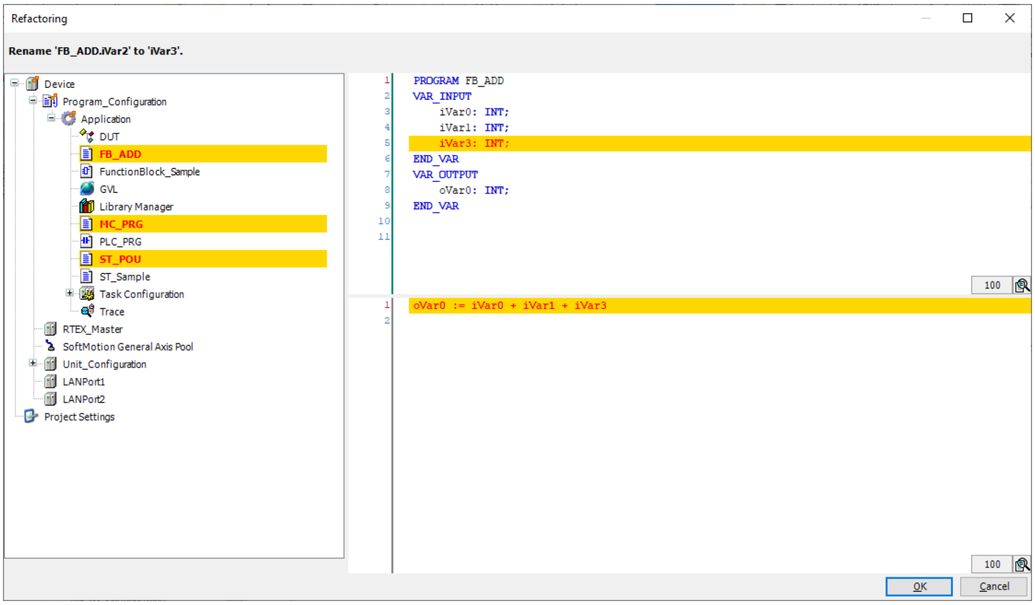

Automatic Refactoring: Rename

You did rename the variable iVar2 to iVar3. Do you want to automatically adapt all references within the project?

2. Click [Yes].

   The "Refactoring" dialog box will be displayed.

   Each section where the changed variable is used is displayed in red and the background of the section is displayed in yellow.

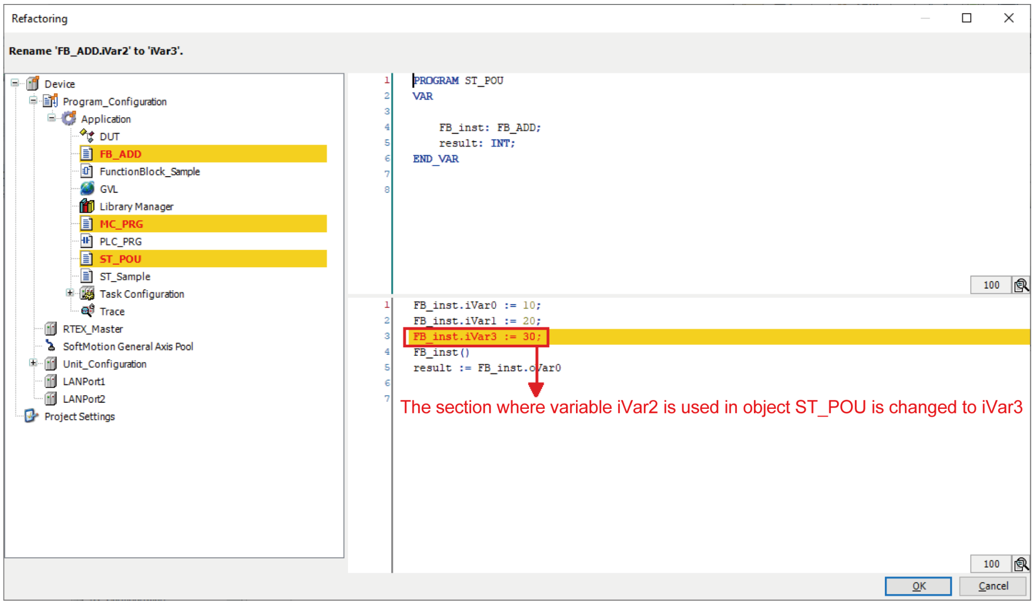


```
PROGRAM FB_ADD
VAR_INPUT
    iVar0: INT;
    iVar1: INT;
    iVar3: INT;
END_VAR
VAR_OUTPUT
    oVar0: INT;
END_VAR
```

```
oVar0 := iVar0 + iVar1 + iVar3
```

By clicking an object in the navigator pane, you can check the change details of the object. At this stage, the changes have not been reflected yet.

When ST_POU object is selected:



```
PROGRAM ST_POU
VAR

    FB_inst: FB_ADD;
    result: INT;
END_VAR
```

```
FB_inst.iVar0 := 10;
FB_inst.iVar1 := 20;
FB_inst.iVar3 := 30;
FB_inst()
result := FB_inst.oVar0
```

The section where variable iVar2 is used in object ST_POU is changed to iVar3

3. Click the [OK] button.
All changes will be reflected.

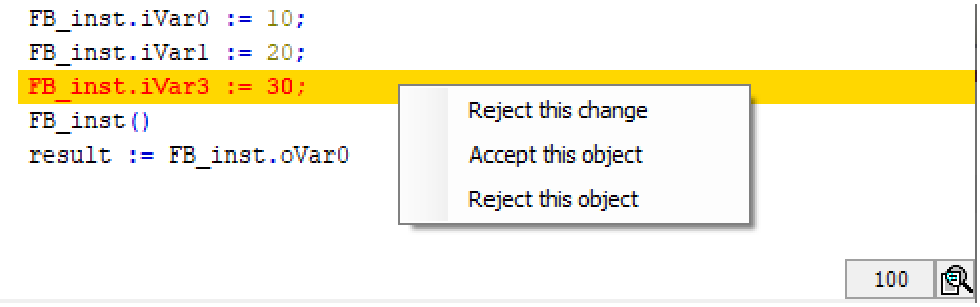
Info.

- When the sections where the changed variable is used are displayed by the refactoring function, you can individually select whether to reflect the change. Right-click in the sections where the changed variable is used and select whether to reflect the change, from the context-sensitive menu that is displayed.

  "Reject this change": Does not reflect the change in the selected section

  "Accept this object": Reflects the change in the object

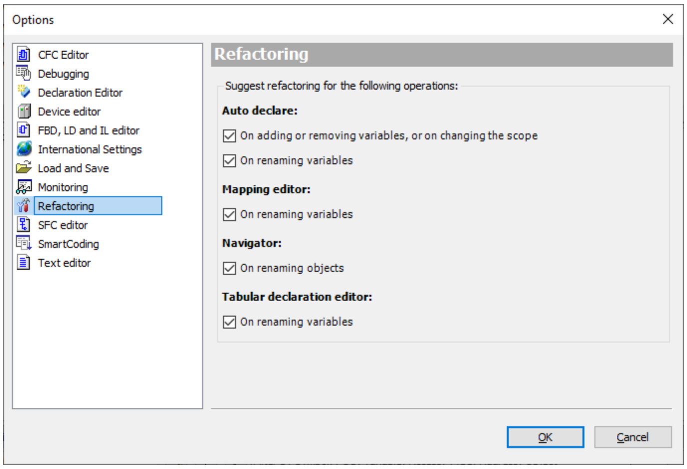
  "Reject this object": Does not reflect the change in the object



```
FB_inst.iVar0 := 10;
FB_inst.iVar1 := 20;
FB_inst.iVar3 := 30;
FB_inst()
result := FB_inst.oVar0
```

- In the Options window, you can specify the situations where the refactoring function is enabled.

  Open the Options window (by selecting **Tools**>**Options**), select the "Refactoring" category, and specify the situations where the refactoring function is enabled.



Suggest refactoring for the following operations:

Auto declare:

☑ On adding or removing variables, or on changing the scope

☑ On renaming variables

Mapping editor:

☑ On renaming variables

Navigator:

☑ On renaming objects

Tabular declaration editor:

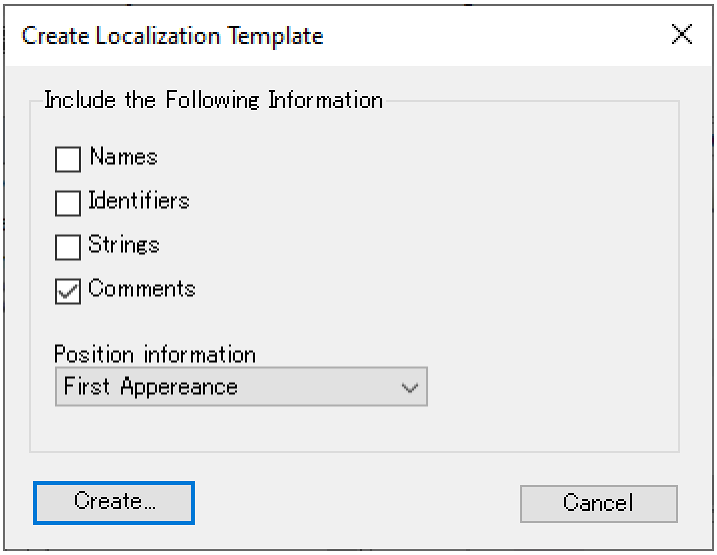
☑ On renaming variables

6.8.8 Displaying Programs in Multiple Languages (Project Localization)

The project localization function allows the user to translate and register comments, titles, and other information in the program to display the translated content in the program window.

1 2 Procedure

1. From the menu bar, select **Project**>**Project Localization**>**Create Localization Template**.
   The "Create Localization Template" dialog box will be displayed.
2. Select information to be translated.
   To translate comments and titles in a program, select the "Comments" check box.
   To add location information to a template, select "First Appearance" or "All" in the Location information drop-down list.



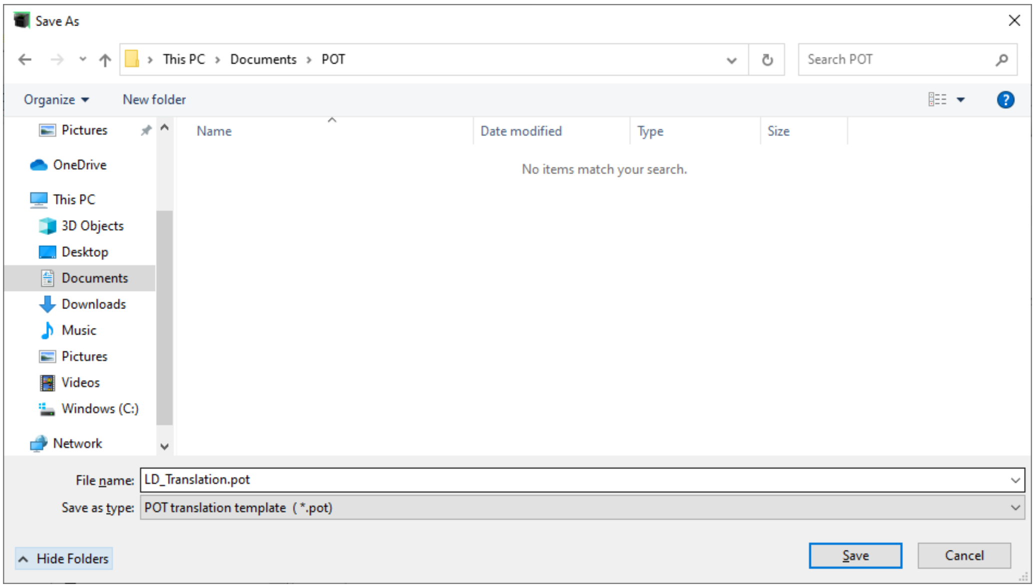
Create Localization Template

Include the Following Information

□ Names
□ Identifiers
□ Strings
☑ Comments

Position information: First Appereance

3. Click the [Create] button.
   The "Save As" dialog box will be displayed.

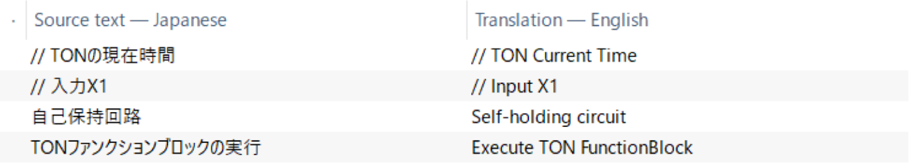


File name: LD_Translation.pot

Save as type: POT translation template (*.pot)

4. Enter a file name and click the Save button.
A POT translation template file (".pot") will be created.

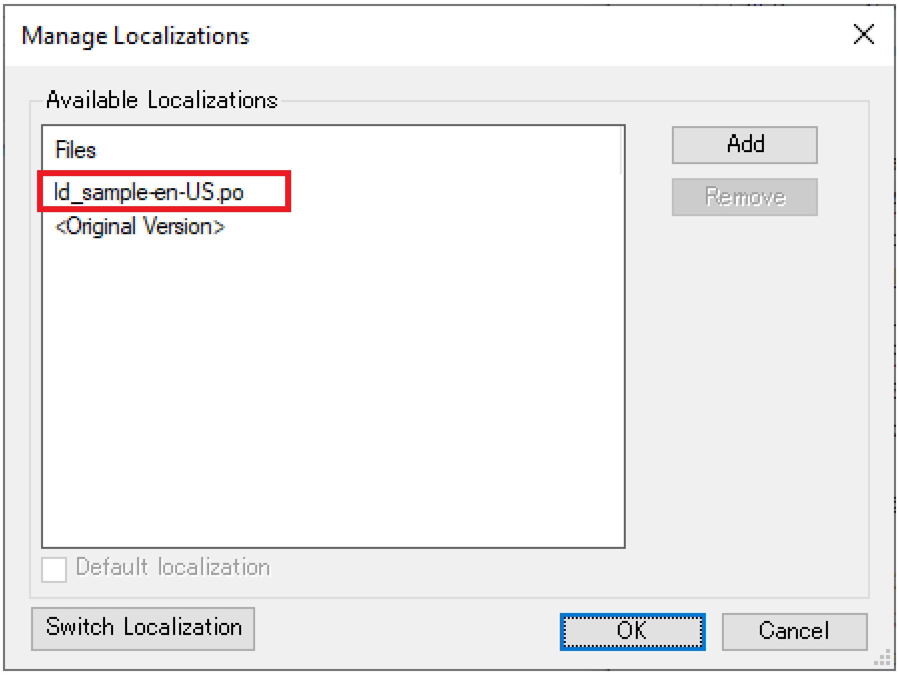
5. For localization, use an editor such as PoEditor to enter translations.
Create a localization file (".po").



Source text — Japanese	Translation — English
// TONの現在時間	// TON Current Time
// 入力X1	// Input X1
自己保持回路	Self-holding circuit
TONファンクションブロックの実行	Execute TON FunctionBlock

6. From the menu bar, select **Project**>**Project Localization**>**Manage Localizations**.
The "Manage Localizations" dialog box will be displayed.

7. Click the [Add] button.
The "Open Localization File" dialog box will be displayed. Select a localization file (".po") that has been created.
The added localization file will be displayed in the "Available Localizations" area.



8. Click the [OK] button.

The "Manage Localizations" dialog box will be closed.

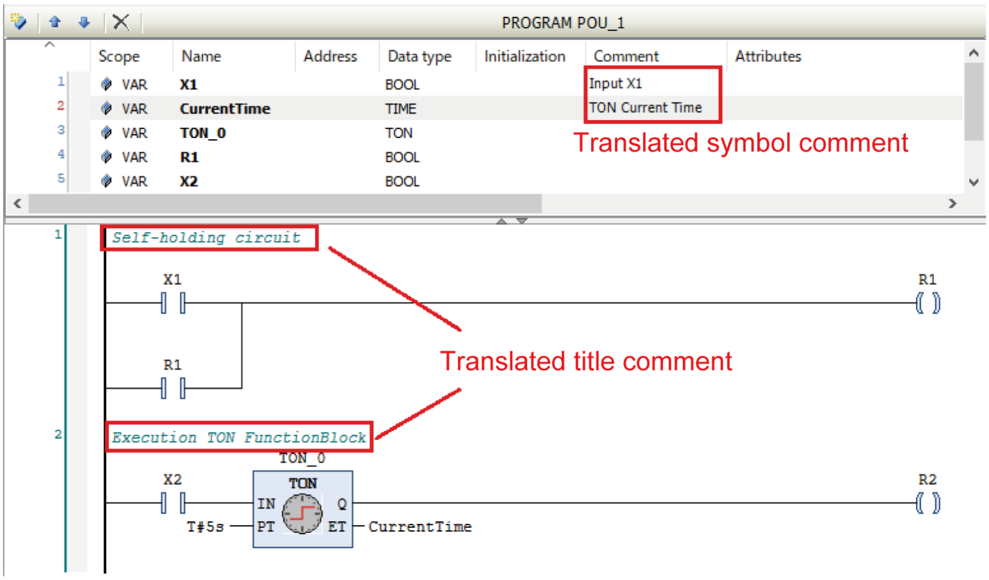
This completes the localization file creation procedure.

Next, the procedure for switching the display is explained below.

9. From the menu bar, select **Project**>**Project Localization**>**Switch Localization**.

Symbol comments and title comments will be displayed according to the translations in the added localization file.

To return the display to its original state, select "Switch Localization" again.



	Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR	X1		BOOL		Input X1	
2	VAR	CurrentTime		TIME		TON Current Time	
3	VAR	TON_0		TON			
4	VAR	R1		BOOL			
5	VAR	X2		BOOL			

Translated symbol comment

Translated title comment

Info.

- You can also switch the localization file by clicking on the toolbar.

(MEMO)

7 Entering programs in each Programming language and Variable

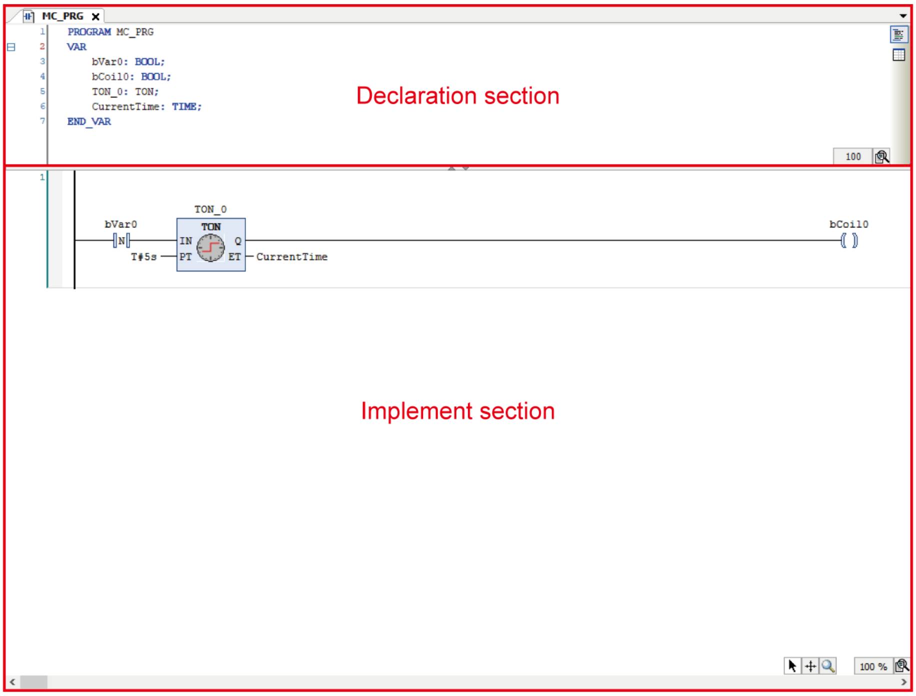
7.1 Programming in Ladder Diagram (LD)

This section explains how to create programs (LD programs) in Ladder Diagram compliant with IEC 61131-3, the international standard for PLC programming languages.

To create LD programs, POU objects for LD programs are required. Set the object setup language to Ladder Diagram (LD).

7.1.1 Inserting Contacts, Coils, and Function Blocks

This section explains how to create an LD program that consists of the normally open contact, coil, and function block TON shown below.

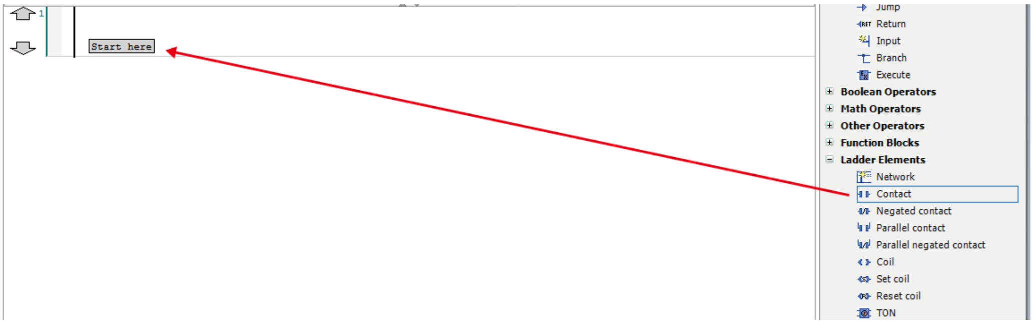


```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
    bCoil0: BOOL;
    TON_0: TON;
    CurrentTime: TIME;
END_VAR
```

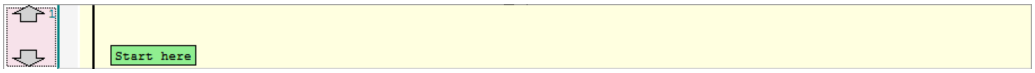
1 2 Procedure

1. In Toolbox, select **Ladder elements**>**Contact** and drag the mouse until "Start here" is displayed in the main pane.

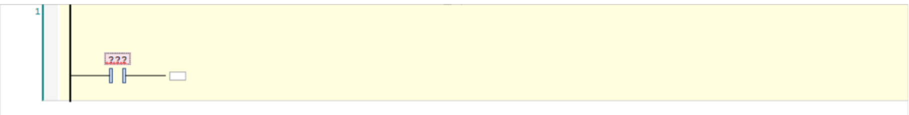
   "Start here" will be displayed in the implementation section.



When you drag the mouse until the position of "Start here" is reached, the display of "Start here" turns green.



When you stop dragging the mouse at the position of "Start here", a normally open contact will be placed in the network (circuit).

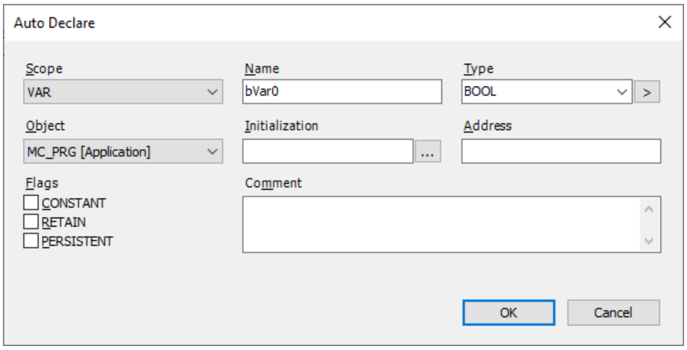


Info.

- You can also insert a normally open contact in the network (circuit) using the following procedure:
  - Right-click in the network (circuit) and select "Insert Contact" from the context-sensitive menu that is displayed.
  - Click the icon on the toolbar.
  - From the menu bar, select **FBD/LD/IL>Insert Contact**.
  - Press the <Ctrl> key + <k> key simultaneously.

2. Select "???" of the normally open contact and enter variable bVar0, and then press the <Enter> key.

The "Auto Declare" dialog box will be displayed.



Auto Declare

Scope: VAR

Name: bVar0

Type: BOOL

Object: MC_PRG [Application]

Initialization:

Address:

Flags

□ CONSTANT

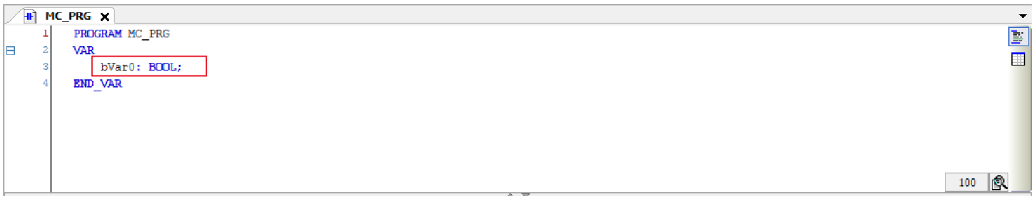
□ RETAIN

□ PERSISTENT

Comment:

3. Click the [OK] button.

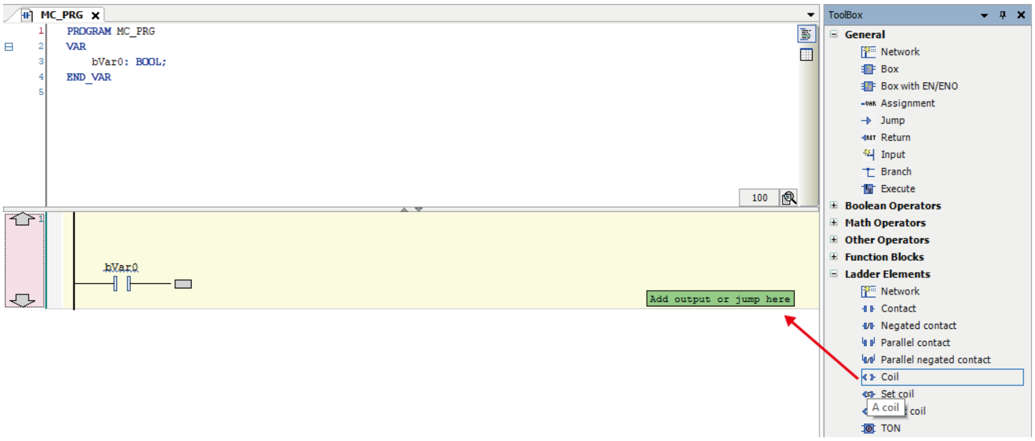
Variable bVar0 will be declared in the declaration section.



```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
END_VAR
```

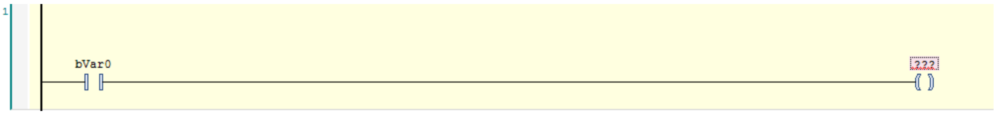
4. In Toolbox, select **Ladder elements**>**Coil** and drag the mouse until "Add output or jump here" is displayed in the main pane.

   When you drag the mouse until the position of "Add output or jump here" is reached, the display of "Add output or jump here" turns green.



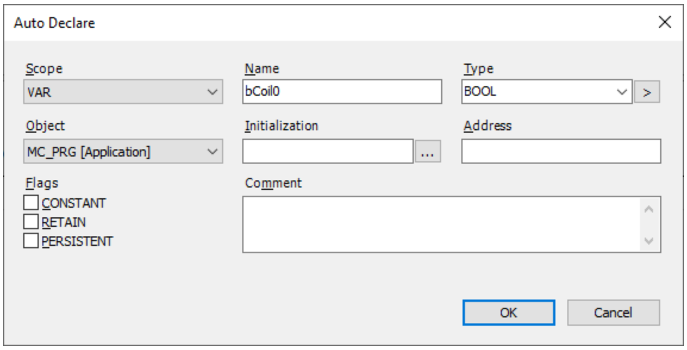
```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
END_VAR
```

When you stop dragging the mouse at the position of Add output or jump here, a coil will be placed in the network (circuit).



5. Select "???" of the coil and enter variable bCoil0, and then press the <Enter> key.

   The "Auto Declare" dialog box will be displayed.



Auto Declare

Scope: VAR

Name: bCoil0

Type: BOOL

Object: MC_PRG [Application]

Initialization:

Address:

Flags

□ CONSTANT

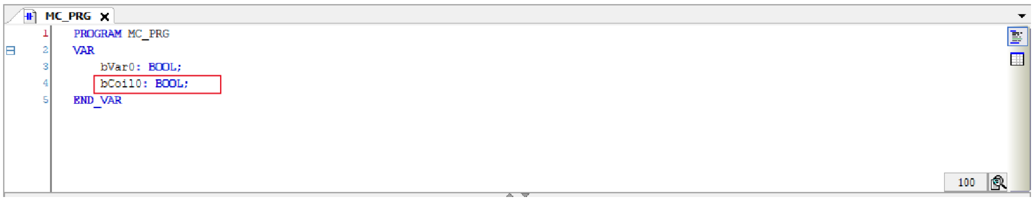
□ RETAIN

□ PERSISTENT

Comment:

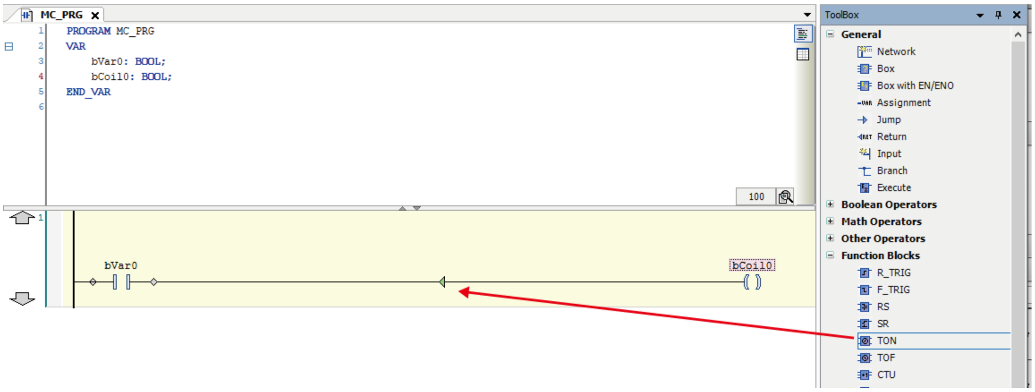
6. Click the [OK] button.

   Variable bCoil0 will be declared in the declaration section.



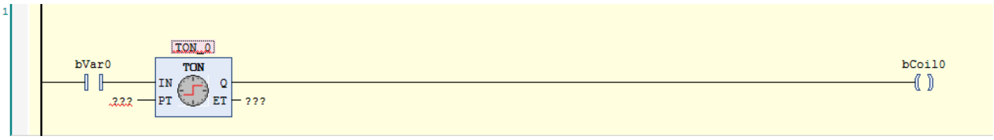
```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
    bCoil0: BOOL;
END_VAR
```

7. In Toolbox, select **Function block**>**TON** and drag the mouse until ◁ is displayed in the main pane.
Function block TON will be displayed.

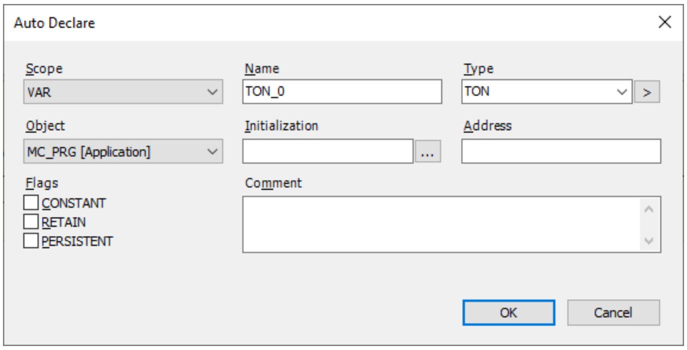


```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
    bCoil0: BOOL;
END_VAR
```

TON will be placed in the implementation section.



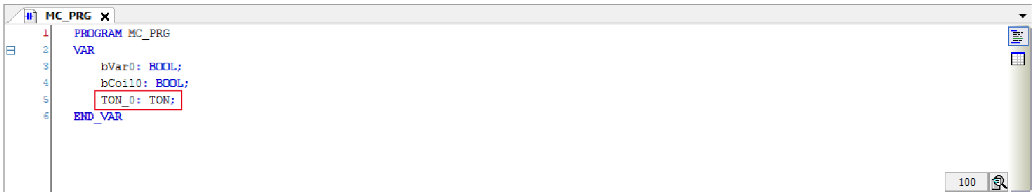
8. Select TON and enter a variable name, and then press the <Enter> key.
The "Auto Declare" dialog box for TON will be displayed.



Auto Declare

Scope: VAR
Name: TON_0
Type: TON
Object: MC_PRG [Application]
Initialization:
Address:
Flags
☐ CONSTANT
☐ RETAIN
☐ PERSISTENT
Comment:

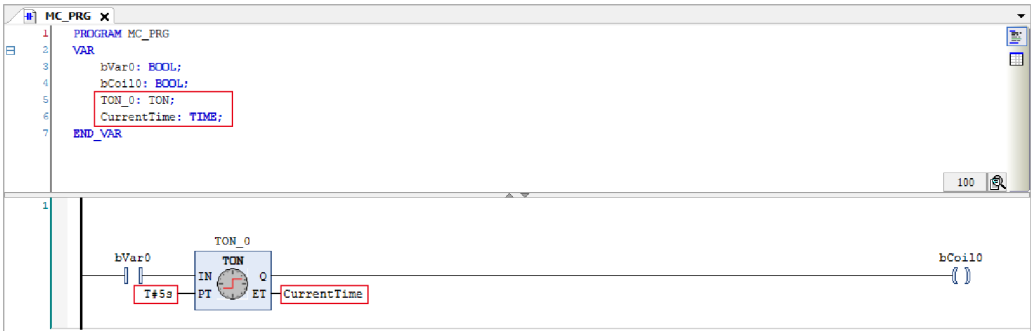
9. Click the [OK] button.
The name of TON will be declared as variable TON_0.



```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
    bCoil0: BOOL;
    TON_0: TON;
END_VAR
```

10. Enter "T#5s" for input PT ("IN PT") and "CurrentTime" for output ET ("O ET"), as shown below.

In the declaration section, declare "CurrentTime" as a TIME type variable.



```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
    bCoil0: BOOL;
    TON_0: TON;
    CurrentTime: TIME;
END_VAR
```

This completes insertion of a normally open contact, coil, and function block TON.

The network (circuit) is now complete.

Info.

- To remove an element that has been placed, select the element and press the <Delete> key.

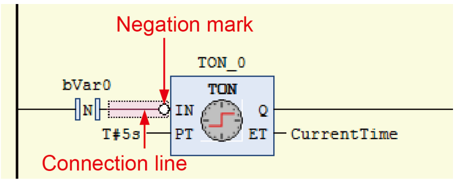
  You can also remove an element by right-clicking the element and selecting "Delete" from the context-sensitive menu that is displayed.

- The normally open contact that is placed can be set as a normally open contact for detecting a rising edge (P). To do so, perform the following procedure with the normally open contact selected.
  - Right-click the normally open contact and select "Edge Detection" from the context-sensitive menu that is displayed.
  - From the menu bar, select **FBD/LD/IL**>**Edge Detection**.
  - Press the <Ctrl> key + <e> key simultaneously.
  - Click the icon on the toolbar.

  If the above procedure is performed one more time, the normally open contact that is placed can be set as a normally open contact for detecting a falling edge (N).

- Input to the function block can be negated. To do so, perform the following procedure with the connection line to the input selected.
  - Right-click the function block and select "Negation" from the context-sensitive menu that is displayed.
  - From the menu bar, select **FBD/LD/IL**>**Negation**.
  - Press the <Ctrl> key + <n> key simultaneously.
  - Click the icon on the toolbar.

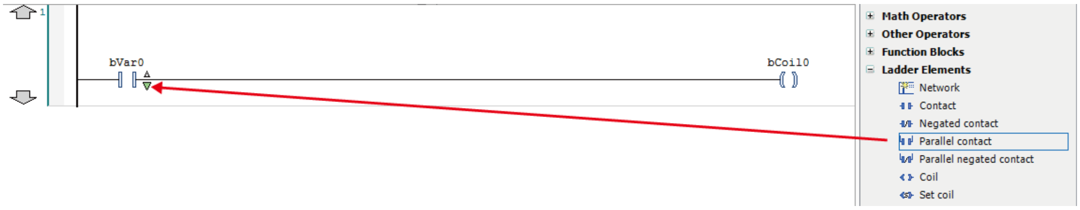
  A negation mark will be displayed on the left side of the input ("IN").



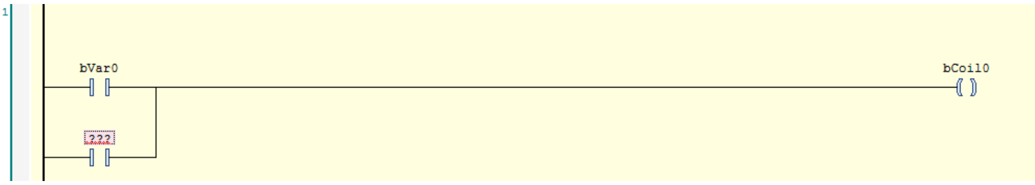
- The LD program can be converted and displayed as an FBD program.
  From the menu bar, select **FBD/LD/IL>View** and select a post-conversion programming language.

7.1.2 Inserting Contacts in Parallel

This section explains how to place a contact in parallel with a normally open contact.
In Toolbox, select **Ladder elements>Parallel contact** and drag the mouse until the position of ▼ displayed on the right side of the normally open contact is reached.

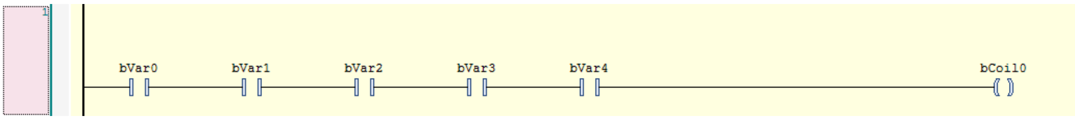


The normally open contacts will be placed in parallel with the other one.

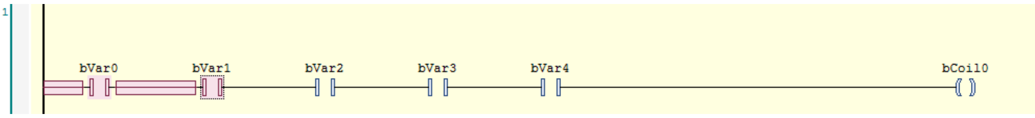


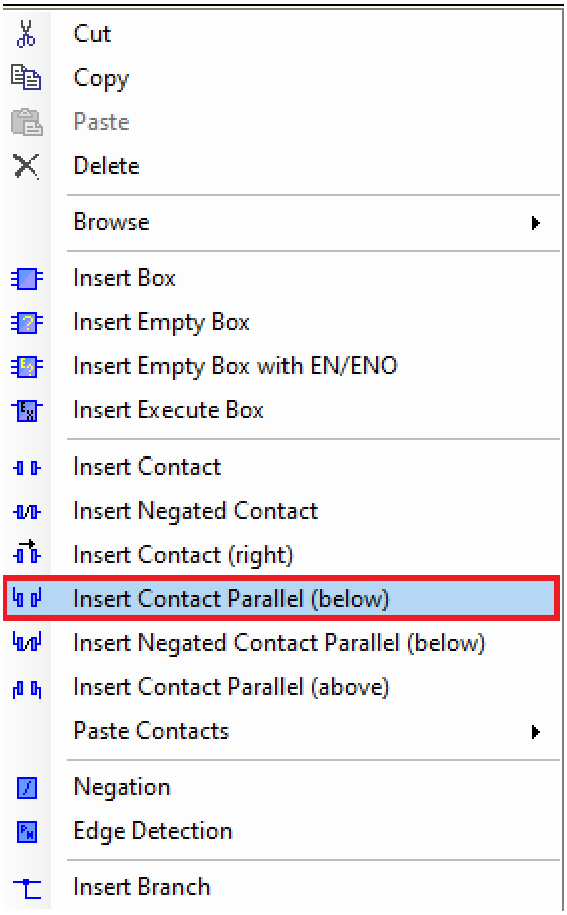
■ Placing a contact in parallel with multiple contacts

The following explains how to place a contact in parallel with multiple contacts.

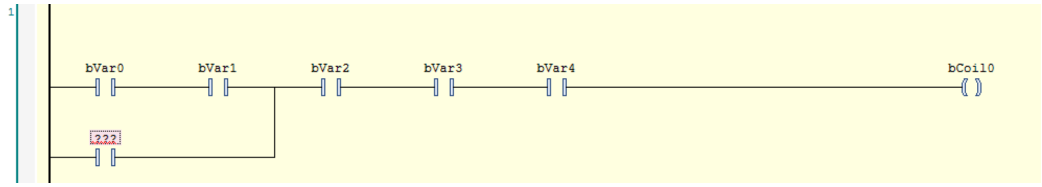


While holding down the <Ctrl> key, select relevant contacts and then right-click.





Select "Insert Contact Parallel (below)" from the context-sensitive menu that is displayed. A contact will be placed in parallel with the selected contacts.

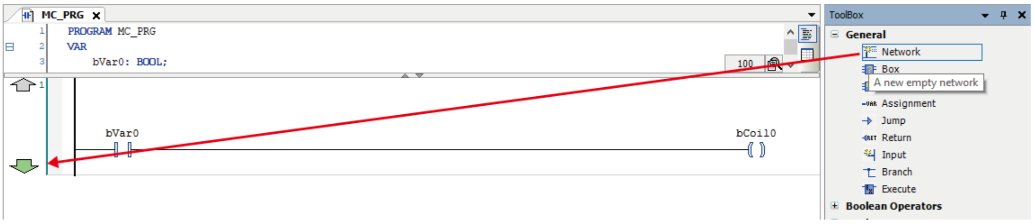


7.1.3 Inserting a Network (Circuit)

This section explains how to insert a new network (circuit).

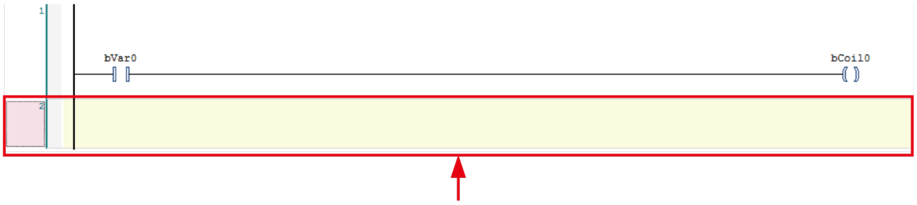
Procedure

1. In Toolbox, select **General**>**Network** and drag the mouse until ⬇ displayed in the network (circuit) in the main pane is reached.



```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
```

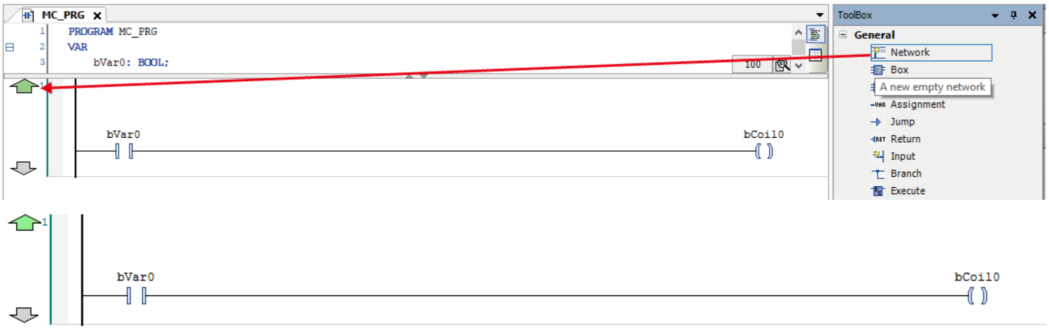
A network (circuit) will be inserted underneath.



Added network

Info.

- You can also insert a network (circuit) underneath using the following procedure:
  - Right-click in the network (circuit) and select "Insert Network (Below)" from the context-sensitive menu that is displayed.
  - From the menu bar, select **FBD/LD/IL**>**Insert Network (below)**.
  - Press the <Ctrl> key + <t> key simultaneously.
- To add a network (circuit) above the existing network, in Toolbox, select **General**>**Network** and drag the mouse until the position of ⬆ is reached.



```
PROGRAM MC_PRG
VAR
    bVar0: BOOL;
```

- To remove a network (circuit), select the network (circuit) and press the <Delete> key.

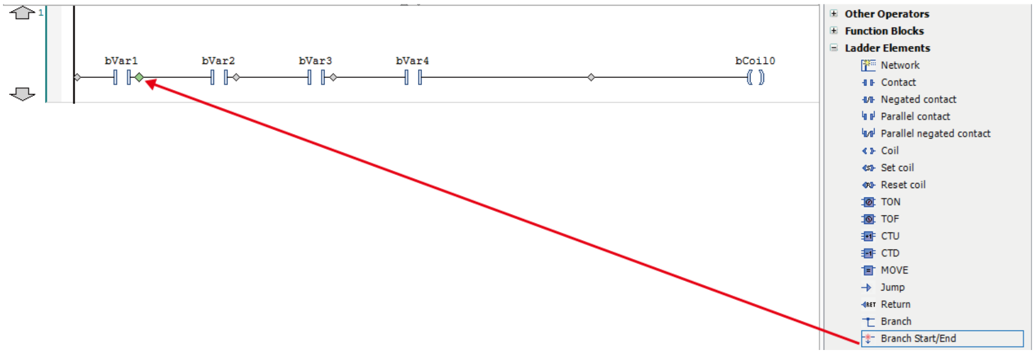
7.1.4 Inserting a Branch

This section explains how to add a branch by specifying the starting point and end point of the branch.

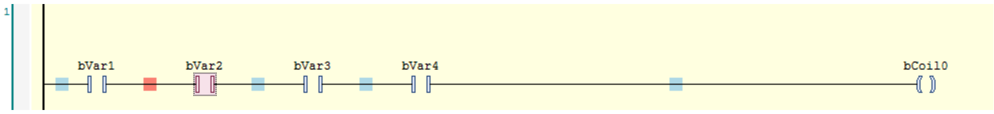
Example: Creating a branch in the following network (circuit) that extends from a point between contacts bVar1 and bVar2 to a point between contacts bVar3 and bVar4

1 2 Procedure

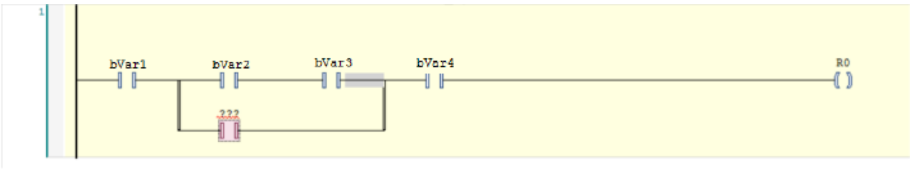
1. In Toolbox, select **Ladder elements**>**Branch Start/End** and drag the mouse until ◆ displayed on the right side of contact bVar1 in the main pane is reached.



When you stop dragging the mouse, a red square mark indicating the starting point of a branch is displayed between contacts bVar1 and bVar2. Blue square marks indicate candidates for the end point of the branch.



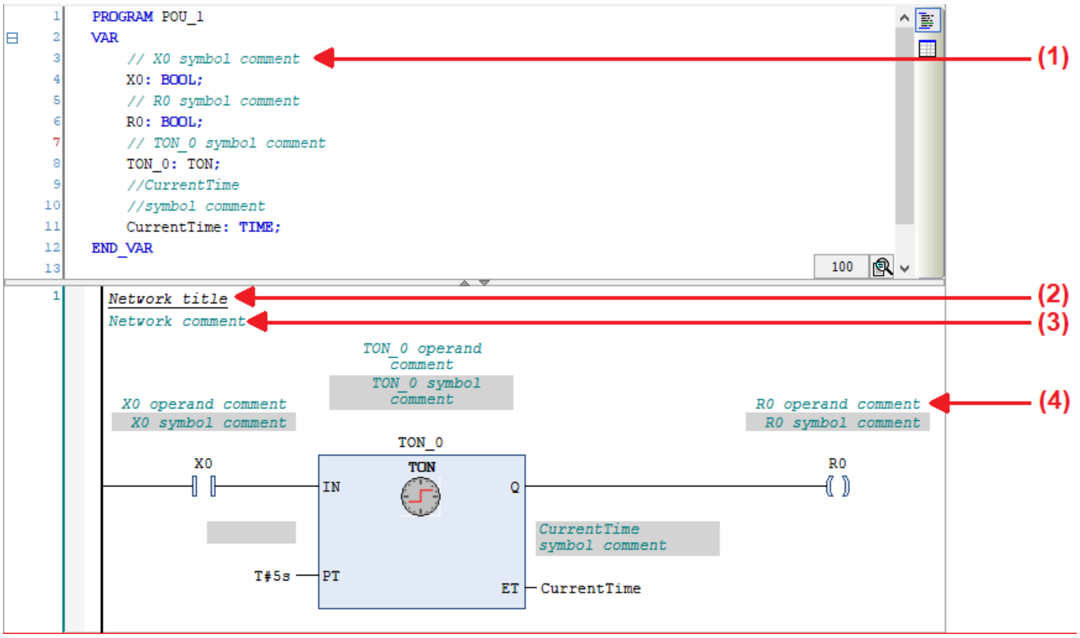
2. Click the blue square mark between contacts bVar3 and bVar4.
A branch that extends from the point between contacts bVar1 and bVar2 to the point between contacts bVar3 and bVar4 will be inserted.



7.1.5 Input of Title and Comment (LD)

The Ladder Diagram programming language allows the user to enter the following four types of titles and comments.

Display examples of titles and comments are shown below.

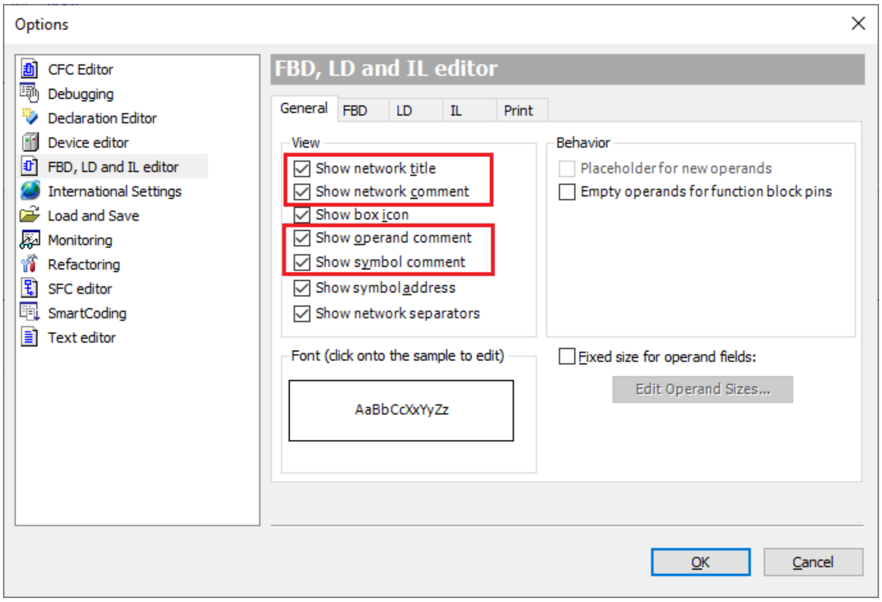


```
PROGRAM POU_1
VAR
    // X0 symbol comment
    X0: BOOL;
    // R0 symbol comment
    R0: BOOL;
    // TON_0 symbol comment
    TON_0: TON;
    //CurrentTime
    //symbol comment
    CurrentTime: TIME;
END_VAR
```

No.	Item	Description
(1)	Symbol comment	This is a comment on a declared variable. The same comment is displayed for the same variable. Enter a comment on a variable in the declaration section. The comment will be displayed in a cell with a black background.
(2)	Network title (circuit title)	A title can be assigned to each network (circuit). Click the top left of the network (circuit) and enter a title.
(3)	Network comment (circuit comment)	A comment can be assigned to each network (circuit). Click the top left of the network (circuit) and enter a comment.
(4)	Operand comment	This is a comment on a variable. Different comments can be assigned to the same variable. Click a position above each variable in the implementation section and enter a comment.

To display titles and comments, you must configure settings.

Open the Options window (by selecting **Tools**>**Options**), select the "FBD, LD, and IL editors" category and then the General tag, and select the items to be displayed in the Display section.



View

- ☑ Show network title
- ☑ Show network comment
- ☑ Show box icon
- ☑ Show operand comment
- ☑ Show symbol comment
- ☑ Show symbol address
- ☑ Show network separators

Behavior

- ☐ Placeholder for new operands
- ☐ Empty operands for function block pins

Font (click onto the sample to edit)

AaBbCcXxYyZz

☐ Fixed size for operand fields:

Info.

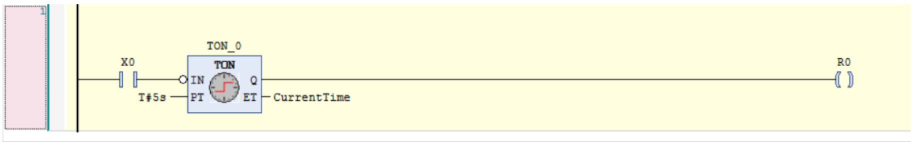
- Titles and comments can be displayed as those translated in a particular language beforehand. For details, refer to "6.8.8 Displaying Programs in Multiple Languages (Project Localization)".

7.1.6 Commenting out a Network (Circuit)

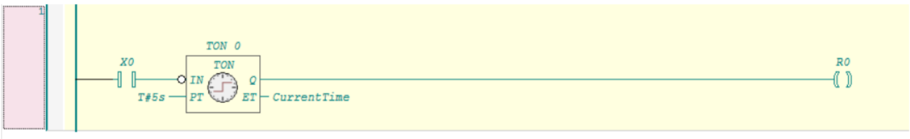
Networks (circuits) can be commented out. A network (circuit) that is commented out cannot be executed.

1 2 Procedure

1. Select a network (circuit) to be commented out.



2. Click the (**) icon (Toggle Network Comment State) on the toolbar.
The network (circuit) will be commented out. To cancel the comment-out state, perform the same operation again.



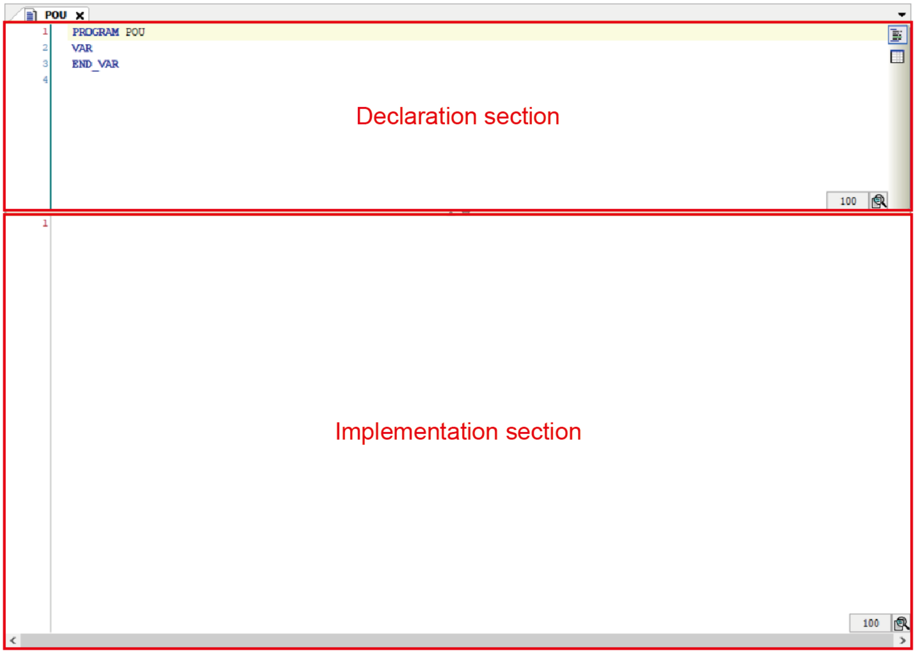
Info.

- You can also comment out the selected network (circuit) using the following procedure:
  - Right-click in the network (circuit) and select "Toggle network comment state" from the context-sensitive menu that is displayed.
  - From the menu bar, select **FBD/LD/IL**>**Toggle network comment state**.
  - Press the <Ctrl> key + <o> key simultaneously.

7.2 Programming in Structured Text (ST)

This section explains how to create programs (ST programs) in Structured Text compliant with IEC 61131-3, the international standard for PLC programming languages.

- To create ST programs, POU objects for ST programs are required. Set the object setup language to Structured Text (ST).
- An ST program is made up by combining expressions and instructions. Expressions and instructions can also be executed under certain conditions or within a loop. Each instruction must end with a semicolon (;).



```
PROGRAM POU
VAR
END_VAR
```

7.2.1 ST Program Syntax

For ST programs, the following syntax can be used.

Item	Example
Assignment statement	The value of the right side is set on the left side. Example: iVar1 : = 4;
Set assignment statement	If the value of the right side is judged to be TRUE, TRUE will be set on the left side. Once the value of the left side is judged to be TRUE, the left side will maintain TRUE even if the value of the right side is judged to be FALSE. Do not leave any spaces between "S" and "=". Example: bVar0S=bVar1;
Reset assignment statement	If the value of the right side is judged to be TRUE, FALSE will be set on the left side. Once the value of the left side is judged to be FALSE, the left side will maintain FALSE even if the value of the right side is judged to be FALSE. Do not leave any spaces between "R" and "=".

Item	Example
	Example: bVar0R=bVar1;

IF instruction	Conditions are judged and subordinate instructions are executed according to the judgment result.
	Example:
	IF (iVar0 = 4) THEN
	iVar1 : = 5;
	ELSIF (iVar0 = 7) THEN
	iVar1 : = 10;
	ELSE
	iVar1 : = 15;
	END_IF;

FOR instruction	Subordinate instructions are executed repeatedly the specified number of times. Example: FOR iVar0 : = 1 TO 10 BY 1 DO iVar1 : = iVar1 + 1; END_FOR;
WHILE instruction	Conditions are judged and subordinate instructions are executed repeatedly as long as the conditions are satisfied. Example: WHILE (iVar0 <> 0) DO iVar1 : = iVar1 * 2; END_WHILE;

CASE instruction	Conditions are judged and subordinate instructions are executed according to the judgment result.
	Example:
	CASE iVar0 OF
	1 : iVar1 : = iVar1 / 2;
	2 : iVar1 : = iVar1 / 4;
	ELSE
	iVar1 : = iVar1 / 8;
	END_CASE;

REPEAT instruction	Conditions are judged and subordinate instructions are executed repeatedly as long as the conditions are satisfied. Example: REPEAT iVar0 : = iVar0 + 1; UNTIL iVar0 = 100 END_REPEAT;
EXIT instruction	The EXIT instruction is used to terminate a loop within the FOR, WHILE, or REPEAT instruction.
RETURN instruction	The RETURN instruction is used to terminate a program organization unit (POU). Instructions within POUs following the RETURN instruction will not be executed.
JMP instruction	The JMP instruction is used to unconditionally move control to the line indicated by the JMP label. Example:

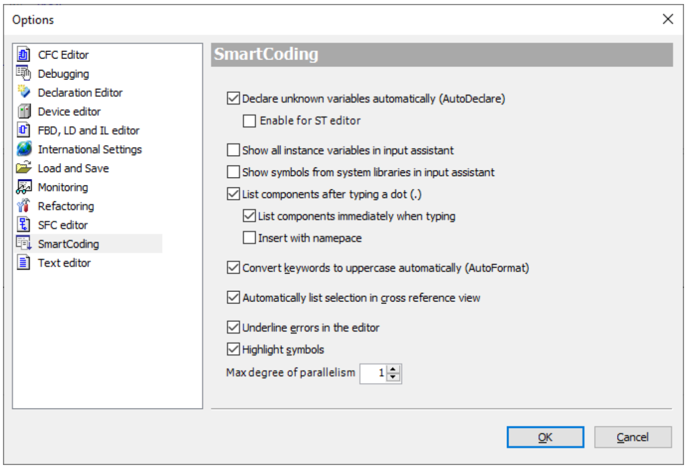
Item	Example

	iVar0 : = 0;
	Label1 : iVar0 : = iVar0 + 1;
	IF (iVar1 = 5) THEN
	JMP Label1;
	END_IF;

CONTINUE instruction	The CONTINUE instruction is used to move control to the beginning of the loop within the FOR, WHILE, or REPEAT instruction.

Info.

- Entered keywords are converted to uppercase letters automatically (AutoFormat). To disable this function, clear the "Convert keywords to uppercase automatically (AutoFormat)" check box in the Options window.



SmartCoding

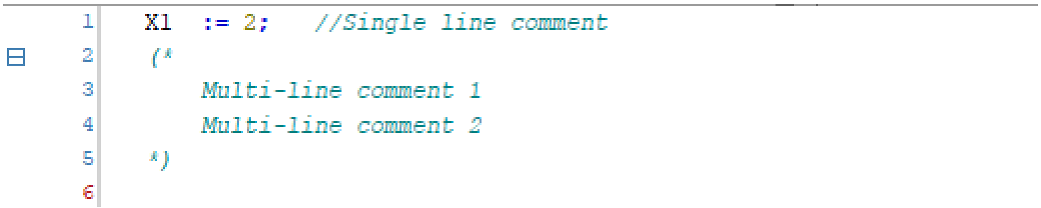
☑ Declare unknown variables automatically (AutoDeclare)
☐ Enable for ST editor
☐ Show all instance variables in input assistant
☐ Show symbols from system libraries in input assistant
☑ List components after typing a dot (.)
☑ List components immediately when typing
☐ Insert with namepace
☑ Convert keywords to uppercase automatically (AutoFormat)
☑ Automatically list selection in cross reference view
☑ Underline errors in the editor
☑ Highlight symbols
Max degree of parallelism 1

7.2.2 Commenting out Code in ST Program

In ST programs, code can be commented out. Program code that is commented out cannot be executed.

Comment type	Description
Single line	Program code from // to the end of the line is treated as a comment. Example: bVar1 := 2; // Single-line comment
Multiple lines	Program code from (* to *) is treated as a comment. (* *) can also be inserted into another comment enclosed between (* and *). Example: (* Multiple-line comment 1 Multiple-line comment 2 *)

Example: Single-line and multiple-line comments

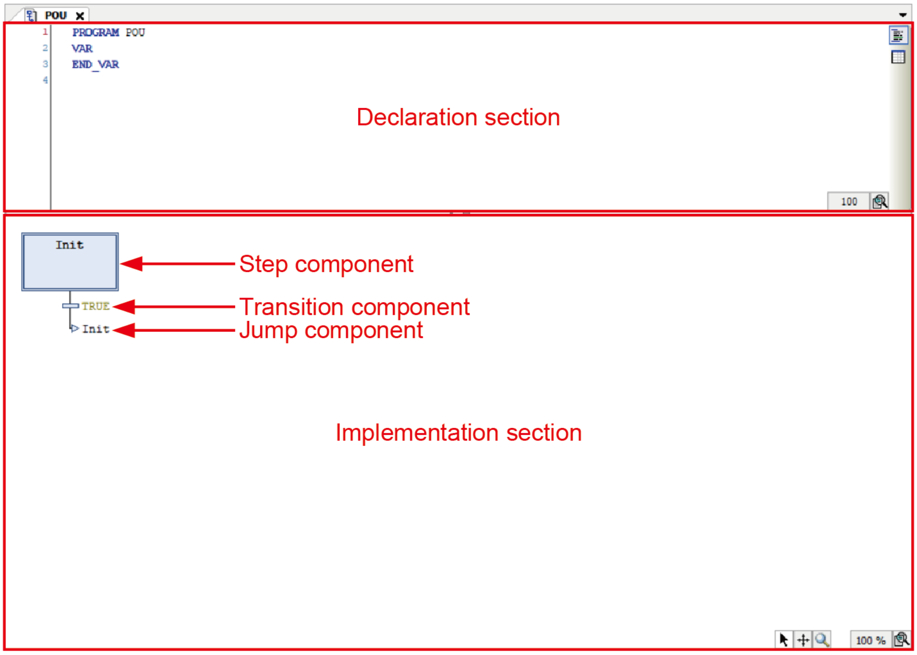


```
X1  := 2;   //Single line comment
(*
    Multi-line comment 1
    Multi-line comment 2
*)

```

7.3 Programming in Sequential Function Chart (SFC)

This section explains how to create programs (SFC programs) in Sequential Function Chart compliant with IEC 61131-3, the international standard for PLC programming languages.

- To create SFC programs, POU objects for SFC programs are required. Set the object setup language to Sequential Function Chart (SFC).
- The SFC program editor is divided into the declaration section and implementation section. Three elements are originally coded in the implementation section.



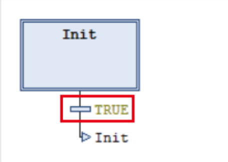
```
PROGRAM POU
VAR
END_VAR
```

7.3.1 Inserting Elements from Menu

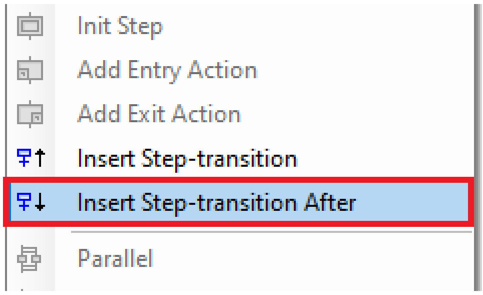
For example, step elements and transition elements can be inserted from the menu, as below.

1 2 Procedure

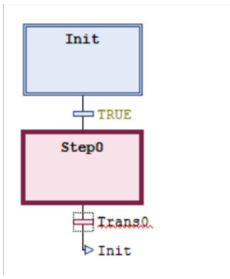
1. Select the TRUE transition element in the implementation section.
The selected transition element will turn red.



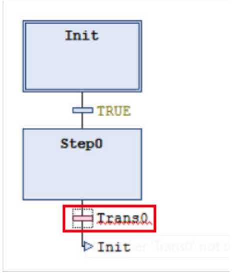
2. Click the right mouse button and select "Insert Step-transition After" from the context-sensitive menu that is displayed.



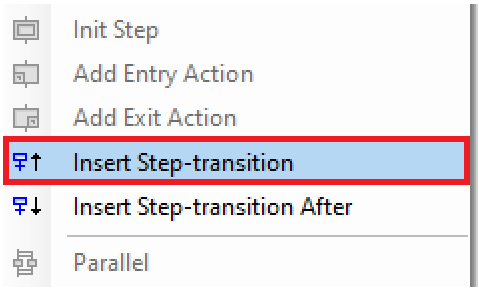
"Step0" step element and "Trans0" transition element will be inserted below the TRUE element.



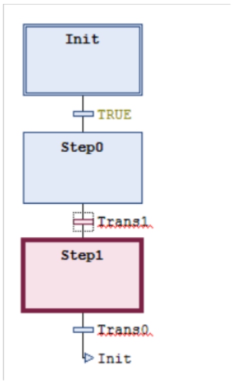
3. Select the "Trans0" element.



4. Click the right mouse button and select "Insert Step-transition" from the context-sensitive menu that is displayed.



"Step1" step element and "Trans1" transition element will be inserted above the "Trans0" transition element.



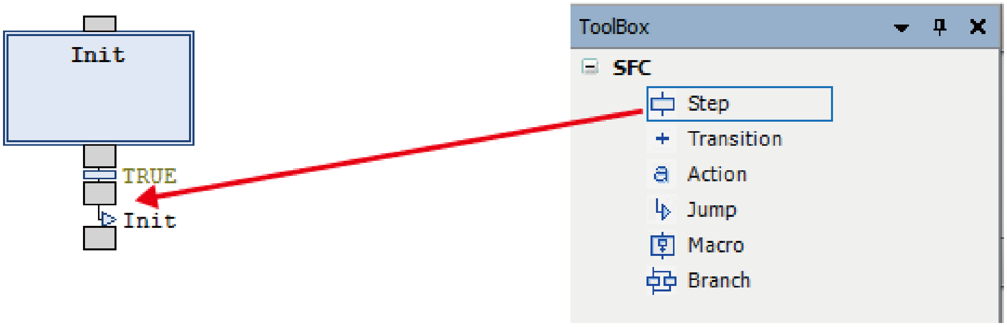
7.3.2 Inserting Elements from Toolbox

Each element can be inserted from Toolbox.

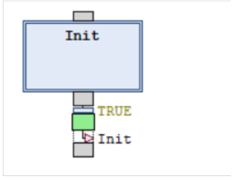
This section explains the procedure for inserting elements from Toolbox, using a step element as an example.

Procedure

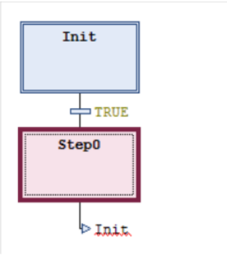
1. In Toolbox, select "Step" and then drag the step element to the position where you want to insert it.



When you drag the step element to the insertion position, the step element is transformed into 🟩.



2. Stop dragging the mouse.
   The step element will be inserted.



Init
TRUE
Step0
Init

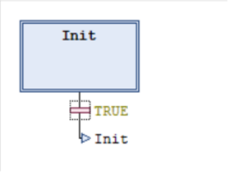
7.3.3 Inserting Elements from Toolbar

For example, step elements and transition elements can be inserted from the toolbar, as below.

1 2 Procedure

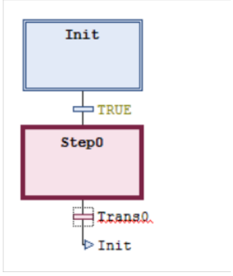
1. Select the TRUE transition element in the implementation section.

The selected transition element will turn red.



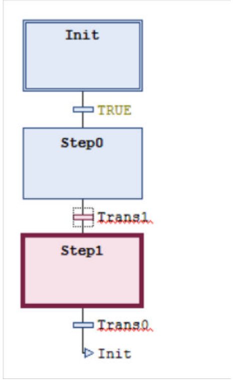
2. Click the icon ("Insert Step-transition After") on the toolbar.

"Step0" step element and "Trans0" transition element will be inserted below the TRUE element.



3. Select the "Trans0" transition element and click the icon ("Insert Step-transition") on the toolbar.

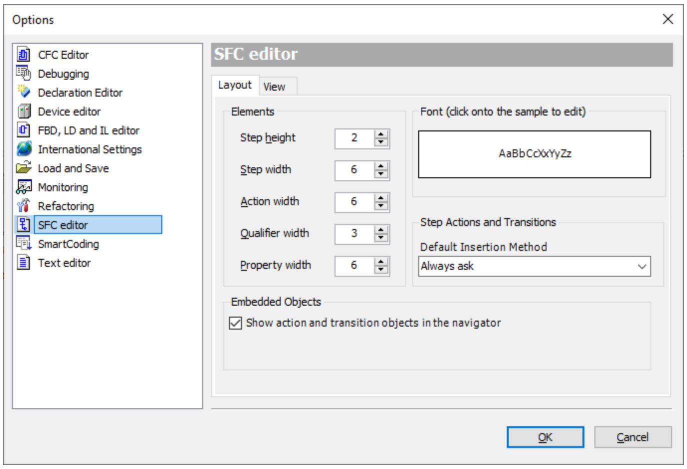
"Step1" step element and "Trans1" transition element will be inserted above the "Trans0" transition element.



7.3.4 Setting up the SFC Editor

For SFC editor elements, you can change step specifications, fonts, and other settings.
From the menu bar, select **Tools**>**Options** to open the "Options" dialog box.
In the "Options" dialog box, select the "SFC editor" category and change the settings.

Layout



Elements

Step height 2

Step width 6

Action width 6

Qualifier width 3

Property width 6

Font (click onto the sample to edit)

AaBbCcXxYyZz

Step Actions and Transitions

Default Insertion Method: Always ask

Embedded Objects

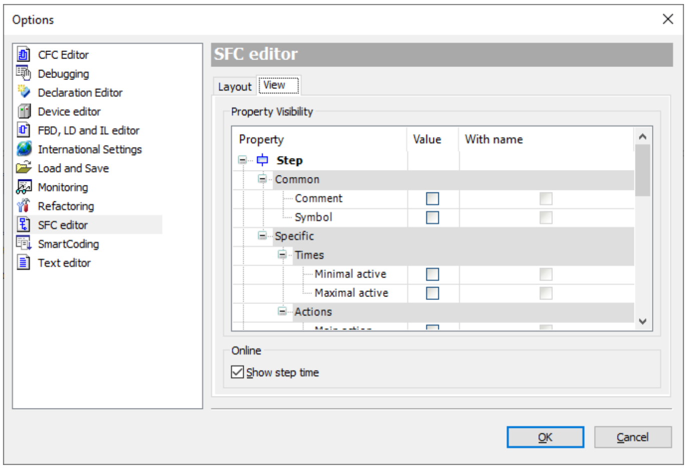
☑ Show action and transition objects in the navigator

Item name		Default value	Function
Elements	Step height	2	Sets the height of a step. Setting range: 1 to 100 matrix units(Note 1)
	Step width	6	Sets the width of a step. Setting range: 2 to 100 matrix units(Note 1)
	Action width	6	Sets the width of an action. Setting range: 2 to 100 matrix units(Note 1)
	Qualifier width	3	Sets the width of a qualifier. Setting range: 2 to 100 matrix units(Note 1)
	Property width	6	Sets the width of a property. Setting range: 2 to 100 matrix units(Note 1)
Font		-	Sets a font to be displayed on the SFC editor.
Step Actions and Transitions	Default insertion method	Always check	Sets the operation to be performed when an action is added to a step. **Copy reference:** When a step is copied, a link to the step action is also copied. The step that is copied invokes the same action. **Copy implementation:** The step action of a step that is copied is embedded. A new action object is copied to a new step.

Item name		Default value	Function
			Always ask: Which of the above operations is to be executed is checked each time an action is initially added to a step.
Embedded Objects	Show actions and transition objects in navigator pane	Selected	This check box is used to specify whether to display the actions embedded in steps in the navigator pane when an action is added to a step by "Copy implementation". **Selected:** An action that is embedded in a step by "Copy implementation" is displayed in the Device view or POU view. **Cleared:** An action that is embedded in a step by "Copy implementation" is not displayed in the Device view or POU view.

(Note 1) "1 matrix unit" is equal to the font size specified in Font in **Options**>**Text editor**>**Text area** tab.

View



Property	Value	With name
Step		
Common		
Comment	☐	☐
Symbol	☐	☐
Specific		
Times		
Minimal active	☐	☐
Maximal active	☐	☐
Actions		

☑ Show step time

Item name		Default value	Settings
Property Visibility		Cleared	Specifies whether to display property values and names beside steps, etc. Selected: Displays property values and names Canceled: Does not display property values or names
Online	Display step time	Selected	Specifies whether to display step active time beside the step in online mode Selected: Displays active time Cleared: Does not display active time

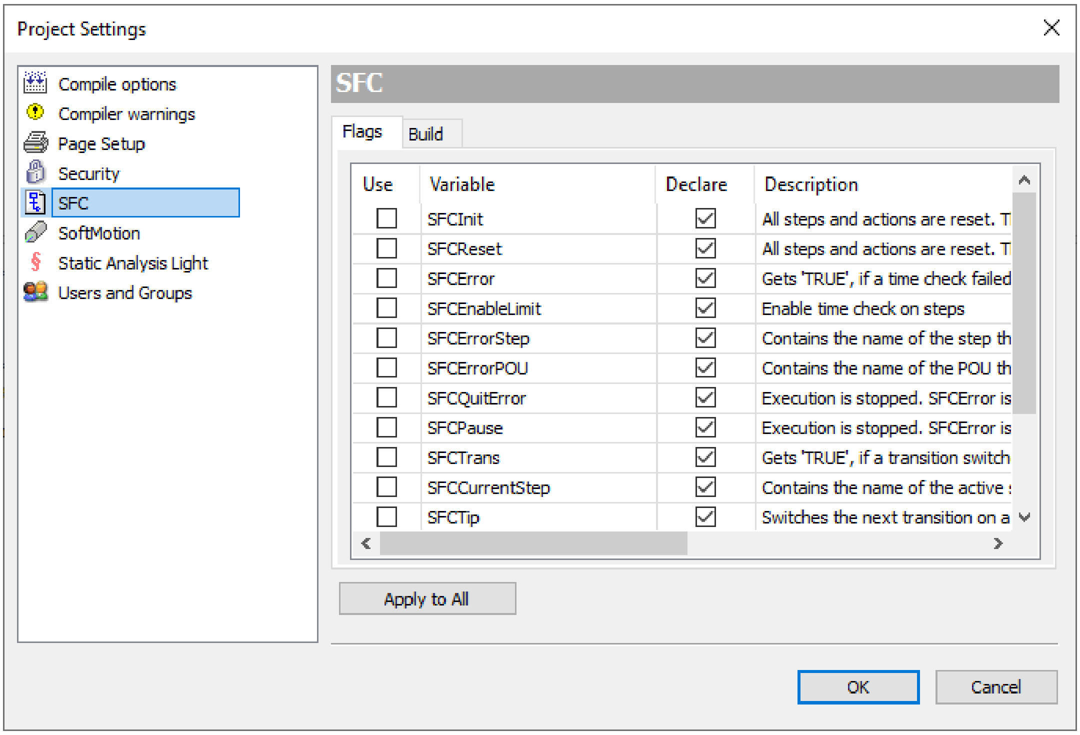
7.3.5 Setting SFC Program Execution Conditions

■ Setting SFC program execution conditions

For SFC programs within a project, you can specify whether to generate code for variables used to check processing or for active transitions during build.

From the menu bar, select **Project**>**Project Settings**. In the Project Settings dialog box, select the "SFC" category.

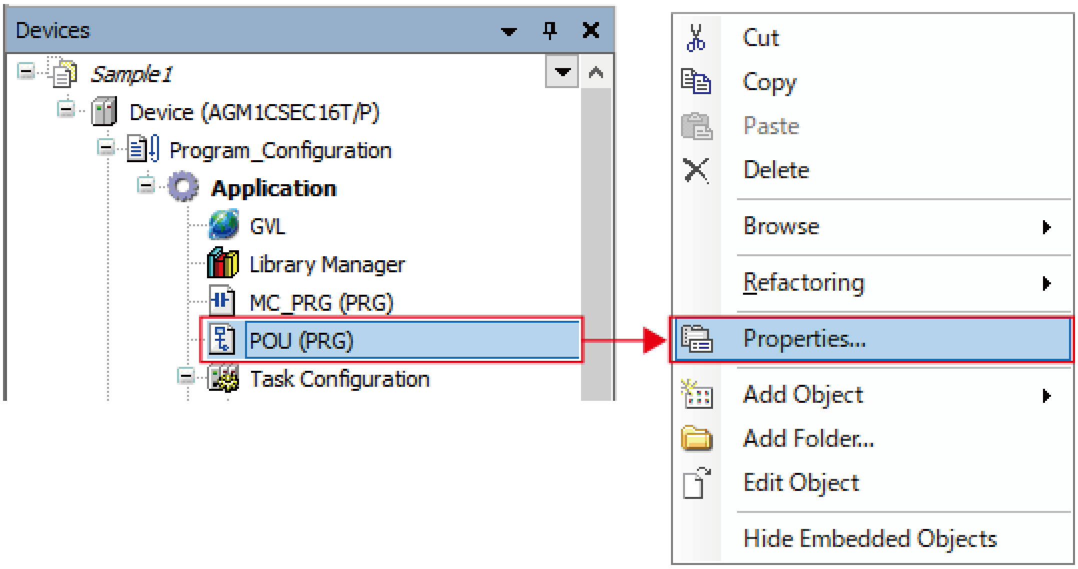
The settings will be applied to all SFC objects.



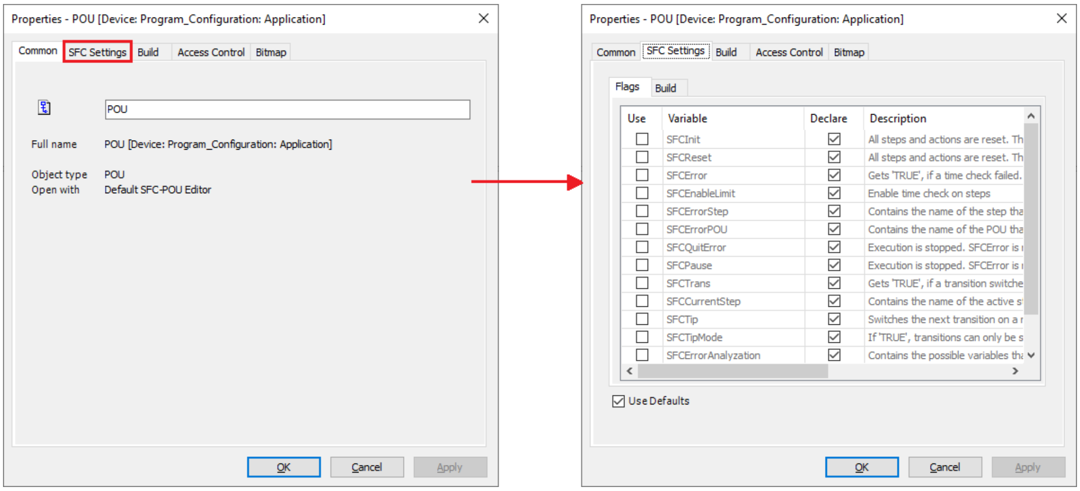
Use	Variable	Declare	Description
□	SFCInit	☑	All steps and actions are reset. T
□	SFCReset	☑	All steps and actions are reset. T
□	SFCError	☑	Gets 'TRUE', if a time check failed
□	SFCEnableLimit	☑	Enable time check on steps
□	SFCErrorStep	☑	Contains the name of the step th
□	SFCErrorPOU	☑	Contains the name of the POU th
□	SFCQuitError	☑	Execution is stopped. SFCError is
□	SFCPause	☑	Execution is stopped. SFCError is
□	SFCTrans	☑	Gets 'TRUE', if a transition switch
□	SFCCurrentStep	☑	Contains the name of the active s
□	SFCTip	☑	Switches the next transition on a

The "Properties" dialog box for SFC objects can be enabled for only particular SFC objects.

Right-click the POU object in the navigator pane and then select "Properties" from the context-sensitive menu that is displayed.



In the "Properties" dialog box, select the "SFC Settings" tab and clear the "Use default" check box.



POU

Full name POU [Device: Program_Configuration: Application]

Object type POU

Open with Default SFC-POU Editor

Use	Variable	Declare	Description
□	SFCInit	☑	All steps and actions are reset. Th
□	SFCReset	☑	All steps and actions are reset. Th
□	SFCError	☑	Gets 'TRUE', if a time check failed.
□	SFCEnableLimit	☑	Enable time check on steps
□	SFCErrorStep	☑	Contains the name of the step tha
□	SFCErrorPOU	☑	Contains the name of the POU tha
□	SFCQuitError	☑	Execution is stopped. SFCError is r
□	SFCPause	☑	Execution is stopped. SFCError is r
□	SFCTrans	☑	Gets 'TRUE', if a transition switche
□	SFCCurrentStep	☑	Contains the name of the active s
□	SFCTip	☑	Switches the next transition on a r
□	SFCTipMode	☑	If 'TRUE', transitions can only be s
□	SFCErrorAnalyzation	☑	Contains the possible variables tha

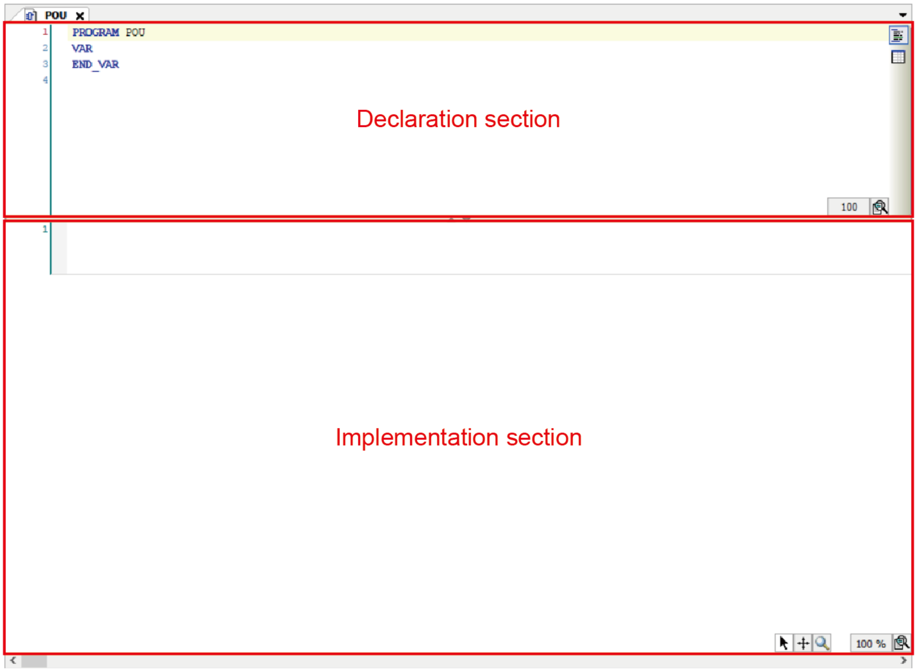
☑ Use Defaults

Clearing the check box allows settings to be changed and the Properties dialog box to be enabled for only particular SFC objects.

7.4 Programming in Function Block Diagram (FBD)

This section explains how to create programs (FBD programs) in Function Block Diagram compliant with IEC 61131-3, the international standard for PLC programming languages.

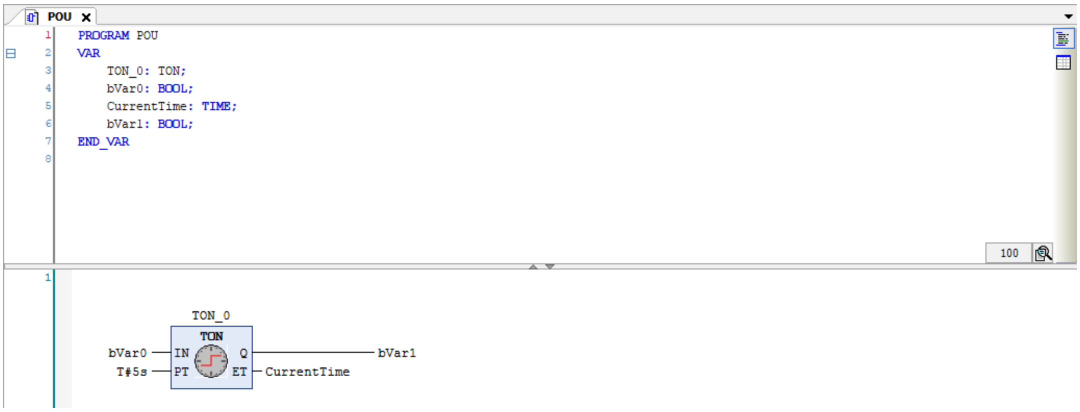
- To create FBD programs, POU objects for FBD programs are required. Set the object setup language to Function Block Diagram (FBD).
- The editor for FBD programs is a window like the one shown below. It consists of the declaration section used to declare variables and the implementation section used to enter program data. Elements can be arranged in the implementation section by selecting them from Toolbox and then dragging and dropping them in the implementation section.



```
PROGRAM POU
VAR
END_VAR
```

7.4.1 Entering Function Blocks

This section explains the procedure for entering function blocks, using an FBD program consisting of the following variables and function block TON as an example.

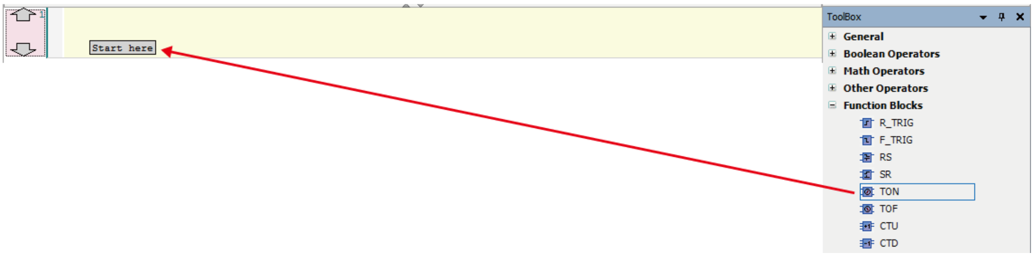


```
PROGRAM POU
VAR
    TON_0: TON;
    bVar0: BOOL;
    CurrentTime: TIME;
    bVar1: BOOL;
END_VAR
```

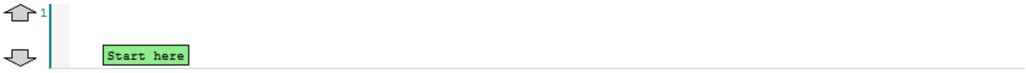
Procedure

1. In Toolbox, select **Function block**>**TON** and drag the mouse until the mouse pointer reaches the position where "Start here" is displayed in the implementation section.

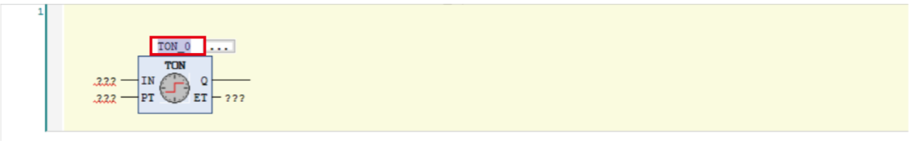
   "Start here" will be displayed in the implementation section.



When you drag the mouse until the position of "Start here" is reached, the display of "Start here" turns green.

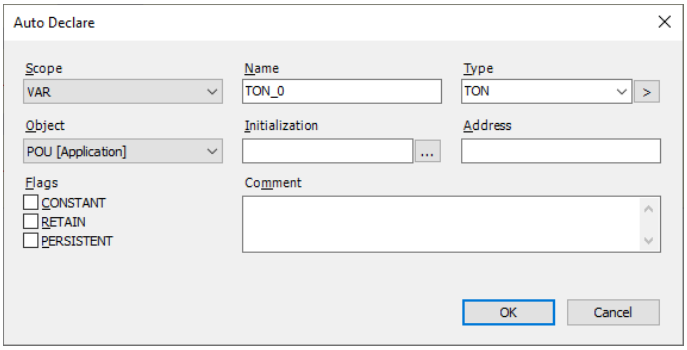


When you stop dragging the mouse at the position of "Start here", function block TON will be placed in the network (circuit).



2. Select TON and enter variable TON_0, and then press the <Enter> key.

   The "Auto Declare" dialog box will be displayed.



Auto Declare

Scope: VAR

Name: TON_0

Type: TON

Object: POU [Application]

Initialization:

Address:

Flags

□ CONSTANT

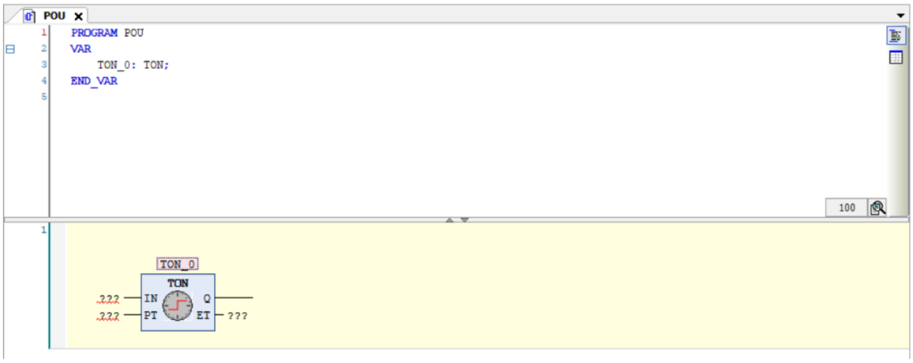
□ RETAIN

□ PERSISTENT

Comment:

3. Click the [OK] button.

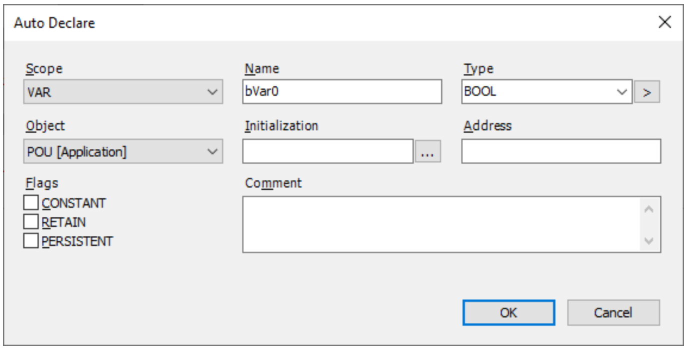
Instance variable TON_0 for function block TON will be declared in the declaration section.



```
PROGRAM POU
VAR
    TON_0: TON;
END_VAR
```

4. Select "???" beside IN of function block TON and enter variable bVar0, and then press the <Enter> key.

The "Auto Declare" dialog box will be displayed.



Auto Declare

Scope: VAR

Name: bVar0

Type: BOOL

Object: POU [Application]

Initialization:

Address:

Flags

□ CONSTANT

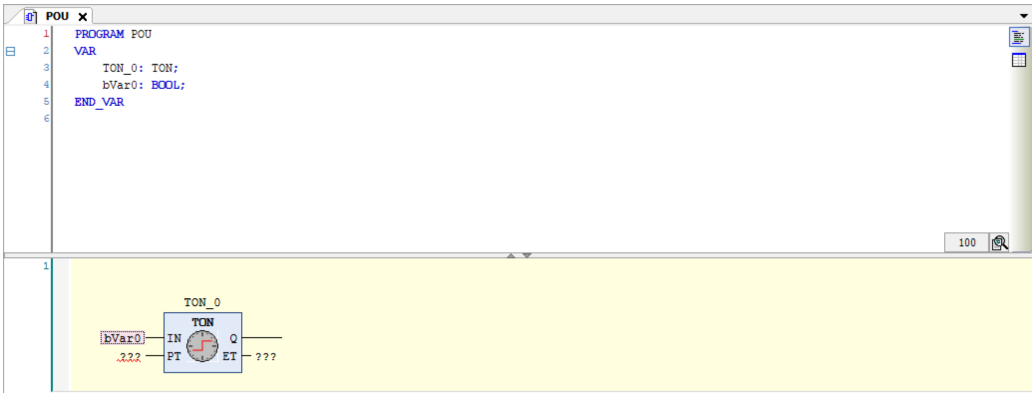
□ RETAIN

□ PERSISTENT

Comment:

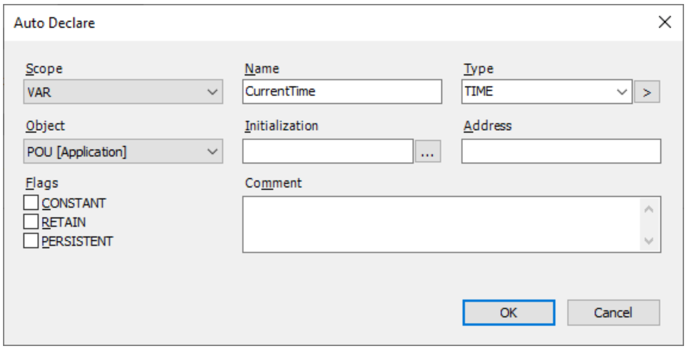
5. Click the [OK] button.

Variable bVar0 will be declared in the declaration section.



```
PROGRAM POU
VAR
    TON_0: TON;
    bVar0: BOOL;
END_VAR
```

6. In function block TON, enter "T#5s" for input PT ("IN PT") and variable "CurrentTime" for output ET ("O ET").
   When "CurrentTime" is entered, the "Auto Declare" dialog box is displayed.



Auto Declare

Scope: VAR

Name: CurrentTime

Type: TIME

Object: POU [Application]

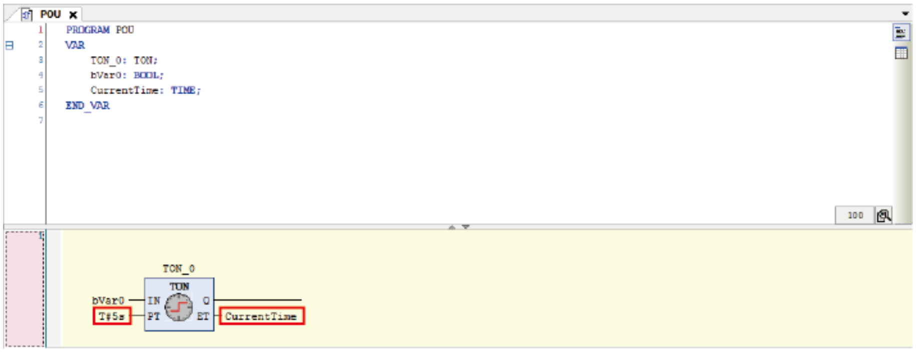
Initialization:

Address:

Flags
□ CONSTANT
□ RETAIN
□ PERSISTENT

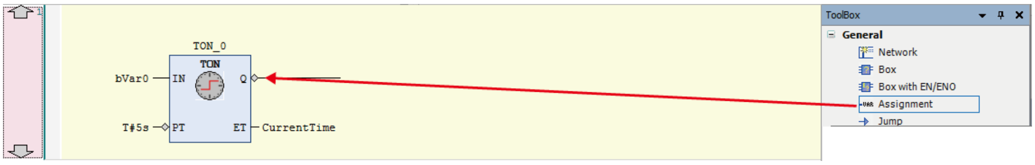
Comment:

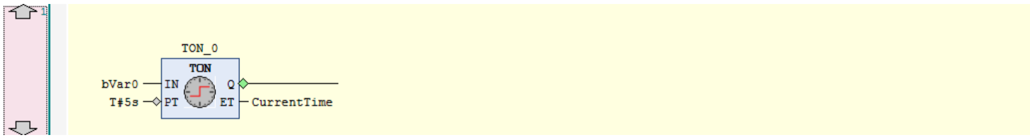
When making a declaration, check that the type is "Time".



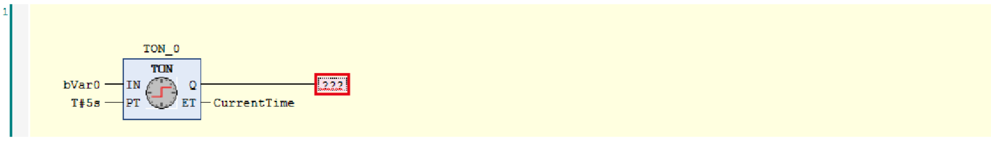
```
PROGRAM POU
VAR
    TON_0: TON;
    bVar0: BOOL;
    CurrentTime: TIME;
END_VAR
```

7. In Toolbox, select **General**>**Assignment** and drag the mouse until a diamond-shaped icon appears on the right side of output "Q" in function block TON.

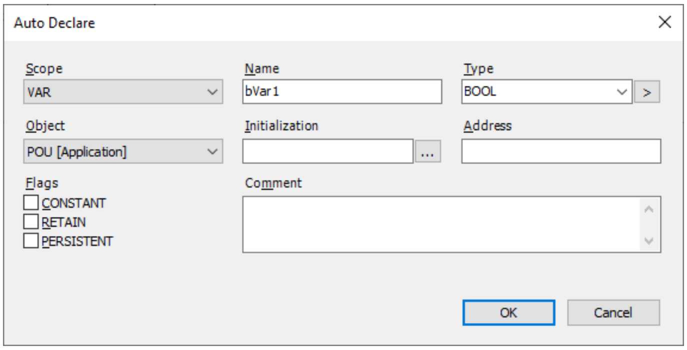




When you stop dragging the mouse, "???" appears on the right side of output "Q".



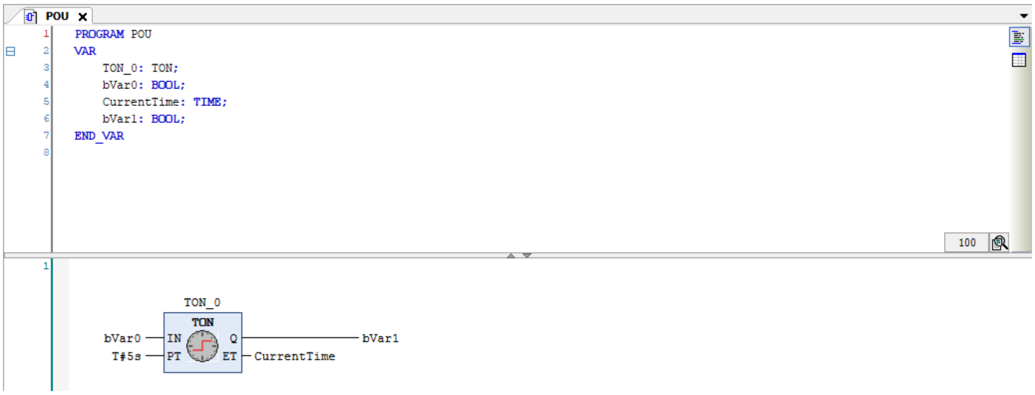
8. Select "???" on the right side of output "Q" and enter variable bVar1, and then press the <Enter> key.
The "Auto Declare" dialog box will be displayed.



Auto Declare

Scope: VAR
Name: bVar1
Type: BOOL
Object: POU [Application]
Initialization:
Address:
Flags
□ CONSTANT
□ RETAIN
□ PERSISTENT
Comment:

Declare variable bVar1.



```
PROGRAM POU
VAR
    TON_0: TON;
    bVar0: BOOL;
    CurrentTime: TIME;
    bVar1: BOOL;
END_VAR
```

Info.

- The FBD program can be converted and displayed as an LD program.
  From the menu bar, select **FBD/LD/IL>View** and select a post-conversion programming language.

7.4.2 Inserting and Commenting out a Network (Circuit)

The procedure for inserting a new network (circuit) is the same as for LD programs.

Refer to "7.1.3 Inserting a Network (Circuit)".

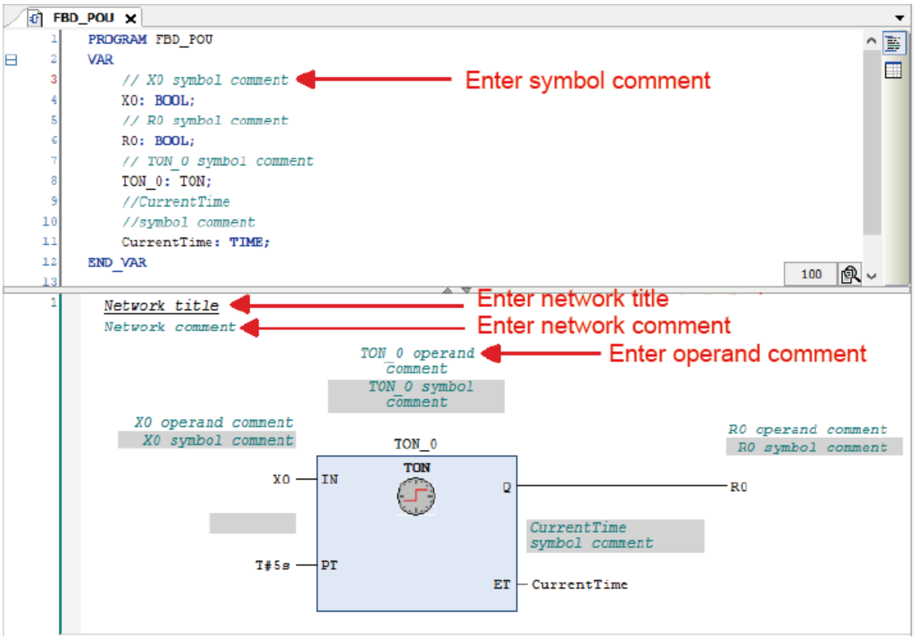
Networks (circuits) can be commented out. The procedure for commenting out a network (circuit) is the same as for LD programs.

Refer to "7.1.6 Commenting out a Network (Circuit)".

7.4.3 Input of Title and Comment (FBD)

In FBD programs, titles and comments can be entered in the same way as for LD programs.

Refer to "7.1.5 Input of Title and Comment (LD)".



```
PROGRAM FBD_POU
VAR
    // X0 symbol comment
    X0: BOOL;
    // R0 symbol comment
    R0: BOOL;
    // TON_0 symbol comment
    TON_0: TON;
    //CurrentTime
    //symbol comment
    CurrentTime: TIME;
END_VAR
```

Enter symbol comment

Enter network title

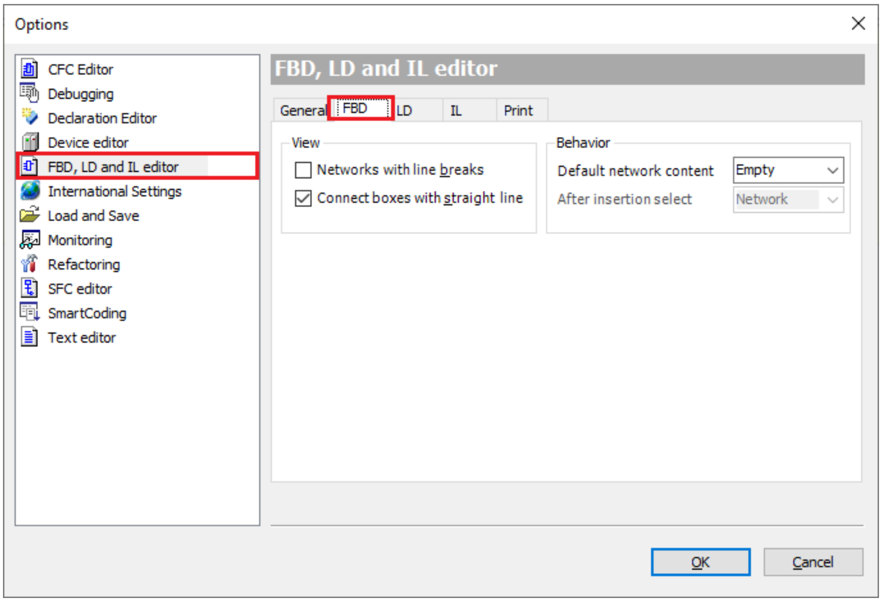
Enter network comment

Enter operand comment

7.4.4 Settings in FBD Program

For FBD programs, the following settings can be configured.

Open the Options window (by selecting **Tools**>**Options**), select the "FBD, LD and IL editors" category and specify settings in the "FBD" tag window.



View

- □ Networks with line breaks
- ☑ Connect boxes with straight line

Behavior

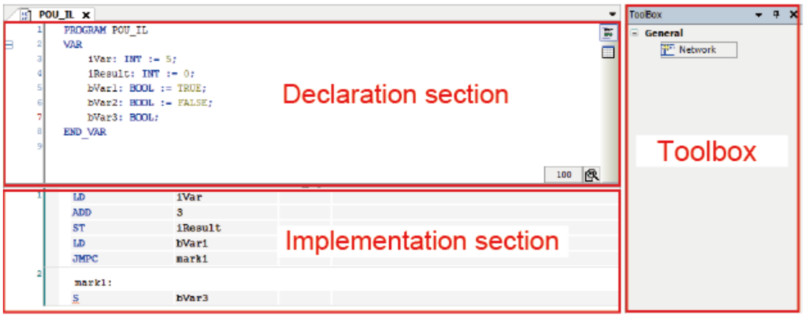
- Default network content: Empty
- After insertion select: Network

Type	Item	Description
View	Networks with line breaks	Selects whether to arrange elements by inserting line breaks automatically so that the display fits in the lateral width of the main pane.
	Connect boxes with straight line	Selects whether to fix the shortest length of a line connecting boxes.
Behavior	Default network content	Selects whether to arrange elements and variables automatically or arrange nothing when a network (circuit) is inserted.
	After insertion select	Selects whether to select a circuit or element after a network (circuit) is inserted.

7.5 Programming in Instruction List (IL)

This section explains how to create programs (IL programs) in Instruction List compliant with IEC 61131-3, the international standard for PLC programming languages.

- IL programs are text-based programs that consist of instructions and operands.
- To create IL programs, POU objects for IL programs are required. Set the object setup language to Instruction List (IL).
- The editor for IL programs is a window like the one shown below. It consists of the declaration section used to declare variables and the implementation section used to enter program data.



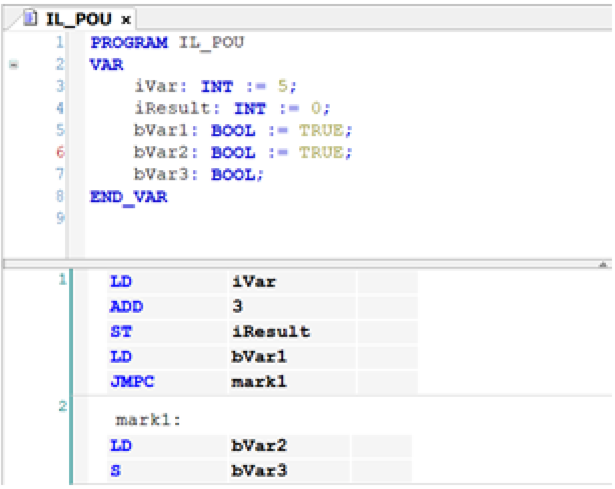
```
PROGRAM POU_IL
VAR
    iVar: INT := 5;
    iResult: INT := 0;
    bVar1: BOOL := TRUE;
    bVar2: BOOL := FALSE;
    bVar3: BOOL;
END_VAR
```

```
LD      iVar
ADD     3
ST      iResult
LD      bVar1
JMPC    mark1

mark1:
S       bVar3
```

- To use POU objects for IL programs, the "Enable IL" check box must be selected as below.
- From the menu bar, select **Tools**>**Options**>**FBD, LD and IL editors**> and select the "Enable IL" check box in the "IL" tab.

7.5.1 Entering Instructions and Operands

This section explains how to create an IL program that consists of the following instructions.

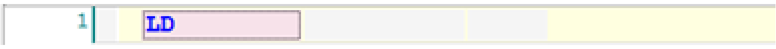


```
PROGRAM IL_POU
VAR
    iVar: INT := 5;
    iResult: INT := 0;
    bVar1: BOOL := TRUE;
    bVar2: BOOL := TRUE;
    bVar3: BOOL;
END_VAR
```

```
LD      iVar
ADD     3
ST      iResult
LD      bVar1
JMPC    mark1

mark1:
LD      bVar2
S       bVar3
```

1 2 Procedure

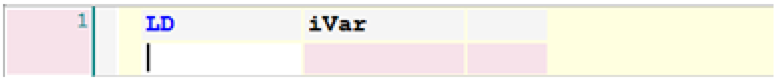
1. Select the cell in the first row and the first column, and enter instruction "LD".
The LD instruction will be entered.



```
1  LD
```

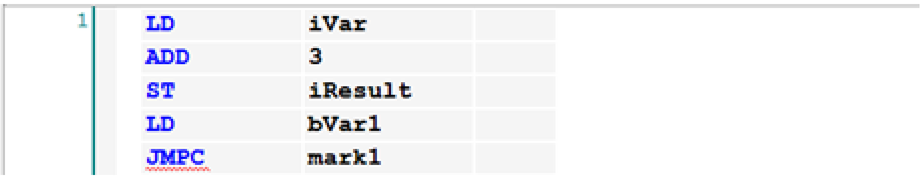
2. Press the <Tab> key to move to another cell and enter operand "iVar".
The operand will be entered and the "Auto Declare" dialog box will be displayed.
In the "Auto Declare" dialog box, declare a variable.

3. Press the <Ctrl> key + <Enter> key simultaneously.
The cursor will move to the next row.



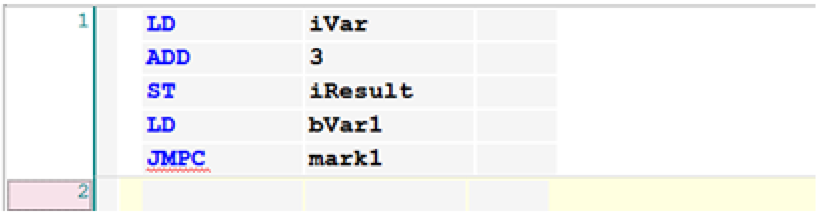
```
1  LD      iVar
   |
```

In the second and subsequent rows, enter instructions and operands in the same way as above.



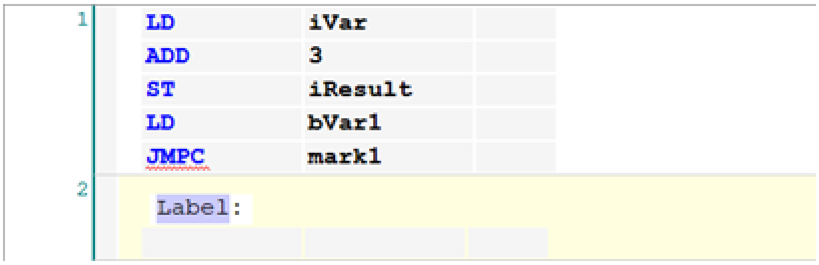
```
1  LD      iVar
   ADD     3
   ST      iResult
   LD      bVar1
   JMPC    mark1
```

4. In Toolbox, select **General**>**Network** and drag the mouse until ⬇ displayed in the network (circuit) in the main pane is reached.
A new network (circuit) will be inserted.



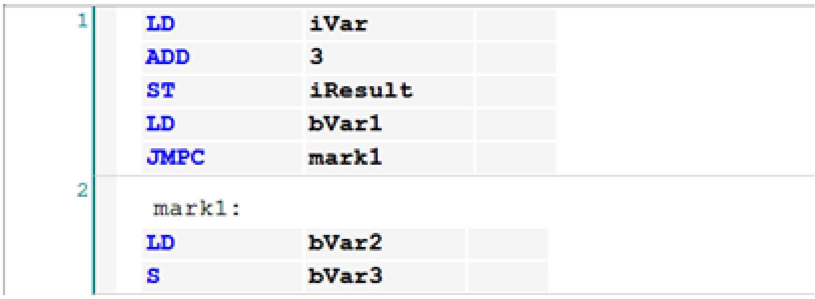
```
1  LD      iVar
   ADD     3
   ST      iResult
   LD      bVar1
   JMPC    mark1
2
```

5. From the menu bar, select **FBD/LD/IL**>**Insert label**.
"Label" will be inserted.



```
1  LD      iVar
   ADD     3
   ST      iResult
   LD      bVar1
   JMPC    mark1
2
    Label:
```

6. Enter label name "mark1" and add instructions and operands to the network (circuit) that has been inserted.



```
1   LD      iVar
    ADD     3
    ST      iResult
    LD      bVar1
    JMPC    mark1
2    mark1:
    LD      bVar2
    S       bVar3
```

Info.

- To create IL programs, the "Enable IL" check box in the Options dialog box must be selected. Refer to "7.5.2 Settings in IL Program".
- To delete a row, select the row and press the <Ctrl > key + <Delete> key simultaneously.
- Networks (circuits) can be commented out. The procedure for commenting out a network (circuit) is the same as for LD programs.

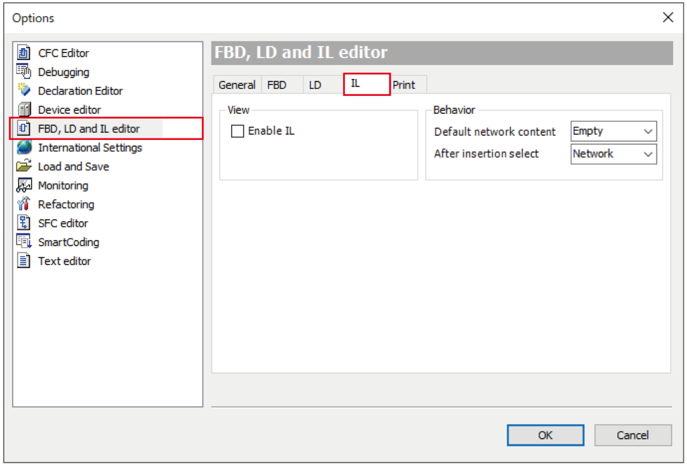
  Refer to "7.1.6 Commenting out a Network (Circuit)".
- The IL program can be converted and displayed as an LD or FBD program.

  From the menu bar, select **FBD/LD/IL**>**View** and select a post-conversion programming language.

7.5.2 Settings in IL Program

For IL programs, the following settings can be configured.

Open the Options window (by selecting **Tools**>**Options**), select the "FBD, LD and IL editors" category and specify settings in the "IL" tab window.



View
□ Enable IL

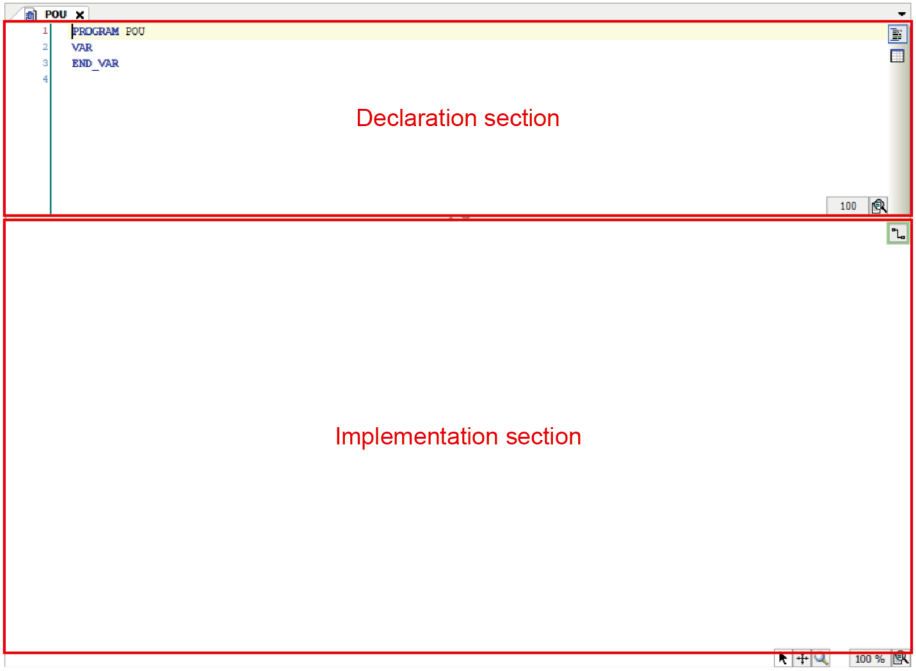
Behavior
Default network content: Empty
After insertion select: Network

Type	Item	Description
View	Enable IL	Enables the use of IL programming language. If this check box is cleared, IL will not be displayed in the list of programming languages for setting objects when a new project is created.
Behavior	Default network content	Selects whether to arrange elements and variables automatically or arrange nothing when a network (circuit) is inserted.
	After insertion select	Selects whether to select a circuit or element after a network (circuit) is inserted.

7.6 Programming in Continuous Function Chart (CFC)

Continuous Function Chart is a graphical programming language that allows programs to be created by arranging elements within the CFC editor. Elements can be freely arranged within the editor and the order of execution is determined according to the list of elements inserted.

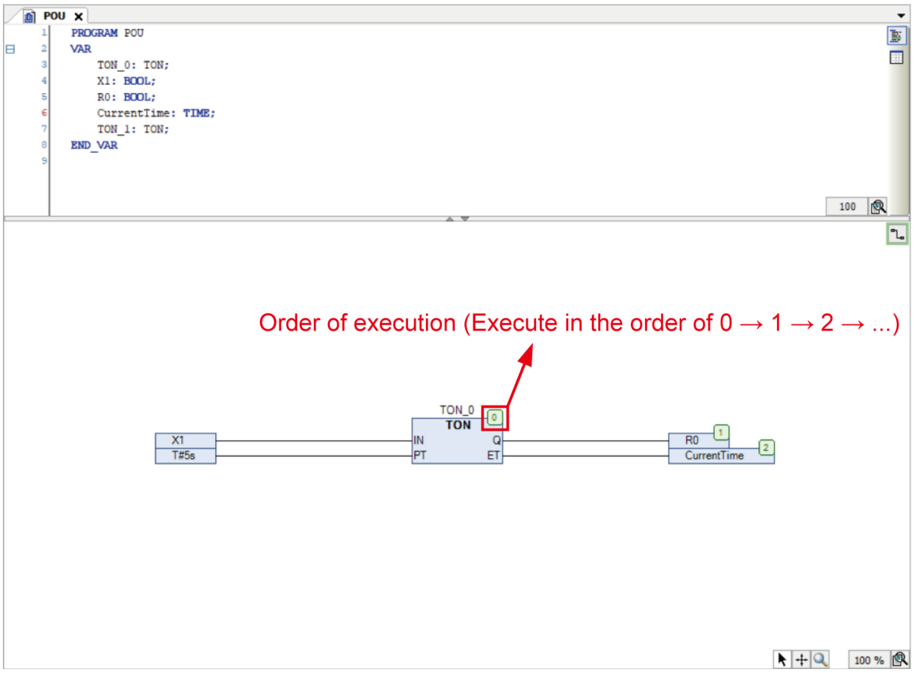
- CFC programs are classified into standard CFC programs and page-oriented CFC programs. Page-oriented CFC programs allow page-based switchover.
- To create CFC programs, POU objects for CFC programs are required. Use "Add Object" to add POU objects for CFC programs.



```
PROGRAM POU
VAR
END_VAR
```

7.6.1 Inserting and Connecting Elements

This section explains how to create a CFC program that consists of the variables and function block TON shown below.



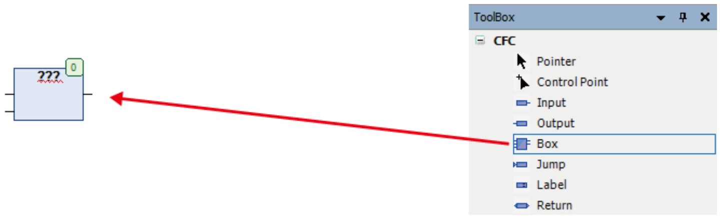
```
PROGRAM POU
VAR
    TON_0: TON;
    X1: BOOL;
    R0: BOOL;
    CurrentTime: TIME;
    TON_1: TON;
END_VAR
```

Order of execution (Execute in the order of 0 → 1 → 2 → ...)

Procedure

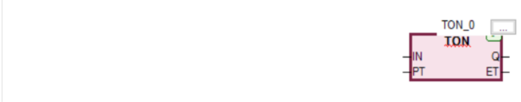
1. In Toolbox, select **CFC**>**Box** and drag the box element and drop it in the implementation section.

   The box element will be placed in the implementation section.



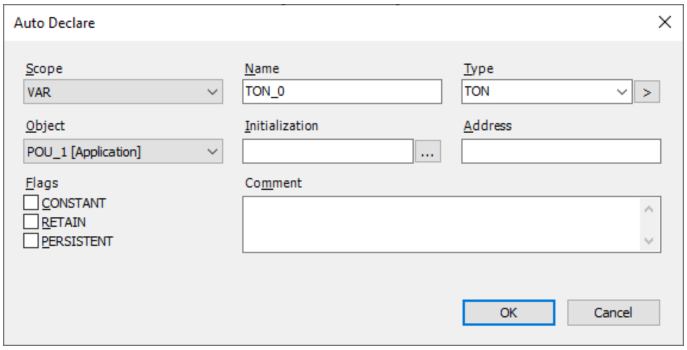
2. Enter an instance name in "???".

   The box element will be transformed into function block TON and an instance name can be entered for function block TON.



3. Select TON and enter variable TON_0, and then press the <Enter> key.

   The "Auto Declare" dialog box will be displayed.



Auto Declare

Scope: VAR

Name: TON_0

Type: TON

Object: POU_1 [Application]

Initialization:

Address:

Flags

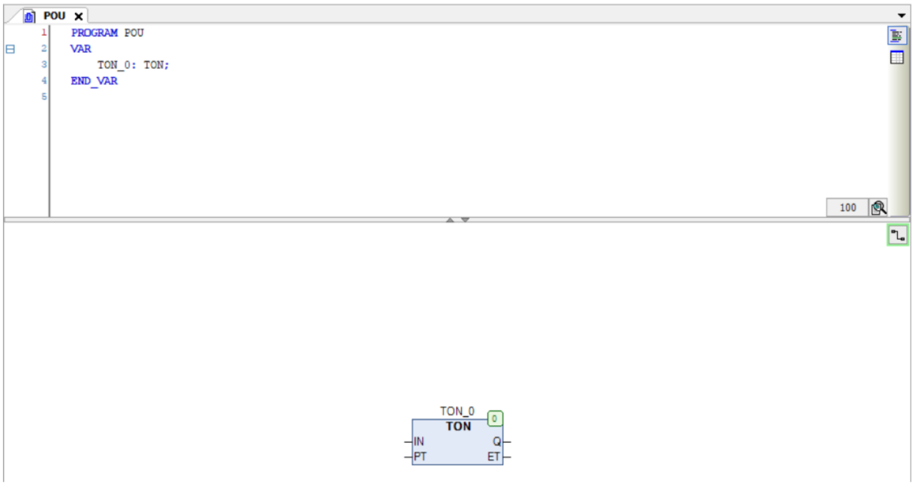
□ CONSTANT

□ RETAIN

□ PERSISTENT

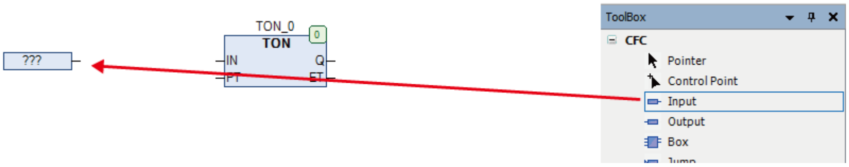
Comment:

4. Click the [OK] button.
   Variable TON_0 for function block TON will be declared in the declaration section.

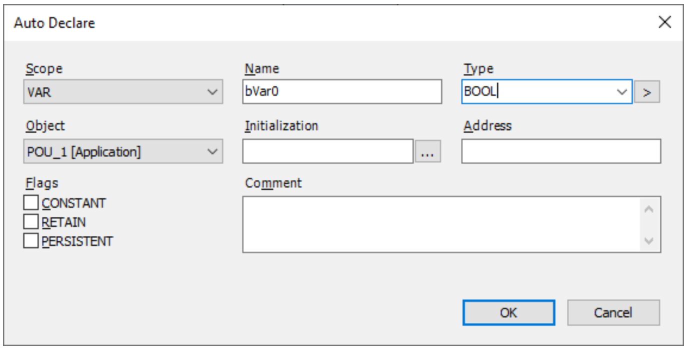


```
PROGRAM POU
VAR
    TON_0: TON;
END_VAR
```

5. In Toolbox, select **CFC>Input** and drag the input element and drop it in the implementation section.
   The input element will be placed in the implementation section.



6. Enter variable bVar0 in input element "???" and press the "Enter" key.
   The "Auto Declare" dialog box will be displayed. Declare Boolean variable bVar0.



Auto Declare

Scope: VAR

Name: bVar0

Type: BOOL

Object: POU_1 [Application]

Initialization:

Address:

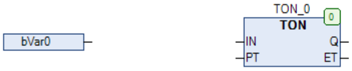
Flags

□ CONSTANT

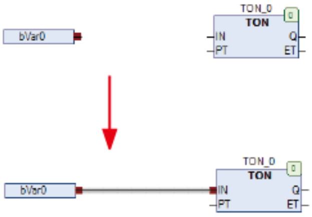
□ RETAIN

□ PERSISTENT

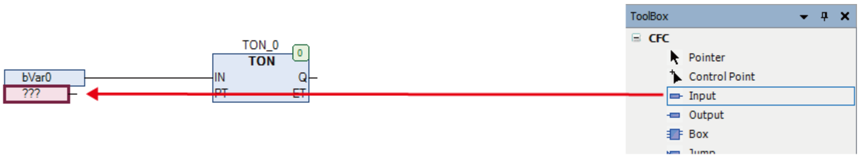
Comment:



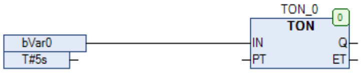
7. Select the pin on input element "variable bVar0" and drag it to "IN" on TON.
   Input element "variable bVar0" and "IN" on TON will be connected with a line.



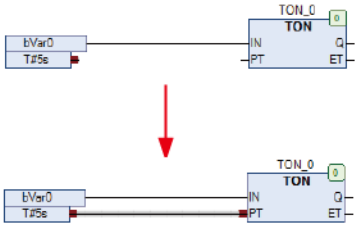
8. In Toolbox, select **CFC**>**Input** and drag the input element and drop it in the implementation section.
   The input element will be placed in the implementation section.



9. Enter "T#5s" in "???" and press the "Enter" key.

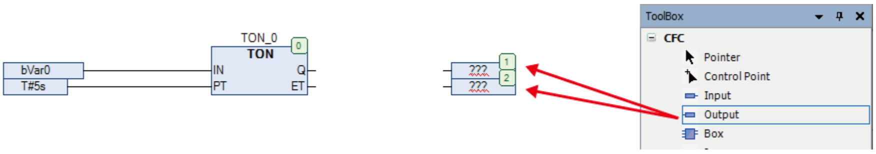


10. Select the pin on input element "T#5s" and connect it to "PT" on TON.
    Input element "T#5S" and "PT" on TON will be connected with a line.



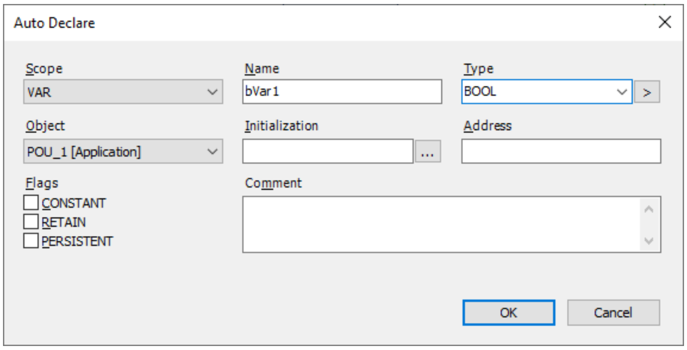
11. In Toolbox, select **CFC**>**Output** and drag the output element and drop it in the implementation section.

The output element will be placed in the implementation section.



12. Enter variable bVar1 and "CurrentTime" in output element "???" and press the "Enter" key.

The "Auto Declare" dialog box will be displayed. Declare a Boolean variable for variable bVar1 and a Time variable for variable "CurrentTime".



Auto Declare

Scope: VAR

Name: bVar1

Type: BOOL

Object: POU_1 [Application]

Initialization:

Address:

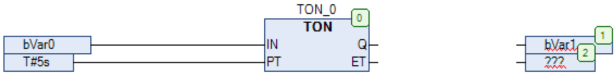
Flags

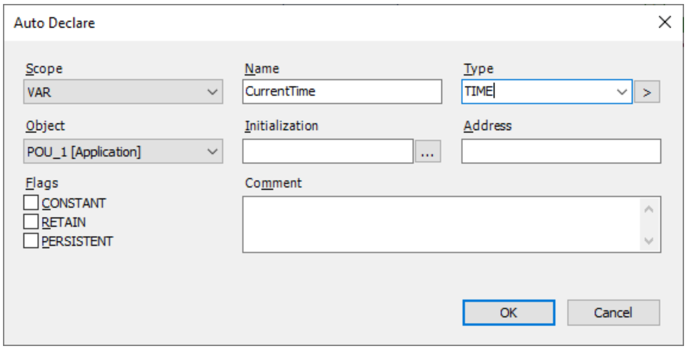
□ CONSTANT

□ RETAIN

□ PERSISTENT

Comment:





Auto Declare

Scope: VAR

Name: CurrentTime

Type: TIME

Object: POU_1 [Application]

Initialization:

Address:

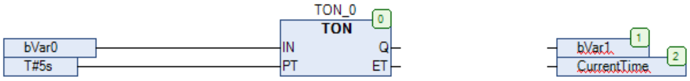
Flags

□ CONSTANT

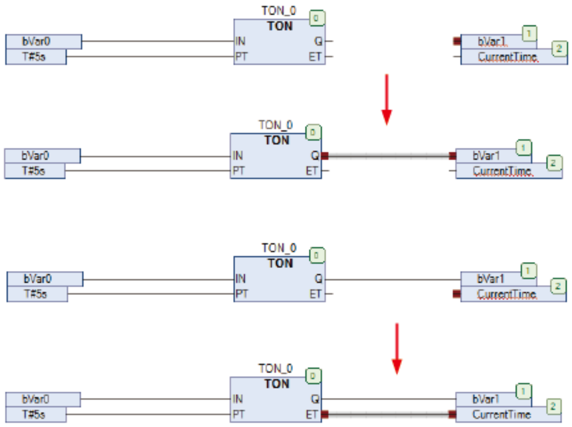
□ RETAIN

□ PERSISTENT

Comment:



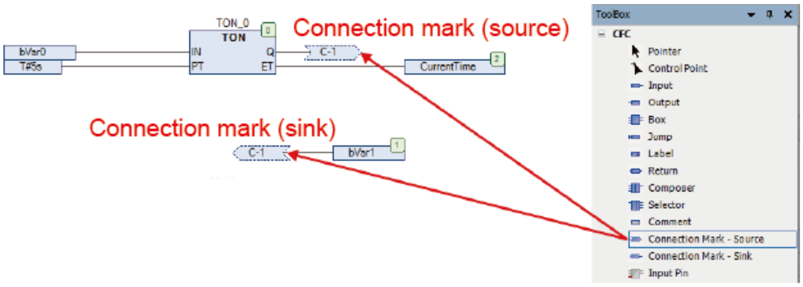
13. Connect output element "variable bVar1" and "Q" on TON and connect output element "CurrentTime" and "ET" on TON.



7.6.2 Connection Mark

Connection marks can be used to separate connection lines.

In Toolbox, select **CFC**>**Connection Mark - Source** and **CFC**>**Connection Mark - Sink**, arrange them in the implementation section, and enter the same name for them.



Info.

- You can also separate a connection line as a connection mark by selecting **CFC**>**Connection Mark** from Toolbox or clicking the ☰ icon on the toolbar, with the connection line selected . Conversely, you can restore the connection mark to a connection line by selecting **CFC**>**Connection Mark** from Toolbox or clicking the ☰ icon on the toolbar, with the connection mark selected.

7.7 Variables

Variables are declared in the main pane of POU objects for programs.

7.7.1 Standard Data Types

GM Programmer allows the following data types to be used as standard data types.

Type	Type	Range	Size (in bits)
Truth	BOOL	TRUE (1) and FALSE (0)	8
Integer	BYTE	0 to 255	8
Integer	WORD	0 to 65535	16
Integer	DWORD	0 to 4294967295	32
Integer	LWORD	0 to 2^{64}-1	64
Integer	SINT	-128 to 127	8
Integer	USINT	0 to 255	8
Integer	INT	-32768 to 32767	16
Integer	UINT	0 to 65535	16
Integer	DINT	-2147483648 to 2147483647	32
Integer	UDINT	0 to 4294967295	32
Integer	LINT	-2^{63} to 2^{63}-1	64
Integer	ULINT	0 to 2^{64}-1	64
Floating-point number	REAL	-3.402823e+38 to 3.402823e+38	32
Floating-point number	LREAL	-1.7976931348623158e+308 to 1.7976931348623158e+308	64
Character string	STRING		(Number of characters + 1) × 8
Character string	WSTRING		(Number of characters + 1) × 16
Time	TIME	0 to 4294967295	32
Time	LTIME	0 to 213503d23h34m33s709ms551us615ns	64
Time	TIME_OF_DAY(TOD)	0 (00:00:00:000) to 4294967295 (11:59:59 PM:999)	32
Date	DATE	0 (1970-01-01) to 4294967295 (2106-02-07)	32
Date and time	DATE_AND_TIME(DT)	0 (1970-01-01,00:00:00) to 4294967295 (2106-02-07,06:28:15)	32

ℹ Info.

- You can also use user-defined data types such as structure, enumeration, alias, and union. For details, refer to "7.7.6 Structure, Enumeration, Alias, and Union Data Types".

7.7.2 STRING type

The STRING type data can be used by enclosing the data with single quotation marks.

Usable characters are half-width English letters (a to z and A to Z), Arabic numerals (0 to 9), symbols, and space characters in the ASCII code.

Characters other than ASCII characters cannot be used correctly.

Possible to set the memory size when declaring variables.

Declaration example of a 35-letter character string:

str : STRING(35):= 'This is a String';

ℹ Info.

- In principle, there are no restrictions on the length of a character string. However, when using character string functions, only the length of 1 to 255 characters are processed.
  Any characters exceeding 255 characters will be truncated from the right.

7.7.3 WSTRING type

The WSTRING type data can be used by enclosing the data with double quotation marks.

Usable characters are Unicode characters.

Typical example: ASCII characters, hiragana characters, katakana characters, kanji characters, symbols, ancient characters, Korean characters, etc.

Some of them may not be used correctly.

Possible to set the memory size when declaring variables.

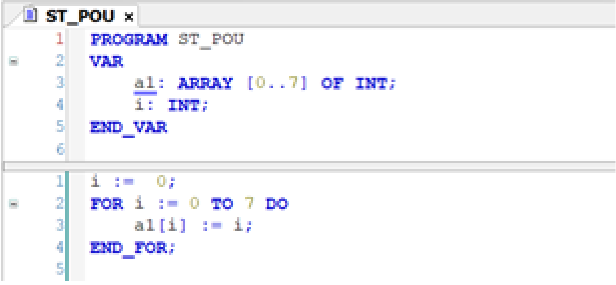
Declaration example of a 35-letter character string:

wstr : WSTRING(35):= "This is a WString";

7.7.4 Array

GM Programmer allows the use of arrays.
Using arrays enables multiple data items to be used as a single variable.
This feature is useful when variables of the same type are handled collectively.

Example: When one-dimensional array a1 with eight INT type data items is declared and used in an ST program



```
PROGRAM ST_POU
VAR
    a1: ARRAY [0..7] OF INT;
    i: INT;
END_VAR

```

```
i := 0;
FOR i := 0 TO 7 DO
    a1[i] := i;
END_FOR;

```

Info.

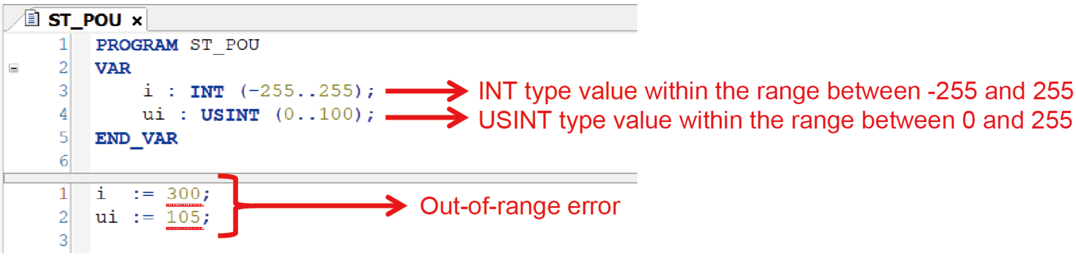
- Whether indexes are within the declared range can be automatically checked when variables in an array are accessed. Use a POU for implicit checks that checks boundaries. For details, refer to "10.4 POU for implicit checks".
- Using the array wizard for auto declaration enables array variables to be declared only by entering index and base types. For details, refer to "6.2.3 Auto Declaration".

7.7.5 Subrange Types

GM Programmer allows the use of subrange types. Subrange types allow the user to specify the range of values for standard data types.

The following is a declaration example of subrange type variables in character string format.

If an attempt is made to substitute an out-of-range value for a variable, an error will be displayed during build.



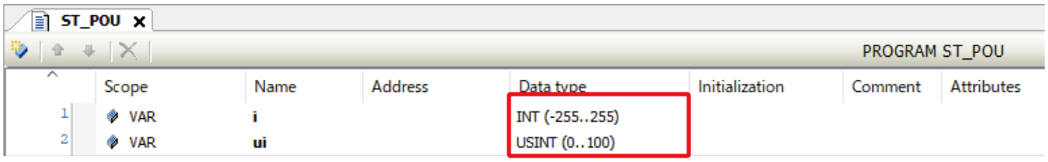
```
PROGRAM ST_POU
VAR
    i : INT (-255..255);
    ui : USINT (0..100);
END_VAR

i  := 300;
ui := 105;
```

INT type value within the range between -255 and 255

USINT type value within the range between 0 and 255

Out-of-range error

Subrange types can also be declared in table format.



	Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR	i		INT (-255..255)			
2	VAR	ui		USINT (0..100)			

Info.

- Whether values are within the declared range can be automatically checked when subrange type variables of the DINT, UDINT, LINT, or ULINT data type are accessed. Use a POU for implicit checks that checks ranges. For details, refer to "10.4 POU for implicit checks".

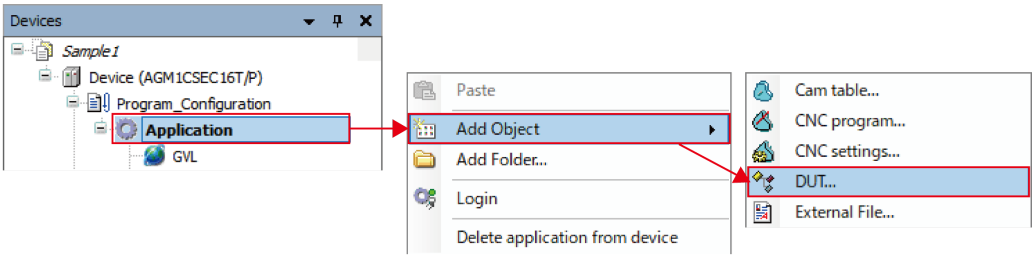
7.7.6 Structure, Enumeration, Alias, and Union Data Types

User-defined structure, enumeration, alias, and union data types can be declared using DUT objects.

When using these data types, add DUT objects to the project.

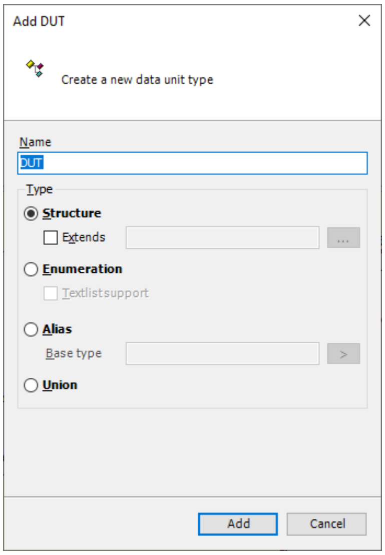
Procedure

1. Right-click the [Application] object in the navigator pane and then select **Add Object**>**DUT** from the context-sensitive menu that is displayed.



The "Add DUT" dialog box will be displayed.

The name in the "Name" field will be used as the name when the data unit type is accessed by programs.


Add DUT

Create a new data unit type

Name: DUT

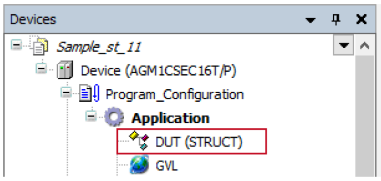
Type

- ● Structure
  - □ Extends
- ○ Enumeration
  - □ Textlistsupport
- ○ Alias
  - Base type
- ○ Union

2. Select a data type to be defined, enter the necessary information, and then click the [Add] button.

A DUT object for defining the selected data type will be added to the navigator pane.

Example: Adding a structure data type

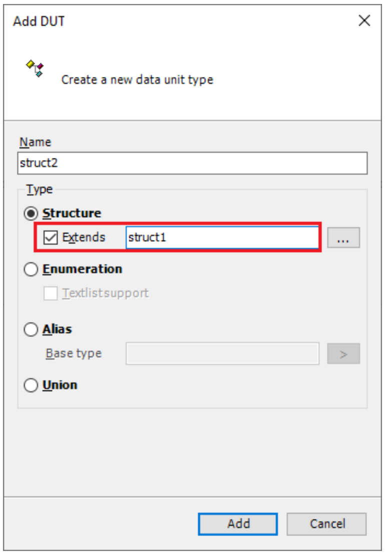


3. Select an object to be added and enter definitions in the main pane.
How to define and use each data type is described below.

Structure

The following is an example of declaring structures struct1 and struct2. struct2 is an extended structure of struct1.

To extend a structure before declaration, select the "Extends" check box in "Step 2" and enter an extension declaration.


Add DUT

Create a new data unit type

Name: struct2

Type

● Structure

☑ Extends: struct1

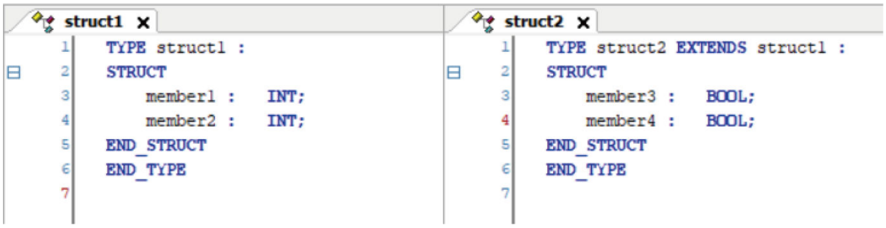
○ Enumeration

□ Textlistsupport

○ Alias

Base type:

○ Union

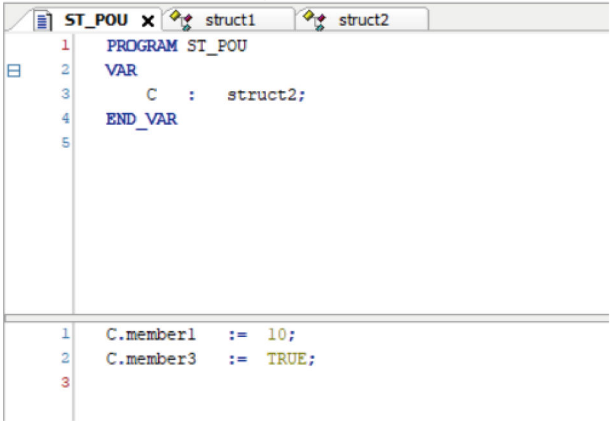


```
TYPE struct1 :
STRUCT
    member1 :   INT;
    member2 :   INT;
END_STRUCT
END_TYPE
```

```
TYPE struct2 EXTENDS struct1 :
STRUCT
    member3 :   BOOL;
    member4 :   BOOL;
END_STRUCT
END_TYPE
```

The variable declared as a variable for structure struct2 can access the members of struct1 and struct2.

Example: An ST program that accesses members of structure struct2



```
PROGRAM ST_POU
VAR
    C   :   struct2;
END_VAR
```

```
C.member1   :=  10;
C.member3   :=  TRUE;
```

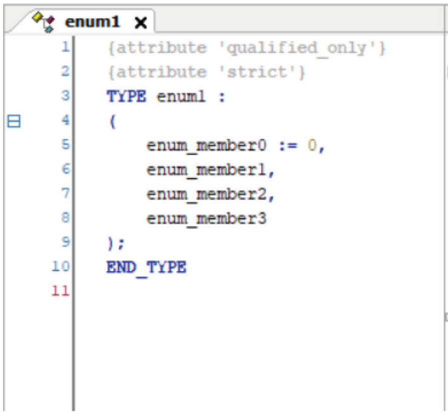
Info.

- The BIT data type can be used as members of a structure. TRUE (1) or FALSE (0) can be used as a value. The size of the BIT data type is one bit.

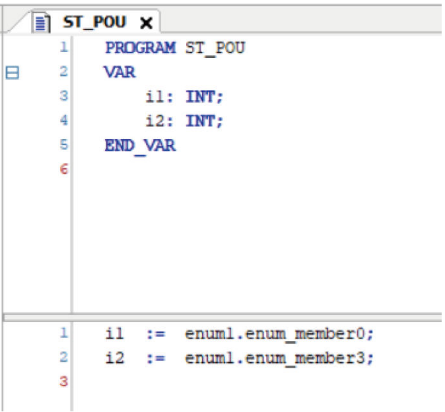
Enumeration

The following are a declaration that defines enumeration enum1 and an example of an ST program that accesses members of enum1.

"0" and "3" are substituted for variables iVar0 and iVar1, respectively.



```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE enum1 :
(
    enum_member0 := 0,
    enum_member1,
    enum_member2,
    enum_member3
);
END_TYPE
```



```
PROGRAM ST_POU
VAR
    i1: INT;
    i2: INT;
END_VAR
```

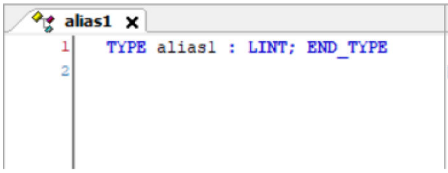
```
i1  :=  enum1.enum_member0;
i2  :=  enum1.enum_member3;
```

Alias

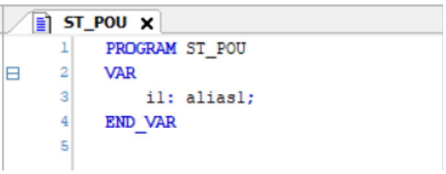
An alias can be used to assign a user-defined name as the name of a data type. Declare a variable by using an alias defined in the declaration section.

The following are an alias declaration that defines alias "alias1" for LINT and an example of a declaration section that declares variable iVar0 of the alias1 data type.

Variable iVar0 declared as the alias1 data type is handled as a variable for the LINT data type.



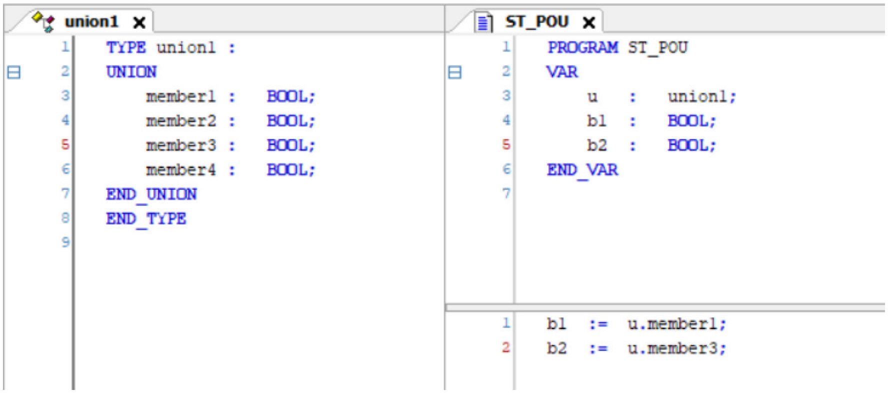
```
TYPE alias1 : LINT; END_TYPE
```



```
PROGRAM ST_POU
VAR
    i1: alias1;
END_VAR
```

Union

The following are a union declaration that defines union "union1" and an example of an ST program that accesses members of union1.



```
TYPE union1 :
UNION
    member1 :   BOOL;
    member2 :   BOOL;
    member3 :   BOOL;
    member4 :   BOOL;
END_UNION
END_TYPE
```

```
PROGRAM ST_POU
VAR
    u   :   union1;
    b1  :   BOOL;
    b2  :   BOOL;
END_VAR
```

```
b1  :=  u.member1;
b2  :=  u.member3;
```

7.7.7 Constants

GM Programmer allows the use of constants.

Constants are declared according to the following syntax.

VAR CONSTANT

Constant name:Type:=Default value;

END_VAR

Type	Constant type	Description
BOOL	BOOL	TRUE (1), FALSE (0)
Integer	Types that can be used as numerical values	Binary, octal, decimal, and hexadecimal numbers For numbers other than decimal numbers, integer constants are entered after number base and #. Examples: 14, 2#0101, 8#27, 16#34AB
Decimals and exponents	REAL / LREAL	Decimals and exponents Examples: 1.4, 2.34e+008
Time	TIME	32-bit time constants compliant with IEC 61131-3 Syntax: t#, T#, time#, TIME# Examples: T#12ms, T#12h32m24s
Time	LTIME	64-bit time constants In addition to TIME constants, the following units can be used. Microsecond: m Nanosecond: ns Syntax: LTIME# Example: LTIME#123m456ns
Time	TIME_OF_DAY	Time Syntax: tod#, TOD#, time_of_day#, TIME_OF_DAY# Example: tod#12:24:20.123
Date	DATE	Date Syntax: d#, D#, date#, DATE# Example: d#2018-01-01
Date and time	DATE_AND_TIME	Date and time Syntax: dt#, DT#, date_and_time#, DATE_AND_TIME# Example: dt#2018-01-01-07:04:13
Character string	STRING	Enclosed with single quotation marks Example: ’Hello World’
	WSTRING	Enclosed with double quotation marks Example："Hello World"

7.7.8 Object for Global Variable Declaration

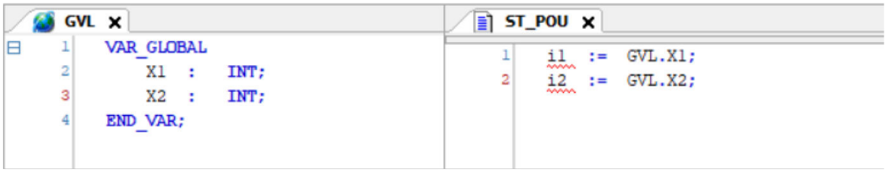
GM Programmer allows the use of global variables that can be used within the entire project.

■ Global variable list

This is an object for declaring global variables.

Variables declared in the global variable list can be accessed by using "object-name.global-variable-name" in the global variable list.

Example: When variables in global variable list "Object GVL" are accessed by an ST program



```
VAR_GLOBAL
    X1  :   INT;
    X2  :   INT;
END_VAR;
```

```
i1  :=  GVL.X1;
i2  :=  GVL.X2;
```

For details, refer to "7.7.9 Global Variables".

■ Persistent variable list

This is an object for declaring global variables that are persistent variables.

From "Add Object", select "Persistent Variables".

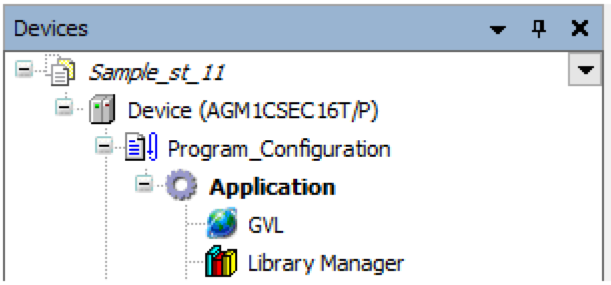
For details, refer to "7.7.10 Persistent Variables".

7.7.9 Global Variables

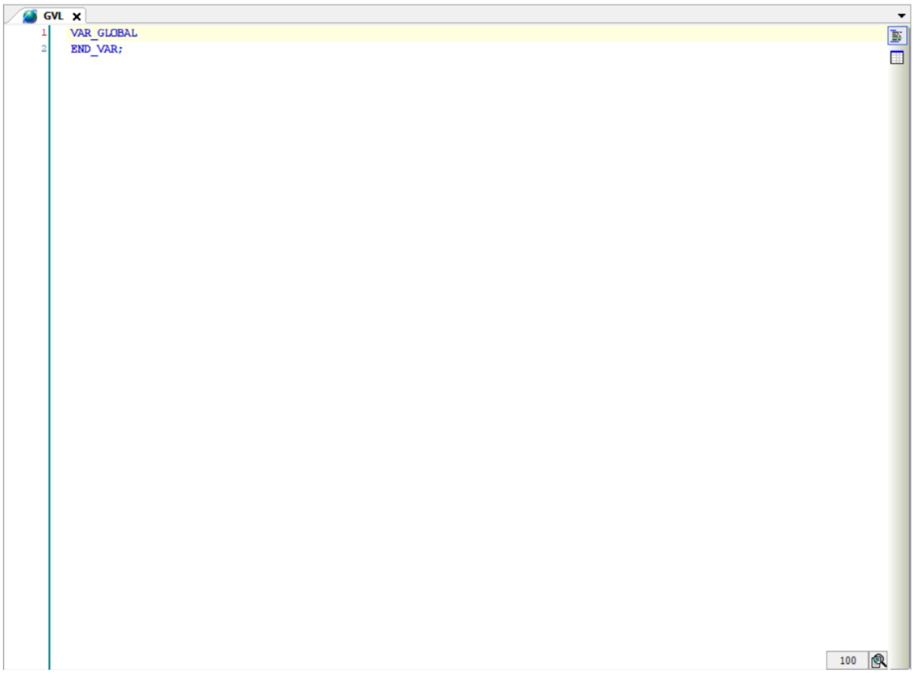
GM Programmer allows the use of global variables that can be used with all projects.
Global variables are declared within the global variable list (GVL) object.
This section explains how to declare global variables and access the declared variables.

1 2 Procedure

1. Double-click the GVL object in the navigator pane.



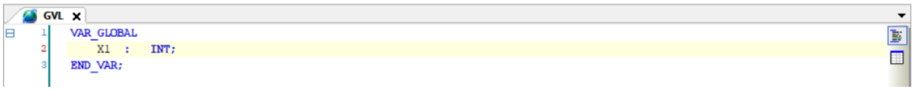
The GVL window will be displayed in the main pane.



```
VAR_GLOBAL
END_VAR;
```

2. Declare variables in the global variable list (GVL).

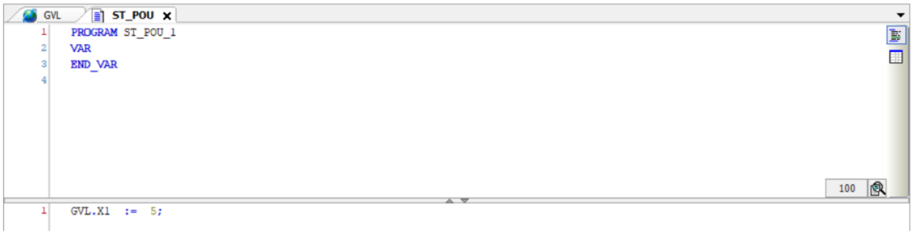
Example: Declaring global variable g_iVar0 of INT data type



```
VAR_GLOBAL
    X1  :   INT;
END_VAR;
```

The declared variable can be accessed from the program by using "name.variable name".

Example: Substituting value "5" for global variable g_iVar0



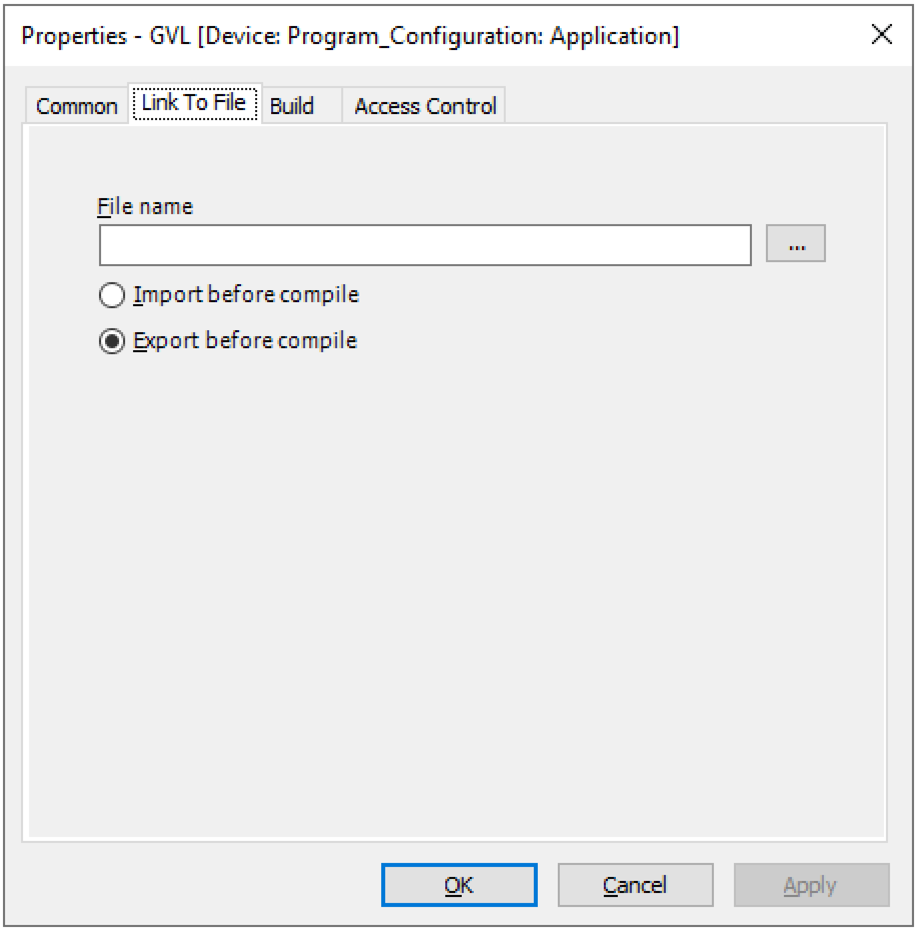
```
PROGRAM ST_POU_1
VAR
END_VAR

```

```
GVL.X1  :=  5;
```

Info.

- Variables declared before the build process can be imported and exported in XML format.

  Right-click the object in the global variable list and then select "Properties" from the context-sensitive menu that is displayed. The "Properties" dialog box will be displayed. Open the "Link To File" tab window, select either the check box for import or the check box for export, and enter the path to the file to be imported or exported in the File name field.



Properties - GVL [Device: Program_Configuration: Application]

File name

○ Import before compile

● Export before compile

7.7.10 Persistent Variables

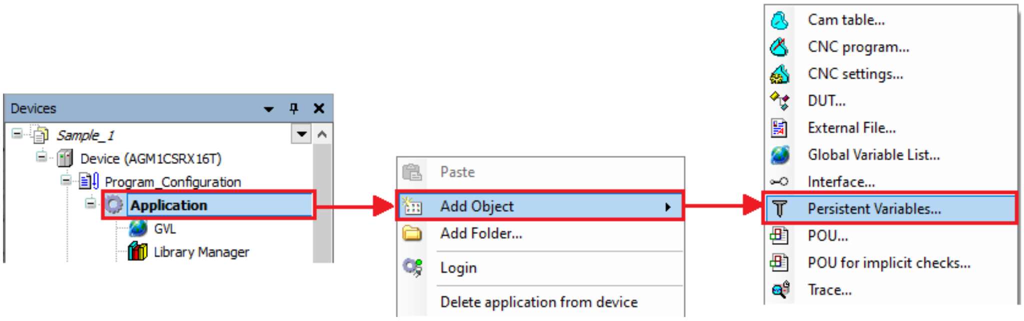
GM Programmer allows the use of global variables that are persistent variables which hold values without initializing them at the time of reset.

Persistent variables that can be used as global variables are declared within the persistent variable list object.

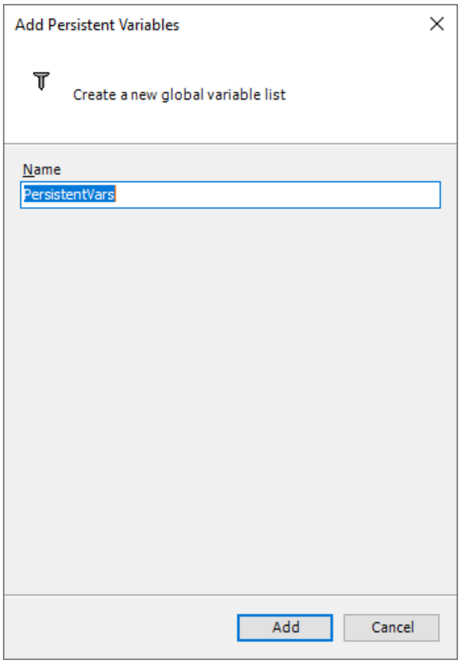
Only one persistent variable list object can be registered.

1 2 Procedure

1. Right-click the [Application] object in the navigator pane and then select **Add Object**>**Persistent Variable** from the context-sensitive menu that is displayed.



The "Add Persistent Variable" dialog box will be displayed.

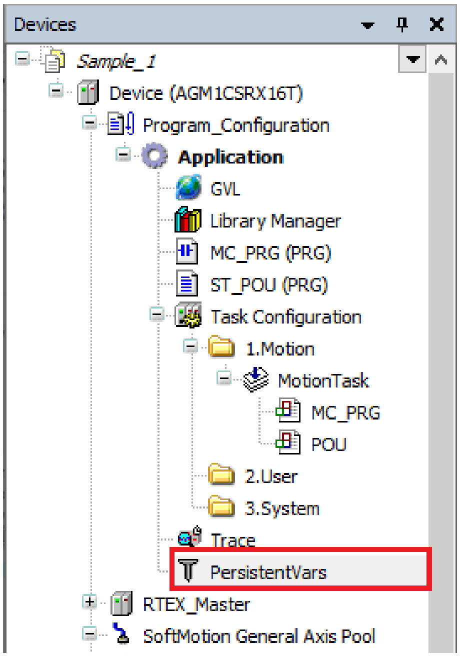


Name
PersistentVars

2. Enter a name for the persistent variable list and click the [Add] button.

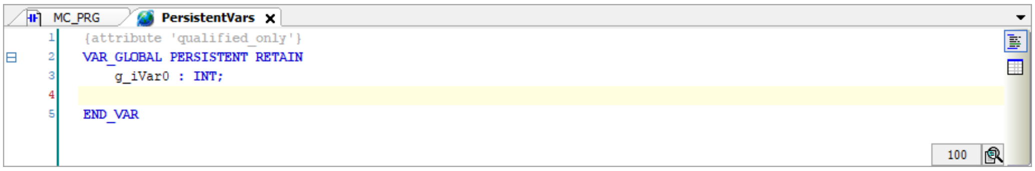
   A "Persistent variable list" object will be added to the navigator pane.

Example: Persistent variable list object with name "PersistentVars"



3. Declare variables in the persistent variable list.

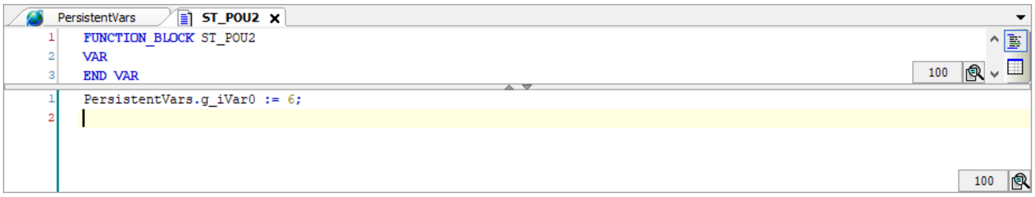
Example: Declaring global variable g_iVar0 that is a persistent variable of INT type



```
{attribute 'qualified_only'}
VAR_GLOBAL PERSISTENT RETAIN
    g_iVar0 : INT;

END_VAR
```

The declared variable can be accessed from the program by using "name.variable name".

Example: Substituting value "6" for global variable g_iVar0 that is a persistent variable



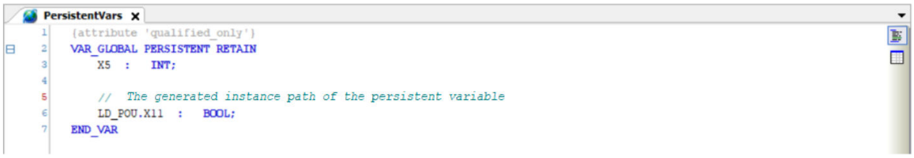
```
FUNCTION_BLOCK ST_POU2
VAR
END_VAR
```

```
PersistentVars.g_iVar0 := 6;
```

Info.

- Persistent variables used as local variables can be declared (as VAR PERSISTENT RETAIN) in the declaration section for each POU object.
- If you update the variable registered in the persistent variable, the retained data of the persistent variable will be cleared at the time of download, so perform the "15.4.5 Saving and restoring persistent variables"operation.
- Instance paths of persistent variables declared in each POU object can be added to the persistent variable list.

  With the persistent variable list declaration section selected, from the menu bar, select **Declare**>**Add All Instance Paths**.



```
{attribute 'qualified_only'}
VAR_GLOBAL PERSISTENT RETAIN
    X5  :   INT;

    //  The generated instance path of the persistent variable
    LD_POU.X11  :   BOOL;
END_VAR
```

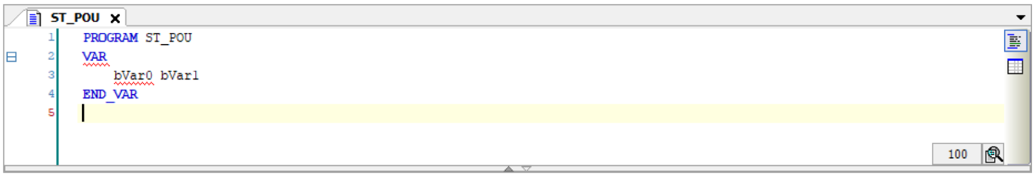
7.7.11 Short Form Function

Using the short form function in the declaration section in character string format enables variables to be declared by entering fewer characters.

■ Example: Declaring Boolean variables bVar0 and bVar1 with the short form function

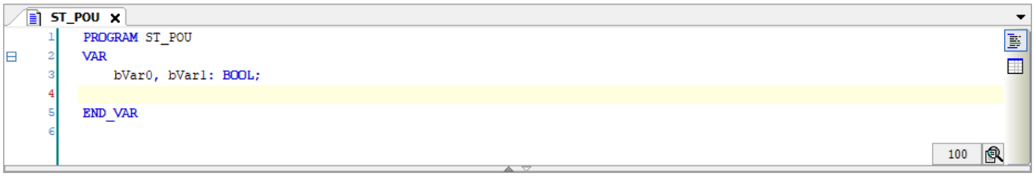
1 2 Procedure

1. Enter variables bVar0 and bVar1, and press the <Ctrl> key + <Enter> key simultaneously.
"bVar0, bVar1:BOOL;" will be entered automatically.



```
PROGRAM ST_POU
VAR
    bVar0 bVar1
END_VAR
```

<Ctrl> key＋<Enter> key



```
PROGRAM ST_POU
VAR
    bVar0, bVar1: BOOL;

END_VAR
```

The table below shows examples of input using the short form function.
Strings entered following a semicolon (;) are treated as comments.

Input in short form	Result after the <Ctrl> key + <Enter> key are pressed simultaneously
bVar0	bVar0:BOOL;
iVar0 iVar1 I 6	iVar0, iVar1: INT := 6;
strVar S 8	strVar: STRING(8)
wVar w; wVar comment	wVar: WORD; // wVar comment

8 Connecting the GM1 Controller and PC

8.1 Flow of Operation Check

This chapter explains how to connect the PC where GM Programmer is installed and the GM1 controller and operate the GM1 controller from the PC.

First, this section explains the flow of operation check for a program that is created.

1. Build(Refer to "6.7 Build")

Execute build to check a program that is created. If an error occurs, correct the program and execute build again. If code generation is executed after build is completed normally, an application will be generated.

2. Connecting to the GM1 controller(Refer to "8.4 Communication Setting")

Connect the PC where GM Programmer is installed to the GM1 controller.

3. Login(Refer to "8.8.1 Login")

Log in to the GM1 controller. When you log in to the GM1 controller, applications are downloaded to the GM1 controller.

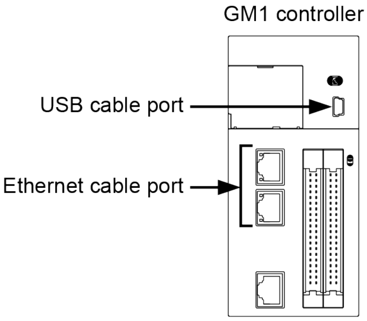
4. Debug(Refer to "9 Debug")

Execute the application and perform debugging. If there are any problems with behaviors, log out of the GM1 controller, correct the problem, and execute build again.

8.2 Connecting the GM1 Controller and PC

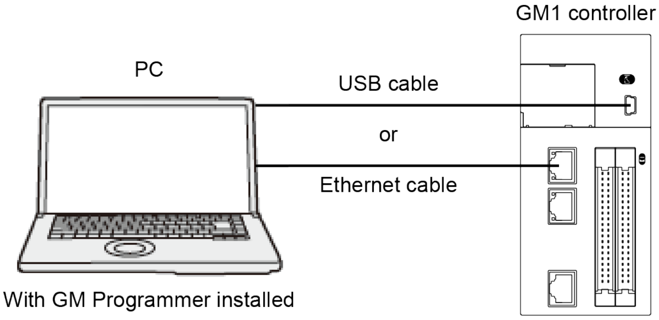
8.2.1 Selecting a Connection Port for GM Programmer

Select either LAN port connection or USB port connection.



8.2.2 Connecting the GM1 Controller and PC with a Cable

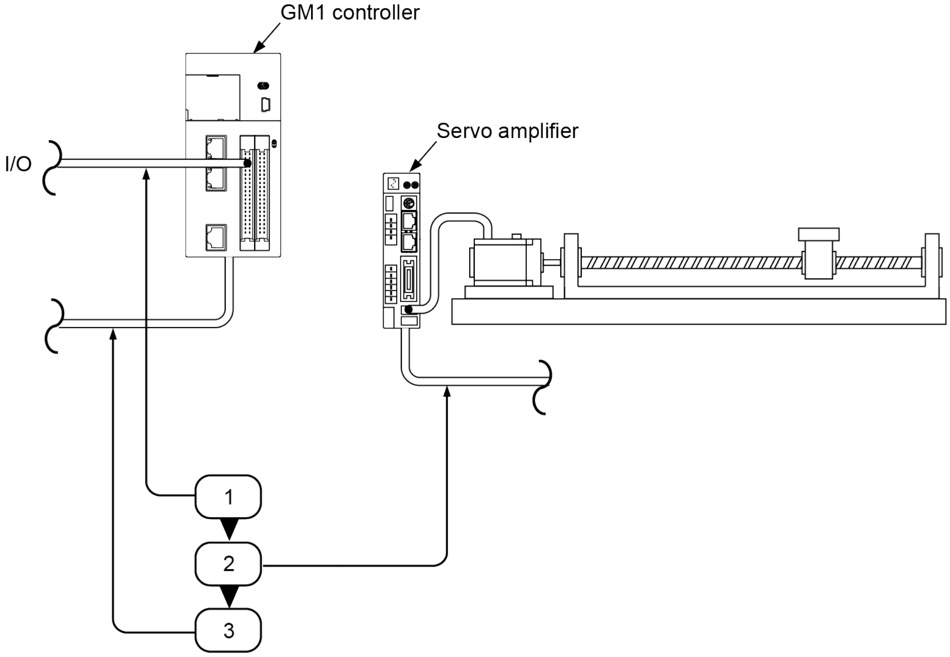
Use an Ethernet cable or USB cable to connect the GM1 controller and a PC on which the GM Programmer is installed.



8.2.3 Operation when Power is ON

When turning ON the power supply to the system incorporating the GM1 controller, consider the nature and statuses of any external devices connected to the system, and take sufficient care so that turning ON the power supply will not initiate unexpected movements.

1. Turn ON the power supplies to the I/O devices connected to the GM1 controller.
2. Turn ON the power supply to the servo amplifier.
3. Turn ON the power supply to the GM1 controller.



8.3 Operation Mode Switching

■ Switching to the RUN mode

There are the following two methods.

- Press the operation button (►) on the GM Programmer while the STOP LED is lit.
- Set the RUN/STOP switch on the GM1 Controller to RUN.

Info.

- The switch cannot be set to the RUN mode if an error that does not allow to continue operation has occurred or if an exceptional situation has occurred.

■ Switching to the STOP mode

There are the following two methods.

- Press the stop button (■) on the GM Programmer while the RUN LED is lit.
- Set the RUN/STOP switch on the GM1 Controller to STOP.

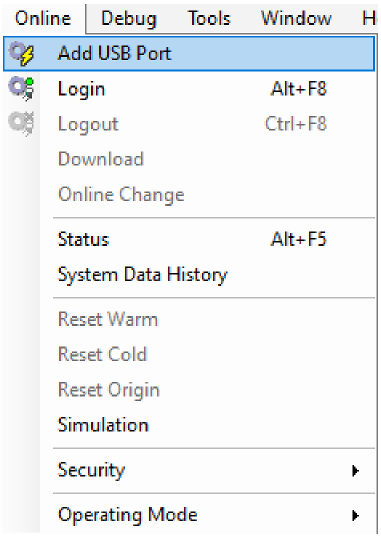
8.4 Communication Setting

8.4.1 Adding USB Ports

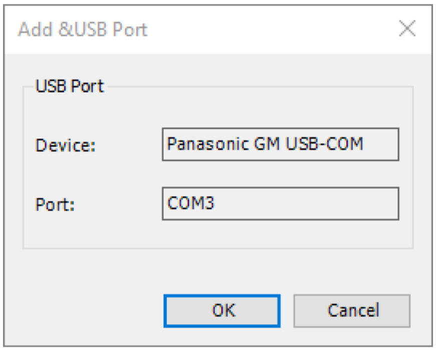
This function allows the user to set a USB port as the communication interface between a tool such as GM Programmer or PANATERM Lite for GM and the GM1 Controller.

1 2 Procedure

1. Connect the GM1 Controller and PC with a USB cable.
2. From the menu bar, select **Online**>**Add USB Port**.



The "Add USB Port" dialog box will be displayed.



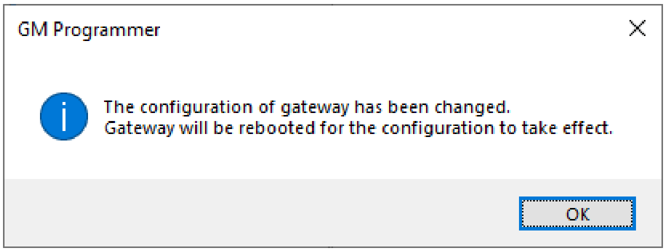
Add &USB Port

USB Port

Device: Panasonic GM USB-COM

Port: COM3

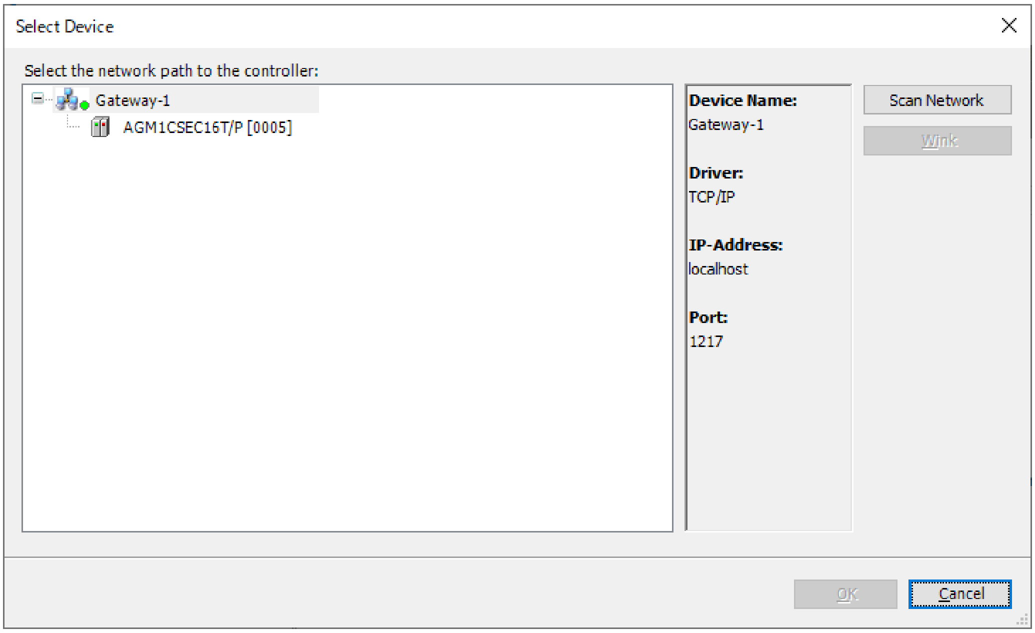
3. Click the [OK] button.
   A dialog box to restart the Gateway will be displayed.



GM Programmer

The configuration of gateway has been changed.
Gateway will be rebooted for the configuration to take effect.

4. Click the [OK] button.
   The "Select Device" dialog box will be displayed.



Select Device

Select the network path to the controller:

Gateway-1
AGM1CSEC16T/P [0005]

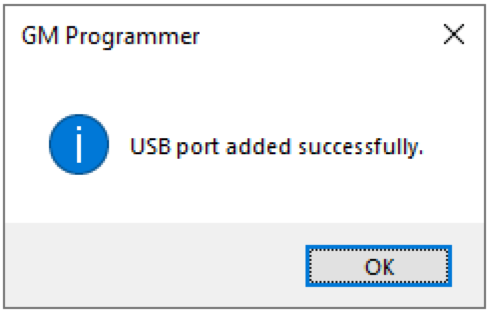
Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

5. Select a GM1 Controller to which you want to connect and click the [OK] button.
   When the connection is completed, a dialog box will be displayed to notify successful connection.



GM Programmer

USB port added successfully.

6. Click the [OK] button.
   A USB port will be added as the communication interface between the PC and GM1 Controller.

8.4.2 Setting the LAN Port

The following table shows the default network settings.

When connecting GM Programmer and the GM1 controller via a LAN port, match the network settings of the PC with those of the GM1 controller.

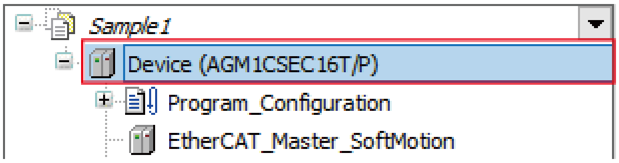
	LAN port 1	LAN port 2
IP address	192.168.1.5	192.168.2.5
Subnet mask	255.255.255.0	255.255.255.0
Default gateway	192.168.1.1	0.0.0.0

Set the IP addresses of LAN port 1 and LAN port 2 so that their network (subnet) addresses are different.

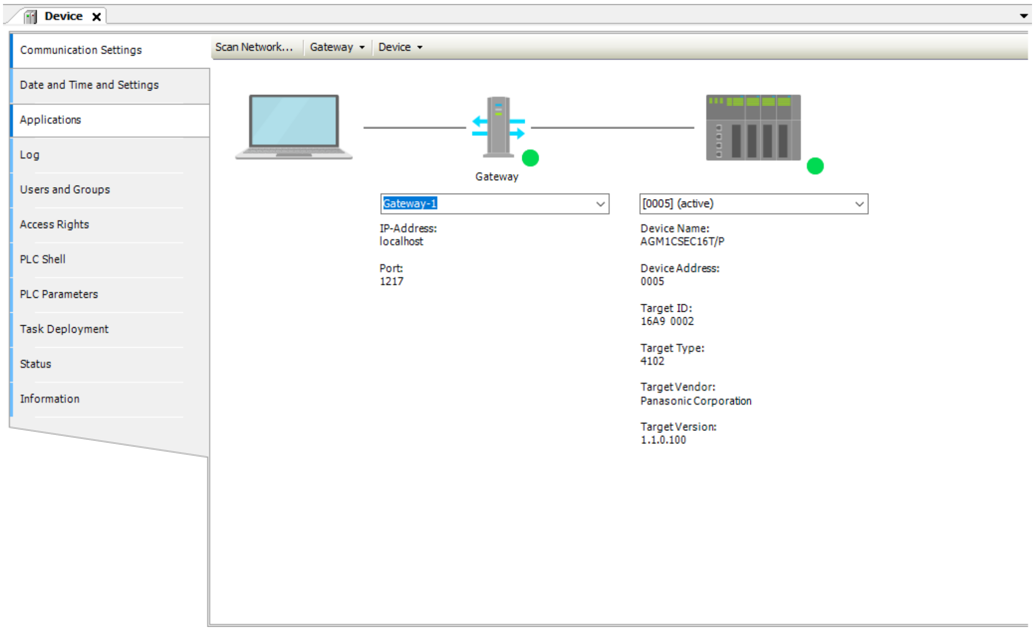
Network settings can be changed using the [PLC parameters] tab of the "Device" object in the navigator pane, as described below.

Procedure

1. Double-click the [Device] object in the navigator pane.



The Device setting window will be displayed.



Gateway-1

IP-Address:
localhost

Port:
1217

[0005] (active)

Device Name:
AGM1CSEC16T/P

Device Address:
0005

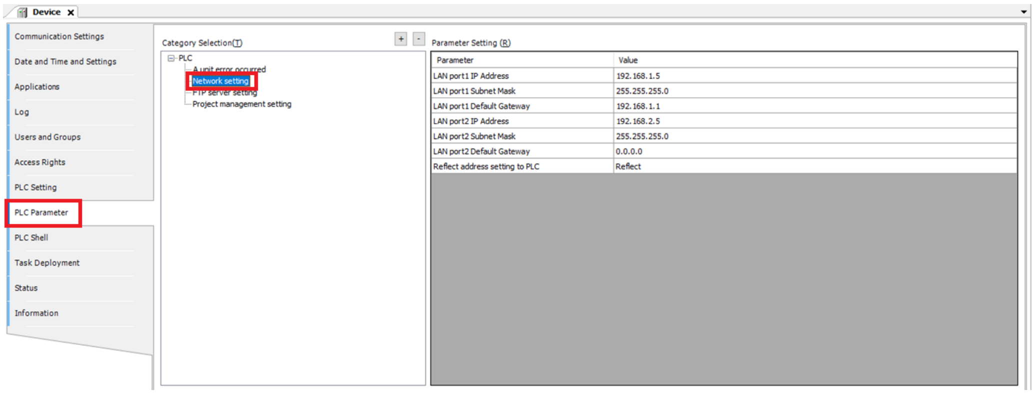
Target ID:
16A9 0002

Target Type:
4102

Target Vendor:
Panasonic Corporation

Target Version:
1.1.0.100

2. Click the "PLC parameters" tab in the Device setting window.



Parameter	Value
LAN port1 IP Address	192.168.1.5
LAN port1 Subnet Mask	255.255.255.0
LAN port1 Default Gateway	192.168.1.1
LAN port2 IP Address	192.168.2.5
LAN port2 Subnet Mask	255.255.255.0
LAN port2 Default Gateway	0.0.0.0
Reflect address setting to PLC	Reflect

3. Set the IP address, subnet mask, and default gateway for each LAN port.

4. Download the project to the GM1 controller.

Info.

- If you change the network settings for the LAN port that connects the GM1 controller and GM Programmer with a LAN cable, the connection will be temporarily disrupted.

8.5 Connecting to the GM1 Controller

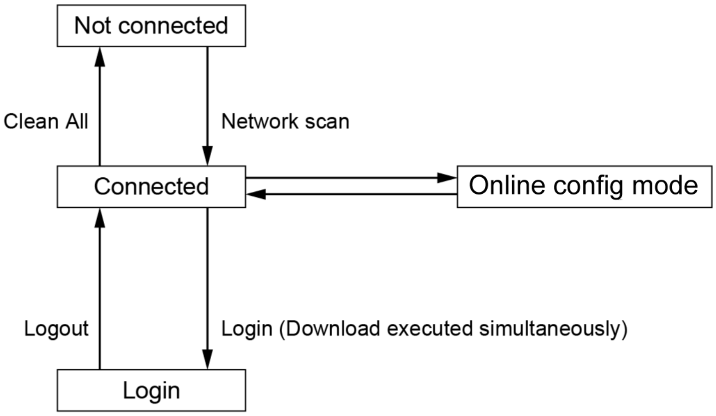
Connect the PC where the GM Programmer is installed to the GM1 Controller.

The connection status of the PC includes "Connected", "Connection as a device user", "Login", and "Online config mode".

Depending on the connection status, operations that can be executed are different.

If the Controller is provided with a device user registration, connection must be made as the device user.

■ Without device user registration



List of available GM1 Controller operations

Function	Not connected	Connected	Login	Online config mode
Setting / acquiring Controller information	×	○	○(Note 1)	×
Application management	×	×(Note 2)	○	×
Reset	×	×(Note 3)	○	×(Note 4)
Security	×	×	○	○
Debug	×	×	○	×
Commissioning	×	×	×	○

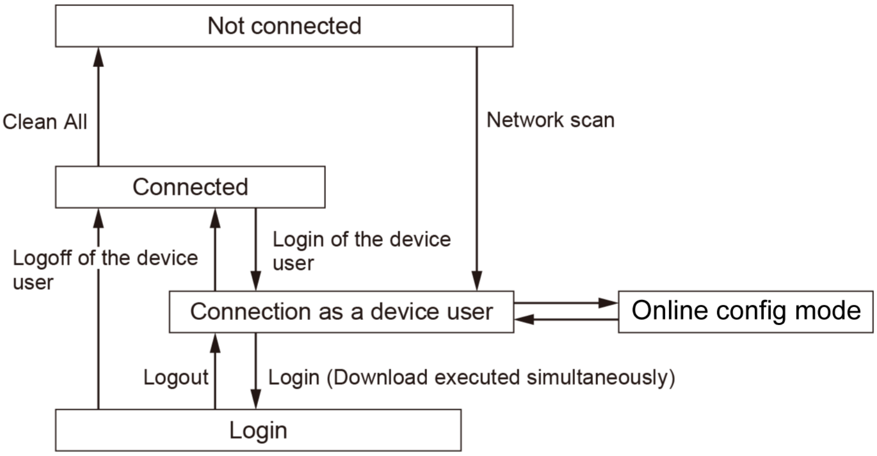
(Note 1) Not possible to operate the PLC Shell.

(Note 2) Possible to upload the source.

(Note 3) Possible to reset the device (PLC initialization) or to delete device application from the device.

(Note 4) Possible to reset the device (PLC initialization).

■ With device user registration



List of available GM1 Controller operations

Function	Not connected	Connected	\as a device user	Login	Online config mode
Setting / acquiring Controller information	×	×	○	○(Note 1)	×
Application management	×	×	×(Note 2)	○	×
Reset	×	×	×(Note 3)	○	×(Note 4)
Security	×	×	○(Note 5)	○	○
Debug	×	×	×	○	×
Commissioning	×	×	×	×	○

(Note 1) Not possible to operate the PLC Shell.

(Note 2) Possible to upload the source.

(Note 3) Possible to reset the device (PLC initialization) or to delete device application from the device.

(Note 4) Possible to reset the device (PLC initialization).

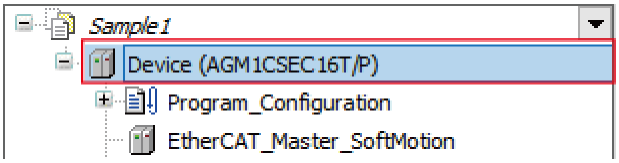
(Note 5) Addition of the device user, changing the password for the device user, or deletion of the device user cannot be made if the user of the Device Editor is not synchronized with "Synchronization" of the group tab.

8.6 Setting Time

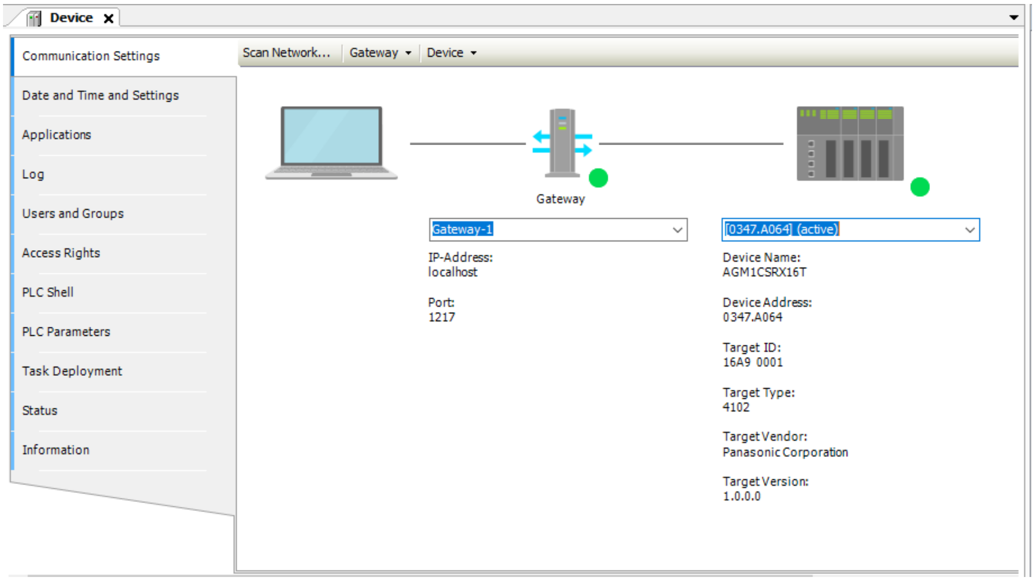
After executing the Network Scan function, set time for the GM1 controller.
You can enter date and time directly or by getting date and time from the PC.

Procedure

1. Double-click the [Device] object in the navigator pane.



The Communication Settings window for the device will be displayed.



Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

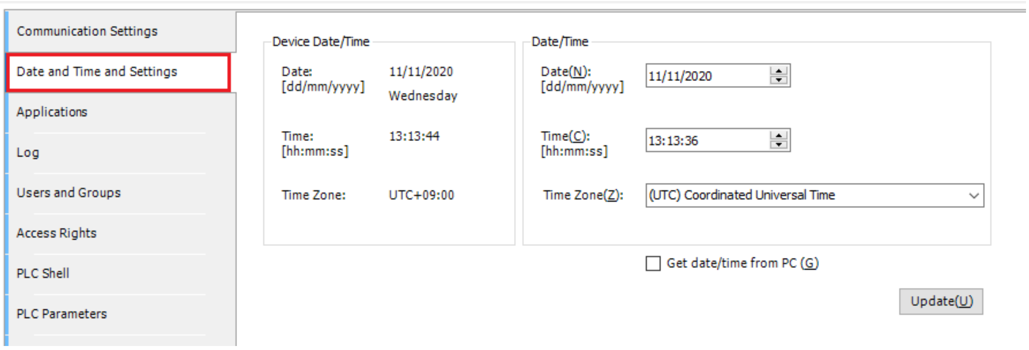
Target ID:
16A9 0001

Target Type:
4102

Target Vendor:
Panasonic Corporation

Target Version:
1.0.0.0

2. Click the "Date and Time and Settings" tab.
The Date and Time Settings window will be displayed.
The "Device Date and Time" section displays the current date and time of the GM1 controller.



Communication Settings
Date and Time and Settings
Applications
Log
Users and Groups
Access Rights
PLC Shell
PLC Parameters

Device Date/Time

Date: [dd/mm/yyyy] 11/11/2020 Wednesday

Time: [hh:mm:ss] 13:13:44

Time Zone: UTC+09:00

Date/Time

Date(N): [dd/mm/yyyy] 11/11/2020

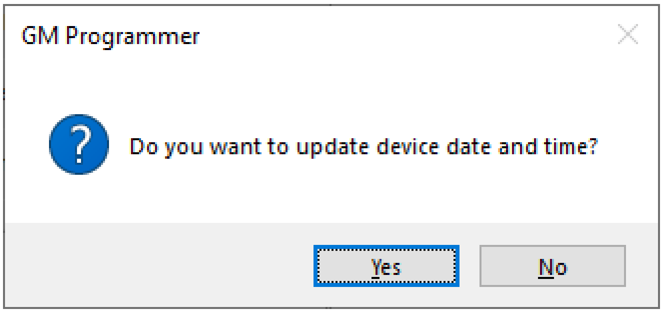
Time(C): [hh:mm:ss] 13:13:36

Time Zone(Z): (UTC) Coordinated Universal Time

□ Get date/time from PC (G)

3. Change the date, time, and time zone in the "Date and Time" section or select the "Get date / time from PC" check box, and click the [Update] button.

A confirmation message will be displayed.



GM Programmer

Do you want to update device date and time?

4. Click the [Yes] button.

The date and time of the GM1 controller will be updated.

8.7 Other Settings

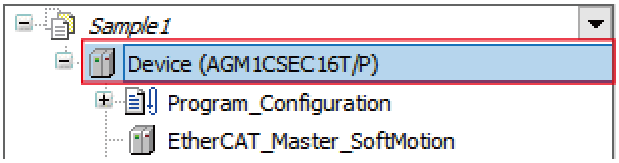
For the connected device, configure settings described in "Change Device Name", "Sending Echo services", "Device preference management", and "Confirmed Online Mode".

8.7.1 Changing the Device Name

This section explains how to change the name of the device connected via "Network Scan".

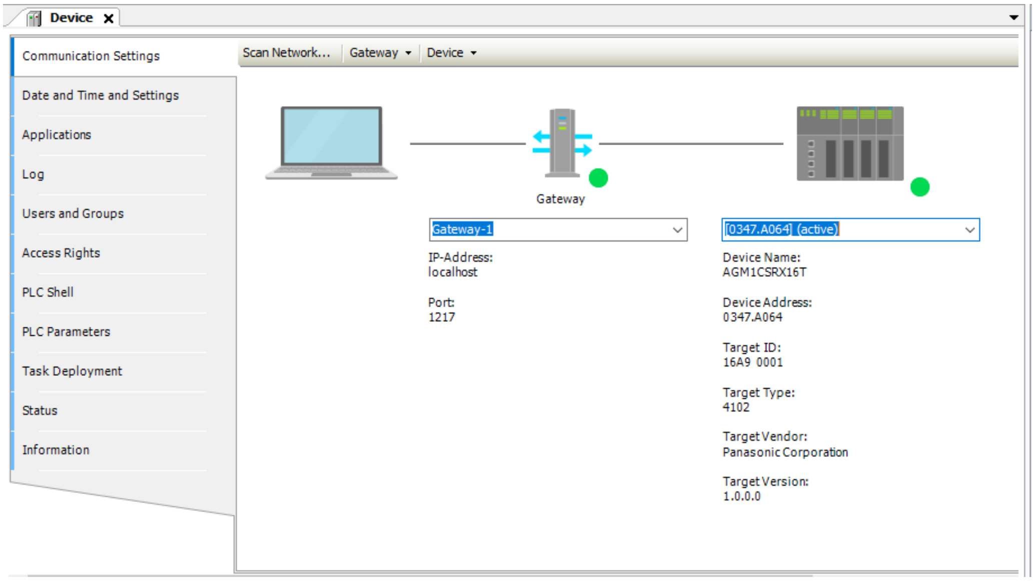
1 2 Procedure

1. Double-click the "Device" object in the navigator pane.



Sample1
Device (AGM1CSEC16T/P)
Program_Configuration
EtherCAT_Master_SoftMotion

The setting pane will be displayed in the main pane.



Device ×

Communication Settings
Date and Time and Settings
Applications
Log
Users and Groups
Access Rights
PLC Shell
PLC Parameters
Task Deployment
Status
Information

Scan Network... Gateway ▾ Device ▾

Gateway

Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

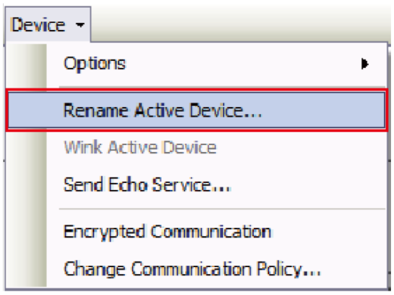
Target ID:
16A9 0001

Target Type:
4102

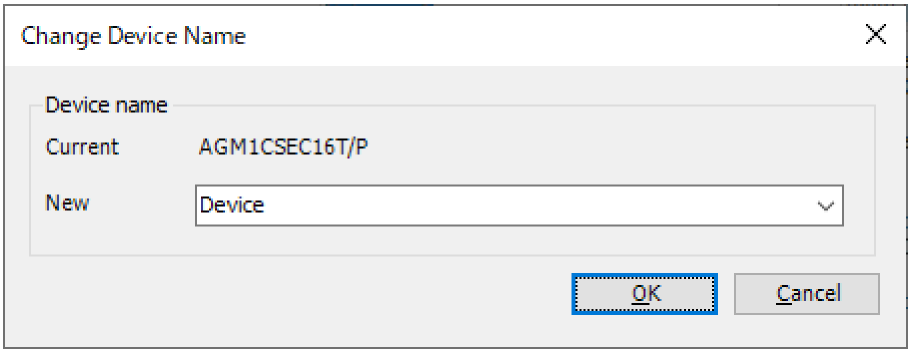
Target Vendor:
Panasonic Corporation

Target Version:
1.0.0.0

2. Select "Rename Active Device" from the "Device" menu.
The "Change Device Name" dialog box will be displayed.



3. Enter a new device name and click the [OK] button.
The device name will be changed.


Change Device Name

Device name

Current AGM1CSEC16T/P

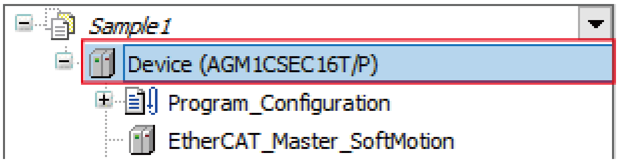
New Device

8.7.2 Sending Echo Services

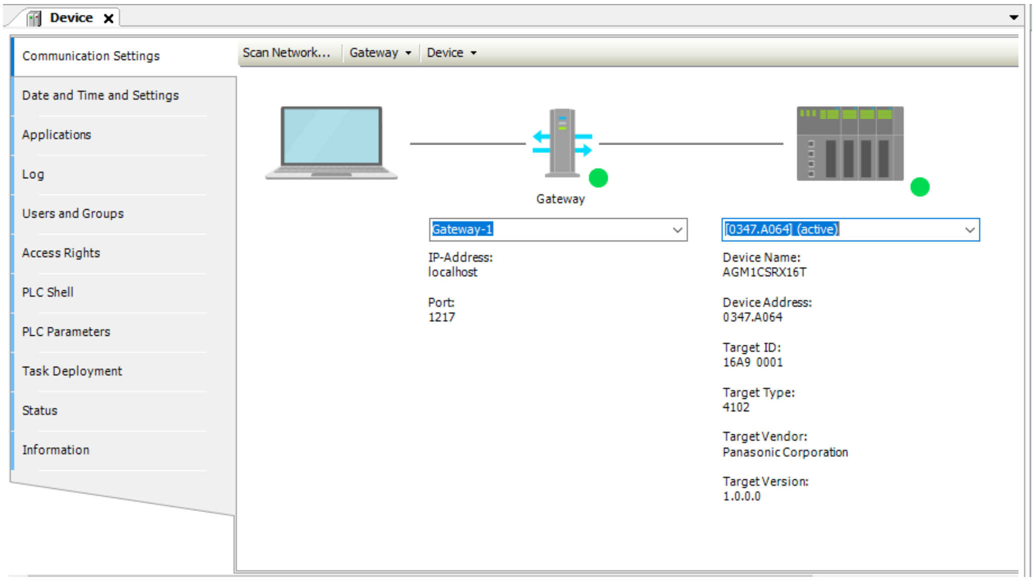
This section explains how to send Echo services to the device connected via "Network Scan". Echo services can be used to conduct a network test.

1 2 Procedure

1. Double-click the "Device" object in the navigator pane.



The setting pane will be displayed in the main pane.



Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

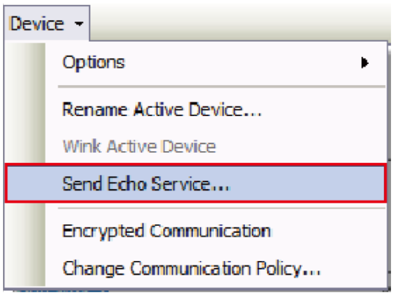
Target ID:
16A9 0001

Target Type:
4102

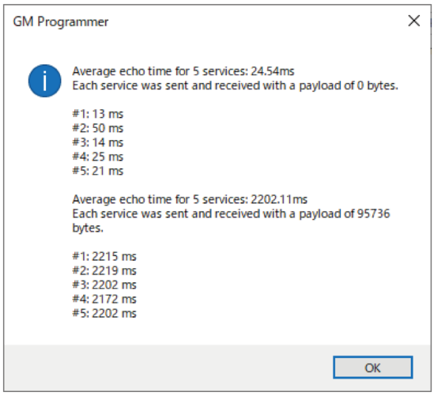
Target Vendor:
Panasonic Corporation

Target Version:
1.0.0.0

2. Select "Send Echo Service" from the "Device" menu.



The results of five transmissions with no data size followed by five transmissions with data size will be displayed.


GM Programmer

Average echo time for 5 services: 24.54ms
Each service was sent and received with a payload of 0 bytes.

#1: 13 ms
#2: 50 ms
#3: 14 ms
#4: 25 ms
#5: 21 ms

Average echo time for 5 services: 2202.11ms
Each service was sent and received with a payload of 95736 bytes.

#1: 2215 ms
#2: 2219 ms
#3: 2202 ms
#4: 2172 ms
#5: 2202 ms

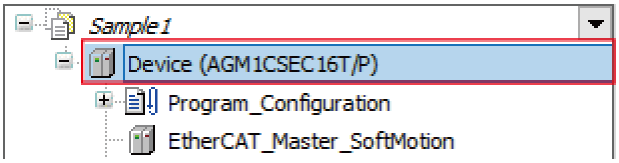
8.7.3 Device Preference Management

Devices connected via "Network Scan" can be managed by registering them as favorite devices.

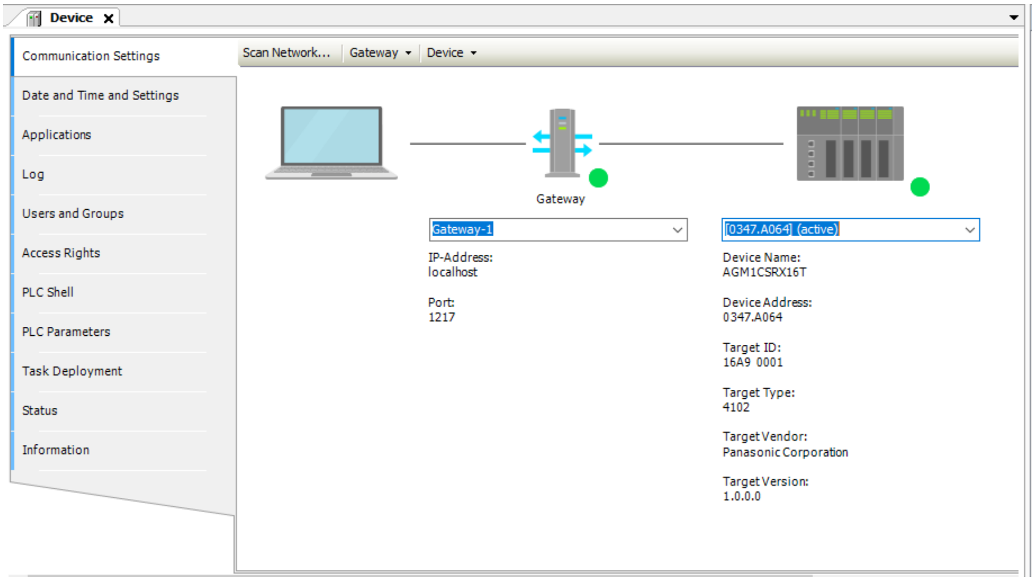
Devices registered as favorite devices will be displayed in the device selection list in the Communication Settings window.

1 2 Procedure

1. Double-click the "Device" object in the navigator pane.



The setting pane will be displayed in the main pane.



Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

Target ID:
16A9 0001

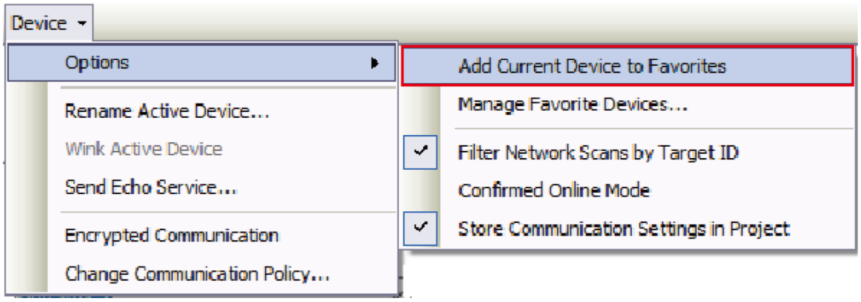
Target Type:
4102

Target Vendor:
Panasonic Corporation

Target Version:
1.0.0.0

2. From the "Device" menu, select "Options" and then "Add Current Device to Favorites".

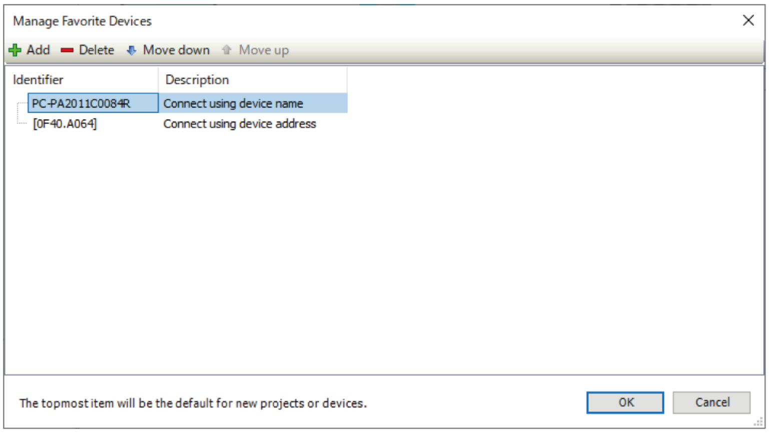
The connected device will be registered as a favorite device.


Device
Options
Rename Active Device...
Wink Active Device
Send Echo Service...
Encrypted Communication
Change Communication Policy...
Add Current Device to Favorites
Manage Favorite Devices...
✓ Filter Network Scans by Target ID
Confirmed Online Mode
✓ Store Communication Settings in Project

Info.

- Devices registered as favorite devices can be viewed in the "Manage Favorite Devices" dialog box.

  The "Manage Favorite Devices" dialog box can be displayed by selecting "Options" and then "Manage Favorite Devices" from the "Device" menu.



Identifier	Description
PC-PA2011C0084R	Connect using device name
[0F40.A064]	Connect using device address

The topmost item will be the default for new projects or devices.

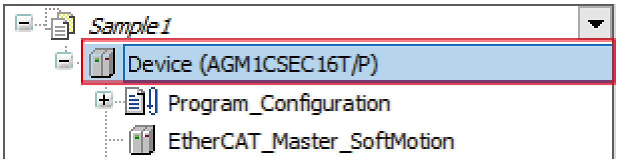
8.7.4 Confirmed Online Mode

A confirmation message can be displayed when an attempt is made to implement the following actions.

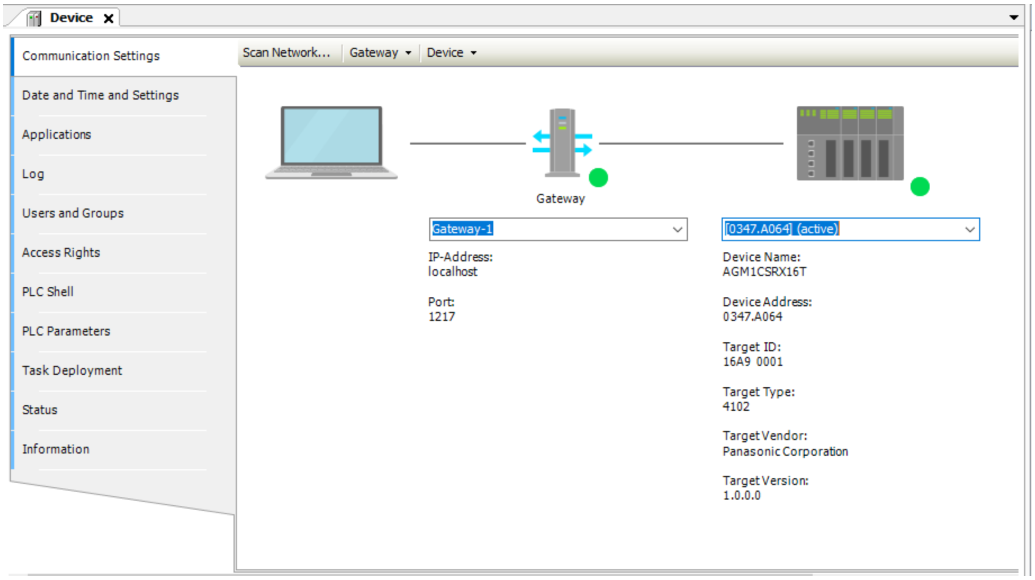
- Login
- Operation
- Stop
- Single Cycle
- Force Values
- Write Values
- Unforce Values

1 2 Procedure

1. Double-click the "Device" object in the navigator pane.



The setting pane will be displayed in the main pane.



Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

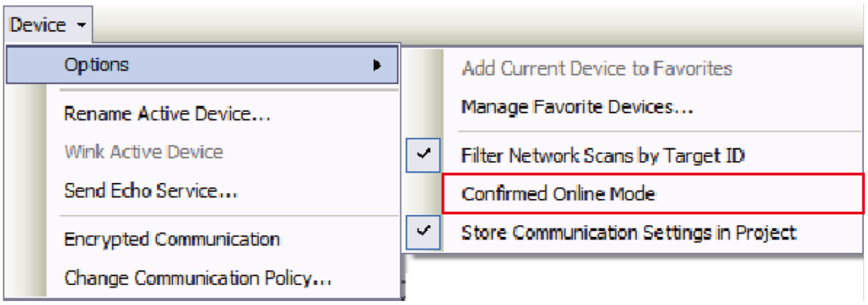
Target ID:
16A9 0001

Target Type:
4102

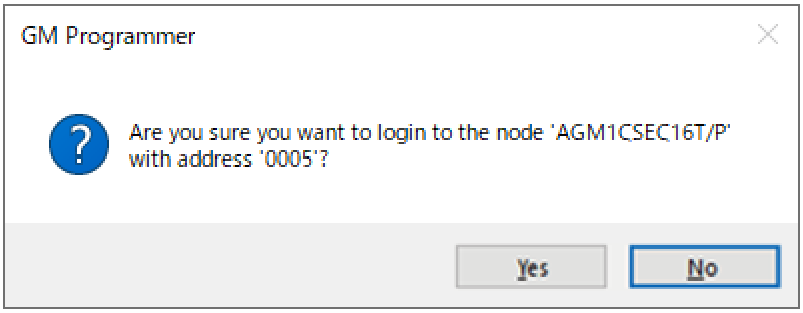
Target Vendor:
Panasonic Corporation

Target Version:
1.0.0.0

2. From the "Device" menu, select "Options" and then "Confirmed Online Mode".


Device
Options
Rename Active Device...
Wink Active Device
Send Echo Service...
Encrypted Communication
Change Communication Policy...
Add Current Device to Favorites
Manage Favorite Devices...
✓ Filter Network Scans by Target ID
Confirmed Online Mode
✓ Store Communication Settings in Project

When an attempt is made to log in with "Confirmed Online Mode" selected, the following message is displayed.


GM Programmer

Are you sure you want to login to the node 'AGM1CSEC16T/P' with address '0005'?

8.8 Login / Logout

GM Programmer allows the user to log in to the GM1 controller.

- During login, the application and source code generated by code generation are downloaded to the GM1 controller.

- The combination of the application and source code downloaded to the GM1 controller differs according to the operations shown in the table below.

○: Downloaded

×: Not downloaded

Operation	Boot application	Source code
Login	○	○
Initial download	○	○
Downloading after changing the program "Update boot project" check box: Selected	○	○
Downloading after changing the program "Update boot project" check box: Cleared	×	×
Downloading after changing the project	○	○
Online change "Update boot project" check box: Selected	○	○
Online change "Update boot project" check box: Cleared	×	×
Generating a boot application	○	○

After logging in to the GM1 controller, you can perform debug operations such as starting or stopping the GM1 controller.

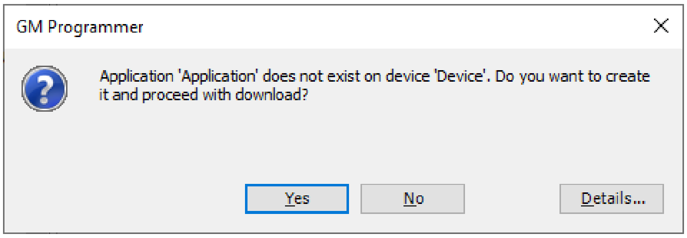
8.8.1 Login

GM Programmer allows the user to log in to the GM1 Controller. When "Login" is executed, applications are downloaded to the GM1 Controller.

1 2 Procedure

1. From the menu bar, select **Online**>**Login**, or press the <Alt> key and the <F8> key simultaneously.

   A confirmation message will be displayed, asking whether to download the applications to the GM1 controller (device).



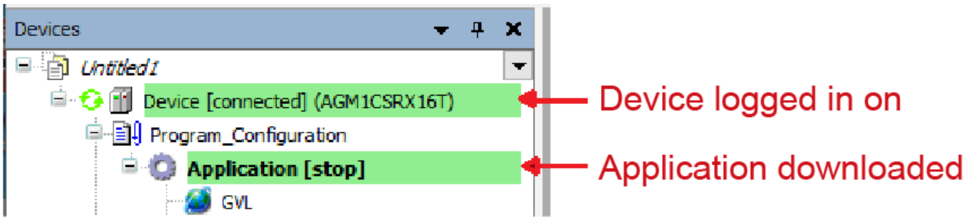
GM Programmer

Application 'Application' does not exist on device 'Device'. Do you want to create it and proceed with download?

2. Click [Yes].

The applications will be downloaded to the GM Programmer at the same time as you log in to the GM1 Controller (device).

"connected" will be displayed at the [Device] object in the navigator pane and the status of the downloaded applications will be displayed.



Info.

- You can also log in by clicking the login button on the toolbar.
- If you log in again after the applications have been downloaded, the confirmation message will not be displayed.

8.8.2 Logout

This function allows the user to log out from the device to which the user logged in.

1 2 Procedure

1. From the menu bar, select **Online**>**Logout**, or press the <Ctrl > + <F8> key simultaneously. You will be logged out.

Info.

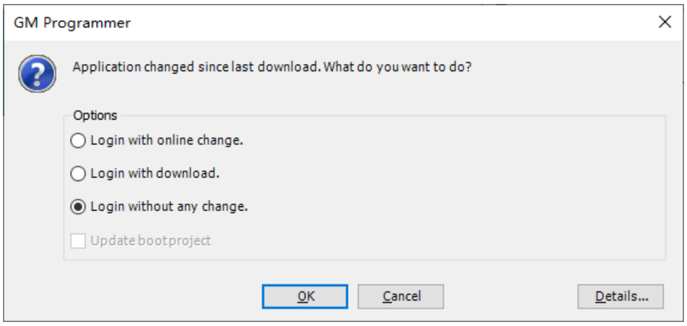
- You can also log out by clicking on the toolbar.

8.8.3 Download

After changing programs and logging in without executing online change, you can download applications while being logged in.
The boot application is also updated during download.

1 2 Procedure

1. When executing "login", select "Login without any change" and click the [OK] button.



GM Programmer

Application changed since last download. What do you want to do?

Options

○ Login with online change.

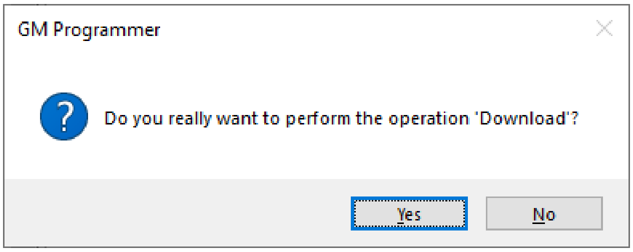
○ Login with download.

● Login without any change.

□ Update bootproject

Applications will not be downloaded to the GM1 controller.

2. From the menu bar, select **Online**>**Download**.
A confirmation message will be displayed, asking whether to download the source code to the GM1 controller.



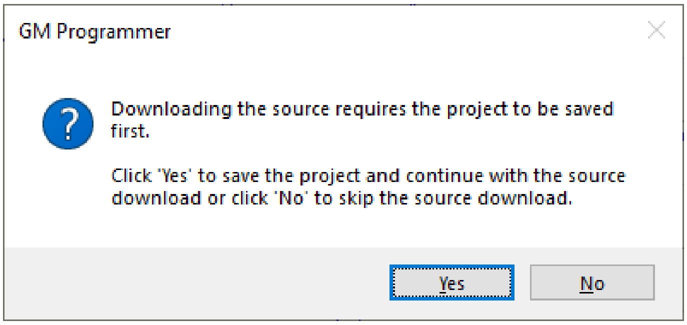
GM Programmer

Do you really want to perform the operation 'Download'?

3. Click [Yes].

   If the project has not yet been saved, a confirmation message will be displayed, asking whether to save the project.

4. If you do not save the project, the source code will not be downloaded. In this case, click the [Yes] button.


GM Programmer

Downloading the source requires the project to be saved first.

Click 'Yes' to save the project and continue with the source download or click 'No' to skip the source download.

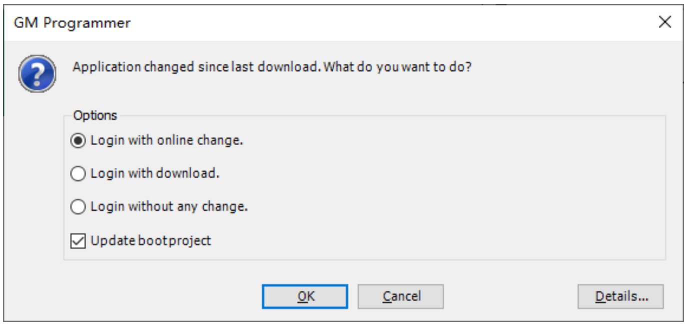
Info.

- To delete the downloaded source code from the GM1 controller, execute "Reset Device". For details on reset, refer to "9.5.1 Reset Warm, Reset Cold, and Reset Origin".

8.8.4 Online Change

Online change allows the user to change applications without having to stop the GM1 controller during operation. Executing online change downloads only different applications to the GM1 controller.

If the applications generated by build during login differ from the applications within the GM1 controller, the following dialog box will be displayed.



GM Programmer

Application changed since last download. What do you want to do?

Options

● Login with online change.

○ Login with download.

○ Login without any change.

☑ Update bootproject

Login with online change

Executes login by downloading only different applications without stopping the GM1 controller

Login with download

Executes login by downloading applications generated by build with the GM1 controller stopped

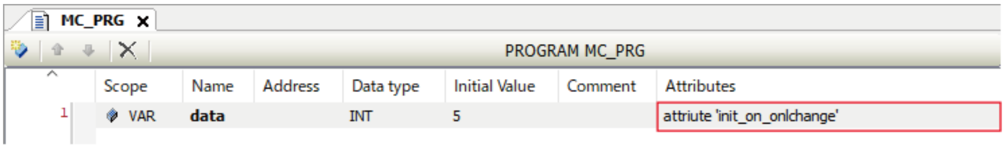
Login without any change

Executes login without downloading the applications generated by build

Info.

- Do not clear the "Update bootproject" check box. If you clear the check box, the applications will not be saved when the GM1 controller is turned OFF.
- When changing the initial value of a variable with online change, be sure to add "attribute 'init_on_onlchange'" declaration to the attribute of the target variable.

  If the declaration is not added, the initial value changed with online change will not be applied.



MC_PRG

PROGRAM MC_PRG

	Scope	Name	Address	Data type	Initial Value	Comment	Attributes
1	VAR	data		INT	5		attriute 'init_on_onlchange'

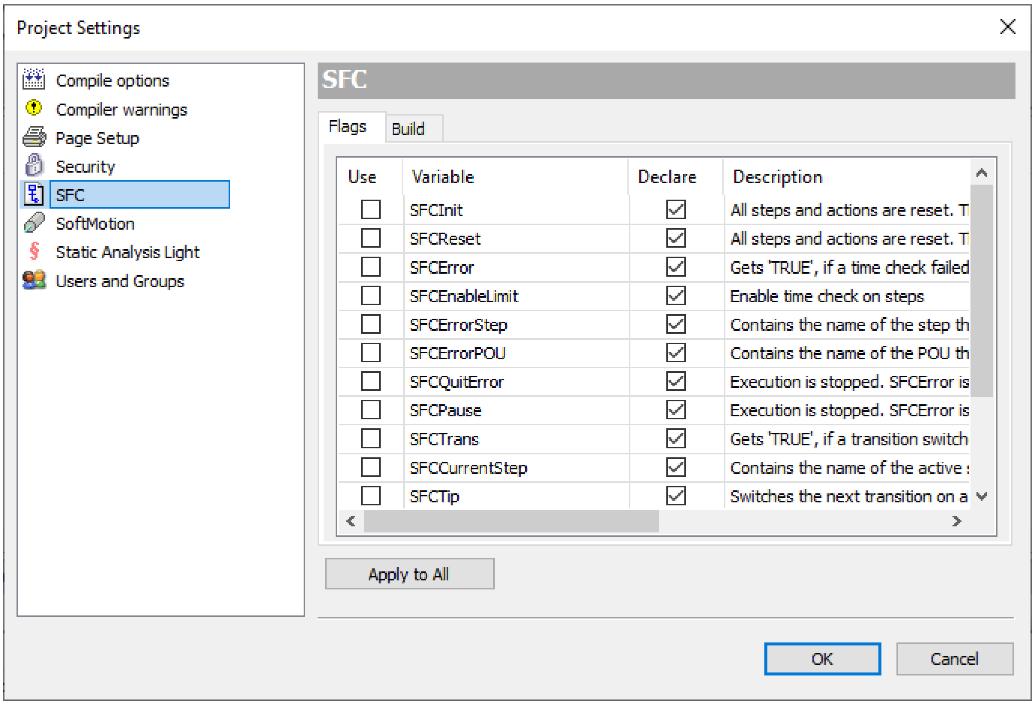
8.8.5 Code Analysis (Static Analysis Light)

Code analysis (Static Analysis Light) can be conducted on programs that are created.
Code analysis can check for the following errors.

Error number	Description
SA0033	Unused variables
SA0028	Overlapping memory areas
SA0006	Write access from several tasks
SA0004	Multiple write accesses on output
SA0027	Multiple uses of identifiers
SA0167	Report of temporary function block instance

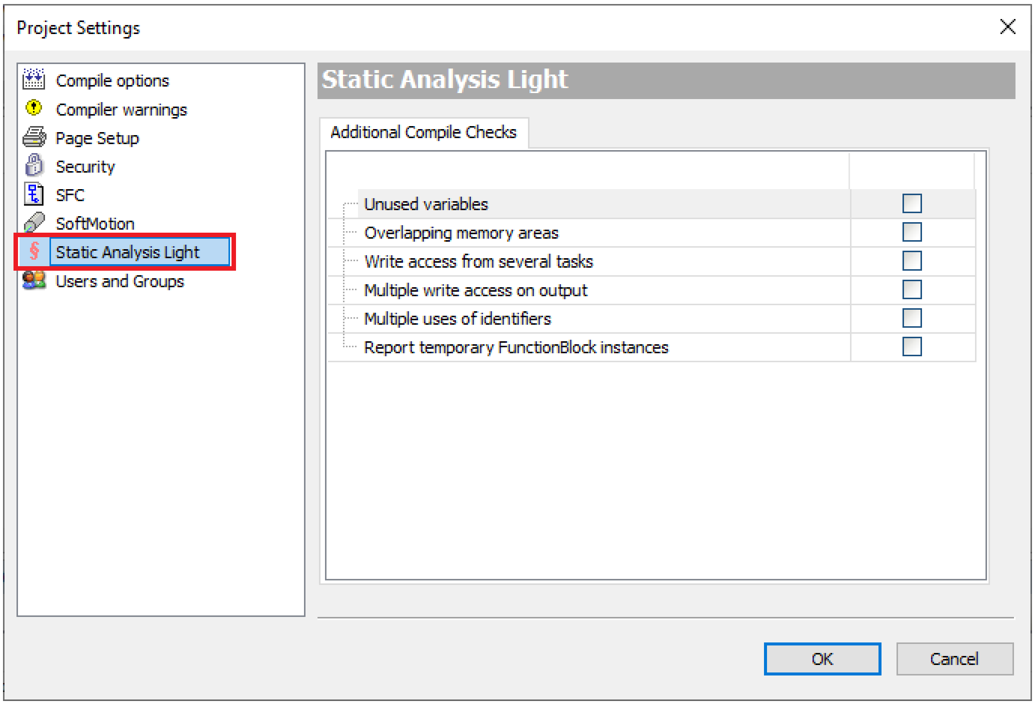
1 2 Procedure

1. From the menu bar, selectProject>Project Settings.
   The Project Settings dialog box will be displayed.



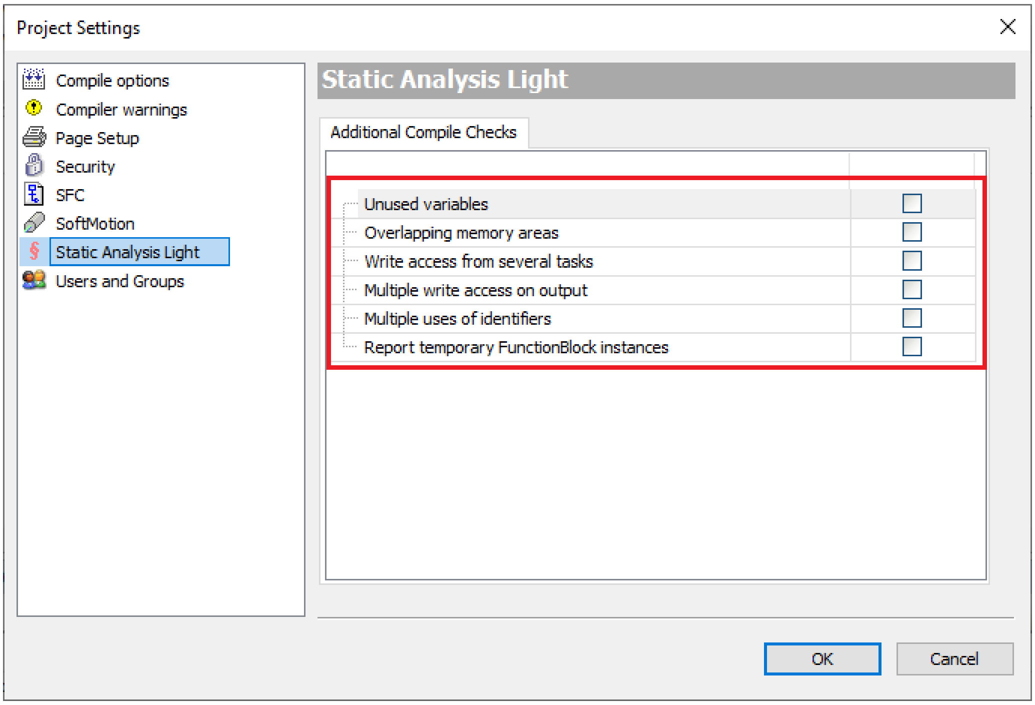
Use	Variable	Declare	Description
□	SFCInit	☑	All steps and actions are reset. T
□	SFCReset	☑	All steps and actions are reset. T
□	SFCError	☑	Gets 'TRUE', if a time check failed
□	SFCEnableLimit	☑	Enable time check on steps
□	SFCErrorStep	☑	Contains the name of the step th
□	SFCErrorPOU	☑	Contains the name of the POU th
□	SFCQuitError	☑	Execution is stopped. SFCError is
□	SFCPause	☑	Execution is stopped. SFCError is
□	SFCTrans	☑	Gets 'TRUE', if a transition switch
□	SFCCurrentStep	☑	Contains the name of the active :
□	SFCTip	☑	Switches the next transition on a

2. In the"Project Settings"dialog box, select the"Static Analysis Light"category.



- Unused variables □
- Overlapping memory areas □
- Write access from several tasks □
- Multiple write access on output □
- Multiple uses of identifiers □
- Report temporary FunctionBlock instances □

3. Select the check boxes of the items to be checked.



- Unused variables □
- Overlapping memory areas □
- Write access from several tasks □
- Multiple write access on output □
- Multiple uses of identifiers □
- Report temporary FunctionBlock instances □

4. Click the[OK]button.

Info.

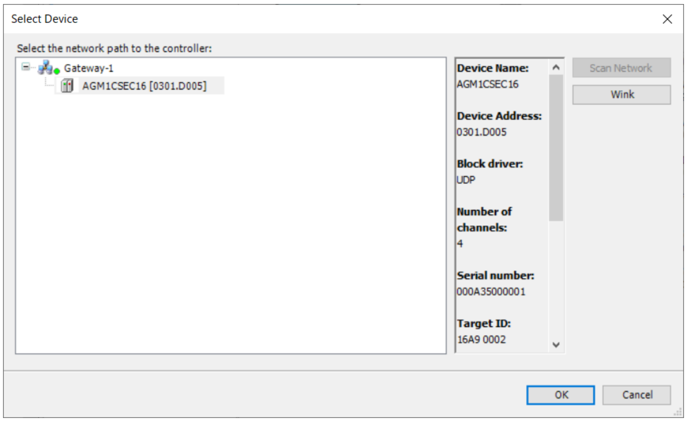
- If the items to be checked are set beforehand, code analysis will be performed automatically during login.

8.9 Source Upload

Upload the source code from the GM1 controller to the PC and retrieve it, as below.

Procedure

1. From the menu bar, select **File**>**Source Upload(Device→PC)**.
   The "Select Device" dialog box will be displayed.



Select Device

Select the network path to the controller:

- Gateway-1
  - AGM1CSEC16 [0301.D005]

Device Name:
AGM1CSEC16

Device Address:
0301.D005

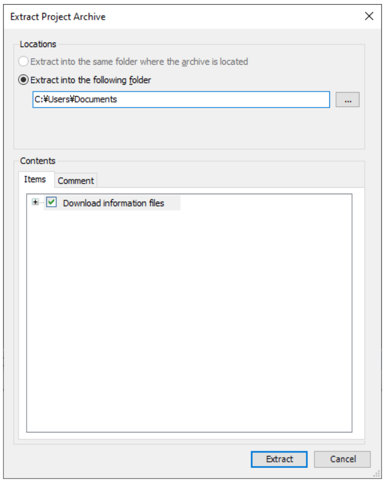
Block driver:
UDP

Number of channels:
4

Serial number:
000A35000001

Target ID:
16A9 0002

2. Select the GM1 controller from which the source code is to be retrieved and click the [OK] button.
   The "Extract Project Archive" dialog box will be displayed.



Extract Project Archive

Locations

○ Extract into the same folder where the archive is located

● Extract into the following folder

C:¥Users¥Documents

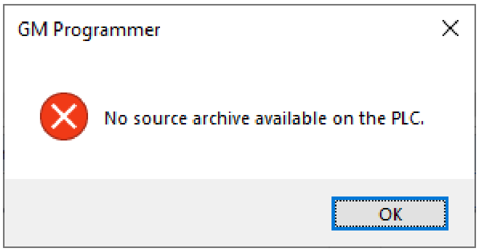
Contents

Items | Comment

☑ Download information files

Info.

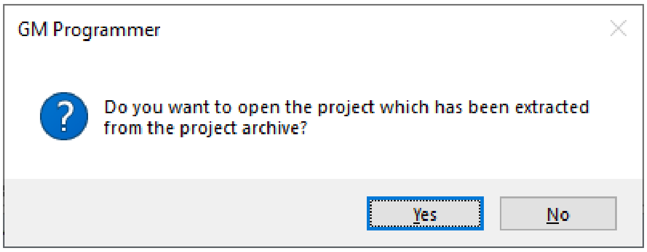
- If the source code does not exist in the selected GM1 controller, the following error message will be displayed.



GM Programmer

No source archive available on the PLC.

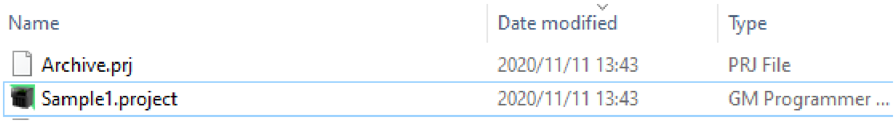
3. Specify the upload destination folder and click the [Extract] button.
A confirmation dialog box will be displayed, asking whether to open the uploaded source code as a project file. Click the [Yes] button to open the uploaded source code as a project file.



GM Programmer

Do you want to open the project which has been extracted from the project archive?

4. In the folder specified as the upload destination, "Archive.prj" and "<project name>.project" will be created.



Name	Date modified	Type
Archive.prj	2020/11/11 13:43	PRJ File
Sample1.project	2020/11/11 13:43	GM Programmer ...

8.10 Commissioning

Commissioning can be conducted using GM Programmer.
To conduct commissioning, the GM1 controller must be connected in online config mode.

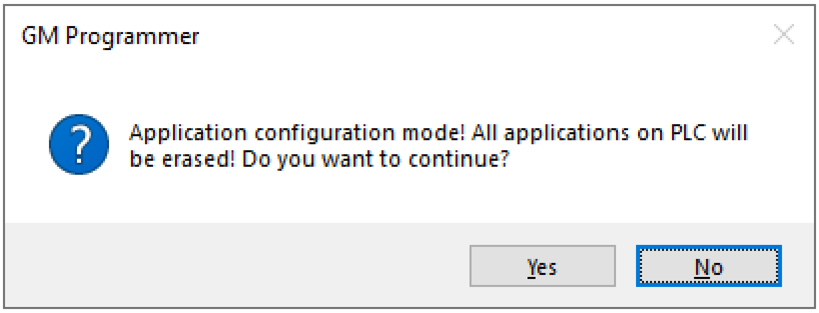
8.10.1 Online Config Mode

When the online config mode is selected, the servo amplifiers are set to be connected to the GM1 Controller.

When using the online config mode, perform the setting as described in "8.4 Communication Setting" in advance.

1 2 Procedure

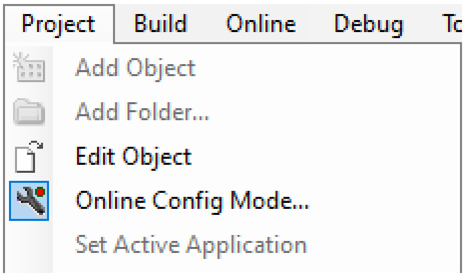
1. From the menu bar, select **Project**>**Online Config Mode**.
   A confirmation message will be displayed, asking whether to remove all applications.



GM Programmer

Application configuration mode! All applications on PLC will be erased! Do you want to continue?

2. Click [Yes].
   All applications will be removed from the GM1 controller, and the GM1 controller and servo amplifiers will be connected in online config mode.
   While online config mode is in progress, "Online Config Mode" in the menu bar remains selected.



Info.

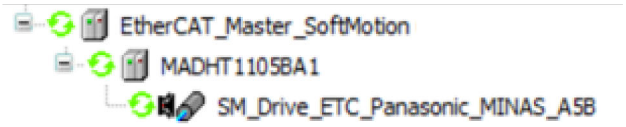
- To cancel the online config mode, select **Project**>**Online Config Mode** from the menu bar again.

8.10.2 Conducting Commissioning for Servo Amplifiers

While in online config mode, you can conduct commissioning for servo amplifiers.
There is no need to create a program for commissioning.
The following is an example of commissioning using the A5B-series servo amplifiers.

Procedure

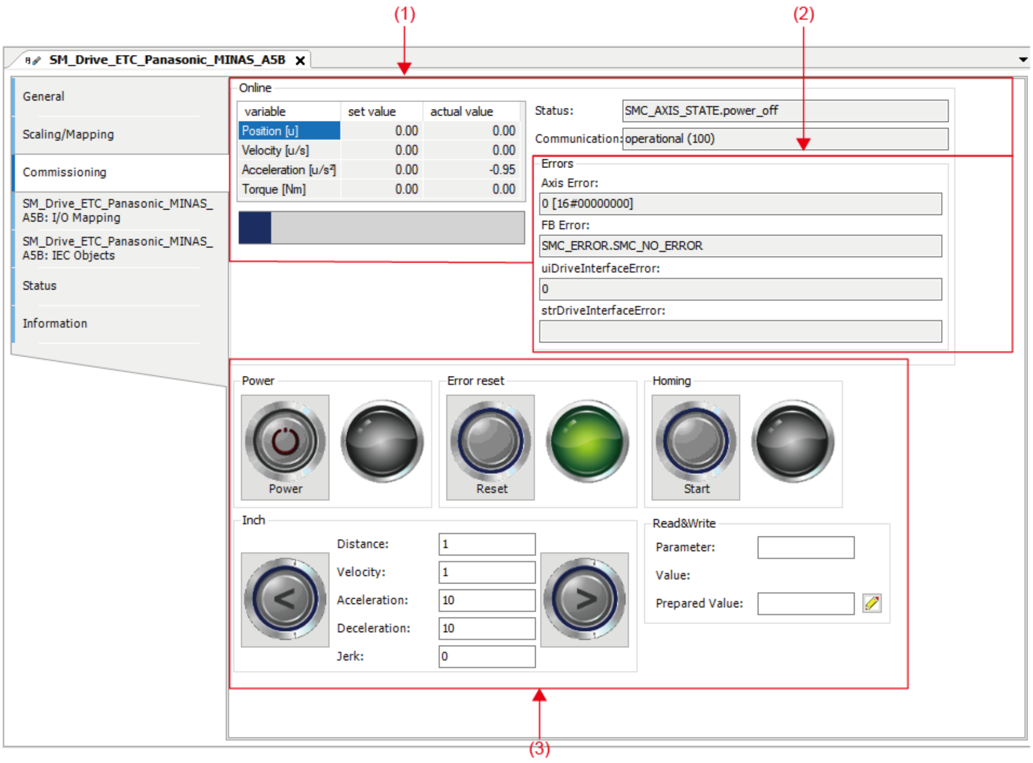
1. Double-click the servo amplifier object in the navigator pane.



EtherCAT_Master_SoftMotion
MADHT1105BA1
SM_Drive_ETC_Panasonic_MINAS_A5B

The "EtherCAT Axis Setting" dialog box will be displayed.

2. Click the "Commissioning" tab.
The Commissioning pane will be displayed.


(1)
(2)
(3)
SM_Drive_ETC_Panasonic_MINAS_A5B
General
Scaling/Mapping
Commissioning
SM_Drive_ETC_Panasonic_MINAS_A5B: I/O Mapping
SM_Drive_ETC_Panasonic_MINAS_A5B: IEC Objects
Status
Information
Power
Power
Error reset
Reset
Homing
Start
Inch
Read&Write

Online

variable	set value	actual value
Position [u]	0.00	0.00
Velocity [u/s]	0.00	0.00
Acceleration [u/s²]	0.00	-0.95
Torque [Nm]	0.00	0.00

Status: SMC_AXIS_STATE.power_off
Communication: operational (100)

Errors
Axis Error: 0 [16#00000000]
FB Error: SMC_ERROR.SMC_NO_ERROR
uiDriveInterfaceError: 0
strDriveInterfaceError:

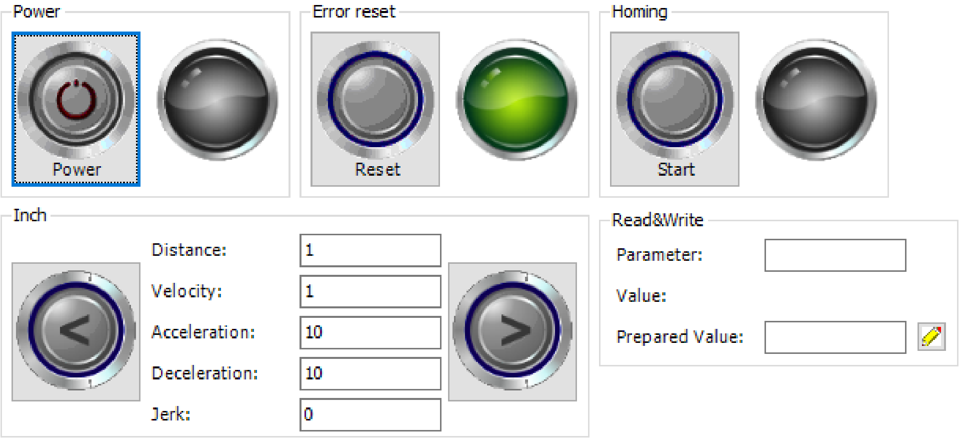
Distance: 1
Velocity: 1
Acceleration: 10
Deceleration: 10
Jerk: 0

Parameter:
Value:
Prepared Value:

No.	Group	Description
(1)	Status	Displays the running status of the servo amplifiers during commissioning.
(2)	Error	Displays errors that occurred during commissioning. Allows the user to clear errors.

No.	Group	Description
(3)	Operation	Allows the user to set commissioning parameters. Allows the user to execute commissioning.

3. Click an appropriate button in the Operation group to start commissioning.
Clicking an icon starts the corresponding commissioning.
To change home return parameters, use the "Program" tab.



Distance: 1
Velocity: 1
Acceleration: 10
Deceleration: 10
Jerk: 0

4. For the servo amplifier status during commissioning, check the "Status" and "Error" groups.

4-1 To clear errors that are displayed, click the [Reset] button in the "Operation" group.
Clicking the [Reset] button clears all errors.

5. From the menu bar, select **Project**>**Online Config Mode**.
Online config mode will be canceled and commissioning will be terminated.

This completes commissioning for servo amplifiers.

Info.

- Even if communication with the servo amplifier is disrupted during "Inching" or "Home Return" operation, the servo amplifier will continue commissioning operation.
- If online config mode is canceled, commissioning will be terminated. To cancel the online config mode, select **Project**>**Online Config Mode** from the menu bar again.

(MEMO)

9 Debug

9.1 Running and Stopping the GM1 Controller

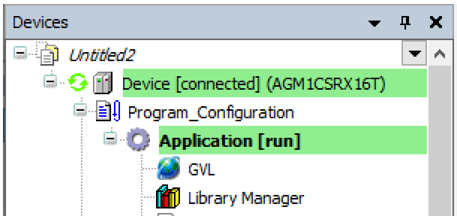
This section explains how to run and stop the GM1 controller and how to run a single cycle operation that is executed in units of a cycle.

9.1.1 Running and Stopping the GM1 Controller

1 2 Procedure

1. After logging in, from the menu bar, select **Debug**>**Start** or press the <F5> key.

   The applications downloaded to the GM1 controller will start running.

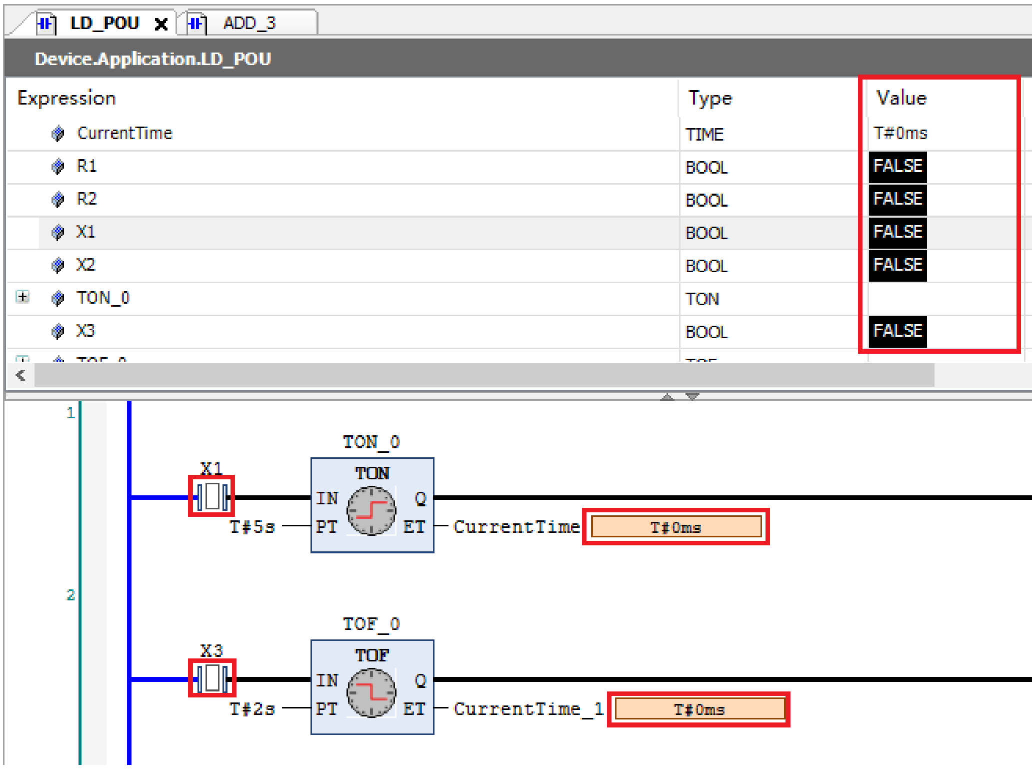
   Devices

   - Untitled2
     - Device [connected] (AGM1CSRX16T)
       - Program_Configuration
         - Application [run]
           - GVL
           - Library Manager

2. From the menu bar, select **Debug**>**Stop**, or press the <Shift> key + <F8> key simultaneously.

   The applications will be stopped.

   During debug operation, you can check the current value of each variable in the declaration section and implementation section.



Expression	Type	Value
CurrentTime	TIME	T#0ms
R1	BOOL	FALSE
R2	BOOL	FALSE
X1	BOOL	FALSE
X2	BOOL	FALSE
TON_0	TON	
X3	BOOL	FALSE

Info.

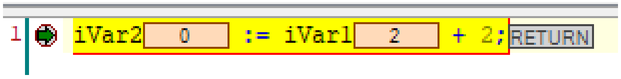
- You can start the GM1 controller by clicking ▶ on the toolbar and also stop it by clicking ■.
- You can select binary, decimal, or hexadecimal as the display format of the variable values to be displayed. From the menu bar, select **Debug**>**Display Mode** and select a display format from those shown.
- If you select confirmed online mode, a confirmation message will be displayed before you start or stop the GM1 controller. For confirmed online mode, refer to "8.7.4 Confirmed Online Mode".

9.1.2 Single Cycle

You can execute the application in simulation mode in a single cycle to check whether a created program is executed as intended.

Procedure

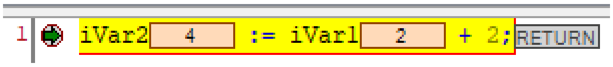
1. After logging in, open the POU.



```
iVar2 0 := iVar1 2 + 2; RETURN
```

2. From the menu bar, select **Debug**>**Single Cycle** or press <Ctrl+F5>.

The opened POU will enter a state in which it has been executed in a single cycle.



```
iVar2 4 := iVar1 2 + 2;RETURN
```

9.2 Breakpoint

By setting a breakpoint in a particular position in a program, you can forcibly stop executing the program and check the variable values.

All programming languages support breakpoints.

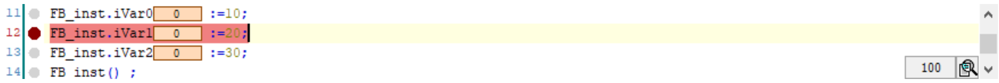
9.2.1 Setting a Breakpoint

1 2 Procedure

1. Select a position where you want to set a breakpoint. From the menu bar, select **Debug**>**Toggle Breakpoint** or press the <F9> key.

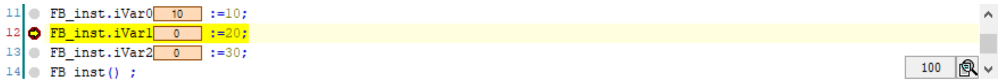
The breakpoint will be enabled.

Example: Setting a breakpoint in line 12 in an ST program



```
11 ○ FB_inst.iVar0[ 0 ] :=10;
12 ● FB_inst.iVar1[ 0 ] :=20;
13 ○ FB_inst.iVar2[ 0 ] :=30;
14 ○ FB_inst() ;
```

If operation is started, the operation will be stopped when the position of the set breakpoint is reached.



```
11 ○ FB_inst.iVar0[ 10 ] :=10;
12 ● FB_inst.iVar1[ 0 ] :=20;
13 ○ FB_inst.iVar2[ 0 ] :=30;
14 ○ FB_inst() ;
```

In the stopped state, the following debug operations can be executed.

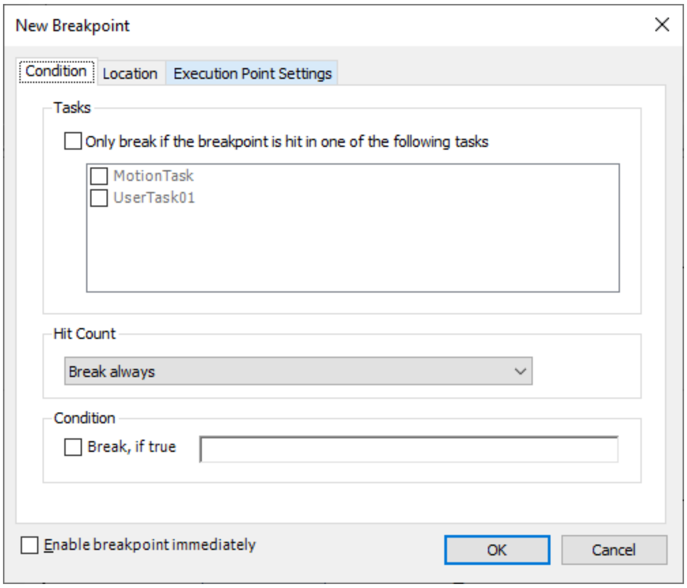
From the menu bar, select "Debug" and then one of the following menu items.

Menu item	Shortcut key	Icon on the toolbar
Step Over	<F10>	↷
Step Into	<F8>	↘
Step Out	<Shift> + <F10>	↗
Run to Cursor	None	→
Set next Statement	None	⇨
Show next Statement	None	⇨

To cancel set breakpoints, from the menu bar, select **Debug**>**Toggle Breakpoint** or click the <F9> key again.

Info.

- You can specify conditions under which operation is stopped when a breakpoint is reached. From the menu bar, select **Debug**>**New Breakpoint**. The "New Breakpoint" dialog box will be displayed. Select the "Condition" tab and specify conditions under which operation is stopped when a breakpoint is reached.



New Breakpoint

Condition | Location | Execution Point Settings

Tasks

□ Only break if the breakpoint is hit in one of the following tasks

□ MotionTask
□ UserTask01

Hit Count

Break always

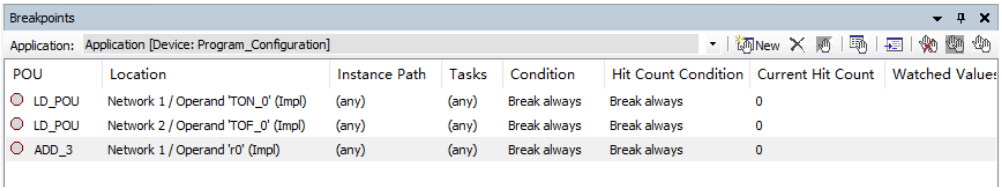
Condition

□ Break, if true

□ Enable breakpoint immediately

- The Breakpoint view allows the user to check a list of set breakpoints. You can check breakpoint positions, break conditions, and the hit count was reached. You can also add, delete, enable, and disable breakpoints.

  To display the Breakpoint view, from the menu bar, select **View**>**Breakpoint**.



Breakpoints

Application: Application [Device: Program_Configuration]

POU	Location	Instance Path	Tasks	Condition	Hit Count Condition	Current Hit Count	Watched Value:
○ LD_POU	Network 1 / Operand 'TON_0' (Impl)	(any)	(any)	Break always	Break always	0	
○ LD_POU	Network 2 / Operand 'TOF_0' (Impl)	(any)	(any)	Break always	Break always	0	
○ ADD_3	Network 1 / Operand 'r0' (Impl)	(any)	(any)	Break always	Break always	0	

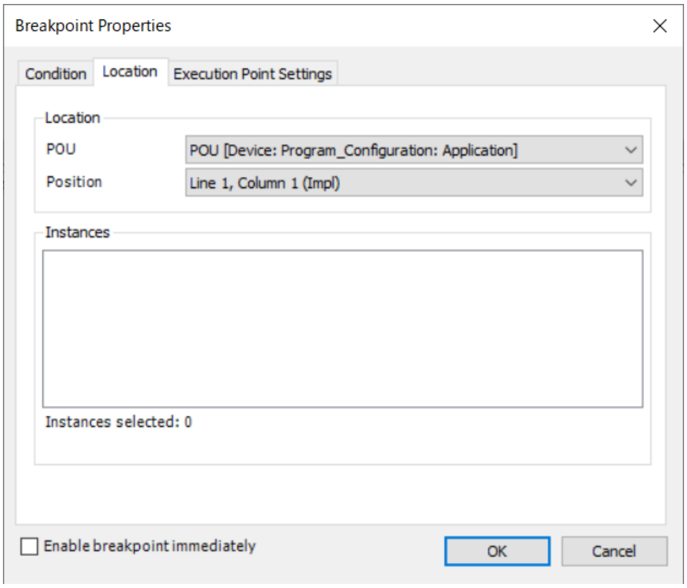
9.2.2 Setting an Execution Point

If an execution point is set, when the position of the execution point is reached, processing that is specified beforehand can be executed and the execution result can be output to the log of the GM1 controller. The application does not stop at the position where an execution point is set.

1 2 Procedure

1. Select a position where you want to set an execution point. From the menu bar, select **Debug**>**New Breakpoint**.

   The "New Breakpoint" dialog box will be displayed.



Breakpoint Properties

Condition | Location | Execution Point Settings

Location

POU: POU [Device: Program_Configuration: Application]

Position: Line 1, Column 1 (Impl)

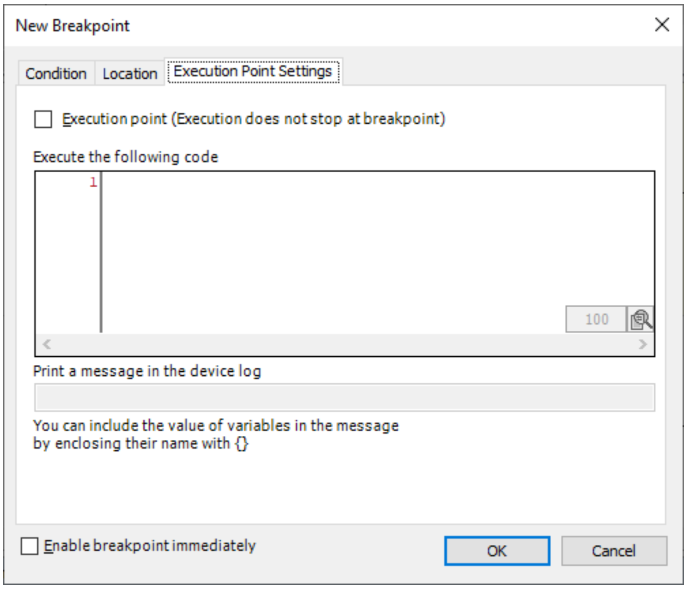
Instances

Instances selected: 0

□ Enable breakpoint immediately

2. Click the "Execution Point Settings" tab.

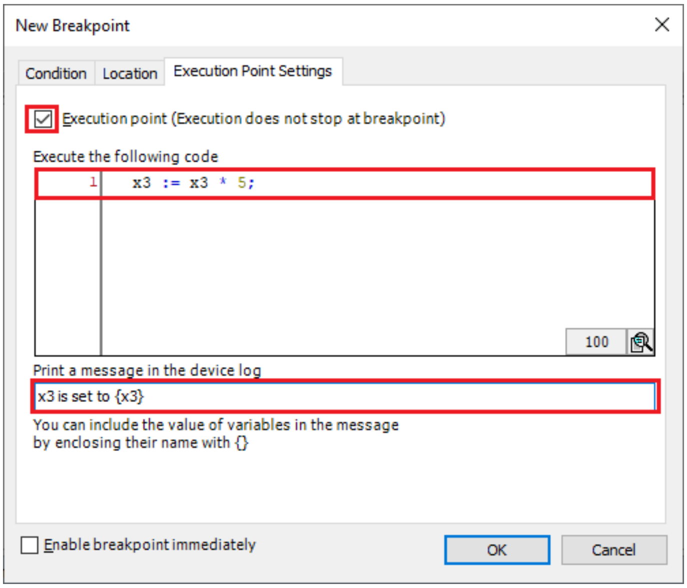
   The Execution Point Settings window will be displayed.


New Breakpoint
Condition
Location
Execution Point Settings
□ Execution point (Execution does not stop at breakpoint)
Execute the following code
100
Print a message in the device log
You can include the value of variables in the message
by enclosing their name with {}
□ Enable breakpoint immediately
OK
Cancel

3. Select the "Execution point" check box and enter the code to be executed at the execution point and the message to be output to the log.

In the "Execute the following code" area, enter executable code in structure text format. In the "Print a message in the device log" field, enter the message to be output to the log.

Example: Multiplying the value of "x3" by 5 and outputting the value to the log


New Breakpoint
Condition
Location
Execution Point Settings
Execute the following code
100
You can include the value of variables in the message
by enclosing their name with {}
□ Enable breakpoint immediately
OK
Cancel

☑ Execution point (Execution does not stop at breakpoint)

Execute the following code

```
x3 := x3 * 5;
```

Print a message in the device log: x3 is set to {x3}

4. Click the [OK] button.

The execution point will be set. When the execution point is enabled, ● appears at the execution point.

Info.

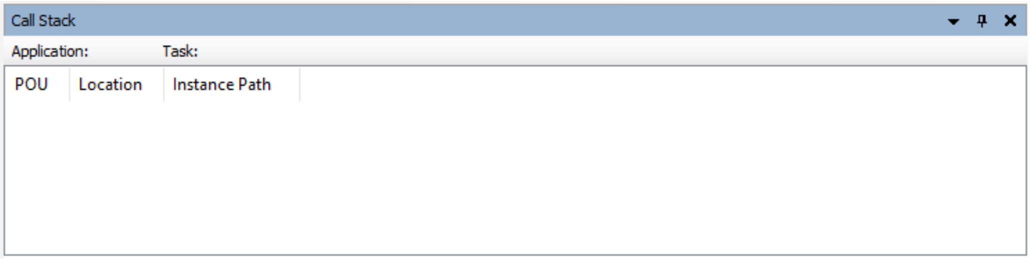
- To output a message to the log when an execution point is reached, from the menu bar, select **Project**>**Project Settings**. In the "Project Settings" dialog box, select the "Compile options" category. Change the setting in **Setting**>**Enable logging in breakpoints** to "Enabled".

9.2.3 Call Stack View

In the Call Stack view, you can check a stop position when operation is stopped due to a breakpoint or for some other reason. If the position is called from another block, the position of the block can also be checked.

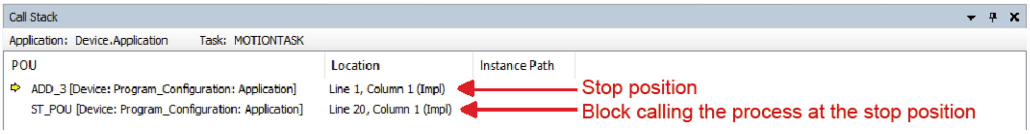
Procedure

1. From the menu bar, select **View**>**Call Stack**.
   The Call Stack view will be displayed.



2. Set a breakpoint and stop the application.
   The stop position and the POU calling the POU at the stop position will be displayed.

Example: When operation stops at line 1 of function "ADD_3" and "ST_POU" calls "ADD_3"



Application: Device.Application Task: MOTIONTASK

POU	Location	Instance Path
➪ ADD_3 [Device: Program_Configuration: Application]	Line 1, Column 1 (Impl)	
ST_POU [Device: Program_Configuration: Application]	Line 20, Column 1 (Impl)	

Stop position

Block calling the process at the stop position

9.3 Debug Operations

This section explains how to perform debug operations such as writing values and watch.

9.3.1 Writing Values and Forcibly Changing Values

Variable values for the GM1 controller can be changed. There are two methods for changing values: Writing values and forcibly changing values.

Writing values: Sets a value (to be changed later) only once. This value can then be changed by the program.

Force Values: Sets a value to be changed in every cycle and maintains the value.

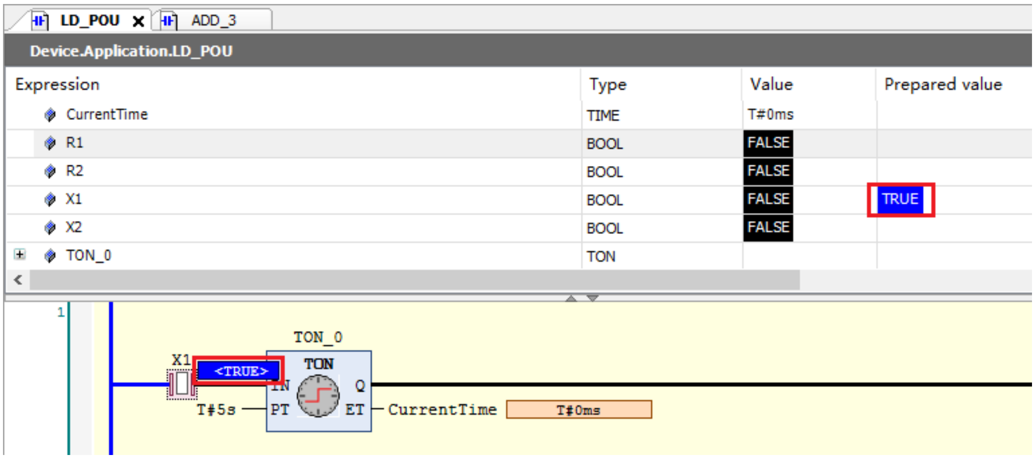
For example, use the following procedure to change the value of Boolean variable "x1" from FALSE to TRUE by writing the value.

1 2 Procedure

1. In the implementation section, double-click the element whose value is to be changed.

A new value will be preset.

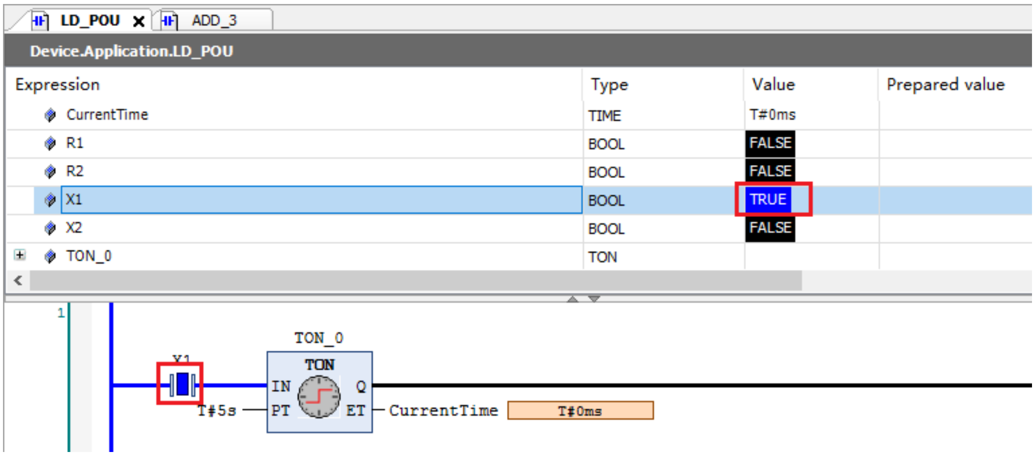
You can also preset a value by clicking a cell in the "Preset value" column of the declaration section.



Expression	Type	Value	Prepared value
CurrentTime	TIME	T#0ms	
R1	BOOL	FALSE	
R2	BOOL	FALSE	
X1	BOOL	FALSE	TRUE
X2	BOOL	FALSE	
TON_0	TON		

2. From the menu bar, select **Debug**>**Write Values**, or press the <Ctrl> key + <F7> key simultaneously.

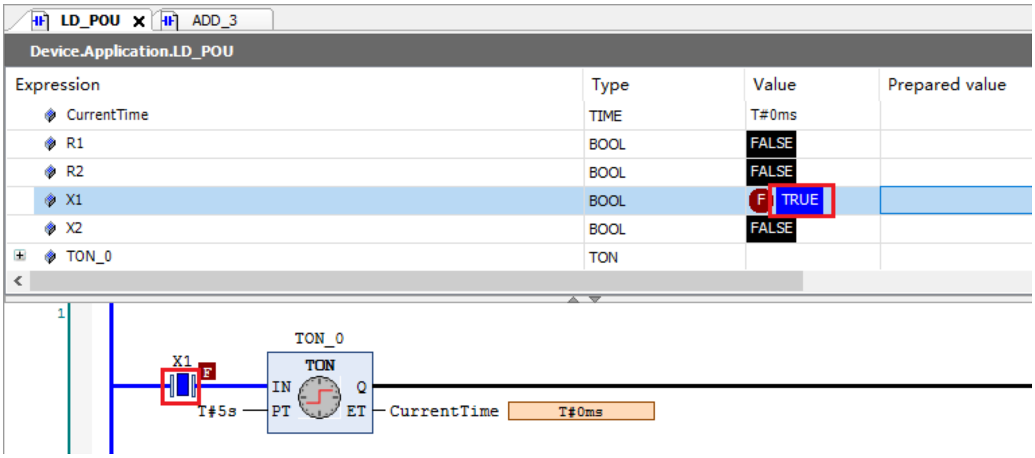
The preset value will be written.



Device.Application.LD_POU

Expression	Type	Value	Prepared value
CurrentTime	TIME	T#0ms	
R1	BOOL	FALSE	
R2	BOOL	FALSE	
X1	BOOL	TRUE	
X2	BOOL	FALSE	
TON_0	TON		

From the menu bar, select **Debug**>**Force Values**, or press the <F7> key. The variabsle value will be forcibly changed. F appears in front of a variable whose value has been forcibly changed, and then the value will not be updated by the program.



Device.Application.LD_POU

Expression	Type	Value	Prepared value
CurrentTime	TIME	T#0ms	
R1	BOOL	FALSE	
R2	BOOL	FALSE	
X1	BOOL	F TRUE	
X2	BOOL	FALSE	
TON_0	TON		

From the menu bar, select **Debug**>**Unforce Values**, or press the <Alt> key + <F7> key simultaneously. Forced value change will be canceled.

9.3.2 Watch

By registering variables in the watch view, you can perform variable value management such as checking or changing variable values.

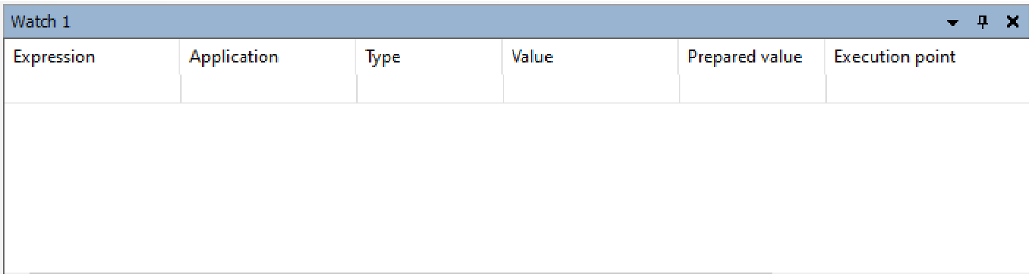
You can use up to four watch views (Watch 1 to Watch 4).

For example, use the following procedure to register variable "x1" in watch view "Watch 1".

1 2 Procedure

1. From the menu bar, select **View**>**Watch**>**Watch 1**.

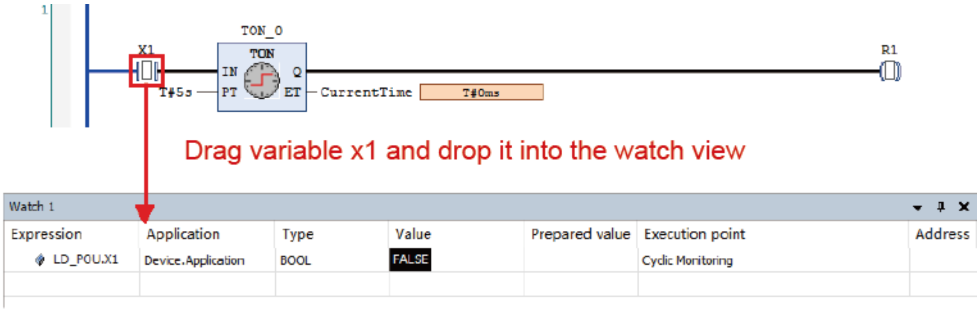
   Watch view "Watch 1" will be displayed.



2. Drag the variable "x1" element in the implementation section and drop it in the watch view.

   Variable "x1" will be registered in the watch view.

   You can also register the variable in the watch view by dragging it from the declaration section and dropping it in the watch view.



Drag variable x1 and drop it into the watch view

This completes the procedure for registering the variable in the watch view. You can check variable values in the Value column.

Info.

- Variables whose values have been forcibly changed are automatically registered in the "Watch all Forces" view.

  From the menu bar, select **View**>**Watch**>**Watch all Forces**.
- If an execution point has been set, the timing of display in the watch view can be set to the point in time when the execution point is reached. In the "Execution point" column, select the execution point that has been set. For details on how to set execution points, refer to "9.2.2 Setting an Execution Point".

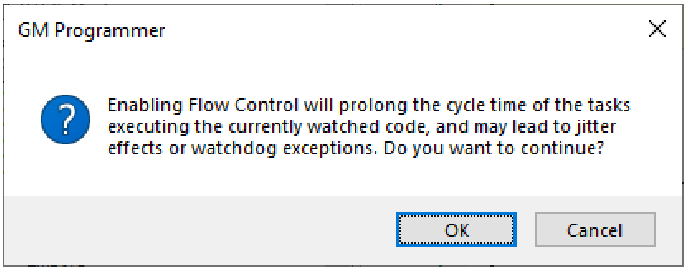
9.3.3 Flow Control

Flow control enables monitoring to be performed by using different colors in positions where the program is executed and in positions where the program is not executed.

Flow control can be used in LD programs, ST programs, and FBD programs.

1 2 Procedure

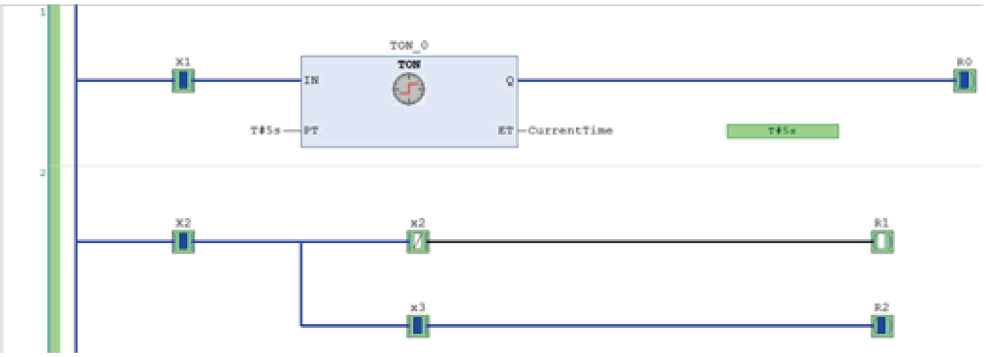
1. After logging in, from the menu bar, select **Debug**>**Toggle Flow Control Mode**.
   The flow control notification dialog box will be displayed.



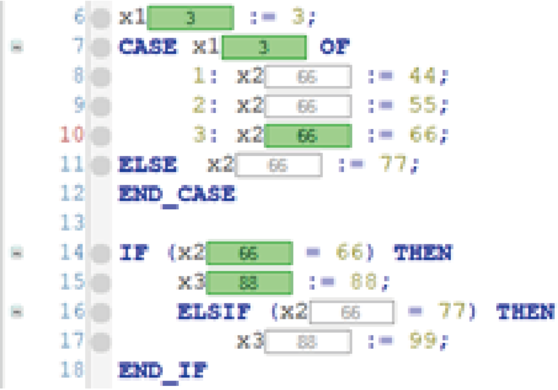
Enabling Flow Control will prolong the cycle time of the tasks executing the currently watched code, and may lead to jitter effects or watchdog exceptions. Do you want to continue?

2. Click the [OK] button.
   The display will be switched to flow control display.
   The positions where the program was executed are displayed in green and the positions where the program was not executed are displayed in white.

Example: Flow control display for LD programs



Example: Flow control display for ST programs



```
x1 3 := 3;
CASE x1 3 OF
        1: x2 66 := 44;
        2: x2 66 := 55;
        3: x2 66 := 66;
ELSE  x2 66 := 77;
END_CASE

IF (x2 66 = 66) THEN
      x3 88 := 88;
      ELSIF (x2 66 = 77) THEN
          x3 88 := 99;
END_IF
```

Info.

- By using confirmed online mode, you can have a confirmation message dialog box displayed before you execute flow control. For confirmed online mode, refer to "8.7.4 Confirmed Online Mode".

9.3.4 Operation Mode

Using the operation mode function makes it possible to prevent some debug operations from being executed. This can prevent incorrect operation of the GM1 controller when it is operated accidentally.

The current operation mode is displayed as an icon on the status bar.

Debug (🔓)

This mode has no restriction.

Locked (🔒)

Start / stop, new breakpoint setting, and forcing values cannot be executed.

Single cycle operation, writing variables, and unforcing values can be executed.

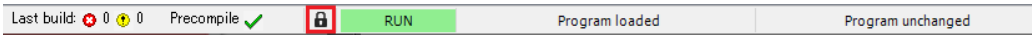
Operational (🔒)

Only writing variables can be executed. Start / stop, new breakpoint setting, forced variable change, single cycle, and canceling forced variable change cannot be executed. To use this mode, the following conditions must be satisfied.

- Application is running
- There is no active breakpoint
- There is no variable whose value has been forcibly changed
- The application created in GM Programmer matches the boot application in the GM1 controller

1 2 Procedure

1. After logging in, from the menu bar, select **Online**>**Operation Mode**>**Locked**.
   The operation mode will be changed from Debug mode to Locked mode.



Last build: 0 0 Precompile ✓ 🔒 RUN Program loaded Program unchanged

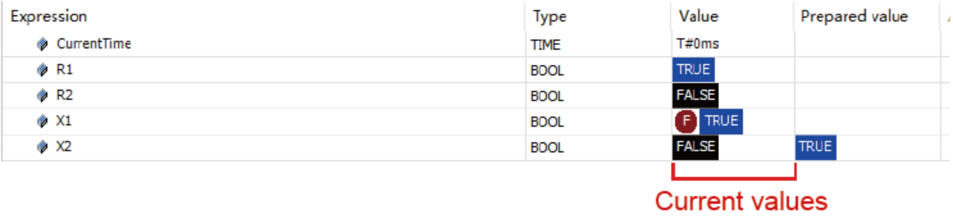
9.4 Monitoring Function

The monitoring function allows the user to check the variables in the program and the current values of the device parameters in real time while being logged in the GM1 controller.

■ Monitoring variables in the declaration editor

The variables declared in the declaration editor can be monitored.

F appears in front of forcibly changed values. For details on forced value change, refer to"9.3.1 Writing Values and Forcibly Changing Values".



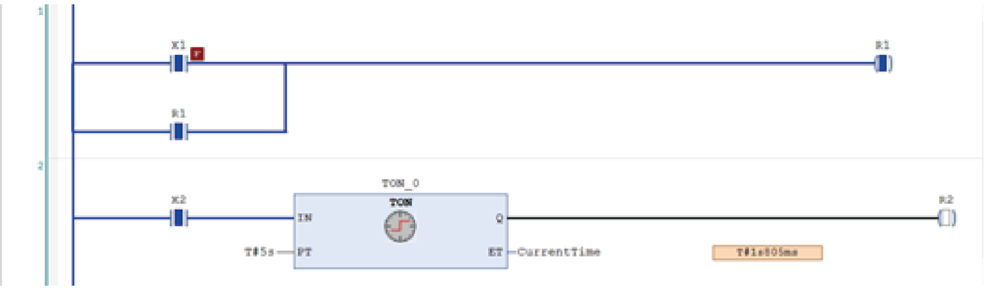
Expression	Type	Value	Prepared value
CurrentTime	TIME	T#0ms	
R1	BOOL	TRUE	
R2	BOOL	FALSE	
X1	BOOL	F TRUE	
X2	BOOL	FALSE	TRUE

Current values

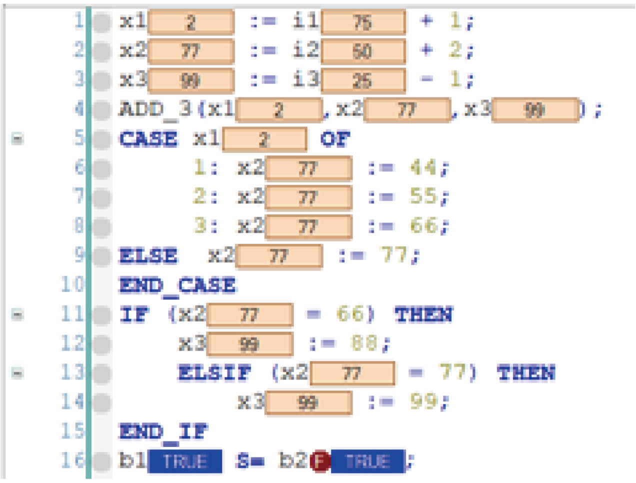
■ Monitoring variables in the implementation section of the program

- Variables can be monitored in the implementation section of the program (inline monitoring).
- The current value is displayed on the right side of each variable, such as T#1s805ms or 2 .
- Contacts, coils, and connection lines are displayed in blue when the current value is TRUE.

<Inline monitoring for LD programs>



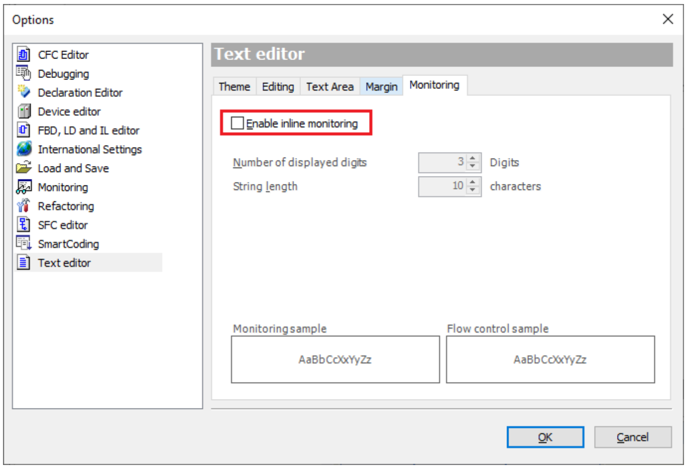
<Inline monitoring for ST programs>



```
x1 := i1 + 1;
x2 := i2 + 2;
x3 := i3 - 1;
ADD_3(x1, x2, x3);
CASE x1 OF
    1: x2 := 44;
    2: x2 := 55;
    3: x2 := 66;
ELSE x2 := 77;
END_CASE
IF (x2 = 66) THEN
    x3 := 88;
    ELSIF (x2 = 77) THEN
        x3 := 99;
END_IF
b1 S= b2;
```

Info.

- Inline monitoring can be disabled. Open the Options window (by selecting**Tools**>**Options**), select the"Text editor"category and then"Monitoring"tab, and clear the"Enable inline monitoring"check box.



Options

Text editor

□ Enable inline monitoring

Number of displayed digits 3 Digits

String length 10 characters

Monitoring sample AaBbCcXxYyZz

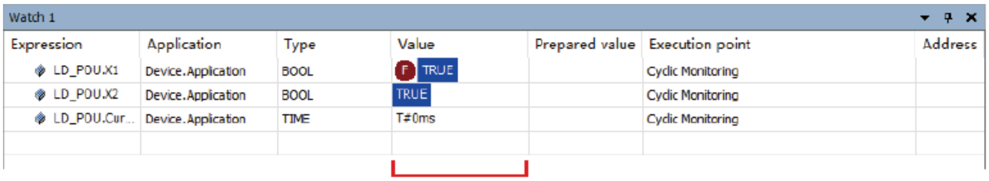
Flow control sample AaBbCcXxYyZz

■ Monitoring variables in the watch view

By registering variables in the watch view, you can monitor the variables.

You can use up to four watch views, as well as a dedicated view where variables whose values are forcibly changed are automatically registered.

For details on how to register variables in the watch view, refer to"9.3.2 Watch".



Watch 1

Expression	Application	Type	Value	Prepared value	Execution point	Address
LD_POU.X1	Device.Application	BOOL	F TRUE		Cyclic Monitoring	
LD_POU.X2	Device.Application	BOOL	TRUE		Cyclic Monitoring	
LD_POU.Cur...	Device.Application	TIME	T#0ms		Cyclic Monitoring	

Current values

9.5 Reset of Controller

Reset operation resets the active applications and initializes the variables and settings.

Reset is divided into the following four types and variables and settings that are initialized differ according to the reset type.

Reset Warm

Initializes variables other than the RETAIN and PERSISTENT variables.

Reset Cold

Initializes variables other than the PERSISTENT variable.

Reset Origin

Initializes all variables. Active applications are deleted from the GM1 controller.

Device Reset

Initializes all variables and device user management information. Applications and source code are deleted from the GM1 controller.

- The following table shows the items that are initialized by reset or other tool operation or controller operation.

○ : Retained × : Initialized Update: Updated

* : If you change the structure of the data by adding or deleting variables, the persistent variables will be cleared if you download it as it is, so please implement it together with "15.4.5 Saving and restoring persistent variables"

Operation		Variables other than "RETAIN / PERSISTENT"	RETAIN variable (RETAIN)	PERSISTENT Variables (PERSISTENT)	(Boot) Application	User management	Source file	IP address	RTC time zone
Tool operation	Stop	○	○	○	○	○	○	○	○
	Reset warm	×	○	○	○	○	○	○	○
	Reset Cold	×	×	○	○	○	○	○	○
	Download	×	×	○*	Update	○	Update	Update	○
	Online change	○	○	○*	Update	○	Update	○	○
	Reset Origin	×	×	×	×	○	○	○	○
	Resetting "Device"	×	×	×	×	×	×	○	○
Controller operation	Power cycling	×	○	○	○	○	○	○	○
	Resetting "Device" by means of	×	×	×	×	×	×	○	○

Operation		Variables other than "RETAIN / PERSISTENT"	RETAIN variable (RETAIN)	PERSISTENT Variables (PERSISTENT)	(Boot) Application	User management	Source file	IP address	RTC time zone
	hard switching								

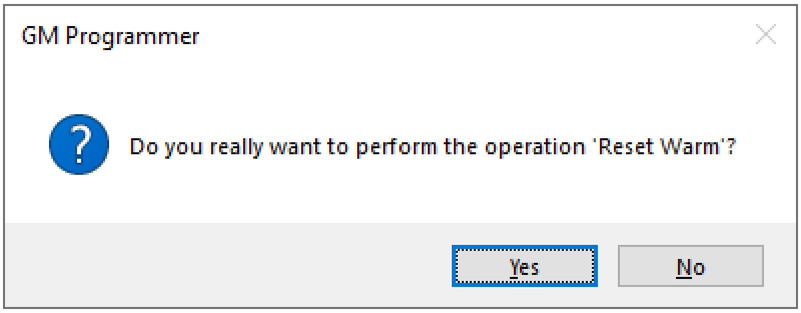
9.5.1 Reset Warm, Reset Cold, and Reset Origin

Execute Warm Reset, Cold Reset, and Reset Origin by selecting them from "Online" on the menu bar. This section explains the execution procedure, using Warm Reset as an example.

1 2 Procedure

1. From the menu bar, select **Online**>**Reset Warm**.

Example: "Reset Warm" execution procedure



GM Programmer

Do you really want to perform the operation 'Reset Warm'?

2. Click the [Yes] button.
Reset warm will be executed.

9.5.2 Executing Device Reset from GM Programmer

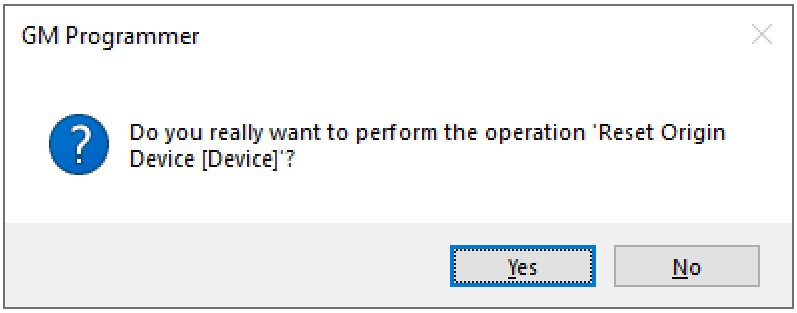
Device reset can be executed from the GM1 controller as well as from GM Programmer.

To execute device reset from GM Programmer, right-click in the navigator pane and execute device reset from the context-sensitive menu that is displayed.

1 2 Procedure

1. Right-click the [Device] object in the navigator pane and then select "Reset Origin" from the context-sensitive menu that is displayed.

   A confirmation message will be displayed, asking whether to execute device reset.



GM Programmer

Do you really want to perform the operation 'Reset Origin Device [Device]'?

2. Click the [Yes] button.

   Device reset will be executed. When device reset is executed, you are logged out.

Info.

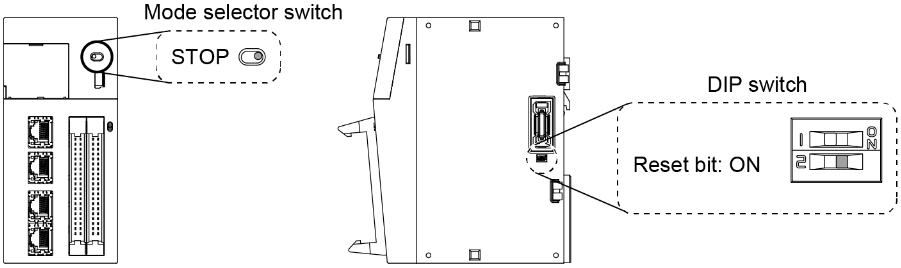
- If you right-click the [Application] object in the navigator pane and select "Delete application from device", the selected application will be removed.

9.5.3 Executing Device Reset from GM1 Controller

Device reset can be executed from the GM1 controller.

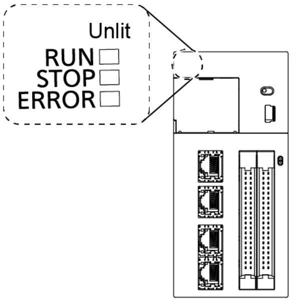
1 2 Procedure

1. Check that the power is OFF, set the mode selector switch to STOP, and set the reset bit of the DIP switch to ON.

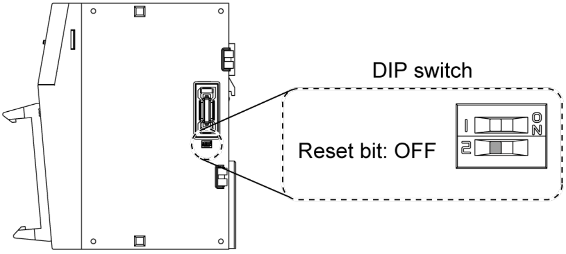


2. Turn the power ON.
   Device reset will be executed.

3. When the "RUN", "STOP", and "ERROR" LEDs go out, device reset is completed.



After device reset is completed, turn the power OFF and set the reset bit of the DIP switch to OFF.



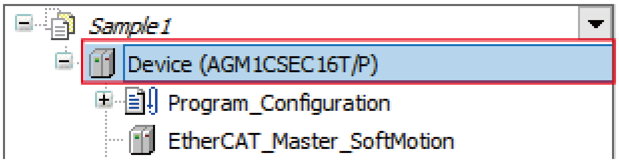
9.6 Checking the Status of GM1 Controller

9.6.1 Checking Logs

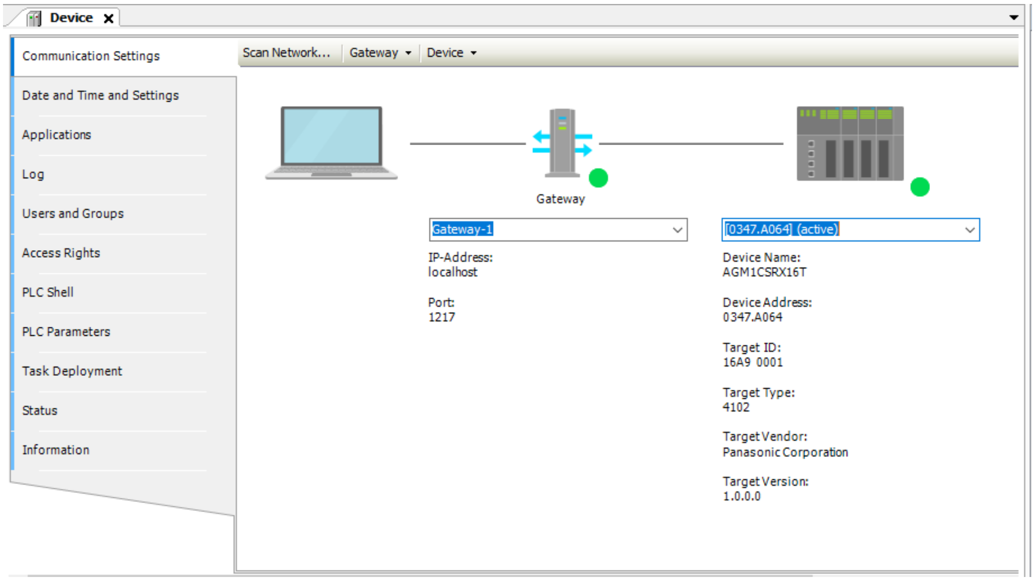
You can check logs of GM1 controller startup, shutdown, application download, and other events.

1 2 Procedure

1. Connect the PC where GM Programmer is installed and the GM1 controller.
For details, refer to "8.5 Connecting to the GM1 Controller".
Double-click the [Device] object in the navigator pane.



The Device setting window will be displayed.



Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

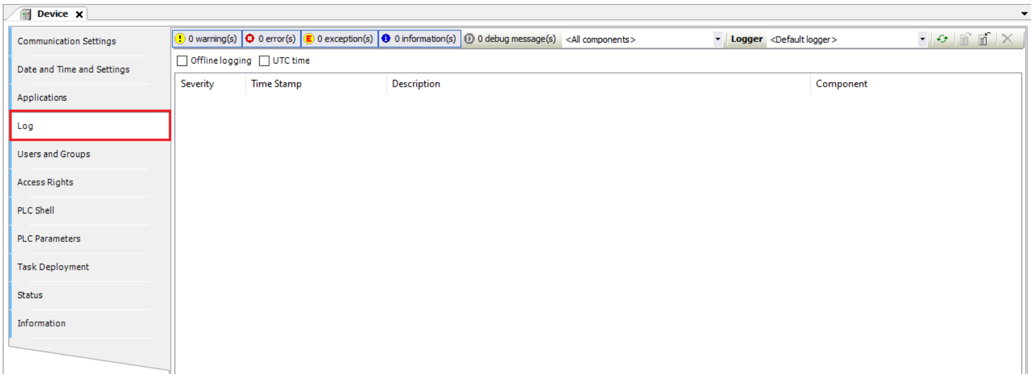
Target ID:
16A9 0001

Target Type:
4102

Target Vendor:
Panasonic Corporation

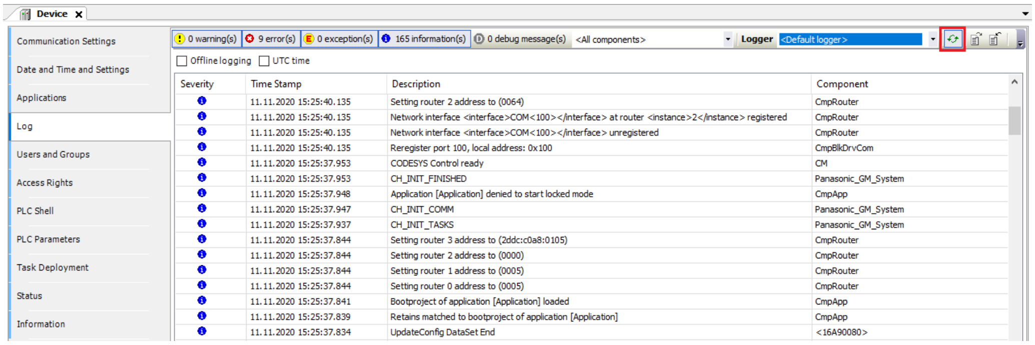
Target Version:
1.0.0.0

2. Click the [Log] tab.
The log window will be displayed.



3. Click the ⟳ icon.

The log will be displayed.



Severity	Time Stamp	Description	Component
ℹ	11.11.2020 15:25:40.135	Setting router 2 address to (0064)	CmpRouter
ℹ	11.11.2020 15:25:40.135	Network interface <interface>COM<100></interface> at router <instance>2</instance> registered	CmpRouter
ℹ	11.11.2020 15:25:40.135	Network interface <interface>COM<100></interface> unregistered	CmpRouter
ℹ	11.11.2020 15:25:40.135	Reregister port 100, local address: 0x100	CmpBlkDrvCom
ℹ	11.11.2020 15:25:37.953	CODESYS Control ready	CM
ℹ	11.11.2020 15:25:37.953	CH_INIT_FINISHED	Panasonic_GM_System
ℹ	11.11.2020 15:25:37.948	Application [Application] denied to start locked mode	CmpApp
ℹ	11.11.2020 15:25:37.947	CH_INIT_COMM	Panasonic_GM_System
ℹ	11.11.2020 15:25:37.937	CH_INIT_TASKS	Panasonic_GM_System
ℹ	11.11.2020 15:25:37.844	Setting router 3 address to (2ddc:c0a8:0105)	CmpRouter
ℹ	11.11.2020 15:25:37.844	Setting router 2 address to (0000)	CmpRouter
ℹ	11.11.2020 15:25:37.844	Setting router 1 address to (0005)	CmpRouter
ℹ	11.11.2020 15:25:37.844	Setting router 0 address to (0005)	CmpRouter
ℹ	11.11.2020 15:25:37.841	Bootproject of application [Application] loaded	CmpApp
ℹ	11.11.2020 15:25:37.839	Retains matched to bootproject of application [Application]	CmpApp
ℹ	11.11.2020 15:25:37.834	UpdateConfig DataSet End	<16A90080>

ℹ Info.

- The displayed log can be exported (by clicking) or imported (by clicking) as an XML file.

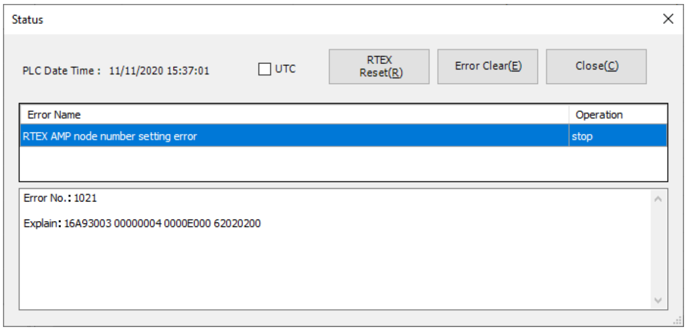
9.6.2 Checking the Status

You can check only one error item that has the highest severity level among all errors currently occurring in the GM1 controller.

Procedure

1. Connect the PC where GM Programmer is installed and the GM1 controller.
   For details, refer to "8.5 Connecting to the GM1 Controller".

2. From the menu bar, select **Online**>**Status**.
   The "Status" dialog box will be displayed. You can check only one error item that has the highest severity level among all errors currently occurring in the GM1 controller.



PLC Date Time : 11/11/2020 15:37:01 □ UTC

Error Name	Operation
RTEX AMP node number setting error	stop

Error No.: 1021

Explain: 16A93003 00000004 0000E000 62020200

3. Click the "Close" button.
   The "Status" dialog box will be closed.

Info.

Error types

The following table shows the types and recovery methods of errors that are displayed.

Clicking the [Error Clear] button deletes the target status item.

Error type	Recovery method
System error (power cycle)	Turn the GM1 controller OFF and then ON.
System error (Reinitialize)	Reinitialize the system. (Applications will be downloaded without executing a reset and the mode will be set to RUN.)
System error (Stop operation)	Click the [Error Clear] button.
System error (Continue operation)	Click the [Error Clear] button.
Incorporation / expansion error	Reinitialize the system.
Unit error	Click the [Error Clear] button or reinitialize the system.

Refer to "9.9.2 Checking and Clearing Errors Using GM Programmer".

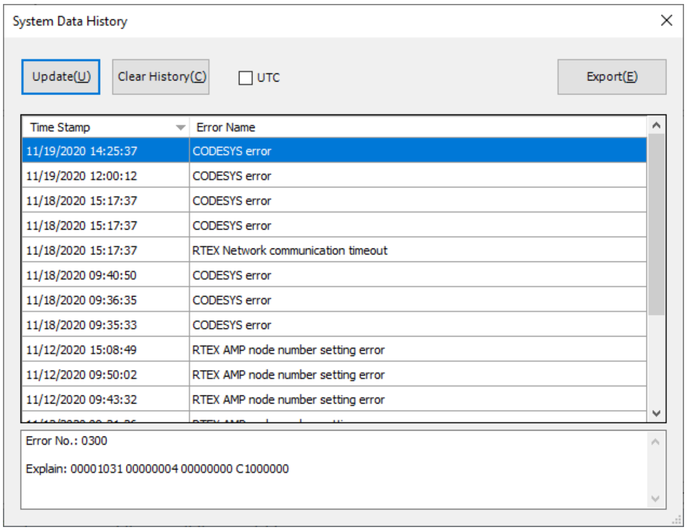
9.6.3 Checking the System Data History

You can check histories of errors that have occurred up until now.

Procedure

1. Connect the PC where GM Programmer is installed and the GM1 controller.
   For details, refer to "8.5 Connecting to the GM1 Controller".

2. From the menu bar, select **Online**>**System Data History**.
   The "System Data History" dialog box will be displayed. You can check errors that have occurred up until now.
   Clicking the [Update] button collects system data histories again.
   Clicking the [Export] button outputs the system data histories displayed in the dialog box to a ".csv" file.
   Clicking the [Clear History] button deletes the system data histories that are displayed.



Time Stamp	Error Name
11/19/2020 14:25:37	CODESYS error
11/19/2020 12:00:12	CODESYS error
11/18/2020 15:17:37	CODESYS error
11/18/2020 15:17:37	CODESYS error
11/18/2020 15:17:37	RTEX Network communication timeout
11/18/2020 09:40:50	CODESYS error
11/18/2020 09:36:35	CODESYS error
11/18/2020 09:35:33	CODESYS error
11/12/2020 15:08:49	RTEX AMP node number setting error
11/12/2020 09:50:02	RTEX AMP node number setting error
11/12/2020 09:43:32	RTEX AMP node number setting error

Error No.: 0300

Explain: 00001031 00000004 00000000 C1000000

3. Click the "×" button.
   The "System Data History" dialog box will be closed.

Info.

For details, refer to "9.9.2 Checking and Clearing Errors Using GM Programmer".

9.6.4 Task Monitoring

You can check the task status, the number of cycles, cycle time, and jitter while being logged in the GM1 controller.
Double-click the "Task Configuration" object in the navigator pane and select the "Monitor" tab.

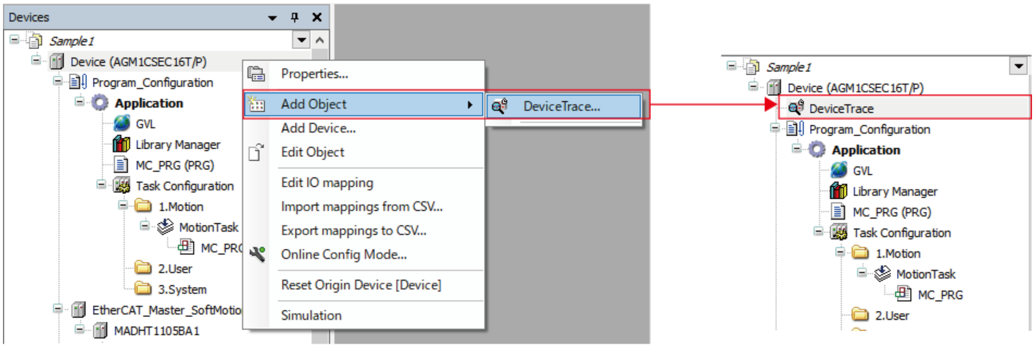
9.7 Device Trace Function

The Device Trace function of GM Programmer can monitor the CPU load factor of the GM1 controller.

This function allows you to check whether the entire task falls within the appropriate CPU time range.

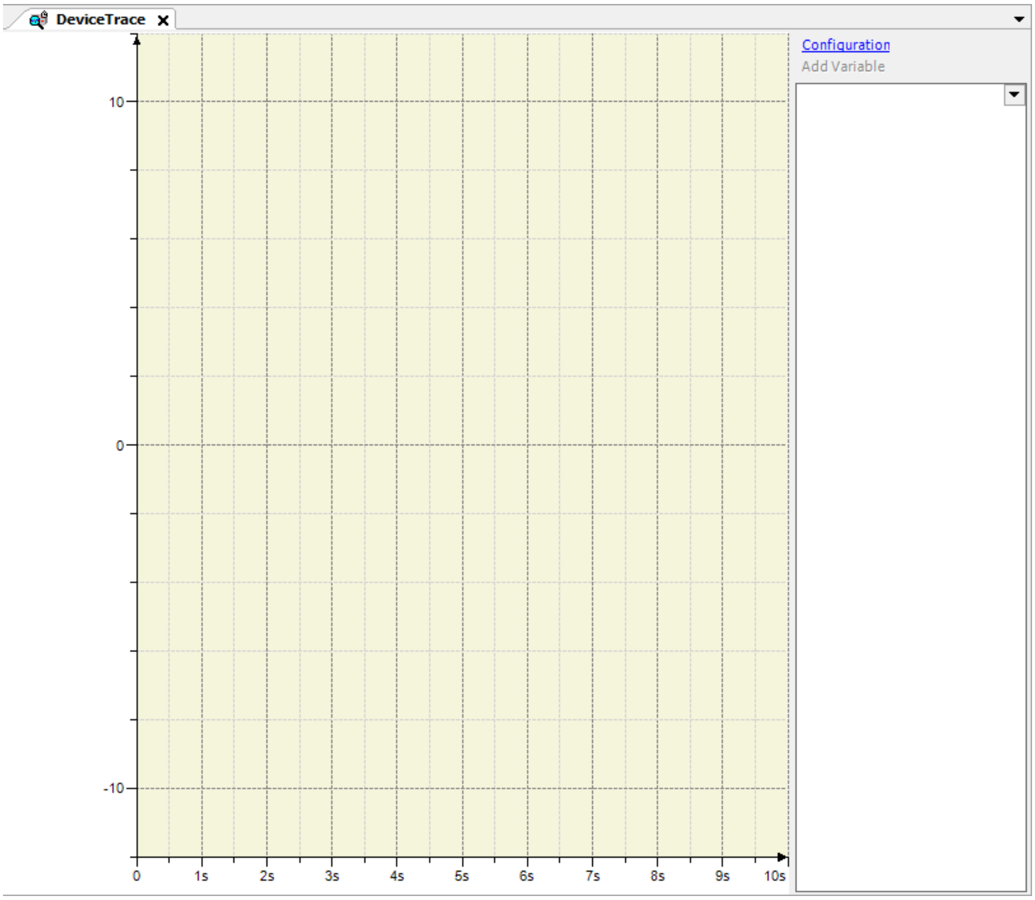
1 2 Procedure

1. Right-click "Device" and select **Add Object**>**DeviceTrace...**.



2. You will be logged in to the device.

3. Double-click the "DeviceTrace" object that has been added. The "DeviceTrace" window will be displayed.

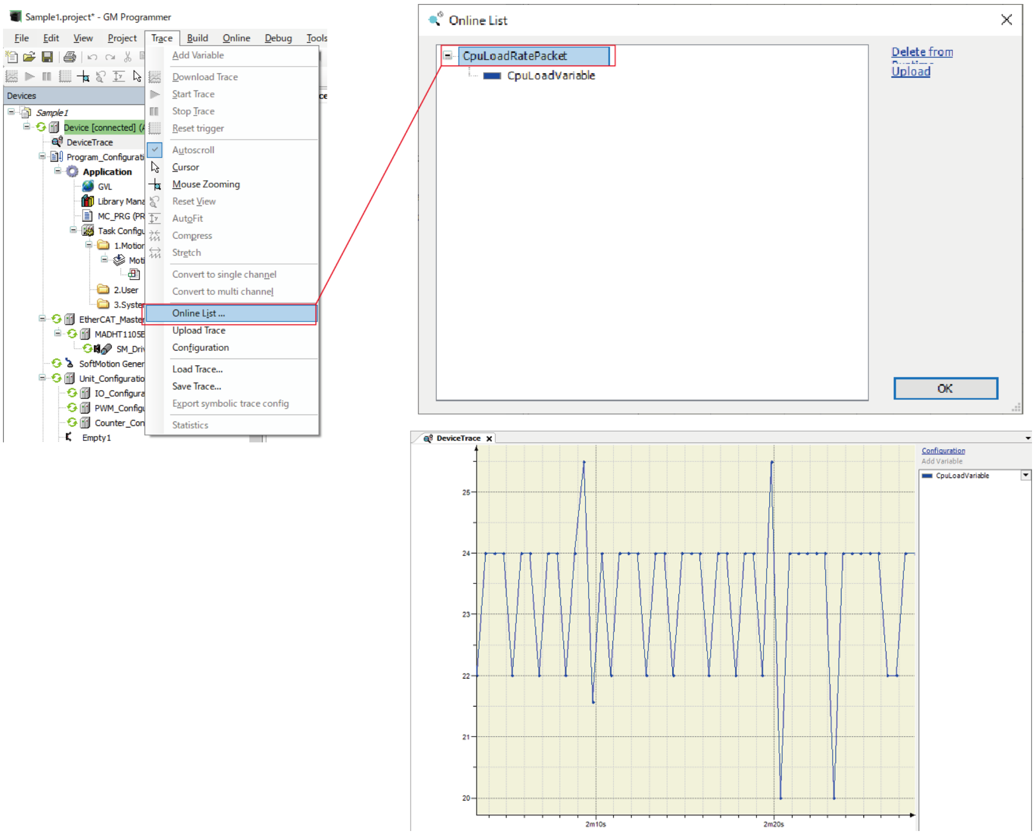


4. From the menu bar, select **Trace**>**Online List...**.

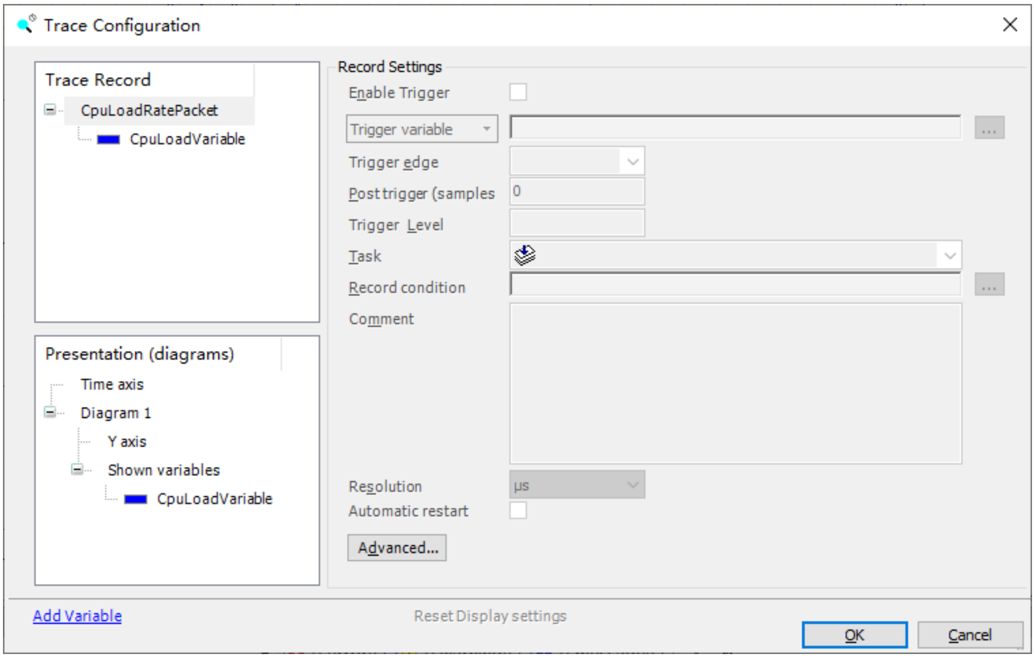
5. The "Online List" window will be displayed. With "CpuLoadRatePacket" selected, click "Upload".

   Plotting the CPU load factor will start.

   Note: As the Online List window is still open, click the [OK] button to close the window.



6. To change the graph display settings, click "Configuration" in the top right corner of the "DeviceTrace" window to open the "Trace Configuration" window.


Trace Configuration

Trace Record
- CpuLoadRatePacket
  - CpuLoadVariable

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - CpuLoadVariable

Record Settings

Enable Trigger □

Trigger variable

Trigger edge

Post trigger (samples 0

Trigger Level

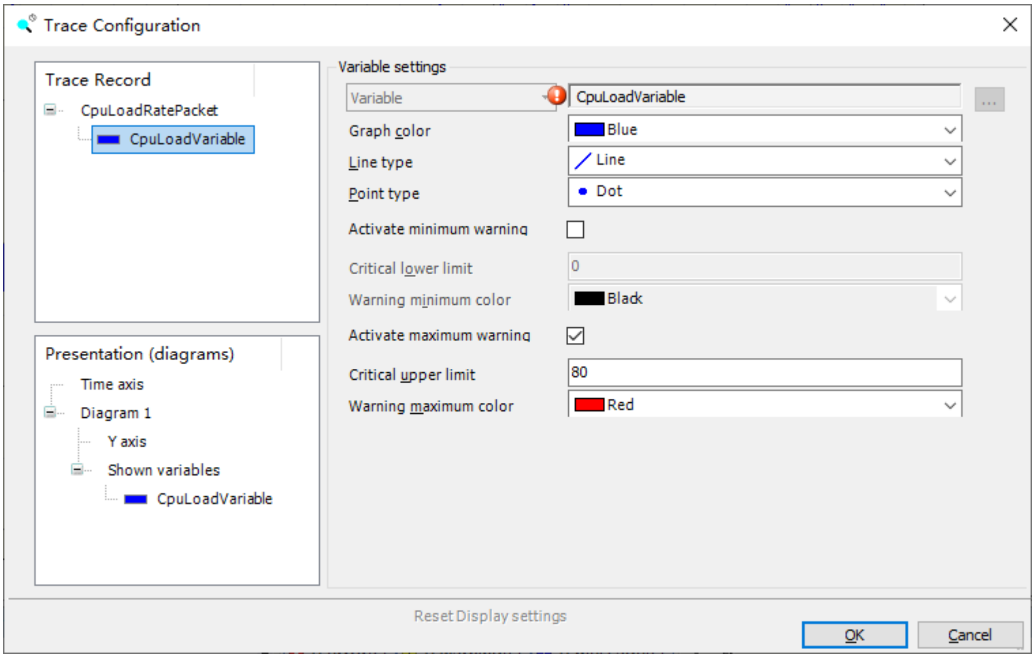
Task

Record condition

Comment

Resolution µs

Automatic restart □



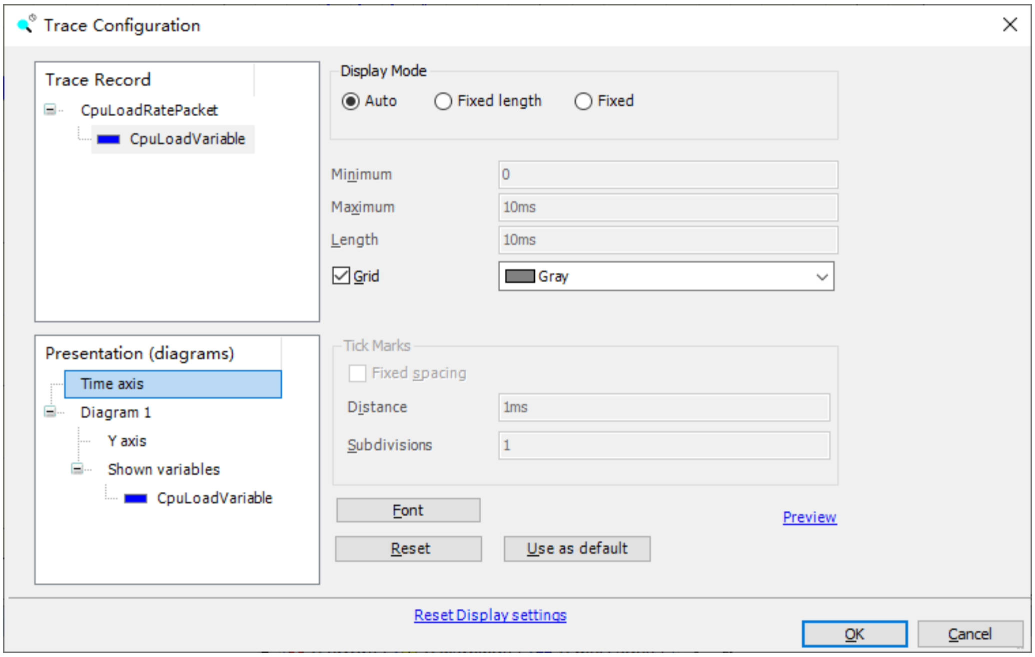
Trace Configuration

Trace Record
- CpuLoadRatePacket
  - CpuLoadVariable

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - CpuLoadVariable

Variable settings

Variable	CpuLoadVariable
Graph color	Blue
Line type	Line
Point type	Dot
Activate minimum warning	□
Critical lower limit	0
Warning minimum color	Black
Activate maximum warning	☑
Critical upper limit	80
Warning maximum color	Red



Trace Configuration

Trace Record
- CpuLoadRatePacket
  - CpuLoadVariable

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - CpuLoadVariable

Display Mode

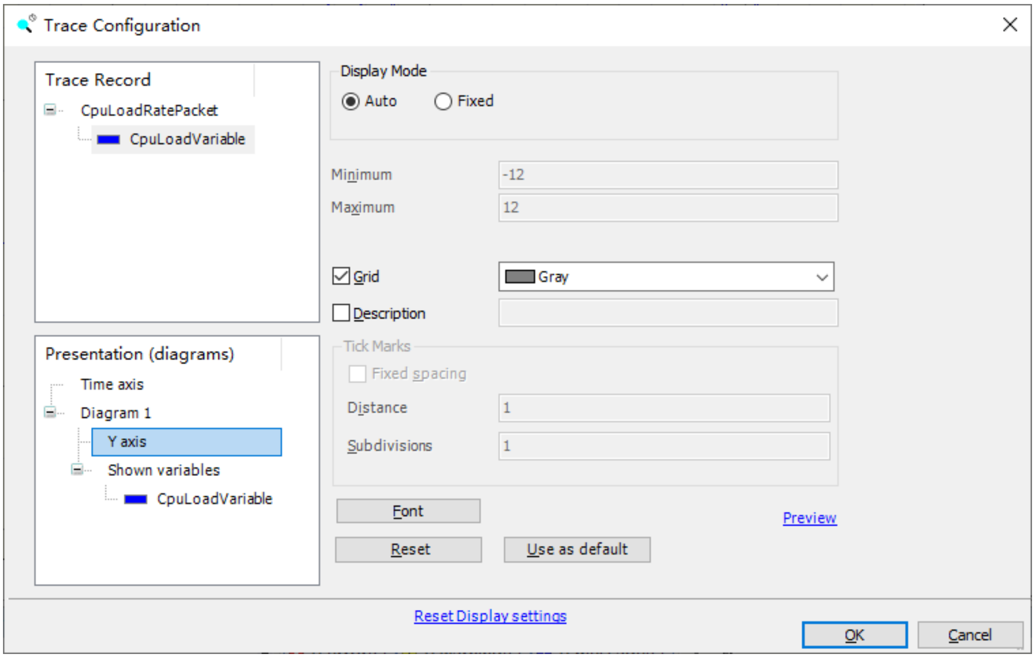
● Auto ○ Fixed length ○ Fixed

Minimum	0
Maximum	10ms
Length	10ms
☑ Grid	Gray

Tick Marks

□ Fixed spacing

Distance	1ms
Subdivisions	1



Trace Configuration

Trace Record
- CpuLoadRatePacket
  - CpuLoadVariable

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - CpuLoadVariable

Display Mode: ● Auto ○ Fixed

Minimum: -12

Maximum: 12

☑ Grid: Gray

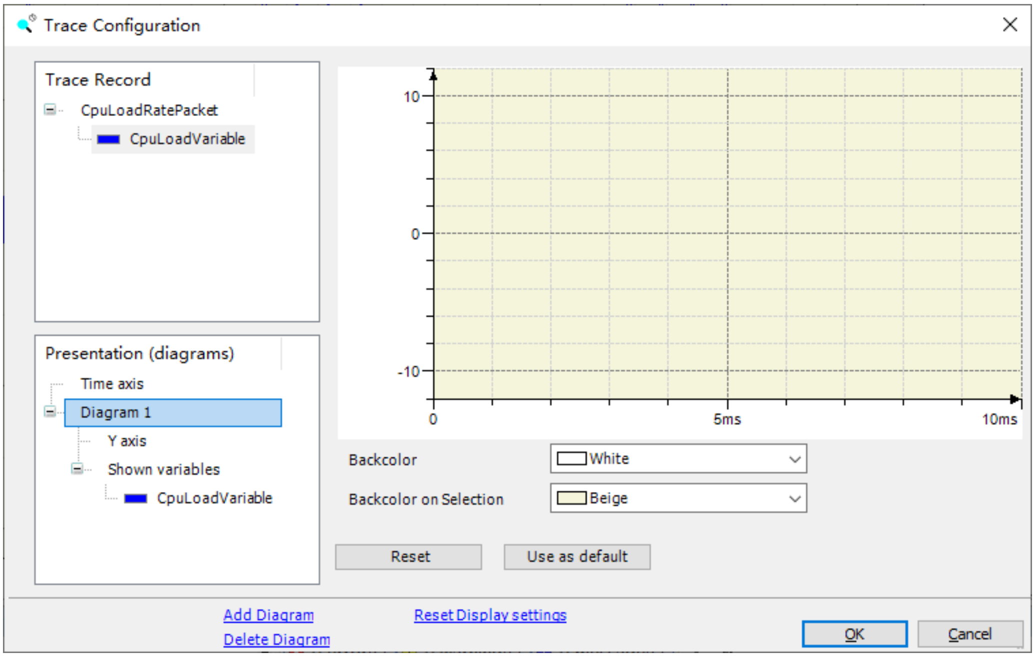
□ Description

Tick Marks

□ Fixed spacing

Distance: 1

Subdivisions: 1



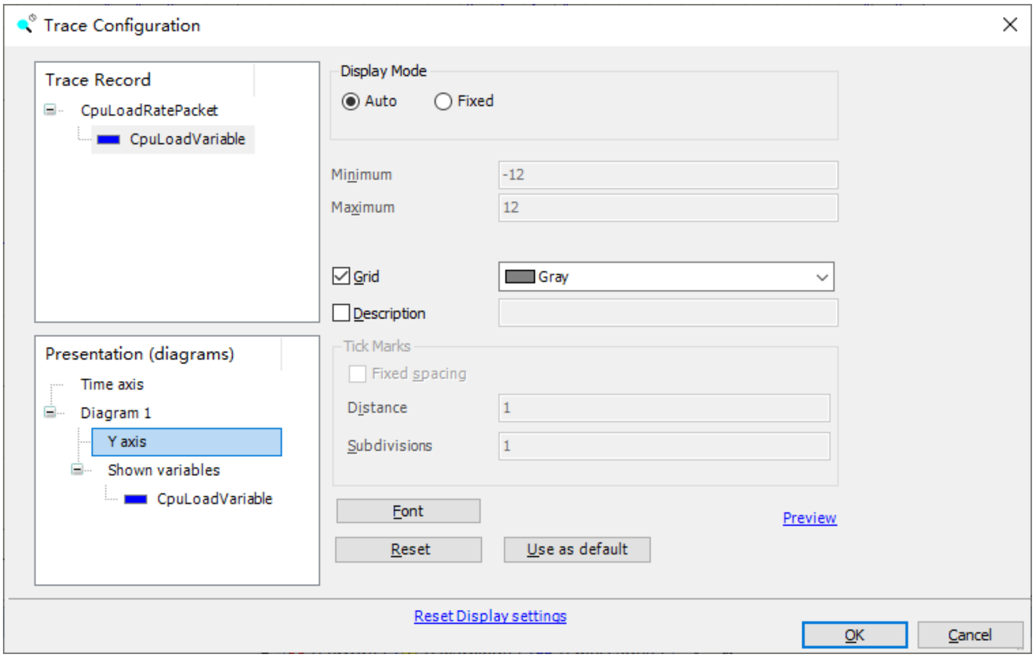
Trace Configuration

Trace Record
- CpuLoadRatePacket
  - CpuLoadVariable

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - CpuLoadVariable

Backcolor: White

Backcolor on Selection: Beige



Trace Configuration

Trace Record
- CpuLoadRatePacket
  - CpuLoadVariable

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - CpuLoadVariable

Display Mode
● Auto ○ Fixed

Minimum -12

Maximum 12

☑ Grid Gray

□ Description

Tick Marks

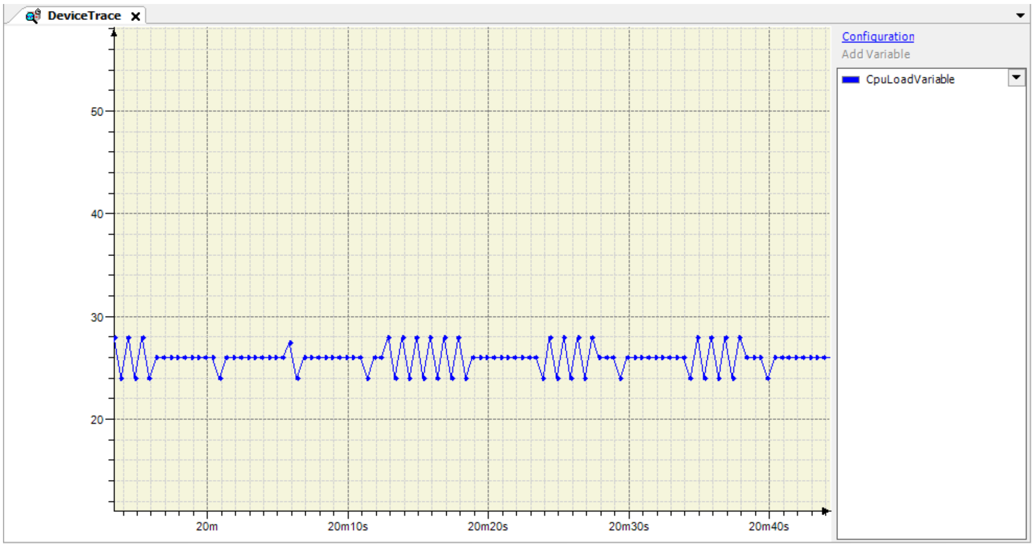
□ Fixed spacing

Distance 1

Subdivisions 1

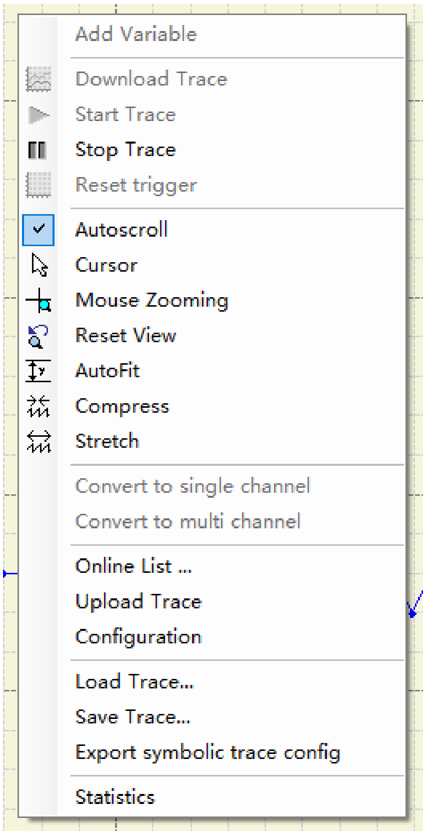
7. The following operations can be performed on the graph.

Dragging the mouse:	To move the time axis
Dragging the mouse while holding down the Ctrl key:	To move the Y-axis
Scrolling the screen:	To lengthen or shorten the time axis
Scrolling the screen while holding down the Ctrl key:	To lengthen or shorten the Y-axis



8. The following right-click menu items can be selected.

Save Trace:	Allows you to save the data plotted on the graph as a file
Load Trace:	Allows you to load the trace file saved by selecting "Save Trace" onto the graph screen



Note

- Use the GM1 controller so that the average CPU load factor is no more than 90%. If 90% is exceeded, stable operation may not be achieved.

9.8 Checking the Performance of GM1 Controller

The GM1 Series motion controller is a system in which multiple tasks run.

To ensure that the GM1 controller operates as a system normally, CPU resources must be allocated to each task properly as described below.

- If CPU resources are insufficient:

  MotionTask and cyclic tasks cannot be operated at the specified intervals. Refer to "9.7 Device Trace Function" and keep the CPU load factor at 80% or less as a guideline.
- If the CPU load factor exceeds 80%:

  Review the program in either one of the following ways.
  - Extend the interval of MotionTask and cyclic tasks.
  - For the program that does not require high speed processing, reassign to the tasks with long intervals.

Also, if the CPU load factor is high, a missing EtherCAT packet may occur. The following section explains how to check a missing EtherCAT packet.

9.8.1 Checking Missing EtherCAT Packets

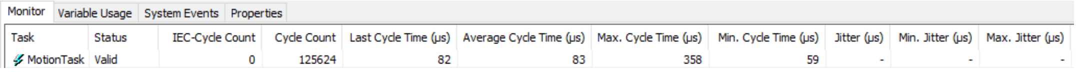
The GM1 controller allows POU to be executed by allocating the POU to a task. To execute POU normally, the processing time (“cycle time") of the task must be smaller than the specified interval (*1) of the task.

*1: For "MotionTask", the specified interval is equivalent to the cycle time (control cycle) of "EtherCAT_Master_SoftMotion".

■ Task processing time

Task processing time can be viewed via the navigator pane and **Task Configuration**>**Monitor**.

In the following example, cycle times of "MotionTask" can be viewed.

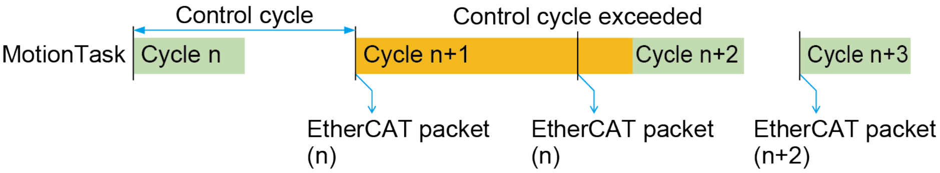


Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)
MotionTask	Valid	0	125624	82	83	358	59	-	-	-

Because cyclic communications are performed over the EtherCAT network, if the cycle time exceeds the task interval, the EtherCAT packet will not be updated in the next cycle, causing the commanded position to remain the same as the previous value. (Missing EtherCAT packet)

■ Example of a Missing EtherCAT Packet

In the following example, a missing EtherCAT packet occurs because the cycle time of cycle n +1 exceeds the cycle time (control cycle).



The occurrence of missing EtherCAT packet can be checked in the following ways.

1. Checking in POU

   Using the "IoDrVEtherCAT.GetStatistics.udiLostFrameCount" method, the occurrence situtation of the missing EtherCAT packet can be checked on the program.

For details on "IoDrVEtherCAT.GetStatistics.udiLostFrameCount", refer to the *GM1 Series Reference Manual (Instruction Edition)*.

2. Checking with GM Programmer settings

   Using the task watchdog timer function makes it possible to check whether the cycle time has exceeded the target value. In the example shown in the figure above, if time=1ms and sensitivity=1 are set, a watchdog timer error will occur in cycle n+1, causing the program to stop.

9.8.2 Performance Check Based on Device Trace

The GM1 controller allows multiple tasks to run. However, use the GM1 controller so that the CPU load factor (processing load of all tasks) is no more than 80%. The CPU load factor can be checked using the"Device Trace"function.

The following section describes how to investigate the influence of the load factor.

1 2 Procedure

1. Using "DeviceTrace", check the state of the load applied to the CPU in operation.

   For details on how to use the Device Trace function, refer to "9.7 Device Trace Function".

2. With the GM1 controller, change the program or change the program in the motion task to user task so that the CPU load factor becomes 90% or less. As an alternative method, change the task period to reduce the load.

Info.

- If the CPU load factor constantly exceeds 95%, the GM1 controller will judge the system to be out of control, causing the system to terminate with an error.

9.9 Error Notification Function

9.9.1 Overview of Errors

The GM1 controller has a self-diagnostic function which identifies errors and stops operation if necessary.

Indications concerning self-diagnosis are as follows.

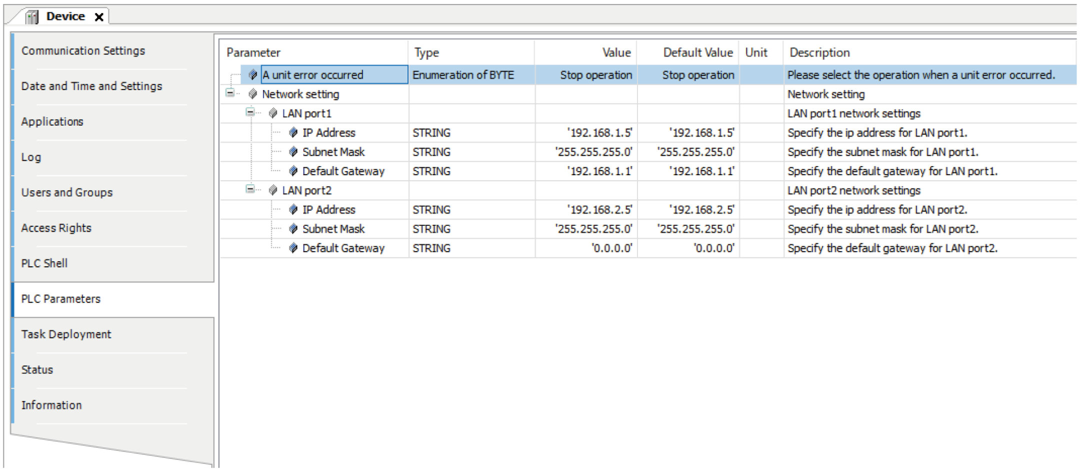
■ LEDs related to self-diagnostic errors

●: Lit, ▲: Flashing, ○: Unlit, ―: Indefinite (Lit or unlit)

	LED display				Description	Operation status
	RUN	STOP	ERROR	ALARM		
Normal	●	○	○	○	Normal operation	Operating
	○	●	○	○	STOP mode	Stopped
Error	●	○	▲	○	When a self-diagnostic error occurs (Operation continues.)	Operating
	○	●	▲	○	When a self-diagnostic error occurs (Operation stops.)	Stopped
	○	●	-	●	System error	Stopped

■ PLC parameter setting

Operation mode at the time of error can be set to continue operation or stop operation in the PLC parameter setting.



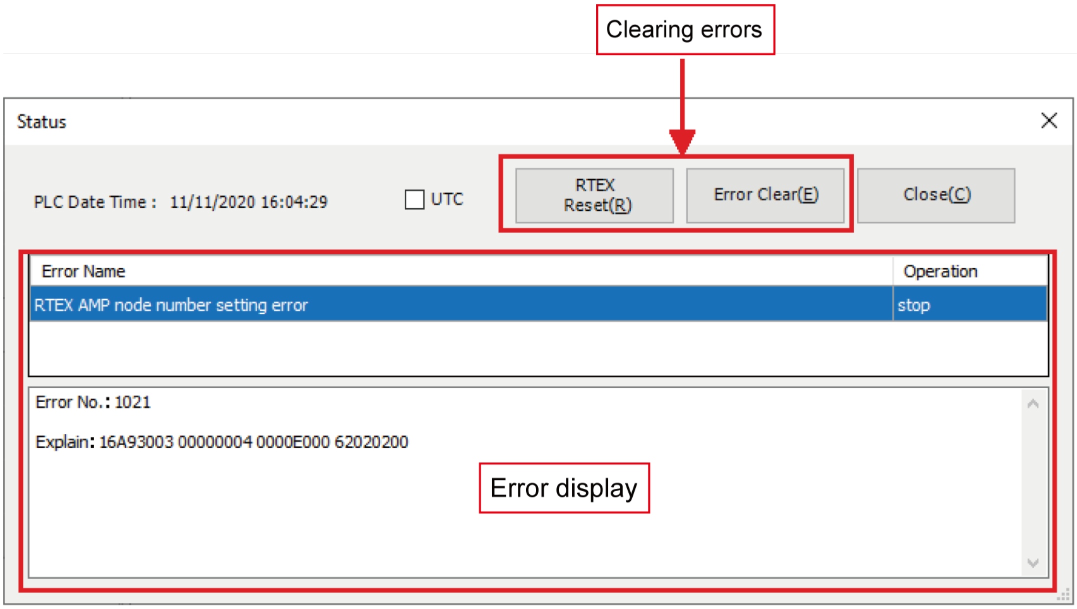
Parameter	Type	Value	Default Value	Unit	Description
A unit error occurred	Enumeration of BYTE	Stop operation	Stop operation		Please select the operation when a unit error occurred.
Network setting					Network setting
LAN port1					LAN port1 network settings
IP Address	STRING	'192.168.1.5'	'192.168.1.5'		Specify the ip address for LAN port1.
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'		Specify the subnet mask for LAN port1.
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'		Specify the default gateway for LAN port1.
LAN port2					LAN port2 network settings
IP Address	STRING	'192.168.2.5'	'192.168.2.5'		Specify the ip address for LAN port2.
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'		Specify the subnet mask for LAN port2.
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'		Specify the default gateway for LAN port2.

9.9.2 Checking and Clearing Errors Using GM Programmer

Error information can be checked in the status window of GM Programmer.
In case of an operation continue error, the error can be resolved by Error Clear.

Info.

- Since the error resolution method varies depending on the error, refer to "9.9.5 Error Code List".



PLC Date Time : 11/11/2020 16:04:29 □ UTC

Error Name	Operation
RTEX AMP node number setting error	stop

Error No.: 1021

Explain: 16A93003 00000004 0000E000 62020200

9.9.3 Obtaining Error Information Using User Programs

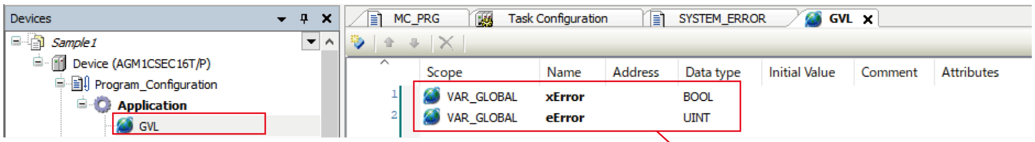
The following function block can be used to obtain error information for the GM1 controller.

- SYS_SysGetSystemError

This function block is used to obtain error information for the GM1 controller from external devices such as display units.

1 2 Procedure

1. Define the variables to be used in the SYS_SysGetSystemError function block, as global variables.

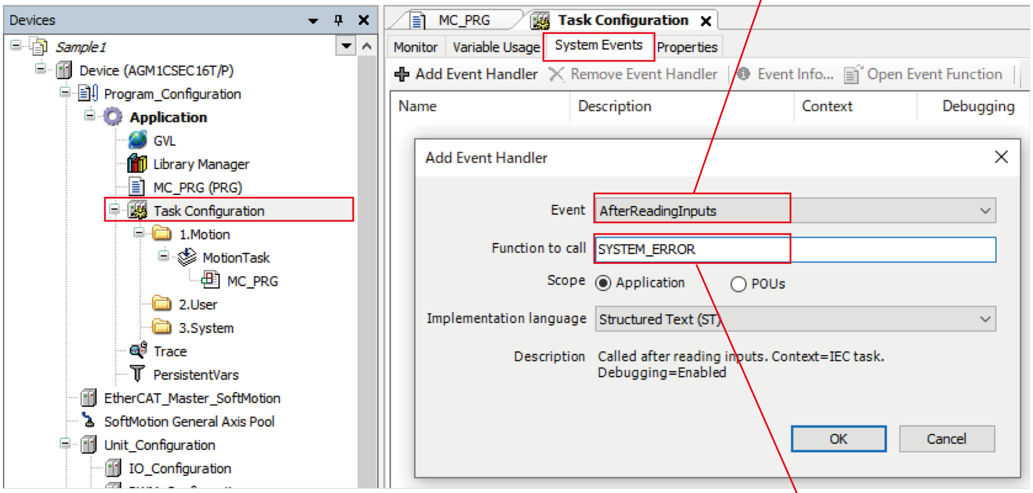


Scope	Name	Address	Data type	Initial Value	Comment	Attributes
VAR_GLOBAL	xError		BOOL			
VAR_GLOBAL	eError		UINT			

Define global variable that are the same data type as parameters of SYS_SysGetSystemError.

2. Select "Task Configuration" and then the "System Events" tab, and register the function to be executed when a particular event occurs.

Select an event.
Select “After Reading Inputs”.



Add Event Handler

Event: AfterReadingInputs

Function to call: SYSTEM_ERROR

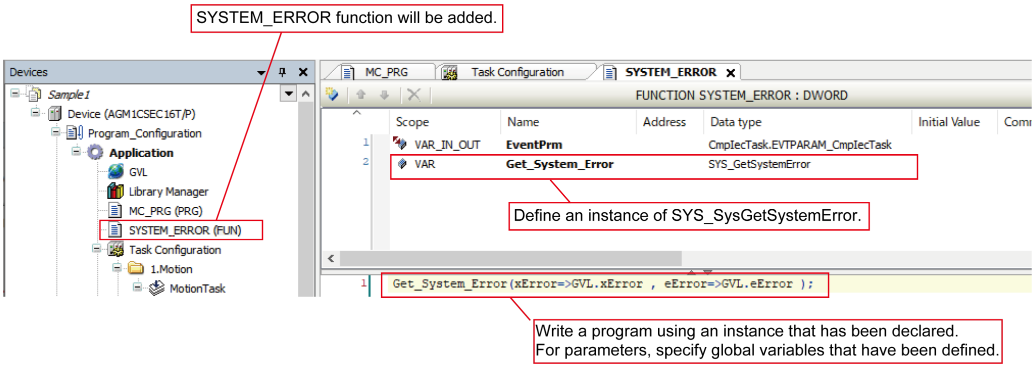
Scope: ● Application ○ POUs

Implementation language: Structured Text (ST)

Description: Called after reading inputs. Context=IEC task.
Debugging=Enabled

Enter the function name (arbitrary name) to be executed when the event occurs.

3. In the function that has been added, write a program for SYS_SysGetSystemError.



SYSTEM_ERROR function will be added.

FUNCTION SYSTEM_ERROR : DWORD

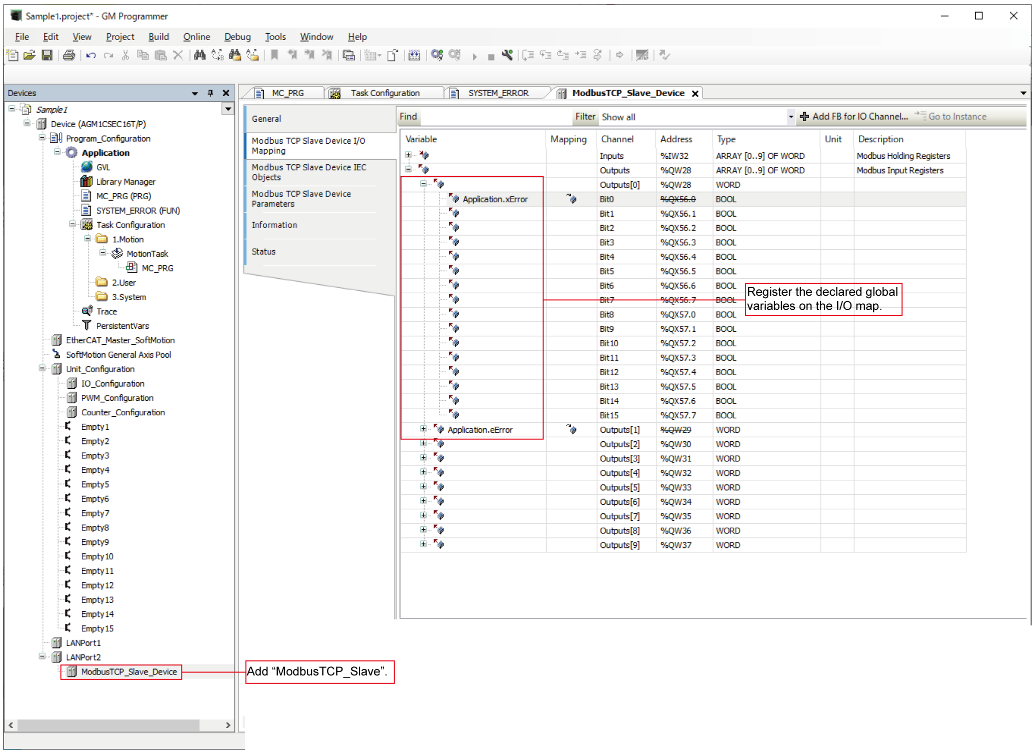
	Scope	Name	Address	Data type	Initial Value	Comı
1	VAR_IN_OUT	EventPrm		CmpIecTask.EVTPARAM_CmpIecTask		
2	VAR	Get_System_Error		SYS_GetSystemError		

Define an instance of SYS_SysGetSystemError.

```
Get_System_Error(xError=>GVL.xError , eError=>GVL.eError );
```

Write a program using an instance that has been declared.
For parameters, specify global variables that have been defined.

4. Map the defined global variables to the I/O map of the ModbusTCP Slave Device to allow external devices to get error information for the GM1 controller.

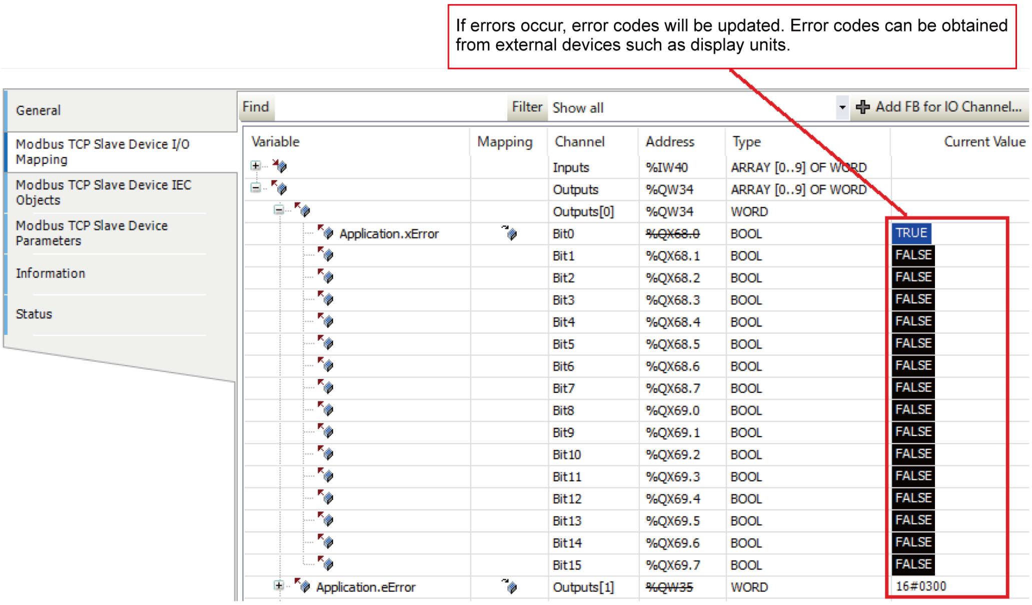


Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%IW32	ARRAY [0..9] OF WORD		Modbus Holding Registers
		Outputs	%QW28	ARRAY [0..9] OF WORD		Modbus Input Registers
		Outputs[0]	%QW28	WORD		
Application.xError		Bit0	~~%QX56.0~~	BOOL		
		Bit1	%QX56.1	BOOL		
		Bit2	%QX56.2	BOOL		
		Bit3	%QX56.3	BOOL		
		Bit4	%QX56.4	BOOL		
		Bit5	%QX56.5	BOOL		
		Bit6	%QX56.6	BOOL		
		Bit7	%QX56.7	BOOL		
		Bit8	%QX57.0	BOOL		
		Bit9	%QX57.1	BOOL		
		Bit10	%QX57.2	BOOL		
		Bit11	%QX57.3	BOOL		
		Bit12	%QX57.4	BOOL		
		Bit13	%QX57.5	BOOL		
		Bit14	%QX57.6	BOOL		
		Bit15	%QX57.7	BOOL		
Application.eError		Outputs[1]	~~%QW29~~	WORD		
		Outputs[2]	%QW30	WORD		
		Outputs[3]	%QW31	WORD		
		Outputs[4]	%QW32	WORD		
		Outputs[5]	%QW33	WORD		
		Outputs[6]	%QW34	WORD		
		Outputs[7]	%QW35	WORD		
		Outputs[8]	%QW36	WORD		
		Outputs[9]	%QW37	WORD		

Register the declared global variables on the I/O map.

Add “ModbusTCP_Slave”.

5. If an error occurs, the error code will be set in the variable, so that external devices can get error information for the GM1 controller.



If errors occur, error codes will be updated. Error codes can be obtained from external devices such as display units.

General
Modbus TCP Slave Device I/O Mapping
Modbus TCP Slave Device IEC Objects
Modbus TCP Slave Device Parameters
Information
Status
Find
Filter Show all
Add FB for IO Channel...

Variable	Mapping	Channel	Address	Type	Current Value
		Inputs	%IW40	ARRAY [0..9] OF WORD	
		Outputs	%QW34	ARRAY [0..9] OF WORD	
		Outputs[0]	%QW34	WORD	
Application.xError		Bit0	~~%QX68.0~~	BOOL	TRUE
		Bit1	%QX68.1	BOOL	FALSE
		Bit2	%QX68.2	BOOL	FALSE
		Bit3	%QX68.3	BOOL	FALSE
		Bit4	%QX68.4	BOOL	FALSE
		Bit5	%QX68.5	BOOL	FALSE
		Bit6	%QX68.6	BOOL	FALSE
		Bit7	%QX68.7	BOOL	FALSE
		Bit8	%QX69.0	BOOL	FALSE
		Bit9	%QX69.1	BOOL	FALSE
		Bit10	%QX69.2	BOOL	FALSE
		Bit11	%QX69.3	BOOL	FALSE
		Bit12	%QX69.4	BOOL	FALSE
		Bit13	%QX69.5	BOOL	FALSE
		Bit14	%QX69.6	BOOL	FALSE
		Bit15	%QX69.7	BOOL	FALSE
Application.eError		Outputs[1]	~~%QW35~~	WORD	16#0300

9.9.4 Error Recovery Processing

Info.

- For errors during simulation, check the log window or each function block.

When an error occurs, normally, stop the operation.

■ When ERROR_LED flashes

A self-diagnostic error has occurred.

■ Solution

Check the condition according to the following procedure.

1. On the GM Programmer, select **Online**>**Status** and check the error content (error code).
2. Switch the mode to the PROG mode.
3. Cancel the situation in accordance with the error code.

■ When ALARM_LED lights up

Timeout of the system watchdog timer has been detected.

■ Solution

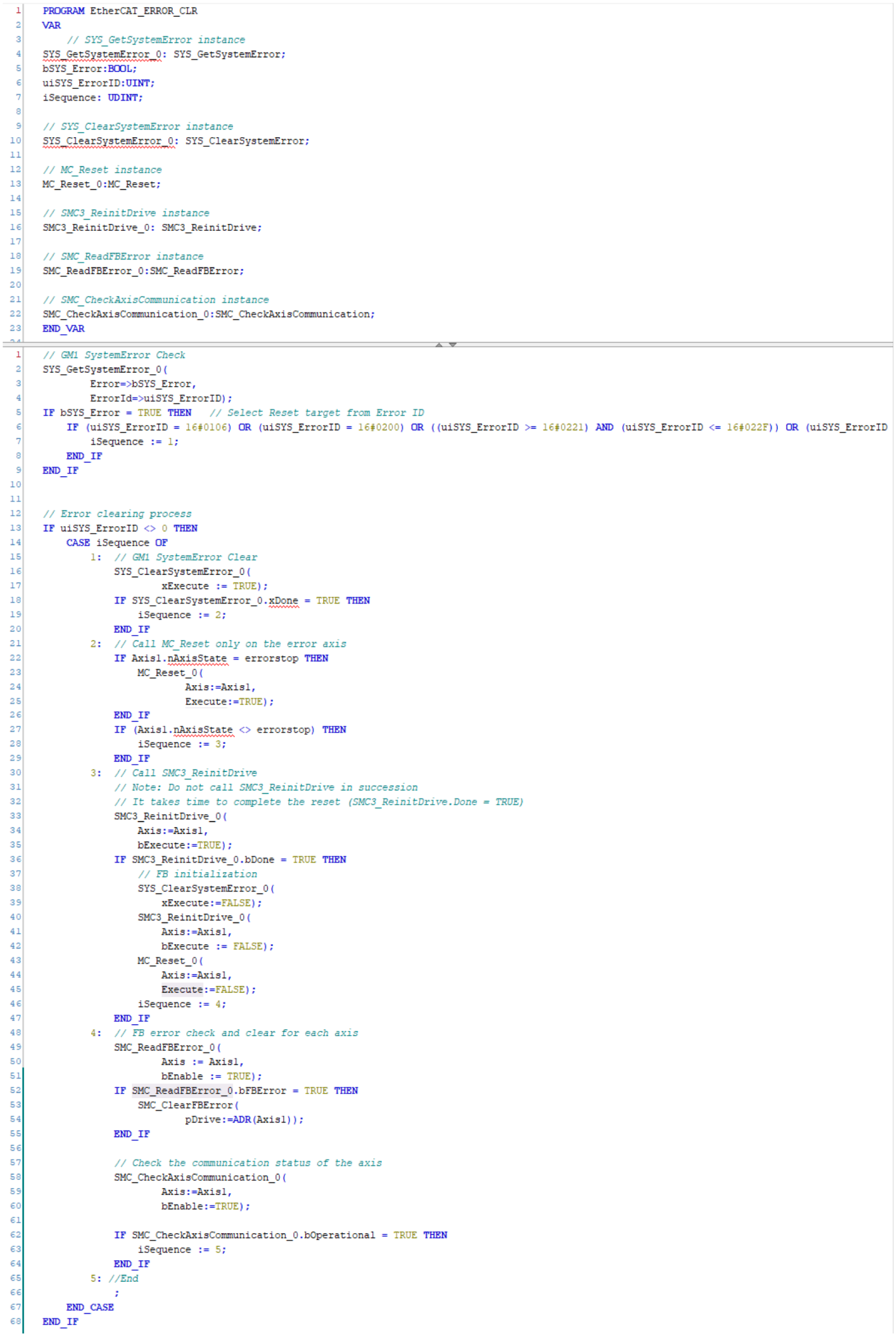
Check the condition according to the following procedure.

1. Turn the controller OFF and then ON.
   If the problem persists, consult your Panasonic representative.

■ Sample code

For a program that is used for error recovery processing, refer to the following.

<Sample program>



```
PROGRAM EtherCAT_ERROR_CLR
VAR
    // SYS_GetSystemError instance
SYS_GetSystemError_0: SYS_GetSystemError;
bSYS_Error:BOOL;
uiSYS_ErrorID:UINT;
iSequence: UDINT;

// SYS_ClearSystemError instance
SYS_ClearSystemError_0: SYS_ClearSystemError;

// MC_Reset instance
MC_Reset_0:MC_Reset;

// SMC3_ReinitDrive instance
SMC3_ReinitDrive_0: SMC3_ReinitDrive;

// SMC_ReadFBError instance
SMC_ReadFBError_0:SMC_ReadFBError;

// SMC_CheckAxisCommunication instance
SMC_CheckAxisCommunication_0:SMC_CheckAxisCommunication;
END_VAR
```

```
// GM1 SystemError Check
SYS_GetSystemError_0(
        Error=>bSYS_Error,
        ErrorId=>uiSYS_ErrorID);
IF bSYS_Error = TRUE THEN   // Select Reset target from Error ID
    IF (uiSYS_ErrorID = 16#0106) OR (uiSYS_ErrorID = 16#0200) OR ((uiSYS_ErrorID >= 16#0221) AND (uiSYS_ErrorID <= 16#022F)) OR (uiSYS_ErrorID
        iSequence := 1;
    END_IF
END_IF


// Error clearing process
IF uiSYS_ErrorID <> 0 THEN
    CASE iSequence OF
        1:  // GM1 SystemError Clear
            SYS_ClearSystemError_0(
                    xExecute := TRUE);
            IF SYS_ClearSystemError_0.xDone = TRUE THEN
                iSequence := 2;
            END_IF
        2:  // Call MC_Reset only on the error axis
            IF Axis1.nAxisState = errorstop THEN
                MC_Reset_0(
                        Axis:=Axis1,
                        Execute:=TRUE);
            END_IF
            IF (Axis1.nAxisState <> errorstop) THEN
                iSequence := 3;
            END_IF
        3:  // Call SMC3_ReinitDrive
            // Note: Do not call SMC3_ReinitDrive in succession
            // It takes time to complete the reset (SMC3_ReinitDrive.Done = TRUE)
            SMC3_ReinitDrive_0(
                Axis:=Axis1,
                bExecute:=TRUE);
            IF SMC3_ReinitDrive_0.bDone = TRUE THEN
                // FB initialization
                SYS_ClearSystemError_0(
                    xExecute:=FALSE);
                SMC3_ReinitDrive_0(
                    Axis:=Axis1,
                    bExecute := FALSE);
                MC_Reset_0(
                    Axis:=Axis1,
                    Execute:=FALSE);
                iSequence := 4;
            END_IF
        4:  // FB error check and clear for each axis
            SMC_ReadFBError_0(
                    Axis := Axis1,
                    bEnable := TRUE);
            IF SMC_ReadFBError_0.bFBError = TRUE THEN
                SMC_ClearFBError(
                        pDrive:=ADR(Axis1));
            END_IF

            // Check the communication status of the axis
            SMC_CheckAxisCommunication_0(
                    Axis:=Axis1,
                    bEnable:=TRUE);

            IF SMC_CheckAxisCommunication_0.bOperational = TRUE THEN
                iSequence := 5;
            END_IF
        5: //End
            ;
    END_CASE
END_IF
```

9.9.5 Error Code List

Error No.	Operation	Name	Error description and action method	Recovery method
0x0001	Stop	System error (serious)	Notify us of the error information or system data history information displayed on the status window of GM Programmer.	Power cycle
0x0002	Stop	System error (CODESYS)	Notify us of the error information or system data history information displayed on the status window of GM Programmer.	Power cycle
0x0003	Continue	System error (minor)	Notify us of the error information or system data history information displayed on the status window of GM Programmer.	Power cycle
0x0106	Continue	Device date and time RTC error	Because the duration of the power failure exceeded the guaranteed period (14 days), the RTC time information was initialized. The device must be left ON for at least 10 minutes. Note: Time information is reset to April 1, 2019.	Clear the error
0x0110 to 0x113	Continue	PWM Cycle setting error	The cycle setting is out of range. Check whether the duty ratio is within the allowable range.	Re-download
0x0120 to 0x123	Continue	PWM Duty ratio setting error	The duty ratio setting is out of range. Check whether the duty ratio is within the allowable range.	Re-download
0x0130 to 0x131	Stop	Counter Overflow error	The counter value exceeds the upper limit. Perform reset or preset operation so that the counter value falls within the range between the upper and lower limits.	Re-download
0x0140 to 0x141	Stop	Counter Underflow error	The counter value drops below the lower limit. Perform reset or preset operation so that the counter value falls within the range between the upper and lower limits.	Re-download
0x0150 to 0x151	Continue	Counter Reset abnormal error	The count value becomes 0 under the reset conditions and goes out of the range between the upper and lower limits. Check the settings of the upper and lower limits of the counter. If 0 does not exist in the range between the upper and lower limits, make a preset request.	Re-download
0x0160 to 0x161	Continue	Counter Preset abnormal error	The count value goes out of the range between the upper and lower limits under the preset conditions. Check whether the preset value falls within the range between the upper and lower limits.	Re-download
0x0170 to 0x171	Continue	Counter Current value change abnormal error	The count value goes out of the range between the upper and lower limits when the current value is changed. After the current value is changed, check whether the new value falls within the range between the upper and lower limits.	Re-download
0x0200	Continue	Expansion unit Communication error	An expansion unit communication error has occurred. Check connections.	Clear the error
0x0201	Stop	Expansion unit	More than 15 expansion units are connected. Check connections.	Power cycle

Error No.	Operation	Name	Error description and action method	Recovery method
		Number of connections exceeded		
0x0203	Stop	Expansion unit Startup error	An attempt to upgrade the expansion unit could have failed. Upgrade again.	Re-download
0x0205	Stop	Expansion unit Startup wait timeout error	Waiting for expansion unit startup has timed out. Check connections.	Re-download
0x020C	Stop	Expansion unit Connection number mismatch	The number of expansion units in the project does not match the number of expansion units mounted. Check connections.	Re-download
0x020D	Stop	Expansion unit Model code mismatch	The expansion unit model in the project does not match the expansion unit model mounted. Check connections.	Re-download
0x020E	Stop	Expansion unit Version mismatch	The version of the expansion unit registered in the project does not match the version of the expansion unit mounted.	Re-download
0x020F	Stop	Expansion unit Unit initialization error	Expansion unit initialization has failed. Check connections.	Re-download
0x0221 to 0x022F	Continue	Expansion unit I/O data error	An I/O data error has occurred in the expansion unit. Check the installation environment.	Clear the error
0x0300	Continue	CODESYS error	A CODESYS error has occurred. Check the error details in the log window of GM Programmer.	Clear the error

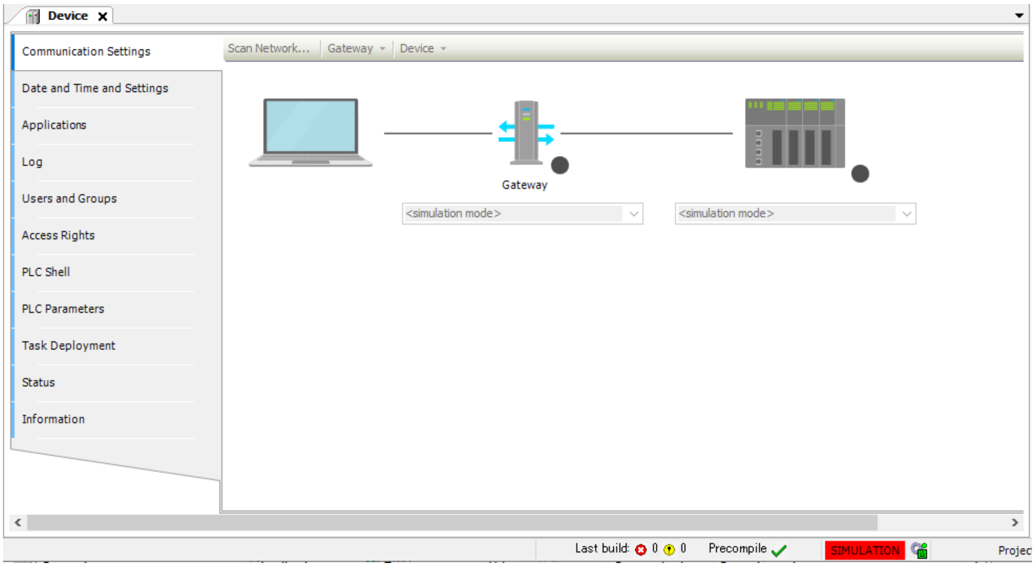
10 Useful Functions of GM Programmer

10.1 Simulation Function

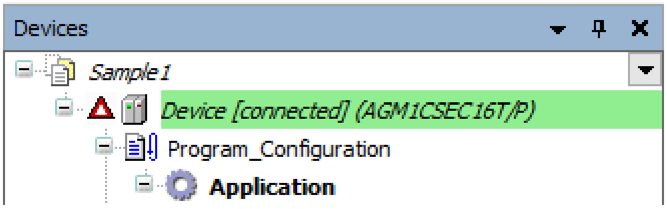
Simulation mode allows the user to perform a login operation without connecting to the GM1 controller. It also allows the user to check behaviors in the same way as if the user logged in.

1 2 Procedure

1. From the menu bar, select **Online**>**Simulation**.
Simulation mode will be invoked and "Simulation" will be displayed on the status bar.



2. From the menu bar, select **Online**>**Login**, or press the <Alt> key and the <F8> key simultaneously.
Login will occur in simulation mode. When login occurs in simulation mode, the device object is displayed in italic.



Simulation will start.

Info.

- ⚠ is displayed in front of the object of a device that is operated in simulation mode.
- The cycle time in simulation mode may not operate according to the set cycle because of a difference in processing time depending on the operating environment.

10.2 Security Function

GM Programmer is equipped with a security function that can implement user management (project user management and device user management) and encrypt project files. This section explains security-related functions such as user management and project encryption, and the procedures for operating each security function.

Stop

- Each user ID and their corresponding password must be different character strings.
- Password strength must be sufficiently high. Do not use any passwords that can be easily guessed.
- Accounts must be managed properly and must not be shared unnecessarily.

!

- Use this controller in a secure network environment.
- Use encryption functions properly to protect information assets.
- Use the device user management function to perform authentication protection for the controller during operation.
- After logging in with the initial password, be sure to change the password.
- Implement password management to prevent passwords from being forgotten. If the password is forgotten, device reset must be performed on the controller.
- Implement password management to prevent passwords from being leaked to third parties.

Item	Description	Reference page
User management	Allows execution permissions for operations (such as executing menu commands and adding, editing, and deleting objects) to be assigned to each group in which users are registered. User management also enables logins to be permitted by assigning permission for login to the device to each user and entering passwords.	"P.10-3"
Encryption / signature	Provides password-based encryption for project files and encrypts connections between the GM1 controller and the PC.	"P.10-16"
Write-protection	Provides write-protection for project files and prevents project files from being modified unintentionally by mistake.	"P.10-24"

10.2.1 User Management

Project user management allows execution permissions for operations (such as executing menu commands and adding, editing, and deleting objects) to be assigned to each group in which users are registered. Logon must be performed by a user of a privileged group.

Device user management also enables logins to be permitted by assigning permission for login to the device to each user and entering passwords.

Project User Management

When a new project is created, Owner users, Owner group, and Everyone group are already registered.

Owner users belong to Owner group and can execute all operations.

All users including Owner users are automatically registered in Everyone group.

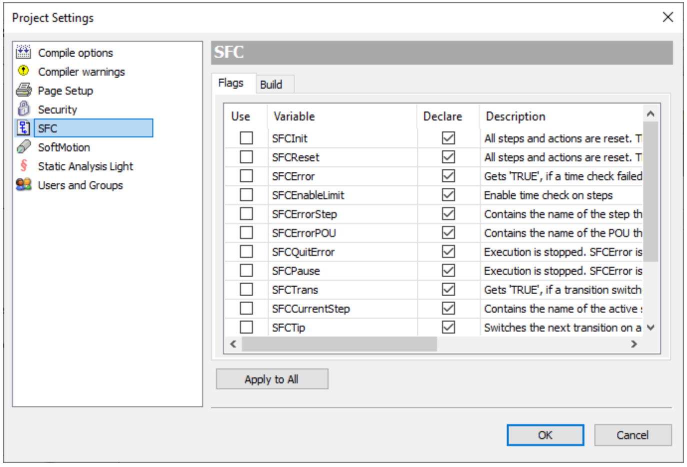
Group	User	Remarks
Owner	Owner	Users can execute all operations. For Owner users, the password field is left blank.
Everyone	Owner	All users are registered automatically.

Creating a New User and Group

In the following example, a group (group name: GroupA) and a user belonging to the group (user name: Fred) are newly created and privileges are set so that users belonging to GroupA can access POU object "POU_1".

Procedure

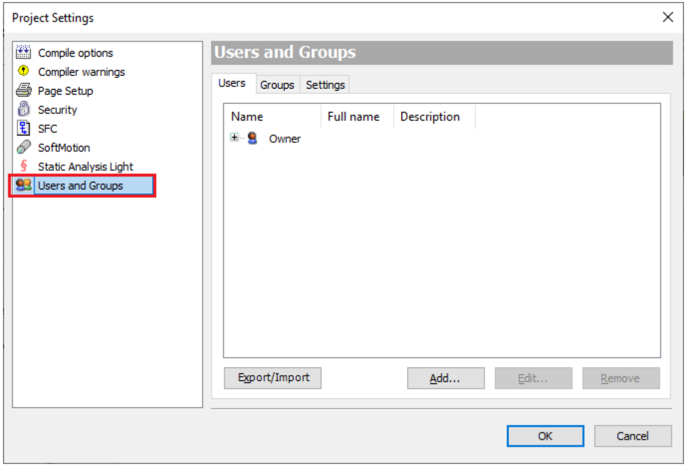
1. From the menu bar, select **Project**>**Project Settings**.
   The "Project Settings" dialog box will be displayed.



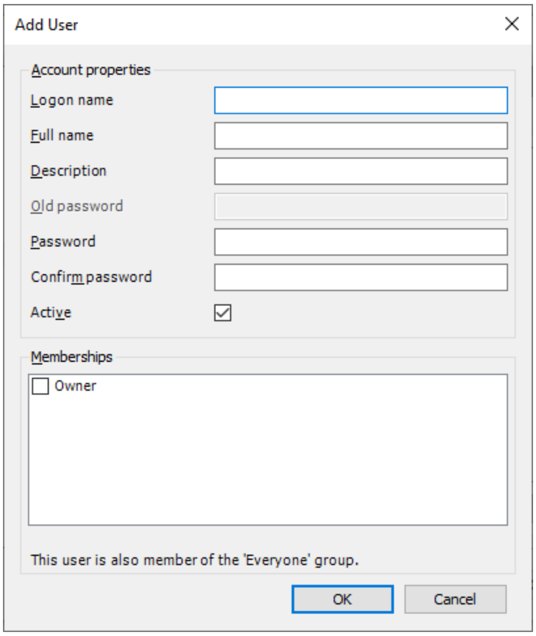
Use	Variable	Declare	Description
□	SFCInit	☑	All steps and actions are reset. T
□	SFCReset	☑	All steps and actions are reset. T
□	SFCError	☑	Gets 'TRUE', if a time check failed
□	SFCEnableLimit	☑	Enable time check on steps
□	SFCErrorStep	☑	Contains the name of the step th
□	SFCErrorPOU	☑	Contains the name of the POU th
□	SFCQuitError	☑	Execution is stopped. SFCError is
□	SFCPause	☑	Execution is stopped. SFCError is
□	SFCTrans	☑	Gets 'TRUE', if a transition switch
□	SFCCurrentStep	☑	Contains the name of the active :
□	SFCTip	☑	Switches the next transition on a

2. In the "Project Settings" dialog box, select the "Users and Groups" category.

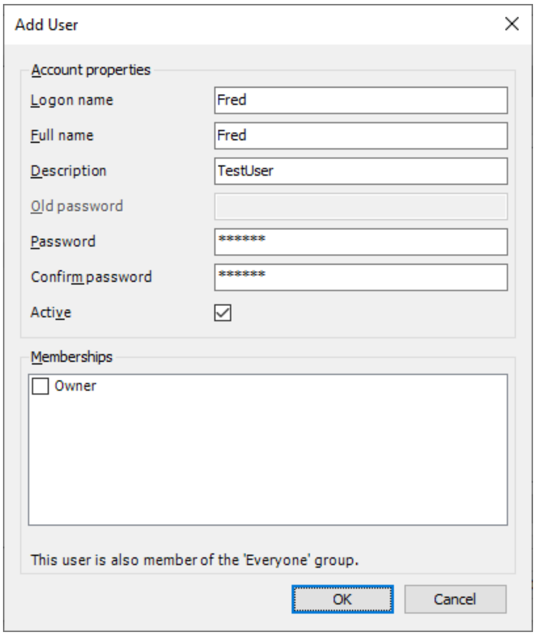
The "Users and Groups" setting pane will be displayed.



3. Click the [Add] button.
The "Add User" dialog box will be displayed.



4. Enter information for a new user to be added.
Enter information about a new user (Fred) to be added.



Add User

Account properties

Logon name: Fred

Full name: Fred

Description: TestUser

Old password:

Password: ******

Confirm password: ******

Active ☑

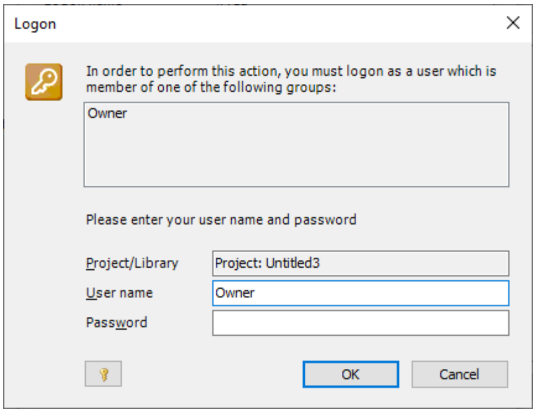
Memberships

□ Owner

This user is also member of the 'Everyone' group.

5. Click the [OK] button.
   The "Logon" dialog box will be displayed. To add a new user, you must log in as an Owner user.

6. In the "User name" field, enter "Owner".
   The default password for "Owner" is not set in the Password field.
   The "Password" field must be left blank.



Logon

In order to perform this action, you must logon as a user which is member of one of the following groups:

Owner

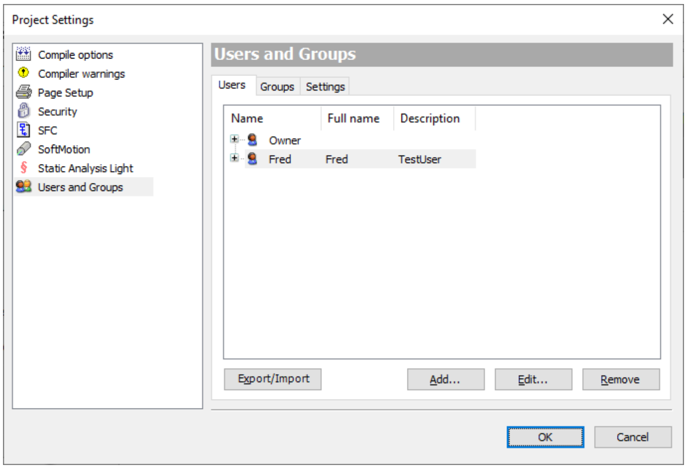
Please enter your user name and password

Project/Library: Project: Untitled3

User name: Owner

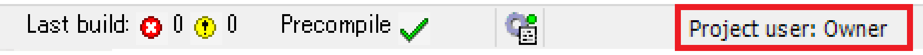
Password:

7. Click the [OK] button.
   Login by the Owner user will be completed and new user "Fred" will be added to the "Users" tab pane.



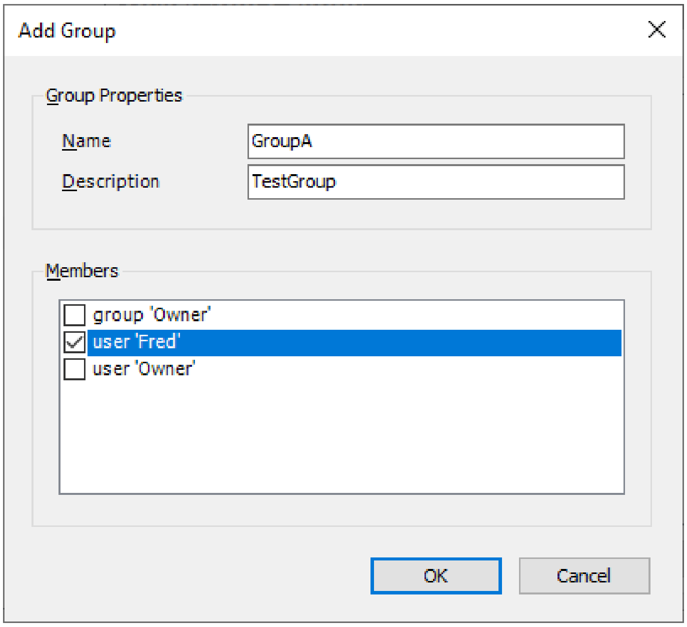
Name	Full name	Description
Owner		
Fred	Fred	TestUser

The name ("Owner") of the user who logged on is displayed in the status field.



8. Select the "Groups" tab and click the [Add] button.
The "Add Group" dialog box will be displayed.

9. Enter information for a new group to be added.
Enter information about a new group (GroupA) to be added.
In the "Members" section, specify a member that belongs to the group. Select the new user added in step "Step 4".

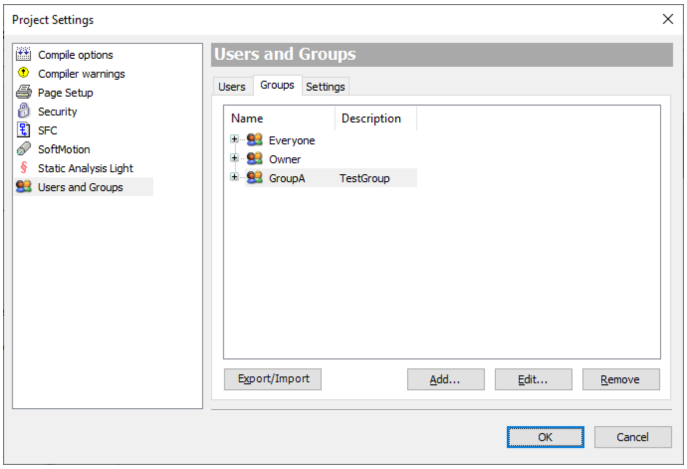


Name: GroupA

Description: TestGroup

- □ group 'Owner'
- ☑ user 'Fred'
- □ user 'Owner'

10. Click the [OK] button.
The new group (GroupA) will be added to the "Groups" tab pane.
User "Fred" is registered in GroupA as a member



11. Click the [OK] button.
The "Project Settings" dialog box will be closed.

This completes the procedure for registering user "Fred" in group "GroupA".
After the procedure is complete, the following groups and users exist in the project.

Group	User	Remarks
Owner	Owner	Group whose users can execute all operations
Everyone	Owner, Fred	Group in which all users are registered automatically
GroupA	Fred	Newly added group

Info.

- User and group information can be exported in XML format. Click the [Export/Import] button in the "Users and Groups" setting pane and select the "Export users and groups" menu item". .users" files can be exported.

  By selecting the "Import User and Group" menu item, you can import ".users" files.

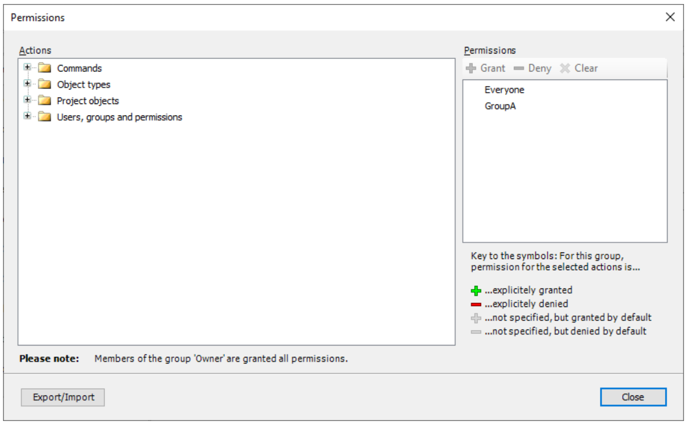
Setting Operation Privileges

In the following example, privileges are set so that users belonging to GroupA can display a POU object (object name: POU_1).

Before performing the following procedure, add a POU object (object name: POU_1) to the project.

Procedure

1. From the menu bar, select **Project**>**User Management**>**Permissions**.
The "Permissions" dialog box will be displayed.

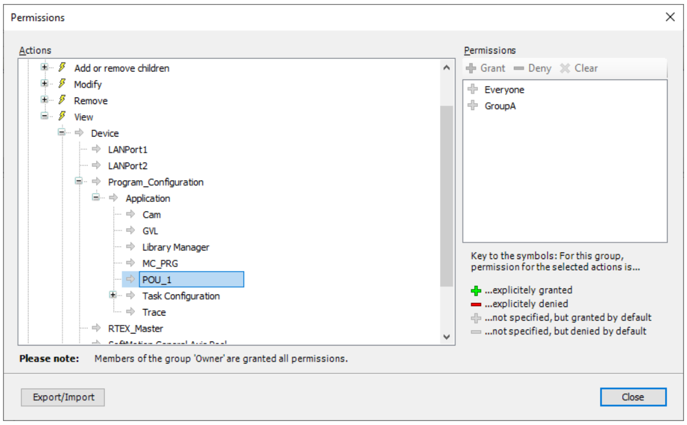


Key to the symbols: For this group, permission for the selected actions is...

✚ ...explicitly granted
▬ ...explicitly denied
✚ ...not specified, but granted by default
▬ ...not specified, but denied by default

Please note: Members of the group 'Owner' are granted all permissions.

2. In the "Action" pane, select the operation to be permitted.
Select **Project Object**>**View**>**Device**>**Program Configuration**>**Application**>**POU_1**.



Key to the symbols: For this group, permission for the selected actions is...

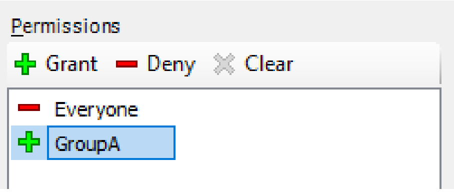
✚ ...explicitly granted
▬ ...explicitly denied
✚ ...not specified, but granted by default
▬ ...not specified, but denied by default

Please note: Members of the group 'Owner' are granted all permissions.

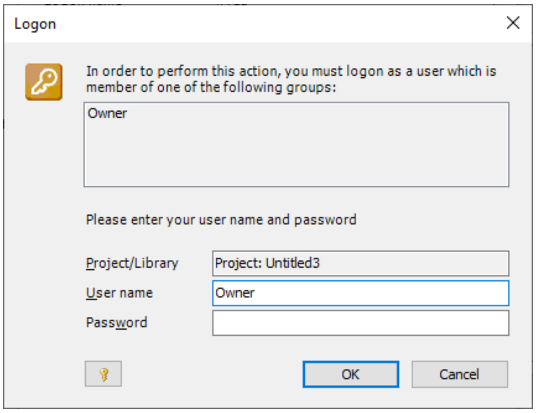
3. In the "Permissions" pane, set privileges to be assigned.

Select "Everyone" and click ▬ Deny. "Reject" will be set.

Select "GroupA" and click ✚ Grant. "Approve" will be set.
This enables only the users of group "GroupA" to display the object.



If the "Logon" dialog box is displayed, enter "Owner" in the "User name" field and leave the "Password" field blank before performing a logon.



In order to perform this action, you must logon as a user which is member of one of the following groups:

Owner

Please enter your user name and password

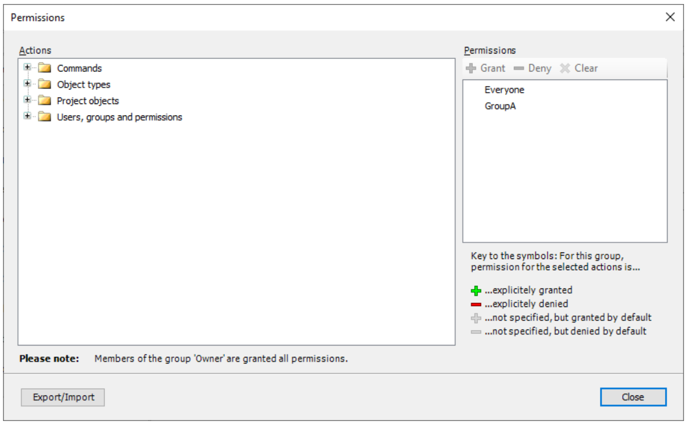
Project/Library: Project: Untitled3

User name: Owner

Password:

4. Click the [Close] button.

The "Permissions" dialog box will be closed.



Key to the symbols: For this group, permission for the selected actions is...

- ✚ ...explicitely granted
- ▬ ...explicitely denied
- ✚ ...not specified, but granted by default
- ▬ ...not specified, but denied by default

Please note: Members of the group 'Owner' are granted all permissions.

Info.

- Settings of operation privileges can be exported in XML format. In the "Permissions" dialog box, click the [Export/Import] button and then select the "Export All Permissions" menu item or "Export selected permissions" menu item". .perms" files can be exported.

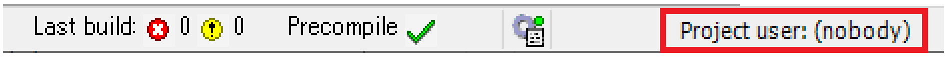
  By selecting the "Import Permission" menu item, you can import ".perms" files.

Performing Operation with Privileges Set

In the following example, an object (POU object: POU_1) is displayed.

Before performing the following procedure, check the status field to see that there is no user who is currently logged on the project.

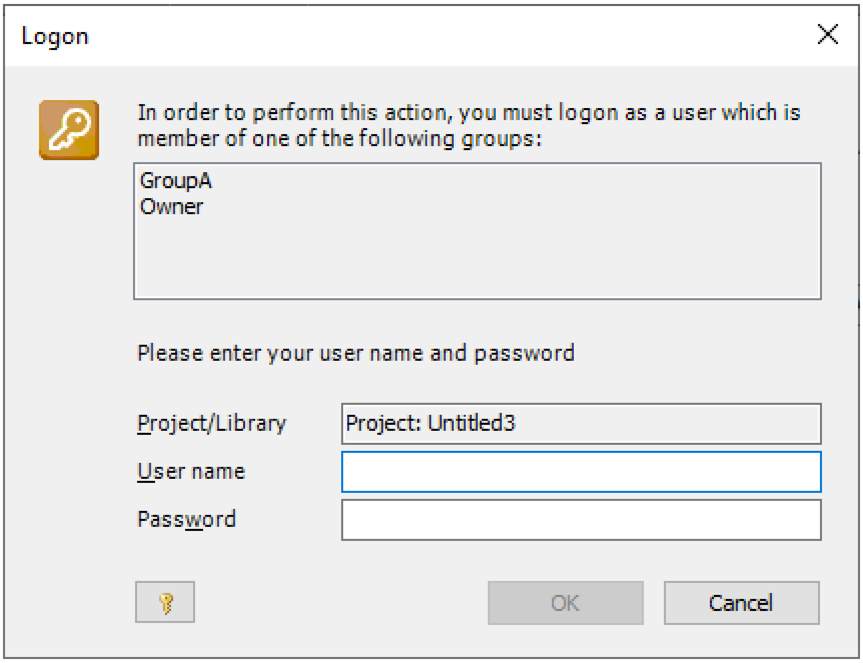
If there are any users who are currently logged on the project, execute logoff by selecting **Project**>**User Management**>**User Logoff**.



Last build: ✖ 0 ⚠ 0 Precompile ✓ Project user: (nobody)

Procedure

1. Double-click the POU_1 object in the navigator pane.
   The "Logon" dialog box will be displayed with object display operations restricted.



Logon

In order to perform this action, you must logon as a user which is member of one of the following groups:

GroupA
Owner

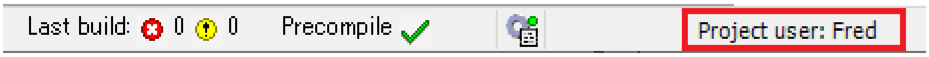
Please enter your user name and password

Project/Library: Project: Untitled3

User name:

Password:

2. Enter appropriate values in the "User name" field and "Password" field, and click the [OK] button.
   Enter the user name and password of the user added in "Creating a New User and Group".
   Logon will be completed and the POU_1 object will be displayed.
   The user name of the user who logged on is displayed on the status field.



Last build: ✖ 0 ⚠ 0 Precompile ✓ Project user: Fred

Device User Management

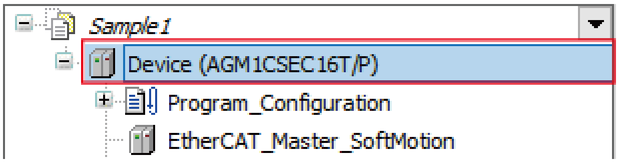
Device user management registers device users and allows only the authorized device users to log in to the device.

A user with user name "Administrator" and password "Administrator" is registered as a device user beforehand.

When you log in as an Administrator user for the first time, you must set any password.

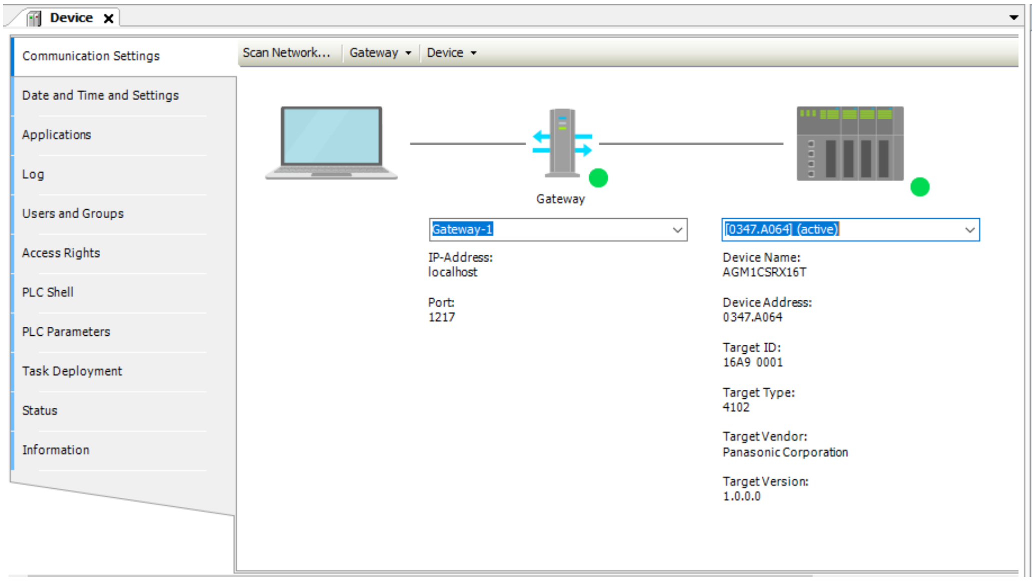
1 2 Procedure

1. Connect the PC where GM Programmer is installed and the GM1 controller.
   For details, refer to "8.5 Connecting to the GM1 Controller".
   Double-click the [Device] object in the navigator pane.



Sample1
Device (AGM1CSEC16T/P)
Program_Configuration
EtherCAT_Master_SoftMotion

The Device setting window will be displayed.



Gateway-1

IP-Address:
localhost

Port:
1217

[0347.A064] (active)

Device Name:
AGM1CSRX16T

Device Address:
0347.A064

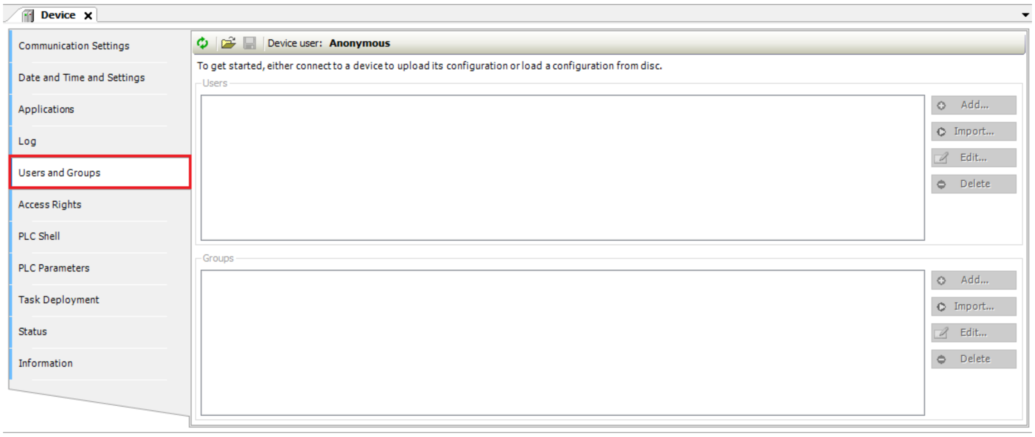
Target ID:
16A9 0001

Target Type:
4102

Target Vendor:
Panasonic Corporation

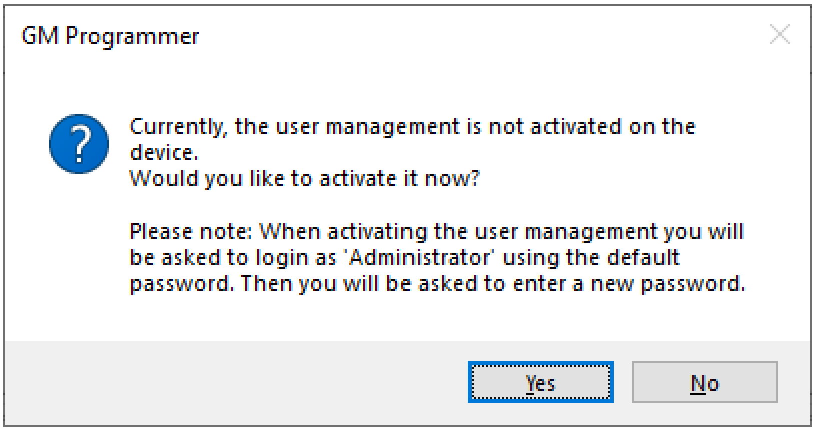
Target Version:
1.0.0.0

2. Click the "User and Group" tab.
   The "User and Group" pane will be displayed.



To get started, either connect to a device to upload its configuration or load a configuration from disc.

3. Click the [♻] icon (Synchronization).
   A confirmation dialog box will be displayed.



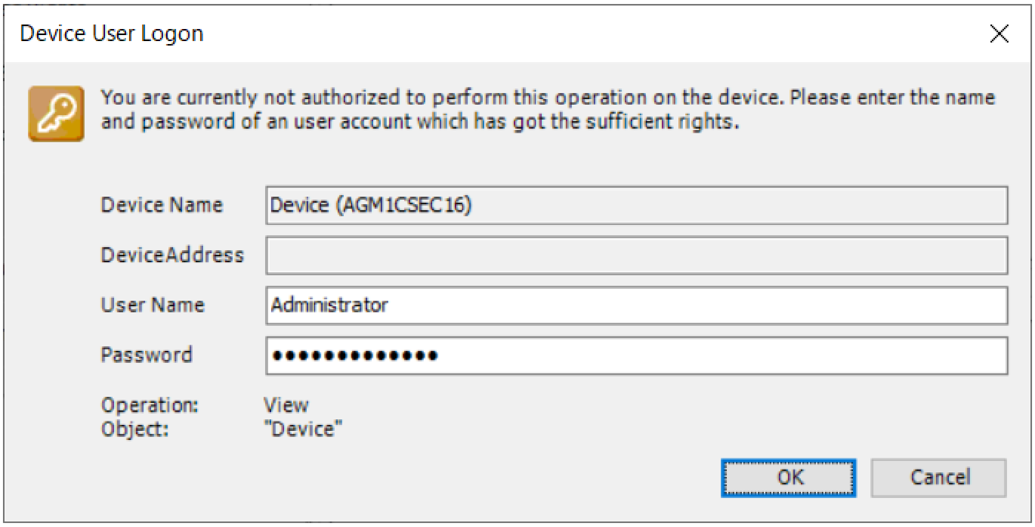
GM Programmer

Currently, the user management is not activated on the device.
Would you like to activate it now?

Please note: When activating the user management you will be asked to login as 'Administrator' using the default password. Then you will be asked to enter a new password.

4. Click the [Yes] button.
   The "Device User Login" dialog box will be displayed.

5. Enter "user name" and "password".
   Enter Administrator in the "User name" field and Administrator in the "Password" field.



Device User Logon

You are currently not authorized to perform this operation on the device. Please enter the name and password of an user account which has got the sufficient rights.

Device Name: Device (AGM1CSEC16)

DeviceAddress:

User Name: Administrator

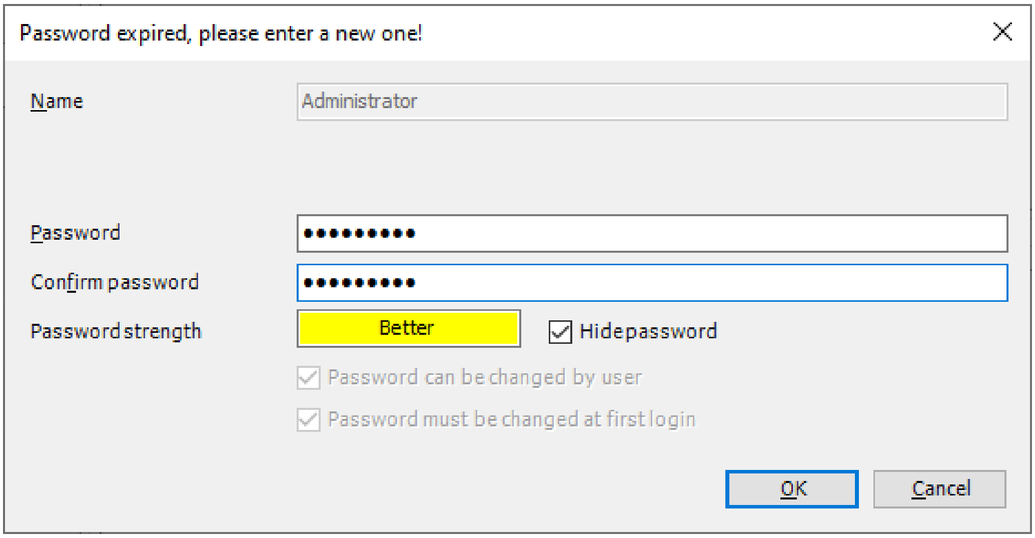
Password: ●●●●●●●●●●●●●

Operation: View

Object: "Device"

6. Click the [OK] button.
   The "Password expired, please enter a new one" dialog box will be displayed.

7. Enter any password.
   To set a password for the Administrator user, enter any password.
   If you forget your password, you cannot log in to the device.



Password expired, please enter a new one!

Name: Administrator

Password: ●●●●●●●●●

Confirm password: ●●●●●●●●●

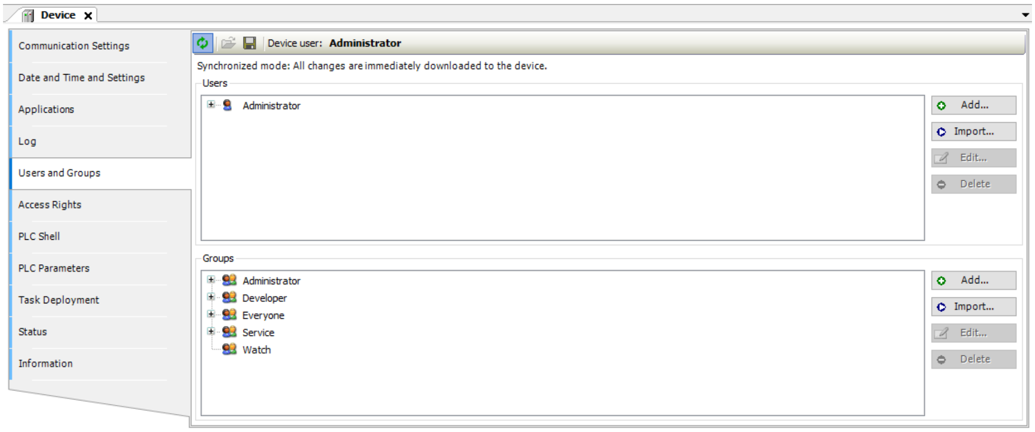
Password strength: Better

☑ Hide password

☑ Password can be changed by user

☑ Password must be changed at first login

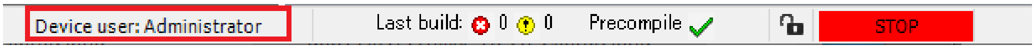
8. Click the [OK] button.
   The password will be set for the Administrator user and you will be logged in as an Administrator user.



Device user: **Administrator**

Synchronized mode: All changes are immediately downloaded to the device.

9. From the menu that is displayed, select **Online**>**Login**.
   You can log in to the device as an Administrator user account.



Device user: Administrator

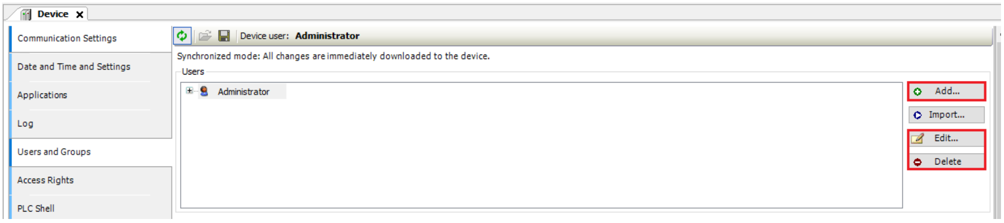
Info.

- To log off logged-in users, from the menu bar, select **Online**>**Security**>**Logoff Current Device User**.
- You can add or remove device users or change their passwords by using the "User and Group" pane.

  To add device users: Add...

  To remove device users: Delete

  To change device user passwords: Edit...



Synchronized mode: All changes are immediately downloaded to the device.

- Users registered by project user management can be imported as device users.

  Import... Clicking the button displays the "Import Users" dialog box. Select a user to be imported and click the [OK] button. In this case, passwords managed by project user management will not be imported. In the "Users and Groups" pane, click the [Edit] button and set a password for the user that has been imported.
- Device user management information can be exported.

  In the "User and Group" pane, click the [💾] icon (Export to Disk).

  XML format files (".dum" files) can be saved.

  To import ".dum" files that have been exported, click the [📂] icon (Import from Disk).
- Device user management information can be initialized by resetting the device.
- If you forget your password, you cannot log in to the GM1 controller. In this case, reset the GM1 controller. For details on how to reset the GM1 controller, refer to the *GM1 Series User's Manual (Hardware Edition)*.

10.2.2 Encryption

This section explains how to encrypt project files.

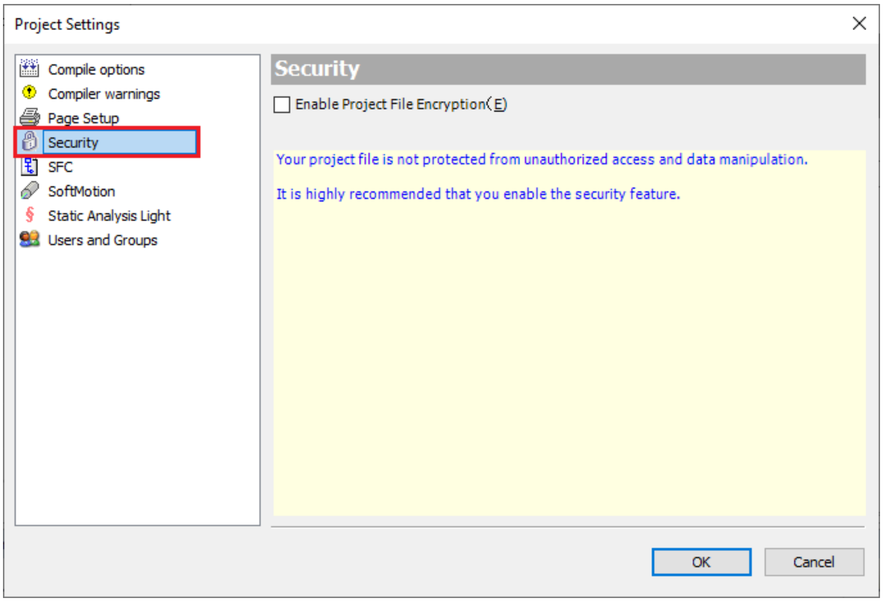
Encrypting Project Files

Project files can be encrypted using passwords. If a password is set, the password must be entered when a project file is opened.

1 2 Procedure

1. From the menu bar, select **Project**>**Project Settings**.
   The "Project Settings" dialog box will be displayed.

2. In the "Project Settings" dialog box, select the "Security" category.
The "Security" pane will be displayed.

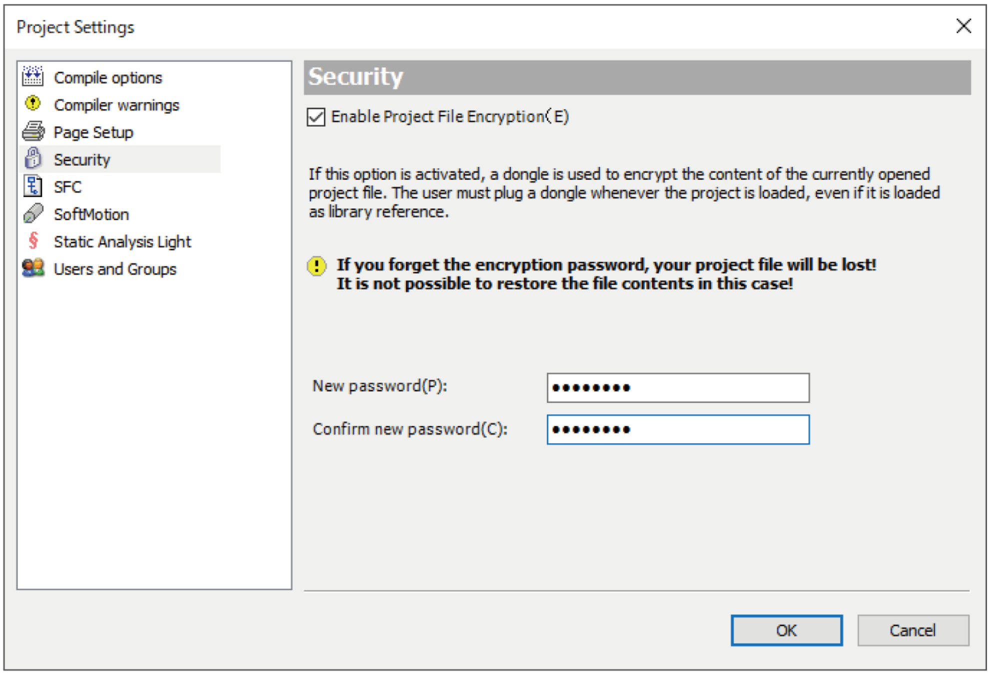


□ Enable Project File Encryption(E)

Your project file is not protected from unauthorized access and data manipulation.

It is highly recommended that you enable the security feature.

3. Select the "Enable Project File Encryption" check box, select the Password option, and then enter a password.



☑ Enable Project File Encryption(E)

If this option is activated, a dongle is used to encrypt the content of the currently opened project file. The user must plug a dongle whenever the project is loaded, even if it is loaded as library reference.

If you forget the encryption password, your project file will be lost!
It is not possible to restore the file contents in this case!

New password(P): ●●●●●●●●

Confirm new password(C): ●●●●●●●●

4. Click the [OK] button.
The specified password will be set for project files.

This completes the password setting procedure.

When an attempt is made to open a project file, a window is displayed, asking the user to enter a password. In this situation, enter the specified password.

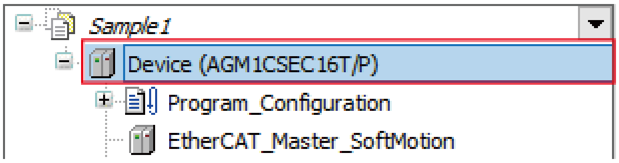
Encrypting the Communication Path: Encrypting Communications Using the Certificate Possessed by the GM1 Controller

Communications between GM Programmer and the GM1 controller can be encrypted using certificates.

This section explains how to encrypt communications by using the certificate possessed by the GM1 controller as a trusted certificate.

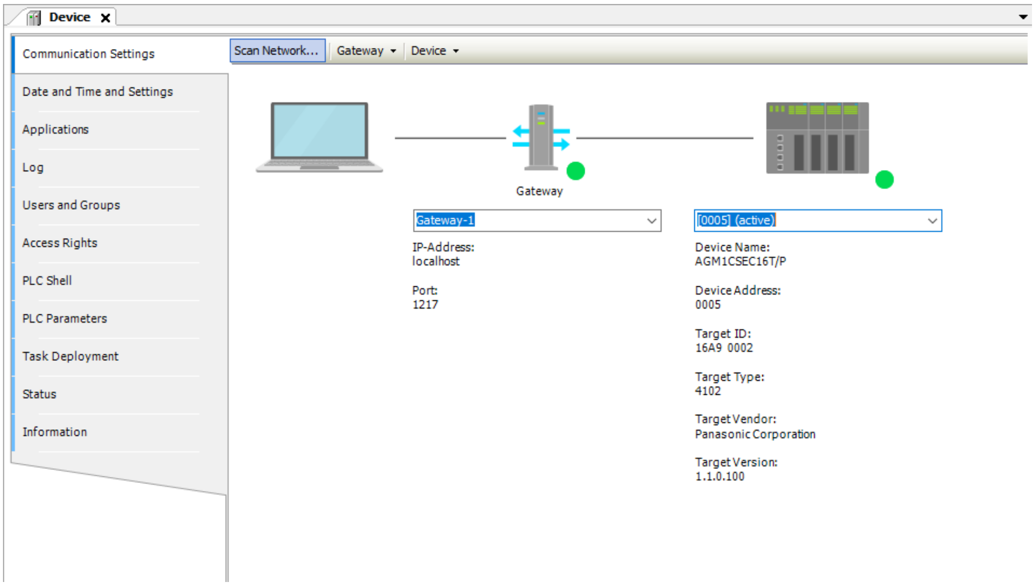
1 2 Procedure

1. Double-click the [Device] object in the navigator pane.



The Device setting window will be displayed.

Open the "Communication Settings" tab.



Gateway-1

IP-Address:
localhost

Port:
1217

[0005] (active)

Device Name:
AGM1CSEC16T/P

Device Address:
0005

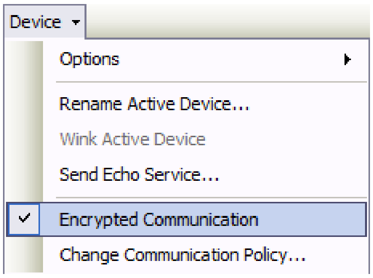
Target ID:
16A9 0002

Target Type:
4102

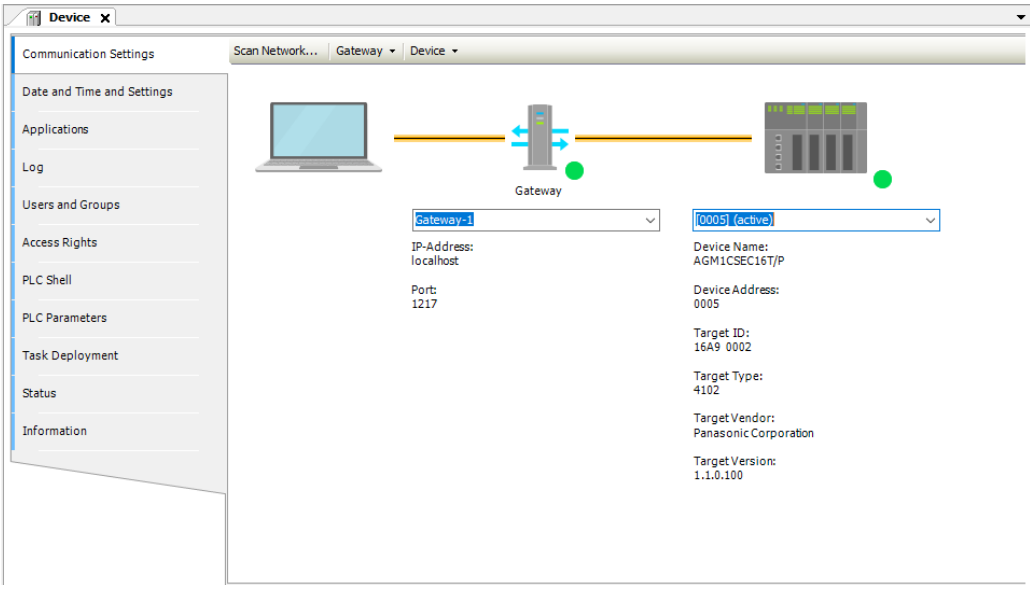
Target Vendor:
Panasonic Corporation

Target Version:
1.1.0.100

2. In the Device menu, select "Encrypted Communication".



If "Encrypted Communication" is selected, the connection lines between the integrated development environment, gateway, and controller will be displayed in yellow.



IP-Address:
localhost

Port:
1217

Device Name:
AGM1CSEC16T/P

Device Address:
0005

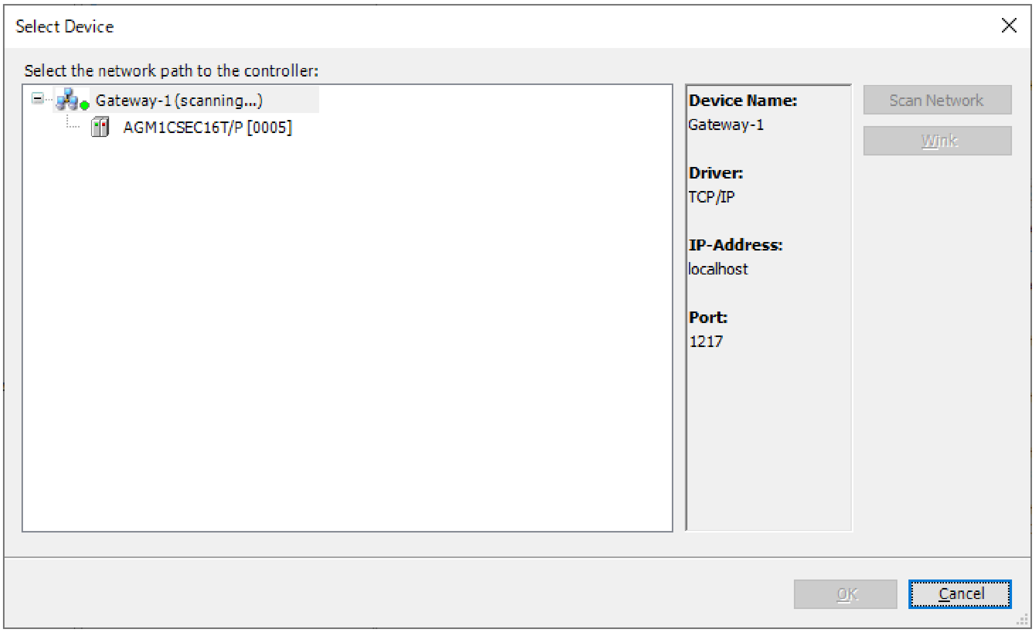
Target ID:
16A9 0002

Target Type:
4102

Target Vendor:
Panasonic Corporation

Target Version:
1.1.0.100

3. Click the Network Scan menu.
The "Select Device" dialog box will be displayed.



Select Device

Select the network path to the controller:

Gateway-1 (scanning...)
AGM1CSEC16T/P [0005]

Device Name: Gateway-1

Driver: TCP/IP

IP-Address: localhost

Port: 1217

4. Select the connected GM1 controller and click the [OK] button.

   A message window will be displayed, indicating that the certificate of the GM1 controller is not certified with a trusted signature for encrypted communication.

5. If the [OK] button is clicked, communications can be encrypted by installing the certificate indicated by the message in local store "Controller Certificates" on the PC to use it as a trusted certificate.

   You can check the registered controller certificate in certmgr.msc in the C:\Windows \System32 folder.

   When the certificate of the GM1 controller is used as a trusted certificate, the validity period of the certificate is 30 days.

Info.

- If the certificate has already expired, the message window shown in step 4 above will be displayed, indicating that the certificate has expired.

  By clicking the [OK] button, you can extend the validity period of the certificate.

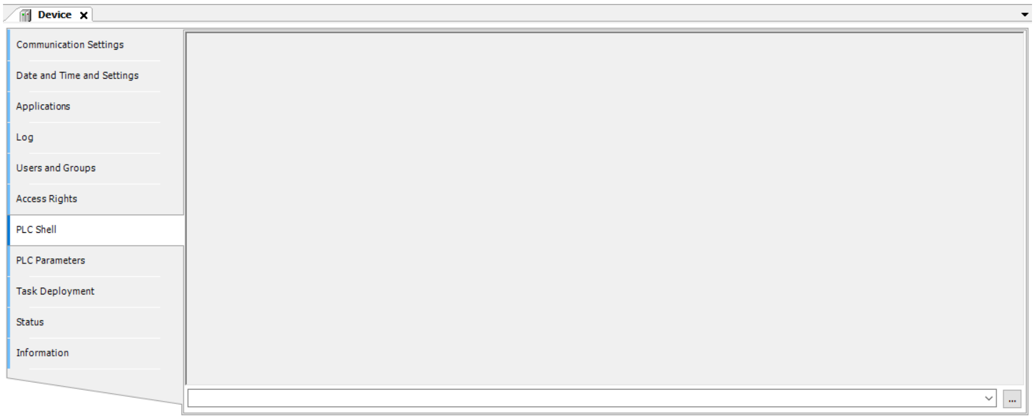
Encrypting the Communication Path: Encrypting Communications Using a Created Certificate

Communications between GM Programmer and the GM1 controller can be encrypted using certificates.

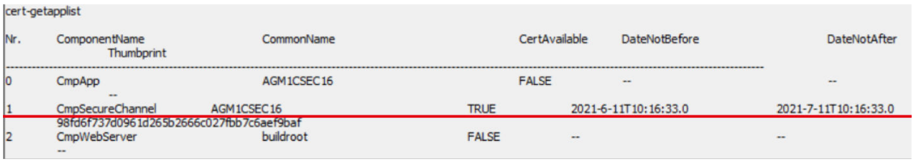
This section explains how to create a trusted certificate for the GM1 controller and encrypt communications using the created certificate.

Procedure

1. Open the device editor and select the "PLC Shell" tab.

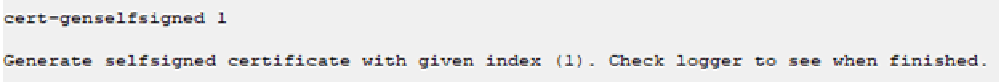


2. Enter the "cert-getapplist" command in the input field.
   All certificates that are used will be displayed.



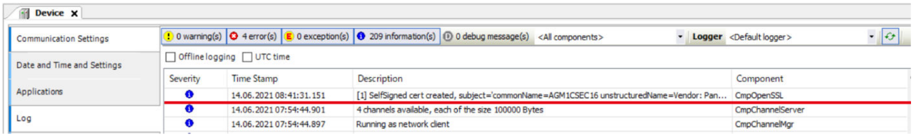
```
cert-getapplist

Nr.     ComponentName                    CommonName                          CertAvailable     DateNotBefore                    DateNotAfter
                 Thumbprint
-----------------------------------------------------------------------------------------------------------------------------
0       CmpApp                           AGM1CSEC16                          FALSE             --                               --
                 --
1       CmpSecureChannel         AGM1CSEC16                         TRUE              2021-6-11T10:16:33.0             2021-7-11T10:16:33.0
        98fd6f737d0961d265b2666c027fbb7c6aef9baf
2       CmpWebServer                     buildroot                  FALSE             --                               --
        --
```

3. Enter the "cert-genselfsigned 1" command in the input field.
   Create a certificate for ComponentName "CmpSecureChannel".



```
cert-genselfsigned 1

Generate selfsigned certificate with given index (1). Check logger to see when finished.
```

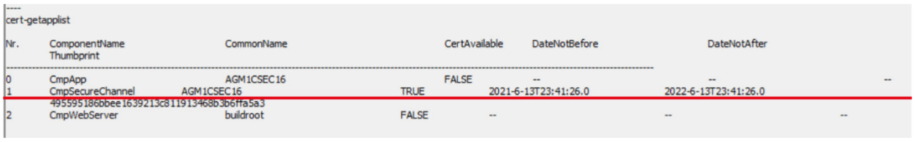
4. Open the "Log" tab of the device editor and click the [Update Information] button.
   Check whether a certificate has been created.



□ Offline logging □ UTC time

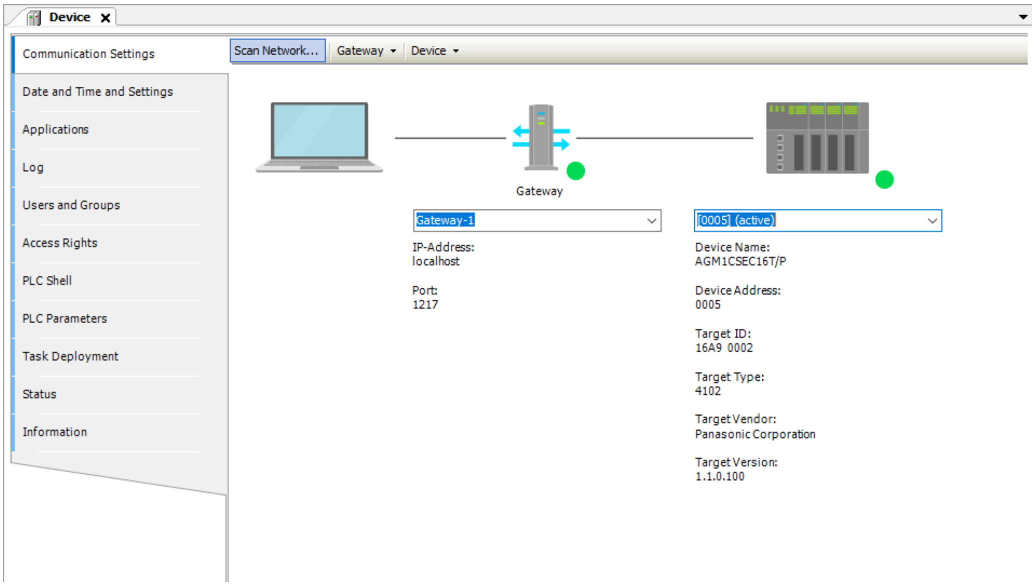
Severity	Time Stamp	Description	Component
●	14.06.2021 08:41:31.151	[1] SelfSigned cert created, subject='commonName=AGM1CSEC16 unstructuredName=Vendor: Pan...	CmpOpenSSL
●	14.06.2021 07:54:44.901	4 channels available, each of the size 100000 Bytes	CmpChannelServer
●	14.06.2021 07:54:44.897	Running as network client	CmpChannelMgr

5. In the device editor, open the "PLC Shell" tab and enter the "cert-getapplist" command in the input field.
   Check whether a certificate has been created for ComponentName "CmpSecureChannel".



```
----
cert-getapplist

Nr.     ComponentName                    CommonName                          CertAvailable     DateNotBefore                    DateNotAfter
        Thumbprint
---------------------------------------------------------------------------------------------------------------
0       CmpApp                           AGM1CSEC16                          FALSE             --                               --              --
1       CmpSecureChannel         AGM1CSEC16                         TRUE              2021-6-13T23:41:26.0             2022-6-13T23:41:26.0
        495595186bbee1639213c811913468b3b6ffa5a3
2       CmpWebServer                     buildroot                  FALSE             --                               --              --
```

6. Open the device editor and select the "Communication Settings" tab.



Gateway-1

IP-Address:
localhost

Port:
1217

[0005] (active)

Device Name:
AGM1CSEC16T/P

Device Address:
0005

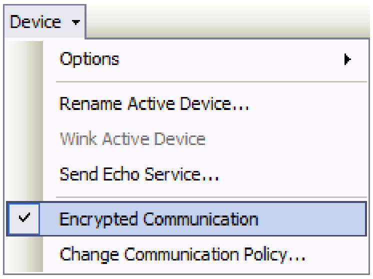
Target ID:
16A9 0002

Target Type:
4102

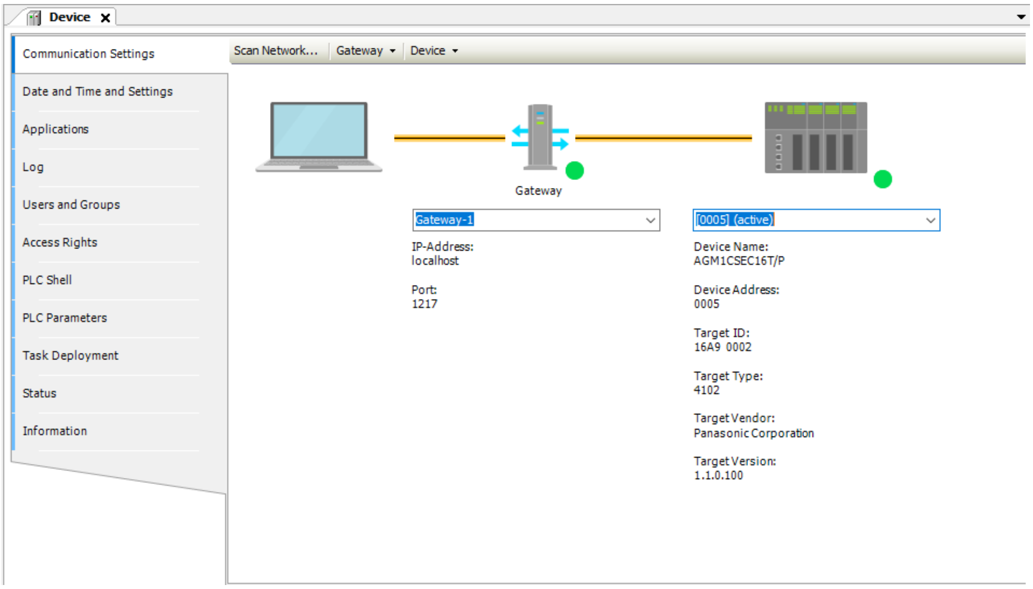
Target Vendor:
Panasonic Corporation

Target Version:
1.1.0.100

7. In the Device menu, select "Encrypted Communication".



If "Encrypted Communication" is selected, the connection lines between the integrated development environment, gateway, and controller will be displayed in yellow.



Gateway-1

IP-Address:
localhost

Port:
1217

[0005] (active)

Device Name:
AGM1CSEC16T/P

Device Address:
0005

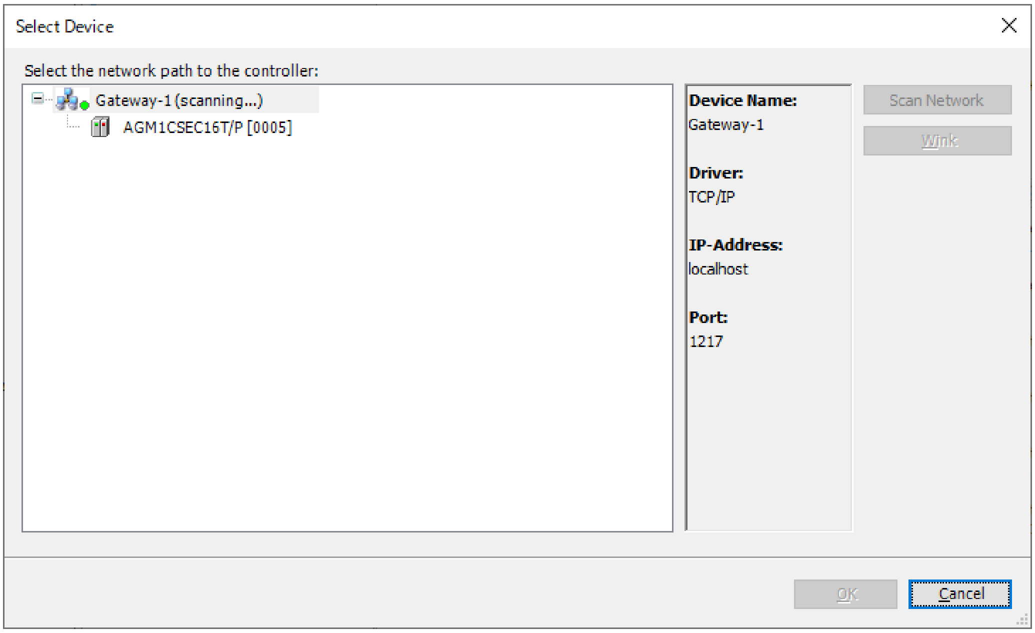
Target ID:
16A9 0002

Target Type:
4102

Target Vendor:
Panasonic Corporation

Target Version:
1.1.0.100

8. Click the Network Scan menu.
   The "Select Device" dialog box will be displayed.



Select the network path to the controller:

Gateway-1 (scanning...)
AGM1CSEC16T/P [0005]

Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

9. Select the connected GM1 controller and click the [OK] button.
   A message window will be displayed, indicating that the certificate of the GM1 controller is not certified with a trusted signature for encrypted communication.

10. If the [OK] button is clicked, communications can be encrypted by installing the certificate indicated by the message in local store "Controller Certificates" on the PC to use it as a trusted certificate.

You can check the registered controller certificate in certmgr.msc in the C:\Windows \System32 folder.

When the created certificate is used as a trusted certificate, the validity period of the certificate is 360 days.

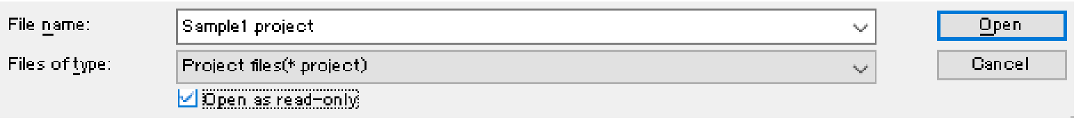
10.2.3 Write-protection

This section explains how to implement write-protection for project files to prevent project files from being modified unintentionally by mistake.

Opening Files in Read-only Mode

Open a project file in read-only mode.

When selecting a project file to be opened, select the "Open in Read-only Mode" check box.



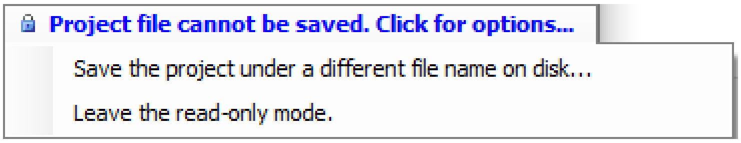
File name: Sample1 project

Files of type: Project files(* project)

☑ Open as read-only

If a file is opened in read-only mode, it cannot be saved.

To save a project file, select "Project file cannot be saved. Click for options" on the menu bar and select an appropriate menu item that is displayed.



Project file cannot be saved. Click for options...

Save the project under a different file name on disk...

Leave the read-only mode.

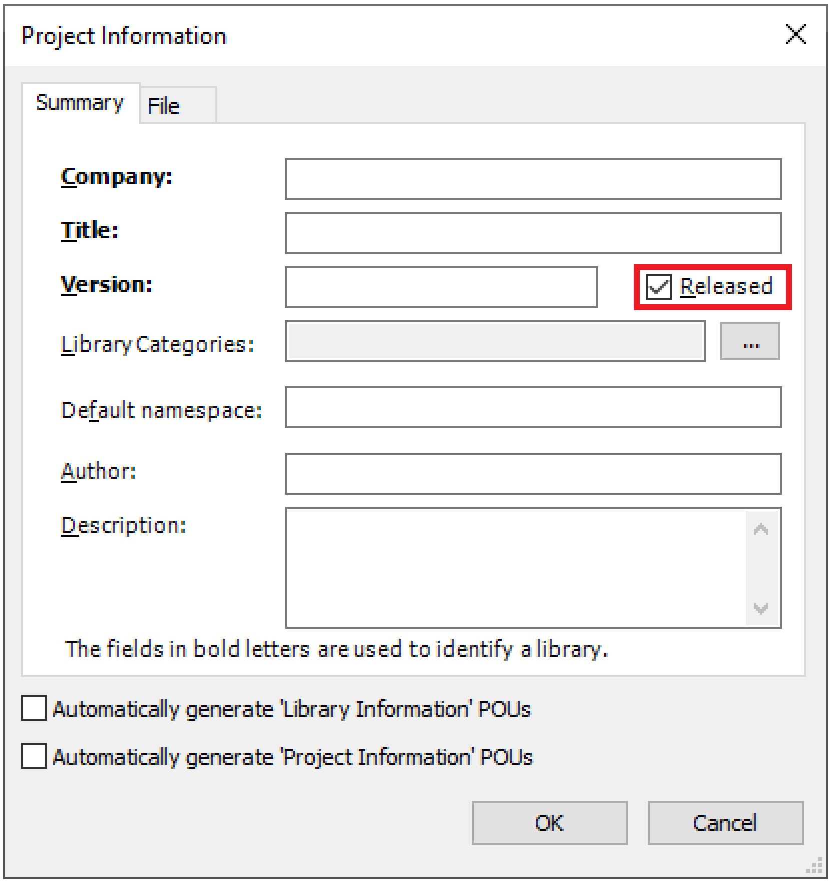
Item	Description
Save the project under a different file name on disk...	Allows the user to rename and save the project file as a writable file.
Leave the read-only mode	Leaves the project file open in read-only mode.

Setting the "Released" Flag

Set a "released" flag in project information in a project file.

If a "released" flag is set in a project file, changes made in the file cannot be saved.

From the menu bar, select **Project>Project Information**, and open the "Summary" tab window and then select the "Released" check box.



Project Information

Summary | File

Company:

Title:

Version:

☑ Released

Library Categories:

Default namespace:

Author:

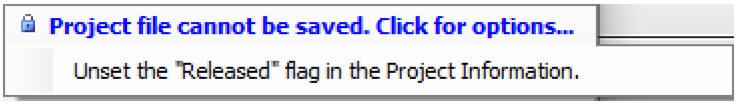
Description:

The fields in bold letters are used to identify a library.

□ Automatically generate 'Library Information' POUs

□ Automatically generate 'Project Information' POUs

To save a project file in which the "Released" flag is set, select "Project file cannot be saved. Click for options" on the menu bar and select a menu item that is displayed.



Project file cannot be saved. Click for options...

Unset the "Released" flag in the Project Information.

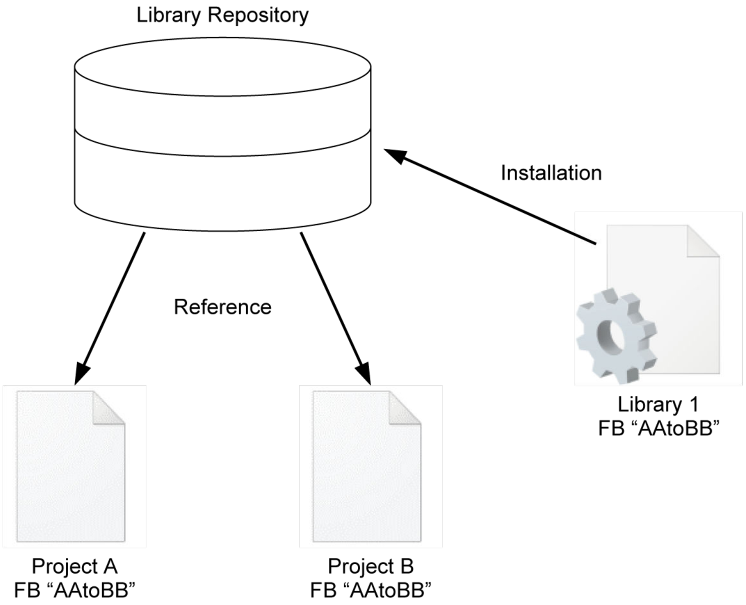
10.3 User Library Function

Combining created functions and function blocks as a library enables other projects to use these functions and function blocks.

The library that has been created must be installed in the library repository. Adding the library installed in the library repository to the project makes it possible to use the functions and function blocks in the library. Libraries in the project are managed by the "Library_Manager" object.

The following sections describe a procedure for creating a library, installing the created library into the library repository, and adding libraries to a project.

Example

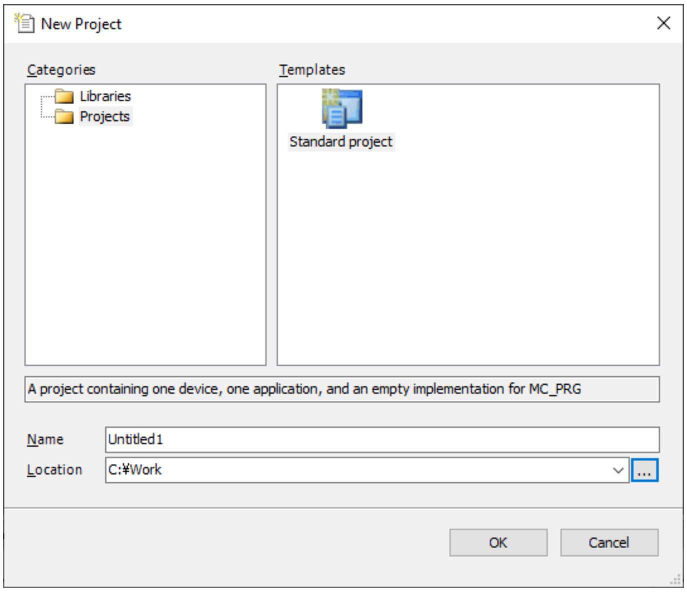


10.3.1 Creating a Library and Adding to the Library Repository

This section explains the entire procedure from creating a project for libraries through to installing libraries in the library repository.

1 2 Procedure

1. From the menu bar, select**File**>**New Project**.
   The"New Project"dialog box will be displayed.



New Project

Categories: Libraries, Projects

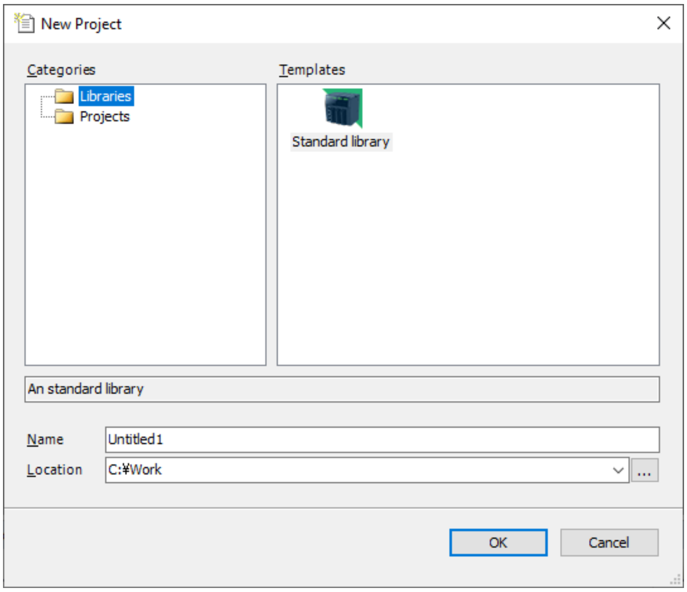
Templates: Standard project

A project containing one device, one application, and an empty implementation for MC_PRG

Name: Untitled1

Location: C:¥Work

2. Select"Libraries"in the Categories pane and then"Standard library"in the Template pane.



New Project

Categories: Libraries, Projects

Templates: Standard library

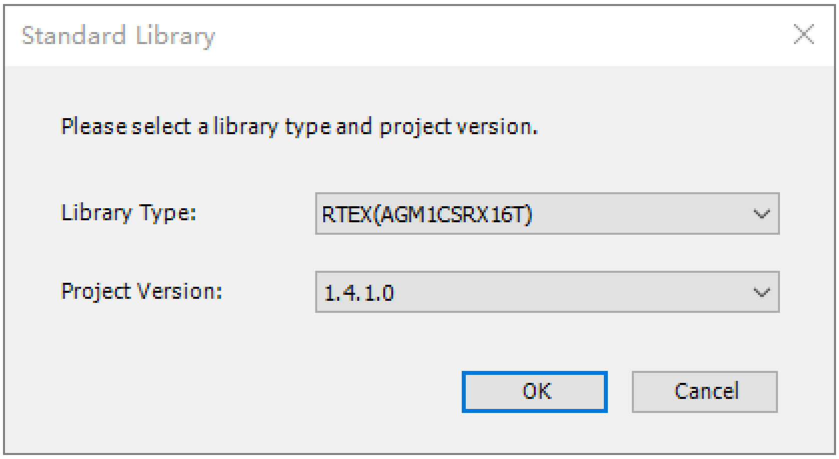
An standard library

Name: Untitled1

Location: C:¥Work

3. Click the [OK] button.

The “Standard library” dialog box will be displayed.



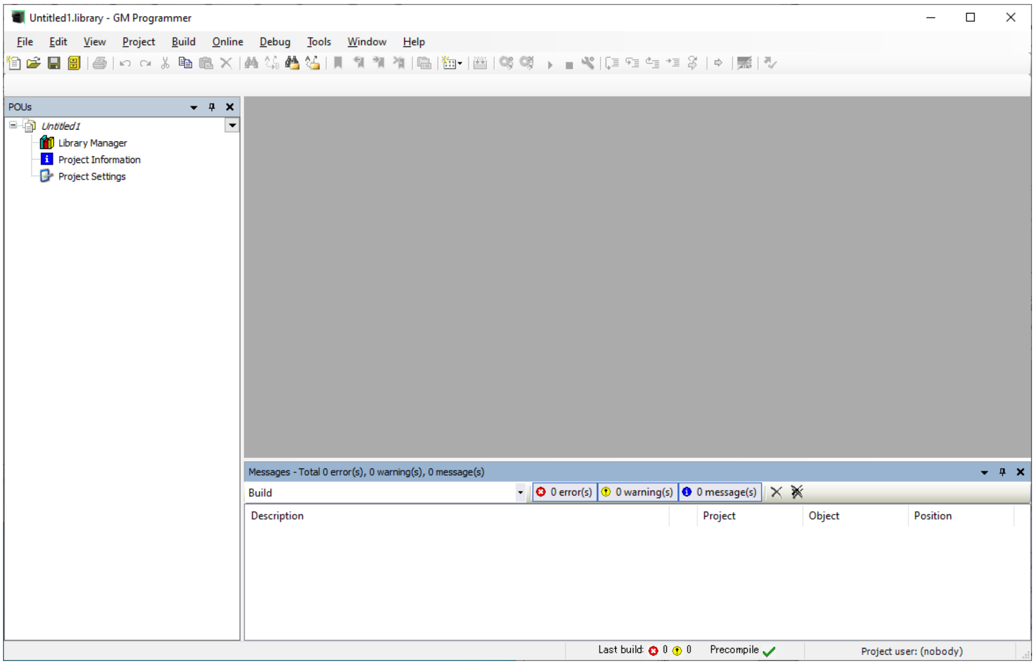
Please select a library type and project version.

Library Type: RTEX(AGM1CSRX16T)

Project Version: 1.4.1.0

4. Click the [OK] button.

A project for libraries will be created. The extension of project files for libraries is "library". For libraries, the POU view is displayed in the navigator pane. Add objects required for creating libraries to the POU view.



Messages - Total 0 error(s), 0 warning(s), 0 message(s)

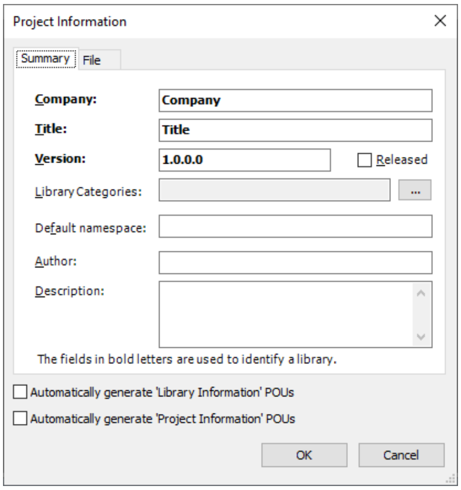
Last build: 0 0 Precompile ✓ Project user: (nobody)

5. Double-click the"Project Information"object in the POU view.

The"Project Information"dialog box will be displayed.

If necessary, change the settings in the"Company","Title", and"Version"fields. The information set in these fields will be displayed in the selection window when created libraries are added to the project.

If the"Release"check box is selected, a confirmation message will be displayed when an attempt is made to change a library.



Project Information

Summary | File

Company: Company

Title: Title

Version: 1.0.0.0 □ Released

Library Categories:

Default namespace:

Author:

Description:

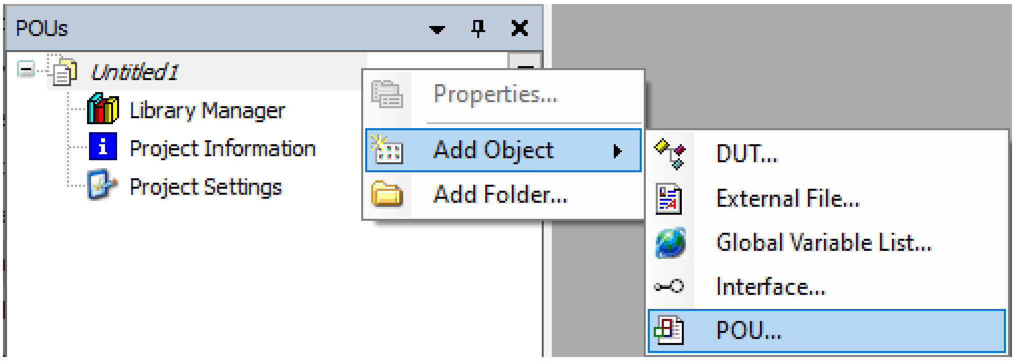
The fields in bold letters are used to identify a library.

□ Automatically generate 'Library Information' POUs

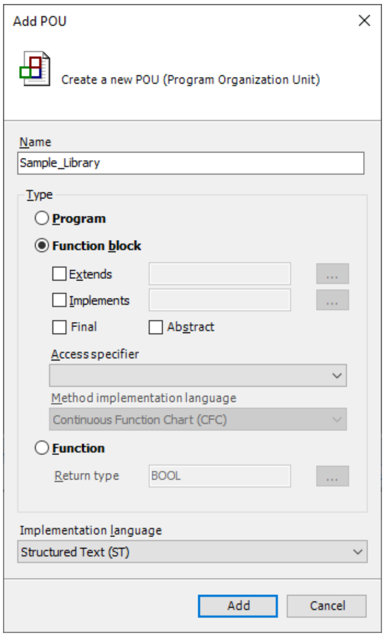
□ Automatically generate 'Project Information' POUs

6. Click the[OK]button.
   The project information will be set.

7. Right-click the <file name> object at the top of the navigator pane and then select**Add Object**>**POU**from the context-sensitive menu that is displayed.
   The"Add POU"dialog box will be displayed.



8. Select the"Function block"check box, enter a name in the Name field, and select a programming language from the Implementation Language drop-down list.
   For details on functions blocks, refer to"6.5 Function and Function Block".



Add POU

Create a new POU (Program Organization Unit)

Name: Sample_Library

Type

○ Program

● Function block

☐ Extends

☐ Implements

☐ Final ☐ Abstract

Access specifier

Method implementation language: Continuous Function Chart (CFC)

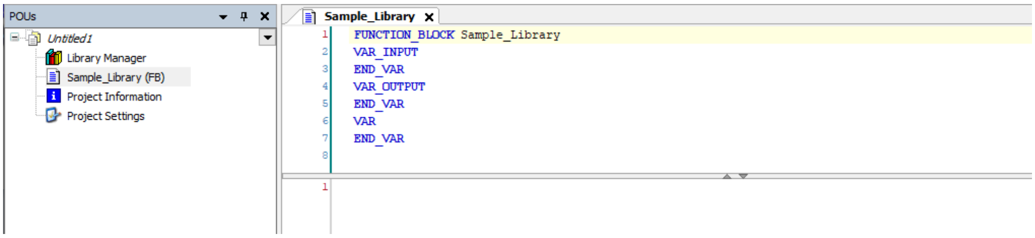
○ Function

Return type: BOOL

Implementation language: Structured Text (ST)

9. Click the[Add]button.

An object of the function block will be added to the POU view in the navigator pane.



POUs

- Untitled1
  - Library Manager
  - Sample_Library (FB)
  - Project Information
  - Project Settings

Sample_Library

```
FUNCTION_BLOCK Sample_Library
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
```

10. Enter a program in the function block.

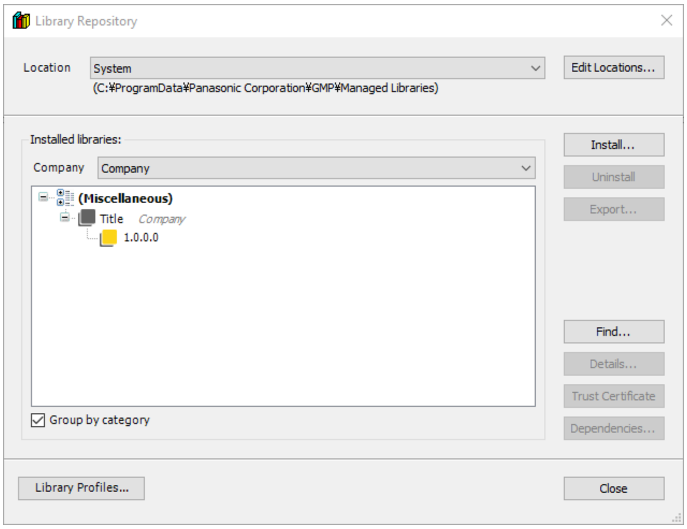
After entering a program, from the menu bar, execute**Build**>**Check All Pooled Objects**. Build will be executed to perform a syntax check. After the above command is executed, if any error is displayed, correct the program and execute build again.

11. From the menu bar, select**File**>**Save Project and Install into Library Repository**.

The library that has been created will be installed in the library repository.

12. From the menu bar, select**Tools**>**Library Repository**.

The"Library Repository"dialog box will be displayed. Check that the created library is displayed in the"Installed library"section.



Location: System
(C:¥ProgramData¥Panasonic Corporation¥GMP¥Managed Libraries)

Edit Locations...

Installed libraries:

Company: Company

- (Miscellaneous)
  - Title *Company*
    - 1.0.0.0

☑ Group by category

Clicking the[Detail]button enables the user to check information such as function blocks included in the library.

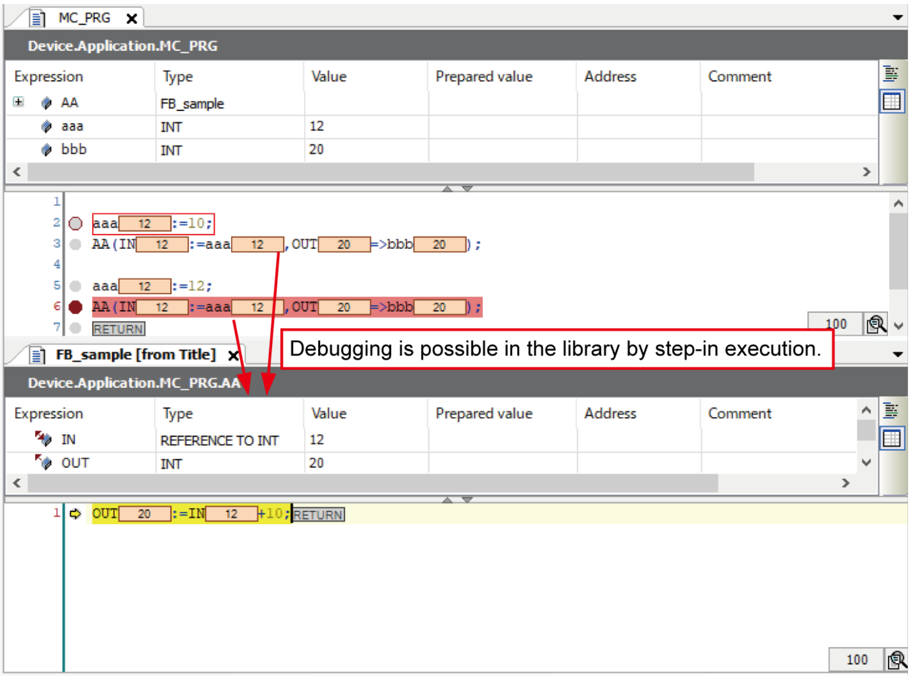
This completes the procedure for installing the library in the library repository.

Info.

- Difference between libraries (.library) and complied libraries (.compiled-library).

  If a library is installed as a library file into the repository, the user can refer to codes by step-in execution during debugging and check execution details.

  The user cannot execute codes in any compiled library while referring to the codes.



MC_PRG

Device.Application.MC_PRG

Expression	Type	Value	Prepared value	Address	Comment
AA	FB_sample				
aaa	INT	12			
bbb	INT	20			

```
aaa 12 :=10;
AA(IN 12 :=aaa 12 ,OUT 20 =>bbb 20 );

aaa 12 :=12;
AA(IN 12 :=aaa 12 ,OUT 20 =>bbb 20 );
RETURN
```

Debugging is possible in the library by step-in execution.

FB_sample [from Title]

Device.Application.MC_PRG.AA

Expression	Type	Value	Prepared value	Address	Comment
IN	REFERENCE TO INT	12			
OUT	INT	20			

```
OUT 20 :=IN 12 +10;RETURN
```

- Method for installing a compiled library

  In Step 10, select**File**>**Save Project as Complied Library**from the menu. The library is saved as a complied library file (.compiled-library).

  Then, click the[Install]button in the"Library Repository"dialog box and select the saved complied library file.

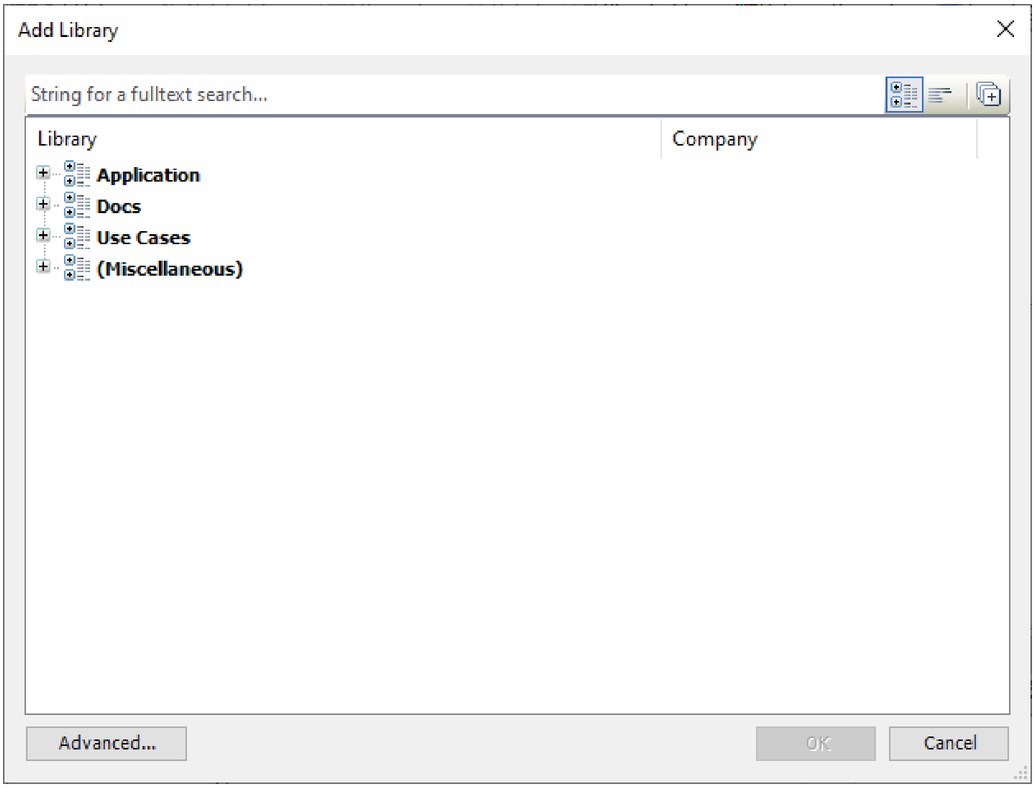
10.3.2 Using Created Libraries

This section explains how to add libraries installed in the library repository to the project.

1 2 Procedure

1. Open the project file and select the "Library Manager" object.
The "Library Manager" window will be displayed.

2. Click the [Add Library] button.
The "Add Library" dialog box will be displayed, showing the libraries added to the library repository.

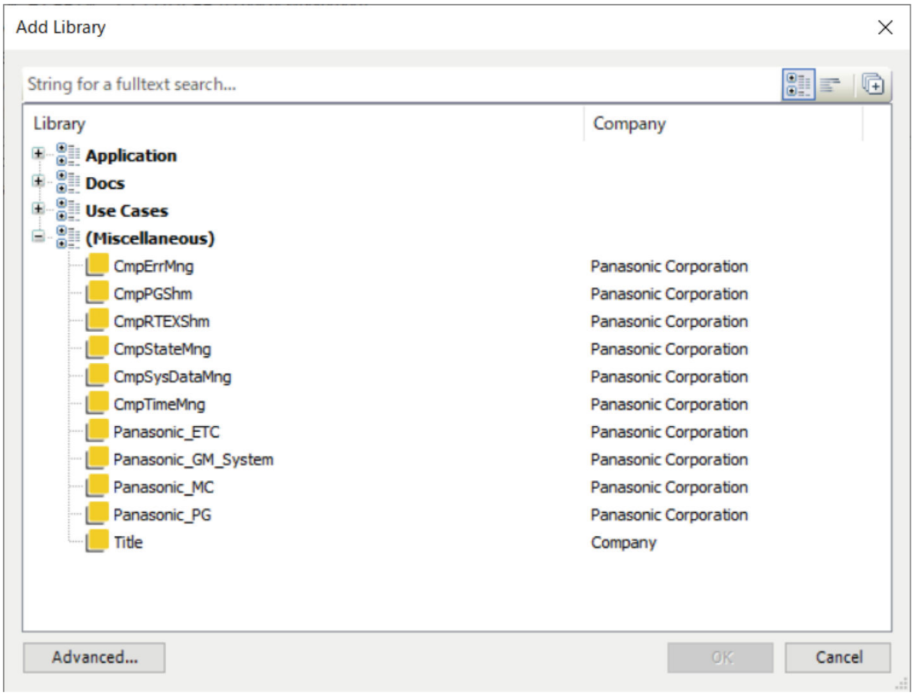


Add Library

String for a fulltext search...

Library	Company
⊞ **Application**	
⊞ **Docs**	
⊞ **Use Cases**	
⊞ **(Miscellaneous)**	

3. Select a created library and click the [OK] button.
The selected library will be added to the application in the project.
The title and company name specified when the library was created will be displayed.



Add Library

String for a fulltext search...

Library	Company
Application	
Docs	
Use Cases	
(Miscellaneous)	
CmpErrMng	Panasonic Corporation
CmpPGShm	Panasonic Corporation
CmpRTEXShm	Panasonic Corporation
CmpStateMng	Panasonic Corporation
CmpSysDataMng	Panasonic Corporation
CmpTimeMng	Panasonic Corporation
Panasonic_ETC	Panasonic Corporation
Panasonic_GM_System	Panasonic Corporation
Panasonic_MC	Panasonic Corporation
Panasonic_PG	Panasonic Corporation
Title	Company

This completes the procedure for adding the library to the application.

The function blocks in the added library can be used in the program.

10.4 POU for implicit checks

POU for implicit checks is an object with functions that check the range of array indexes or subrange type variables, validity of pointer addresses, and division by zero. Adding this object to the project makes it possible to automatically call these functions and perform checks, without calling the functions explicitly within the program.

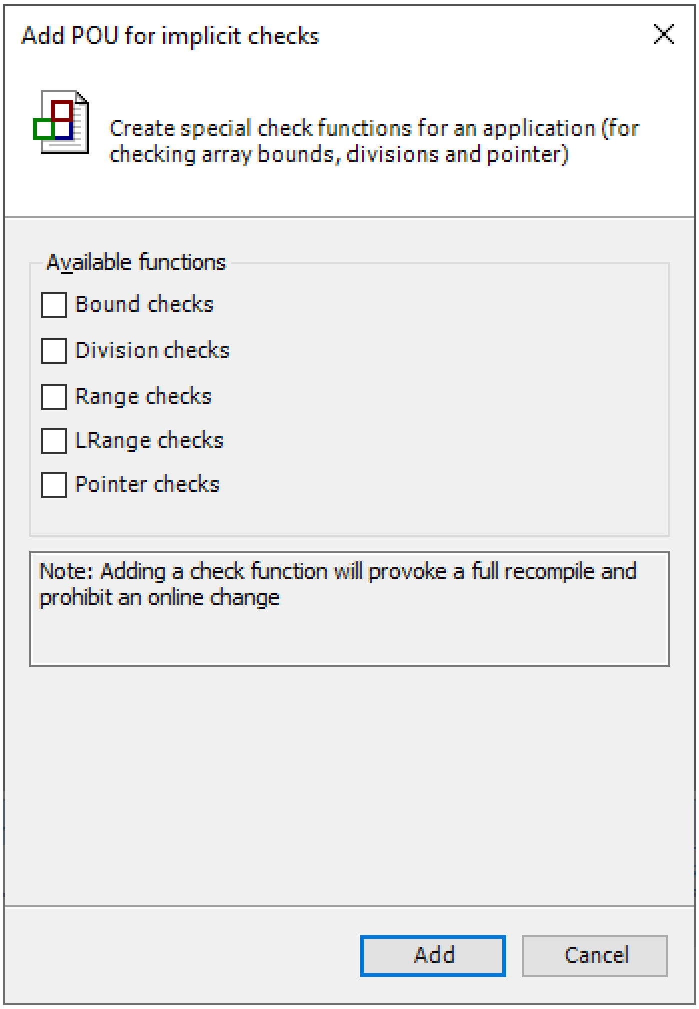
10.4.1 Setting up POU for implicit checks

This section explains how to add POU for implicit checks objects.

1 2 Procedure

1. Right-click the[Application]object in the navigator pane and then select **Add Object**>**POU for implicit checks** from the context-sensitive menu that is displayed.

   The"Add POU for implicit checks"dialog box will be displayed.



Add POU for implicit checks

Create special check functions for an application (for checking array bounds, divisions and pointer)

Available functions

□ Bound checks

□ Division checks

□ Range checks

□ LRange checks

□ Pointer checks

Note: Adding a check function will provoke a full recompile and prohibit an online change

2. Select the check box of a function that you want to add.
   Multiple functions (objects) will be added, depending on the item that you select.

Check type	Function name (object name)
Bound checks	CheckBounds
Division checks	CheckDivDInt
	CheckDivLInt
	CheckDivLReal
	CheckDivReal
Range checks	CheckRangeSigned
	CheckRangeUnsigned
LRange checks	CheckLRangeSigned
	CheckLRangeUnsigned
Pointer checks	CheckPointer

3. Click the[Add]button.
   The function (object) for the selected item will be added.
   If necessary, edit the implementation section of the object that has been added.

The check details of each check item are as below.

Bound Checks

Checks whether the bound is violated.

Division Checks

Checks whether anything is divided by zero.

Range Checks

Checks whether values of DINT or UDINT subrange type variables are within the specified range.

LRange Checks

Checks whether values of LINT or ULINT subrange type variables are within the specified range.

Pointer Checks

Checks whether the returned pointer refers to a valid memory address or whether the contents of a memory address that is referred to match the variable type that refers to the pointer.

For Pointer Checks, a program must be created for the function.

10.5 Interface Function

The interface object defines common methods and properties that are used between different function blocks in the same way.

The interface object is one of the means of implementing object-oriented programming.

The interface object contains only method and property declarations but does not contain implementation.

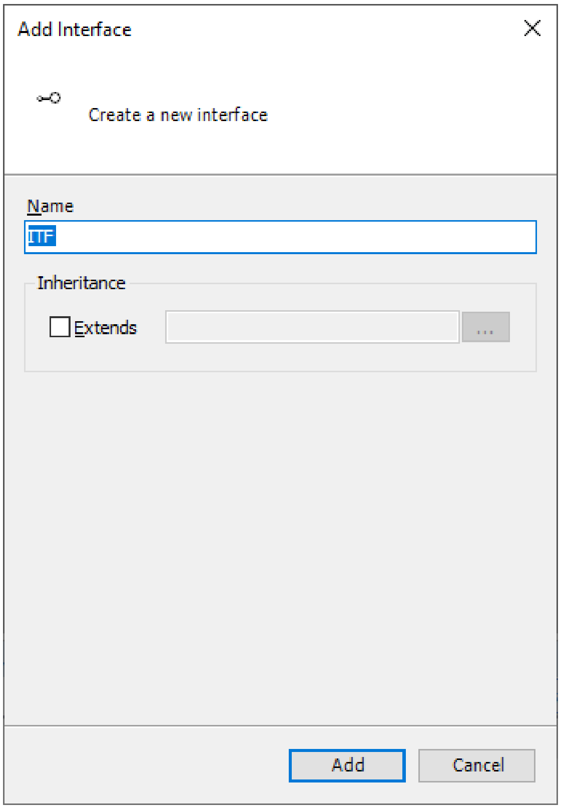
10.5.1 Setting up an Interface Object

This section explains how to add an interface object.

1 2 Procedure

1. Right-click the "Application" object in the navigator pane and then select **Add Object**>**Interface** from the context-sensitive menu that is displayed.

   The "Add Interface" dialog box will be displayed.



Add Interface

Create a new interface

Name: ITF

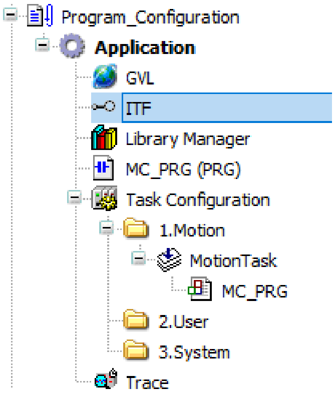
Inheritance

☐ Extends

2. Enter a name and click the [Add] button.

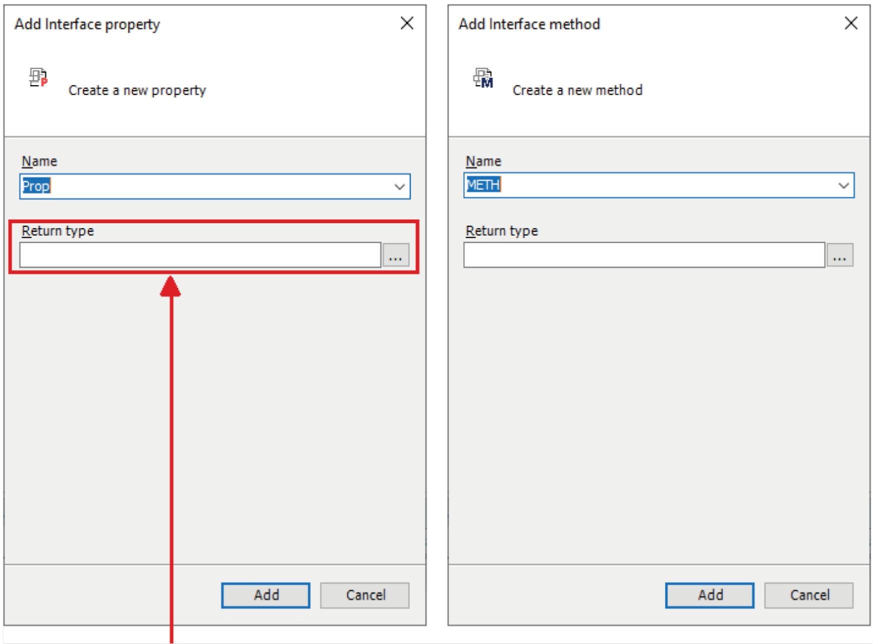
   An interface object will be added.

   If the "Extends" check box is selected, the interface entered in the input field can be inherited and extended.



3. Right-click the "ITF" object added to the navigator pane and then select **Add Object**>**Interface property** or **method** from the context-sensitive menu that is displayed.

The "Add Interface property" dialog box or "Add Interface method" dialog box will be displayed.


Add Interface property

Create a new property

Name: Prop

Return type:

Add Interface method

Create a new method

Name: METH

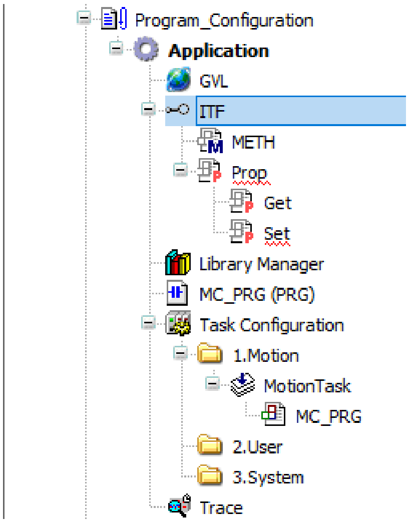
Return type:

* For interface properties, the data type of return values must be entered.

4. Enter a name and click the [Add] button.

For interface properties, be sure to enter a value in the "Return type" field.

"Prop" or "METH" object will be added under the "ITF" object.



5. The added "METH" and "Prop" objects are used to define methods and properties, respectively.
   This completes the procedure for creating an interface object.

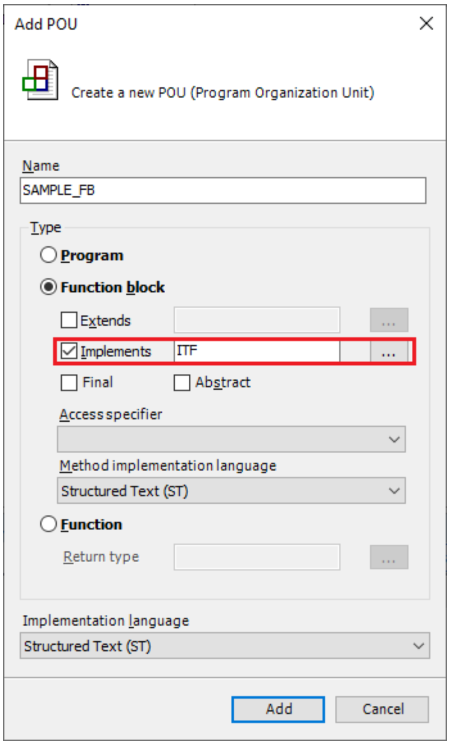
10.5.2 Implementing in New Function Block

This section explains how to implement an interface in a new function block to be created.

Procedure

1. Right-click the "Application" object in the navigator pane and then select **Add Object**>**POU** from the context-sensitive menu that is displayed.

   The "Add POU" dialog box will be displayed.



Add POU

Create a new POU (Program Organization Unit)

Name: SAMPLE_FB

Type

○ Program

● Function block

□ Extends

☑ Implements: ITF

□ Final □ Abstract

Access specifier:

Method implementation language: Structured Text (ST)

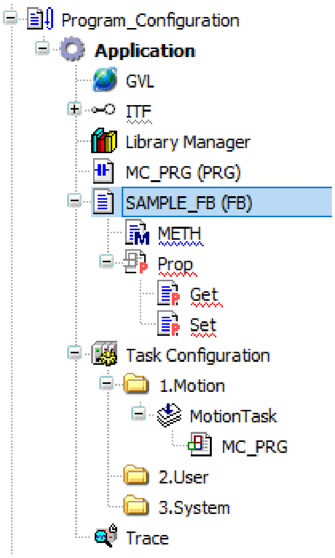
○ Function

Return type:

Implementation language: Structured Text (ST)

2. Enter a name. In the Type section, select the Function block option, select the "Implements" check box, and enter an interface to be implemented. Click the [Add] button.

   A function block with the interface implemented will be added under the "Application" object.



3. Open the respective editors for the added "METH" and "Prop" objects, and implement internal processing for the methods and properties.

Note

- If methods and properties are added under the "interface" object later, they will not be automatically added to the function block with the interface implemented. Therefore, if they need to be added, perform the procedure starting from "Step 3" 3 in "10.5.3 Implementing in Existing Function Block".

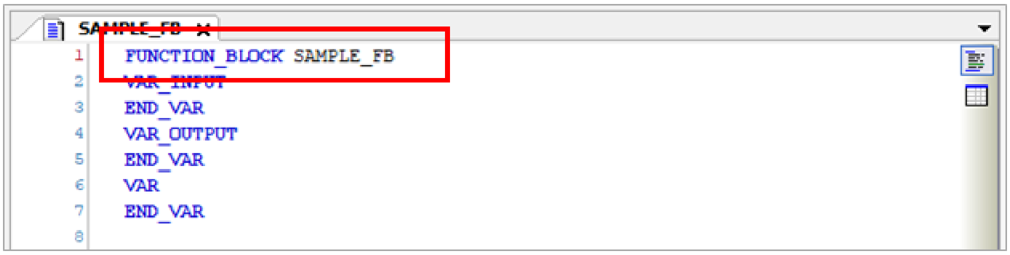
10.5.3 Implementing in Existing Function Block

This section explains how to implement an interface in an existing function block.

1 2 Procedure

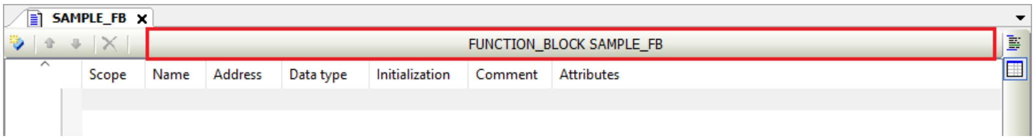
1. Open the editor of the existing function block from the navigator pane.

Character string format



```
FUNCTION_BLOCK SAMPLE_FB
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
```

Table format

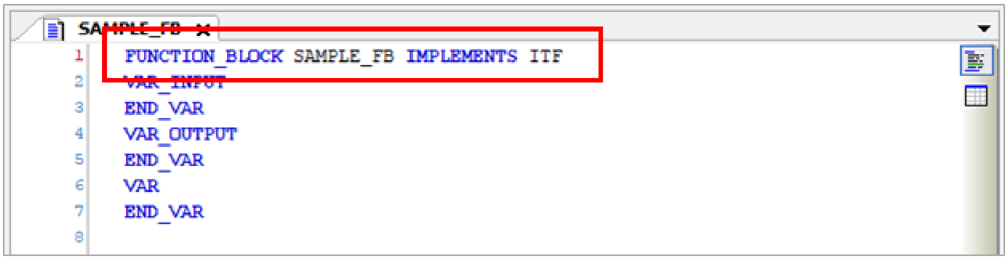


FUNCTION_BLOCK SAMPLE_FB

Scope	Name	Address	Data type	Initialization	Comment	Attributes

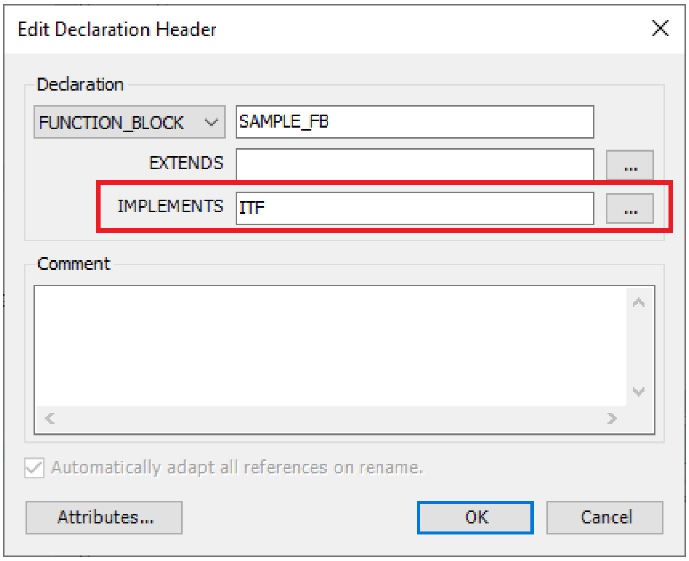
2. For character string format, add "IMPLEMENTS <interface name>" to the declaration header section. For table format, open the "Edit Declaration Header" dialog box, enter an interface name in the "IMPLEMENTS" field, and click the [OK] button.

Character string format



```
FUNCTION_BLOCK SAMPLE_FB IMPLEMENTS ITF
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
```

Table format


Edit Declaration Header

Declaration

FUNCTION_BLOCK SAMPLE_FB

EXTENDS

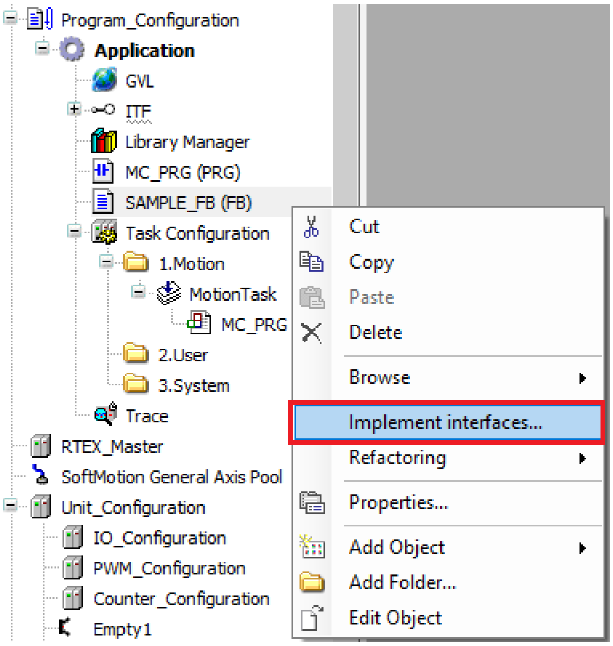
IMPLEMENTS ITF

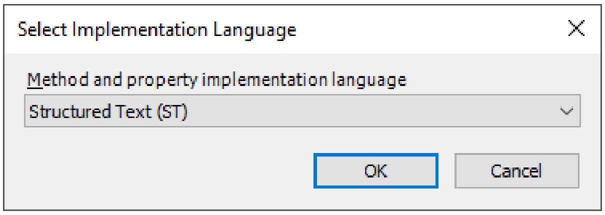
Comment

☑ Automatically adapt all references on rename.

3. Right-click the existing function block in the navigator pane and select "Implement Interfaces" from the context-sensitive menu that is displayed.

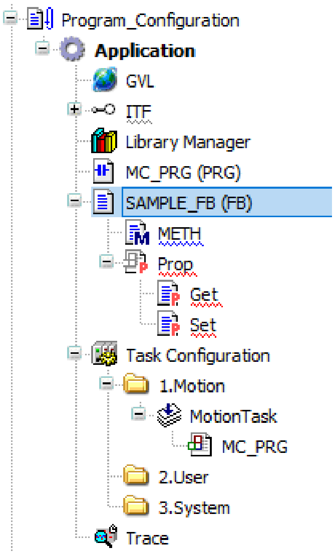
   The "Select implementation language" dialog box will be displayed.




Select Implementation Language

Method and property implementation language: Structured Text (ST)

4. Select a desired programming language and click the [OK] button. The method and property objects will be added under the function block object.



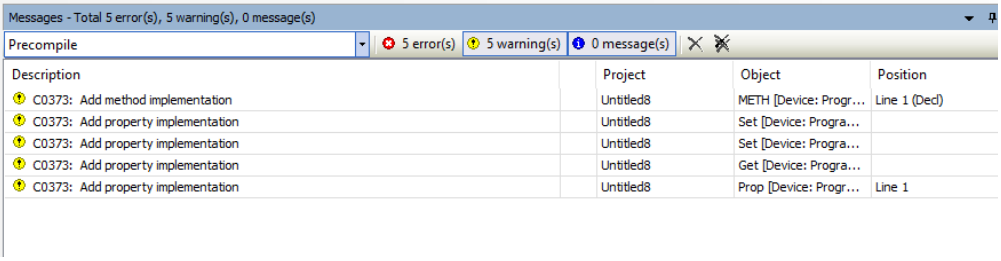
5. Open the respective editors for the added "METH" and "Prop" objects, and implement internal processing for the methods and properties.

Note

- If methods and properties are added under the "interface" object later, they will not be automatically added to the function block with the interface implemented. Therefore, if they need to be added, perform the procedure starting from Step "Step 3".

Info.

- When methods and properties are implemented, warning messages are automatically implemented. Therefore, when compilation is executed, the following warning messages are displayed.

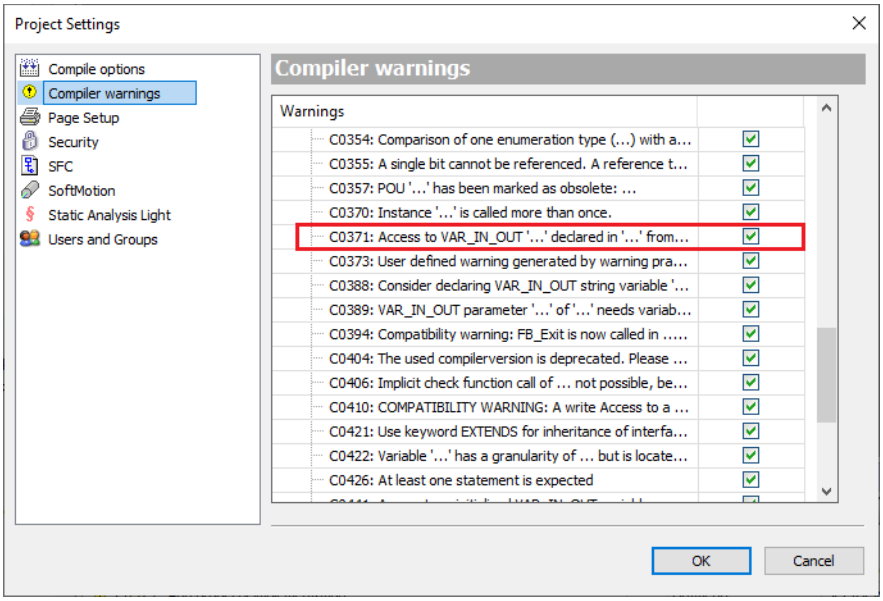


Messages - Total 5 error(s), 5 warning(s), 0 message(s)

Precompile | 5 error(s) | 5 warning(s) | 0 message(s)

Description	Project	Object	Position
C0373: Add method implementation	Untitled8	METH [Device: Progr…	Line 1 (Decl)
C0373: Add property implementation	Untitled8	Set [Device: Progra…	
C0373: Add property implementation	Untitled8	Set [Device: Progra…	
C0373: Add property implementation	Untitled8	Get [Device: Progra…	
C0373: Add property implementation	Untitled8	Prop [Device: Progr…	Line 1

These warning messages do not indicate any problems. However, to prevent particular warning messages from being displayed, from the menu bar, select **Project**>**Project Settings** and then clear the check boxes of the target warning messages in the Compiler Warnings pane.



Project Settings

Compiler warnings

Warnings	
C0354: Comparison of one enumeration type (…) with a…	☑
C0355: A single bit cannot be referenced. A reference t…	☑
C0357: POU '…' has been marked as obsolete: …	☑
C0370: Instance '…' is called more than once.	☑
C0371: Access to VAR_IN_OUT '…' declared in '…' from…	☑
C0373: User defined warning generated by warning pra…	☑
C0388: Consider declaring VAR_IN_OUT string variable '…	☑
C0389: VAR_IN_OUT parameter '…' of '…' needs variab…	☑
C0394: Compatibility warning: FB_Exit is now called in …..	☑
C0404: The used compilerversion is deprecated. Please …	☑
C0406: Implicit check function call of … not possible, be…	☑
C0410: COMPATIBILITY WARNING: A write Access to a …	☑
C0421: Use keyword EXTENDS for inheritance of interfa…	☑
C0422: Variable '…' has a granularity of … but is locate…	☑
C0426: At least one statement is expected	☑

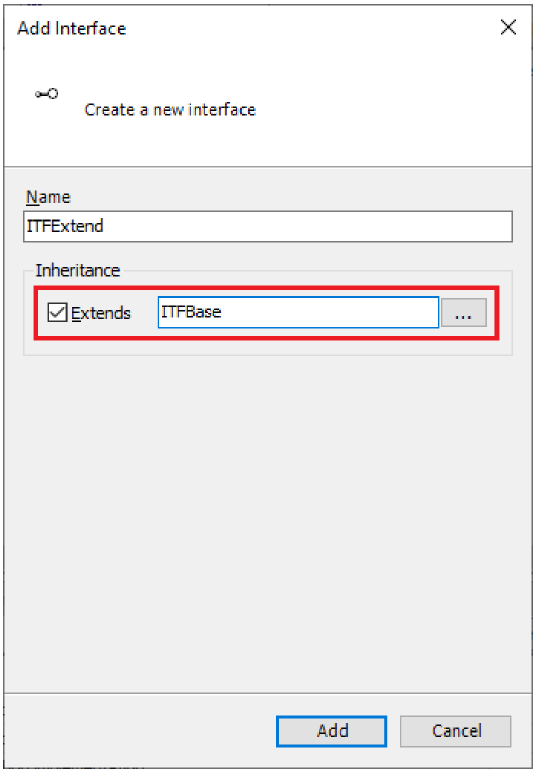
10.5.4 Extending the Interface

Existing interfaces can be inherited and extended.

This section explains how to extend existing interface "ITFBase" and create new interface "ITFExtend".
In this example, when interface "ITFBase" exists, create a new interface as below.

1 2 Procedure

1. Right-click the "Application" object in the navigator pane and then select **Add Object>Interface** from the context-sensitive menu that is displayed.
   The "Add Interface" dialog box will be displayed.



Add Interface

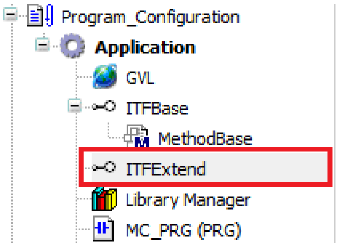
Create a new interface

Name: ITFExtend

Inheritance

☑ Extends ITFBase

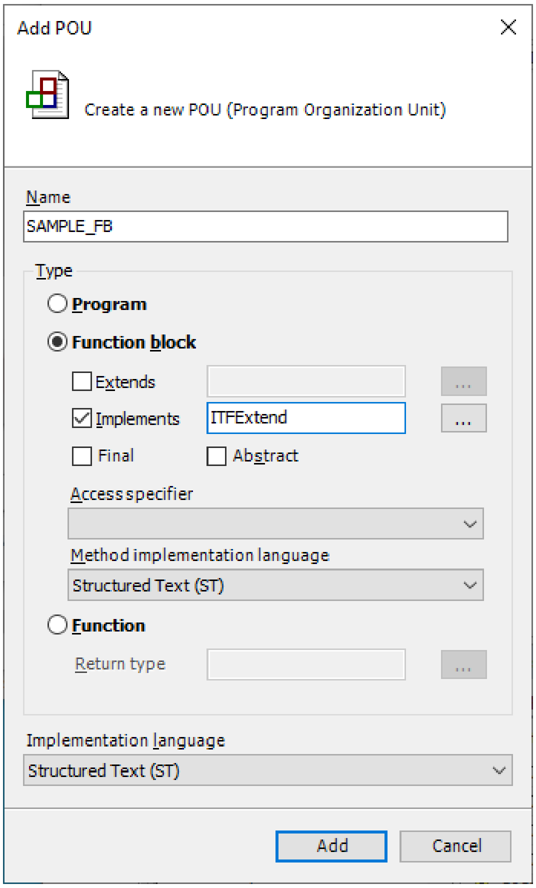
2. In the "Add Interface" dialog box, select the "Extends" check box and enter an interface to be inherited.
   Interface "ITFExtend" will be created.



3. Right-click the "ITFExtend" object added to the navigator pane and then select **Add Object**>**Interface property** or **method** from the context-sensitive menu that is displayed in order to add a property or method for the "ITFExtend" object.

4. Right-click the "Application" object in the navigator pane and then select **Add Object**>**POU** from the context-sensitive menu that is displayed.

The "Add POU" dialog box will be displayed.


Add POU

Create a new POU (Program Organization Unit)

Name
SAMPLE_FB

Type
○ **Program**
● **Function block**
☐ Extends
☑ Implements ITFExtend
☐ Final ☐ Abstract
Access specifier
Method implementation language
Structured Text (ST)
○ **Function**
Return type

Implementation language
Structured Text (ST)

5. Enter a name. In the Type section, select the Function block option, select the "Implement" check box, and enter "ITFExtend". Click the [Add] button.

A function block with properties and methods for both interfaces "ITFExtend" and "ITFBase" will be added under the "Application" object.

10.6 External File Functions

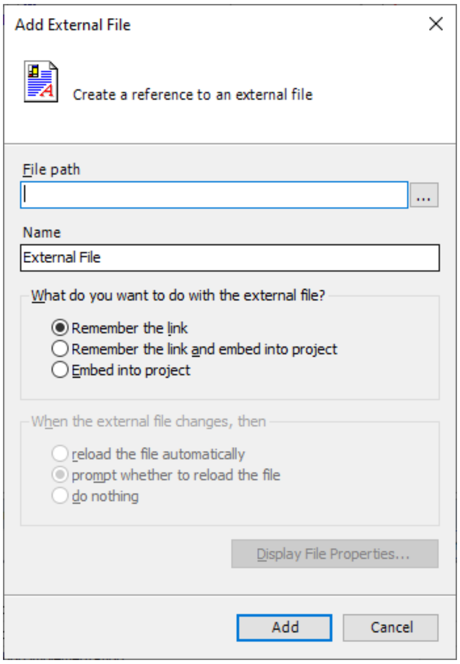
The external file object allows text files, image files, and other files to be saved in the project.

10.6.1 Setting up an External File Object

This section explains how to add an external file object.

Procedure

1. Right-click the "Application" object in the navigator pane and then select **Add Object**>**External File** from the context-sensitive menu that is displayed.
   The "Add External File" dialog box will be displayed.



Add External File

Create a reference to an external file

File path:

Name: External File

What do you want to do with the external file?

● Remember the link
○ Remember the link and embed into project
○ Embed into project

When the external file changes, then

○ reload the file automatically
● prompt whether to reload the file
○ do nothing

2. In the File path field, specify a file to be registered. In the Name field, enter the name of the file.
   In the "What do you want to do with the external file?" section, select an appropriate option as the method for registering the external file in the project.
   If you select the "Remenber the link and embed into project" option in the "What do you want to do with the external file?" section, select an appropriate option in the "When the external file changes, then" section to specify settings for update processing to be performed when the external file is changed.
   "What do you want to do with the external file?":

   "Remember the link"
   Stores only the path to the file in the project.
   If the file does not exist at the link destination, it cannot be used in the project.

"Remember the link and embed into project"

Saves a copy of the file and a link to the file to the project at the same time.

As long as the file exists at the link destination, the setting of the "When the external file changes, then" section applies.

If the file does not exist, a version of the file that is stored in the project will be used.

"Embed into project"

Saves a copy of the file to the project.

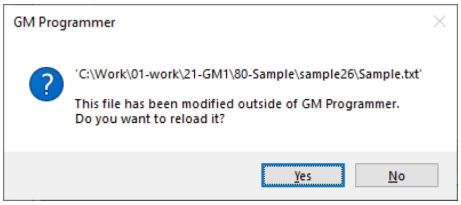
"When the external file changes, then":

"reload the file automatically"

Updates a copy of an external file in the project when the corresponding external file is changed.

"prompt whether to reload the file"

Opens a dialog box, asking whether to update a copy of an external file in the project, when the corresponding external file is changed.



GM Programmer

'C:\Work\01-work\21-GM1\80-Sample\sample26\Sample.txt'

This file has been modified outside of GM Programmer.
Do you want to reload it?

"do nothing"

Does not update a copy of an external file in the project even if the corresponding external file is changed.

3. Click the [Add] button.

The external file will be added under the "Application" object.

Info.

- When the added external file is opened in GM Programmer, a copy of the file is temporarily created in the following folder:

  C:\Users\<user name>\AppData\Local\Temp
- Files added as "external file" objects cannot be accessed from programs such as POU.

10.7 Servo Amplifier / Motor Operation Function (PANATERM Lite for GM)

You can start PANATERM Lite for GM, which allows you to check the settings of servo amplifiers, the operating states of servo amplifiers and motors, and the error status of servo amplifiers via the GM1 controller.

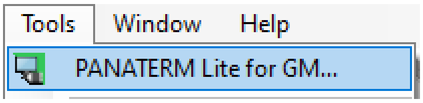
For details, refer to "16 Overview of PANATERM Lite for GM".

10.7.1 Starting PANATERM Lite for GM

This section explains how to start PANATERM Lite for GM from GM Programmer.

1 2 Procedure

1. From the menu bar, select **Tools**>**PANATERM Lite for GM**. PANATERM Lite for GM will be started.



11 Motion Control

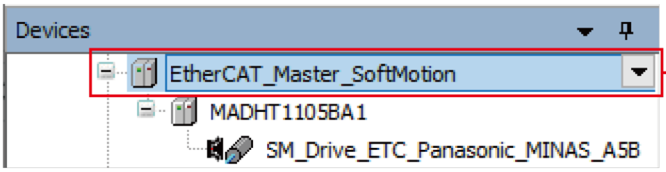
11.1 Setting up the EtherCAT master

11.1.1 EtherCAT_Master_SoftMotion

■ Overview

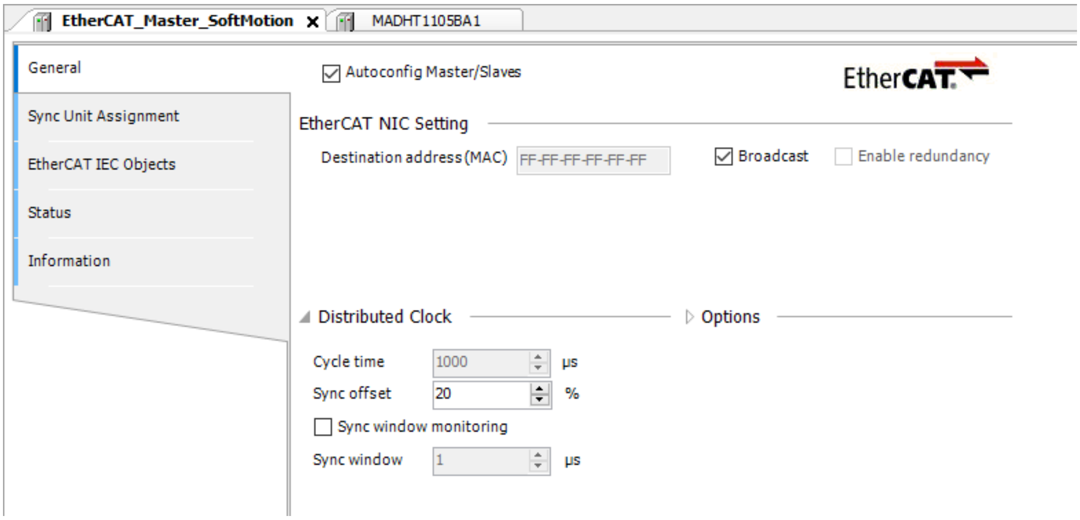
The EtherCAT master must be configured as described below.

This section explains various tabs and settings of EtherCAT_Master_SoftMotion editor.



EtherCAT master setting

Set EtherCAT Master parameters on the General window.



☑ Autoconfig Master/Slaves

EtherCAT NIC Setting

Destination address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast □ Enable redundancy

Distributed Clock

Cycle time 1000 µs

Sync offset 20 %

□ Sync window monitoring

Sync window 1 µs

List of parameters

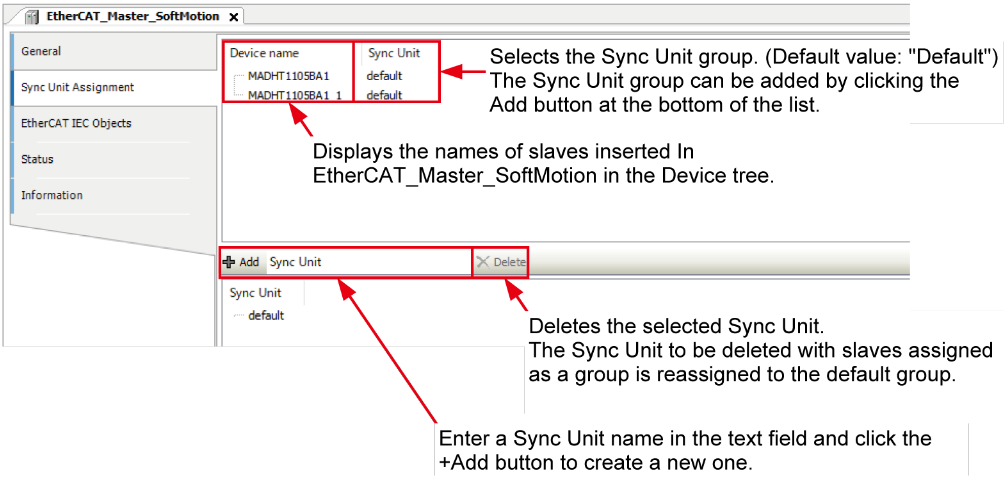
Name		Description
Automatic Master / Slave Configuration		When this parameter is enabled, it automatically performs configuration of the master and slave. When this parameter is disabled, all configuration settings for the master and slave must be made manually. Since configuration settings require expertise, it is recommended to set to automatic settings.
EtherCAT NIC Setting	Storage location address (MAC)	Specifies the destination address.
	Broadcast	When this setting is enabled, a broadcast address (FF-FF-FF-FF-FF-FF) is set in the storage location address (destination). When this setting is disabled, set a multicast address in the storage location address (destination).
Distributed Clock	Cycle Time	Sets the EtherCAT communication cycle. The MotionTask cycle is set.

Name		Description
	Synchronous Offset	Sets the delay time between the DC reference time of the EtherCAT slave and the EtherCAT task (PLC's cycle start time).
	Synchronous Window Monitoring	When this setting is enabled, it monitors slave synchronization.
	Synchronous Window	Sets the synchronous window monitoring time When all slave's synchronizations occur within the set time, xSyncInWindow (IoDrvEtherCAT) turns to TRUE.
Options	Use LRW instead of LWR / LRD.	When this setting is enabled, the LRW command (read/write) is used. When this setting is disabled, the LRD command (read) and LWR command (write) are used as separate commands.
	Enable message for every task.	When this setting is enabled, the PDO mapped variables are updated at the point of use of the POU. When this setting is disabled, the PDO mapped variables are updated by a MotionTask.
	Autostart slave.	When a communication breakdown occurs, automatically the master attempts to restart the slave immediately.
Master Configuration		It can be set only when Automatic Master / Slave Configuration is disabled.
	Image In Address	Sets the first logical address of the first slave used for PDO input data.
	Image Out Address	Sets the first logical address of the first slave used for PDO output data.

■ Sync Unit Assignment

In Sync Unit Assignment, multiple slaves can be grouped together and managed as a Sync Unit.

For example, one slave within a Sync Unit becomes unknown, the rest of the slaves in the Sync Unit are displayed as unknown. Since other Sync Units not affected remain operable, the unknown group can be promptly corrected using the device diagnostics.



Selects the Sync Unit group. (Default value: "Default")
The Sync Unit group can be added by clicking the Add button at the bottom of the list.

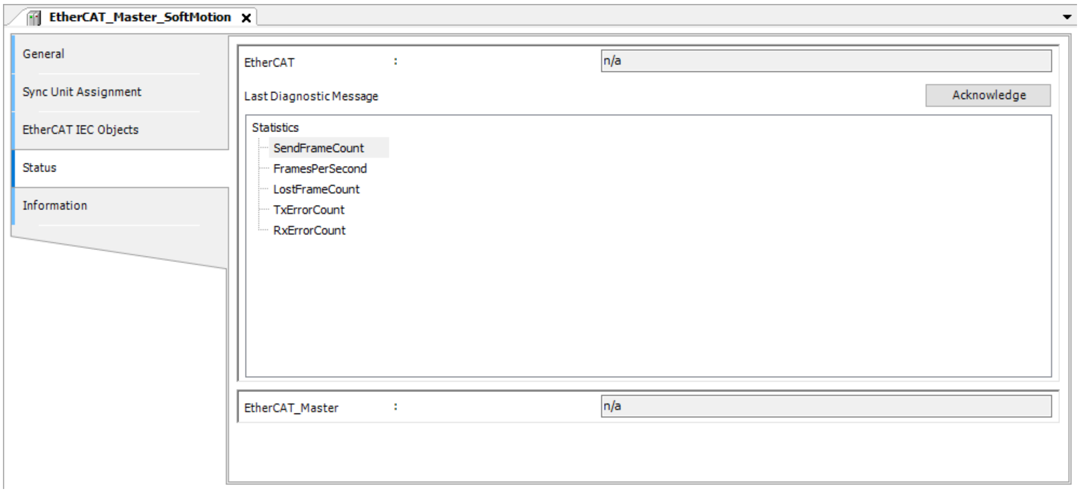
Displays the names of slaves inserted In EtherCAT_Master_SoftMotion in the Device tree.

Deletes the selected Sync Unit.
The Sync Unit to be deleted with slaves assigned as a group is reassigned to the default group.

Enter a Sync Unit name in the text field and click the +Add button to create a new one.

■ Status tab

Diagnostic messages of the EtherCAT_Master_SoftMotion device are displayed. They can be checked with the program using the"GetStatistics"method of"IoDevEtherCAT".



EtherCAT : n/a

Last Diagnostic Message

Statistics
- SendFrameCount
- FramesPerSecond
- LostFrameCount
- TxErrorCount
- RxErrorCount

EtherCAT_Master : n/a

Item	Description
SendFrameCount	The number of EtherCAT send frames is displayed.
FramesPerSecond	The number of send frames per second is displayed.
LostFrameCount(Note 1)	The number of frame losses is displayed.
TxErrorCount	The number of send errors is counted.
RxErrorCount	The number of receive errors is counted.

(Note 1) Check the following when the number of frame losses is other than 0.

- Influence of noise
- Connection status with slaves
- Program load

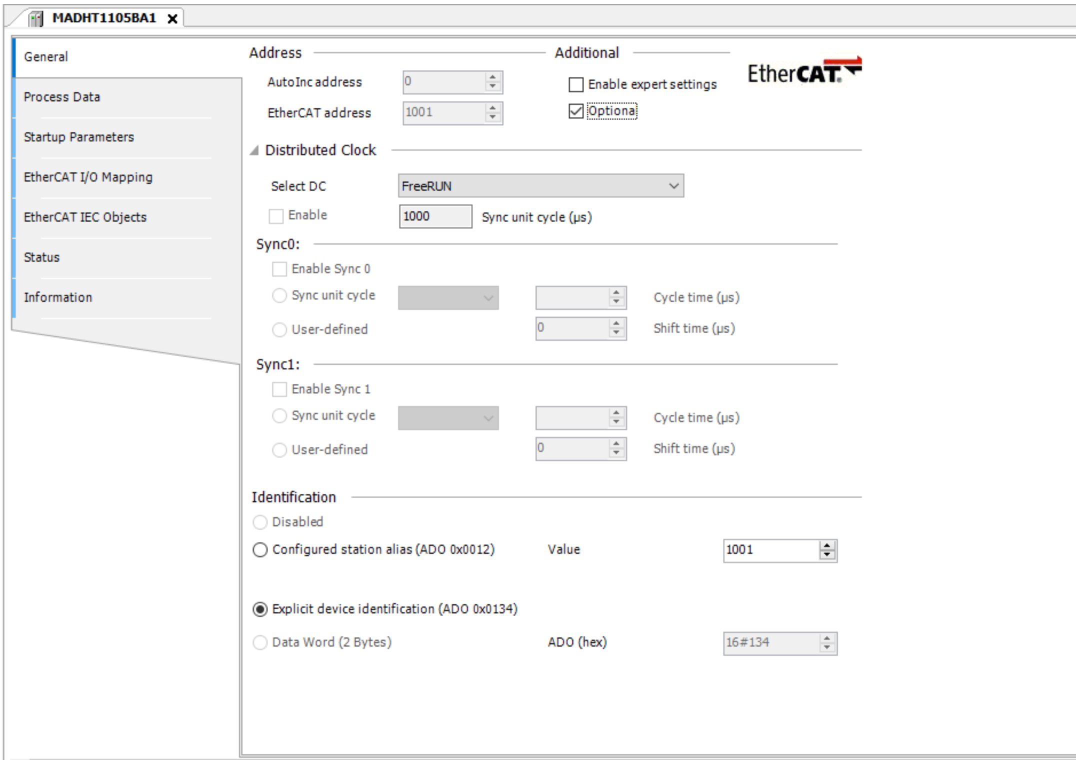
11.2 EtherCAT Slave Settings

11.2.1 EtherCAT Slave Settings

■ Overview

The EtherCAT slave must be configured as described below. The description below explains how to set the servo amplifier (A5B: MADHT1105BA1) manufactured by Panasonic.

Set EtherCAT slave parameters on the General window. In the Sync Unit cycle, the cycle set in MotionTask is set.



MADHT1105BA1

General
Process Data
Startup Parameters
EtherCAT I/O Mapping
EtherCAT IEC Objects
Status
Information

Address
AutoInc address: 0
EtherCAT address: 1001

Additional
□ Enable expert settings
☑ Optiona

Distributed Clock
Select DC: FreeRUN
□ Enable
1000 Sync unit cycle (µs)

Sync0:
□ Enable Sync 0
○ Sync unit cycle — Cycle time (µs)
○ User-defined — 0 — Shift time (µs)

Sync1:
□ Enable Sync 1
○ Sync unit cycle — Cycle time (µs)
○ User-defined — 0 — Shift time (µs)

Identification
○ Disabled
○ Configured station alias (ADO 0x0012) — Value: 1001
● Explicit device identification (ADO 0x0134)
○ Data Word (2 Bytes) — ADO (hex): 16#134

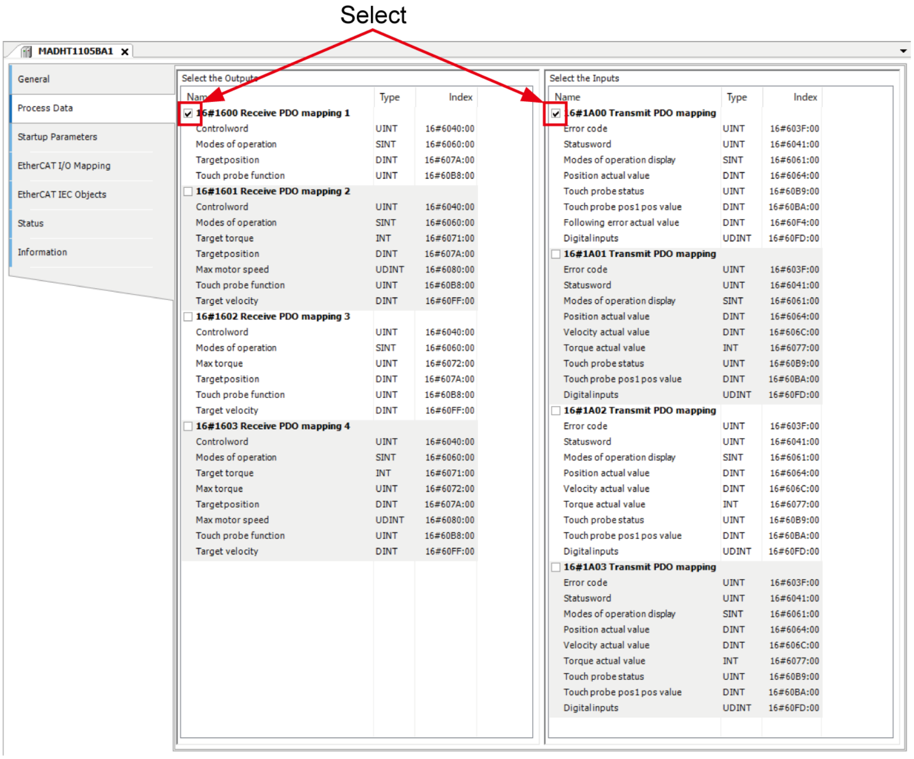
List of parameters

Name		Description
Address		It can be set only when"Automatic Master / Slave Configuration"is disabled in EtherCAT master setting.
	AutoInc address	Sets a tentative address used before the master assign an EtherCAT address to the slave.
	EtherCAT address	Sets an EtherCAT address to the slave.
Add	Enable expert settings	When this setting is enabled, expert setting can be made. Since expert settings are not required for standard applications, it is recommended to disable the settings.
	Option	When this setting is enabled, the slave is defined as an option. As a result, an error message is not generated even when a device is

Name		Description
		not found in the bus system. When this setting is enabled, assign a unique ID in the setting for the identification section. ● Requirements for use • The"Automatic Master / Slave Configuration".parameter on the"General"tab of EtherCAT_Master_SoftMotion editor is enabled. • The EtherCAT slave supports this function.
Distributed Clock	DC selection	Sets the synchronization method between the master and the slave. Example: ● FreeRUN: Non-synchronize with master’s cycle time. ● DC Sync0: Based on the time of the 1st axis, synchronize time information of other slaves.
Identification	Disable	Slave identification is not checked.
	Configured station alias value	Sets the address stored in the slave device EEPROM.
	Explicit device identification	Sets a device identification number for hardware such as a DIP switch of the slave device.
	Data word (2 bytes)	A 2-byte identification number is saved. (MINAS: Not supported)

■ Process data

PDO (Process Data Object) is data that is updated between the master and the slave for every EtherCAT communication cycle. The configuration of the PDO varies depending on the slave. Select an appropriate one for each axis control method. The selected PDO mapping is reflected on the"EtherCAT I/O mapping"and can be used in the program.



Select

Select the Outputs

Name	Type	Index
☑ **16#1600 Receive PDO mapping 1**		
Controlword	UINT	16#6040:00
Modes of operation	SINT	16#6060:00
Targetposition	DINT	16#607A:00
Touch probe function	UINT	16#60B8:00
☐ **16#1601 Receive PDO mapping 2**		
Controlword	UINT	16#6040:00
Modes of operation	SINT	16#6060:00
Target torque	INT	16#6071:00
Targetposition	DINT	16#607A:00
Max motor speed	UDINT	16#6080:00
Touch probe function	UINT	16#60B8:00
Target velocity	DINT	16#60FF:00
☐ **16#1602 Receive PDO mapping 3**		
Controlword	UINT	16#6040:00
Modes of operation	SINT	16#6060:00
Max torque	UINT	16#6072:00
Targetposition	DINT	16#607A:00
Touch probe function	UINT	16#60B8:00
Target velocity	DINT	16#60FF:00
☐ **16#1603 Receive PDO mapping 4**		
Controlword	UINT	16#6040:00
Modes of operation	SINT	16#6060:00
Target torque	INT	16#6071:00
Max torque	UINT	16#6072:00
Targetposition	DINT	16#607A:00
Max motor speed	UDINT	16#6080:00
Touch probe function	UINT	16#60B8:00
Target velocity	DINT	16#60FF:00

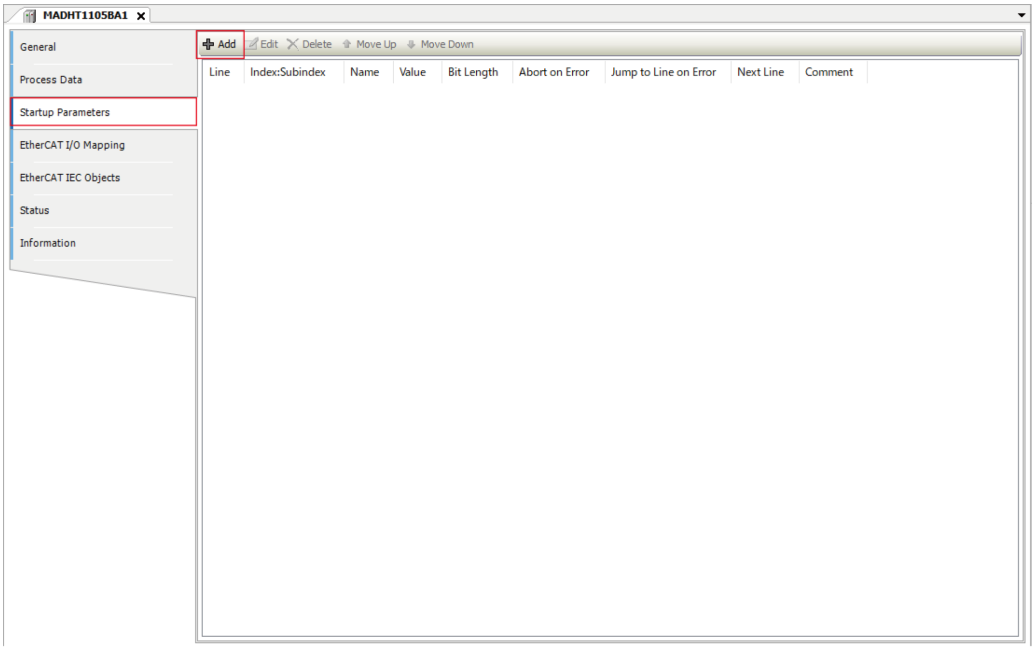
Select the Inputs

Name	Type	Index
☑ **16#1A00 Transmit PDO mapping**		
Error code	UINT	16#603F:00
Statusword	UINT	16#6041:00
Modes of operation display	SINT	16#6061:00
Position actual value	DINT	16#6064:00
Touch probe status	UINT	16#60B9:00
Touch probe pos1 pos value	DINT	16#60BA:00
Following error actual value	DINT	16#60F4:00
Digitalinputs	UDINT	16#60FD:00
☐ **16#1A01 Transmit PDO mapping**		
Error code	UINT	16#603F:00
Statusword	UINT	16#6041:00
Modes of operation display	SINT	16#6061:00
Position actual value	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Torque actual value	INT	16#6077:00
Touch probe status	UINT	16#60B9:00
Touch probe pos1 pos value	DINT	16#60BA:00
Digitalinputs	UDINT	16#60FD:00
☐ **16#1A02 Transmit PDO mapping**		
Error code	UINT	16#603F:00
Statusword	UINT	16#6041:00
Modes of operation display	SINT	16#6061:00
Position actual value	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Torque actual value	INT	16#6077:00
Touch probe status	UINT	16#60B9:00
Touch probe pos1 pos value	DINT	16#60BA:00
Digitalinputs	UDINT	16#60FD:00
☐ **16#1A03 Transmit PDO mapping**		
Error code	UINT	16#603F:00
Statusword	UINT	16#6041:00
Modes of operation display	SINT	16#6061:00
Position actual value	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Torque actual value	INT	16#6077:00
Touch probe status	UINT	16#60B9:00
Touch probe pos1 pos value	DINT	16#60BA:00
Digitalinputs	UDINT	16#60FD:00

■ Startup parameter

Using the SDO (Service Data Object), slave parameters can be set at the start of RUN.

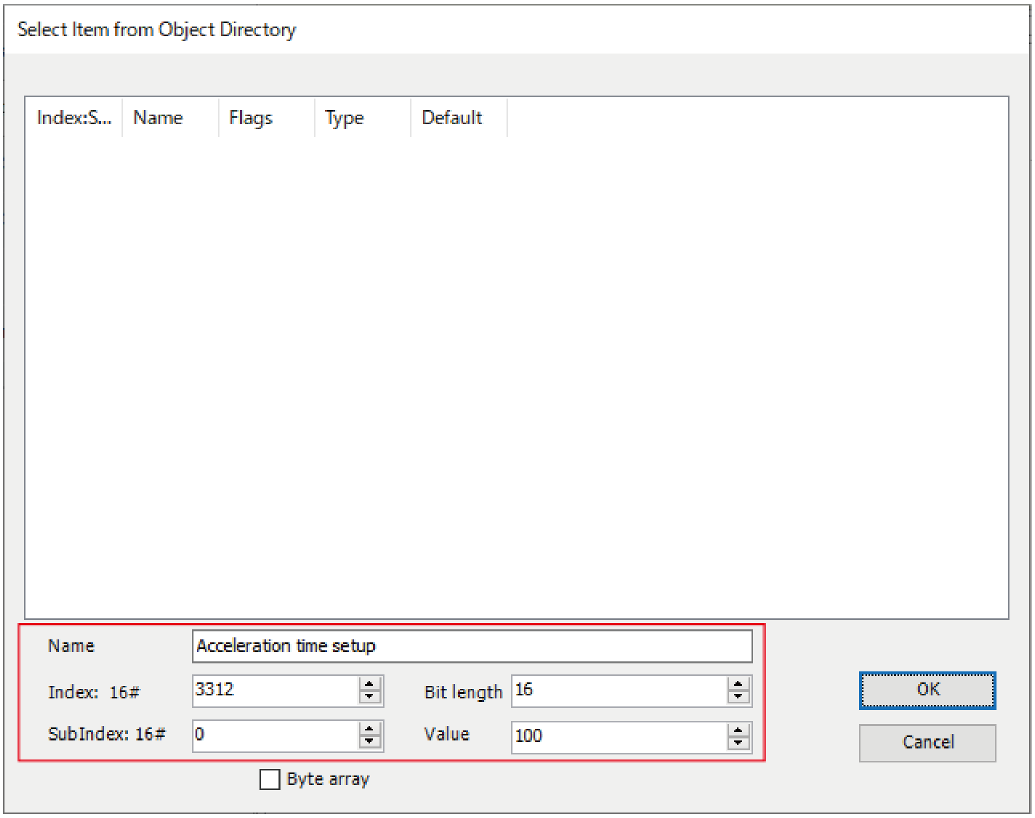
1. Click"Add".



2. Enter object information to be changed.

   The following explanation is provided for an example where "Acceleration time setting" is changed.

   - Name: Any name
   - Index 16#: 3312
   - Sub-index 16#: 0
   - Bit length: 16
   - Value: 100



Select Item from Object Directory

Index:S...	Name	Flags	Type	Default

Name: Acceleration time setup

Index: 16# 3312

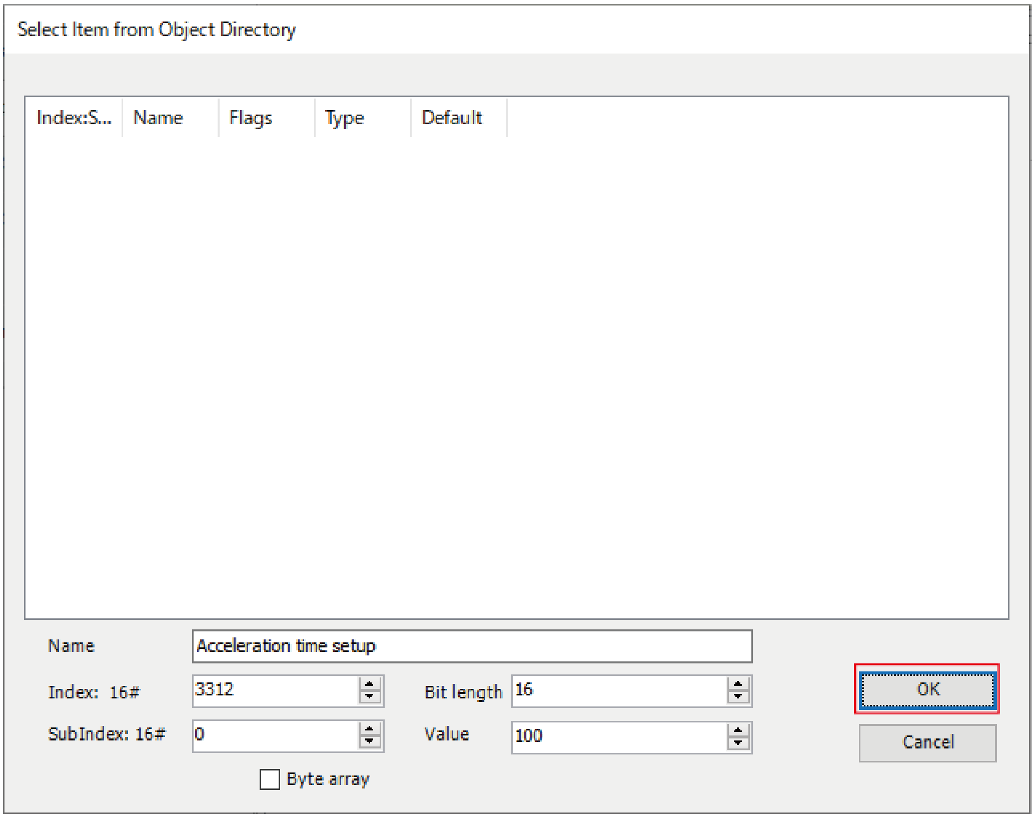
Bit length 16

SubIndex: 16# 0

Value 100

□ Byte array

3. Click the[OK]button.



Select Item from Object Directory

Index:S...	Name	Flags	Type	Default

Name: Acceleration time setup

Index: 16# 3312

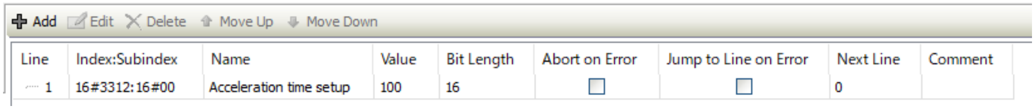
Bit length 16

SubIndex: 16# 0

Value 100

□ Byte array

The data will be displayed as follows.



Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Error	Next Line	Comment
1	16#3312:16#00	Acceleration time setup	100	16	□	□	0	

■ Online

When set to online, the status of the ESM (EtherCAT State Machine) can be changed and displayed, firmware files can be written or read, and configuration information can be written or read to EEPROM.

- Requirements for use
  - "Enable expert settings"on the"General"tab is enabled.
  - You are already logged in to the controller.

Note

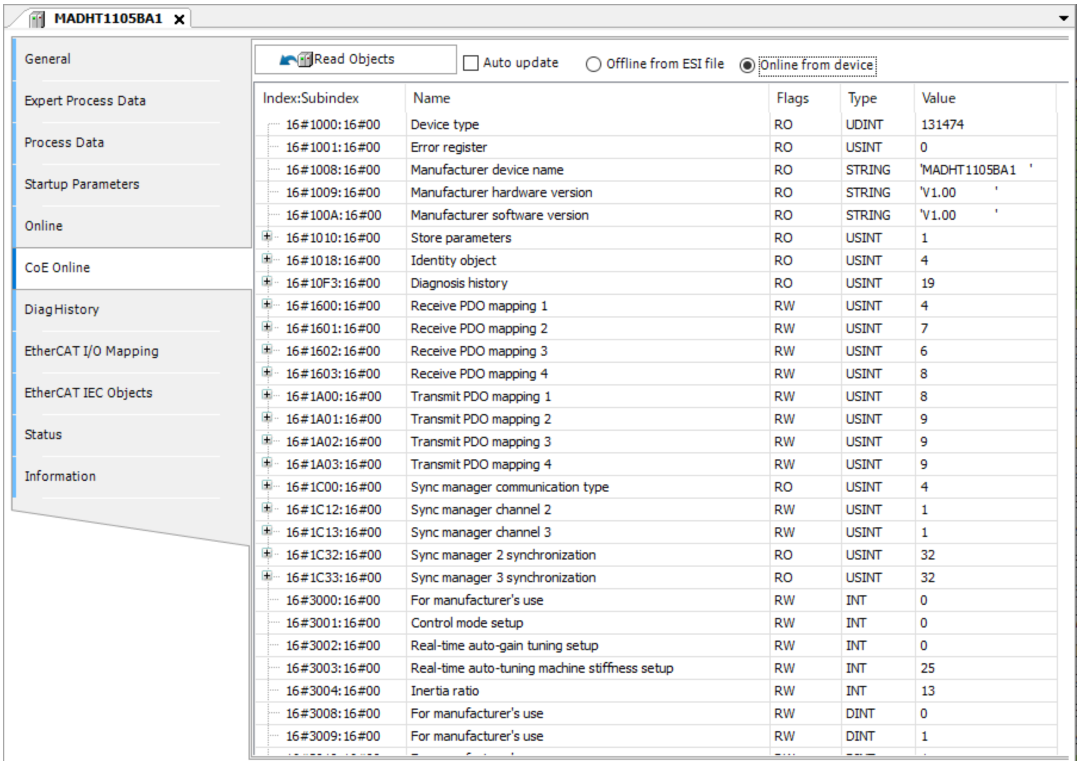
- Writing a wrong file or incorrect configuration information may cause the slave to operate in an abnormal manner. Be careful when using the online mode.

■ CoE online tab

Set values can be written or read from the ESI file or the object dictionary of the slave device.

- Requirements for use
  - "Enable expert settings"on the"General"tab is enabled.

- You are already logged in to the controller.



□ Auto update ○ Offline from ESI file ● Online from device

Index:Subindex	Name	Flags	Type	Value
16#1000:16#00	Device type	RO	UDINT	131474
16#1001:16#00	Error register	RO	USINT	0
16#1008:16#00	Manufacturer device name	RO	STRING	'MADHT1105BA1 '
16#1009:16#00	Manufacturer hardware version	RO	STRING	'V1.00 '
16#100A:16#00	Manufacturer software version	RO	STRING	'V1.00 '
16#1010:16#00	Store parameters	RO	USINT	1
16#1018:16#00	Identity object	RO	USINT	4
16#10F3:16#00	Diagnosis history	RO	USINT	19
16#1600:16#00	Receive PDO mapping 1	RW	USINT	4
16#1601:16#00	Receive PDO mapping 2	RW	USINT	7
16#1602:16#00	Receive PDO mapping 3	RW	USINT	6
16#1603:16#00	Receive PDO mapping 4	RW	USINT	8
16#1A00:16#00	Transmit PDO mapping 1	RW	USINT	8
16#1A01:16#00	Transmit PDO mapping 2	RW	USINT	9
16#1A02:16#00	Transmit PDO mapping 3	RW	USINT	9
16#1A03:16#00	Transmit PDO mapping 4	RW	USINT	9
16#1C00:16#00	Sync manager communication type	RO	USINT	4
16#1C12:16#00	Sync manager channel 2	RW	USINT	1
16#1C13:16#00	Sync manager channel 3	RW	USINT	1
16#1C32:16#00	Sync manager 2 synchronization	RO	USINT	32
16#1C33:16#00	Sync manager 3 synchronization	RO	USINT	32
16#3000:16#00	For manufacturer's use	RW	INT	0
16#3001:16#00	Control mode setup	RW	INT	0
16#3002:16#00	Real-time auto-gain tuning setup	RW	INT	0
16#3003:16#00	Real-time auto-tuning machine stiffness setup	RW	INT	25
16#3004:16#00	Inertia ratio	RW	INT	13
16#3008:16#00	For manufacturer's use	RW	DINT	0
16#3009:16#00	For manufacturer's use	RW	DINT	1

11.3 EtherCAT Axis Settings

11.3.1 Overview of EtherCAT Axis Settings

To use the motion function, you must configure axis settings for EtherCAT.
This section explains how to set axis information for EtherCAT in GM Programmer.

Info.

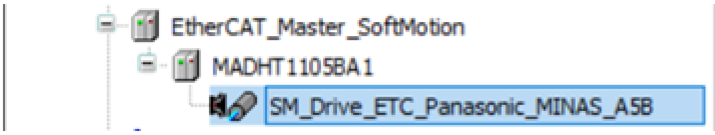
- For details on how to set up EtherCAT and add axes, refer to "5.3 Setting up Motion Control".

11.3.2 Basic Settings for EtherCAT Axes

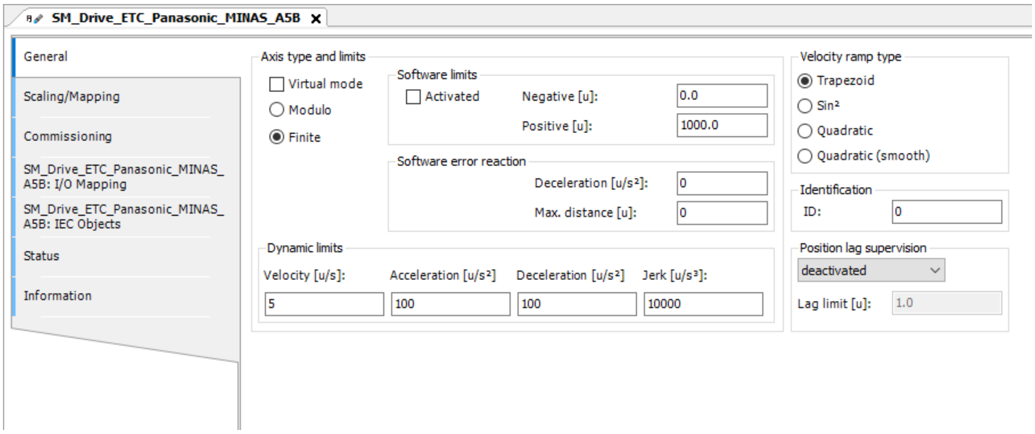
- Be sure to configure settings for EtherCAT axes.

Procedure

1. Double-click the servo amplifier object in the navigator pane.



The setting pane will be displayed in the main pane.


SM_Drive_ETC_Panasonic_MINAS_A5B
General
Scaling/Mapping
Commissioning
SM_Drive_ETC_Panasonic_MINAS_A5B: I/O Mapping
SM_Drive_ETC_Panasonic_MINAS_A5B: IEC Objects
Status
Information

Axis type and limits
□ Virtual mode
○ Modulo
● Finite

Software limits
□ Activated
Negative [u]: 0.0
Positive [u]: 1000.0

Software error reaction
Deceleration [u/s²]: 0
Max. distance [u]: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
5	100	100	10000

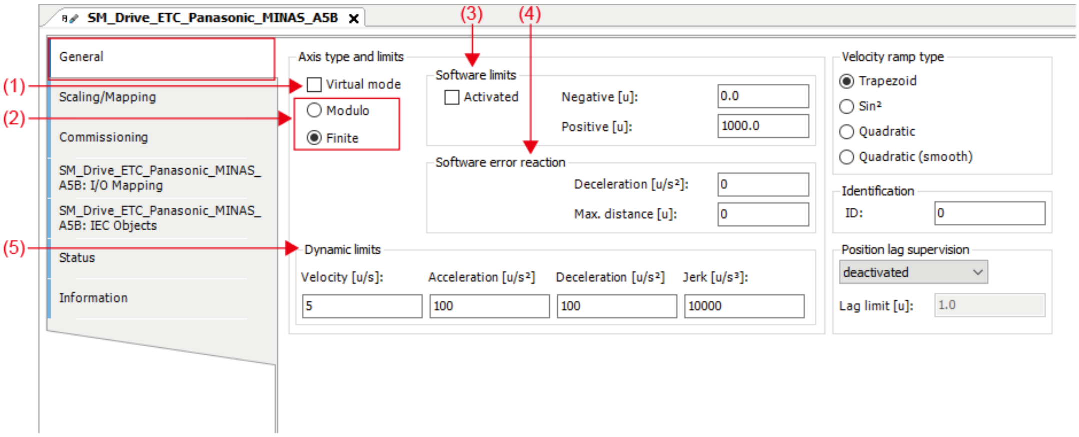
Velocity ramp type
● Trapezoid
○ Sin²
○ Quadratic
○ Quadratic (smooth)

Identification
ID: 0

Position lag supervision
deactivated
Lag limit [u]: 1.0

General Settings

Select the "General" tab and set the following items.


SM_Drive_ETC_Panasonic_MINAS_A5B
General
Scaling/Mapping
Commissioning
SM_Drive_ETC_Panasonic_MINAS_A5B: I/O Mapping
SM_Drive_ETC_Panasonic_MINAS_A5B: IEC Objects
Status
Information
(1)
(2)
(3)
(4)
(5)

Axis type and limits

□ Virtual mode
○ Modulo
● Finite

Software limits

□ Activated
Negative [u]: 0.0
Positive [u]: 1000.0

Software error reaction

Deceleration [u/s²]: 0
Max. distance [u]: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
5	100	100	10000

Velocity ramp type

● Trapezoid
○ Sin²
○ Quadratic
○ Quadratic (smooth)

Identification

ID: 0

Position lag supervision

deactivated
Lag limit [u]: 1.0

(1) Virtual mode

You can set real axes or virtual axes.

Use of real axes: Real axes are used to actually control the servo amplifier.

Use of virtual axes: Virtual axes create and execute a virtual servo amplifier within the GM1 Controller.

(2) Modulo / Finite

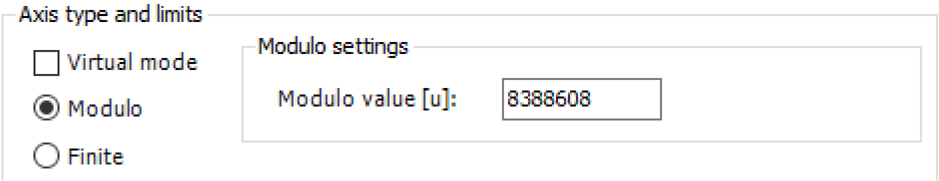
The axis type can be specified.

- Modulo

  Modulo: The motor rotates infinitely (belt drive, etc.) without limiting the travel range.

  - The value of the commanded position keeps looping between 0 and modulo value.
  - The maximum settable modulo value is "255×units in application" (*1).

    *1: For "units in application", specify settings in the "Scaling / Mapping" tab.
  - A negative value cannot be set. (A warning is issued. If a download is performed without correcting the value, an error will occur when the GM1 Controller is started.



Axis type and limits

□ Virtual mode
● Modulo
○ Finite

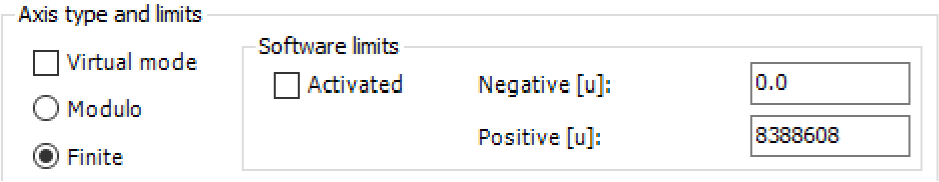
Modulo settings

Modulo value [u]: 8388608

- Finite

  The set value of the commanded position is a finite value.

  Soft limits can be set. Note that an error will occur if a 32-bit real number is exceeded.



Axis type and limits

□ Virtual mode
○ Modulo
● Finite

Software limits

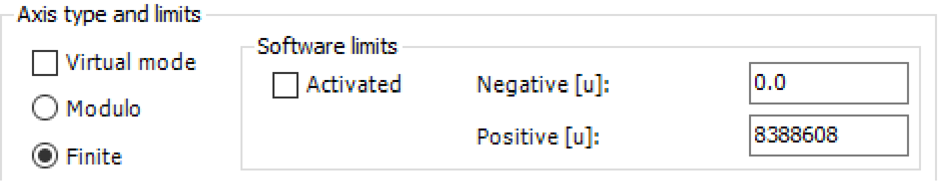
□ Activated
Negative [u]: 0.0
Positive [u]: 8388608

(3) Soft limits

If the axis type is set to "Finite", soft limits can be set.

If the commanded position falls outside the soft limit setting range, an error stop will occur, causing the operation to stop.

If the operation is stopped because the soft limit setting range is exceeded, the value specified in "Deceleration" or "Max. distance" in "Soft error reaction" or the value specified in "Deceleration" in "Dynamic limits" will be applied, whichever is the shortest time from when deceleration starts until a stoppage occurs.



Axis type and limits

□ Virtual mode

○ Modulo

● Finite

Software limits

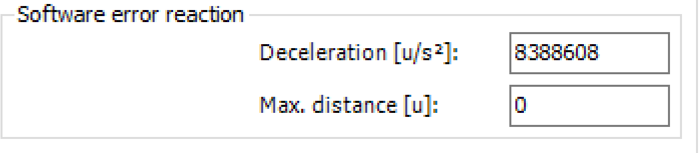
□ Activated

Negative [u]: 0.0

Positive [u]: 8388608

(4) Software error reaction

Settings can be configured to stop operation when an error occurs.



Software error reaction

Deceleration [u/s²]: 8388608

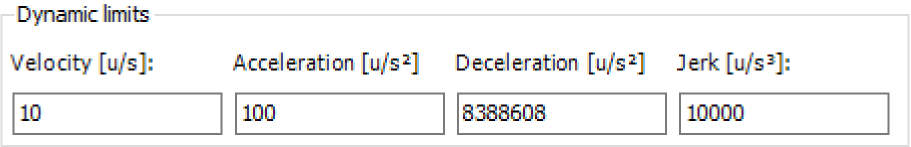
Max. distance [u]: 0

Info.

- If the mode is switched from run to stop during operation, an emergency stop will be executed, regardless of the settings in "Software error reaction".
- For stop operation that takes place when an error stop occurs or when the soft limit range is exceeded, one of the values specified in the following items is applied, whichever is the shortest time from when deceleration starts until a stoppage occurs.
  - "Deceleration" in "Software error reaction"
  - "Max. distance" in "Software error reaction"
  - "Deceleration" in "Dynamic limits"
- If "Deceleration" and "Max. distance" in "Software error reaction" are set to 0, these settings will be disabled. In this case, operation is stopped according to the value specified in "Deceleration" in "Dynamic limits".

(5) Dynamic limits

"Velocity", "Acceleration", and "Deceleration" cannot be set to 0. If they are set to 0, a warning will be issued.



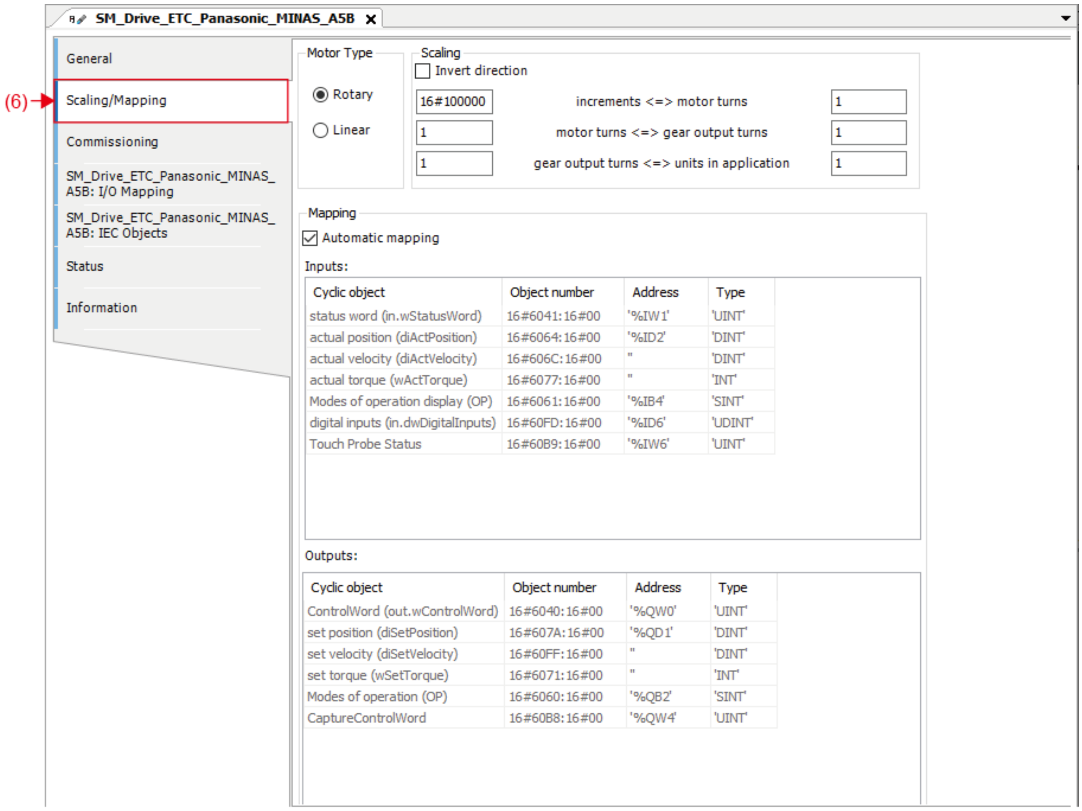
Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
10	100	8388608	10000

The "SMC_CheckLimits" function block can be used to check whether the values set in "Dynamic limits" were exceeded during axis operations. Note that the "SMC_CheckLimits" function block cannot detect whether the value set in "Jerk" was exceeded. Therefore, do not use the "Jerk" field.

Scaling / Mapping Settings

Select the "Scaling / Mapping" tab and set the following items.



SM_Drive_ETC_Panasonic_MINAS_A5B

- General
- (6) Scaling/Mapping
- Commissioning
- SM_Drive_ETC_Panasonic_MINAS_A5B: I/O Mapping
- SM_Drive_ETC_Panasonic_MINAS_A5B: IEC Objects
- Status
- Information

Motor Type

● Rotary

○ Linear

Scaling

□ Invert direction

16#100000	increments <=> motor turns	1
1	motor turns <=> gear output turns	1
1	gear output turns <=> units in application	1

Mapping

☑ Automatic mapping

Inputs:

Cyclic object	Object number	Address	Type
status word (in.wStatusWord)	16#6041:16#00	'%IW1'	'UINT'
actual position (diActPosition)	16#6064:16#00	'%ID2'	'DINT'
actual velocity (diActVelocity)	16#606C:16#00	''	'DINT'
actual torque (wActTorque)	16#6077:16#00	''	'INT'
Modes of operation display (OP)	16#6061:16#00	'%IB4'	'SINT'
digital inputs (in.dwDigitalInputs)	16#60FD:16#00	'%ID6'	'UDINT'
Touch Probe Status	16#60B9:16#00	'%IW6'	'UINT'

Outputs:

Cyclic object	Object number	Address	Type
ControlWord (out.wControlWord)	16#6040:16#00	'%QW0'	'UINT'
set position (diSetPosition)	16#607A:16#00	'%QD1'	'DINT'
set velocity (diSetVelocity)	16#60FF:16#00	''	'DINT'
set torque (wSetTorque)	16#6071:16#00	''	'INT'
Modes of operation (OP)	16#6060:16#00	'%QB2'	'SINT'
CaptureControlWord	16#60B8:16#00	'%QW4'	'UINT'

(6) Scaling / Mapping

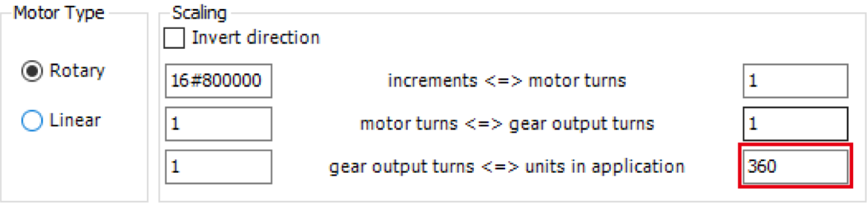
- Rotary type

  When the axis type is set to "Modulo", the ratio in conversion from drive increments to the application unit is set.

  The unit on the servo amplifier and the unit on the application (POU) are converted.

  Example:

  One revolution of the MINAS A6B MADHT1105BA1 is 0x800000. To treat one revolution as 360 in the application, set this ratio to 360.



Motor Type

● Rotary

○ Linear

Scaling

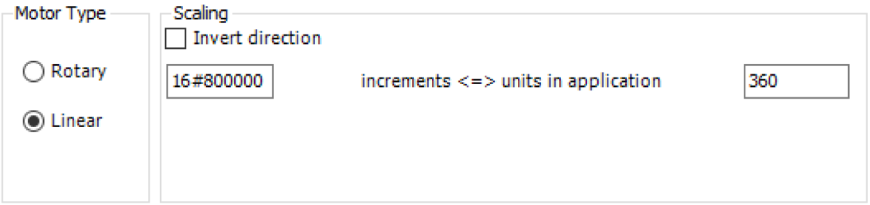
□ Invert direction

16#800000	increments <=> motor turns	1
1	motor turns <=> gear output turns	1
1	gear output turns <=> units in application	360

  Invert direction: The direction is reversed.

- Linear type

  When the axis type is set to "Finite", the ratio in conversion from drive increments to the application unit is set.



Motor Type

○ Rotary

● Linear

Scaling

□ Invert direction

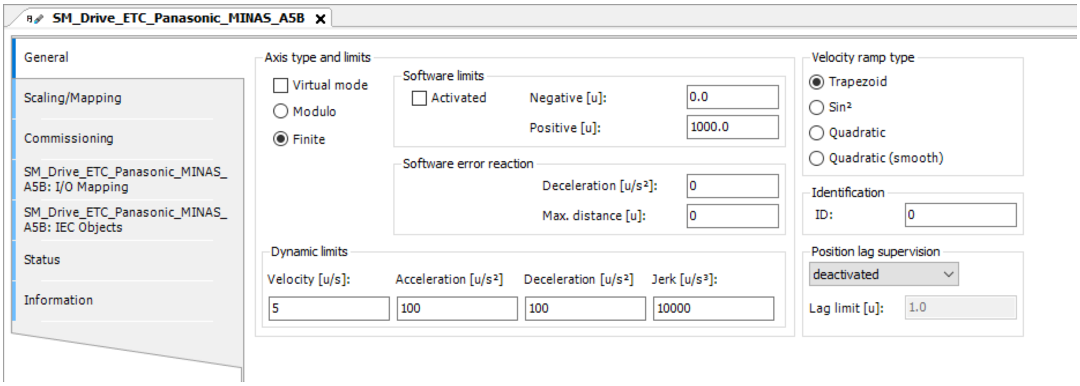
16#800000 increments <=> units in application 360

Invert direction: The direction is reversed.

11.3.3 EtherCAT Axis Extended Setting

Configure extended settings as required.

Right-click the object in the navigator pane and then select"Edit Object"from the context-sensitive menu that is displayed. The"EtherCAT Axis Setting"dialog box will be displayed.



SM_Drive_ETC_Panasonic_MINAS_A5B

General | Scaling/Mapping | Commissioning | SM_Drive_ETC_Panasonic_MINAS_A5B: I/O Mapping | SM_Drive_ETC_Panasonic_MINAS_A5B: IEC Objects | Status | Information

Axis type and limits

□ Virtual mode

○ Modulo

● Finite

Software limits

□ Activated

Negative [u]: 0.0

Positive [u]: 1000.0

Software error reaction

Deceleration [u/s²]: 0

Max. distance [u]: 0

Dynamic limits

Velocity [u/s]: 5

Acceleration [u/s²]: 100

Deceleration [u/s²]: 100

Jerk [u/s³]: 10000

Velocity ramp type

● Trapezoid

○ Sin²

○ Quadratic

○ Quadratic (smooth)

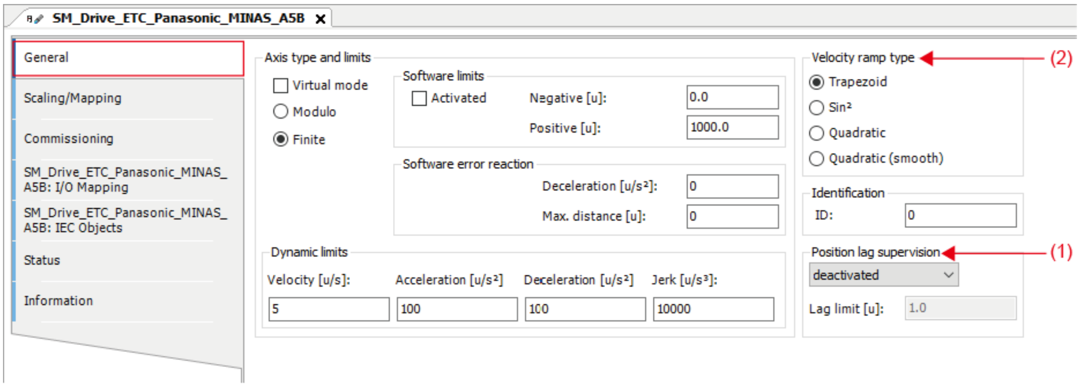
Identification

ID: 0

Position lag supervision: deactivated

Lag limit [u]: 1.0

Select the"General"tab and set the following items.



SM_Drive_ETC_Panasonic_MINAS_A5B

General | Scaling/Mapping | Commissioning | SM_Drive_ETC_Panasonic_MINAS_A5B: I/O Mapping | SM_Drive_ETC_Panasonic_MINAS_A5B: IEC Objects | Status | Information

Axis type and limits

□ Virtual mode

○ Modulo

● Finite

Software limits

□ Activated

Negative [u]: 0.0

Positive [u]: 1000.0

Software error reaction

Deceleration [u/s²]: 0

Max. distance [u]: 0

Dynamic limits

Velocity [u/s]: 5

Acceleration [u/s²]: 100

Deceleration [u/s²]: 100

Jerk [u/s³]: 10000

Velocity ramp type (2)

● Trapezoid

○ Sin²

○ Quadratic

○ Quadratic (smooth)

Identification

ID: 0

Position lag supervision (1): deactivated

Lag limit [u]: 1.0

(1) Position lag supervision

Axis operations automatically stop when the difference (absolute value) between the command position of the axis and the actual position has exceeded the value set in Lag limit. The following stop methods are available.

You can enter a positive real number greater than or equal to 0 in Lag limit. Negative values cannot be entered.

The axis state is Errorstop after the operation stops with an error. Execute MC_Reset to recover from the error.

Stop methods	Details
deactivated	The position lag supervision function is disabled. (Default)
disable drive	Power is turned Off. In the program, the bRegulatorOn parameter of the axis structure (AXIS_REF_SM3) is forcibly switched to OFF.
do quick stop	The axis operation comes to a quick stop. Do not use.
stay enabled	The axis is brought to a stop according to the settings in Software error reaction. The axis remains powered On.

(2) Velocity ramp type

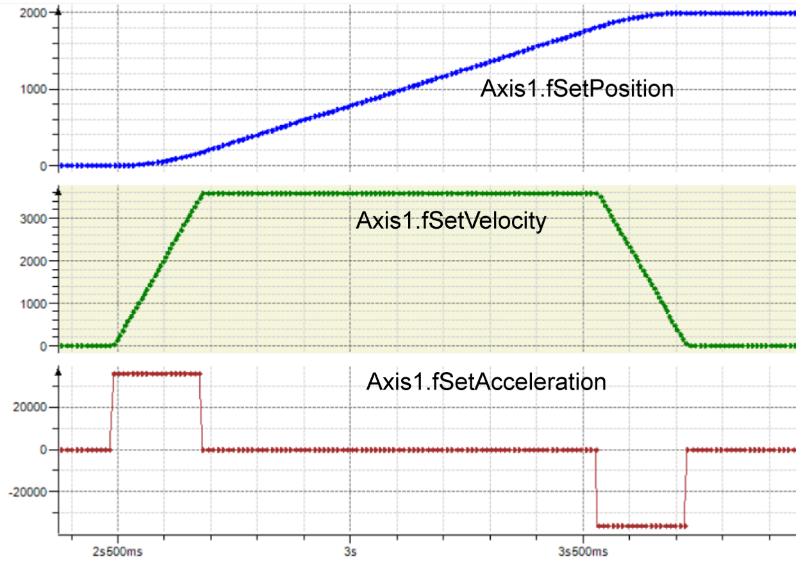
Velocity profiles are defined for each axis.

- Trapezoid

  In the trapezoidal velocity profile, velocity continues linearly.

  Therefore, acceleration can rise sharply.

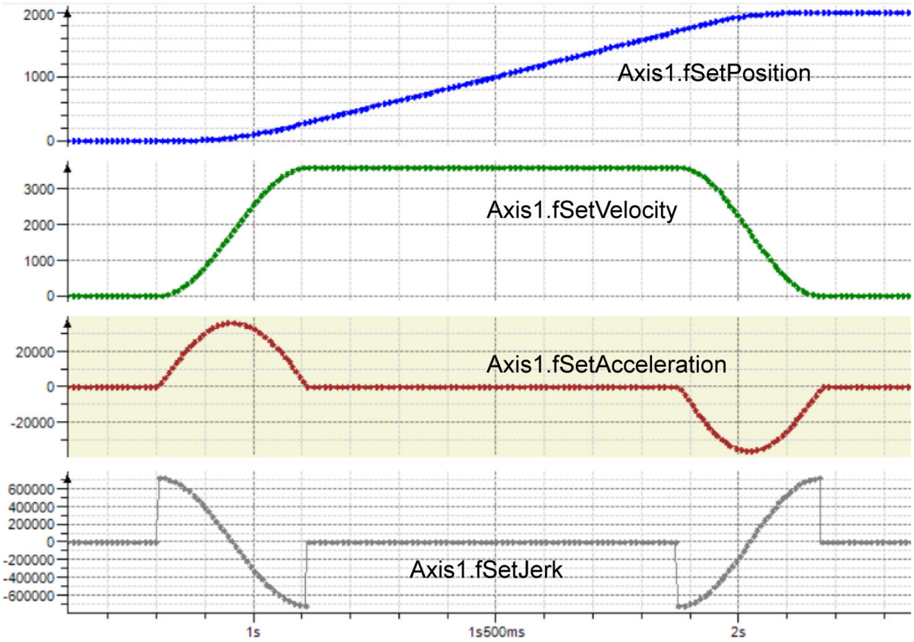
  In this mode, jerk restriction does not work in each function block.



- Sin^2

  In the velocity profile defined with the Sin^{2}function, transition motion within each section of the velocity profile is smooth and acceleration rises less sharply.

  In this mode, jerk restriction does not work in each function block.

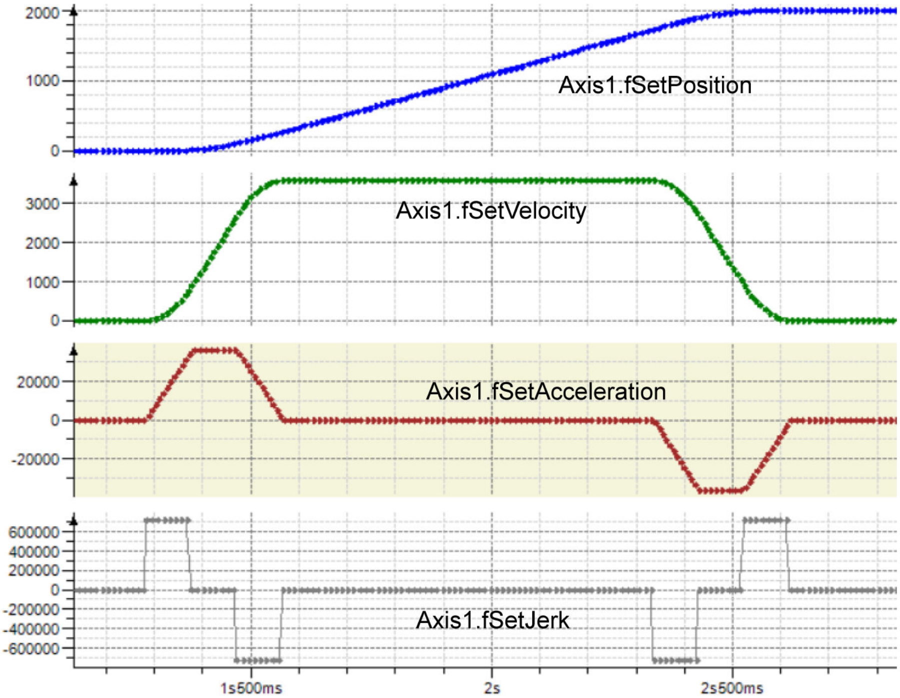


- Quadratic

  This is a trapezoidal acceleration profile with jerk restriction.

  Acceleration changes partially linearly and continuously and jerk rises sharply.

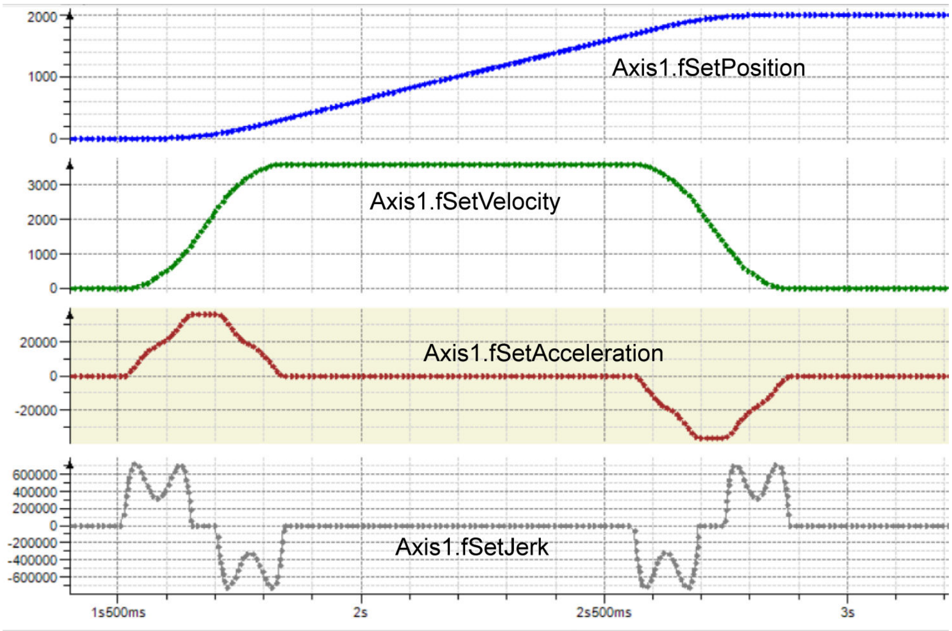
  The maximum value of jerk can be limited by jerk values in each function block.



- Quadratic (Smooth)

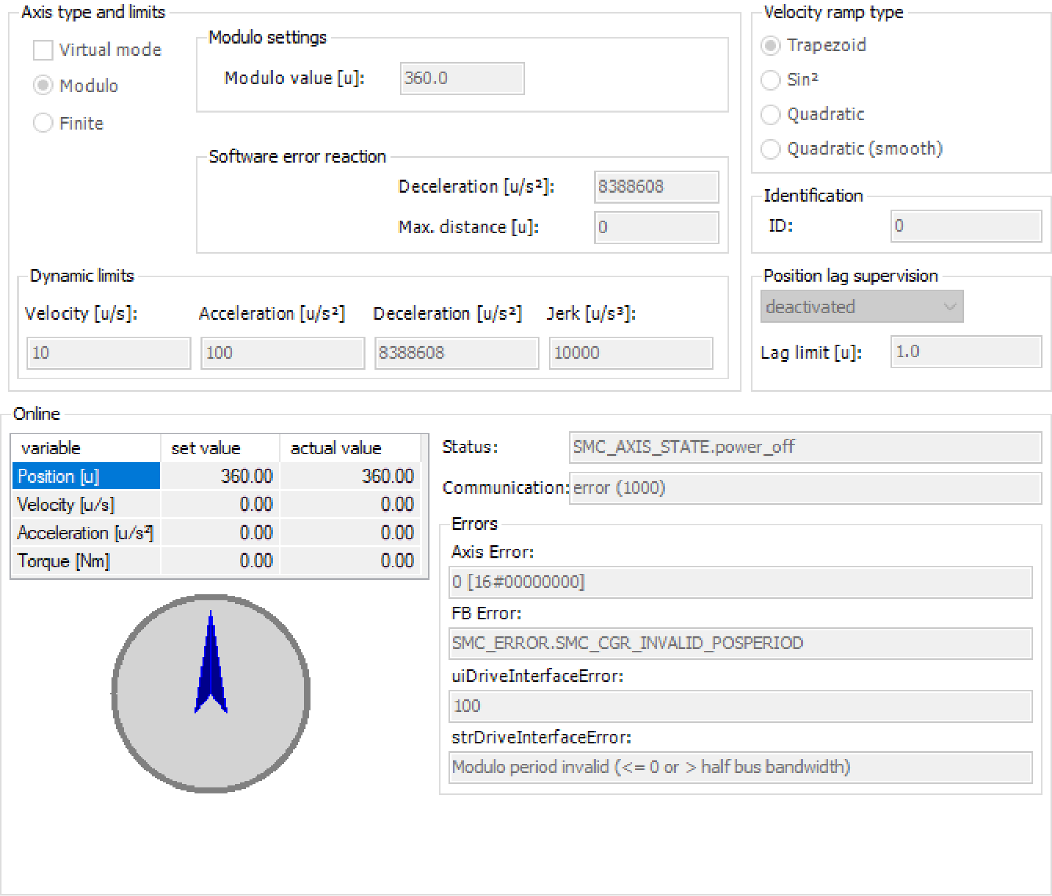
  This is similar to "Quadratic", except that jerk does not rise sharply.

The maximum value of jerk can be limited by jerk values in each function block.



(3) Axis state window in online mode

When "Modulo" is selected



Axis type and limits

□ Virtual mode

● Modulo

○ Finite

Modulo settings

Modulo value [u]: 360.0

Software error reaction

Deceleration [u/s^2]: 8388608

Max. distance [u]: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s^2]	Deceleration [u/s^2]	Jerk [u/s^3]:
10	100	8388608	10000

Velocity ramp type

● Trapezoid

○ Sin^2

○ Quadratic

○ Quadratic (smooth)

Identification

ID: 0

Position lag supervision

deactivated

Lag limit [u]: 1.0

Online

variable	set value	actual value
Position [u]	360.00	360.00
Velocity [u/s]	0.00	0.00
Acceleration [u/s^2]	0.00	0.00
Torque [Nm]	0.00	0.00

Status: SMC_AXIS_STATE.power_off

Communication: error (1000)

Errors

Axis Error:

0 [16#00000000]

FB Error:

SMC_ERROR.SMC_CGR_INVALID_POSPERIOD

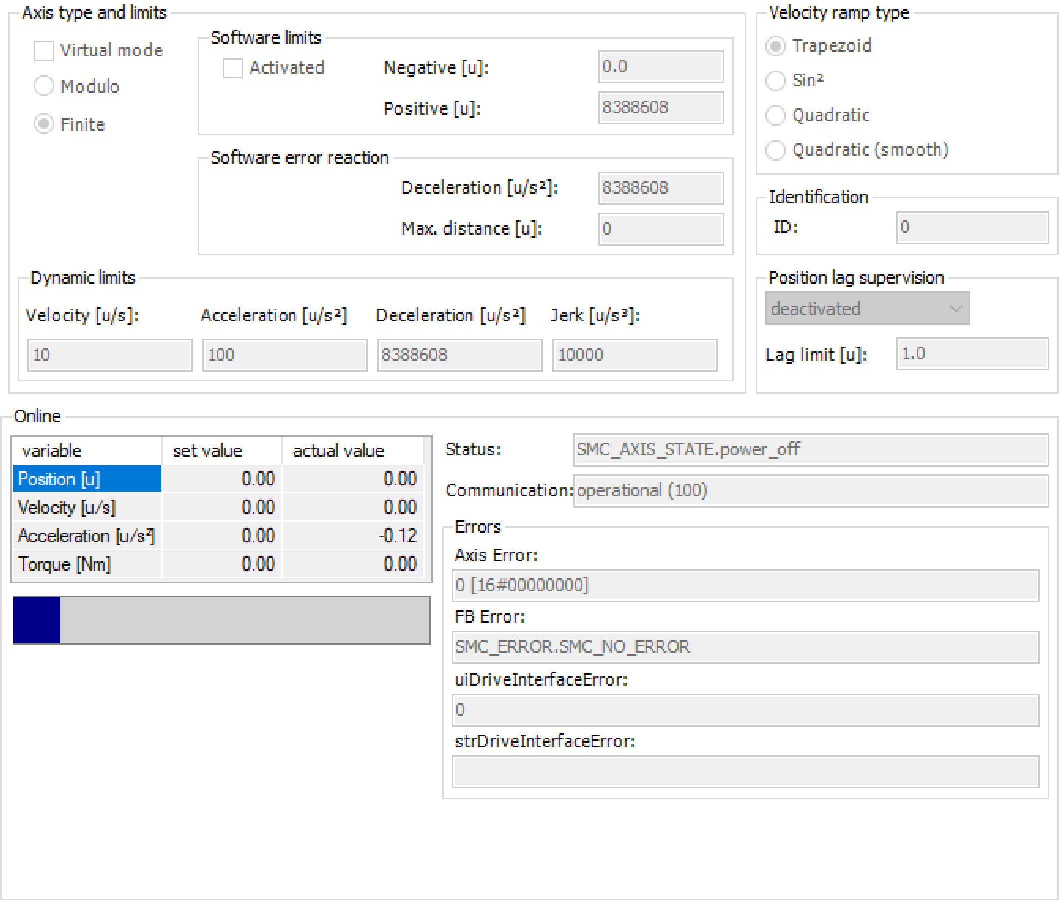
uiDriveInterfaceError:

100

strDriveInterfaceError:

Modulo period invalid (<= 0 or > half bus bandwidth)

When "Finite" is selected



Axis type and limits

□ Virtual mode

○ Modulo

● Finite

Software limits

□ Activated

Negative [u]: 0.0

Positive [u]: 8388608

Software error reaction

Deceleration [u/s^2]: 8388608

Max. distance [u]: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s^2]	Deceleration [u/s^2]	Jerk [u/s^3]:
10	100	8388608	10000

Velocity ramp type

● Trapezoid

○ Sin^2

○ Quadratic

○ Quadratic (smooth)

Identification

ID: 0

Position lag supervision

deactivated

Lag limit [u]: 1.0

Online

variable	set value	actual value
Position [u]	0.00	0.00
Velocity [u/s]	0.00	0.00
Acceleration [u/s^2]	0.00	-0.12
Torque [Nm]	0.00	0.00

Status: SMC_AXIS_STATE.power_off

Communication: operational (100)

Errors

Axis Error: 0 [16#00000000]

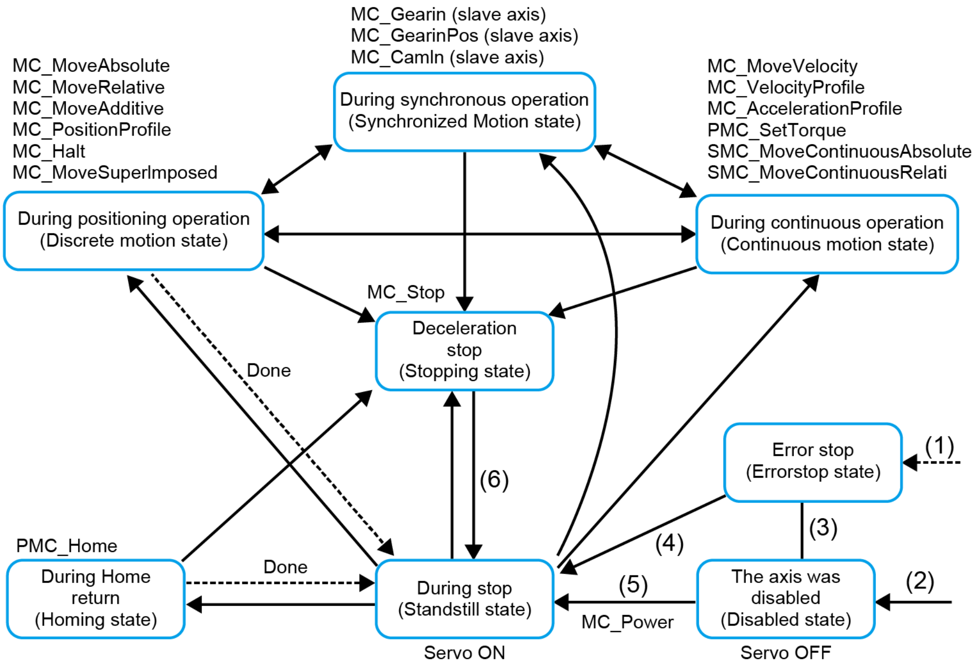
FB Error: SMC_ERROR.SMC_NO_ERROR

uiDriveInterfaceError: 0

strDriveInterfaceError:

(4) "Online" - "State"

This indicates one of the axis states in the following figure.



(5) "Online" - "Communication"

This indicates one of the following communication states.

Stopped
Variable initialized
Basic communication initialized
Drive initialized
Drive synchronization on standby
Initialization done
Operational
Re-initialized
Error
Unknown

(6) "Online" - "Error"

- FB error

  The oldest error that occurred on the axis, "SMC_ERROR", is displayed.

  This is the same error as the one that can be obtained by the "SMC_ReadFBError" function block.

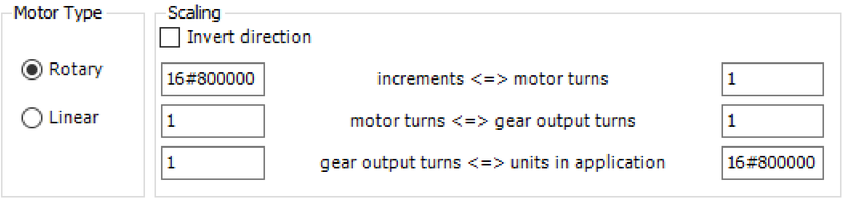
- uiDriveInterfaceError / strDriveInterfaceError

  This is an internal error in the GM1 series.

(7) Scaling / Mapping

- Rotary type

  When the axis type is set to "Modulo", the ratio in the conversion from the drive increment to the application unit is set.



Motor Type

● Rotary

○ Linear

Scaling

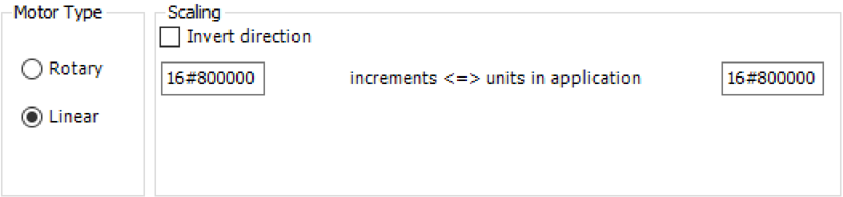
□ Invert direction

16#800000	increments <=> motor turns	1
1	motor turns <=> gear output turns	1
1	gear output turns <=> units in application	16#800000

Note: Invert direction: The direction is inverted.

- Linear type

When the axis type is set to "Finite", the ratio in the conversion from the drive increment to the application unit is set.



Motor Type

○ Rotary

● Linear

Scaling

□ Invert direction

16#800000	increments <=> units in application	16#800000

Note: Invert direction: The direction is inverted.

11.4 Operation up to running

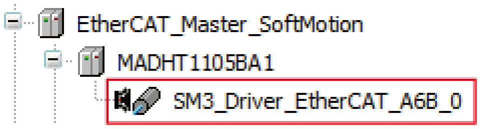
This section explains how to run and stop the motor.

11.4.1 Servo ON or OFF

To turn ON or OFF the servo motor, use the "MC_Power" function block.

■ Explanation of function block

- For input "Axis", specify the axis corresponding to the servo motor.
  Example) In the case of SM3_Driver_EtherCAT_A6B_0, substitute "SM3_Driver_EtherCAT_A6B_0" for “Axis” of MC_Power.



EtherCAT_Master_SoftMotion
MADHT1105BA1
SM3_Driver_EtherCAT_A6B_0

- When inputs "Enable", "bRegulatorOn", and "bDriveStart" are set to TRUE, the servo turns ON.
  When input "bRegulatorOn" is set to FALSE, the servo turns OFF.

■ Program examples

The following are LD program and ST program examples that execute the "MC_Power" function block.
Setting variable "SW_Power" to TRUE turns ON the servo and setting it to FALSE turns OFF the servo.

ST program

- Declaration section

```
PROGRAM ST_POU
VAR
    MC_Power_0: MC_Power;
    SW_power : BOOL := FALSE;
    PStatus  : BOOL := FALSE;
END_VAR
```

- Implementation section

```
MC_Power_0(
    Axis := Axisl ,
    Enable := TRUE ,
    bRegulatorOn := SW_power ,
    bDriveStart := TRUE ,
    Status => PStatus ,
    bRegulatorRealState => ,
    bDriveStartRealState => ,
    Busy => ,
    Error => ,
    ErrorID =>
    );
```

i Info.

- When executing "MC_Power", confirm in advance that communication has been established. The communication state can be checked using the "SMC_CheckAxisCommunication" function block.

11.4.2 Home Return

Home return is an operation that returns the motor to its home position.

Home return can be achieved by executing the "MC_Home" function block.

Info.

- Home Return,refer to the Manuals of each manufacturer.
- MC_Home,refer to the*GM1 Series Reference Manual (Instruction)*.

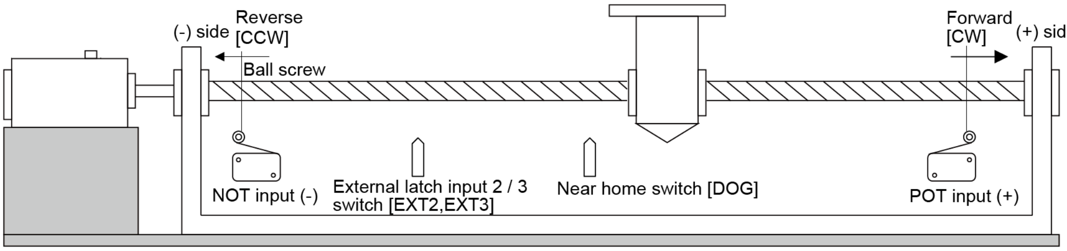
Types of home return

Home return is a function that moves the axis to the preset reference position (home position) and set the coordinates of the position to 0.

If an incremental encoder is used for the servomotor, the home return methods shown in the table below can be selected.

Info.

- The GM1 controller also supports home return using an absolute encoder. When implementing home return using an absolute encoder, use MINAS V1.06 or later.
- For details, refer to the users manual(Servo amplifier)



Settings and operations of home return	Behavior overview
Edge detection of near home switch + Home position (Z phase) based on front edge	After the rising edge (front edge) of the near home switch (DOG) is detected, the rising edge of the first home position (Z phase) is detected and the motor stops. The stopping position is set as the home position.
Edge detection of near home switch	The rising edge of the near home switch (DOG) is detected and the motor stops. The stopping position is set as the home position.
Edge detection of near home switch + Home position (Z phase) based on rear edge	After the falling edge (rear edge) of the near home switch (DOG) is detected, the rising edge of the first home position (Z phase) in the home return direction is detected and the motor stops. The stopping position is set as the home position.
Edge detection of limit switch + Home position (Z phase) based on front edge	After the rising edge of the limit switch on the opposite side of the home return direction is detected, the rotation of the motor is reversed. Then, the rising edge of the first home position (Z phase) is detected and the motor stops. The stopping position is set as the home position.

Settings and operations of home return	Behavior overview
Edge detection of limit switch	The rising edge of the limit switch in the home return direction is detected and the motor stops. The stopping position is set as the home position.
Edge detection of home position (Z phase)	The axis moves from the current value toward the direction of home return. Then, the rising edge of the first home position (Z phase) is detected and the motor stops. The stopping position is set as the home position.
Data set method	The current value is set as the home position.

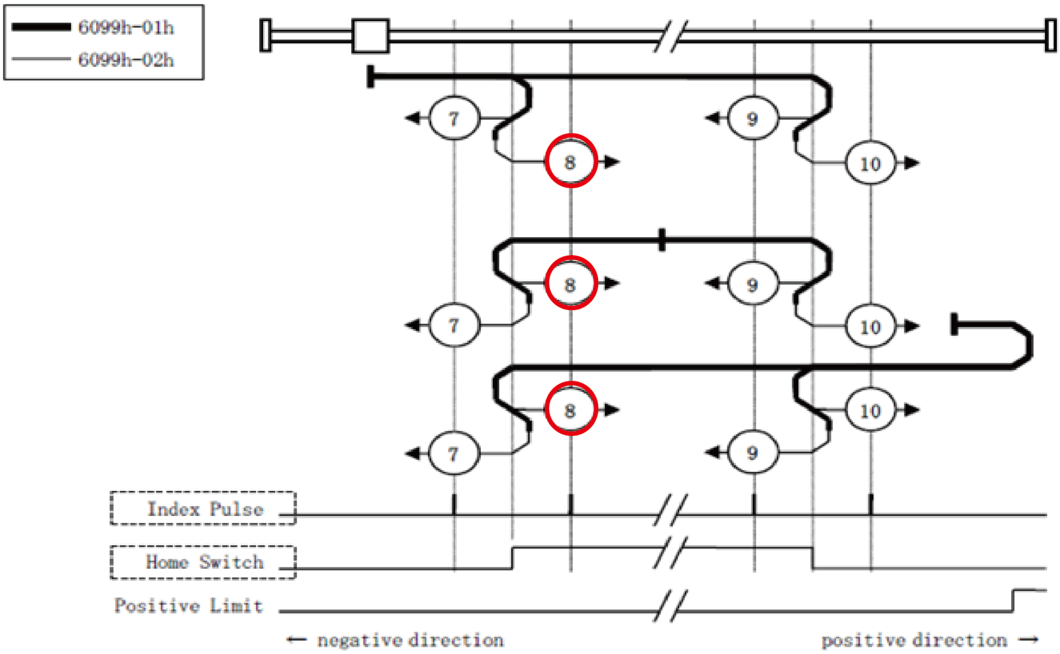
Edge detection of near home switch + Home position (Z phase) based on front edge

After the rising edge (front edge) of the near home switch (DOG) is detected, the rising edge of the first home position (Z phase) is detected and the motor stops. The stopping position is set as the home position.

(Note 1) If the home position (Z phase) is ON at the time of startup, it will not be regarded as a home position (Z phase). Searches for a near home switch (DOG) will be started.

■ Method 8 (Home return direction: positive direction)

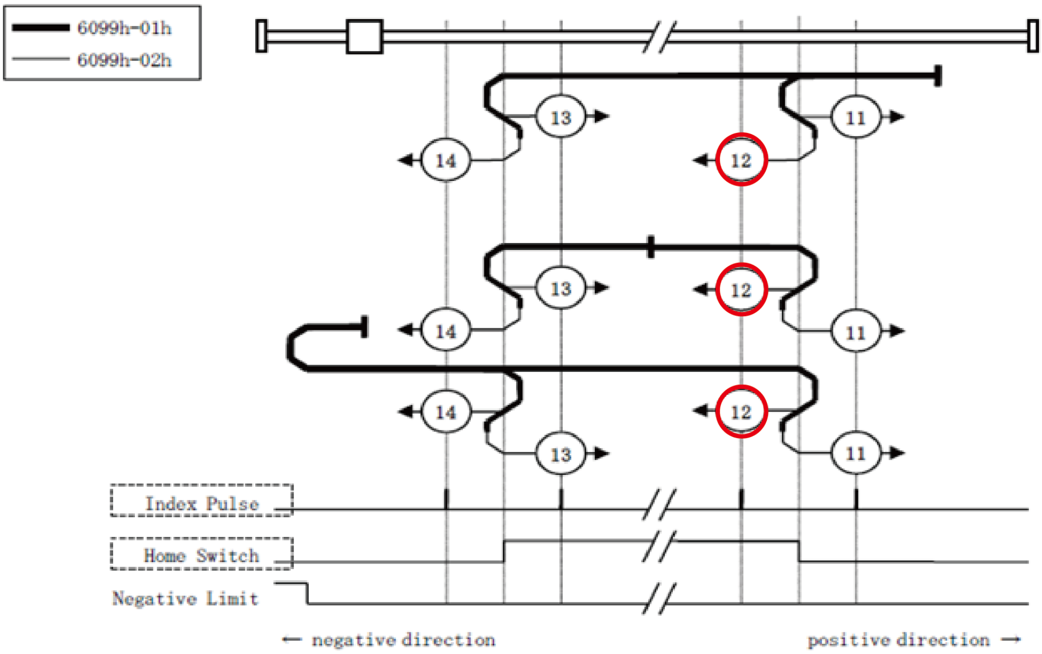
- This method uses the home switch (HOME) and the index pulse (Z phase).
- The initial motion direction is a negative direction when the home switch (HOME) is active at the start of operation.
- The home detection position is the index pulse (Z phase) near the rising edge or the falling edge of the home switch (HOME). (Refer to the following figure.)
- If HOME and NOT are not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



Homing on home switch and index pulse - positive initial motion

■ Method 12 (Home return direction: negative direction)

- This method uses the home switch (HOME) and the index pulse (Z phase).
- The initial motion direction is a positive direction when the home switch (HOME) is active at the start of operation.
- The home detection position is the index pulse (Z phase) near the rising edge or the falling edge of the home switch (HOME). (Refer to the following figure.)
- If HOME and NOT are not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



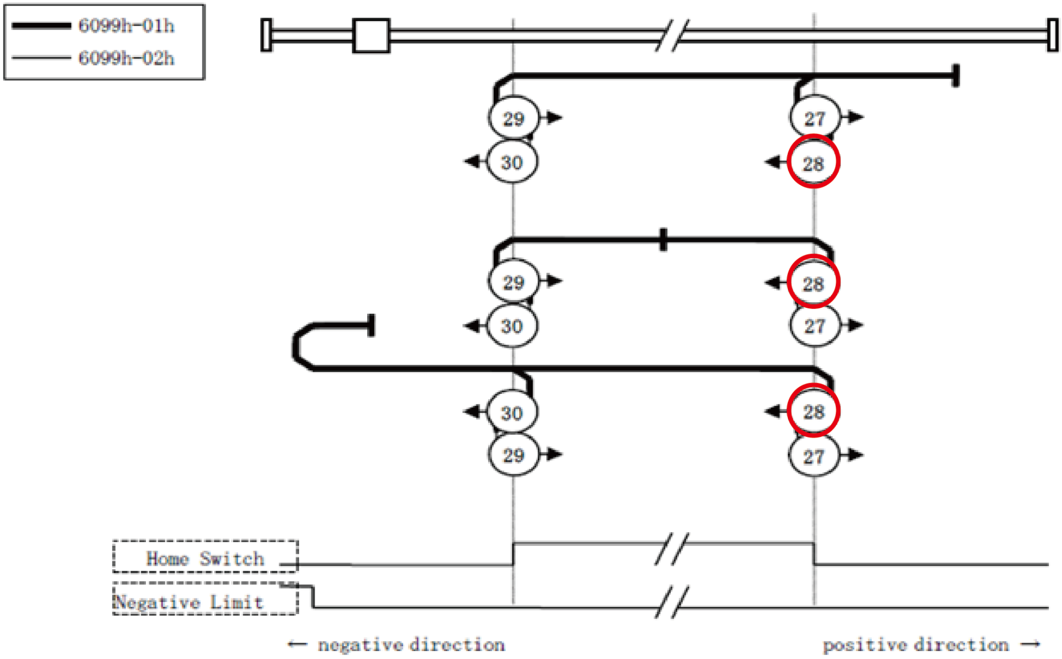
Homing on home switch and index pulse - negative initial motion

Edge detection of near home switch

The rising edge of the near home switch (DOG) is detected and the motor stops. The stopping position is set as the home position.

■ Method 28 (Home return direction: negative direction)

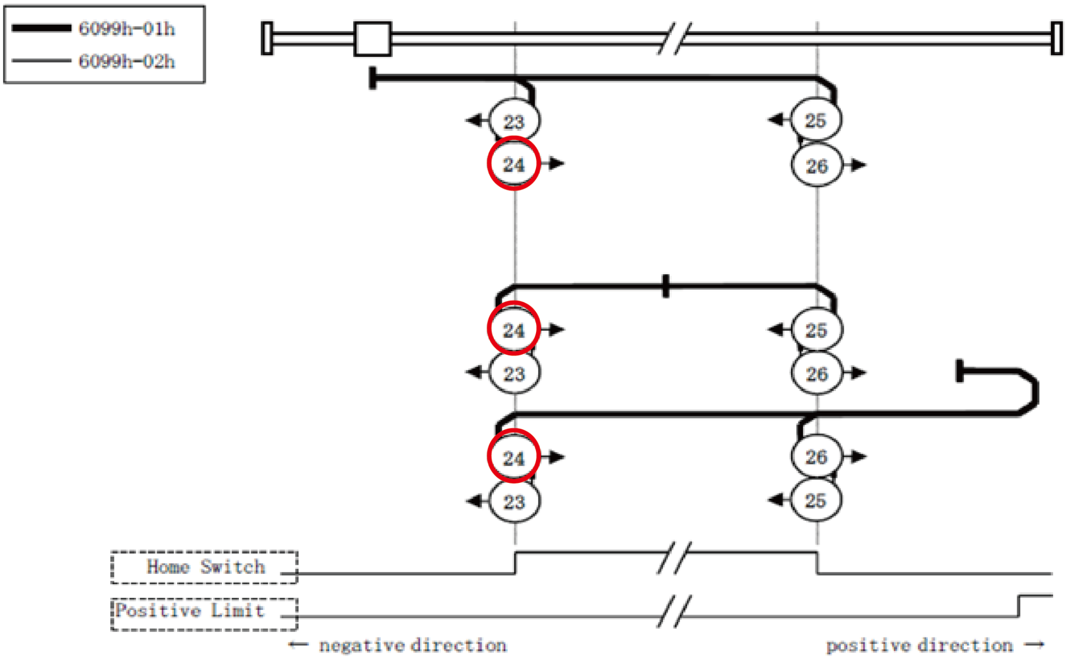
- This method uses the home switch (HOME).
- The initial motion direction is a positive direction when the home switch (HOME) is active at the start of operation.
- The home detection position is the position where the home switch (HOME) changes its state. (Refer to the following figure.)
- If HOME and NOT are not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



Homing on home switch and index pulse - negative initial motion

■ Method 24 (Home return direction: positive direction)

- This method uses the home switch (HOME).
- The initial motion direction is a negative direction when the home switch (HOME) is active at the start of operation.
- The home detection position is the position where the home switch (HOME) changes its state. (Refer to the following figure.)
- If HOME and NOT are not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



Homing on home switch and index pulse - positive initial motion

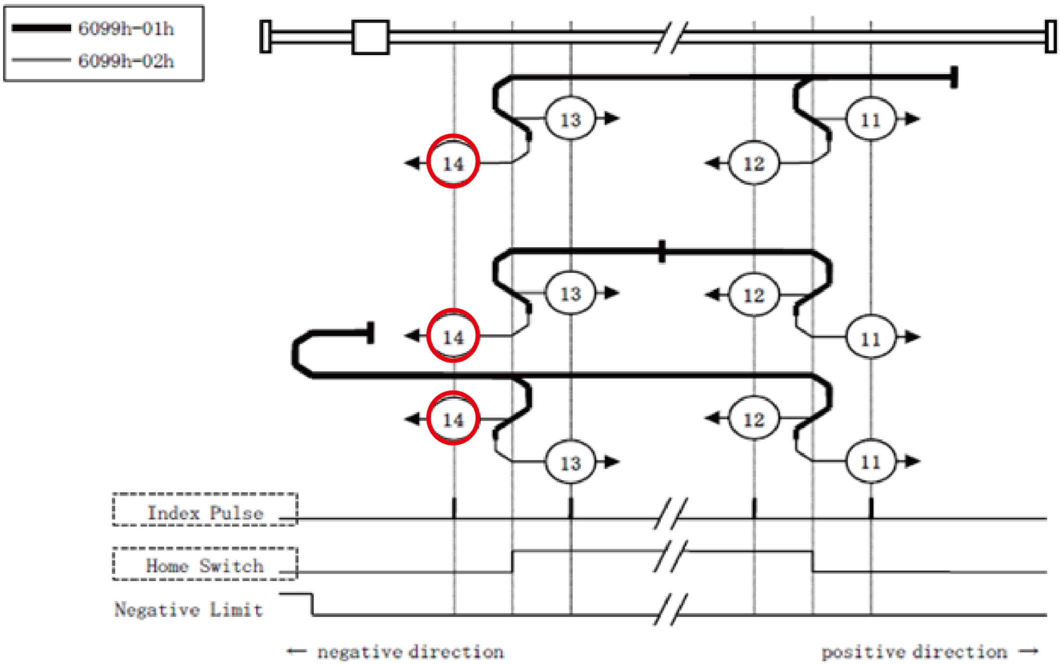
Edge detection of near home switch + Home position (Z phase) based on rear edge

After the falling edge (rear edge) of the near home switch (DOG) is detected, the rising edge of the first home position (Z phase) in the home return direction is detected and the motor stops. The stopping position is set as the home position.

(Note 1) If the home position (Z phase) is ON at the time of startup, it will not be regarded as a home position (Z phase). Searches for a near home switch (DOG) will be started.

■ Method 14 (Home return direction: negative direction)

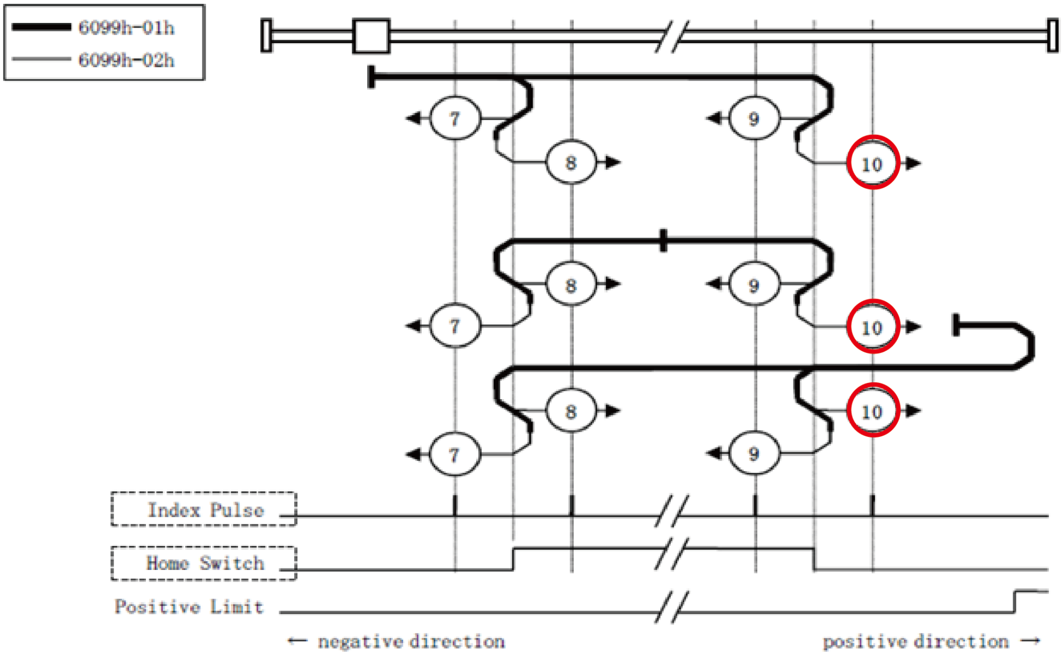
- This method uses the home switch (HOME) and the index pulse (Z phase).
- The initial motion direction is a positive direction when the home switch (HOME) is active at the start of operation.
- The home detection position is the index pulse (Z phase) near the rising edge or the falling edge of the home switch (HOME). (Refer to the following figure.)
- If HOME and NOT are not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



Homing on home switch and index pulse - negative initial motion

■ Method 10 (Home return direction: positive direction)

- This method uses the home switch (HOME) and the index pulse (Z phase).
- The initial motion direction is a negative direction when the home switch (HOME) is active at the start of operation.
- The home detection position is the index pulse (Z phase) near the rising edge or the falling edge of the home switch (HOME). (Refer to the following figure.)
- If HOME and NOT are not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



Homing on home switch and index pulse - positive initial motion

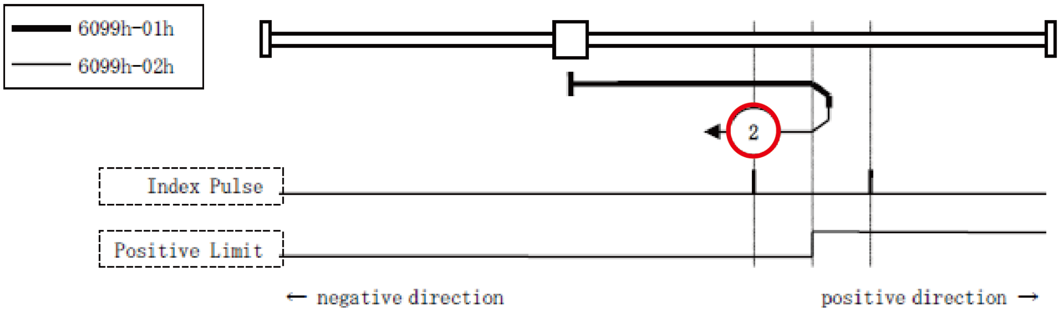
Edge detection of limit switch + Home position (Z phase) based on front edge

After the rising edge of the limit switch on the opposite side of the home return direction is detected, the rotation of the motor is reversed. Then, the rising edge of the first home position (Z phase) is detected and the motor stops. The stopping position is set as the home position.

(Note 1) If the home position (Z phase) is ON at the time of startup, it will not be regarded as a home position (Z phase). Searches for a limit switch will be started.

■ Method 2 (Home return direction: negative direction)

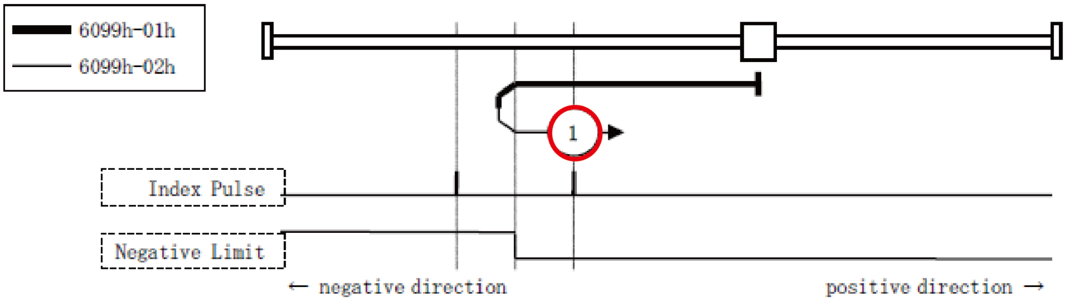
- In this method, the initial motion direction is a positive direction when the positive limit switch (POT) is inactive. (The inactive state is indicated as a low level state in the figure.)
- The home detection position is the first index pulse (Z phase) detection position on the negative side after the positive limit signal (POT) turns inactive. (Refer to the following figure.)
- If POT is not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



Homing on positive limit switch and index pulse

■ Method 1 (Home return direction: positive direction)

- In this method, the initial motion direction is a negative direction when the negative limit switch (NOT) is inactive. (The inactive state is indicated as a low level state in the figure.)
- The home detection position is the first index pulse (Z phase) detection position on the positive side after the negative limit signal (NOT) turns inactive.
- If NOT is not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



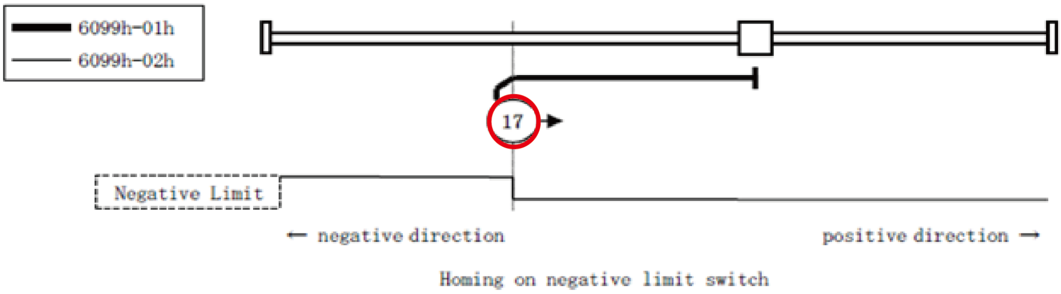
Homing on negative limit switch and index pulse

Edge detection of limit switch

The rising edge of the limit switch in the home return direction is detected and the motor stops. The stopping position is set as the home position.

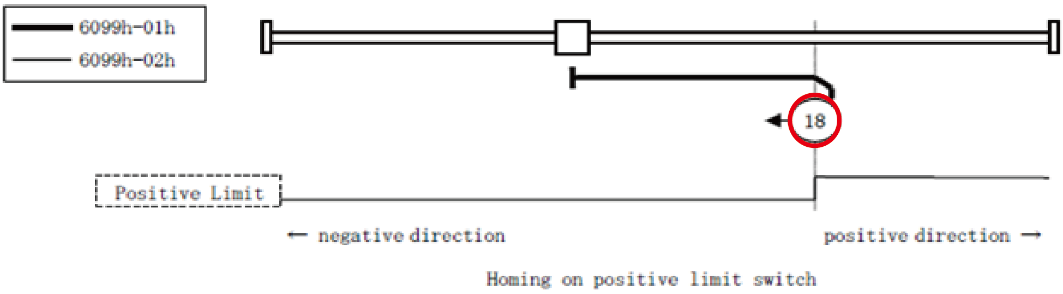
■ Method 17 (Home return direction: positive direction)

- In this method, the initial motion direction is a negative direction when the negative limit switch (NOT) is inactive. (The inactive state is indicated as a low level state in the figure.)
- The home detection position is the position where the negative limit switch (NOT) changes its state.
- If NOT is not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.



■ Method 18 (Home return direction: negative direction)

- In this method, the initial motion direction is a positive direction when the positive limit switch (POT) is inactive. (The inactive state is indicated as a low level state in the figure.)
- The home detection position is the position where the positive limit switch (POT) changes its state.
- If NOT is not assigned, an error occurs in the MC_Home function block. For details on signal assignment, refer to relevant manuals for the MINAS series.

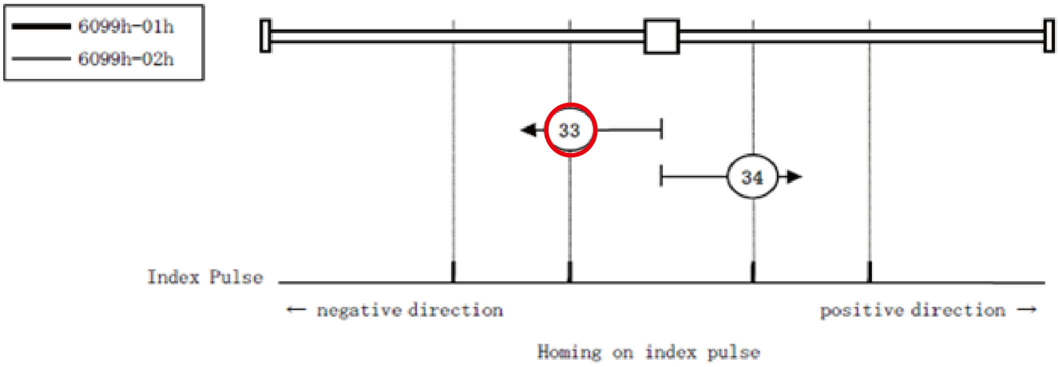


Edge detection of home position (Z phase)

The axis moves from the current value toward the direction of home return. Then, the rising edge of the first home position (Z phase) is detected and the motor stops. The stopping position is set as the home position.

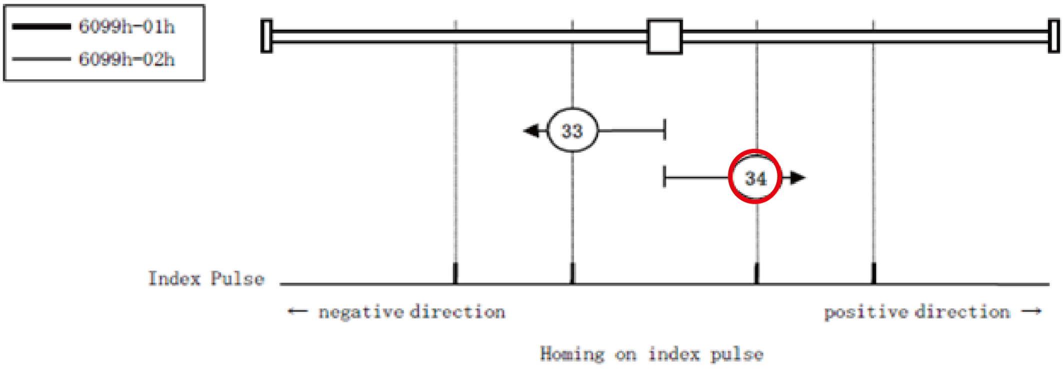
■ Method 33 (Home return direction: negative direction)

- This method uses only the index pulse (Z phase).
- The index pulse (Z phase) that is detected after operating in the direction shown in the figure is regarded as the home detection position.
- For detailed specifications, refer to relevant manuals for the MINAS series.



■ Method 34 (Home return direction: positive direction)

- This method uses only the index pulse (Z phase).
- The index pulse (Z phase) that is detected after operating in the direction shown in the figure is regarded as the home detection position.
- For detailed specifications, refer to relevant manuals for the MINAS series.



Data set method

The current position is set as the home position.

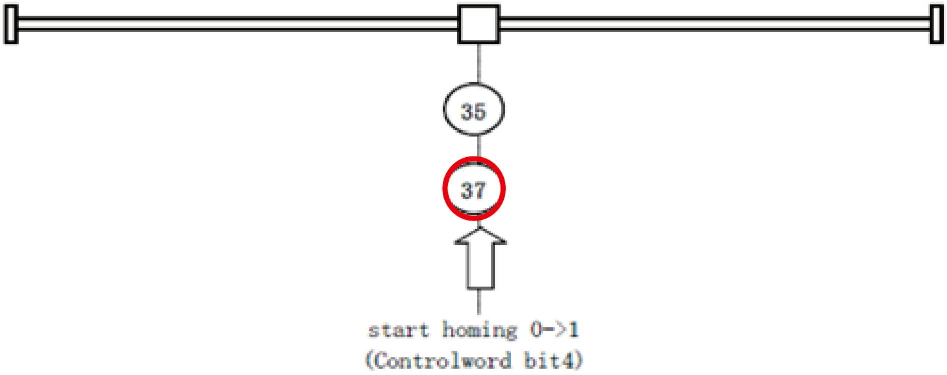
■ Method 37

- This method is used when setting the coordinate system (position information setting) of the servo amplifier.

  The following objects are initialized (preset) based on the position where MC_Home is executed.

  6062h (Position demand value) = 6064h (Position actual value) = 607Ch (Home offset)

  6063h (Position actual internal value) = 60FCh (Position demand internal value) = 0
- For detailed specifications, refer to relevant manuals for the MINAS series.



Settings and operations of home return

■ Settings in GM Programmer

To use the home return function, set servo amplifiers of each manufacturer using the ETC_CO_SdoWrite function block and execute "MC_Home".

Specify the following settings according to the home return function to be used. The following is an example of setting the MINAS A6B.

Index	Sub-Index	Name	Description
6098h	00h	Homing method	Home return method
6099h	01h	Speed during search for switch	Home return speed
6099h	02h	Speed during search for zero	Home return creep speed
609Ah	00h	Homing acceleration	Home return acceleration / deceleration

11.5 Single-axis Operation

This section explains single-axis operations using function blocks.

11.5.1 Overview of Single-axis Operation

In single-axis operation, it supports Home Return,Switching the Control Mode,Stop and JOG / Inching.

Single-axis Operation type	Overview	reference
Ssevo ON	Turn Servo ON and make it ready for operation.	"11.4.1 Servo ON or OFF"
Home Return	Home return of the axis can be performed.	EtherCAT:"11.4.2 Home Return"
Switching the Control Mode	Switching between position, control, and torque control can be performed.	"11.5.2 Switching the Control Mode"
Stop	Decelerates the axis to a stop.	*GM1 Series Reference Manual (Instruction).*
JOG / Inching	JOG: Moves forward or reverse at a constant speed. Inching: The axis moves forward or reverse by a specified relative distance.	

11.5.2 Switching the Control Mode

Supports position control, velocity control, and torque control as control modes.
Switching the Control Mode can be used while switching control according to the purpose.
The control mode can be switched by"SMC_SetControllerMode"of the function block.

■ Control Mode and Corresponding Function Blocks

Control mode	Value	Name	Description	Corresponding Function Blocks
SMC_torque	1	Torque control mode	Torque control is performed at the specified torque.	PMC_SetTorque, SMC_SetTorque
SMC_velocity	2	Velocity control mode	Controls the motor to move at a specified speed.	MC_MoveVelocity, MC_VelocityProfile, MC_AccelerationProfile
SMC_position	3	Position control mode (Default value)	Controls the motor to move at a specified position or distance.	MC_MoveAbsolute, MC_MoveRelative, MC_MoveAdditive, MC_MoveSuperImposed,MC_PositionProfile, SMC_MoveContinuousAbsolute, SMC_MoveContinuousRelative, MC_MoveVelocity, MC_VelocityProfile, MC_AccelerationProfile

Info.

- For details, refer to the *GM1 Series Reference Manual (Instruction).*

11.6 CNC Control

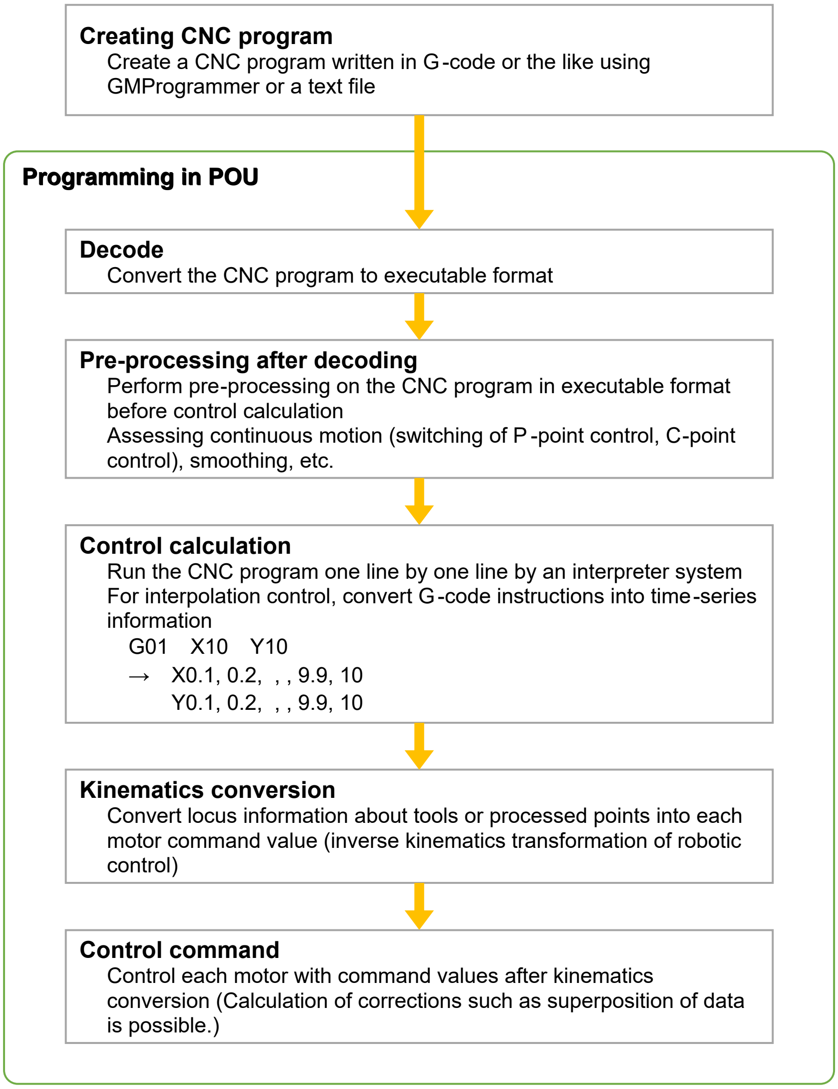
CNC, the acronym of computer numerical control, is control that has been generally used to automate primarily machine tools and improve work efficiency and quality. Information on tool tracks is generated from drawing data using computer-aided design (CAD) or computer-aided manufacturing (CAM) software to give machine control instructions to machine tools and implement continuous automated operation. A unified language called G-code is used in machine control instructions, and basic movements such as machine feeding (specified position, velocity, and acceleration) and fast-forwarding can be implemented irrespective of manufacturer.

CNC data containing G-code is general-purpose text information and thus is also applied to coating and dispensing processes as well as robot control that require locus control.

11.6.1 Overview of CNC Control and How to Use It

You must execute the following series of processes to perform CNC control with the GM1 controller.

- Create a CNC program written in G-code using the CNC editor or other tools.
- Decode the CNC program to executable format.
- Compute command data from the decoded CNC program at every cycle of motor control to control the motor.



Creating CNC program
Create a CNC program written in G-code or the like using GMProgrammer or a text file

Programming in POU

Decode
Convert the CNC program to executable format

Pre-processing after decoding
Perform pre-processing on the CNC program in executable format before control calculation
Assessing continuous motion (switching of P-point control, C-point control), smoothing, etc.

Control calculation
Run the CNC program one line by one line by an interpreter system
For interpolation control, convert G-code instructions into time-series information

```
G01   X10   Y10
→   X0.1, 0.2,  , , 9.9, 10
    Y0.1, 0.2,  , , 9.9, 10
```

Kinematics conversion
Convert locus information about tools or processed points into each motor command value (inverse kinematics transformation of robotic control)

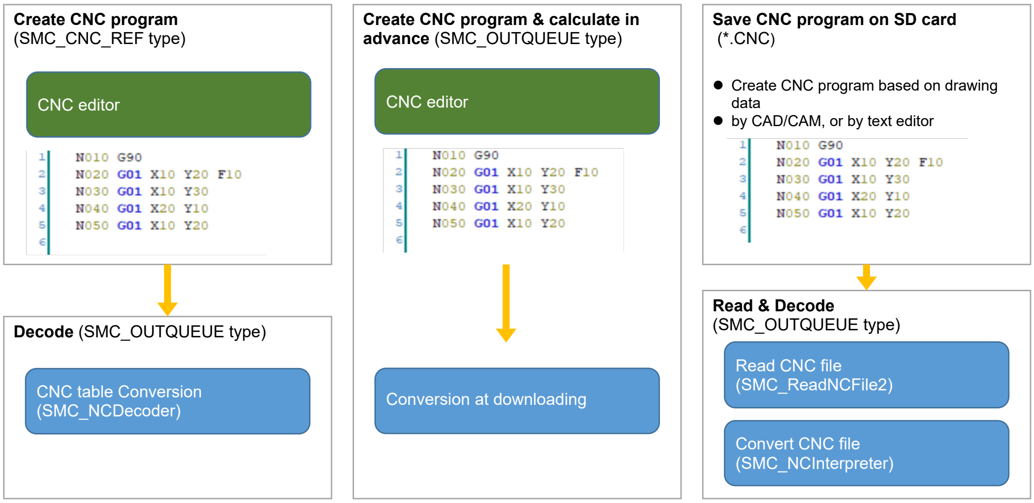
Control command
Control each motor with command values after kinematics conversion (Calculation of corrections such as superposition of data is possible.)

Procedure

1. Create a CNC program and decode it

   A CNC program (SMC_OUTQUEUE type) written in executable format is necessary to program CNC control using GM Programmer. The CNC program in executable format can be created by any of the following three methods.

   - After creating a CNC program (SMC_CNC_REF type) using the CNC editor, decoding the CNC program to that of the SMC_OUTQUEUE type by SMC_NCDecoder.
   - Creating a CNC program (SMC_OUTQUEUE type) using the CNC editor. The created CNC program is converted into executable format when it is downloaded.
   - Reading a CNC file (*.CNC), which is created by a text editor or CAD/CAM, via a SD card using SMC_ReadNCFile2 and decoding the CNC file to that of the SMC_OUTQUEUE type by SMC_NCInterpreter.


Create CNC program (SMC_CNC_REF type)
CNC editor
Decode (SMC_OUTQUEUE type)
CNC table Conversion (SMC_NCDecoder)
Create CNC program & calculate in advance (SMC_OUTQUEUE type)
CNC editor
Conversion at downloading
Save CNC program on SD card (*.CNC)
● Create CNC program based on drawing data
● by CAD/CAM, or by text editor
Read & Decode (SMC_OUTQUEUE type)
Read CNC file (SMC_ReadNCFile2)
Convert CNC file (SMC_NCInterpreter)

```
N010 G90
N020 G01 X10 Y20 F10
N030 G01 X10 Y30
N040 G01 X20 Y10
N050 G01 X10 Y20
```

```
N010 G90
N020 G01 X10 Y20 F10
N030 G01 X10 Y30
N040 G01 X20 Y10
N050 G01 X10 Y20
```

```
N010 G90
N020 G01 X10 Y20 F10
N030 G01 X10 Y30
N040 G01 X20 Y10
N050 G01 X10 Y20
```

It is necessary to set a compile mode according to the method of CNC program creation.

For details on how to create a CNC program using the CNC editor and set a compile mode for the CNC program, refer to "11.6.2 Creating a CNC Program".

2. Programming CNC control in POU

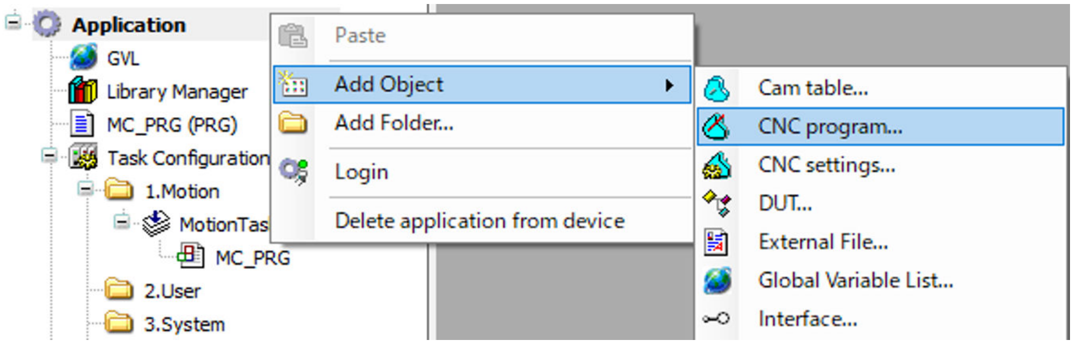
Program a series of the processes from decoding of the CNC program and motor control in POU.

For details about programming CNC control in POU, function blocks, and G-code, refer to the *GM1 Series Reference Manual (Instruction Edition)*.

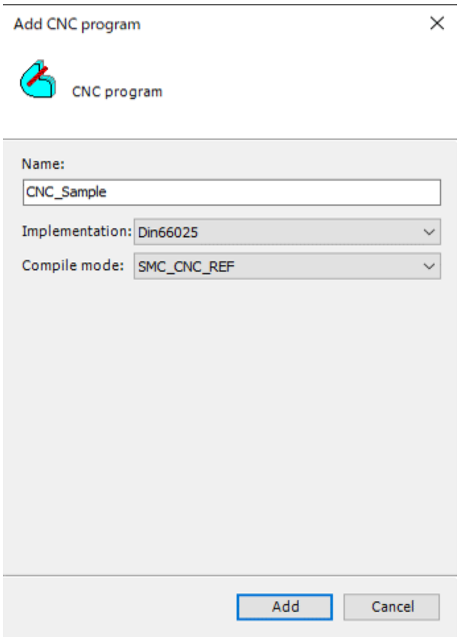
11.6.2 Creating a CNC Program

You can create a CNC program by writing it with GM Programmer or on a text editor. The description below explains a method using GM Programmer.

Right-click the"Application"object in the navigator pane and then select**Add Object**>**CNC program**from the context-sensitive menu that is displayed.



When "CNC program" is selected, the following window is displayed. Specify settings and click [Add].



Name: CNC_Sample

Implementation: Din66025

Compile mode: SMC_CNC_REF

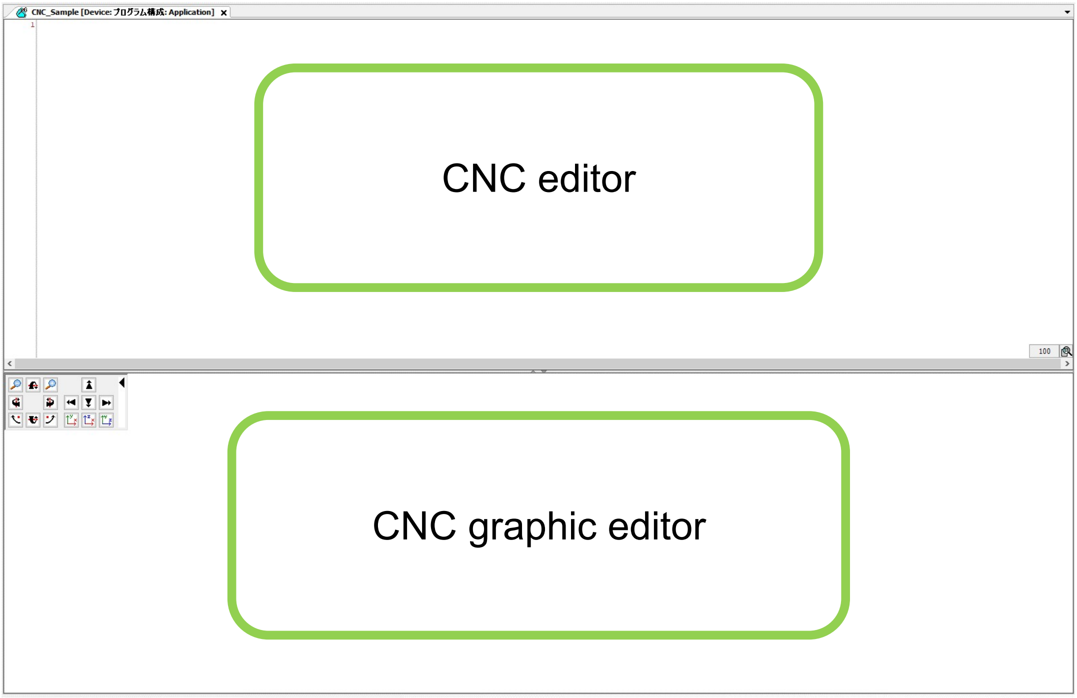
For CNC programs, three compile modes are available. You can select one at the time of creating a table.

Setting item		Description
Name	Settings	Specify a name for a CNC program to be created.
Imple mentat ion	Din66025	Your program will be written in text format. Be sure to use this setting.
	Table	Do not use.

Setting item		Description
Compile mode	SMC_CNC_REF	Normally, use this setting.
	SMC_OUTQUEUE	The movement speed improves due to the prior calculation method, but pre-processing cannot be performed, CNC program settings cannot be changed, and CNC programs cannot be joined together.
	File	CNC programs are generated as external reference files. Even files created by external text editors can similarly work.

When Add is clicked, CNC settings and a CNC program are created in the Device tree. The CNC program is displayed as the name registered earlier (CNC_Sample).

The CNC editor and CNC graphic editor are displayed in the right pane of the GM Programmer screen.



■ CNC editor

Write a CNC program.

For details on the method of writing, refer to the *GM1 Series Reference Manual (Instruction Edition)*.

■ CNC graphic editor

The movement loci coded in the CNC editor are graphically displayed in real time. You can also rotate the display and change the scale.

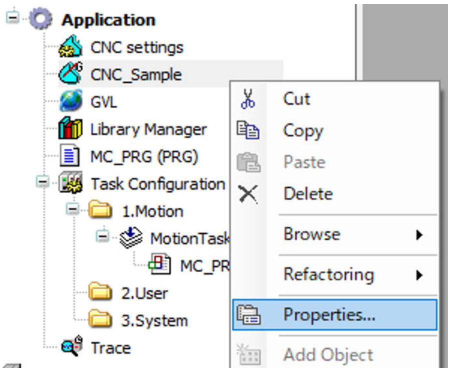
It is possible to make a change to the loci in the CNC graphic editor, but it should be noted that the change will also be reflected in the code in the CNC editor.

If variables are used in CNC programs, the CNC graphic editor cannot be displayed.

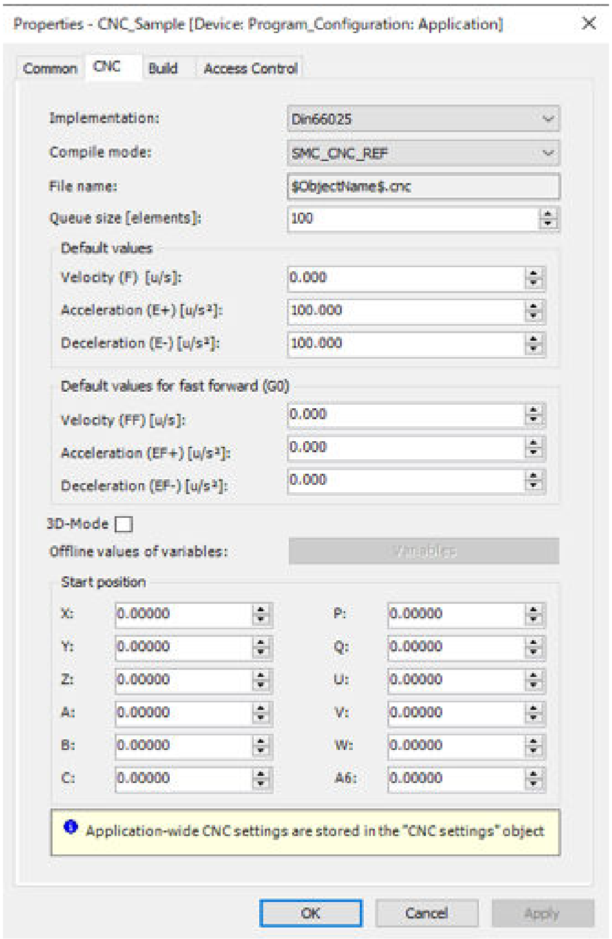
■ "CNC program" - "Properties"

In the Properties window for the CNC program, you can change the compile mode and configure parameter settings.

In the Device tree of GM Programmer, right-click **Application**>**CNC program**>**Properties** in this order.



- "CNC" tab


Properties - CNC_Sample [Device: Program_Configuration: Application]
Common
CNC
Build
Access Control

Implementation: Din66025

Compile mode: SMC_CNC_REF

File name: $ObjectName$.cnc

Queue size [elements]: 100

Default values

Velocity (F) [u/s]: 0.000

Acceleration (E+) [u/s²]: 100.000

Deceleration (E-) [u/s²]: 100.000

Default values for fast forward (G0)

Velocity (FF) [u/s]: 0.000

Acceleration (EF+) [u/s²]: 0.000

Deceleration (EF-) [u/s²]: 0.000

3D-Mode □

Offline values of variables: Variables

Start position

X:	0.00000	P:	0.00000
Y:	0.00000	Q:	0.00000
Z:	0.00000	U:	0.00000
A:	0.00000	V:	0.00000
B:	0.00000	W:	0.00000
C:	0.00000	A6:	0.00000

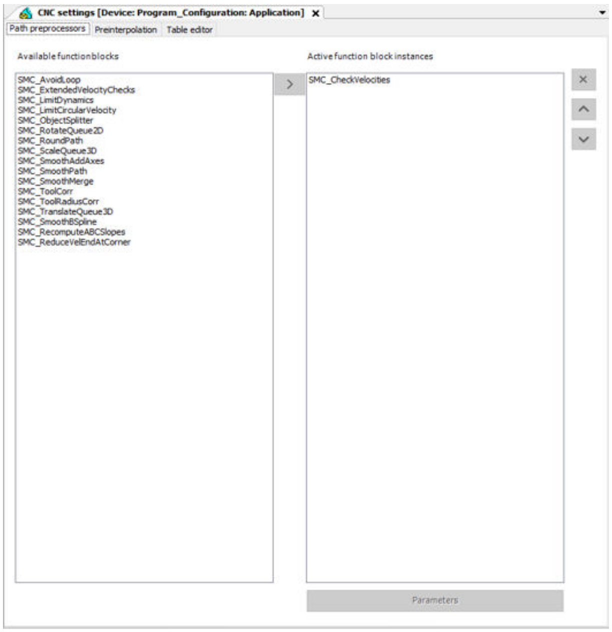
Application-wide CNC settings are stored in the "CNC settings" object

Setting item		Description
Implementation	Din66025	Your program will be written in text format. Be sure to use this setting.
	Table	Do not use.
Compile mode	SMC_CNC_REF	Normally, use this setting.
	SMC_OUTQUEUE	The movement speed improves due to the prior calculation method, but pre-processing cannot be performed, CNC program settings cannot be changed, and CNC programs cannot be joined together.
	File	CNC programs are generated as external reference files. Even files created by external text editors can similarly work.
File name		If the Compile mode is set to File, the name of the external reference file must be entered.
Queue size		This is the number of CNC program elements managed under SMC_OUTQUEUE at runtime. (Recommended number: 10 to 30)
Default values		Specify default values of the velocity, acceleration, and deceleration for interpolation operation. Unless velocity, acceleration, and deceleration are specified using G-code, interpolation control will be performed using these default values.
Default values for fast forward (G0)		Specify default values of the velocity, acceleration, and deceleration for interpolation operation with G0. Unless velocity, acceleration, and deceleration are specified with G0, interpolation control will be performed using these default values.
3D-Mode		XY 2-axis interpolation control is set by default. To execute XYZ 3-axis interpolation control without G16 to G19 plane specification, make this setting enabled. If the program is executed with an external file, the argument in the function block FB (SMC_ReadNCFile2) serves as the similar setting.
Start position		For the SMC_OUTQUEUE type, specify the coordinates from which interpolation control is to be started.

■ "CNC settings" - Setting window

In the "CNC settings” object added concurrently with the CNC program, you can configure path processing settings for CNC programs created under the SMC_OUTQUEUE type.

- "Path preprocessors" tab


CNC settings [Device: Program_Configuration: Application]

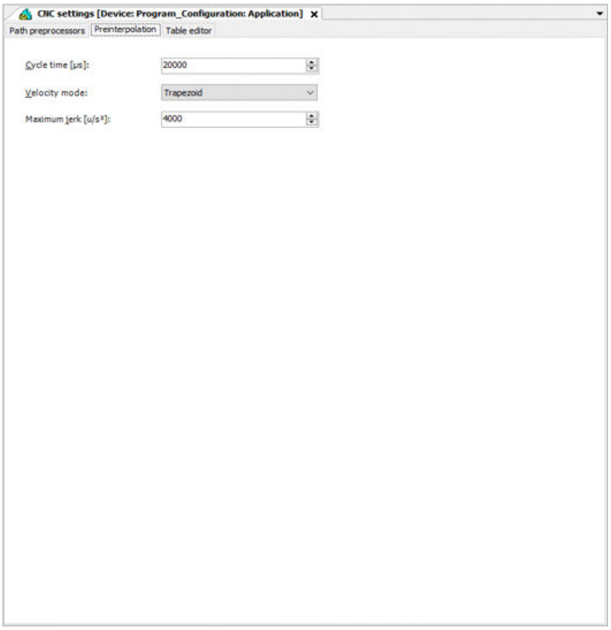
Available functionblocks

SMC_AvoidLoop
SMC_ExtendedVelocityChecks
SMC_LimitDynamics
SMC_LimitCircularVelocity
SMC_ObjectSplitter
SMC_RotateQueue2D
SMC_RoundPath
SMC_ScaleQueue3D
SMC_SmoothAddAxes
SMC_SmoothPath
SMC_SmoothMerge
SMC_ToolCorr
SMC_ToolRadiusCorr
SMC_TranslateQueue3D
SMC_SmoothBSpline
SMC_RecomputeABCSlopes
SMC_ReduceVelEndAtCorner

Active function block instances

SMC_CheckVelocities

- "Preinterpolation" tab

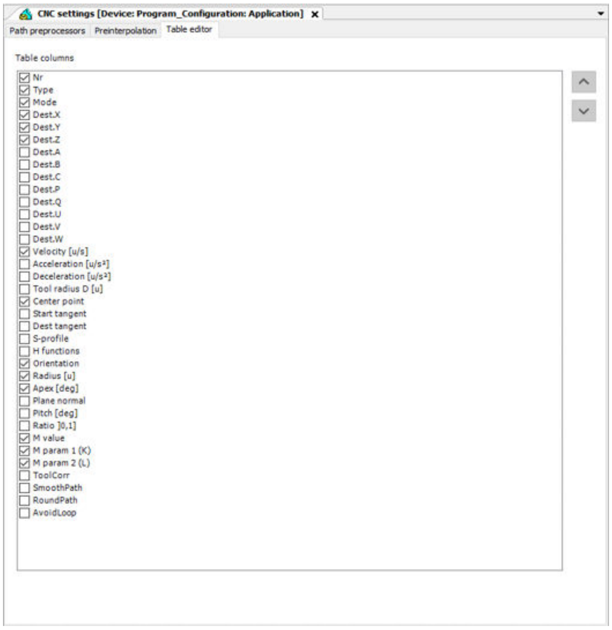

CNC settings [Device: Program_Configuration: Application]

Cycle time [µs]: 20000

Velocity mode: Trapezoid

Maximum jerk [u/s³]: 4000

- "Table editor" tab



CNC settings [Device: Program_Configuration: Application]

Path preprocessors | Preinterpolation | Table editor

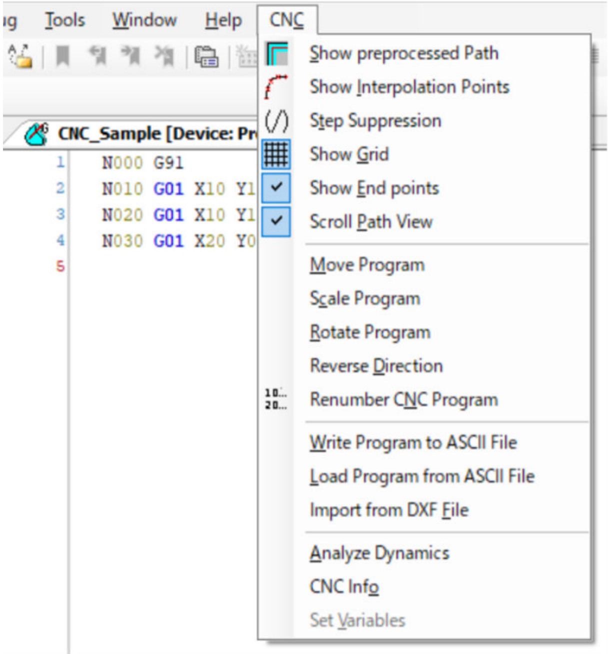
Table columns

- ☑ Nr
- ☑ Type
- ☑ Mode
- ☑ Dest.X
- ☑ Dest.Y
- ☑ Dest.Z
- □ Dest.A
- □ Dest.B
- □ Dest.C
- □ Dest.P
- □ Dest.Q
- □ Dest.U
- □ Dest.V
- □ Dest.W
- ☑ Velocity [u/s]
- □ Acceleration [u/s^2]
- □ Deceleration [u/s^2]
- □ Tool radius D [u]
- ☑ Center point
- □ Start tangent
- □ Dest tangent
- □ S-profile
- □ H functions
- ☑ Orientation
- ☑ Radius [u]
- ☑ Apex [deg]
- □ Plane normal
- □ Pitch [deg]
- □ Ratio]0,1]
- ☑ M value
- ☑ M param 1 (K)
- ☑ M param 2 (L)
- □ ToolCorr
- □ SmoothPath
- □ RoundPath
- □ AvoidLoop

Setting item		Description
Tab name	Settings	
Path preprocessors	-	For SMC_OUTQUEUE type programs, pre-processing is required to be defined here in advance. Even for SMC_CNC_REF type programs or external files, results of pre-processing can be displayed and checked in the CNC graphic editor.
Preinterpolation	Cycle time	Specify time for MotionTask.
	Velocity mode	Specify an acceleration/deceleration profile. The setting can be changed by the argument in SMC_Interpolator when the program is executed.
	Maximum jerk	Specify the maximum jerk value for interpolation control operation. The setting can be changed by the argument in SMC_Interpolator when the program is executed.
Table editor	-	This tab is used to perform CNC in table format. Do not use.

■ CNC menu

When the CNC program editing window is opened, the menu bar displays the "CNC" menu, and you can configure settings for the following items.



Setting items	Overview
Show preprocessed Path	Select to show or hide the path with preprocessing in the CNC graphic editor.
Show Interpolation Points	Select to show or hide interpolation points in the CNC graphic editor.
Step Suppression	Select to activate or deactivate the step suppression.
Show Grid	Select to show or hide the grid in the CNC graphic editor.
Show End Points	Select to show or hide end points in the CNC graphic editor.
Scroll Path View	If this is activated, the view of the path on a line selected in the CNC editor is automatically scrolled and zoomed such that the path is visible in the CNC graphic editor window.
Move Program	This opens the translation vector dialog box where you can enter movement amounts for the axes. This moves the CNC program along the axes by specified amounts.
Scale Program	This opens the scale program dialog box where you can enter a scaling factor and select a target axis. This scales up or down the CNC program by the specified scaling factor along the selected axis.
Rotate Program	This opens the rotation parameters dialog box where you can enter an angle of rotation and a rotation factor for each axis. This causes the CNC program to rotate about each axis by the specified angle of rotation × rotation factor.
Reverse Direction	This sets the path elements of the CNC program in reverse order.
Renumber CNC Program	This refreshes the numbering of program lines in the CNC program. The numbers start at 0 and are incremented by 10.
Write Program to ASCII File	This opens the dialog box for saving a file. This causes the CNC program to be saved as an ASCII file.
Load Program from ASCII File	This opens the dialog box for selecting a file. This loads CNC program data from the selected ASCII file.
Import from DXF File	Do not use.

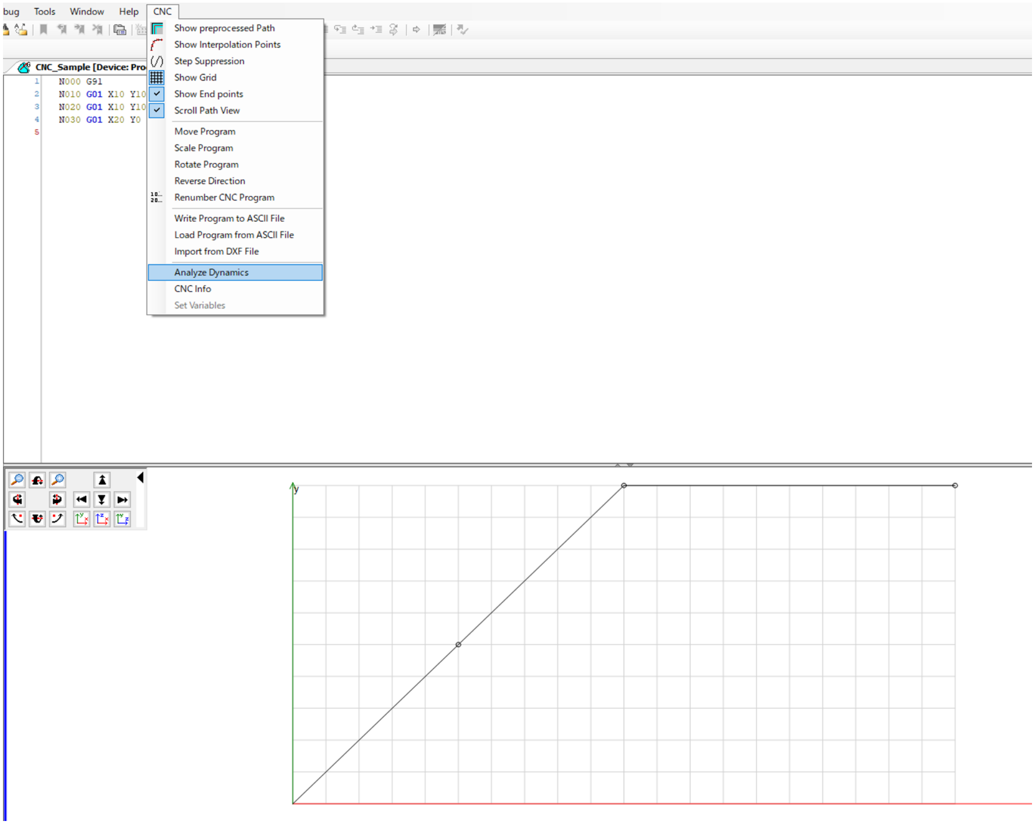
Setting items	Overview
Analyze Dynamics	A dynamical analysis of the CNC program allows you to preliminarily check how each axis moves. For details, refer to "11.6.3 Dynamical Analysis for CNC Program".
CNC Info	Do not use.
Set Variables	Do not use.

11.6.3 Dynamical Analysis for CNC Program

■ Preliminarily checking motion trace

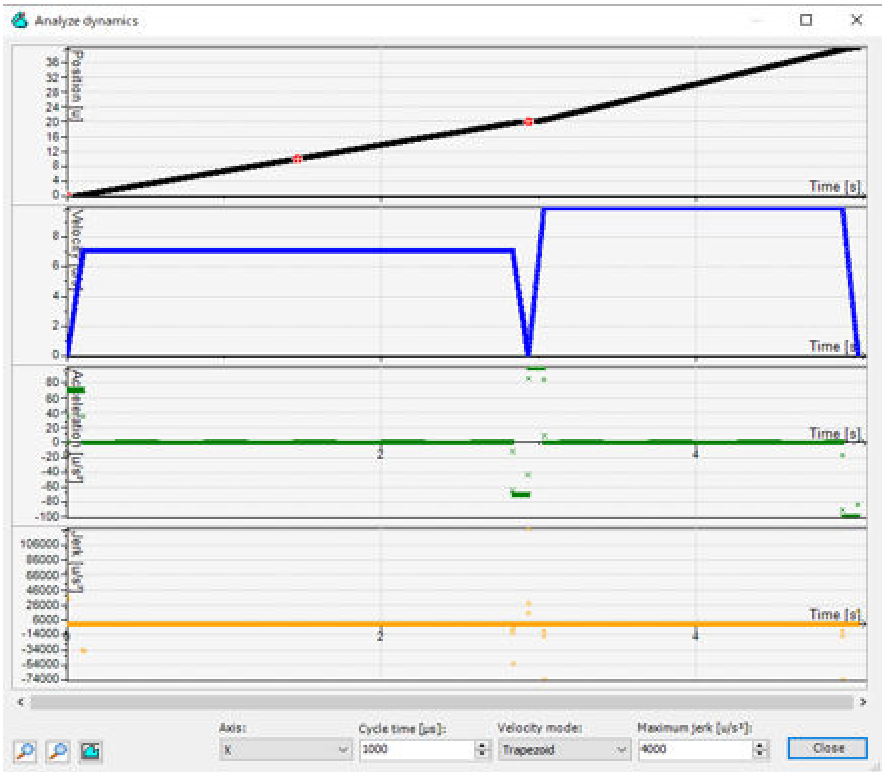
A dynamical analysis of the created CNC program allows you to preliminarily check an expected motion curve of each axis.

1. In the CNC program editing window, from the menu bar, select **CNC**>**Analyze Dynamics**.



```
N000 G91
N010 G01 X10 Y10
N020 G01 X10 Y10
N030 G01 X20 Y0
```

2. The “Dynamical Analysis” window will be displayed. Specify appropriate settings in the "Axis", "Cycle time", "Velocity mode", and "Maximum jerk" fields.



Axis: X

Cycle time [µs]: 1000

Velocity mode: Trapezoid

Maximum jerk [u/s³]: 4000

Note

- If variables are used in CNC programs, the "Dynamical Analysis" window cannot be displayed.
- If starting coordinates are specified or CNC programs are joined or repeated in POU, actual movements will differ from the results of dynamical analysis.

11.7 Motion Function Errors

11.7.1 Overview of Motion Function Errors

Motion function errors can be classified as below.

■ Function block (FB) errors

These errors occur when motion function blocks are executed. They are defined as SMC_ERROR in CODESYS.

Errors can be classified as below.

- Errors resulting in errorstop

  In CODESYS, if an error that is judged to be operation non-continuable occurs, the axis state will be set to "errorstop" and control for the relevant axes will stop.

  In a function block that is being executed, the "CommandAborted" flag is set to TRUE, causing the function block to terminate.

  The "errorstop" state is released by executing the MC_Reset function block.

- Errors not resulting in errorstop

  In CODESYS, if an error that is judged to be operation continuable (an error that can be cleared by re-executing the function block, such as a parameter error to the function block) occurs, the "Error" flag of the function block will be set to TRUE and an error code will be set in ErrorID. The error is cleared by setting the "Execute" flag in the function block to FALSE, enabling the function block to be re-executed.

■ DriveInterfaceError

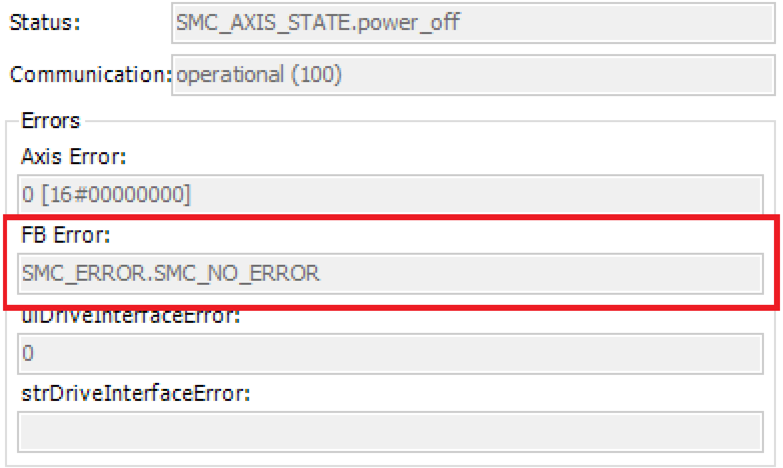
Errors that occur in SM_Drive_ETC_Panasonic_MINAS_A5B are output as DriveInterfaceError. DriveInterfaceError is an internal error.

■ Amplifier alarm

Alarms and warnings occur in servo amplifiers.

11.7.2 Error Check Method

■ Online monitor in the axis setting window for SM_Drive_ETC_Panasonic_MINAS_A5B



Status: SMC_AXIS_STATE.power_off

Communication: operational (100)

Errors

Axis Error: 0 [16#00000000]

FB Error: SMC_ERROR.SMC_NO_ERROR

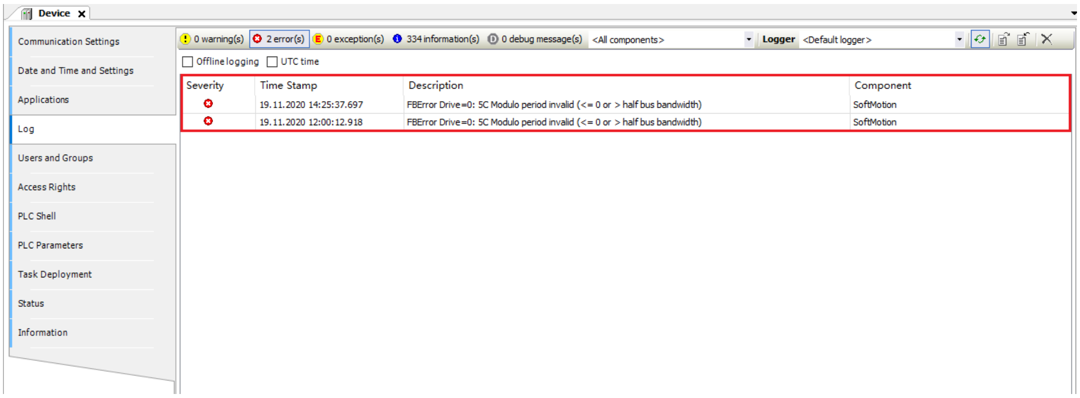
uiDriveInterfaceError: 0

strDriveInterfaceError:

The oldest error is displayed in the "FB error" field. This is the same error result as that output by the MC_ReadFBError function block.

* Some FB errors are not displayed (SMC_RP_REQUESTING_ERROR, etc.).

■ Device log



0 warning(s) 2 error(s) 0 exception(s) 334 information(s) 0 debug message(s) <All components> Logger <Default logger>

□ Offline logging □ UTC time

Severity	Time Stamp	Description	Component
⊗	19.11.2020 14:25:37.697	FBError Drive=0: 5C Modulo period invalid (<= 0 or > half bus bandwidth)	SoftMotion
⊗	19.11.2020 12:00:12.918	FBError Drive=0: 5C Modulo period invalid (<= 0 or > half bus bandwidth)	SoftMotion

All errors that occurred in the past are displayed. This is regardless of whether "errorstop" occurred.

* Some FB errors are not displayed (SMC_RP_REQUESTING_ERROR, etc.).

■ Errors that can be output to programs

Return value to function block	When an FB error occurs, the "Error" flag is set to TRUE and the error ID of SMC_ERROR is output to ErrorID.
MC_ReadFBError	FB errors that occurred are displayed, regardless of "errorstop". Some FB errors are not displayed (SMC_RP_REQUESTING_ERROR, etc.).

Amplifier alarm	ETC_CO_SdoRead FB or PDO mapping "Error code"

11.7.3 Clearing Errors

When errorstop occurs	Execute MC_Reset.
When an error returns to a function block	Execute the same function block with the "Execute" flag set to FALSE.
When an amplifier alarm occurs	Execute SMC3_ReinitDrive.

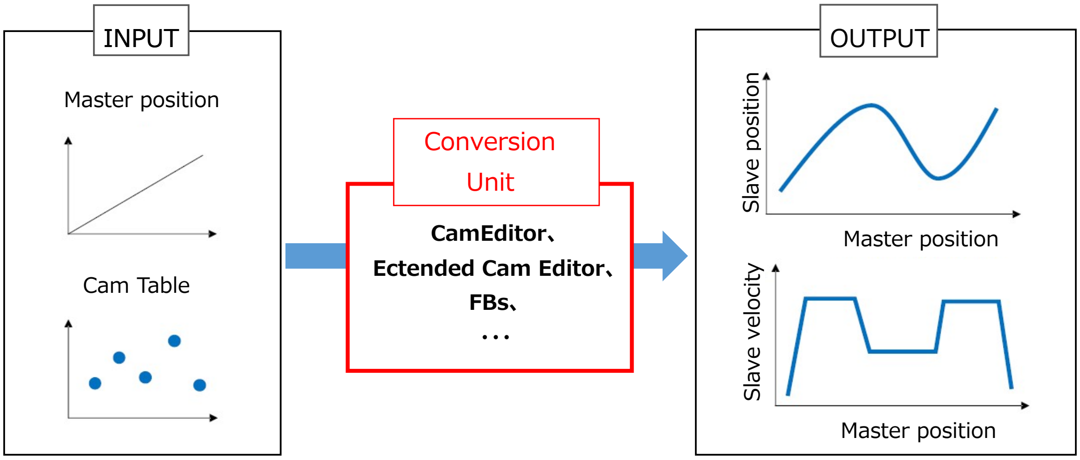
(MEMO)

12 Cam synchronous Control

12.1 Overview of Cam Control and How to Use It

This section describes an overview of cam control and how to use cam control with GM Programmer.

Cam control is control used to synchronize the master axis and slave axis in accordance with information (a cam profile) in which the position, velocity, and acceleration of the slave axis are defined based on the phase of the master axis (positional information). In the cam profile, information about the position and velocity of the slave axis is created from a cam table, in which representative points are defined, and the phase of the master axis (positional information) through a converter.



At the converter, conversion is performed largely through the following three steps to output information on slave position from information on master position.

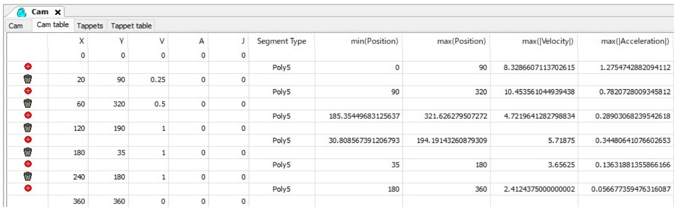
1 2 Procedure

1. Create a cam table

Create a cam table to define representative points that are elements of a cam profile (waveform information).

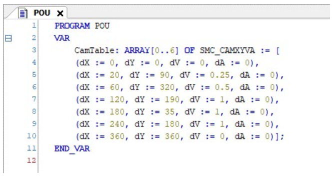
Configure representative points into which an operation section are divided and specify how these representative points are connected to each other to create a cam table. With GM Programmer, you can create it by the following three types of methods.

- Cam editor
- Expansion cam editor
- Programming in POU



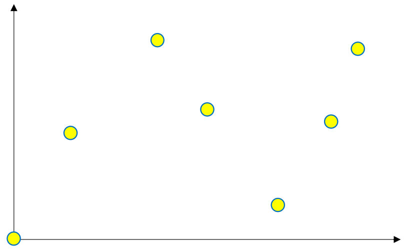
Cam | Cam table | Tappets | Tappet table

	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	90	8.3286607113702615	1.2754742882094112
🗑	20	90	0.25	0	0					
⊕						Poly5	90	320	10.453561044939438	0.7820728009345812
🗑	60	320	0.5	0	0					
⊕						Poly5	185.35449683125637	321.626279507272	4.7219641282798834	0.28903068239542618
🗑	120	190	1	0	0					
⊕						Poly5	30.808567391206793	194.19143260879309	5.71875	0.34480641076602653
🗑	180	35	1	0	0					
⊕						Poly5	35	180	3.65625	0.13631881355866166
🗑	240	180	1	0	0					
⊕						Poly5	180	360	2.4124375000000002	0.056677359476316087
	360	360	0	0	0					



```
PROGRAM POU
VAR
    CamTable: ARRAY[0..6] OF SMC_CAMXYVA := [
    (dX := 0, dY := 0, dV := 0, dA := 0),
    (dX := 20, dY := 90, dV := 0.25, dA := 0),
    (dX := 60, dY := 320, dV := 0.5, dA := 0),
    (dX := 120, dY := 190, dV := 1, dA := 0),
    (dX := 180, dY := 35, dV := 1, dA := 0),
    (dX := 240, dY := 180, dV := 1, dA := 0),
    (dX := 360, dY := 360, dV := 0, dA := 0)];
END_VAR
```

Cam Table

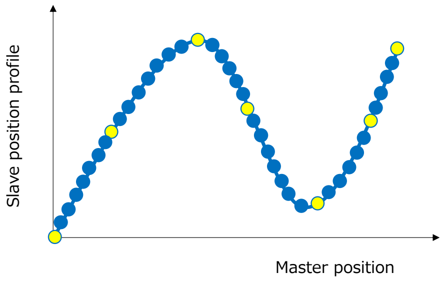


2. Convert the cam table into executable format

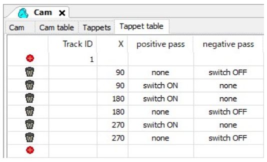
   Read the created cam table and convert it into executable format.

   First, convert the cam table into profile data in which each section between the adjacent representative points is divided by information on the master position. Next, create a structure MC_CAM_REF that includes tappet settings in addition to the profile data about position. The tappet settings are a list of data (tappet table) about points that are to be switched ON or OFF in response to the master and slave positions. The data about points includes ON/OFF operation settings aside from the positional information.

   Cam tables created by the Cam editor are automatically converted into profile data and MC_CAM_REF.

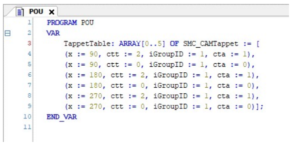


Tappet Settings

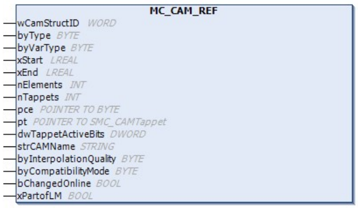


Cam | Cam table | Tappets | Tappet table

	Track ID	X	positive pass	negative pass
⊕	1			
🗑		90	none	switch OFF
🗑		90	switch ON	none
🗑		180	switch ON	none
🗑		180	none	switch OFF
🗑		270	switch ON	none
🗑		270	none	switch OFF
⊕				



```
PROGRAM POU
VAR
    TappetTable: ARRAY[0..5] OF SMC_CAMTappet := [
    (x := 90, ctt := 2, iGroupID := 1, cta := 1),
    (x := 90, ctt := 0, iGroupID := 1, cta := 0),
    (x := 180, ctt := 2, iGroupID := 1, cta := 1),
    (x := 180, ctt := 0, iGroupID := 1, cta := 0),
    (x := 270, ctt := 2, iGroupID := 1, cta := 1),
    (x := 270, ctt := 0, iGroupID := 1, cta := 0)];
END_VAR
```



MC_CAM_REF

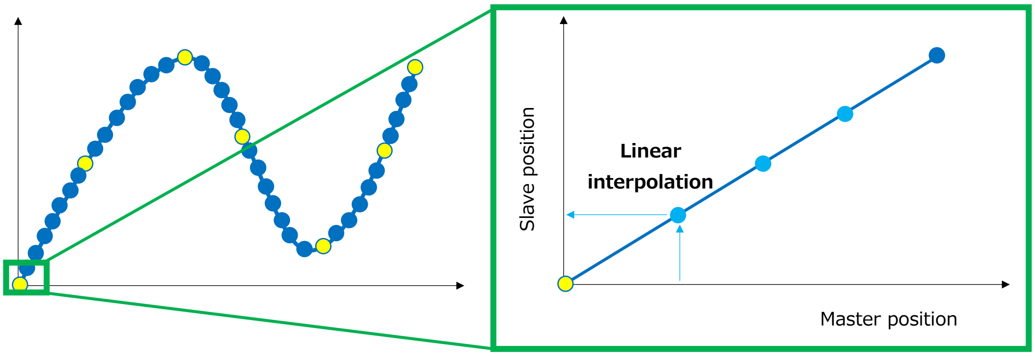
wCamStructID *WORD*
byType *BYTE*
byVarType *BYTE*
xStart *LREAL*
xEnd *LREAL*
nElements *INT*
nTappets *INT*
pce *POINTER TO BYTE*
pt *POINTER TO SMC_CAMTappet*
dwTappetActiveBits *DWORD*
strCAMName *STRING*
byInterpolationQuality *BYTE*
byCompatibilityMode *BYTE*
bChangedOnline *BOOL*
xPartofLM *BOOL*

3. Calculate axis command position from the profile data

   Read the profile data about position from the created MC_CAM_REF and compute a slave position to calculate a command position.

The command position is derived from information about the master position by linearly interpolating data between the divided points of the slave axis position profile. Cam synchronous control is performed by causing the slave axis to move according to the command position.

To calculate a command position with GM Programmer, you must do programming using function blocks in POU.



Info.

- For the procedure for creating MC_CAM_REF using the Cam editor, refer to"12.2.1 Creating MC_CAM_REF by Cam Editor".
- For the procedure for creating MC_CAM_REF using the Expansion cam editor, refer to "12.2.2 Creating MC_CAM_REF by Ectended Cam Editor".
- Function blocks used in cam control are not just for interpolation calculation but also for acquisition of tappet switching action, axis parameter calculation, and other purposes. For details, refer to the GM1 Series Reference Manual (Instruction Edition).

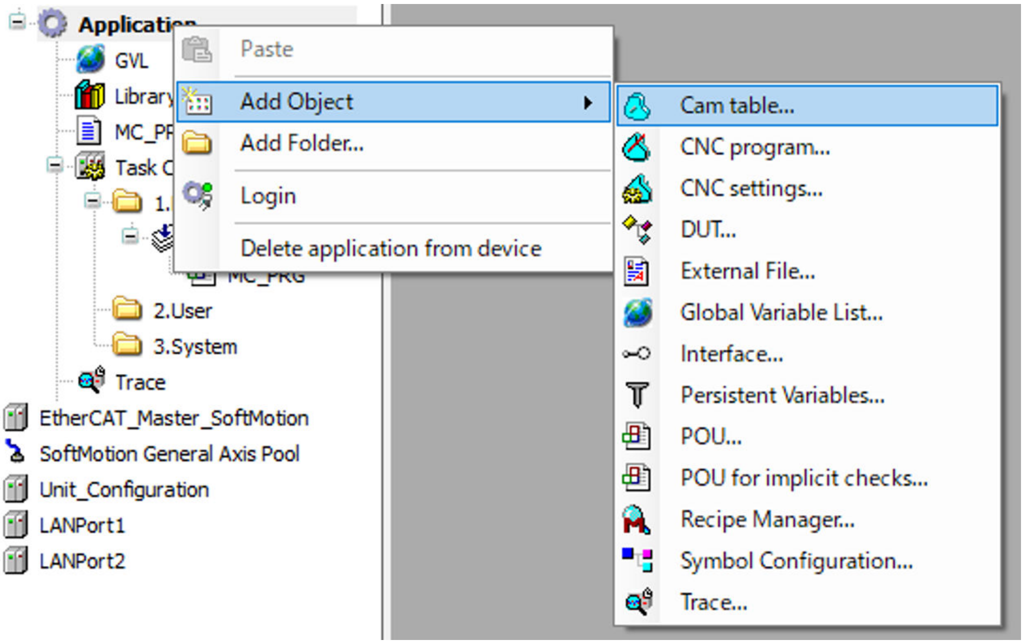
12.2 Creating Cam Data

12.2.1 Creating MC_CAM_REF by Cam Editor

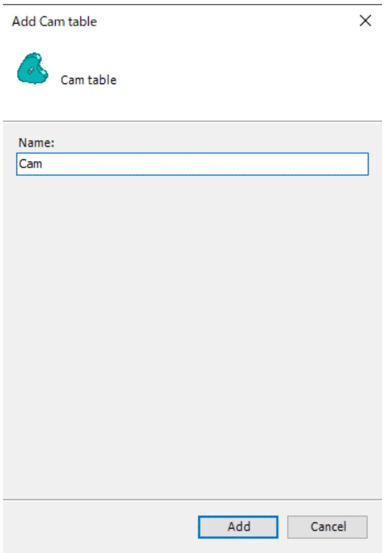
This section explains how to add a cam object and how to create MC_CAM_REF.

Procedure

1. Right-click the “Application” object in the navigation pane and then select**Add Object**>**Cam Table**from the context-sensitive menu that is displayed.
   The “Add cam table” dialog box will be displayed.



2. Enter a cam table name and then click the [Add] button.
   The name specified here will be used for an entry in a function block.



Add Cam table

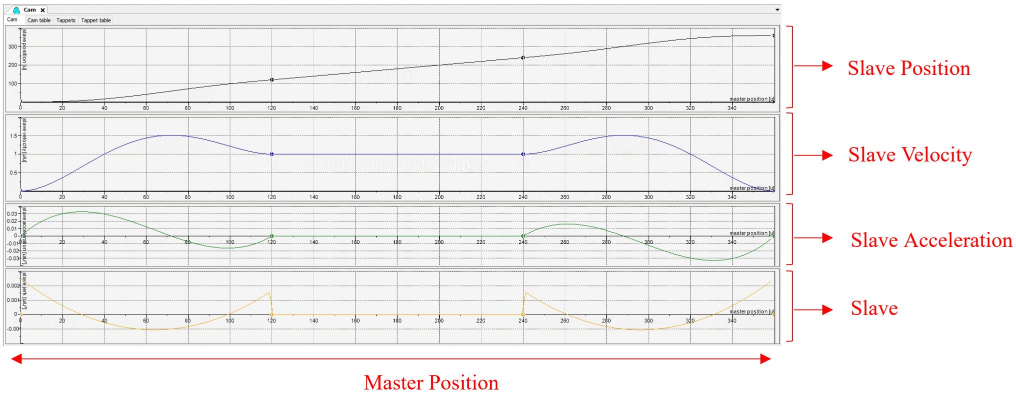
Cam table

Name: Cam

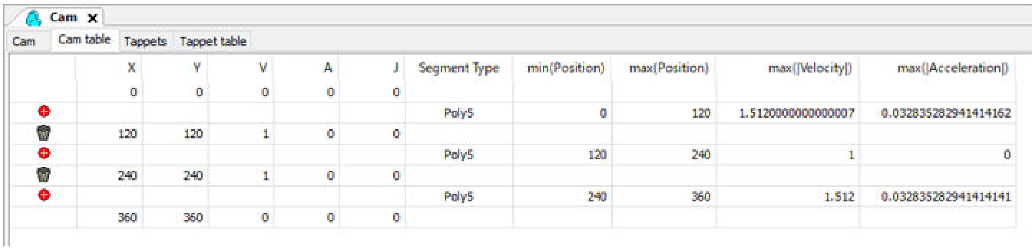
3. When the cam object is added, the “Cam” tab appears. Change the default path setting if necessary.

The “Cam” tab displays the following four items one by one in a vertical direction: position, velocity, acceleration, and jerk of a slave (vertical axis) relative to the master position (horizontal axis). The “Cam Table” tab displays numerical values that represent the graphs displayed in the “Cam” tab.

- “Cam” tab



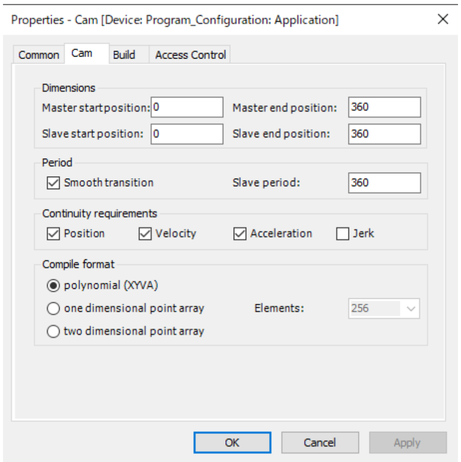
- “Cam table” tab



	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	120	1.5120000000000007	0.03283528294141462
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.03283528294141414141
	360	360	0	0	0					

You can change the maximum and minimum values on the master axis and slave axis in the cam table. Right-click the “Cam” object in the navigator pane and then select

"Properties" from the context-sensitive menu that is displayed. In the Properties window, click the "Cam" tab and change the values.



Properties - Cam [Device: Program_Configuration: Application]

Dimensions

Master start position: 0
Master end position: 360
Slave start position: 0
Slave end position: 360

Period

☑ Smooth transition
Slave period: 360

Continuity requirements

☑ Position ☑ Velocity ☑ Acceleration □ Jerk

Compile format

● polynomial (XYVA)
○ one dimensional point array
Elements: 256
○ two dimensional point array

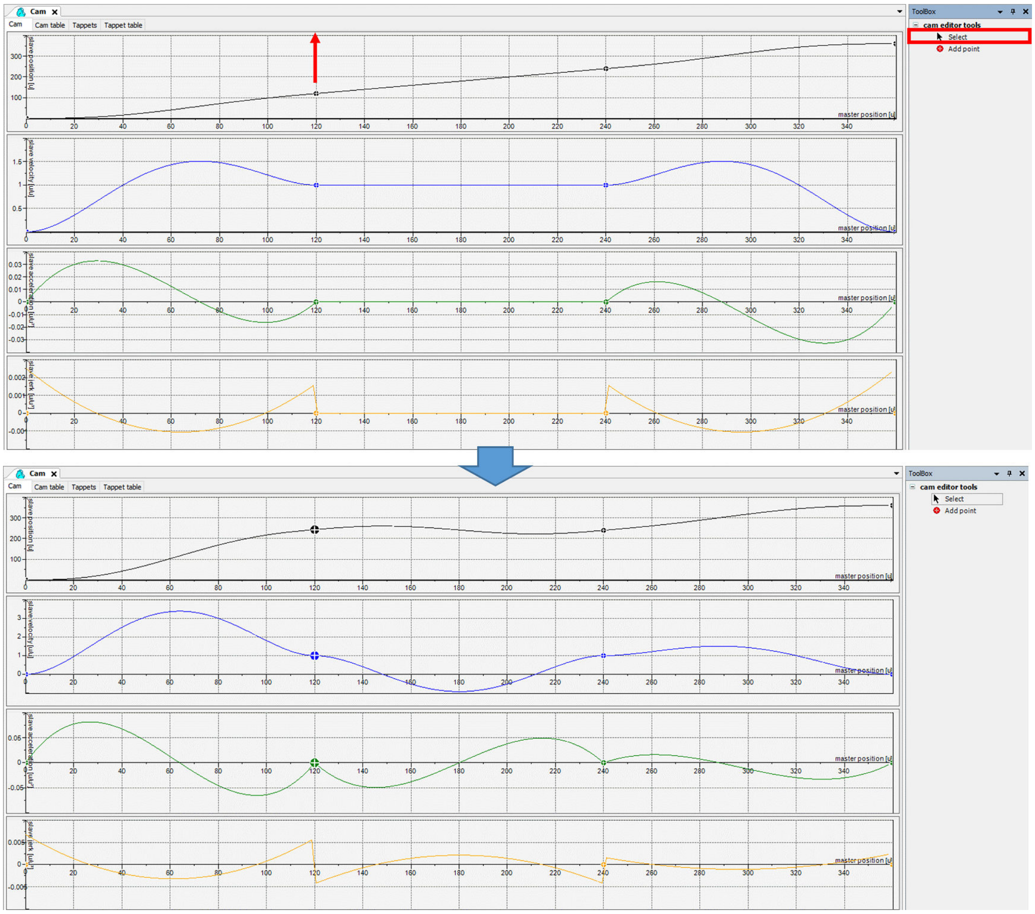
Item		Corresponding variables in MC_CAM_REF	Description
Dimensions	Master start/end position	xStart/xEnd	Defines scale values on the cam horizontal axis (master axis). A cam table can be created only within this range.
	Slave start / end position	-	Defines scale values on the cam vertical axis (slave axis). Slave position points on the Cam editor can be set only within this range. However, this does not apply to maximum and minimum values between position points.
Period	Smooth transition	-	Enabled: When a path is edited, the path is automatically adjusted such that the position, velocity, and acceleration values of the slave axis are continuous for the specified "Slave period" value. Details will be described later in step 4.
	Period	-	Specifies a period for the slave position. The specified value is reflected when the smooth transition setting is enabled.
Continuity requirements	Position, Velocity, Acceleration, Jerk	-	Continuity of parameters for which the check box is selected (whether data before and after the representative point is continuous) is checked during build.
Compile format(Note 1)	Polynomial (XYVA)	byType = 3	A cam table is made up of master position, slave position, slave velocity, and slave axis acceleration items.
	one dimensional point array	byType = 1	A cam table is made up of only a slave position item. (master position is

Item		Corresponding variables in MC_CAM_REF	Description
			divided into equal intervals by the number of elements.)
	two dimensional point array	byType = 2	A cam table is made up of master position and slave position items.
	Elements	-	This can be used for one dimensional and two dimensional point arrays. Specifies the number of elements that make up the array.

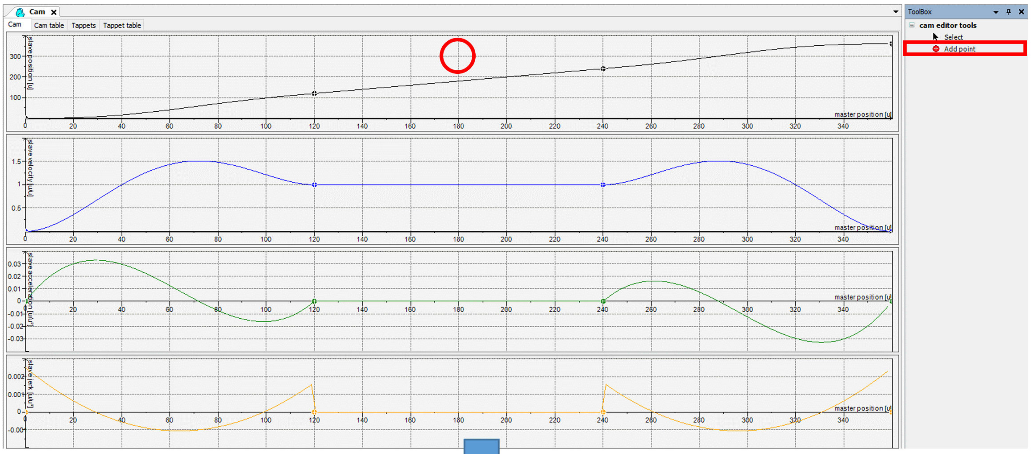
(Note 1) For details, refer to the GM1 Series Reference Manual (Instruction Edition).

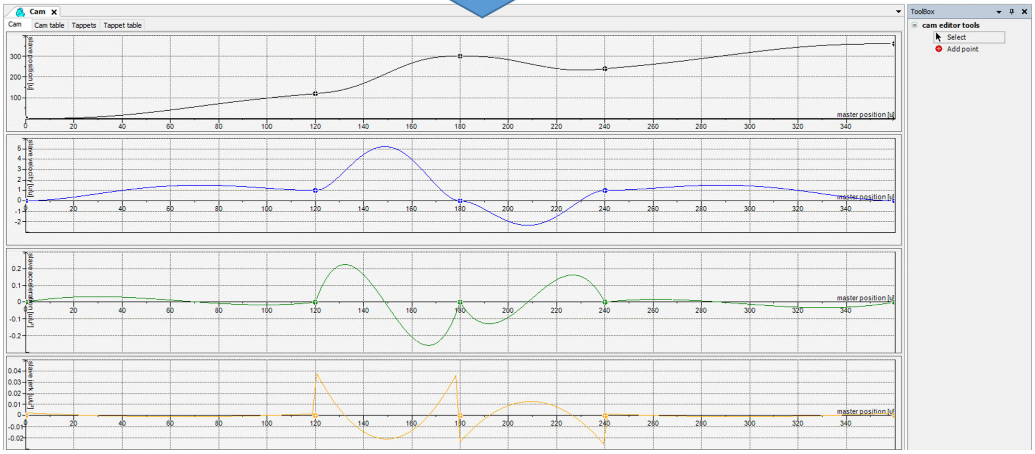
4. To change the path on the “Cam” tab, use Toolbox.

To change the path by using a point already set, select “Select” in Toolbox and drag the point you want to change. Points can move only upward or downward. To delete a point, select the point with the “Select” tool and then press the “Del” key.

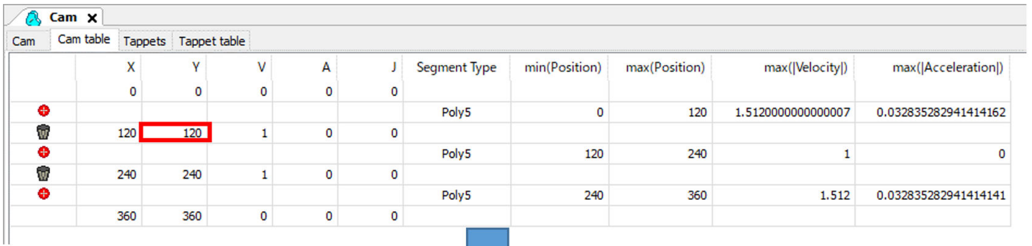


If you want change another part other than existing points, you must add a point to the path. Select “Add point” in Toolbox and then select a place on the path to which you want to add the point. The added point can be handled in the same way as points set by default.

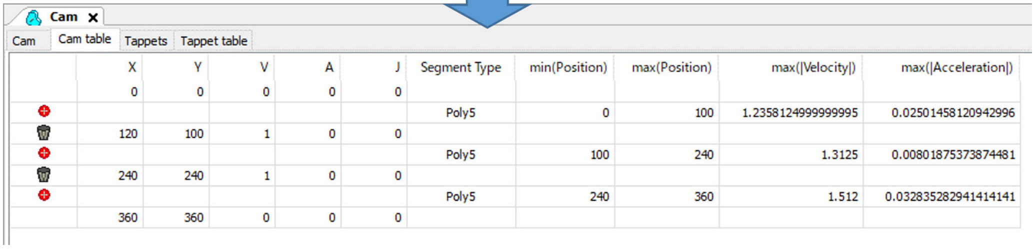




To edit the path on the “Cam table” tab, directly enter values. Entered values will also be reflected in graphs in the “Cam” tab.

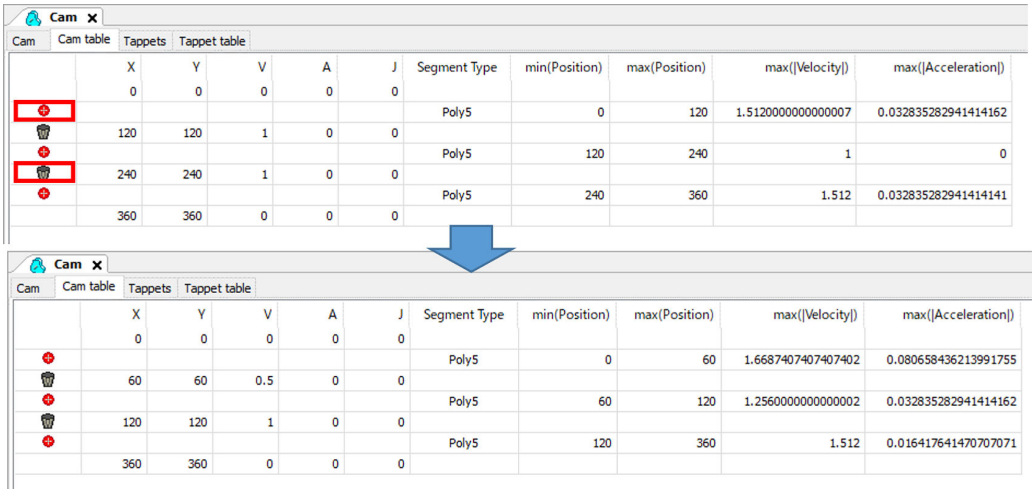


	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	120	1.5120000000000007	0.032835282941414162
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.032835282941414141
	360	360	0	0	0					



	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	100	1.2358124999999995	0.02501458120942996
🗑	120	100	1	0	0					
⊕						Poly5	100	240	1.3125	0.00801875373874481
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.032835282941414141
	360	360	0	0	0					

To add a point, click the ⊕ mark at a place where you want to add it. To delete the point, click the 🗑 mark. Note that after a change is made, segment types are automatically revised to ensure continuity.



	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	120	1.5120000000000007	0.032835282941414162
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.032835282941414141
	360	360	0	0	0					

	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	60	1.6687407407407402	0.080658436213991755
🗑	60	60	0.5	0	0					
⊕						Poly5	60	120	1.2560000000000002	0.032835282941414162
🗑	120	120	1	0	0					
⊕						Poly5	120	360	1.512	0.016417641470707071
	360	360	0	0	0					

Item	Description
X	Sets the position of the master axis.
Y	Sets the position of the slave axis.
V	Sets the velocity of the slave axis.
A	Sets the acceleration of the slave axis.
J	Sets the jerk of the slave axis.
Segment type	Sets an interpolation format for the path between the points. ● Poly5: Interpolates data by a fifth degree polynomial. This enables smooth connection in position and velocity. ● Line: linearly interpolates slave position. The slave velocity gets constant.
min(Position)	Minimum slave position between the points is calculated.
max(Position)	Maximum slave position between the points is calculated.
max(|Velocity|)	Maximum slave velocity (absolute value) between the points is calculated.
max(|Acceleration|)	Maximum slave axis acceleration (absolute value) between the points is calculated.

Info.

- When a path is edited, the position, velocity, acceleration, and jerk are each changed accordingly irrespective of the edited parameter.
- Editable parameters differ depending on the segment type. Editable parameters are as shown in the following table.

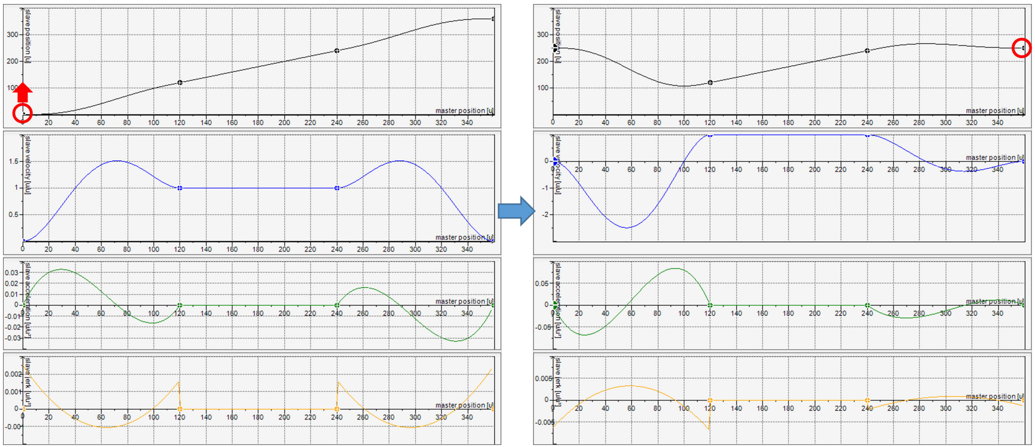
	Segment type	
	Poly5	**Line**
X	Editable	
Y	Editable	
V	Editable	Not editable
A	Editable	Not editable
J	Not editable	
min(Position)	Automatic computation	
max(Position)	Automatic computation	
max(|Velocity|)	Automatic computation	
max(|Acceleration|)	Automatic computation	

- Smooth transition setting

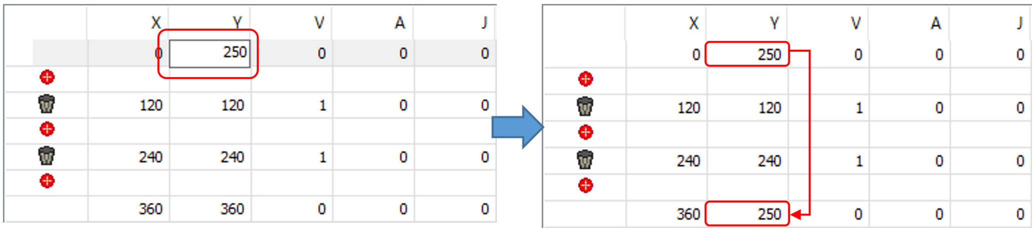
When a cam table is edited with the “Smooth transition” setting enabled, the position, velocity, and acceleration values are each automatically changed such that the slave axis parameters are continuous at the start and end points of the path. Whether the slave axis parameters at the start and end positions of the master axis match is checked to determine continuity.

The “Slave period” setting is set to a period for the slave position. The position of the slave axis returns to 0 when the position reaches the “Slave period” setting. In other words, the position of the slave axis is offset by a constant times the “Slave period” setting in some cases.
The following is an example in which the position settings (0, 0) are changed to (0, 250).

Smooth transition = Enabled, Slave period = 360
When the point is moved from the positions (0, 0) to (0, 250) on the “Cam” tab, the path changes accordingly such that the slave position represents the same value at the start and end points on the Cam table (master position). At this time, the (X, Y) values on the “Cam table” tab are also updated to (360, 250).



When the positions (X, Y) = (0, 0) are changed to input values (0, 250) on the “Cam table” tab, values (X, Y) = (360, 360) are changed to (360, 250) to update the slave position.

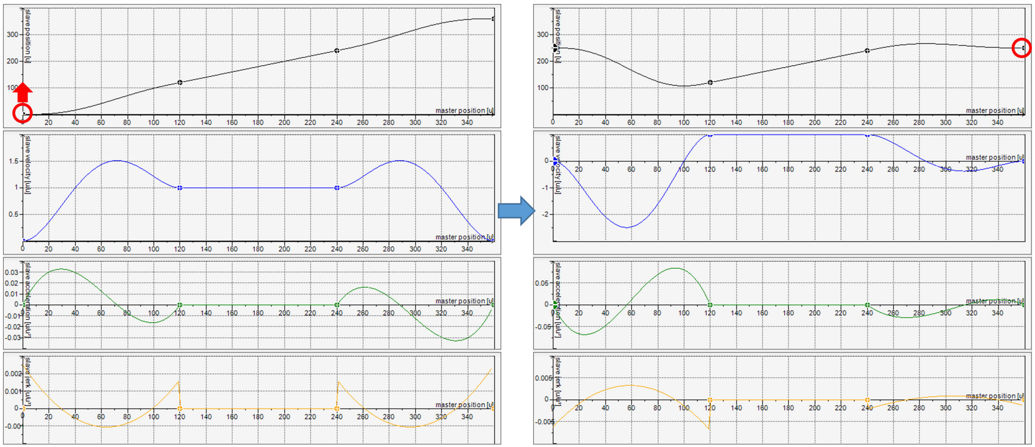


X	Y	V	A	J
0	250	0	0	0
120	120	1	0	0
240	240	1	0	0
360	360	0	0	0

X	Y	V	A	J
0	250	0	0	0
120	120	1	0	0
240	240	1	0	0
360	250	0	0	0

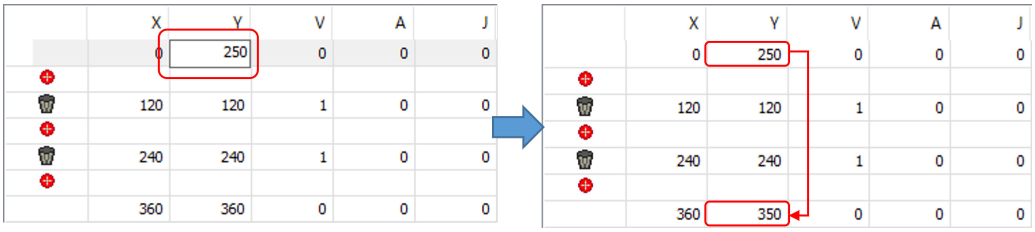
Smooth transition = Enabled, Slave period = 100

When the point is moved from the positions (0, 0) to (0, 250) on the “Cam” tab, in the same way as the case in which the Slave period is set to 360, the path and values on the “Cam table” tab are updated such that the slave position is continuous at the start and end points on the Cam table. The updated values (X, Y) = (360, 250).



On the “Cam ”tab, the slave position changes accordingly so as to represent the same value. However, on the “Cam table” tab, the slave position does not necessarily represent the same value but may differ by a constant times the “Slave period” setting.

When the positions (X, Y) = (0, 0) are changed to input values (0, 250) on the “Cam table” tab, values (X, Y) = (360, 360) are updated to (360, 350) such that the slave position is continuous. Since the Slave period is set to 100 in this case, the slave position represents 350 = 250 + 100 (= 100 × 1 cycle of the Slave period). In other words, the slave position is considered to be moved ahead by one Slave period to a point of 250 and is determined to be continuous despite the different values.



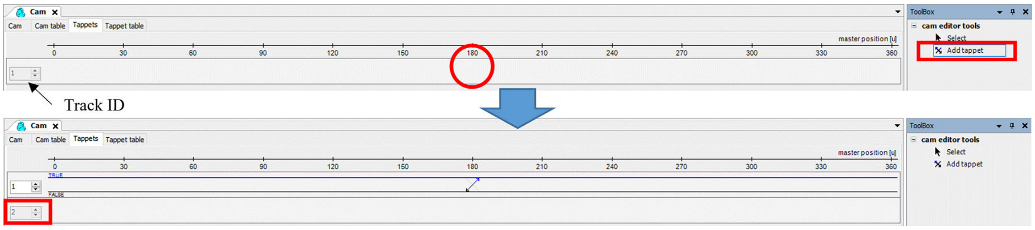
X	Y	V	A	J
0	250	0	0	0
120	120	1	0	0
240	240	1	0	0
360	360	0	0	0

X	Y	V	A	J
0	250	0	0	0
120	120	1	0	0
240	240	1	0	0
360	350	0	0	0

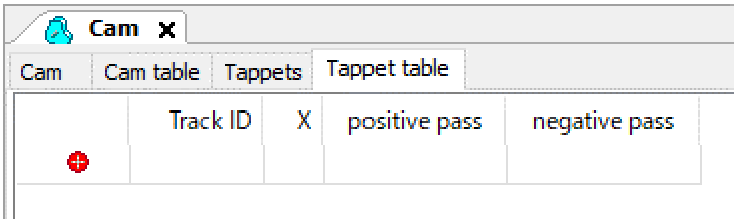
5. To change tappets on the “Tappets” tab, use Toolbox.

Select “Add tappet” in Toolbox and then select a place to which you want to add a tappet with a track ID. The tappet is added to the place at which you click the mouse. If you add a tappet to a track ID with which no tappet is present, a new track ID is created.

The tappet position and switch operation can be changed using “Select” in Toolbox. To delete a tappet, select the tappet you want to delete with the “Select” tool and then press the “Del” key.

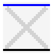
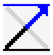
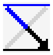
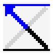
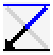
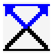


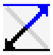
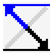
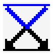
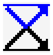
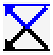
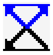
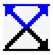
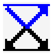
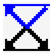
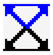
In the same way as paths, settings for tappets can be configured by entering numerical values, for example, through use of “Tappet table”.



Item	Description
Track ID	Sets an ID of the tappet path. All tappets of a shared track ID refer to the same tappet switching (a variable of the BOOL type). Up to 99 track IDs can be set within a range of 1 to 99.
X	Sets the position of the tappet. Up to three tappets can be specified for a shared point X irrespective of track ID. If four or more tappets are specified, an error occurs during cam synchronous control.
Positive pass	Sets action of the tappet performed when the axis passes through the position of the tappet in positive direction.
Negative pass	Sets action of the tappet performed when the axis passes through the position of the tappet in negative direction.

The following table shows tappet switch varieties. On the “Tappets” tab, you can switch the attribute of a tappet between ON and OFF by clicking the arrow part of the tappet symbol you want to switch.

Positive pass	Negative pass	Symbol
No action	No action	
Switch to ON	No action	
Switch to OFF	No action	
No action	Switch to ON	
No action	Switch to OFF	
Switch to ON	Switch to ON	

Positive pass	Negative pass	Symbol
Switch to ON	Switch to OFF	
Switch to OFF	Switch to ON	
Switch to OFF	Switch to OFF	
Invert	No action	
No action	Invert	
Switch to ON	Invert	
Invert	Switch to ON	
Invert	Switch to OFF	
Switch to OFF	Invert	
Invert	Invert	

Info.

- Up to 300 tappets can be used.

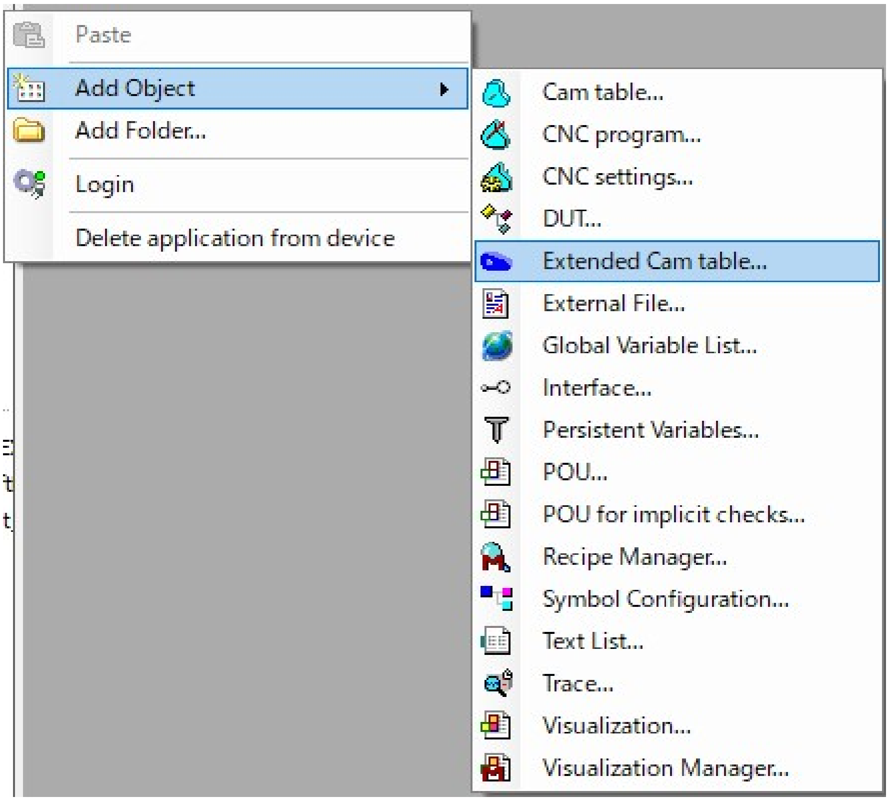
12.2.2 Creating MC_CAM_REF by Ectended Cam Editor

This section explains how to add a cam object and how to create MC_CAM_REF.

Procedure

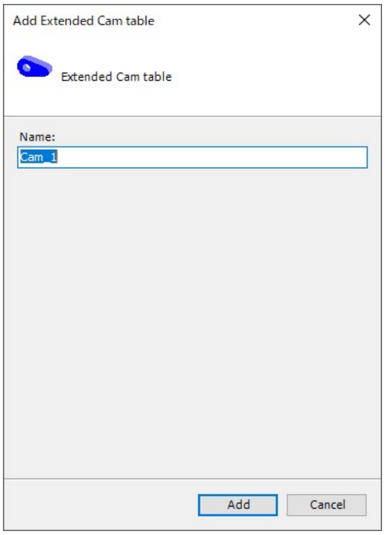
1. Right-click the “Application” object in the navigation pane and then select **Add Object**>**Ectended Cam Table** from the context-sensitive menu that is displayed.

The “Add cam table” dialog box will be displayed.



2. Enter a cam table name and then click the [Add] button.

The name specified here will be used for an entry in a function block.



Add Extended Cam table

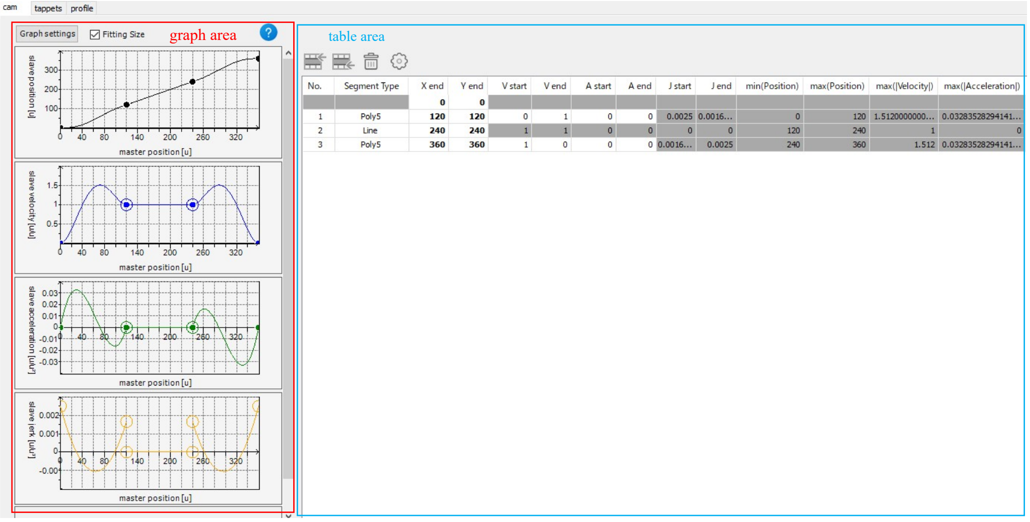
Extended Cam table

Name:

Cam_1

3. When the cam object is added, the “Cam” tab appears. Change the default path setting if necessary.

The “Cam” tab consists of a graph area and a table area.



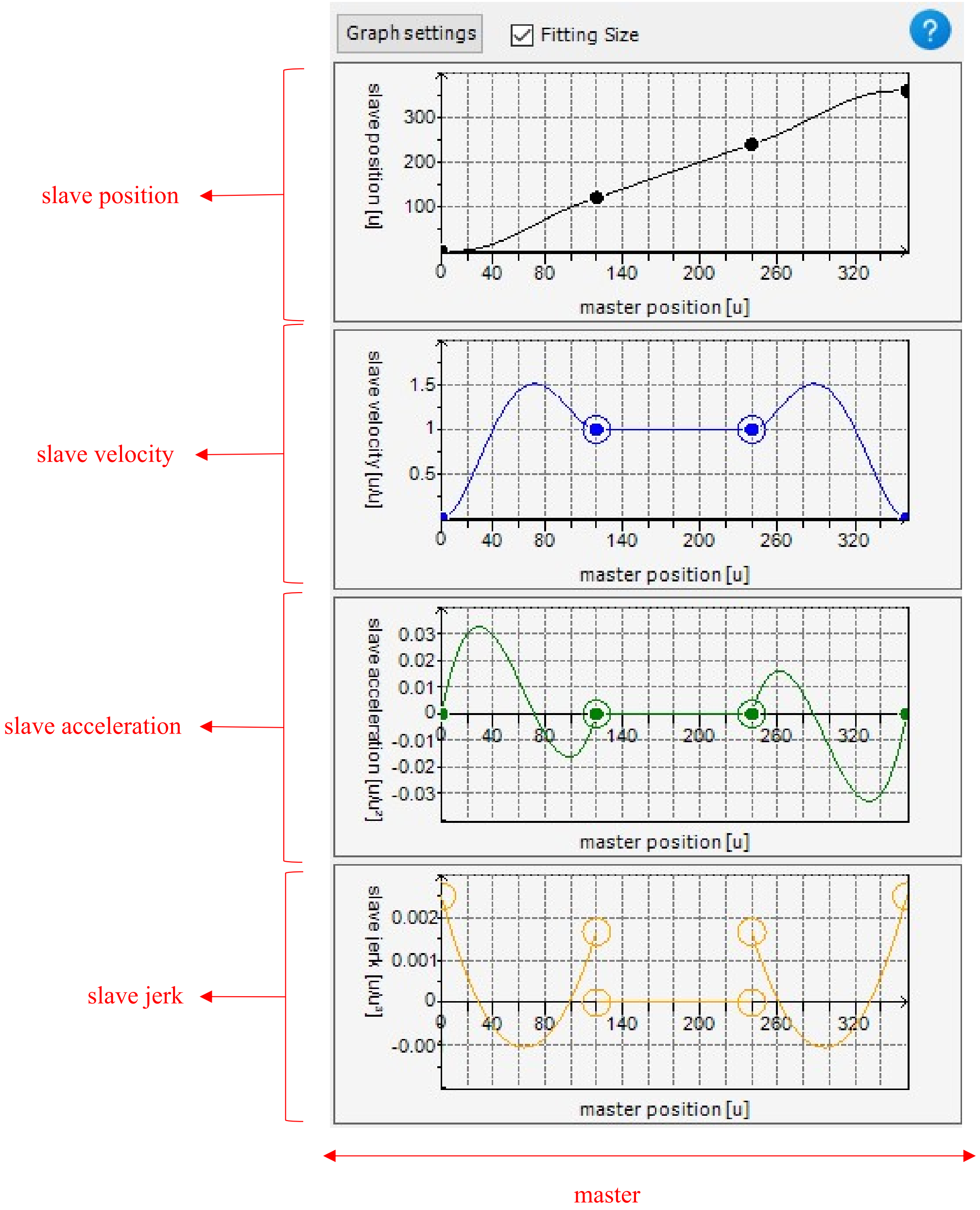
No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
		0	0										
1	Poly5	120	120	0	1	0	0	0.0025	0.0016...	0	120	1.5120000000...	0.03283528294141...
2	Line	240	240	1	1	0	0	0	0	120	240	1	0
3	Poly5	360	360	1	0	0	0	0.0016...	0.0025	240	360	1.512	0.03283528294141...

In the graph area, the following four items are displayed one by one in a vertical direction: position, velocity, acceleration, and jerk of a slave (vertical axis) relative to the master position (horizontal axis).

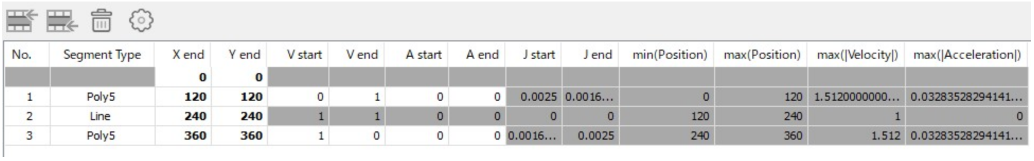
The [Graph setting] button is used to set the interval between graduations for the master position and also the maximum and minimum values of the slave graduations.

When the [Automatic adjustment] check box is enabled, the graph area size is automatically adjusted to match the window size.

Press the [?] button, and a dialog box that explains nodes will be displayed.



In the table area, numerical values that corresponds with graphs in the graph area are displayed.

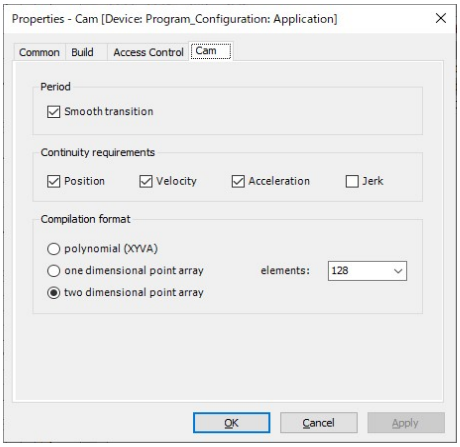


No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
		0	**0**										
1	Poly5	**120**	**120**	0	1	0	0	0.0025	0.0016...	0	120	1.5120000000...	0.03283528294141...
2	Line	**240**	**240**	1	1	0	0	0	0	120	240	1	0
3	Poly5	**360**	**360**	1	0	0	0	0.0016...	0.0025	240	360	1.512	0.03283528294141...

4. [Cam Table Property Setting]

Right-click the “Cam” object and then select “Properties” from the context-sensitive menu that is displayed.

In the Properties window, click the “Cam” tab and change the values.



Properties - Cam [Device: Program_Configuration: Application]

Common | Build | Access Control | Cam

Period

☑ Smooth transition

Continuity requirements

☑ Position ☑ Velocity ☑ Acceleration □ Jerk

Compilation format

○ polynomial (XYVA)

○ one dimensional point array elements: 128

● two dimensional point array

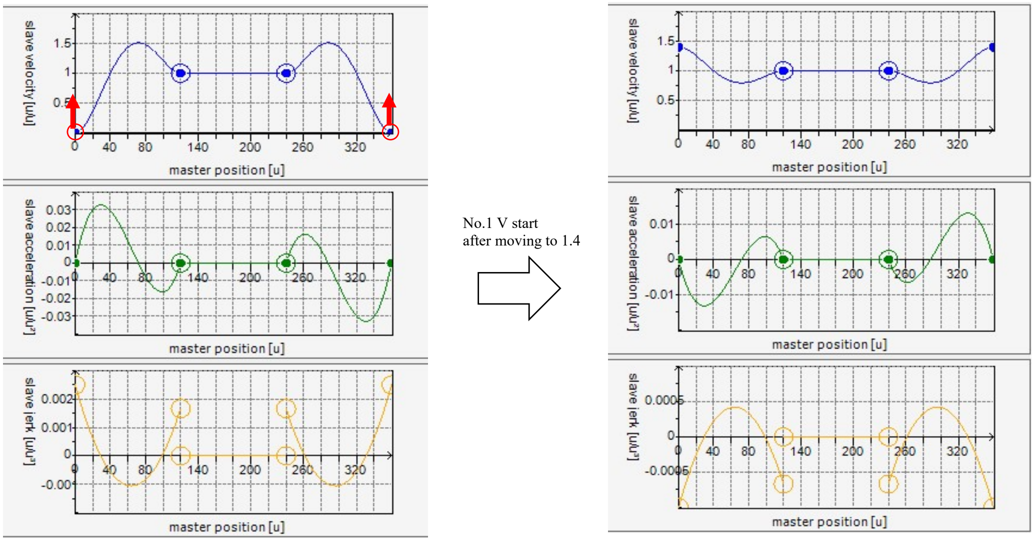
Item		Corresponding variables in MC_CAM_REF	Description
Transition	Smooth transition	-	Enabled: When a path is edited, the path is automatically adjusted such that the velocity and acceleration values of the slave axis at the start position and end position of the master axis are continuous.
Continuous condition check	Position, velocity, acceleration, and jerk	-	Continuity of parameters for which the check box is selected (whether data before and after the representative point is continuous) is checked during build.
Compile format(Note 1)	Polynomial (XYVA)	byType = 3	A cam table is made up of master position, slave position, slave velocity, and slave axis acceleration items. This compile format can be selected only for the segment type that can be expressed in polynomial form. Details will be described later in Section 12.3.
	One dimensional point array	byType = 1	A cam table is made up of only a slave position item. (master position is divided into equal intervals by the number of elements.)
	Two dimensional point array	byType = 2	A cam table is made up of master position and slave position items.
	Elements	-	This can be used for one dimensional and two dimensional point arrays. Specifies the number of elements that make up the array.

(Note 1) For details, refer to the GM1 Series Reference Manual (Instruction Edition).

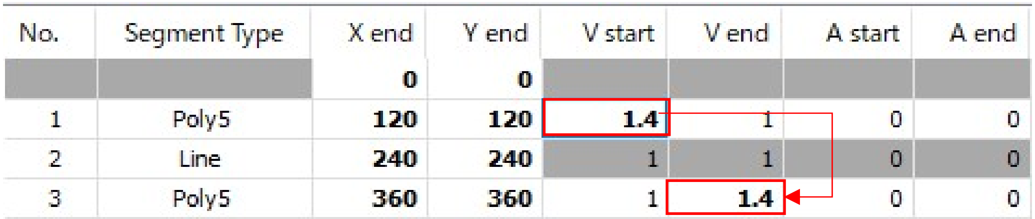
- Smooth transition setting

  When a cam graph or table is edited with the “Smooth transition” setting enabled, the path is automatically adjusted such that the velocity and acceleration values of the slave axis at the start position and end position of the master axis are continuous. Whether the slave axis parameters at the start and end positions of the master axis match is checked to determine continuity.

The following is an example in which (master axis position, slave axis velocity) settings are changed from (0, 0) to (0, 1.4). In the graph for the slave axis velocity, when the node (master axis position, slave axis velocity) is moved from (0, 0) to (0, 1.4), the path changes accordingly such that the slave velocity represents the same value at the start and end points of the master axis position. At this time, the values in the table area are also updated in conjunction with the graphs.



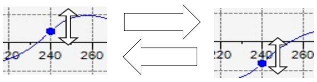
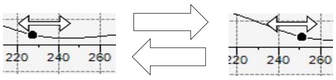
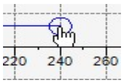
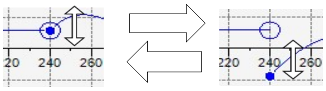
In the table area, if the start point Vstart is changed from 0 to 1.4, the end point Vend is also updated from 0 to 1.4.



No.	Segment Type	X end	Y end	V start	V end	A start	A end
		0	0				
1	Poly5	120	120	1.4	1	0	0
2	Line	240	240	1	1	0	0
3	Poly5	360	360	1	1.4	0	0

5. Moving Node on a Graph

The node can be moved by operating the mouse. It cannot be moved beyond ±999,999.

Node	Characteristic	Operation Of Dragging Nodes	<Shift>key + Operation Of Dragging Nodes
● Movable Node	The value of the slave axis can be changed	Drag Down  Drag Up	The value of the master axis can be changed. Only the slave axis position graph can be dragged left and right, but the result of the change is also reflected in the velocity, acceleration, and jerk of the slave axis. Drag Right  Drag Left
○ Fixed Node	The value of the axis cannot be changed	Can't move up or down 	
◉ Mixed Node	Has the Characteristic of an active node on one side and a fixed node on the other side	Drag Down(Separation)  Drag movable node up(sorption)	

6. Addition or Deletion of Segments in the Table Area

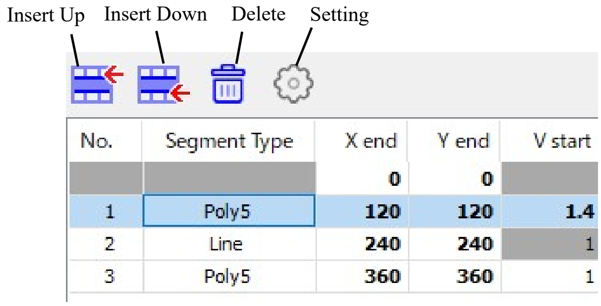
Segments can be added to or deleted through operations in the table area.

Select a row and then click the button at the top of the table area or select an item from the menu displayed when the row is right-clicked.

“Delete” can be executed simultaneously on multiple rows.

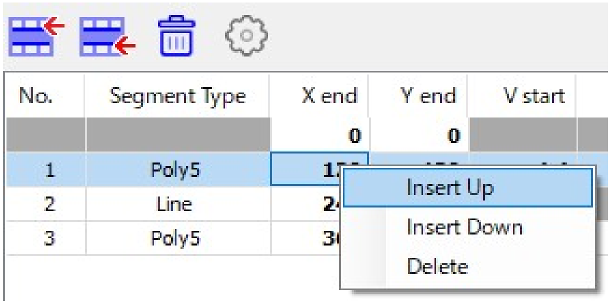
The settings are enabled when “Floating Trapezoid” is selected as a segment type. For details, refer to “"12.3.3 Floating Trapezoid"”.

Button



No.	Segment Type	X end	Y end	V start
		0	0	
1	Poly5	120	120	1.4
2	Line	240	240	1
3	Poly5	360	360	1

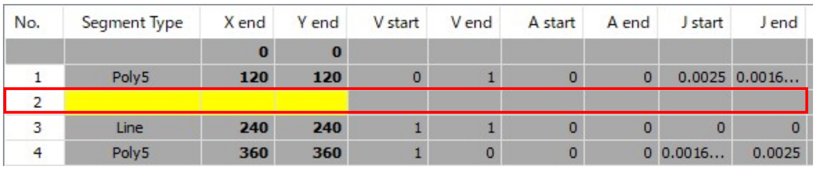
Right-click menu



No.	Segment Type	X end	Y end	V start
		0	0	
1	Poly5	1		
2	Line	2		
3	Poly5	3		

The newly added segment is in an unconfirmed state right after the addition as shown in the figure below.

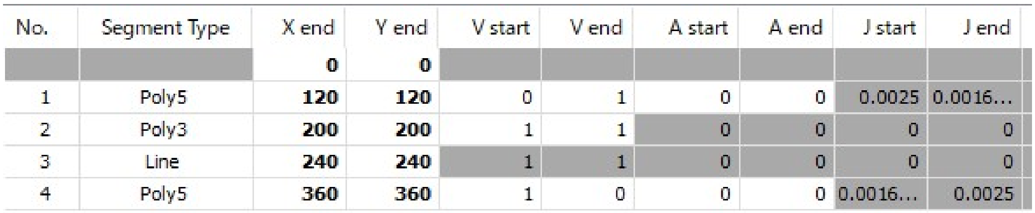
Entry to items other than “Segment type”, “Xend”, and “Yend” is prohibited.



Indeterminate segment

No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Poly5	120	120	0	1	0	0	0.0025	0.0016...
2									
3	Line	240	240	1	1	0	0	0	0
4	Poly5	360	360	1	0	0	0	0.0016...	0.0025

Once you enter values in “Segment type”, “Xend”, and “Yend” for the unconfirmed segment, the segment becomes a confirmed state and the entry prohibited state is cleared.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Poly5	120	120	0	1	0	0	0.0025	0.0016...
2	Poly3	200	200	1	1	0	0	0	0
3	Line	240	240	1	1	0	0	0	0
4	Poly5	360	360	1	0	0	0	0.0016...	0.0025

7. Editing segments

To edit a segment, enter numerical values in the table cells with white background and then press the <Enter> key.

Info.

- The range of numerical values that can be entered is between -999,999 and 999,999.
- When the segment is edited, the graph area is updated in conjunction with the edited contents.
- The following table shows a list of segment editing and display items.

Column name	Description
Segment type	Specify the curve type of each segment. Curve types will be described later in Section "12.3 Types and Setting of Cam Curves".
Xend	Indicates the master axis position at the end point of a segment. The start point of the next adjacent segment is at the same master axis position.
Yend	Indicates the slave axis position at the end point of a segment. The start point of the next adjacent segment is at the same slave axis position.
Vstart	Indicates the slave axis velocity at the start point of a segment.
Vend	Indicates the slave axis velocity at the end point of a segment.
Astart	Indicates the slave axis acceleration at the start point of a segment.
Aend	Indicates the slave axis acceleration at the end point of a segment.
Jstart	Indicates the slave axis jerk at the start point of a segment.
Jend	Indicates the slave axis jerk at the end point of a segment.
max(Position)	The maximum value of the slave axis position in the segment is displayed.
min(Position)	The minimum value of the slave axis position in the segment is displayed.
max(|Velocity|)	The maximum value in absolute value of the slave axis velocity in the segment is displayed.

Column name	Description
max(|Acceleration|)	The maximum value of the slave axis acceleration in the segment is displayed.

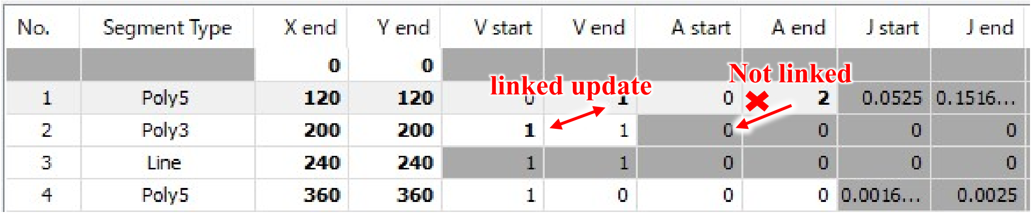
Items that can be entered vary depending on the segment type.

Details will be described later in Section "12.3 Types and Setting of Cam Curves".

Fine font numerical values indicate values calculated automatically and bold font numerical values indicate values manually entered.

If the borders between adjacent segments are both movable nodes, the numerical value entered in one movable node is also reflected on the other node.

If the borders between adjacent segments are mixed nodes (one is movable node and the other is fixed node), the numerical value entered in the movable node is not reflected on the fixed node.

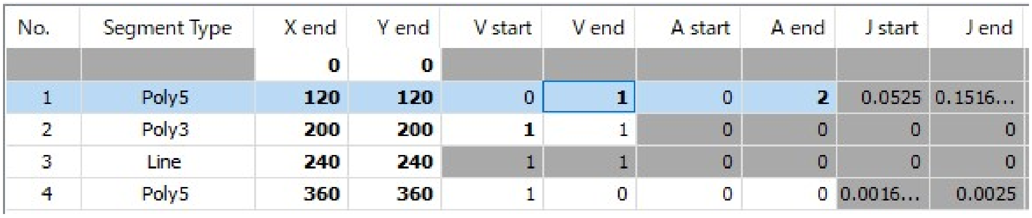


No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	**0**						
1	Poly5	**120**	**120**	0	**1**	0	**2**	0.0525	0.1516...
2	Poly3	**200**	**200**	**1**	1	0	0	0	0
3	Line	**240**	**240**	1	1	0	0	0	0
4	Poly5	**360**	**360**	1	0	0	0	0.0016...	0.0025

Select a bold numerical value, press the <Delete> key, and then press the <Enter> key to confirm the deletion.

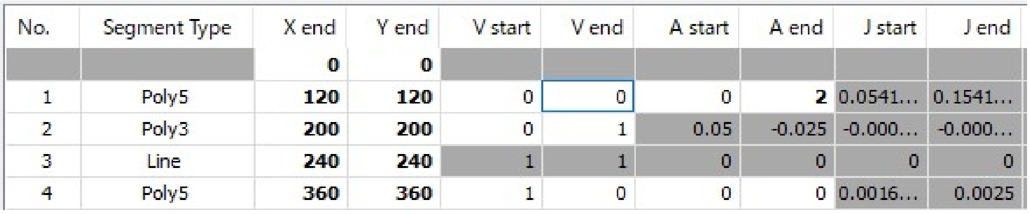
The change made to the node is reset. In the reset cell, an automatically calculated numerical value is displayed in fine font.

Note that Xend and Yend are not automatically calculated. Therefore, set them correctly.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	**0**						
1	Poly5	**120**	**120**	0	**1**	0	**2**	0.0525	0.1516...
2	Poly3	**200**	**200**	**1**	1	0	0	0	0
3	Line	**240**	**240**	1	1	0	0	0	0
4	Poly5	**360**	**360**	1	0	0	0	0.0016...	0.0025

No.1 V end
<Delete>



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	**0**						
1	Poly5	**120**	**120**	0	0	0	**2**	0.0541...	0.1541...
2	Poly3	**200**	**200**	0	1	0.05	-0.025	-0.000...	-0.000...
3	Line	**240**	**240**	1	1	0	0	0	0
4	Poly5	**360**	**360**	1	0	0	0	0.0016...	0.0025

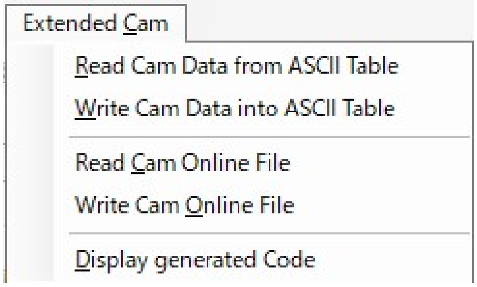
8. Tappet Settings

Since operations of the “Tappet” tab are the same as for the Cam editor, refer to “"12.2.1 Creating MC_CAM_REF by Cam Editor"”.

With the Cam editor, tappet and tappet table are displayed under separate tabs. With the Ectended Cam editor, they are displayed on the same tab.

12.2.3 File Import/Export Function

The file import/export function can be selected from the menu bar. When using the Cam editor, select the menu from the “Cam” menu, and when using the Ectended Cam editor, select it from the “Ectended Cam” menu.



A cam table or a tappet table created by the Cam editor or Ectended Cam editor can be saved and reused by exporting (writing) and importing (reading) files.

On the cam menu, you can use the following items.

Item	Description
Reading cam data from ASCII table	Reads cam data from a text file saved in ASCII table format to create a cam table. The number of cam tables is the number of arrays saved in ASCII table format.
Writing cam data to ASCII table	Saves cam data to a text file in ASCII table format. The maximum number of arrays that can be saved is 100,000. Details will be described later. Information on tappets is not saved.
Reading cam online file	Reads cam data and tappet data from a cam file saved in binary format to create a cam table and a tappet table. A waveform condition that was saved is restored. Cam files created by SMC_WriteCAM cannot be read.
Writing cam online file	Saves cam data and tappet data at the present to a cam file in binary format. Details will be described later. Any saved cam file cannot be read by SMC_ReadCAM.
Showing generated code	This function can be used to display MC_CAM_REF program code created by the Cam editor. The code can be reused in POUs or others by copying & pasting.

Info.

- ASCII table format file

An ASCII table format file can be used to read or write master position (X) and slave position (Y) values of a cam table created by the Cam editor. The number of stored points that can be specified is 100,000 or less, and X, Y values of a cam table that are divided by the number of stored points at equal intervals relative to the master position are saved. When an ASCII table format file is read, a cam table is created with the number of points specified when the file was saved.

Velocity, segment type, tappet table, and property settings are not saved. Thus, even when an ASCII table format file is read, a cam table that was written cannot be perfectly reproduced. Note that with an increase in the number of stored points, Cam editor operation speed gets slower despite improved accuracy with which the waveform is reproduced from a read file.

The extension of saved files is .txt. An example of a saved ASCII table format file is shown below.

```
0; 0
                    2.8346456692913384; 0.0092022399826652158
                    5.6692913385826769; 0.07123944761839393
                    ...
```

- Cam online file

A cam online file can be used to read or write cam tables, tappet tables, and information on each profile. By reading a cam online file, a cam table and a tappet table that were saved can be reproduced. To save a cam online file, the file can contain up to 100 cam tables (100 segments). For more than 100 cam tables (100 segments), the system does not operate properly.

Cam online files saved by the Cam editor will have extension of “.cam” and cam online files saved by Ectended Cam editor will have extension of “.gmcam”. Be careful that cam online files generated by the Cam editor or Ectended Cam editor are different as shown below.

If you edit the saved Cam online file using a text editor or similar software, it may no longer be correctly loaded by the GM Programmer.

	(*.cam)	(*.gmcam)
Cam editor	R/W (Possible to read or write.)	- (Not possible to read or write.)
Ectended Cam editor	R (Possible to read only.)	R/W (Possible to read or write.)
Information to be saved	● Input information of the “Cam table” tab (Note that the segment type is not saved.) ● Input information of the “Tappet table” tab ● * Input information of the “Properties” dialog box is not saved.	● Input information of the “Cam” tab ● Input information of the “Tappet” tab ● Input information of the “Cam” tab on the “Properties” dialog box ● Input information of the “Profile definitions” tab

12.3 Types and Setting of Cam Curves

The following table shows the types of cam curves (segment types) that connect a specified section.

Segment type	Characteristics
Line Linear	The slave velocity stays constant. Used to set a constant velocity section.
Poly3 Cubic polynomial	Used to smoothly connect to adjacent velocity curves.
Poly5 Quintic polynomial	Used to smoothly connect to adjacent velocity or acceleration curves.
Spline 4-3-4 beginning seg. Spline 4-3-4 (Start segment)	Used to create a free curve that is continuously interpolated with splines for the position, velocity, or acceleration.
Spline 4-3-4 Spline 4-3-4 (Middle segment)	
Spline 4-3-4 end seg. Spline 4-3-4 (End segment)	
Floating Trapezoid Floating trapezoid	Used to lower the maximum value of the speed or acceleration within the curve while smoothly connecting to adjacent velocity or acceleration curves.
Rotary Knife A Rotary Knife A	Used to create a cam curve of the rotary knife whose blade edge draws an arc.

The following table shows a summary of applicable items for each segment type.

Segment type	Applicable only to the expansion cam	Specifiable slave axis parameter					Compile format		
		Yend	Vstart	Vend	Astart	Aend	Type=1	Type=2	Type=3
Line Linear	-	○	-	-	-	-	○	○	○
Poly3 Cubic polynomial	○	○	○	○	-	-	○	○	○
Poly5 Quintic polynomial	-	○	○	○	○	○	○	○	○
Spline 4-3-4 beginning seg. Spline 4-3-4 (Start segment)	○	○	○	-	○	-	○	○	○
Spline 4-3-4 Spline 4-3-4 (Middle segment)	○	○	-	-	-	-	○	○	○
Spline 4-3-4 end seg. Spline 4-3-4 (End segment)	○	○	-	○	-	○	○	○	○
Floating Trapezoid Floating trapezoid	○	○	○	○	○	○	○	○	-

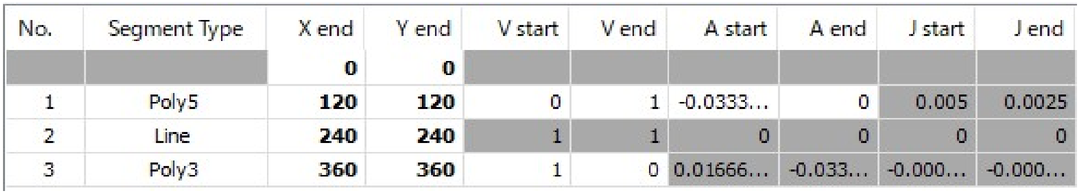
Segment type	Applicable only to the expansion cam	Specifiable slave axis parameter					Compile format		
		Yend	Vstart	Vend	Astart	Aend	Type=1	Type=2	Type=3
Rotary Knife A(Note 1) Rotary Knife A	○	○	-	-	-	-	○	○	-

(Note 1) It cannot be connected to other segments. It can be used only as a single segment.

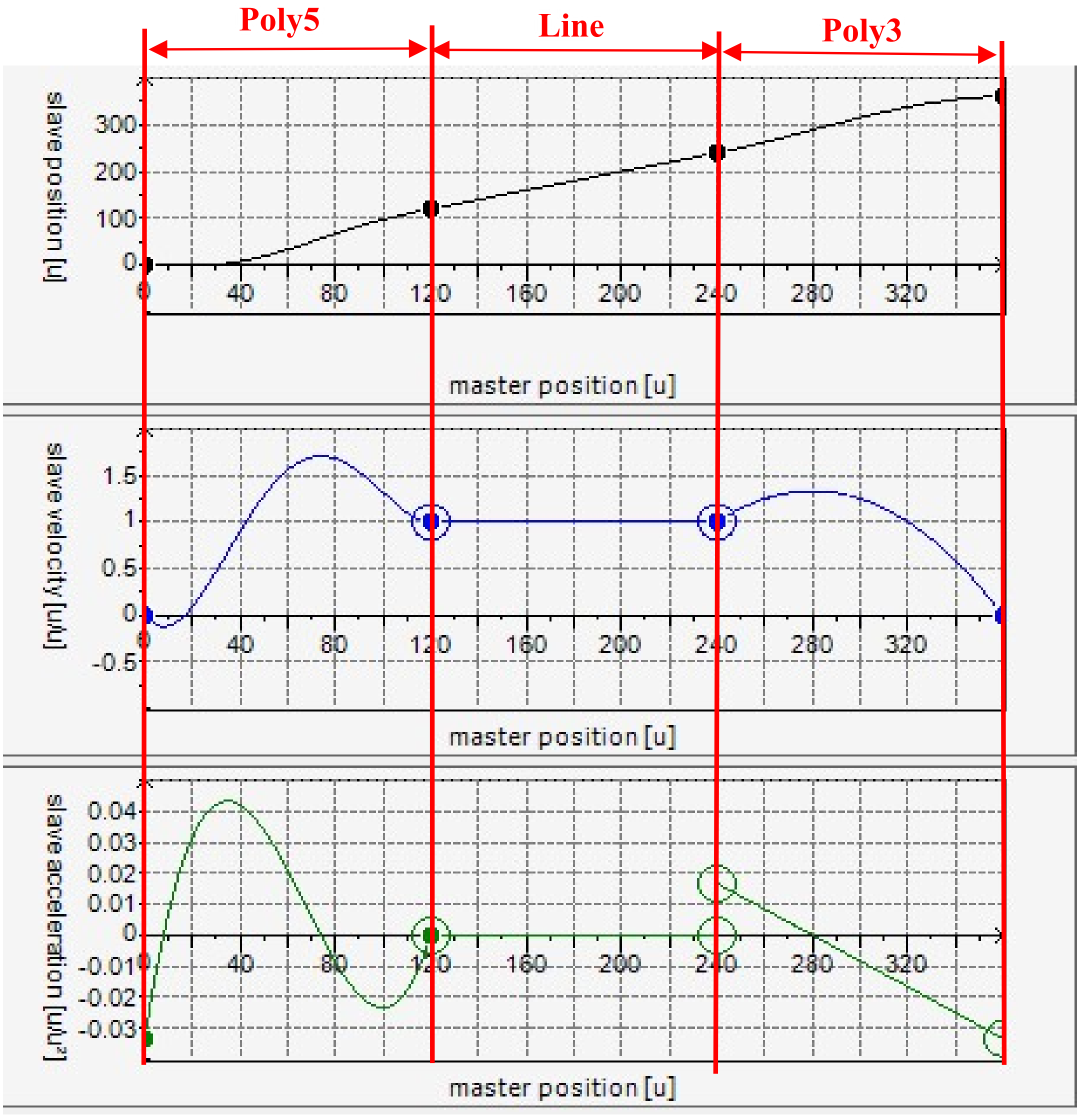
12.3.1 Line, Poly3, Poly5

A cam curve is generated by a single polynomial.

- Line is a first degree polynomial and keeps the velocity of the slave axis constant. Used to set a constant velocity section.
- Poly3 is a cubic polynomial and can be used to specify the velocity at both ends. Used to smoothly connect to adjacent velocity curves.
- Poly5 is a quintic polynomial and can be used to specify the velocity and acceleration at both ends. Used to smoothly connect to adjacent velocity or acceleration curves.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Poly5	120	120	0	1	-0.0333...	0	0.005	0.0025
2	Line	240	240	1	1	0	0	0	0
3	Poly3	360	360	1	0	0.01666...	-0.033...	-0.000...	-0.000...


Poly5
Line
Poly3
slave position [u]
300
200
100
0
0
40
80
120
160
200
240
280
320
master position [u]
slave velocity [u/u]
1.5
1
0.5
0
-0.5
0
40
80
120
160
200
240
280
320
master position [u]
slave acceleration [u/u²]
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
0
40
80
120
160
200
240
280
320
master position [u]

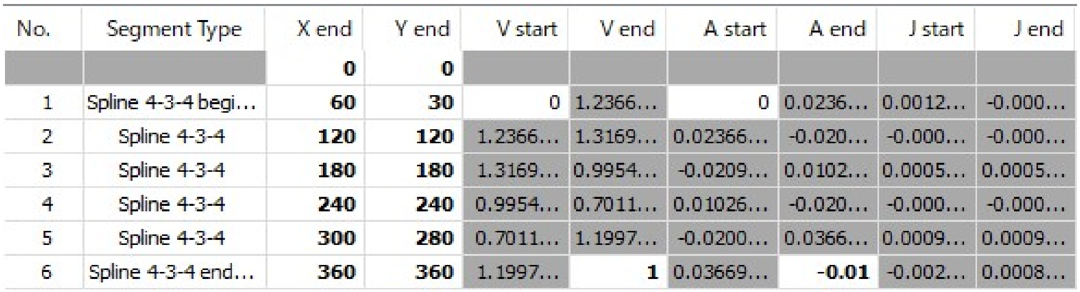
12.3.2 Spline 4-3-4

Used to create a free curve that is continuously interpolated with splines for the position, velocity, or acceleration.

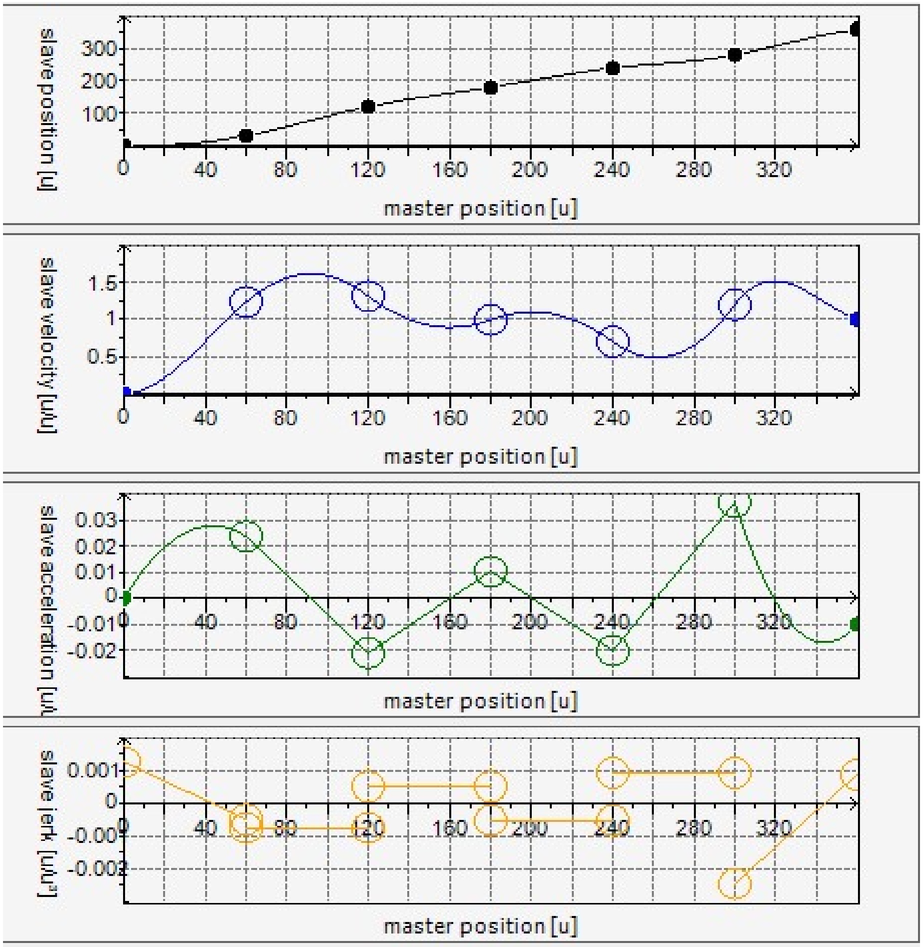
The curve is created by connecting multiple middle segments between the start segment and the end segment.

Since a continuous curve for the velocity or acceleration can be obtained without specifying the velocity or acceleration values, free curves can be created more easily than when connecting multiple Poly5 or Poly3 segments.

Since it is possible to specify the velocity or acceleration for the starting point of the start segment and the ending point of the end segment, you can smoothly connect to other curves.

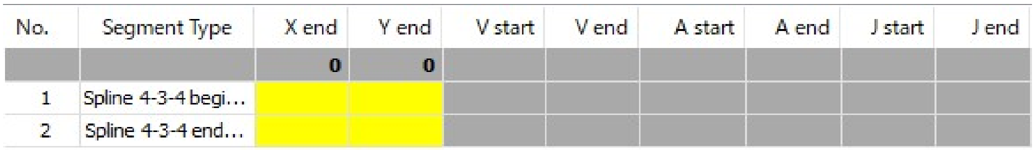


No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	**0**						
1	Spline 4-3-4 begi...	**60**	**30**	0	1.2366...	0	0.0236...	0.0012...	-0.000...
2	Spline 4-3-4	**120**	**120**	1.2366...	1.3169...	0.02366...	-0.020...	-0.000...	-0.000...
3	Spline 4-3-4	**180**	**180**	1.3169...	0.9954...	-0.0209...	0.0102...	0.0005...	0.0005...
4	Spline 4-3-4	**240**	**240**	0.9954...	0.7011...	0.01026...	-0.020...	-0.000...	-0.000...
5	Spline 4-3-4	**300**	**280**	0.7011...	1.1997...	-0.0200...	0.0366...	0.0009...	0.0009...
6	Spline 4-3-4 end...	**360**	**360**	1.1997...	**1**	0.03669...	**-0.01**	-0.002...	0.0008...



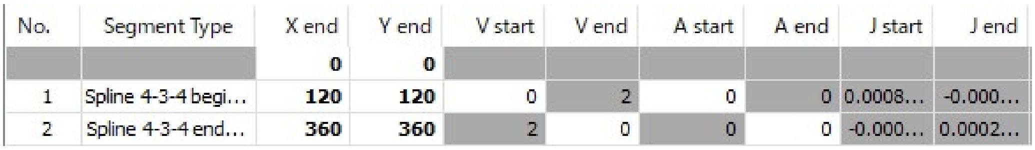
Procedure

1. If you select "Spline 4-3-4” on the dropdown list for the segment type, one start segment and one end segment are automatically generated.



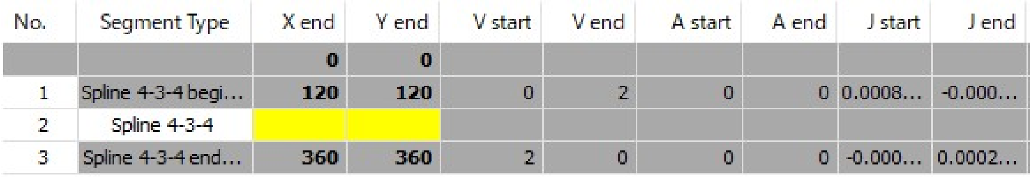
No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Spline 4-3-4 begi…								
2	Spline 4-3-4 end…								

2. Enter the Xend and Yend values for the start segment and the end segment to make the segments in the defined state.



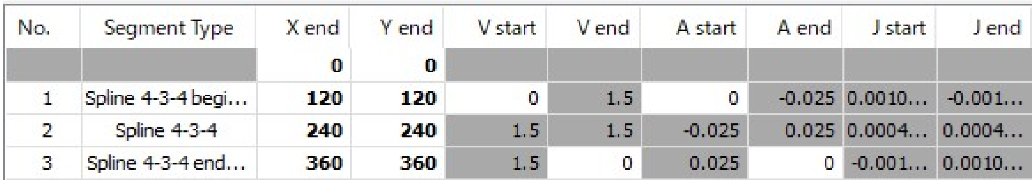
No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Spline 4-3-4 begi…	120	120	0	2	0	0	0.0008…	-0.000…
2	Spline 4-3-4 end…	360	360	2	0	0	0	-0.000…	0.0002…

3. If you add a new segment between the start segment and end segment, a middle segment is generated.



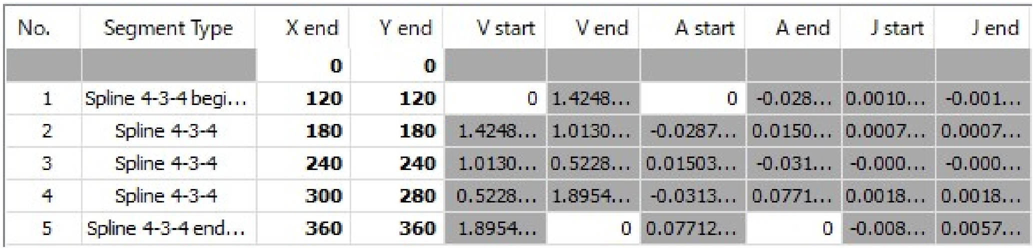
No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Spline 4-3-4 begi...	120	120	0	2	0	0	0.0008...	-0.000...
2	Spline 4-3-4								
3	Spline 4-3-4 end...	360	360	2	0	0	0	-0.000...	0.0002...

4. Enter the Xend and Yend values for the middle segment to make the segment in the defined state.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Spline 4-3-4 begi...	120	120	0	1.5	0	-0.025	0.0010...	-0.001...
2	Spline 4-3-4	240	240	1.5	1.5	-0.025	0.025	0.0004...	0.0004...
3	Spline 4-3-4 end...	360	360	1.5	0	0.025	0	-0.001...	0.0010...

5. In the same procedure, you can increase middle segments.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Spline 4-3-4 begi...	120	120	0	1.4248...	0	-0.028...	0.0010...	-0.001...
2	Spline 4-3-4	180	180	1.4248...	1.0130...	-0.0287...	0.0150...	0.0007...	0.0007...
3	Spline 4-3-4	240	240	1.0130...	0.5228...	0.01503...	-0.031...	-0.000...	-0.000...
4	Spline 4-3-4	300	280	0.5228...	1.8954...	-0.0313...	0.0771...	0.0018...	0.0018...
5	Spline 4-3-4 end...	360	360	1.8954...	0	0.07712...	0	-0.008...	0.0057...

6. The following table shows behaviors when a segment is deleted or when the segment type is changed.

Target	Deletion of segment	Change of segment type
Start segment	The start segment is deleted and the adjacent middle segment becomes a new start segment.	The change is reflected and the adjacent middle segment becomes a new start segment.
Middle segment	The selected middle segment is deleted.	Not applicable
End segment	The end segment is deleted and the adjacent middle segment becomes a new end segment.	The change is reflected and the adjacent middle segment becomes a new end segment.

12.3.3 Floating Trapezoid

You can specify the velocity and acceleration at both ends. In addition, you can provide a sub section where the velocity is constant or acceleration is constant within the curve.

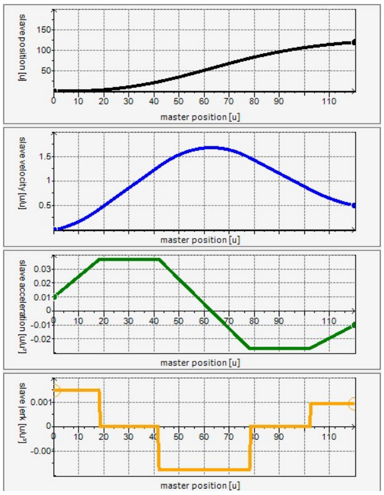
Used to lower the maximum value of the speed or acceleration within the curve while smoothly connecting to adjacent velocity or acceleration curves.

Set the number of divisions (5 divisions/7 divisions) of sub sections or the length using the Set button at the top of the table area.

The specifications and waveform examples of the sub sections with 5 divisions and 7 divisions are shown below.

5 divisions

< Waveform examples >



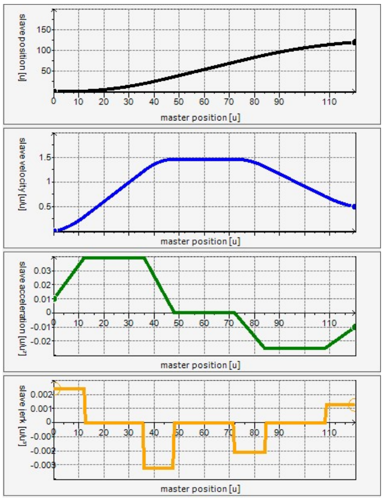
< Specifications of each sub section >

If we name divisions of sub sections as I, II, III, IV, and V in the order starting from the lowest master axis position, the divisions are as follows:

- I, III, and V: The slave axis acceleration makes a monotonic increase or monotonic decrease.
- II and IV: The slave axis acceleration stays constant.

7 divisions

< Waveform examples >



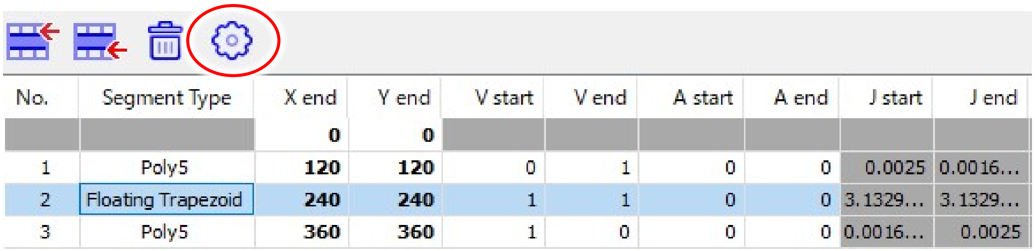
< Specifications of each sub section >

If we name divisions of sub sections as I, II, III, IV, V, VI, and VII in the order starting from the lowest master axis position, the divisions are as follows:

- I, III, V, and VII: The slave axis acceleration makes a monotonic increase or monotonic decrease.
- II and VI: The slave axis acceleration stays constant.
- IV: The slave axis acceleration is zero, that is, the slave axis velocity stays constant.

1 2 Procedure

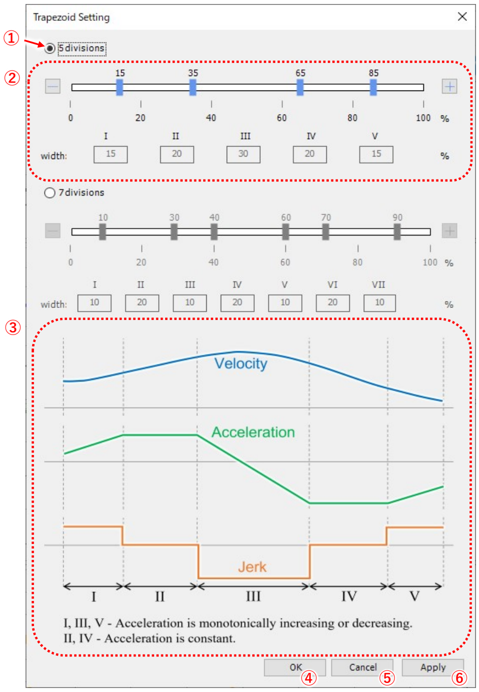
1. Specify "Floating Trapezoid” on the dropdown list for the segment type. When the "Floating Trapezoid” row is selected, the Set button at the top of the table area becomes active.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Poly5	120	120	0	1	0	0	0.0025	0.0016...
2	Floating Trapezoid	240	240	1	1	0	0	3.1329...	3.1329...
3	Poly5	360	360	1	0	0	0	0.0016...	0.0025

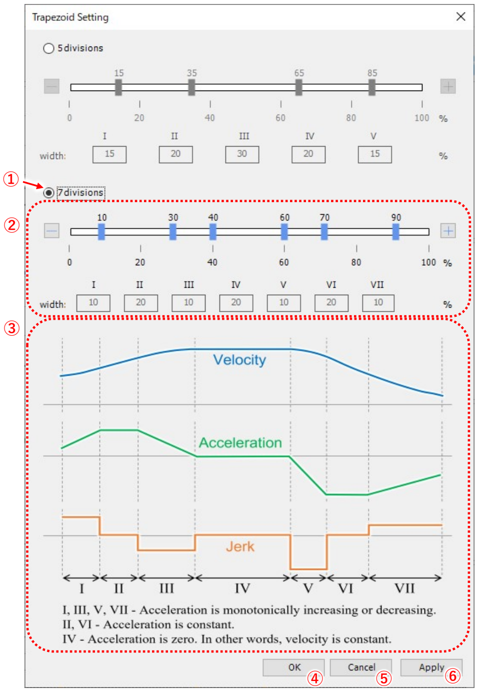
2. Click the Set button, and the “Trapezoid Setting” dialog box will be displayed. Set the number of divisions of sub sections and the length of each sub section. In conjunction with the operation of the dialog box, the cam curve in the graph area is updated.

5 division selection



I, III, V - Acceleration is monotonically increasing or decreasing.
II, IV - Acceleration is constant.

7 division selection



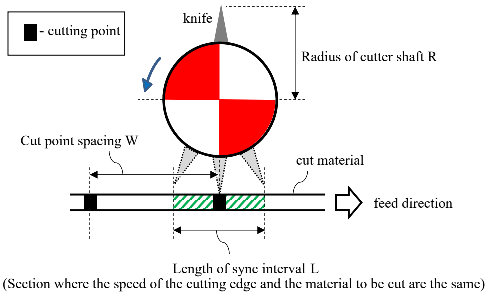
I, III, V, VII - Acceleration is monotonically increasing or decreasing.
II, VI - Acceleration is constant.
IV - Acceleration is zero. In other words, velocity is constant.

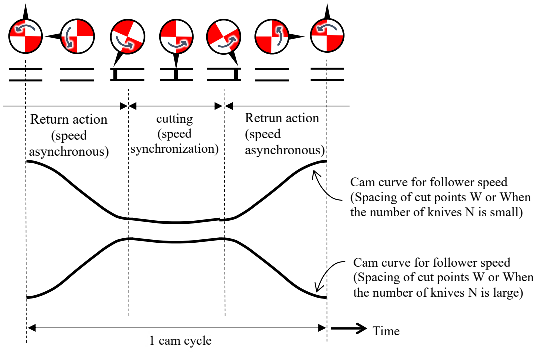
Dialog item	Description
① Division setting (radio button)	Select how many sub sections to create by dividing one segment. It is possible to select either 5 divisions or 7 divisions.
② Split setting	Adjust the length of the sub section according to either one of the following operations. ・ Drag the slider. ・ Enter a numerical value in the “Width” field. The length of the sub section is displayed in a numerical value in the “Width” field. Left-click the slider, and the “Width” fields for the two sub sections divided by the slider will become active to allow you to input numerical values.
③ Schematic diagram	A schematic diagram that shows shape specifications of the sub section is displayed. The diagram switches between 5 divisions and 7 divisions as selected.
④ "OK” button	Saves the settings and then closes the dialog box.
⑤ "Cancel” button	Closes the dialog box without saving the settings.
⑥ "Apply” button	Saves the settings.

12.3.4 Creating Rotary Knife A

Used for cam operation where a rotary knife cuts the material moved in one direction.

A cam curve is generated so that the blade edge velocity becomes the same as the cut material velocity in the vicinity of the cutting point. The master axis position is at a position in the feed direction of the cut material and the slave axis position corresponds to the rotation angle of the rotary knife. The interval between one cutting and the next corresponds to one cycle of the cam curve.

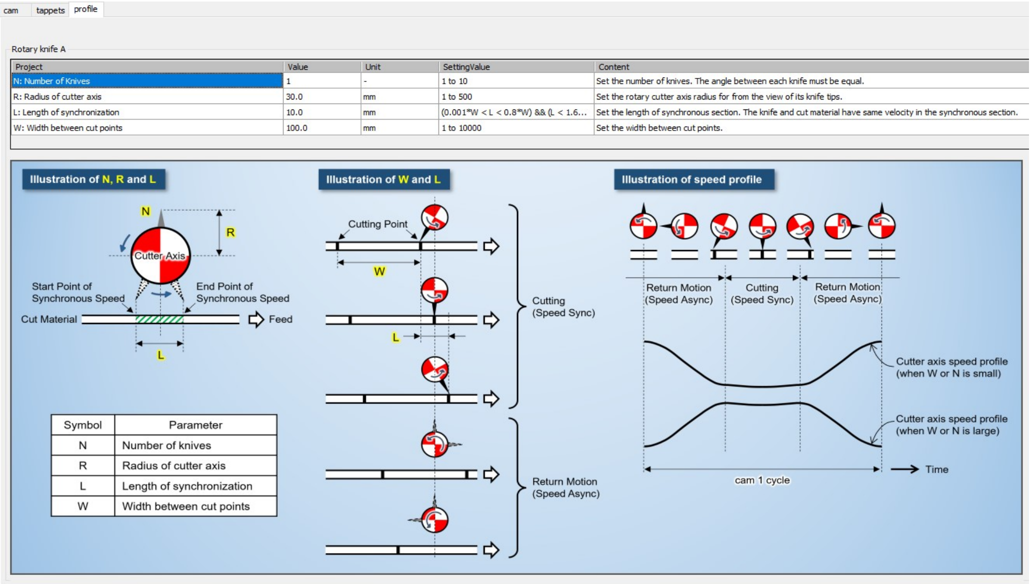

■ - cutting point
knife
Radius of cutter shaft R
Cut point spacing W
cut material
feed direction
Length of sync interval L
(Section where the speed of the cutting edge and the material to be cut are the same)



At the midpoint of the cam curve, the blade edge reaches the bottom dead center (a position where the blade cuts the material at the right angle). At the start point and end point of the cam curve, the blade edge is farthest from the bottom dead center.

1 2 Procedure

1. On the "Profile Definition” tab, enter the mechanical dimension parameters of the rotary knife to edit the waveform.



Rotary knife A

Project	Value	Unit	SettingValue	Content
N: Number of Knives	1	-	1 to 10	Set the number of knives. The angle between each knife must be equal.
R: Radius of cutter axis	30.0	mm	1 to 500	Set the rotary cutter axis radius for from the view of its knife tips.
L: Length of synchronization	10.0	mm	(0.001*W < L < 0.8*W) && (L < 1.6...	Set the length of synchronous section. The knife and cut material have same velocity in the synchronous section.
W: Width between cut points	100.0	mm	1 to 10000	Set the width between cut points.

Illustration of N, R and L
N
R
Cutter Axis
Start Point of Synchronous Speed
End Point of Synchronous Speed
Cut Material
Feed
L
Illustration of W and L
Cutting Point
W
L
Cutting (Speed Sync)
Return Motion (Speed Async)
Illustration of speed profile
Return Motion (Speed Async)
Cutting (Speed Sync)
Return Motion (Speed Async)
Cutter axis speed profile (when W or N is small)
Cutter axis speed profile (when W or N is large)
cam 1 cycle
Time

Symbol	Parameter
N	Number of knives
R	Radius of cutter axis
L	Length of synchronization
W	Width between cut points

Rotary Knife A setting parameters (mechanical dimension parameters of the rotary knife)

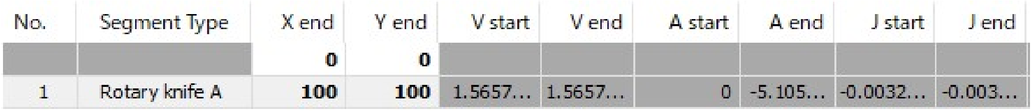
Setting item	Unit	Setting range	Description
N: Number of knives	-	1 to 10	Set the number of knives. The angle between knives must be equal.

Setting item	Unit	Setting range	Description
R: Radius of the cutter shaft	mm	1 to 500	Set the cutter shaft radius viewed from the knife tip.
L: Length of the synchronous section	mm	(0.001*W < L < 0.8*W) && (L < 1.6*R)	Specify the length of the synchronous section. The knife velocity becomes equal to that of the cut material in this synchronous section.
W: Interval between cut points	mm	1 to 10000	Set the interval between two cut points.

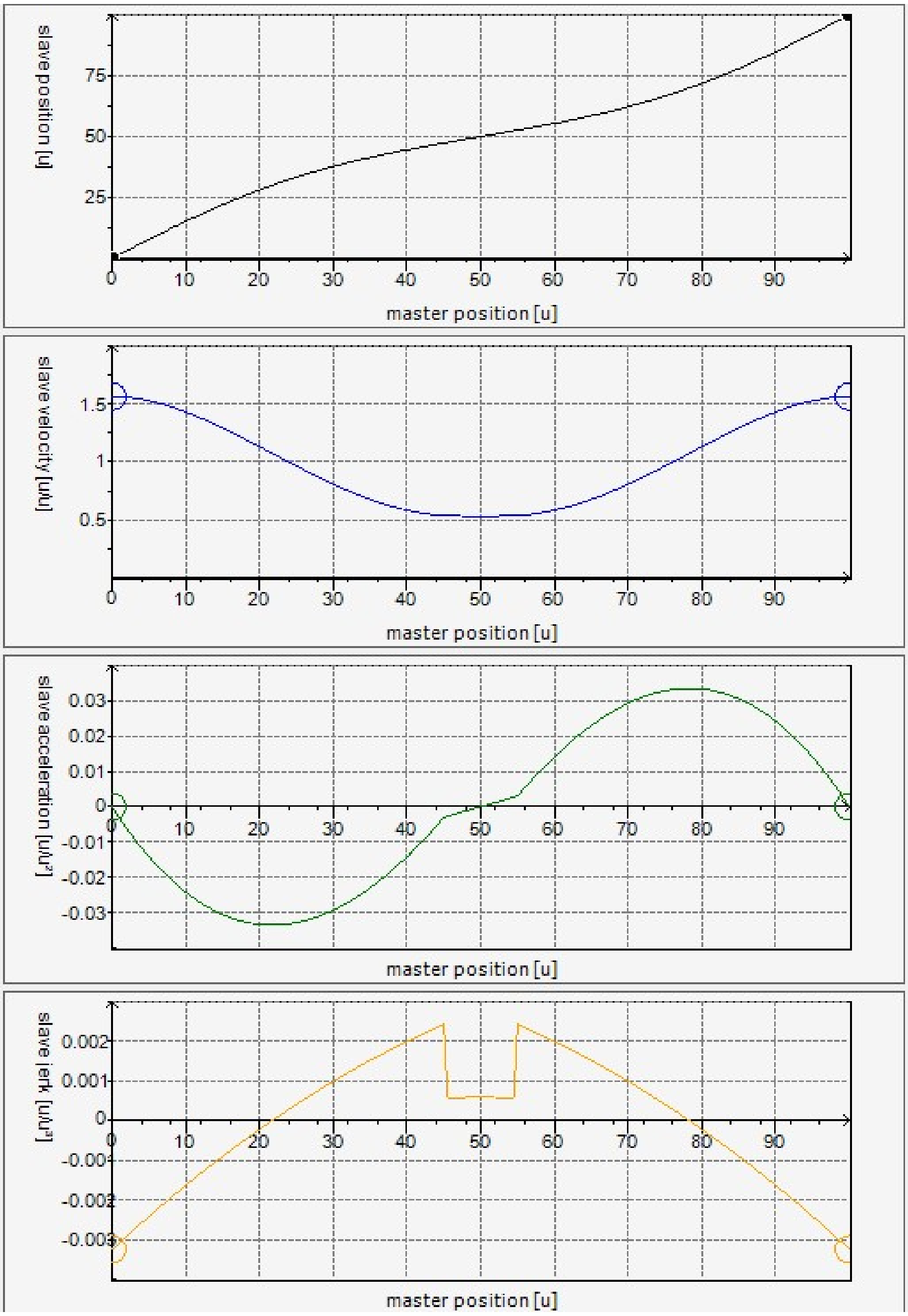
When the entered parameter value is outside the set range, [!] is displayed next to the corresponding parameter. At this time, the parameter is not reflected internally, so enter a correct value.

2. Select "Rotary Knife A” on the dropdown list for the segment type.

   An entire cam curve is generated as a single segment. It cannot be combined with other segments.



No.	Segment Type	X end	Y end	V start	V end	A start	A end	J start	J end
		0	0						
1	Rotary knife A	100	100	1.5657...	1.5657...	0	-5.105...	-0.0032...	-0.003...


slave position [u]
75
50
25
0
10
20
30
40
50
60
70
80
90
master position [u]
slave velocity [u/u]
1.5
1
0.5
0
10
20
30
40
50
60
70
80
90
master position [u]
slave acceleration [u/u²]
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
0
10
20
30
40
50
60
70
80
90
master position [u]
slave jerk [u/u³]
0.002
0.001
0
-0.001
-0.002
-0.003
0
10
20
30
40
50
60
70
80
90
master position [u]

12.4 Cam Control (POU Programming)

With the GM1 controller, through use of function blocks, you can perform cam synchronous control of the axes using cam objects.

1 2 Procedure

1. Execute MC_CamTableSelect to specify the master axis, the slave axis, and a cam table (MC_CAM_REF) that is converted into executable format.
2. When MC_CamIn is executed, the slave axis moves in response to the master position in accordance with the cam profile and gets synchronized.
3. To release the cam synchronization, execute MC_CamOut.
4. Moreover, using auxiliary function blocks for cam synchronous control, you can monitor tappet information, check minimum/ maximum values of the slave velocity and acceleration from the cam profile before operation, and read or write cam data.

Info.

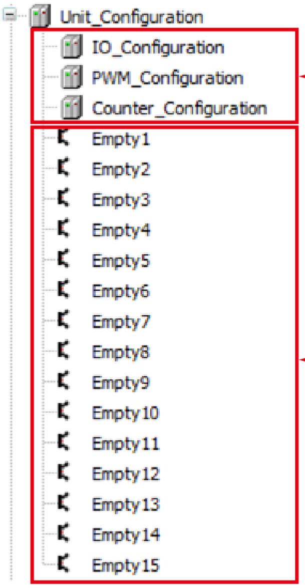
- For details on each function block, refer to the GM1 Series Reference Manual (Instruction Edition).

13 Unit Control

This chapter explains unit control for the GM1 controller.

13.1 Overview of Unit Control

Unit control provides control for the controller functions (I/O, PWM, and counter) and expansion units.



Unit_Configuration
IO_Configuration
PWM_Configuration
Counter_Configuration
Empty1
Empty2
Empty3
Empty4
Empty5
Empty6
Empty7
Empty8
Empty9
Empty10
Empty11
Empty12
Empty13
Empty14
Empty15

Controller functions (IO, PWM, and Counter)

Expansion units

13.2 IO Parameters for Unit Control

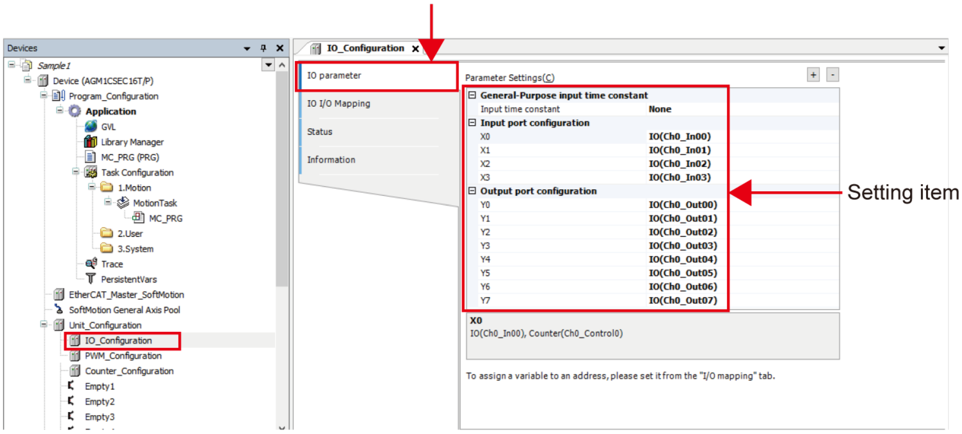
The parameter window is used to set up parameters for the controller functions (I/O, PWM, and counter) and each expansion unit.

This section explains I/O mapping, using IO_Configuration as an example.

In the navigator pane, double-click the "IO_Configuration" object. The "IO_Configuration" pane will be displayed.

In the "IO_Configuration" pane, click the "IO Parameters" tab. The parameter settings sub-pane will be displayed. Change the settings according to your need.

Select the IO Parameters tab



⊟ General-Purpose input time constant	
Input time constant	None
⊟ Input port configuration	
X0	IO(Ch0_In00)
X1	IO(Ch0_In01)
X2	IO(Ch0_In02)
X3	IO(Ch0_In03)
⊟ Output port configuration	
Y0	IO(Ch0_Out00)
Y1	IO(Ch0_Out01)
Y2	IO(Ch0_Out02)
Y3	IO(Ch0_Out03)
Y4	IO(Ch0_Out04)
Y5	IO(Ch0_Out05)
Y6	IO(Ch0_Out06)
Y7	IO(Ch0_Out07)

X0
IO(Ch0_In00), Counter(Ch0_Control0)

To assign a variable to an address, please set it from the "I/O mapping" tab.

Setting item

13.3 I/O Mapping for Unit Control

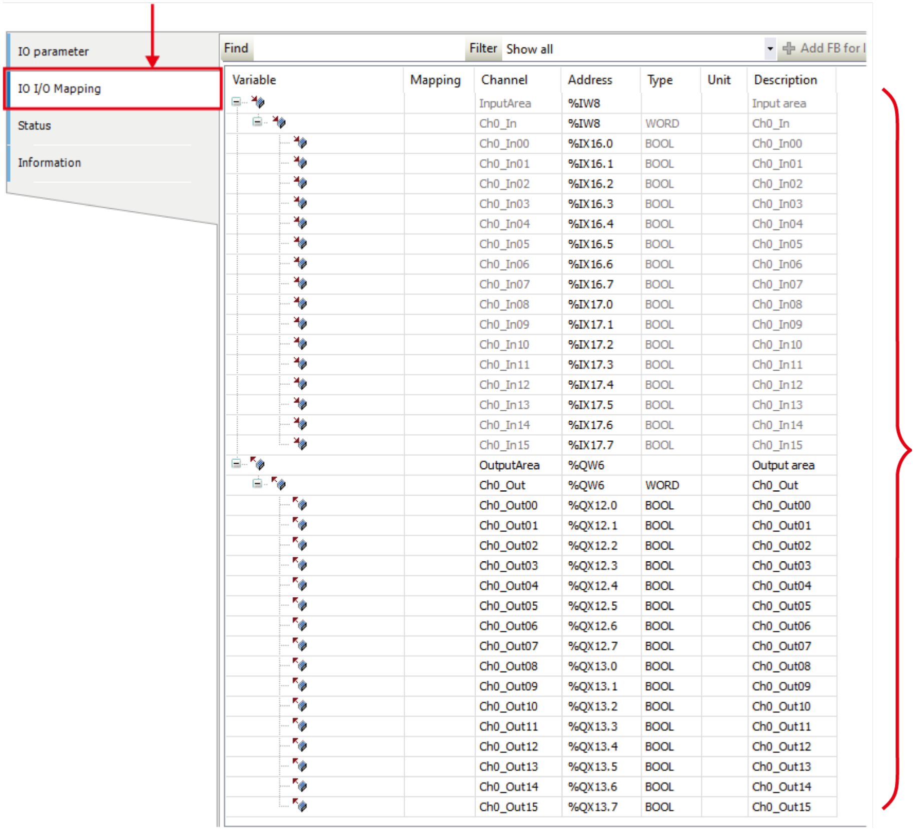
The I/O mapping window is used to allocate variables to I/O mapping for the controller functions (I/O, PWM, and counter) and each expansion unit.

This section explains I/O mapping, using IO_Configuration as an example.

Click the "I/O Mapping" tab in the Device pane. The I/O mapping pane will be displayed.

Register variables directly from the variable or I/O mapping window declared in the Application object.

Select the I/O Mapping tab



Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%IW8			Input area
		Ch0_In	%IW8	WORD		Ch0_In
		Ch0_In00	%IX16.0	BOOL		Ch0_In00
		Ch0_In01	%IX16.1	BOOL		Ch0_In01
		Ch0_In02	%IX16.2	BOOL		Ch0_In02
		Ch0_In03	%IX16.3	BOOL		Ch0_In03
		Ch0_In04	%IX16.4	BOOL		Ch0_In04
		Ch0_In05	%IX16.5	BOOL		Ch0_In05
		Ch0_In06	%IX16.6	BOOL		Ch0_In06
		Ch0_In07	%IX16.7	BOOL		Ch0_In07
		Ch0_In08	%IX17.0	BOOL		Ch0_In08
		Ch0_In09	%IX17.1	BOOL		Ch0_In09
		Ch0_In10	%IX17.2	BOOL		Ch0_In10
		Ch0_In11	%IX17.3	BOOL		Ch0_In11
		Ch0_In12	%IX17.4	BOOL		Ch0_In12
		Ch0_In13	%IX17.5	BOOL		Ch0_In13
		Ch0_In14	%IX17.6	BOOL		Ch0_In14
		Ch0_In15	%IX17.7	BOOL		Ch0_In15
		OutputArea	%QW6			Output area
		Ch0_Out	%QW6	WORD		Ch0_Out
		Ch0_Out00	%QX12.0	BOOL		Ch0_Out00
		Ch0_Out01	%QX12.1	BOOL		Ch0_Out01
		Ch0_Out02	%QX12.2	BOOL		Ch0_Out02
		Ch0_Out03	%QX12.3	BOOL		Ch0_Out03
		Ch0_Out04	%QX12.4	BOOL		Ch0_Out04
		Ch0_Out05	%QX12.5	BOOL		Ch0_Out05
		Ch0_Out06	%QX12.6	BOOL		Ch0_Out06
		Ch0_Out07	%QX12.7	BOOL		Ch0_Out07
		Ch0_Out08	%QX13.0	BOOL		Ch0_Out08
		Ch0_Out09	%QX13.1	BOOL		Ch0_Out09
		Ch0_Out10	%QX13.2	BOOL		Ch0_Out10
		Ch0_Out11	%QX13.3	BOOL		Ch0_Out11
		Ch0_Out12	%QX13.4	BOOL		Ch0_Out12
		Ch0_Out13	%QX13.5	BOOL		Ch0_Out13
		Ch0_Out14	%QX13.6	BOOL		Ch0_Out14
		Ch0_Out15	%QX13.7	BOOL		Ch0_Out15

Allocate variables

13.4 General-purpose I/O

13.4.1 Overview of General-purpose I/O Function

The general-purpose I/O function allows use of up to 16 input points and 16 output points.

* However, the general-purpose I/O function shares some ports with the high-speed counter function and PWM output function. Therefore, use IO parameter settings to select functions to be used.

■ Specification overview

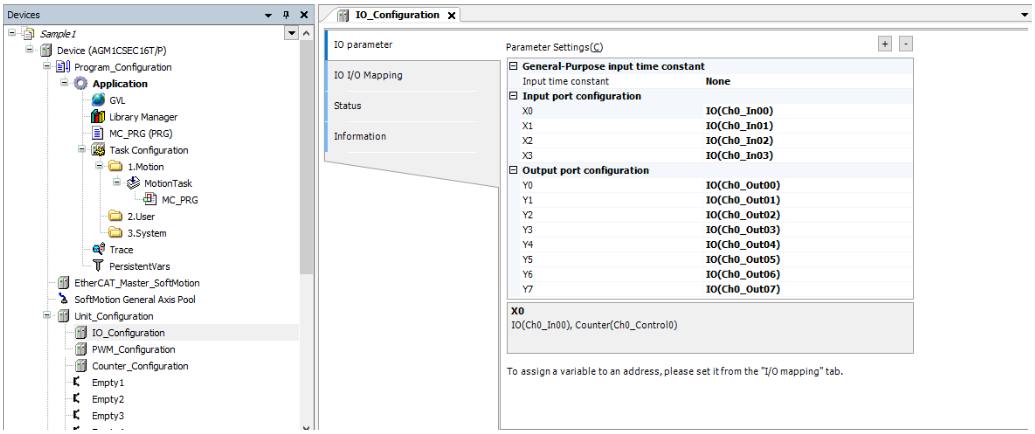
Item	Specifications	Remarks
Number of input ports	Max. 16 points	X0 to X3 are shared with the counter function. Use IO parameter settings to select functions to be used.
Number of output ports	Max. 16 points	Y0 to Y3 are shared with the counter function and Y4 to Y7 are shared with the PWM output function. Use IO parameter settings to select functions to be used.
Input port number	X0 to X15	
Output port number	Y0 to Y15	

13.4.2 Setting Parameters with GM Programmer

1 2 Procedure

1. In the navigator pane, double-click the "IO_Configuration" object.

The general-purpose IO setting pane will be displayed.



Parameter Settings(C)

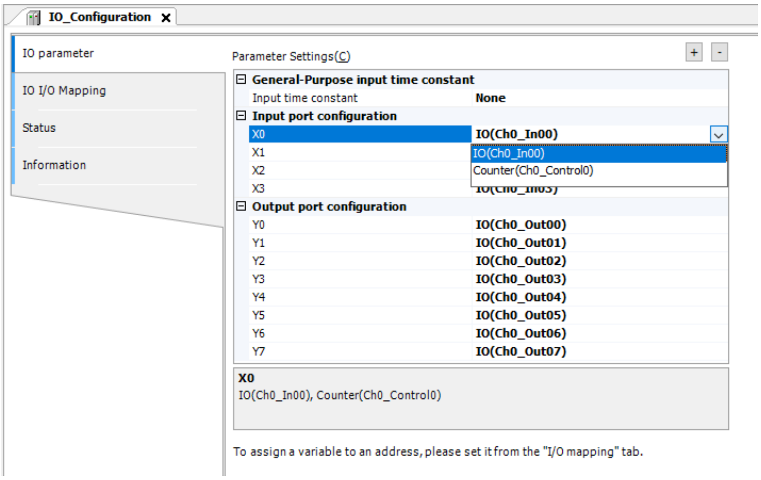
⊟ **General-Purpose input time constant**	
Input time constant	**None**
⊟ **Input port configuration**	
X0	**IO(Ch0_In00)**
X1	**IO(Ch0_In01)**
X2	**IO(Ch0_In02)**
X3	**IO(Ch0_In03)**
⊟ **Output port configuration**	
Y0	**IO(Ch0_Out00)**
Y1	**IO(Ch0_Out01)**
Y2	**IO(Ch0_Out02)**
Y3	**IO(Ch0_Out03)**
Y4	**IO(Ch0_Out04)**
Y5	**IO(Ch0_Out05)**
Y6	**IO(Ch0_Out06)**
Y7	**IO(Ch0_Out07)**

X0
IO(Ch0_In00), Counter(Ch0_Control0)

To assign a variable to an address, please set it from the "I/O mapping" tab.

2. Set up general-purpose IO parameters.

Select an IO parameter to be changed and then select a desired item from the drop-down list.



Parameter Settings(C)

⊟ **General-Purpose input time constant**	
Input time constant	**None**
⊟ **Input port configuration**	
X0	**IO(Ch0_In00)**

X1	IO(Ch0_In00)
	Counter(Ch0_Control0)

X2	
X3	**IO(Ch0_In03)**
⊟ **Output port configuration**	
Y0	**IO(Ch0_Out00)**
Y1	**IO(Ch0_Out01)**
Y2	**IO(Ch0_Out02)**
Y3	**IO(Ch0_Out03)**
Y4	**IO(Ch0_Out04)**
Y5	**IO(Ch0_Out05)**
Y6	**IO(Ch0_Out06)**
Y7	**IO(Ch0_Out07)**

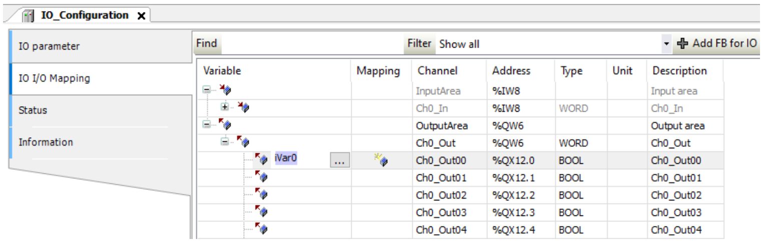
X0
IO(Ch0_In00), Counter(Ch0_Control0)

To assign a variable to an address, please set it from the "I/O mapping" tab.

3. Select the "IO I/O Mapping" tab and set the correspondence (mapping) between the channel and variable in the mapping setting pane.

   Click the "Variable" column corresponding to the channel to be used by the program and enter a variable name.

   Clicking the mark in the "Mapping" column allows you to change the type of mapping.



Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%IW8			Input area
		Ch0_In	%IW8	WORD		Ch0_In
		OutputArea	%QW6			Output area
		Ch0_Out	%QW6	WORD		Ch0_Out
iVar0		Ch0_Out00	%QX12.0	BOOL		Ch0_Out00
		Ch0_Out01	%QX12.1	BOOL		Ch0_Out01
		Ch0_Out02	%QX12.2	BOOL		Ch0_Out02
		Ch0_Out03	%QX12.3	BOOL		Ch0_Out03
		Ch0_Out04	%QX12.4	BOOL		Ch0_Out04

Info.

- You can copy the variable name set in the Channel column.

  Select a channel (CH0 or CH1) in the "Category Selection" column, right-click, and then select [Copy] from the context-sensitive menu that is displayed. Next, select another channel, right-click, and then select [Paste] from the context-sensitive menu that is displayed.

13.4.3 Setting Items of IO_Configuration Parameters

■ IO parameters

Setting item	Setting item	Settings	Default value	Description
General-purpose input time constant	Input time constant	None 0.1 ms 0.5 ms 1 ms 5 ms 10 ms 20 ms 70 ms	None	Input time constant

Input function setting	X0	IO(Ch0_In00)
		Counter(Ch0_Control0)
			IO(Ch0_In00)	Select X0

	X1	IO(Ch0_In01)
		Counter(Ch0_Control1)
			IO(Ch0_In01)	Select X1

	X2	IO(Ch0_In02)
		Counter(Ch1_Control0)
			IO(Ch0_In02)	Select X2

	X3	IO(Ch0_In03)
		Counter(Ch1_Control1)
			IO(Ch0_In03)	Select X3

Output function setting	Y0	IO(Ch0_Out00)
		Counter(Ch0_ExternalOutput0)
			IO(Ch0_Out00)	Select Y0

	Y1	IO(Ch0_Out01)
		Counter(Ch0_ExternalOutput1)
			IO(Ch0_Out01)	Select Y1

	Y2	IO(Ch0_Out02)
		Counter(Ch1_ExternalOutput0)
			IO(Ch0_Out02)	Select Y2

	Y3	IO(Ch0_Out03)
		Counter(Ch1_ExternalOutput1)
			IO(Ch0_Out03)	Select Y3

	Y4	IO(Ch0_Out04)
		PWM(Ch0_PWM_Output)
			IO(Ch0_Out04)	Select Y4

	Y5	IO(Ch0_Out05)
		PWM(Ch1_PWM_Output)
			IO(Ch0_Out05)	Select Y5

	Y6	IO(Ch0_Out06)
		PWM(Ch2_PWM_Output)
			IO(Ch0_Out06)	Select Y6

	Y7	IO(Ch0_Out07)
		PWM(Ch3_PWM_Output)
			IO(Ch0_Out07)	Select Y07

13.4.4 I/O Mapping for General-purpose I/O

Channel	Type	Description	Remarks
Ch0_In	WORD	Ch0_In	
Ch0_Out	WORD	Ch0_Out	

■ Ch0_In

Channel	Type	Description	Remarks
Ch0_In00	BOOL	Ch0_In00	
Ch0_In01	BOOL	Ch0_In01	
Ch0_In02	BOOL	Ch0_In02	
Ch0_In03	BOOL	Ch0_In03	
Ch0_In04	BOOL	Ch0_In04	
Ch0_In05	BOOL	Ch0_In05	
Ch0_In06	BOOL	Ch0_In06	
Ch0_In07	BOOL	Ch0_In07	
Ch0_In08	BOOL	Ch0_In08	
Ch0_In09	BOOL	Ch0_In09	
Ch0_In10	BOOL	Ch0_In10	
Ch0_In11	BOOL	Ch0_In11	
Ch0_In12	BOOL	Ch0_In12	
Ch0_In13	BOOL	Ch0_In13	
Ch0_In14	BOOL	Ch0_In14	
Ch0_In15	BOOL	Ch0_In15	

■ Ch0_Out

Channel	Type	Description	Remarks
Ch0_Out00	BOOL	Ch0_Out00	
Ch0_Out01	BOOL	Ch0_Out01	
Ch0_Out02	BOOL	Ch0_Out02	
Ch0_Out03	BOOL	Ch0_Out03	
Ch0_Out04	BOOL	Ch0_Out04	
Ch0_Out05	BOOL	Ch0_Out05	
Ch0_Out06	BOOL	Ch0_Out06	
Ch0_Out07	BOOL	Ch0_Out07	
Ch0_Out08	BOOL	Ch0_Out08	
Ch0_Out09	BOOL	Ch0_Out09	
Ch0_Out10	BOOL	Ch0_Out10	
Ch0_Out11	BOOL	Ch0_Out11	

Channel	Type	Description	Remarks
Ch0_Out12	BOOL	Ch0_Out12	
Ch0_Out13	BOOL	Ch0_Out13	
Ch0_Out14	BOOL	Ch0_Out14	
Ch0_Out15	BOOL	Ch0_Out15	

13.5 PWM Output

13.5.1 Overview of PWM Output

The PWM output function enables up to 100 kHz of PWM output to be obtained within a range of 0% to 100%.

■ Outline of specifications

Item	Specifications	Remarks
Number of output channels	Max. 4 channels	
Output port number	Y4 to Y7	
Output frequency	1 Hz to100 kHz (Settable by 1 Hz)(Note 1)	
Output duty ratio	0% to 100% (Settable by 0.1%)	
Control input	Enable request or start request	

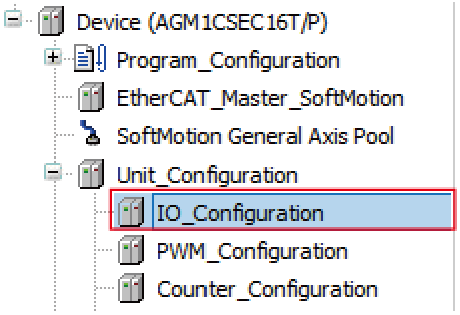
(Note 1) This specification applies when push-pull is set and output current is 0.1 A. It varies according to loads.

13.5.2 Setting Output Ports with GM Programmer

You can set output ports for PWM output via **IO_Configuration**>**Edit Object** in GM Programmer.

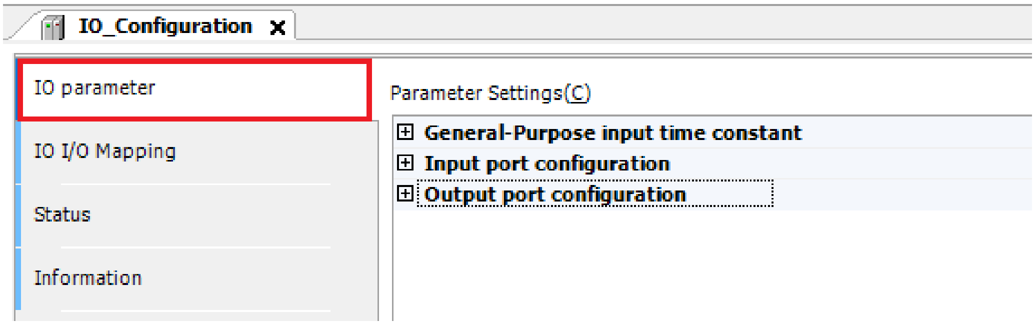
1 2 Procedure

1. From "Device view" in the navigator pane, double-click "IO_Configuration".

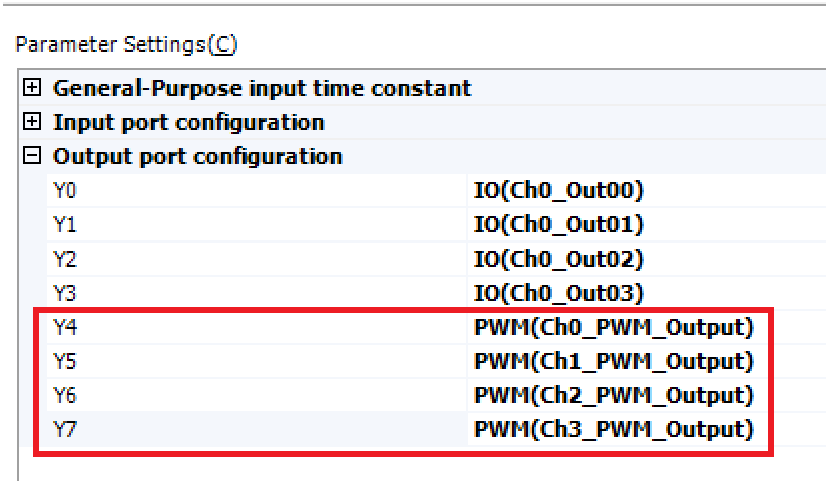


```
Device (AGM1CSEC16T/P)
  Program_Configuration
  EtherCAT_Master_SoftMotion
  SoftMotion General Axis Pool
  Unit_Configuration
    IO_Configuration
    PWM_Configuration
    Counter_Configuration
```

2. Click the "IO Parameters" tab.



3. If necessary, change Y4 to Y7 in "Output port configuration" to PWM(Ch0_PWM_Output) to PWM(Ch3_PWM_Output), respectively.


Parameter Settings(C)

⊞ General-Purpose input time constant
⊞ Input port configuration
⊟ Output port configuration

Y0	IO(Ch0_Out00)
Y1	IO(Ch0_Out01)
Y2	IO(Ch0_Out02)
Y3	IO(Ch0_Out03)
Y4	PWM(Ch0_PWM_Output)
Y5	PWM(Ch1_PWM_Output)
Y6	PWM(Ch2_PWM_Output)
Y7	PWM(Ch3_PWM_Output)

13.5.3 I/O Mapping for PWM Output

The high-speed counter function is controlled by user programs.

■ InputArea (input area)

Channel	Type	Description	Remarks
PwmStatusRegister	WORD	Input area	-

■ PwmStatusRegister (PWM status register)

Channel	Type	Description	Remarks
Ch*_PwmStatus	BOOL	Ch* PWM output status	Indicates the PWM output state. FALSE: OFF TRUE: ON

■ OutputArea (output area)

Channel	Type	Description	Remarks
PwmRequestRegister	WORD	PWM request register	-
Ch*_FrequestValue	UDINT	Ch* frequency set value	Unit: Hz (0 to 100,000 Hz)
Ch* DutyValue	UINT	Ch* duty ratio set value	Unit: 0.1% (0 to 100.0%)

■ PwmRequestRegister (PWM request register)

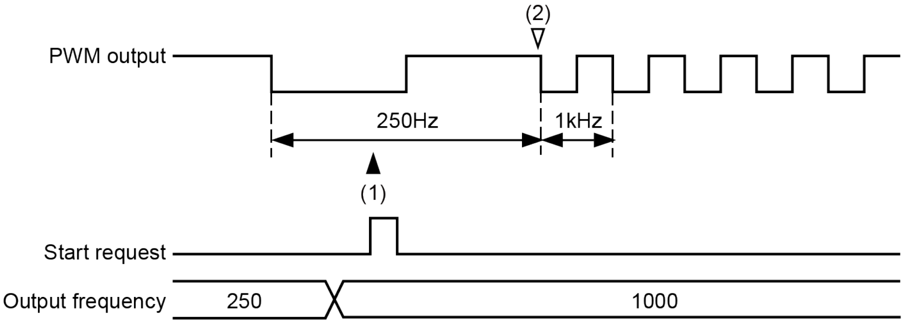
Channel	Type	Description	Remarks
Ch*_PwmStartRequest	BOOL	Ch* start request	PWM output is started at the rising edge.
Ch*_PwmEnableRequest	BOOL	Ch* enable request	FALSE: Disables PWM output TRUE: Enables PWM output

13.5.4 Data Update Timing (Output Frequency)

The data update timing for output frequency during PWM output is descried below.

■ Data update at the rising edge of start request bit

In this mode, the frequency value to be changed is written to the frequency set value and updated with data at the point in time when the start request bit is switched from OFF to ON. The frequency value changed in this timing is reflected at the falling edge of the pulse that is being output.



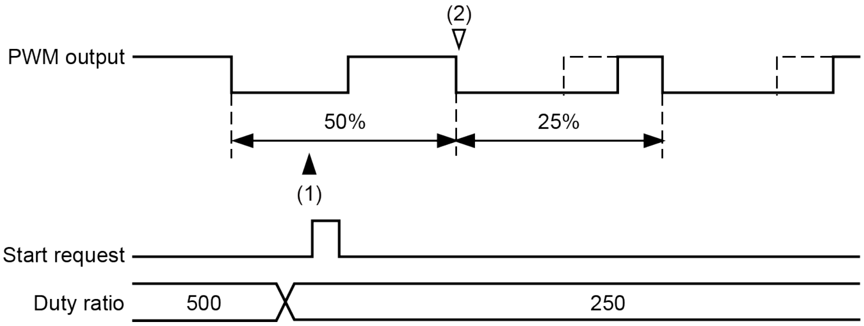
No.	Name	Description
(1)	Data update timing	Indicates the timing in which the frequency set value is reflected
(2)	Output update timing	Indicates the timing in which the changed frequency is reflected as actual output

13.5.5 Data Update Timing (Duty Ratio)

The data update timing for duty ratios during PWM output is descried below.

■ Data update at the rising edge of start request bit

In this mode, the duty ratio to be changed is written to the duty ratio set value and updated with data at the point in time when the start request bit is switched from OFF to ON. The duty ratio changed in this timing is reflected at the falling edge of the pulse that is being output.

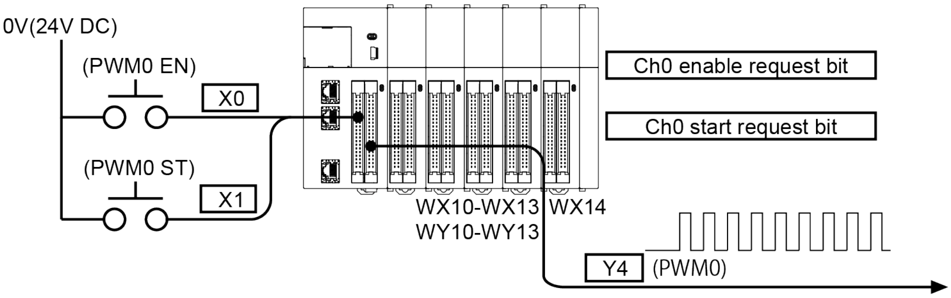


No.	Name	Description
(1)	Data update timing	Indicates the timing in which the duty ratio set value is reflected
(2)	Output update timing	Indicates the timing in which the changed duty ratio is reflected as actual output

13.5.6 PWM Output Setting Example

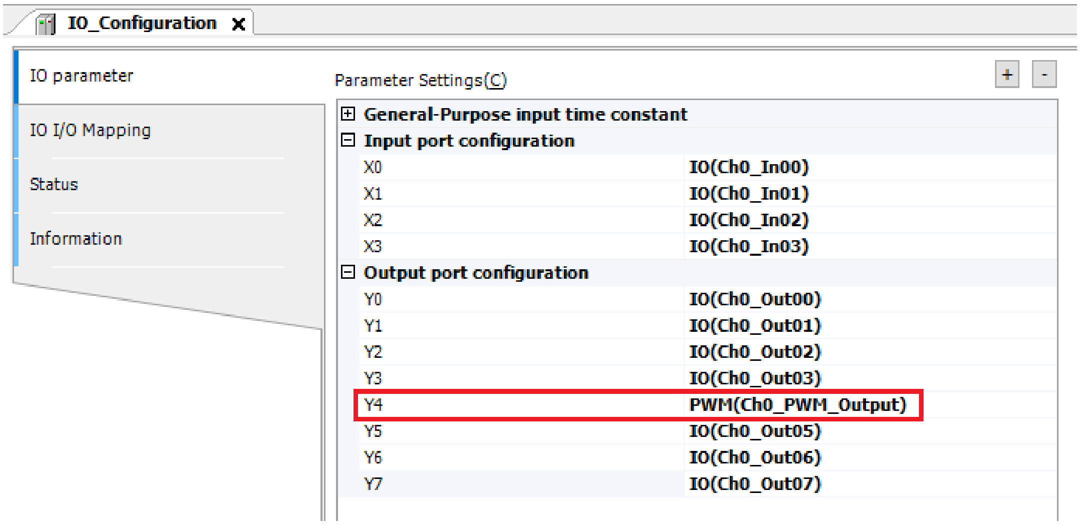
■ Overview

PWM output is performed. It is controlled by the switch input (X0 or X1) connected to the GM1 controller. If the start request bit is turned ON when the enable request bit is ON, PWM output will be started.



■ Settings in GM Programmer

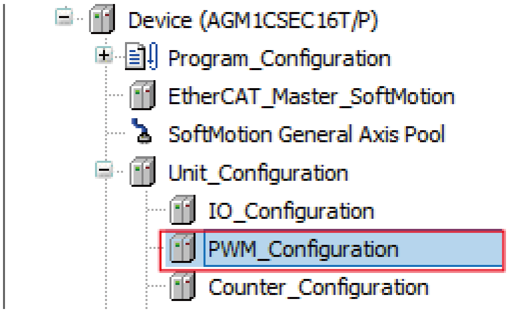
For example, to set PWM output for Y4, use the procedure in "13.5.2 Setting Output Ports with GM Programmer" to configure settings as shown below.



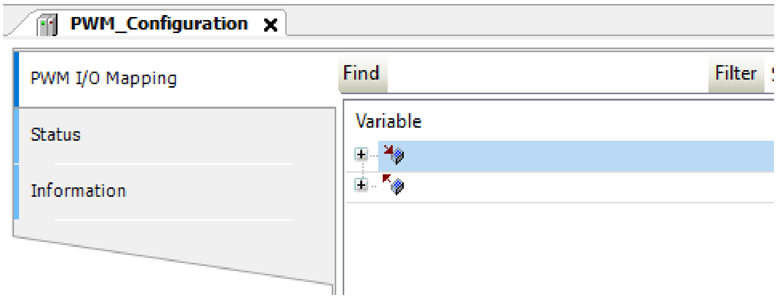
⊞ General-Purpose input time constant	
⊟ Input port configuration	
X0	IO(Ch0_In00)
X1	IO(Ch0_In01)
X2	IO(Ch0_In02)
X3	IO(Ch0_In03)
⊟ Output port configuration	
Y0	IO(Ch0_Out00)
Y1	IO(Ch0_Out01)
Y2	IO(Ch0_Out02)
Y3	IO(Ch0_Out03)
Y4	PWM(Ch0_PWM_Output)
Y5	IO(Ch0_Out05)
Y6	IO(Ch0_Out06)
Y7	IO(Ch0_Out07)

Next, perform I/O mapping for variables created in POU.

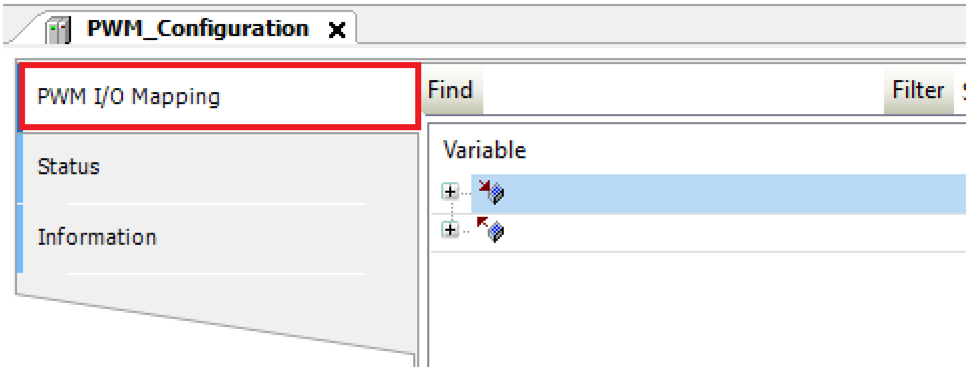
1. From "Device view" in the navigator pane, double-click "PWM_Configuration".



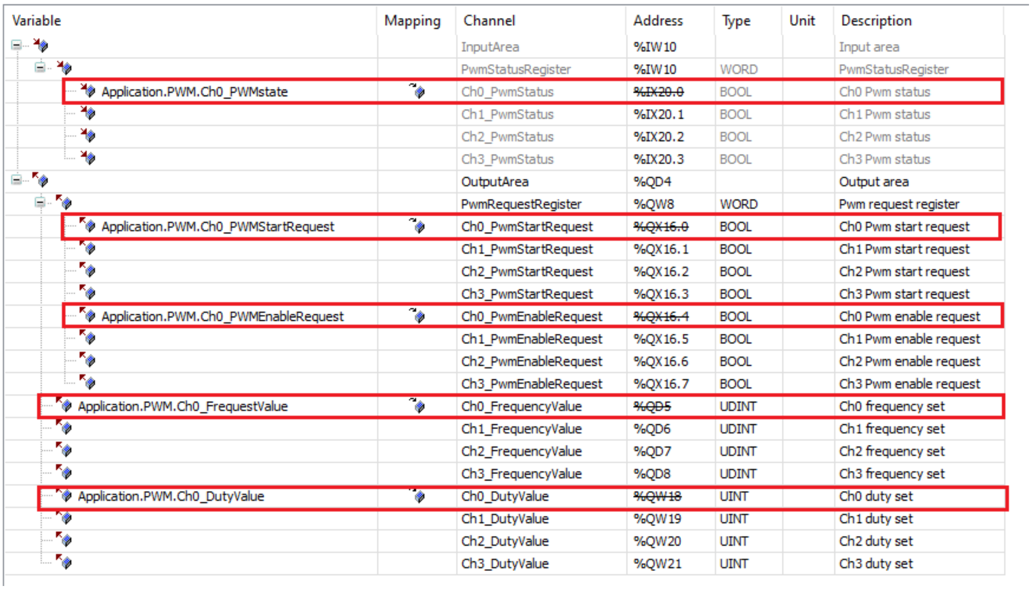
The "PWM_Configuration" window will be displayed.



2. In the "PWM_Configuration" window, click the "PWM I/O Mapping" tab.

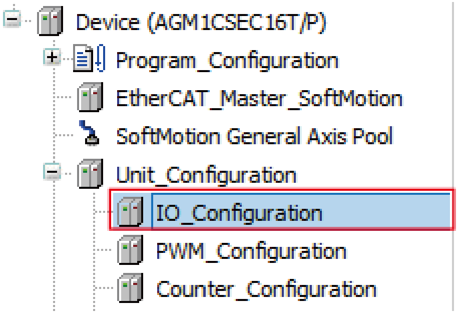


3. In this example, variables with the same name as the channel name are mapped to the channels required for PWM Ch0 (the name of sample POU is created as "PWM").

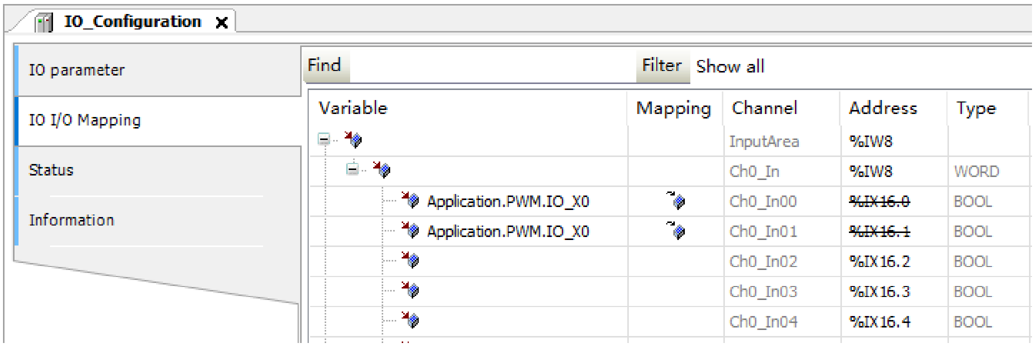


Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%IW10			Input area
		PwmStatusRegister	%IW10	WORD		PwmStatusRegister
Application.PWM.Ch0_PWMstate		Ch0_PwmStatus	~~%IX20.0~~	BOOL		Ch0 Pwm status
		Ch1_PwmStatus	%IX20.1	BOOL		Ch1 Pwm status
		Ch2_PwmStatus	%IX20.2	BOOL		Ch2 Pwm status
		Ch3_PwmStatus	%IX20.3	BOOL		Ch3 Pwm status
		OutputArea	%QD4			Output area
		PwmRequestRegister	%QW8	WORD		Pwm request register
Application.PWM.Ch0_PWMStartRequest		Ch0_PwmStartRequest	~~%QX16.0~~	BOOL		Ch0 Pwm start request
		Ch1_PwmStartRequest	%QX16.1	BOOL		Ch1 Pwm start request
		Ch2_PwmStartRequest	%QX16.2	BOOL		Ch2 Pwm start request
		Ch3_PwmStartRequest	%QX16.3	BOOL		Ch3 Pwm start request
Application.PWM.Ch0_PWMEnableRequest		Ch0_PwmEnableRequest	~~%QX16.4~~	BOOL		Ch0 Pwm enable request
		Ch1_PwmEnableRequest	%QX16.5	BOOL		Ch1 Pwm enable request
		Ch2_PwmEnableRequest	%QX16.6	BOOL		Ch2 Pwm enable request
		Ch3_PwmEnableRequest	%QX16.7	BOOL		Ch3 Pwm enable request
Application.PWM.Ch0_FrequestValue		Ch0_FrequencyValue	~~%QD5~~	UDINT		Ch0 frequency set
		Ch1_FrequencyValue	%QD6	UDINT		Ch1 frequency set
		Ch2_FrequencyValue	%QD7	UDINT		Ch2 frequency set
		Ch3_FrequencyValue	%QD8	UDINT		Ch3 frequency set
Application.PWM.Ch0_DutyValue		Ch0_DutyValue	~~%QW18~~	UINT		Ch0 duty set
		Ch1_DutyValue	%QW19	UINT		Ch1 duty set
		Ch2_DutyValue	%QW20	UINT		Ch2 duty set
		Ch3_DutyValue	%QW21	UINT		Ch3 duty set

4. From Device view in the navigator pane, double-click "IO_Configuration".

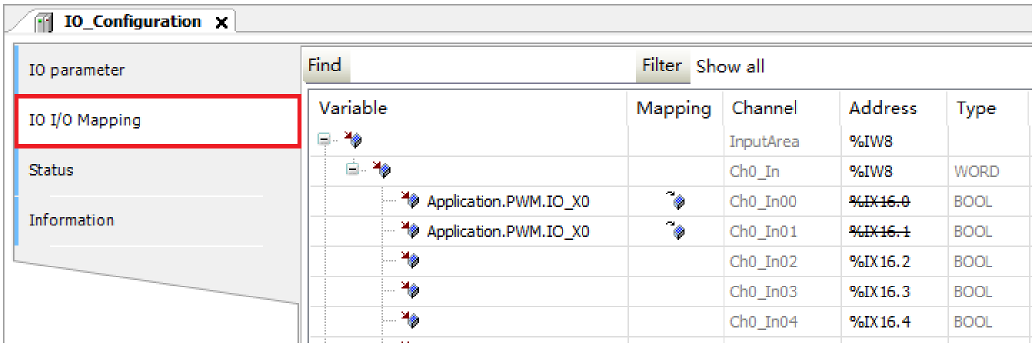


The "IO_Configuration" window will be displayed.



Variable	Mapping	Channel	Address	Type
		InputArea	%IW8	
		Ch0_In	%IW8	WORD
Application.PWM.IO_X0		Ch0_In00	~~%IX16.0~~	BOOL
Application.PWM.IO_X0		Ch0_In01	~~%IX16.1~~	BOOL
		Ch0_In02	%IX16.2	BOOL
		Ch0_In03	%IX16.3	BOOL
		Ch0_In04	%IX16.4	BOOL

5. Click the "IO I/O Mapping" tab.



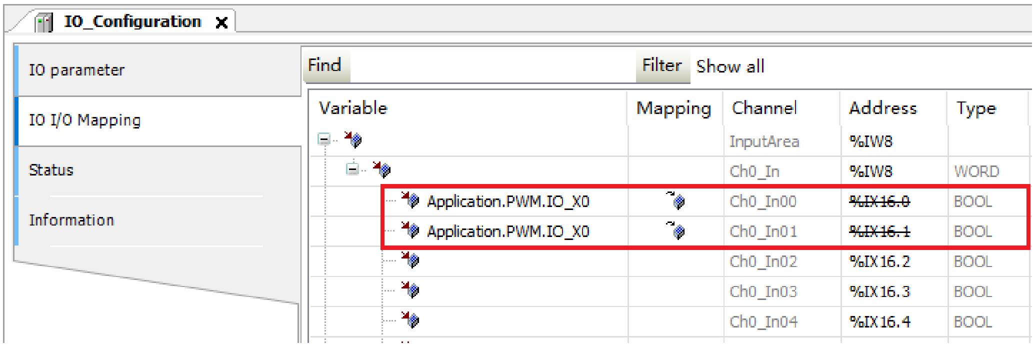
IO_Configuration

- IO parameter
- IO I/O Mapping
- Status
- Information

Find Filter Show all

Variable	Mapping	Channel	Address	Type
		InputArea	%IW8	
		Ch0_In	%IW8	WORD
Application.PWM.IO_X0		Ch0_In00	~~%IX16.0~~	BOOL
Application.PWM.IO_X0		Ch0_In01	~~%IX16.1~~	BOOL
		Ch0_In02	%IX16.2	BOOL
		Ch0_In03	%IX16.3	BOOL
		Ch0_In04	%IX16.4	BOOL

6. Variables will be mapped.



IO_Configuration

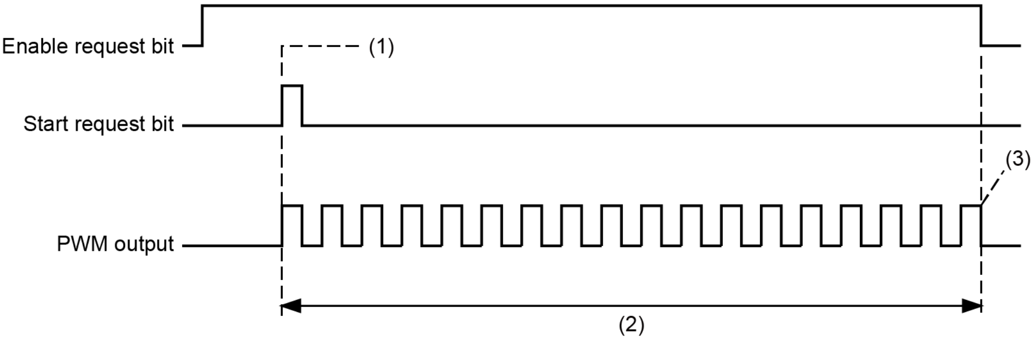
- IO parameter
- IO I/O Mapping
- Status
- Information

Find Filter Show all

Variable	Mapping	Channel	Address	Type
		InputArea	%IW8	
		Ch0_In	%IW8	WORD
Application.PWM.IO_X0		Ch0_In00	~~%IX16.0~~	BOOL
Application.PWM.IO_X0		Ch0_In01	~~%IX16.1~~	BOOL
		Ch0_In02	%IX16.2	BOOL
		Ch0_In03	%IX16.3	BOOL
		Ch0_In04	%IX16.4	BOOL

■ Timing chart

If the rising edge of the start request bit is detected when the enable request bit is ON, Y4 will start PWM output. When the enable request bit is set to OFF, PWM output stops.



No.	Description
(1)	If the rising edge of the start request bit is detected when the enable request bit is ON, PWM output will be started.
(2)	PWM output is performed with a duty ratio of 50% and at a frequency of 100 Hz.
(3)	When the enable request bit turns OFF, PWM output stops.

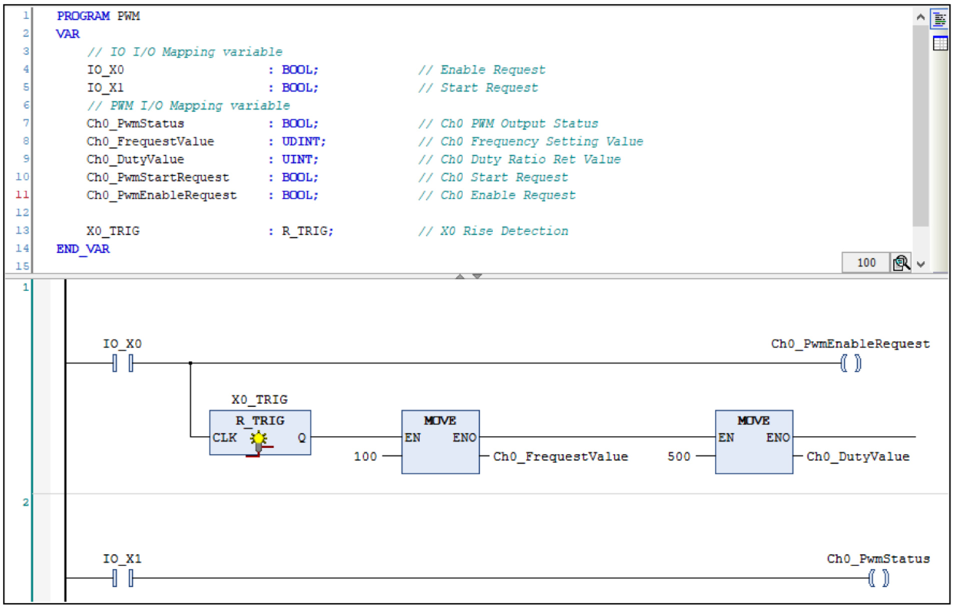
■ Sample program

The following are LD program and ST program examples for sample POU (PWM).

The state of X0 is output to the Ch0 Enable Request bit.

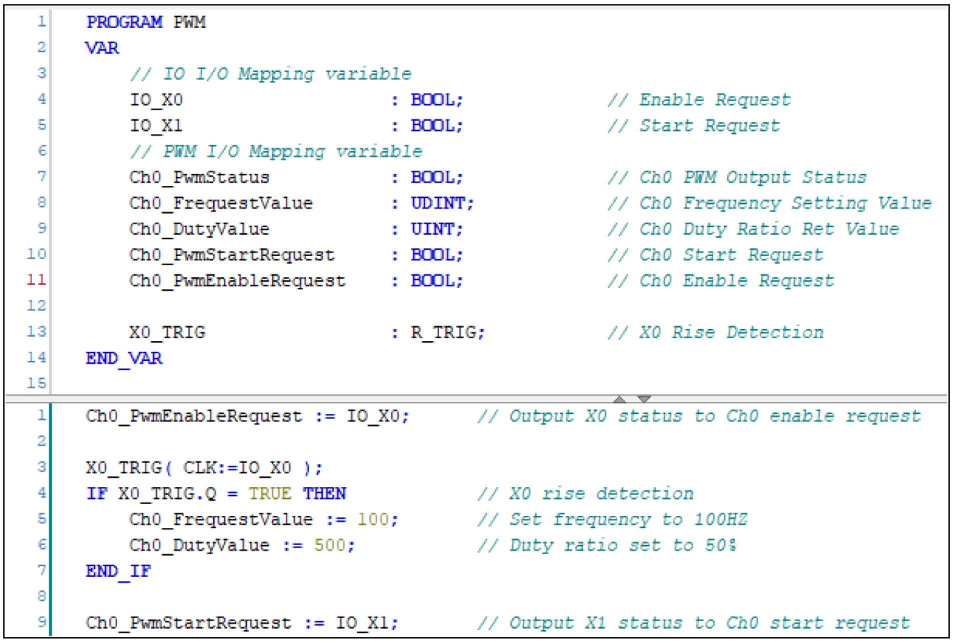
Ch0 frequency and Ch0 duty ratio are set at the rising edge of X0.
The state of X1 is output to the Ch0 Start Request bit.

LD program



```
PROGRAM PWM
VAR
    // IO I/O Mapping variable
    IO_X0                   : BOOL;             // Enable Request
    IO_X1                   : BOOL;             // Start Request
    // PWM I/O Mapping variable
    Ch0_PwmStatus           : BOOL;             // Ch0 PWM Output Status
    Ch0_FrequestValue       : UDINT;            // Ch0 Frequency Setting Value
    Ch0_DutyValue           : UINT;             // Ch0 Duty Ratio Ret Value
    Ch0_PwmStartRequest     : BOOL;             // Ch0 Start Request
    Ch0_PwmEnableRequest    : BOOL;             // Ch0 Enable Request

    X0_TRIG                 : R_TRIG;           // X0 Rise Detection
END_VAR
```

ST program



```
PROGRAM PWM
VAR
    // IO I/O Mapping variable
    IO_X0                   : BOOL;             // Enable Request
    IO_X1                   : BOOL;             // Start Request
    // PWM I/O Mapping variable
    Ch0_PwmStatus           : BOOL;             // Ch0 PWM Output Status
    Ch0_FrequestValue       : UDINT;            // Ch0 Frequency Setting Value
    Ch0_DutyValue           : UINT;             // Ch0 Duty Ratio Ret Value
    Ch0_PwmStartRequest     : BOOL;             // Ch0 Start Request
    Ch0_PwmEnableRequest    : BOOL;             // Ch0 Enable Request

    X0_TRIG                 : R_TRIG;           // X0 Rise Detection
END_VAR
```

```
Ch0_PwmEnableRequest := IO_X0;      // Output X0 status to Ch0 enable request

X0_TRIG( CLK:=IO_X0 );
IF X0_TRIG.Q = TRUE THEN            // X0 rise detection
    Ch0_FrequestValue := 100;       // Set frequency to 100HZ
    Ch0_DutyValue := 500;           // Duty ratio set to 50%
END_IF

Ch0_PwmStartRequest := IO_X1;       // Output X1 status to Ch0 start request
```

13.6 High-speed Counter Function

13.6.1 Overview of High-speed Counter Function

■ Two 4-MHz, signed 32-bit high-speed counters are provided

- High-speed counting of input signals is available for up to the maximum frequency 4 MHz (or 16 MHz for 2-phase input 4 multiple). Two-phase input (phase differential input), individual input, or direction identification input can be selected according to the input device such as encoders or sensors.

■ 24 VDC, 12 VDC, and 5 VDC inputs and line driver input are supported

- The count input circuit supports both open collector output and line driver output (differential output: equivalent to AM26LS31).

■ Ring counter or linear counter can be selected

- Both the ring counter and linear counter are supported. Both types can use the Z-phase of an encoder as count reset timing.

■ Internal clock counting is possible

- Internal clocks can be selected as count input signals. High-accuracy time measurements can be made with a maximum resolution of 0.25 μs. Selectable internal clocks are 0.25 μs (4 MHz), 1 μs (1 MHz), 10 μs (100 kHz), and 100 μs (10 kHz).

■ Capture function and sampling capture function are provided

- It is possible to store the count value at the moment of the occurrence of a trigger assigned to a capture flag. Count values can be checked, independently of I/O refresh.
- The capture function executes capturing at the rising and falling edges of a capture flag. The sampling capture function executes capturing at every sampling time according to the input of a capture flag.

■ Various counter operations can be selected

1. Enable count operation
2. Reset count operation
3. Preset count operation
4. Enable reset count operation
5. Enable preset count operation

■ Band comparison function and target value match comparison function are provided

- The band comparison function allows up to 16 pairs of upper and lower limits to be set for each counter. Up to 16 comparison match flags can be turned ON or OFF for each comparison condition.
- The target value match comparison function allows up to16 target values to be set for each counter. Target values can be set or reset individually according to the direction of counting (incrementation or decrementation) when the count value reaches the target value or when up to 16 comparison match flags for each comparison condition match their respective target values.

- For each counter, up to two comparison match flags and external output signals can be linked with each counter.

■ ON hold time setting (for band comparison function only)

- The ON state of external output signals is retained for the set time (1 to 1,000 ms).

■ Input time constant (noise filter)

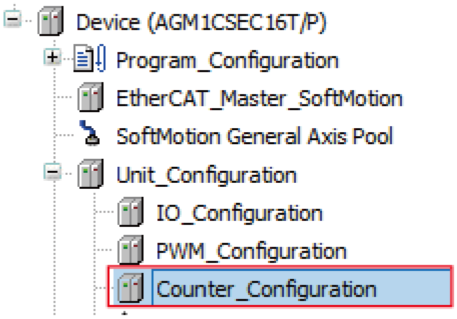
- Input time constants can be set as input signals (A-phase, B-phase, and Z-phase) and control signals for each counter.

13.6.2 Setting Parameters with GM Programmer

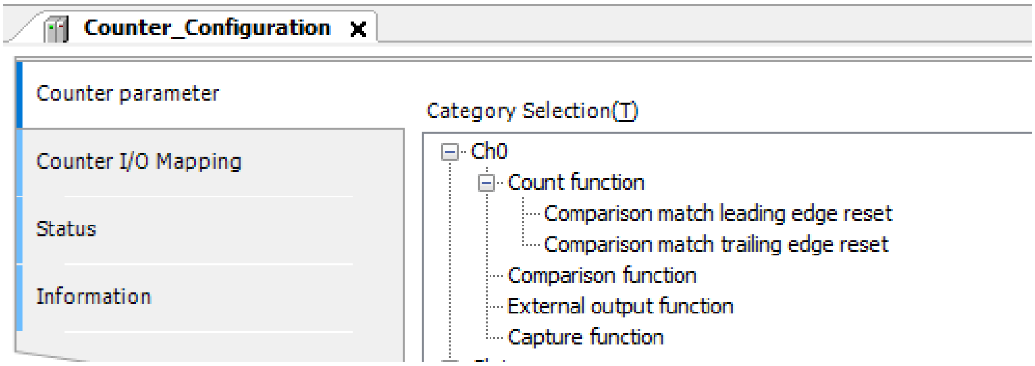
You can set parameters for the high-speed counter via Couter_Configuration in GM Programmer.

1 2 Procedure

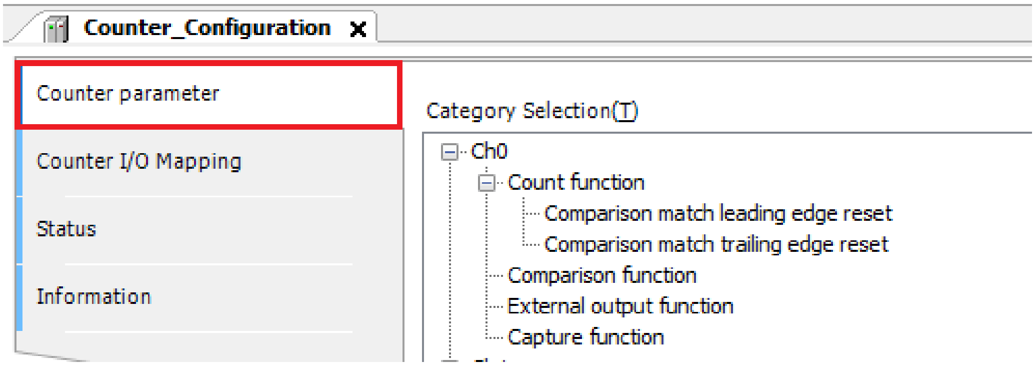
1. From "Device view" in the navigator pane, double-click "Counter_Configuration".



The "Counter_Configuration" setting window will be displayed.



2. In the "Counter_Configuration" window, click the "Counter parameter" tab.



Counter_Configuration

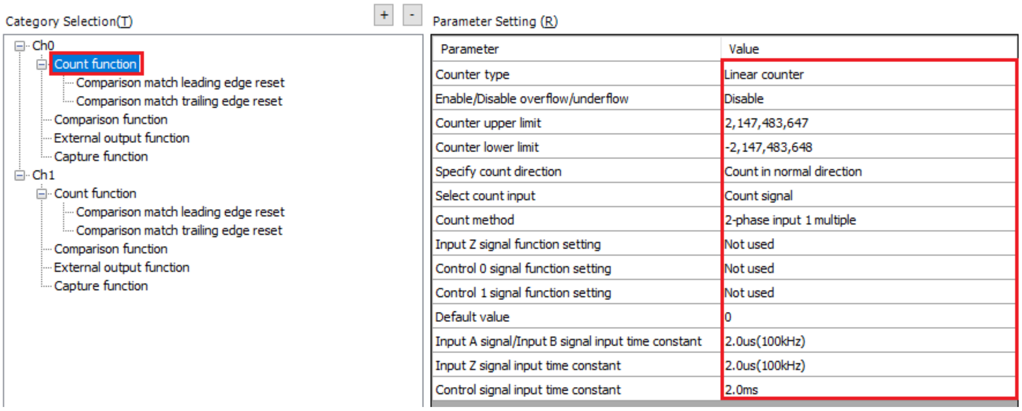
- Counter parameter
- Counter I/O Mapping
- Status
- Information

Category Selection(T)

- Ch0
  - Count function
    - Comparison match leading edge reset
    - Comparison match trailing edge reset
  - Comparison function
  - External output function
  - Capture function

3. Set up parameters of each function for each counter.

Counter function



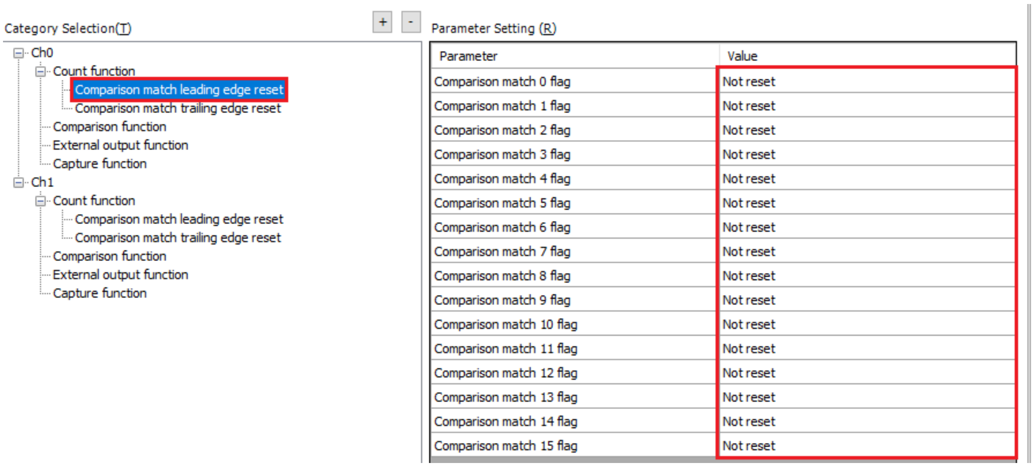
Category Selection(T)

- Ch0
  - Count function
    - Comparison match leading edge reset
    - Comparison match trailing edge reset
  - Comparison function
  - External output function
  - Capture function
- Ch1
  - Count function
    - Comparison match leading edge reset
    - Comparison match trailing edge reset
  - Comparison function
  - External output function
  - Capture function

Parameter Setting (R)

Parameter	Value
Counter type	Linear counter
Enable/Disable overflow/underflow	Disable
Counter upper limit	2,147,483,647
Counter lower limit	-2,147,483,648
Specify count direction	Count in normal direction
Select count input	Count signal
Count method	2-phase input 1 multiple
Input Z signal function setting	Not used
Control 0 signal function setting	Not used
Control 1 signal function setting	Not used
Default value	0
Input A signal/Input B signal input time constant	2.0us(100kHz)
Input Z signal input time constant	2.0us(100kHz)
Control signal input time constant	2.0ms

Comparison match rising edge reset / Comparison match falling edge reset



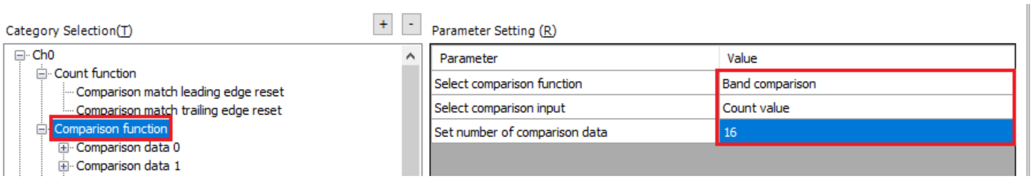
Category Selection(T)

- Ch0
  - Count function
    - Comparison match leading edge reset
    - Comparison match trailing edge reset
  - Comparison function
  - External output function
  - Capture function
- Ch1
  - Count function
    - Comparison match leading edge reset
    - Comparison match trailing edge reset
  - Comparison function
  - External output function
  - Capture function

Parameter Setting (R)

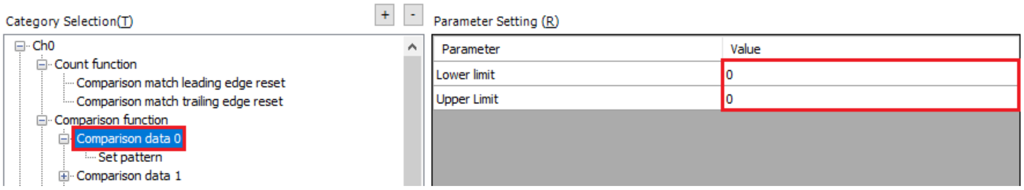
Parameter	Value
Comparison match 0 flag	Not reset
Comparison match 1 flag	Not reset
Comparison match 2 flag	Not reset
Comparison match 3 flag	Not reset
Comparison match 4 flag	Not reset
Comparison match 5 flag	Not reset
Comparison match 6 flag	Not reset
Comparison match 7 flag	Not reset
Comparison match 8 flag	Not reset
Comparison match 9 flag	Not reset
Comparison match 10 flag	Not reset
Comparison match 11 flag	Not reset
Comparison match 12 flag	Not reset
Comparison match 13 flag	Not reset
Comparison match 14 flag	Not reset
Comparison match 15 flag	Not reset

Comparison function (band comparison)



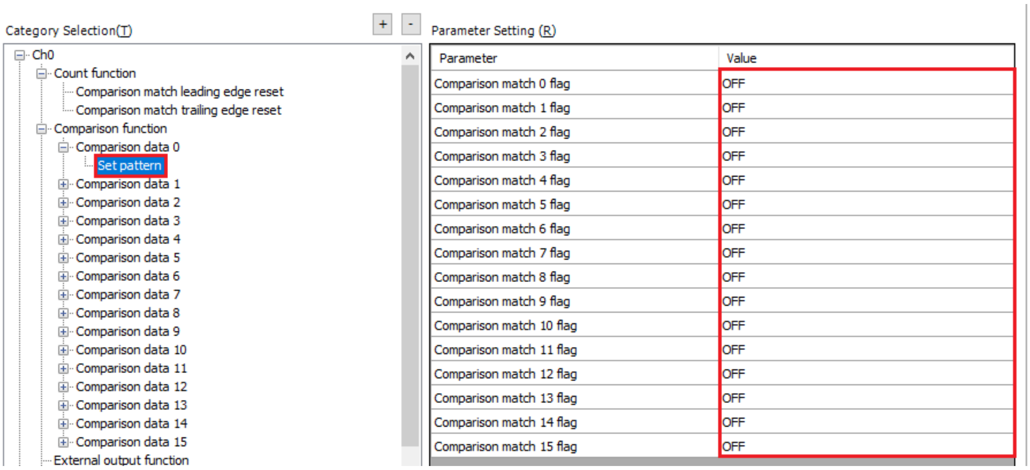
Parameter	Value
Select comparison function	Band comparison
Select comparison input	Count value
Set number of comparison data	16

Comparison function (band comparison) comparison data



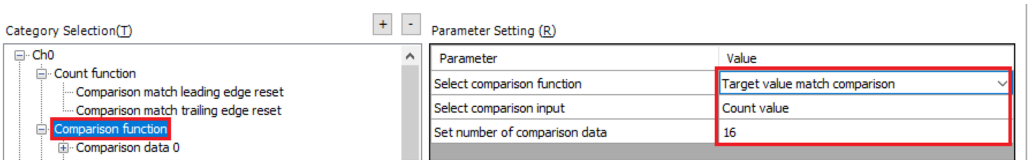
Parameter	Value
Lower limit	0
Upper Limit	0

Set patterns for comparison function (band comparison) comparison data



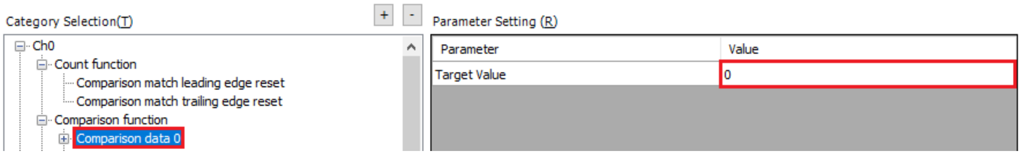
Parameter	Value
Comparison match 0 flag	OFF
Comparison match 1 flag	OFF
Comparison match 2 flag	OFF
Comparison match 3 flag	OFF
Comparison match 4 flag	OFF
Comparison match 5 flag	OFF
Comparison match 6 flag	OFF
Comparison match 7 flag	OFF
Comparison match 8 flag	OFF
Comparison match 9 flag	OFF
Comparison match 10 flag	OFF
Comparison match 11 flag	OFF
Comparison match 12 flag	OFF
Comparison match 13 flag	OFF
Comparison match 14 flag	OFF
Comparison match 15 flag	OFF

Comparison function (target value match comparison)



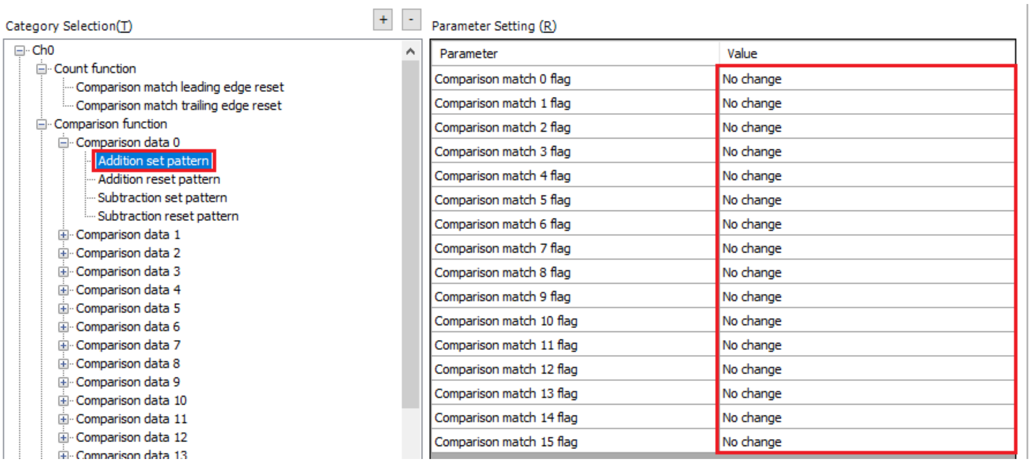
Parameter	Value
Select comparison function	Target value match comparison
Select comparison input	Count value
Set number of comparison data	16

Comparison function (target value match comparison) comparison data



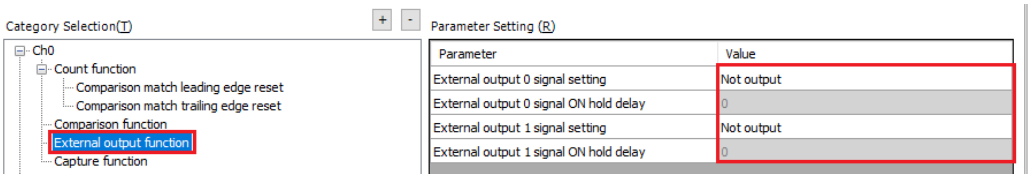
Parameter	Value
Target Value	0

Set / reset patterns for comparison function (target value match comparison) comparison data



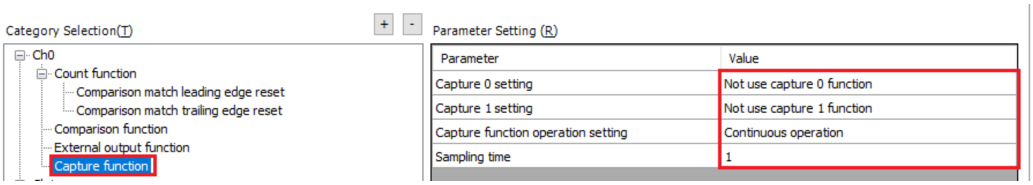
Parameter	Value
Comparison match 0 flag	No change
Comparison match 1 flag	No change
Comparison match 2 flag	No change
Comparison match 3 flag	No change
Comparison match 4 flag	No change
Comparison match 5 flag	No change
Comparison match 6 flag	No change
Comparison match 7 flag	No change
Comparison match 8 flag	No change
Comparison match 9 flag	No change
Comparison match 10 flag	No change
Comparison match 11 flag	No change
Comparison match 12 flag	No change
Comparison match 13 flag	No change
Comparison match 14 flag	No change
Comparison match 15 flag	No change

External output function



Parameter	Value
External output 0 signal setting	Not output
External output 0 signal ON hold delay	0
External output 1 signal setting	Not output
External output 1 signal ON hold delay	0

Capture function



Parameter	Value
Capture 0 setting	Not use capture 0 function
Capture 1 setting	Not use capture 1 function
Capture function operation setting	Continuous operation
Sampling time	1

Info.

- You can copy the parameter set in a counter. To do so, select a channel (Ch0 or Ch1) in the "Category Selection" column and click the [Copy] button.

  Next, select another counter and click the [Paste] button.
- For details on each parameter, refer to "13.6.3 Counter Parameter Setting Items".

13.6.3 Counter Parameter Setting Items

■ Count function (Settable for each counter)

Setting item	Settings	Default value
Counter type	Linear counter / Ring counter	Linear counter
Enable/Disable overflow/ underflow	Disable / Enable	Disable

Setting item	Settings	Default value
Counter upper limit	-2,147,483,647 to 2,147,483,647	2,147,483,647
Counter lower limit	-2,147,483,648 to 2,147,483,646	-2,147,483,648
Specify count direction	Count in normal direction / Count in reverse direction	Count in normal direction
Select count input	Count signal Internal clock 0.25 μs (4 MHz) Internal clock 1.00 μs (1 MHz) Internal clock 10 μs (100 kHz) Internal clock 100 μs (10 kHz)	Count signal
Count method	2-phase input 1 multiple / 2-phase input 2 multiple / 2-phase input 4 multiple / Individual input 1 multiple / Individual input 2 multiple / Direction detection input 1 multiple / Direction detection input 2 multiple	2-phase input 1 multiple
Input Z signal function setting	Not used Reset operation at rising edge Reset operation at falling edge Positive logic reset operation Negative logic reset operation Preset operation at rising edge Preset operation at falling edge Positive logic preset operation Negative logic preset operation	Not used
Control 0 signal function setting	Not used Positive logic enable operation Negative logic enable operation Positive logic enable operation, reset operation at rising edge Negative logic enable operation and reset operation at falling edge Positive logic enable operation and preset operation at rising edge Negative logic enable operation and preset operation at falling edge	Not used
Control 1 signal function setting	Not used Positive logic enable operation Negative logic enable operation	Not used
Default value	Overwrites the count value with the default value when the power is turned ON	0
Input A signal/Input B signal input time constant	No input time constant / 0.1 μs (2 MHz) / 0.2 μs (1 MHz) / 0.5 μs (500 kHz) / 1.0 μs (250 kHz) / 2.0 μs (100 kHz) / 10.0 μs (10 kHz)	2.0 μs (100 kHz)
Input Z signal input time constant	No input time constant / 0.1 μs (2 MHz) / 0.2 μs (1 MHz) / 0.5 μs (500 kHz) / 1.0 μs (250 kHz) / 2.0 μs (100 kHz) / 10.0 μs (10 kHz)	
Control signal input time constant	No input time constant / 2 μs / 5 μs / 10 μs / 20 μs / 50 μs / 100 μs / 500 μs / 1.0 ms / 2.0 ms / 5.0 ms / 10.0 ms	2.0 ms

(Note 1) The control 0 signal and control 1 signal cannot be assigned to the capture function if they are assigned to the enable operation.

Setting item	Settings	Default value
Comparison match leading edge reset / Comparison match trailing edge reset	Selects a comparison match flag that resets the count value at the rising edge or falling edge. Not reset / Reset	Not reset

■ Comparison function (Settable for each counter)

Setting item	Settings	Default value
Select comparison function	Not use / Band comparison / Target value match comparison	Not use
Select comparison input	Count value	Count value
Set number of comparison data	Sets the number of data items to be compared Setting range: 1 to 16	16

If you select "Band comparison" or "Target value match comparison" for "Select comparison function", set parameters for each comparison data item.

Setting item	Settings	Default value
Comparison data 0 to Comparison data 15 (for band comparison)	Specifies the lower and upper limits for each comparison data (Note 1) Setting range: -2,147,483,648 to 2,147,483,647	0
	Specifies the state of the comparison match flag when the current value falls within the specified band ON / OFF	OFF
Comparison data 0 to Comparison data 15 (for target value match comparison)	Specifies target values for each comparison data Setting range: -2,147,483,648 to 2,147,483,647	0
	Comparison match flags to be set or reset can be selected for each status (incrementation or decrementation) of comparison data that has reached the target value. ● Addition set pattern: Set / No change ● Addition reset pattern: Reset / No change ● Subtraction set pattern: Set / No change ● Subtraction reset pattern: Reset / No change	No change

(Note 1) Lower and upper limits can be set within the range between the lower and upper limits for the counter.

For linear counters, set each limit so that the lower limit is less than the upper limit. For ring counters, lower and upper limits can be set in any range.

■ External output function (Settable for each counter)

Setting item	Settings	Default value
External output 0 signal setting	Not output / Output	Not output
External output 0 signal ON hold time	Setting range: 0 to 1,000 (ms)	0 ms
External output 1 signal setting	Not output / Output	Not output

Setting item	Settings	Default value
External output 1 signal ON hold delay	Setting range: 0 to 1,000 (ms)	0 ms

(Note 1) "ON hold time" is enabled only when the band comparison function is used.

■ Capture function (Settable for each counter)

Setting item	Settings	Default value
Capture 0 setting	Not use capture 0 function Capture function at rising edge of control 0 signal Capture function at falling edge of control 0 signal Capture function at rising edge of control 1 signal Capture function at falling edge of control 1 signal Control 0 signal positive logic sampling capture function Control 0 signal negative logic sampling capture function Control 1 signal positive logic sampling capture function Control 1 signal negative logic sampling capture function Output relay (Y relay) sampling capture function	Not use capture 0 function
Capture 1 setting	Not use capture 1 function Capture function at rising edge of control 0 signal Capture function at falling edge of control 0 signal Capture function at rising edge of control 1 signal Capture function at falling edge of control 1 signal	Not use capture 1 function
Capture function operation setting	One operation / Continuous operation	Continuous operation
Sampling time (ms)	1 to 65,535	1

(Note 1) If any value related to the sampling capture function is selected for "Capture 0 setting", "Capture 1 setting" will be disabled.

13.6.4 I/O Mapping for High-speed Counter Output

The high-speed counter function is controlled by user programs.

■ InputArea (input area)

Channel	Type	Description	Remarks
Ch*_StatusRegister	WORD	Ch* status register	-
Ch*_ComparisonMatchRegister	WORD	Ch* comparison match flag	-
Ch*_CountValue	DINT	Ch* count value	-
Ch*_Capture0Value	DINT	Ch* capture 0 value	-
Ch*_Capture1Value	DINT	Ch* capture 1 value	-

Channel	Type	Description	Remarks
Ch*_CaptureDifferenceValue	DINT	Ch* capture differential value	Stores the value (Capture 1 value - Capture 0 value)

■ Ch*_StatusRegister (Ch* status register)

Channel	Type	Description	Remarks
Ch*_OperationReadyStatus	BOOL	Ch* operation ready status	Indicates whether the count function is ready to run. 0: Getting ready, 1: Ready
CH*_CountEnableStatus	BOOL	Ch* count enable status	Indicates whether count operation is in progress. 0: Stopped, 1: Operation in progress
Ch*_CountDirectionStatus	BOOL	Ch* count direction status	Indicates the direction of counting. 0: Reverse rotation (decrementation direction), 1: Forward rotation (incrementation direction)
Ch*_Capture0Status	BOOL	Ch* capture 0 status	Indicates that the count value is stored as capture 0 value at the rising edge or falling edge of the control signal, whichever is enabled.
Ch*_Capture1Status	BOOL	Ch* capture 1 status	Indicates that the count value is stored as capture 1 value at the rising edge or falling edge of the control signal, whichever is enabled.
Ch*_ExternalOutput0Status	BOOL	Ch* external output 0 status	Indicates the output status of external output 0 signal. 0: Output OFF, 1: Output ON
Ch*_ExternalOutput1Status	BOOL	Ch* external output 1 status	Indicates the output status of external output 1 signal. 0: Output OFF, 1: Output ON
Ch*_InputAStatus	BOOL	Ch* input A status	Indicates the input status of input A signal. 0: Output OFF, 1: Output ON
Ch*_InputBStatus	BOOL	Ch* input B status	Indicates the input status of input B signal. 0: Output OFF, 1: Output ON
Ch*_InputZStatus	BOOL	Ch* input Z status	Indicates the input status of input Z signal. 0: Output OFF, 1: Output ON
Ch*_Control0Status	BOOL	Ch* control 0 status	Indicates the input status of control 0 signal. 0: Output OFF, 1: Output ON
Ch*_Control1Status	BOOL	Ch* control 1 status	Indicates the input status of control 1 signal. 0: Output OFF, 1: Output ON

■ Ch*_ComparisonMatchRegister (Ch* comparison match flag)

Channel	Type	Description	Remarks
Ch*_ComparisonMatch0Status to	BOOL	Ch* comparison match 0 flag	Outputs the result of the band comparison function or target value match function. 0: Unmatched, 1: Matched

Channel	Type	Description	Remarks
Ch*_ComparisonMatch15Status		to Ch* comparison match 15 flag	

■ OutputArea (output area)

Channel	Type	Description	Remarks
Ch*_RequestRegister	WORD	Ch* request register	-
Ch*_TemporaryPresetValue	DINT	Ch* temporary preset value	Stores the value to replace the preset value. -2,147,483,648 to 2,147,483,647
Ch*_TemporaryCurrentValue	DINT	Ch* temporary current value	Stores the value to replace the count value. -2,147,483,648 to 2,147,483,647

■ Ch*_RequestRegister (Ch* request register)

Channel	Type	Description	Validity condition	Remarks
Ch*_OperationReadyRequest	BOOL	Ch* operation ready request	Level	Specifies whether to enable operation preparation for the count function. 0: Disable, 1: Enable
Ch*_CountEnableRequest	BOOL	Ch* count enable request	Level	Specifies whether to enable count operation. 0: Disable, 1: Enable
Ch*_ResetRequest	BOOL	Ch* reset request	ON edge	Specifies whether to reset the count value. 0: Do not reset 1: Reset
Ch*_PresetRequest	BOOL	Ch* preset request	ON edge	Specifies whether to change the count value to a preset value. 0: Do not change, 1: Change
Ch*_ResetEnableRequest	BOOL	Ch* reset enable request	Level	Capture function: Specifies whether to enable reset count operation for the input Z signal or comparison match flag. 0: Disable, 1: Enable Sampling capture function Specifies whether to enable reset count operation. 0: Disable, 1: Enable
Ch*_CurrentValueChangeRequest	BOOL	Ch* current value change request	ON edge	Specifies whether to change the count value to a temporarily current value. 0: Do not change, 1: Change
Ch*_PresetValueChangeRequest	BOOL	Ch* preset value change request	ON edge	Specifies whether to change the preset value to a temporarily preset value. 0: Do not change, 1: Change

Channel	Type	Description	Validity condition	Remarks
Ch*_CaptureEnableRequest	BOOL	Ch* capture enable request(Note 1)	Level	Capture function: Specifies whether to enable the capture function. 0: Disable, 1: Enable Sampling capture function Used as a capture flag.
Ch*_ExternalOutput0ForcedONRequest	BOOL	Ch* external output 0 forced ON request	Level	Relay to forcibly turn on the external output 0 signal
Ch*_ExternalOutput0ForcedOFFRequest	BOOL	Ch* external output 0 forced OFF request	Level	Relay to forcibly turn off the external output 0 signal
Ch*_ExternalOutput1ForcedONRequest	BOOL	Ch* external output 1 forced ON request	Level	Relay to forcibly turn on the external output 1 signal
Ch*_ExternalOutput1ForcedOFFRequest	BOOL	Ch* external output 1 forced OFF request	Level	Relay to forcibly turn off the external output 1 signal
Ch*_ErrorClearRequest	BOOL	Ch* error clearing request	ON edge	Specifies whether to clear the error. 0: Do not clear, 1: Clear

(Note 1) The behavior of the "Ch* capture enable request" bit differs according to the function to be used.

13.6.5 Operation Ready Request

■ Operation ready request program

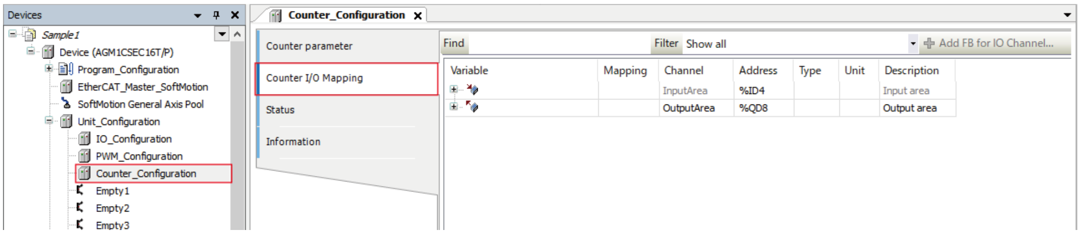
To enable the settings of the high-speed counter function, you must configure parameter settings with GM Programmer and issue an operation ready request.

Ensure that the following operation ready request is issued before the high-speed counter is used.

Example: A program to request preparation for CH0 operation of the high-speed counter function

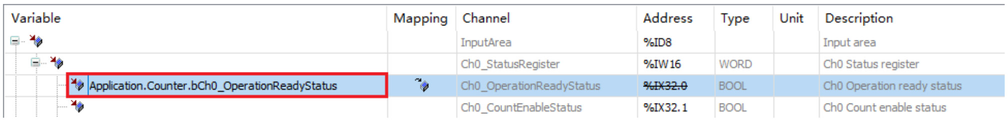
First, perform I/O mapping for variables created in POU.

From "Device view" in the navigator pane, double-click "Counter_Configuration" and select the "Counter I/O mapping" tab.

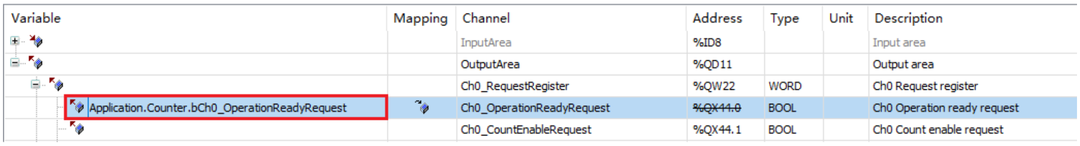


Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID4			Input area
		OutputArea	%QD8			Output area

In this example, variables are mapped to the channel used for high-speed counter Ch0 (the name of sample POU is created as "Counter").



Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		Ch0_StatusRegister	%IW16	WORD		Ch0 Status register
Application.Counter.bCh0_OperationReadyStatus		Ch0_OperationReadyStatus	~~%IX32.0~~	BOOL		Ch0 Operation ready status
		Ch0_CountEnableStatus	%IX32.1	BOOL		Ch0 Count enable status

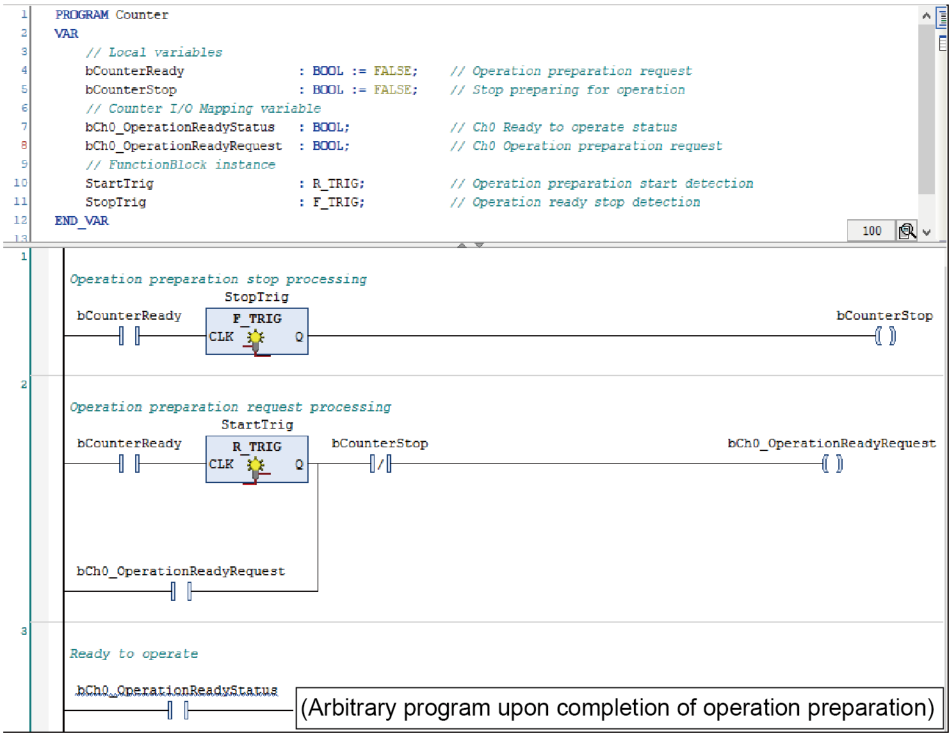


Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		OutputArea	%QD11			Output area
		Ch0_RequestRegister	%QW22	WORD		Ch0 Request register
Application.Counter.bCh0_OperationReadyRequest		Ch0_OperationReadyRequest	~~%QX44.0~~	BOOL		Ch0 Operation ready request
		Ch0_CountEnableRequest	%QX44.1	BOOL		Ch0 Count enable request

The following are LD program and ST program examples for sample POU (Counter).

When the "OperationReadyRequest" flag is set to TRUE, operation preparation is started.

LD program

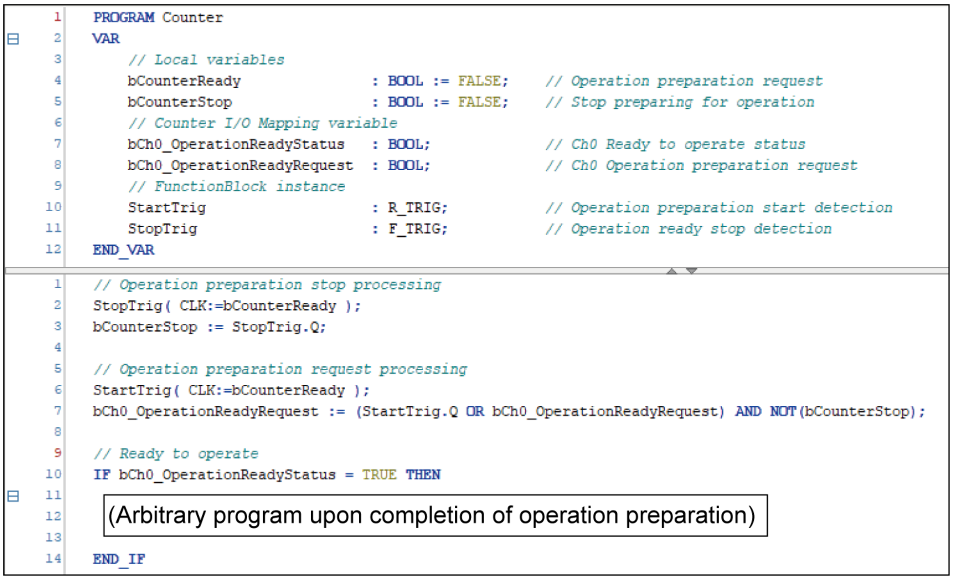


```
PROGRAM Counter
VAR
    // Local variables
    bCounterReady                : BOOL := FALSE;    // Operation preparation request
    bCounterStop                 : BOOL := FALSE;    // Stop preparing for operation
    // Counter I/O Mapping variable
    bCh0_OperationReadyStatus    : BOOL;             // Ch0 Ready to operate status
    bCh0_OperationReadyRequest   : BOOL;             // Ch0 Operation preparation request
    // FunctionBlock instance
    StartTrig                    : R_TRIG;           // Operation preparation start detection
    StopTrig                     : F_TRIG;           // Operation ready stop detection
END_VAR
```

100
1
Operation preparation stop processing
StopTrig
bCounterReady
F_TRIG
CLK
Q
bCounterStop
2
Operation preparation request processing
StartTrig
bCounterReady
R_TRIG
CLK
Q
bCounterStop
bCh0_OperationReadyRequest
bCh0_OperationReadyRequest
3
Ready to operate
bCh0_OperationReadyStatus

(Arbitrary program upon completion of operation preparation)

ST program



```
PROGRAM Counter
VAR
    // Local variables
    bCounterReady                : BOOL := FALSE;    // Operation preparation request
    bCounterStop                 : BOOL := FALSE;    // Stop preparing for operation
    // Counter I/O Mapping variable
    bCh0_OperationReadyStatus    : BOOL;             // Ch0 Ready to operate status
    bCh0_OperationReadyRequest   : BOOL;             // Ch0 Operation preparation request
    // FunctionBlock instance
    StartTrig                    : R_TRIG;           // Operation preparation start detection
    StopTrig                     : F_TRIG;           // Operation ready stop detection
END_VAR
```

```
// Operation preparation stop processing
StopTrig( CLK:=bCounterReady );
bCounterStop := StopTrig.Q;

// Operation preparation request processing
StartTrig( CLK:=bCounterReady );
bCh0_OperationReadyRequest := (StartTrig.Q OR bCh0_OperationReadyRequest) AND NOT(bCounterStop);

// Ready to operate
IF bCh0_OperationReadyStatus = TRUE THEN

  (Arbitrary program upon completion of operation preparation)

END_IF
```

■ Downloading IO parameters

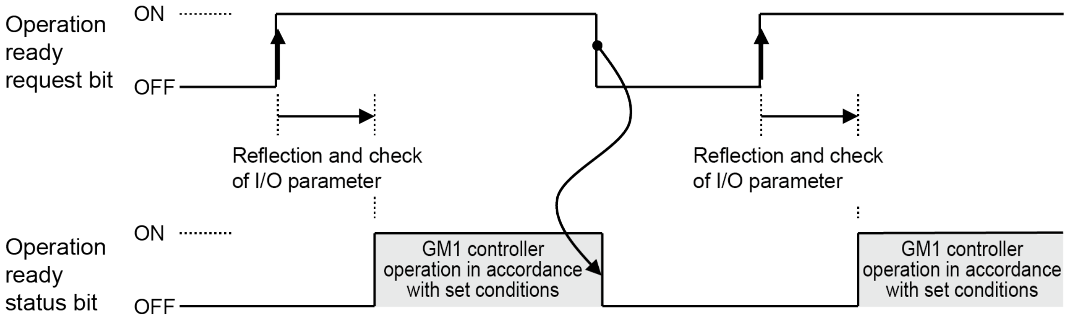
The IO parameters of the high-speed counter function are downloaded to the GM1 controller together with user programs.

When an operation ready request is issued in RUN mode and the operation ready status bit turns ON, each function of the high-speed counter function is enabled.

■ Behaviors when operation ready request program is executed

When the operation ready request program is executed and each behavior set in the GM1 controller becomes executable, the operation ready status bit turns ON.

Behaviors of operation ready request bit


Operation ready request bit
ON
OFF
Reflection and check of I/O parameter
Reflection and check of I/O parameter
Operation ready status bit
ON
OFF
GM1 controller operation in accordance with set conditions
GM1 controller operation in accordance with set conditions

(Note 1) Ensure that the operation ready request bit remains ON when the high-speed counter is used.

Info.

- If the default value or preset value is out of range, the operation ready status bit will not turn ON.

13.6.6 Count Function

■ Setup procedure

1. From "Device view" in the navigator pane, double-click "Counter_Configuration".
2. Click the "Counter Parameters" tab.
3. For each channel, select the count function and set up each parameter.

Info.

- For details on how to set up parameters, refer to "13.6.2 Setting Parameters with GM Programmer".

■ Counter upper limit and Counter lower limit

Set the upper limit and lower limit values for each counter.

Counter upper limit: Any value between -2,147,483,647 and 2,147,483,647 can be set (Default value: 2,147,483,647)

Counter lower limit: Any value between -2,147,483,648 and 2,147,483,646 can be set (Default value: -2,147,483,648)

■ Specify count direction

Set the rotational direction of count input.

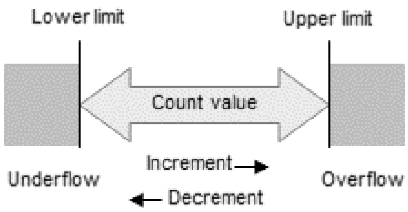
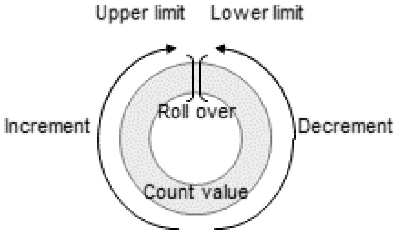
Count in normal direction: Counts in the direction stated in the manual

Count in reverse direction: Counts in the direction reverse to the one stated in the manual

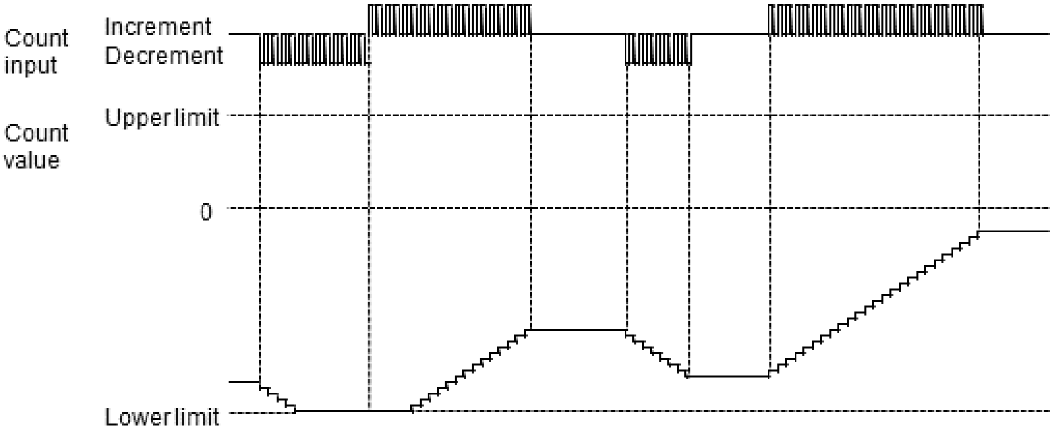
■ Counter type

The counter behavior differs according to the type of the counter as below.

Difference in behavior between counter types

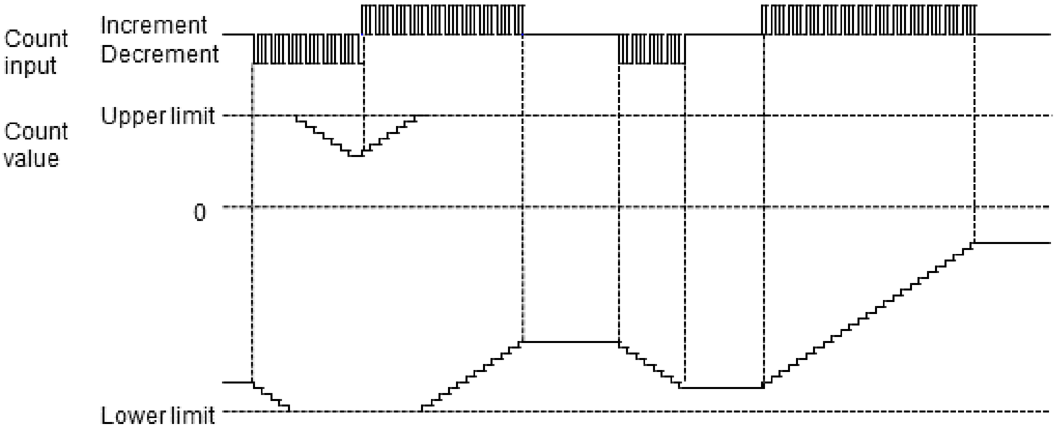
Item	Linear counter	Ring counter
Behavior image	 Lower limit Upper limit Count value Underflow Increment Decrement Overflow	 Upper limit Lower limit Increment Roll over Decrement Count value

Behavior example of linear counter



Behavior example of ring counter

- If the count is decremented from the lower limit, the counter will roll over the count and continue a decremental count from the upper limit.
- If the count is incremented from the upper limit, the counter will roll over the count and continue an incremental count from the lower limit.



■ Enable / Disable overflow / underflow (for linear counters only)

Set the counter behavior to be performed when the count value reaches the specified upper limit or lower limit.

Disable: Continues counting within the countable range for the system (-2,147,483,648 to 2,147,483,647) even if the specified upper limit or lower limit is reached.

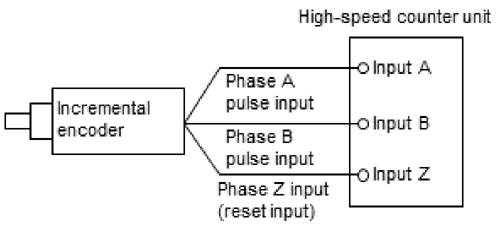
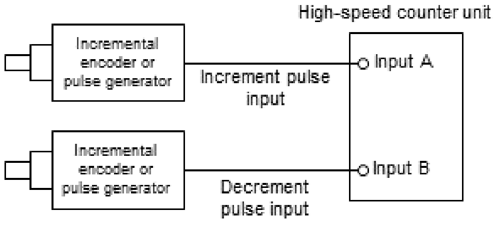
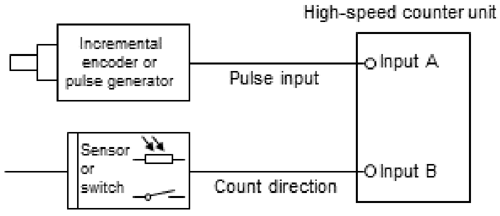
■ Select count input

- To import input signals from external devices, select "Count input".
- To measure the frequencies of external input signals or time based on the internal clock, select "Internal clock".
- For internal clocks, you can select from 0.25 μs (4 MHz), 1.00 μs (1 MHz), 10 μs (100 kHz), and 100 μs (10 kHz).

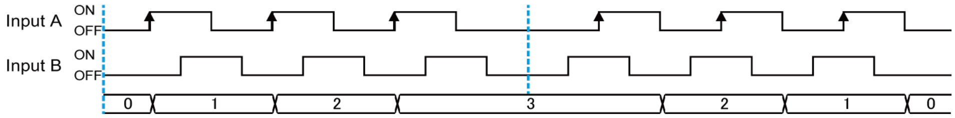
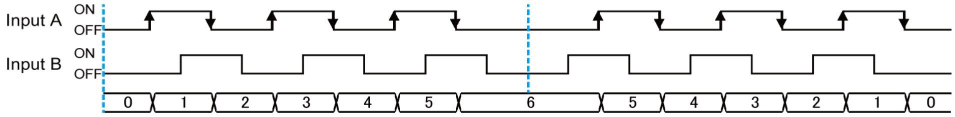
■ Count method

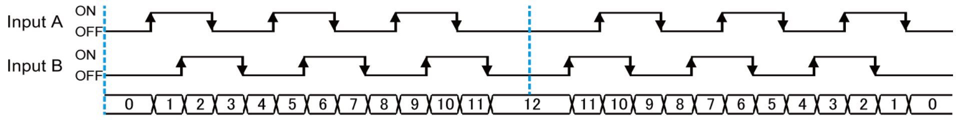
- You can select from the three types shown in the table below according to the input device to be connected.
- The count behavior changes according to the settings of a multiplication factor, as described on the following pages.

Count method

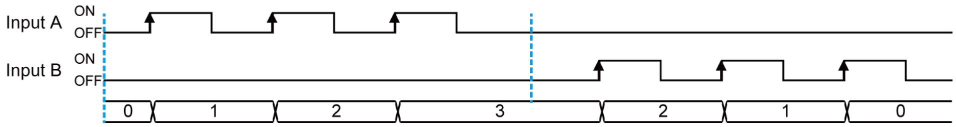
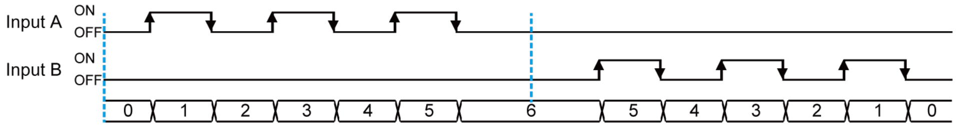
Method	Connected	Count
2-phase (Phase difference)	 High-speed counter unit Incremental encoder Phase A pulse input Phase B pulse input Phase Z input (reset input) Input A Input B Input Z	For 2-phase input, the input A signal and input B signal of each counter are connected to phase A and phase B, respectively, in the encoder. The count direction depends on the phase difference between phases A and B. When phase A is ahead of phase B by 90 degrees in terms of the electrical angle, the count value is incremented. When phase A is behind phase B by 90 degrees in terms of the electrical angle, the count value is decremented.
Individual	 High-speed counter unit Incremental encoder or pulse generator Increment pulse input Incremental encoder or pulse generator Decrement pulse input Input A Input B	For individual input, the counter is incremented when the level of the input A signal rises or falls, and decremented when the level of the input B signal rises or falls.
Direction detection	 High-speed counter unit Incremental encoder or pulse generator Pulse input Sensor or switch Count direction Input A Input B	For direction detection input, the count signal is connected to the input A signal. The count direction is controlled by the direction signal level of the input B signal. When the input B signal is OFF, the counter is incremented when the level of the input A signal rises or falls. When the input B signal is ON, the counter is decremented.

Count operation of 2-phase input (Phase difference input)

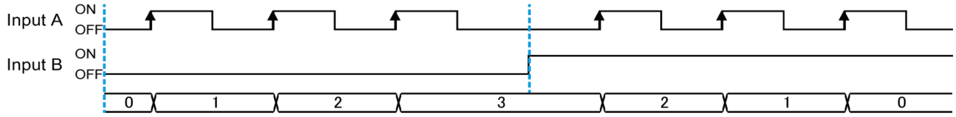
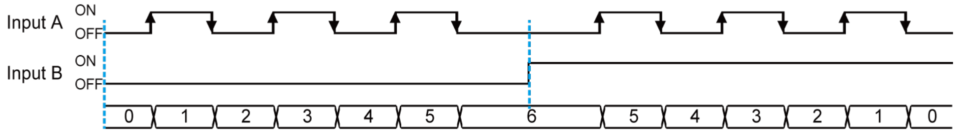
Multiplication	Timing chart		
		Incrementation	Decrementation
Multiply by 1	 Input A ON OFF Input B ON OFF 0 1 2 3 2 1 0		
Multiply by 2	 Input A ON OFF Input B ON OFF 0 1 2 3 4 5 6 5 4 3 2 1 0		

<table>
<tr><th rowspan="2">Multiplication</th><th colspan="3">Timing chart</th></tr>
<tr><th></th><th>Incrementation</th><th>Decrementation</th></tr>
<tr><td>Multiply by 4</td><td colspan="3"> </td></tr>
</table>

Count operation of individual input

<table>
<tr><th rowspan="2">Multiplication</th><th colspan="3">Timing chart</th></tr>
<tr><th></th><th>Incrementation</th><th>Decrementation</th></tr>
<tr><td>Multiply by 1</td><td colspan="3"> </td></tr>
<tr><td>Multiply by 2</td><td colspan="3"> </td></tr>
</table>

Count operation of direction detection input

<table>
<tr><th rowspan="2">Multiplication</th><th colspan="3">Timing chart</th></tr>
<tr><th></th><th>Incrementation</th><th>Decrementation</th></tr>
<tr><td>Multiply by 1</td><td colspan="3"> </td></tr>
<tr><td>Multiply by 2</td><td colspan="3"> </td></tr>
</table>

■ Input time constant

Input time constants can be set for input signals (phases A, B, and Z) and control signals.

Input signal name	Settings
Input A signal, input B signal (The same time constant for both signals)	No input time constant / 0.1 μs (2 MHz) / 0.2 μs (1 MHz) / 0.5 μs (500 kHz) / 1.0 μs (250 kHz) / 2.0 μs (100 kHz) / 10.0 μs (10 kHz)
Input Z signal	No input time constant / 0.1 μs (2 MHz) / 0.2 μs (1 MHz) / 0.5 μs (500 kHz) / 1.0 μs (250 kHz) / 2.0 μs (100 kHz) / 10.0 μs (10 kHz)
Control signal	No input time constant / 2 μs / 5 μs / 10 μs / 20 μs / 50 μs / 100 μs / 500 μs / 1.0 ms / 2.0 ms / 5.0 ms / 10.0 ms

■ Types of count operation

Enable count operation

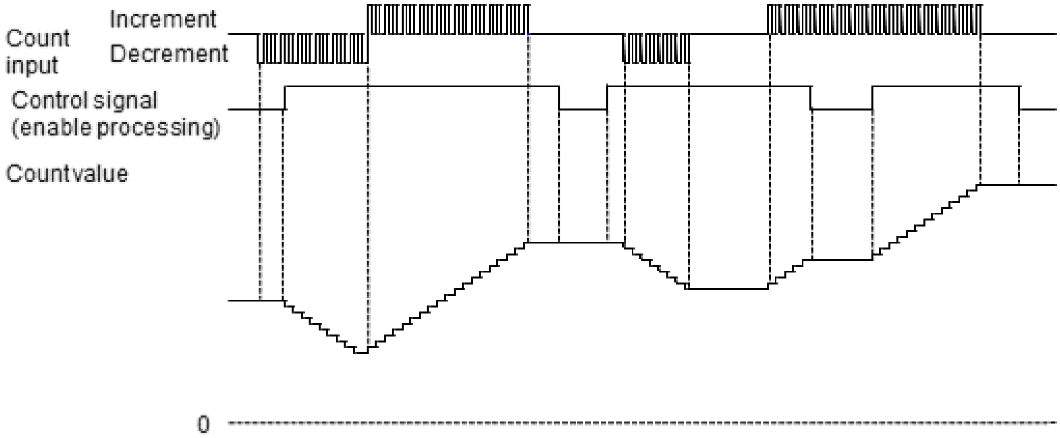
- Enable count operation is used by allocating the enable function to the control flag. Count operation is performed while the control flag is enabled.
- Enable count operation can be set using the methods shown in the following table.

Types of control flag (enable processing)

Signal	Setting method using the Counter_Configuration parameters	Enable condition	
		ON OFF ⊓	ON OFF ⊔
Count enable request bit	- (No need to set)	●	
Control 0 signal	Setting the enable operation conditions by setting up the control 0 signal function	●	●
Control 1 signal	Setting the enable operation conditions by setting up the control 1 signal function	●	●

(Note 1) If you set the control 0 signal or control 1 signal as the enable operation condition, do not use the count enable request bit.

Count enable operation example



■ Reset count operation

- Reset count operation is used by allocating the reset function to the control flag.
- The count value is reset to 0 according to the change (rising, falling, positive logic, or negative logic) of the control flag.
- Reset count operation can be set using the methods shown in the following table.

Types of control flag (reset processing)

Signal	Setting method using the Counter_Configuration parameters	Reset condition			
		ON OFF ↑	ON OFF ↓	ON OFF ⊓	ON OFF ⊔
Reset request bit	- (No need to set)	●			

Signal	Setting method using the Counter_Configuration parameters	Reset condition			
		ON OFF ↑	ON OFF ↓	ON OFF ⊓	ON OFF ⊔
Input Z signal	Setting the reset operation conditions by setting up the input Z signal function	●	●	●	●
Comparison match status bit	Setting the "Comparison match rising edge reset" or "Comparison match falling edge reset" function to "Reset"	●	●		

(Note 1) If you set the control 0 signal or control 1 signal as the enable operation condition, the count enable request bit will be disabled.

Info.

- To use the input Z signal and comparison match flag as reset signals, turn ON the reset enable request bit through user programs.

■ Preset count operation

- Preset count operation is used by allocating the preset function to the control flag.
- Preset count operation rewrites the count value as the preset value according to the change (rising, falling, positive logic, or negative logic) of the control flag.
- Preset count operation can also be used to start counter operation from the preset value.
- Preset count operation can be set using the methods shown in the following table.

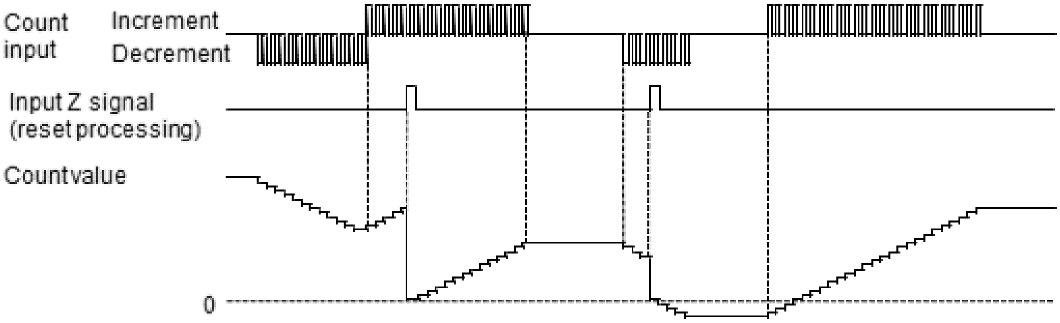
Types of control flag (preset processing)

Signal	Setting method using the Counter_Configuration parameters	Preset condition			
		ON OFF ↑	ON OFF ↓	ON OFF ⊓	ON OFF ⊔
Preset request bit	- (No need to set)	●			
Input Z signal	Setting the preset operation conditions by setting up the input Z signal function	●	●	●	●

Info.

- To set a preset value, you must use a user program to set a temporary preset value and turn ON the preset value change request bit.

Reset (preset) count operation example



■ Enable reset count operation

- ● Enable reset count operation is used by allocating the enable function and reset function to the control flag.
- ● The count value is reset to zero when the counter becomes enabled due to the change of the control flag (enable reset processing).

Types of control flag (enable reset processing)

Signal	Setting method using the Counter_Configuration parameters	Reset condition		Enable condition	
		ON OFF ↑	ON OFF ↓	ON OFF ⊓	ON OFF ⊔
Control 0 signal	Setting the following conditions by setting up the control 0 signal function Positive logic enable operation and reset operation at rising edge Negative logic enable operation and reset operation at falling edge	•	•	•	•

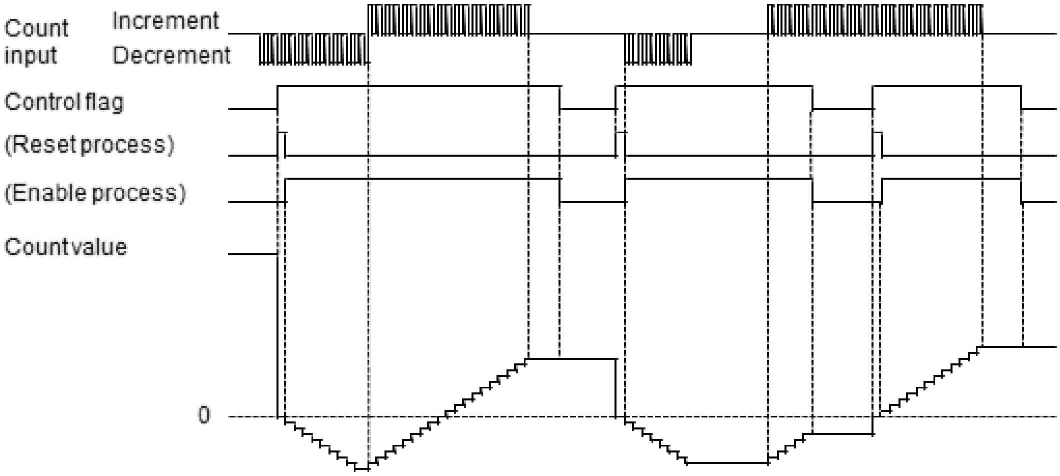
■ Enable preset count operation

- ● Enable preset count operation is used by allocating the enable function and preset function to the control flag.
- ● The count value is set as a preset value when the counter becomes enabled due to the change of the control flag (enable preset processing).

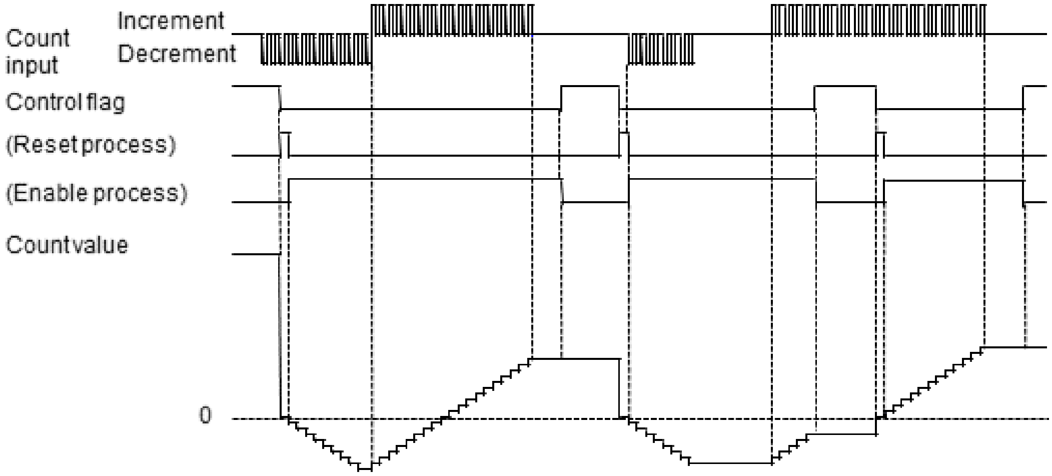
Types of control flag (enable preset processing)

Signal	Setting method using the Counter_Configuration parameters	Preset condition		Enable condition	
		ON OFF ↑	ON OFF ↓	ON OFF ⊓	ON OFF ⊔
Control 0 signal	Setting the following conditions by setting up the control 0 signal function Positive logic enable operation and preset operation at rising edge Negative logic enable operation and preset operation at falling edge	•	•	•	•

Positive logic enable operation and reset (preset) operation at rising edge

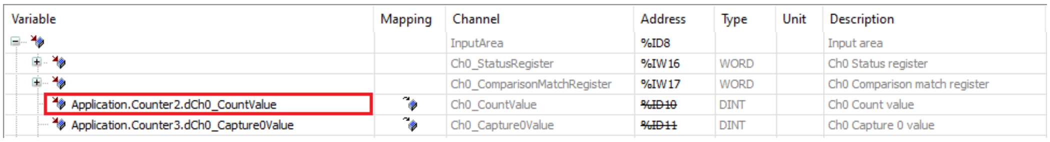


Negative logic enable operation and reset (preset) operation at falling edge

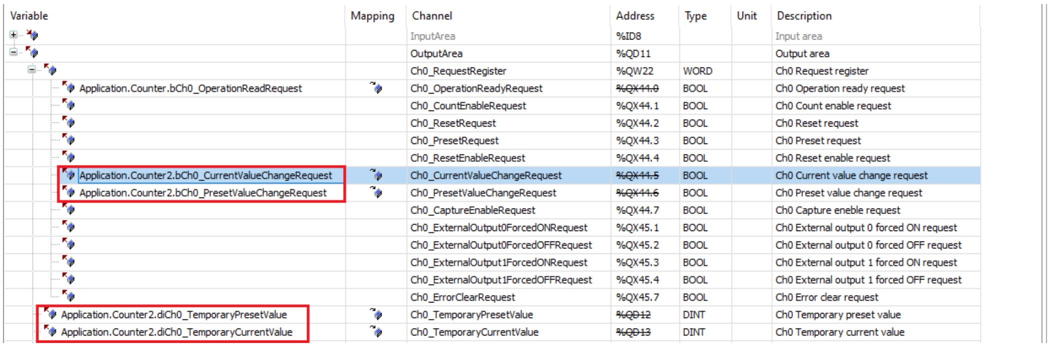


■ Reading the count value or changing the current count value or preset value

- To read and write channel data, variables are mapped to channels in the same way as in "13.6.5 Operation Ready Request". In this example, variables are mapped to channels used for reading the count value of Counter Ch0 and changing the current count value and preset value.



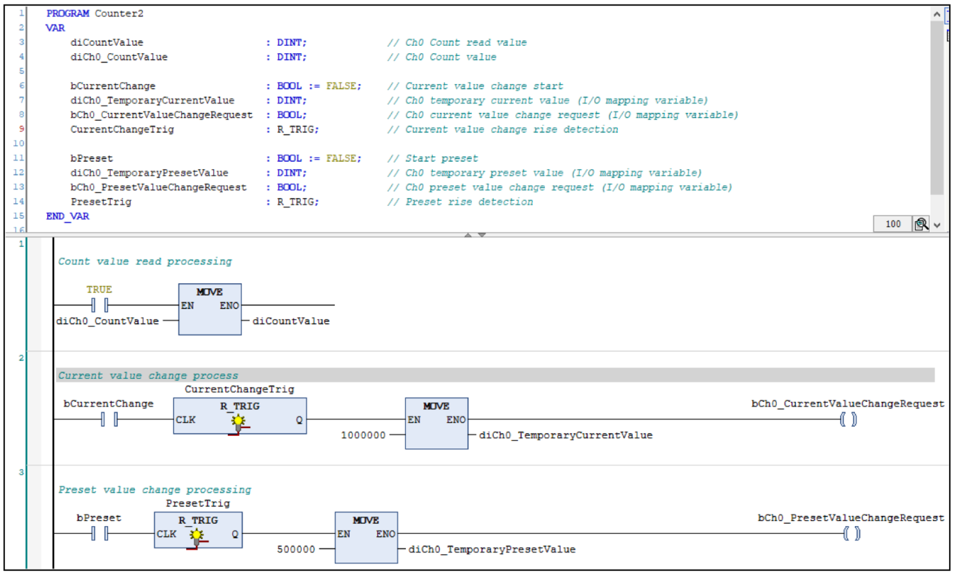
Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		Ch0_StatusRegister	%IW16	WORD		Ch0 Status register
		Ch0_ComparisonMatchRegister	%IW17	WORD		Ch0 Comparison match register
Application.Counter2.dCh0_CountValue		Ch0_CountValue	~~%ID10~~	DINT		Ch0 Count value
Application.Counter3.dCh0_Capture0Value		Ch0_Capture0Value	~~%ID11~~	DINT		Ch0 Capture 0 value



Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		OutputArea	%QD11			Output area
		Ch0_RequestRegister	%QW22	WORD		Ch0 Request register
Application.Counter.bCh0_OperationReadRequest		Ch0_OperationReadyRequest	~~%QX44.0~~	BOOL		Ch0 Operation ready request
		Ch0_CountEnableRequest	%QX44.1	BOOL		Ch0 Count enable request
		Ch0_ResetRequest	%QX44.2	BOOL		Ch0 Reset request
		Ch0_PresetRequest	%QX44.3	BOOL		Ch0 Preset request
		Ch0_ResetEnableRequest	%QX44.4	BOOL		Ch0 Reset enable request
Application.Counter2.bCh0_CurrentValueChangeRequest		Ch0_CurrentValueChangeRequest	~~%QX44.5~~	BOOL		Ch0 Current value change request
Application.Counter2.bCh0_PresetValueChangeRequest		Ch0_PresetValueChangeRequest	~~%QX44.6~~	BOOL		Ch0 Preset value change request
		Ch0_CaptureEnableRequest	%QX44.7	BOOL		Ch0 Capture eneble request
		Ch0_ExternalOutput0ForcedONRequest	%QX45.1	BOOL		Ch0 External output 0 forced ON request
		Ch0_ExternalOutput0ForcedOFFRequest	%QX45.2	BOOL		Ch0 External output 0 forced OFF request
		Ch0_ExternalOutput1ForcedONRequest	%QX45.3	BOOL		Ch0 External output 1 forced ON request
		Ch0_ExternalOutput1ForcedOFFRequest	%QX45.4	BOOL		Ch0 External output 1 forced OFF request
		Ch0_ErrorClearRequest	%QX45.7	BOOL		Ch0 Error clear request
Application.Counter2.diCh0_TemporaryPresetValue		Ch0_TemporaryPresetValue	~~%QD12~~	DINT		Ch0 Temporary preset value
Application.Counter2.diCh0_TemporaryCurrentValue		Ch0_TemporaryCurrentValue	~~%QD13~~	DINT		Ch0 Temporary current value

- The following are LD program and ST program examples for sample POU (Counter2).
  - In this example, count values are read for each scan.
  - When the current value change start bit is set to TRUE, the current value of Ch0 is set to 1000000.
  - When the start preset bit is set to TRUE, the preset value of Ch0 is set to 500000.

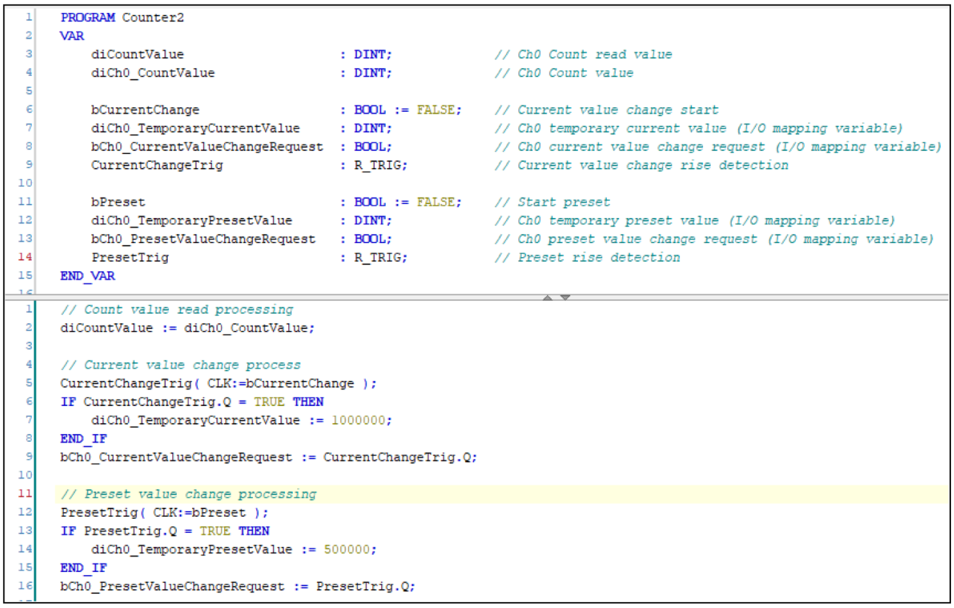
LD program



```
PROGRAM Counter2
VAR
    diCountValue                    : DINT;             // Ch0 Count read value
    diCh0_CountValue                : DINT;             // Ch0 Count value

    bCurrentChange                  : BOOL := FALSE;    // Current value change start
    diCh0_TemporaryCurrentValue     : DINT;             // Ch0 temporary current value (I/O mapping variable)
    bCh0_CurrentValueChangeRequest  : BOOL;             // Ch0 current value change request (I/O mapping variable)
    CurrentChangeTrig               : R_TRIG;           // Current value change rise detection

    bPreset                         : BOOL := FALSE;    // Start preset
    diCh0_TemporaryPresetValue      : DINT;             // Ch0 temporary preset value (I/O mapping variable)
    bCh0_PresetValueChangeRequest   : BOOL;             // Ch0 preset value change request (I/O mapping variable)
    PresetTrig                      : R_TRIG;           // Preset rise detection
END_VAR
```

100
1
Count value read processing
TRUE
MOVE
EN
ENO
diCh0_CountValue
diCountValue
2
Current value change process
CurrentChangeTrig
bCurrentChange
R_TRIG
CLK
Q
MOVE
EN
ENO
1000000
diCh0_TemporaryCurrentValue
bCh0_CurrentValueChangeRequest
3
Preset value change processing
PresetTrig
bPreset
R_TRIG
CLK
Q
MOVE
EN
ENO
500000
diCh0_TemporaryPresetValue
bCh0_PresetValueChangeRequest

ST program



```
PROGRAM Counter2
VAR
    diCountValue                  : DINT;            // Ch0 Count read value
    diCh0_CountValue              : DINT;            // Ch0 Count value

    bCurrentChange                : BOOL := FALSE;   // Current value change start
    diCh0_TemporaryCurrentValue   : DINT;            // Ch0 temporary current value (I/O mapping variable)
    bCh0_CurrentValueChangeRequest : BOOL;           // Ch0 current value change request (I/O mapping variable)
    CurrentChangeTrig             : R_TRIG;          // Current value change rise detection

    bPreset                       : BOOL := FALSE;   // Start preset
    diCh0_TemporaryPresetValue    : DINT;            // Ch0 temporary preset value (I/O mapping variable)
    bCh0_PresetValueChangeRequest : BOOL;            // Ch0 preset value change request (I/O mapping variable)
    PresetTrig                    : R_TRIG;          // Preset rise detection
END_VAR
```

```
// Count value read processing
diCountValue := diCh0_CountValue;

// Current value change process
CurrentChangeTrig( CLK:=bCurrentChange );
IF CurrentChangeTrig.Q = TRUE THEN
    diCh0_TemporaryCurrentValue := 1000000;
END_IF
bCh0_CurrentValueChangeRequest := CurrentChangeTrig.Q;

// Preset value change processing
PresetTrig( CLK:=bPreset );
IF PresetTrig.Q = TRUE THEN
    diCh0_TemporaryPresetValue := 500000;
END_IF
bCh0_PresetValueChangeRequest := PresetTrig.Q;
```

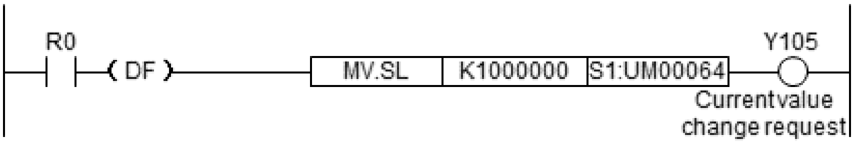
■ Count value when the power is turned on

- When the power is turned on, the count value is "0".

■ Changing the current count value

- The current count value can be changed to any value as necessary.
- Set a value in the temporary current value channel and turn ON the current value change request bit.

Example: A program to change the current value of CH0 to 1000000



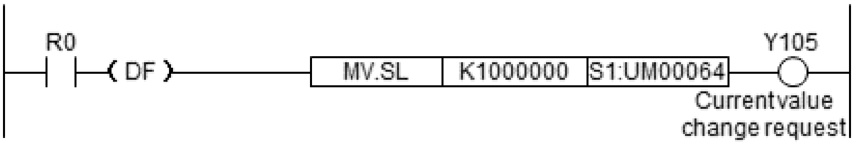
Info.

- Count values cannot be written directly to the count value channel (Ch*_CountValue).

■ Changing the preset value

- The preset value can be changed to any value as necessary.
- Set a value in the temporary preset value channel and turn ON the preset value change request bit.

Example: A program to change the preset value of CH0 to 1000000



Info.

- Preset values that are set cannot be read by programs.

13.6.7 Comparison Function

■ Setup procedure

1. From "Device view" in the navigator pane, double-click "Counter_Configuration".
2. Click the "Counter Parameters" tab.
3. For each counter, select the comparison function and set up each parameter.

Info.

- For details on how to set up parameters, refer to "13.6.2 Setting Parameters with GM Programmer".

■ Types of comparison function

This function compares the current value of the high-speed counter with a preset target value and, when these values match, it reflects the value in the comparison match flag.

- There are target value match comparison and band comparison.
- A total of 16 comparison data items can be set.
- Comparison results can be output externally.
- Comparison methods can be selected for each counter.

Comparison match function specifications

Item	Specifications
Set number of comparison data	Up to 16 data items for each counter (Comparison data 0 to comparison data 15)
Comparison match flag	Up to 16 flags for each counter (Comparison match 0 flag to comparison match 15 flag) Behaviors of 16 comparison match flags can be set for a single comparison data item.
Select comparison function	Target value match comparison Sets or resets the comparison match flag when the elapsed value matches the target value.
	Band comparison Turns ON or OFF the comparison match flag when the elapsed value falls within the range between the lower and upper limits that are set.
External output function	Up to two flags for each counter Comparison match 0 flag or comparison match 1 flag can be allocated to external output.
	ON hold time can be set only when the band comparison function is used. ON hold time: 0 to 1,000 ms

Info.

- Only the comparison match 0 flag or comparison match 1 flag can be set as the external output function.
- By default, the external output 0 and external output 1 signals are set to "Not output". When necessary, change the setting in the Counter_Configuration parameter window.
- There is no need to arrange comparison data items in ascending or descending order.

■ Target value match comparison and band comparison

The main differences are as below.

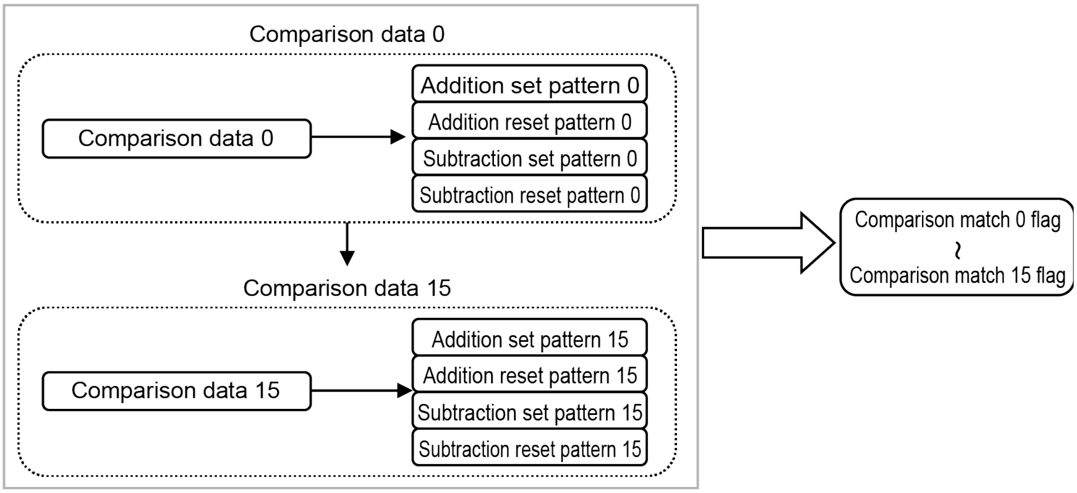
Main differences in characteristics

Item	Target value match comparison	Band comparison
Setting of comparison value data	Comparison value data is specified as a target value.	Comparison value data is specified as a band (lower and upper limits).
Setting for comparison value data match	One of the following four options is specified as the comparison match flag behavior to be performed when the target value is reached. "Addition set pattern": Sets the flag when the current value and comparison value match at the time of incrementation "Addition reset pattern": Resets the flag when the current value and comparison value match at the time of incrementation "Subtraction set pattern": Sets the flag when the current value and comparison value match at the time of decrementation "Subtraction reset pattern": Resets the flag when the current value and comparison value match at the time of decrementation	ON or OFF is specified as the behavior of the comparison match flag when the current value falls within the specified band.
Behavior when comparison value data matches	The behavior that is performed when the current value matches the same comparison value data may differ between incremental count and decremental count, depending on the settings.	The behavior that is performed when the current value matches the same comparison value data is the same for incremental count and decremental count.
External output signal ON hold time	Cannot be set	ON hold time: 0 to 1,000 ms

■ Parameter settings for target value match comparison

Parameter setting procedure

1. Click **Counter parameter**>**Counter (Ch0 or Ch1)**>**Comparison function**, change the "Select comparison function" parameter to "Target value match comparison", and execute "Set number of comparison data".
2. Click **Counter parameter**>**Counter (Ch0 or Ch1)**>**Comparison function**>**Comparison data** and specify target values for each comparison data item.
3. Select "Addition set pattern", "Addition reset pattern", "Subtraction set pattern", and "Subtraction reset pattern" separately and set ""No change", "Set output", or "Reset output"" for each comparison match flag.
4. Configure these settings for each comparison data item.



Info.

- Settings can be configured individually according to the count direction (incremental or decremental direction) at the time of comparison data match.
- A total of 16 comparison match flags can be set separately for "Addition set pattern", "Addition reset pattern", "Subtraction set pattern", and "Subtraction reset pattern".
- For details on how to set up parameters, refer to "13.6.2 Setting Parameters with GM Programmer".

■ Setting example for target value match function

Output setting example

Comparison data	Target value	Output setting	Comparison match 0 flag	Comparison match 1 flag	Comparison match 2 flag	Comparison match 3 flag
0	+500	Addition set pattern		○		
		Addition reset pattern	○			
		Subtraction set pattern				
		Subtraction reset pattern		○	○	○
1	+1,250	Addition set pattern			○	
		Addition reset pattern		○		○
		Subtraction set pattern				
		Subtraction reset pattern				
2	+2,500	Addition set pattern				○
		Addition reset pattern				
		Subtraction set pattern		○		○
		Subtraction reset pattern				
3	+3,750	Addition set pattern		○		
		Addition reset pattern			○	

Comparison data	Target value	Output setting	Comparison match 0 flag	Comparison match 1 flag	Comparison match 2 flag	Comparison match 3 flag
		Subtraction set pattern				
		Subtraction reset pattern				○
4	+5,000	Addition set pattern				
		Addition reset pattern				○
		Subtraction set pattern			○	○
		Subtraction reset pattern	○	○		
5	+6,250	Addition set pattern			○	
		Addition reset pattern				
		Subtraction set pattern				
		Subtraction reset pattern				○
6	+7,500	Addition set pattern	○			○
		Addition reset pattern		○		
		Subtraction set pattern	○	○		
		Subtraction reset pattern				
7	+8,750	Addition set pattern				
		Addition reset pattern			○	
		Subtraction set pattern				○
		Subtraction reset pattern				

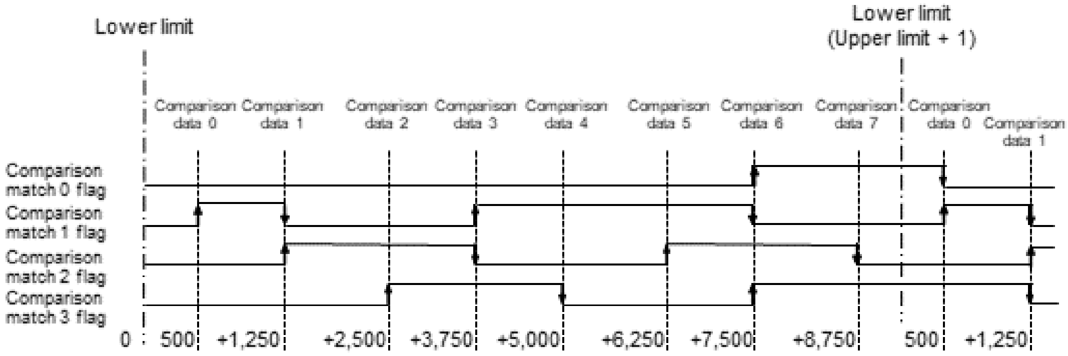
Info.

- The behavior of the comparison match flag that is performed when the count value reaches the target value can be changed separately for incrementation and decrementation.
- If the contents of comparison data 0 to 15 are duplicated and reset conditions are different, comparison data is prioritized in the following order.

  (High) 0 > 1 > 2 > 3 > 4 > 5 > 6 > 7 > 8 > 9 > 10 > 11 > 12 > 13 > 14 > 15 (Low)

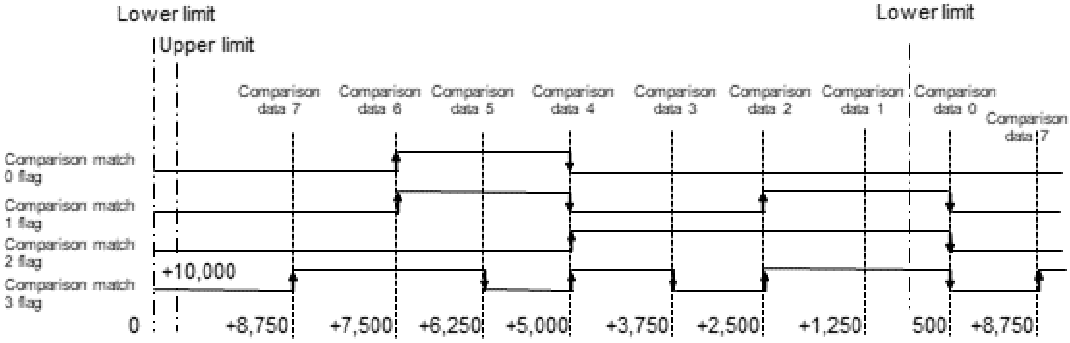
Behavior of comparison match flag during incremental count

When the current value matches the comparison data, the comparison match flag behaves according to the setting of "Addition set pattern" or "Addition reset pattern".



Behavior of comparison match flag during decrementation count

When the current value matches the comparison data, the comparison match flag behaves according to the setting of "Subtraction set pattern" or "Subtraction reset pattern".



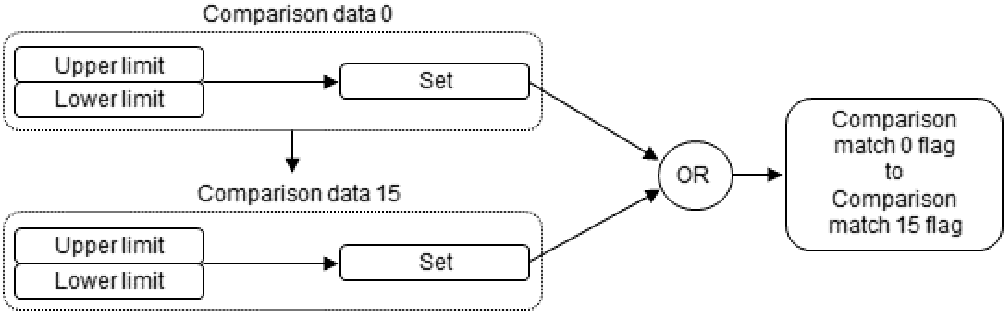
Info.

- For ring counters, comparison data can be set in an area including the lower and upper limits where count values are rolled over.

■ Parameter settings for band comparison

Parameter setting procedure

1. Click **Counter parameter**>**Counter (Ch0 or Ch1)**>**Comparison function**, change the "Select comparison function" parameter to "Band comparison", and execute "Set number of comparison data".
2. Click **Counter parameters**>**Counter (Ch0 or Ch1)**>**Comparison function**>**Comparison data** and specify upper and lower limits for each comparison data item.
3. Set whether to turn ON or OFF the comparison match flag when the count value exists in the zone (between the upper and lower limits).
4. Configure these settings for each comparison data item.



Info.

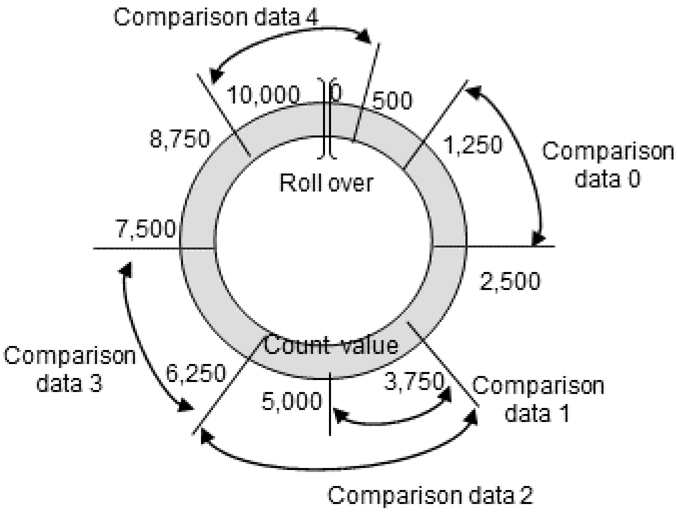
- A total of 16 comparison match flags can be set separately for each comparison data item.
- Multiple comparison data items can be set for the same band.
- For details on how to set up parameters, refer to "13.6.2 Setting Parameters with GM Programmer".

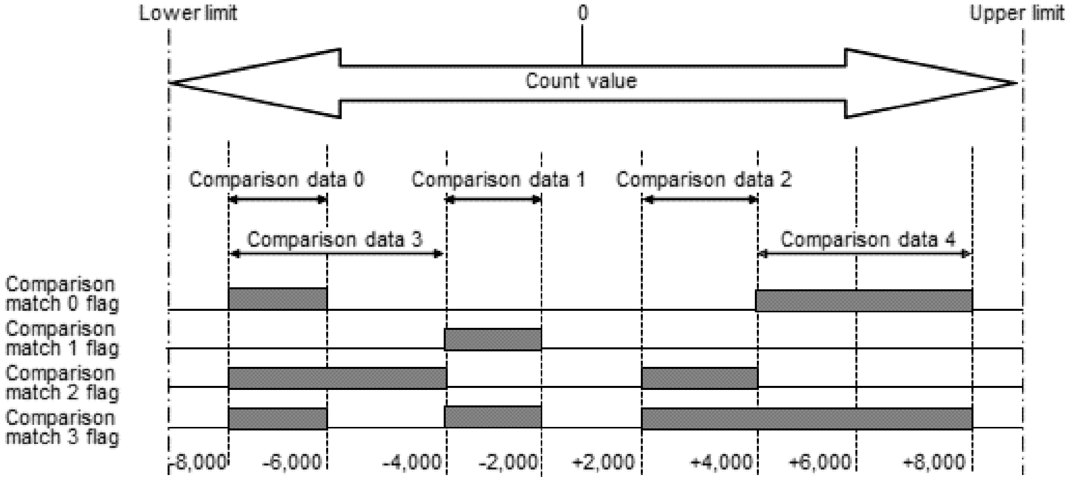
■ Setting example for band comparison

Output setting example

Comparison data	Band comparison value		Output setting	Comparison match 0 flag	Comparison match 1 flag	Comparison match 2 flag	Comparison match 3 flag
	Lower limit	Upper limit					
0	+1,250	2,500	Set	ON	OFF	OFF	OFF
1	+3,750	+5,000	Set	OFF	ON	OFF	ON
2	+3,750	+7,500	Set	OFF	OFF	OFF	ON
3	+6,250	+7,500	Set	ON	ON	ON	ON
4	+8,750	+500	Set	OFF	OFF	ON	OFF

Behavior of comparison match flag during band comparison





Info.

- For ring counters, comparison data can be set in an area including the lower and upper limits where count values are rolled over.

■ Parameter settings for external output function

- The comparison match 0 flag and comparison match 1 flag can be output externally using parameter settings.
- The Counter_Configuration parameter window is used to allocate the comparison match 0 flag and comparison match 1 flag to the external output function.

Info.

- For details on how to set up parameters, refer to "13.6.2 Setting Parameters with GM Programmer".

13.6.8 External Output Function

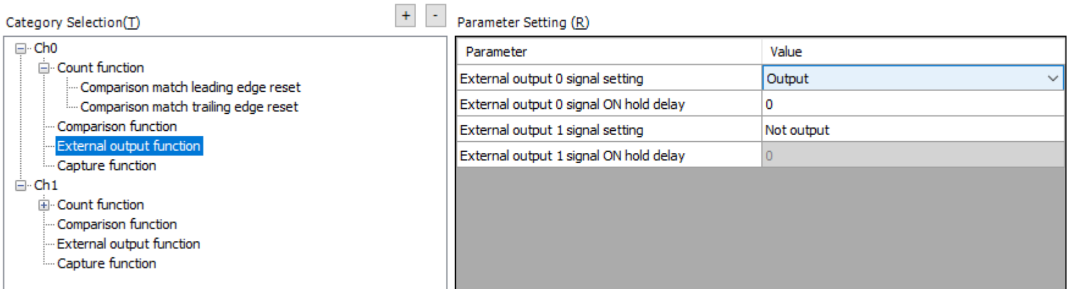
■ Overview of external output function

The comparison match 0 flag and comparison match 1 flag can be output externally using parameter settings.

Setup procedure

1. From "Device view" in the navigator pane, double-click "Counter_Configuration".
2. Click the "Counter parameter" tab.
3. For each counter, select "External output function "and set up each parameter.

Counter_Configuration parameter setting example



Category Selection(T)

- Ch0
  - Count function
    - Comparison match leading edge reset
    - Comparison match trailing edge reset
  - Comparison function
  - External output function
  - Capture function
- Ch1
  - Count function
  - Comparison function
  - External output function
  - Capture function

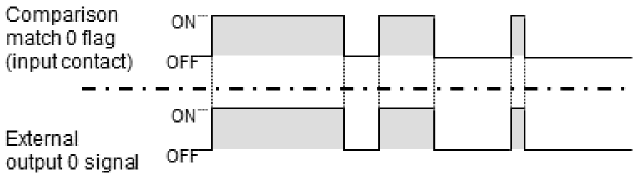
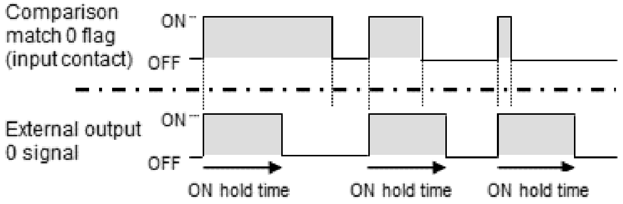
Parameter Setting (R)

Parameter	Value
External output 0 signal setting	Output
External output 0 signal ON hold delay	0
External output 1 signal setting	Not output
External output 1 signal ON hold delay	0

■ ON hold time (for band comparison only)

When the band comparison function is used, ON hold time can be set as an output signal.

Differences in behavior between settings

ON hold time	Timing chart for comparison match flag and external output signal
0	 Comparison match 0 flag (input contact) ON OFF External output 0 signal ON OFF
1 to 1,000 ms	 Comparison match 0 flag (input contact) ON OFF External output 0 signal ON OFF ON hold time ON hold time ON hold time

■ Forced output function

- If the Ch* external output 0* forced ON / OFF request bit is used, the external output 0 signal and external output 1 signal can be turned ON or OFF through user programs.

- The forced output function can be used to check wiring and for other purposes.

13.6.9 Capture Function

■ Setup procedure

1. From "Device view" in the navigator pane, double-click "Counter_Configuration".
2. Click the "Counter Parameters" tab.
3. For each counter, select "Capture function" and set up each parameter.

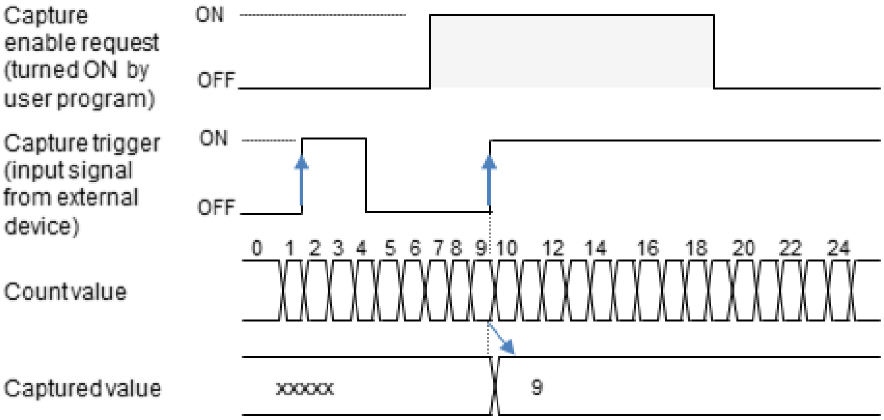
Info.

- For details on how to set up parameters, refer to "13.6.2 Setting Parameters with GM Programmer".

■ Types of capture function

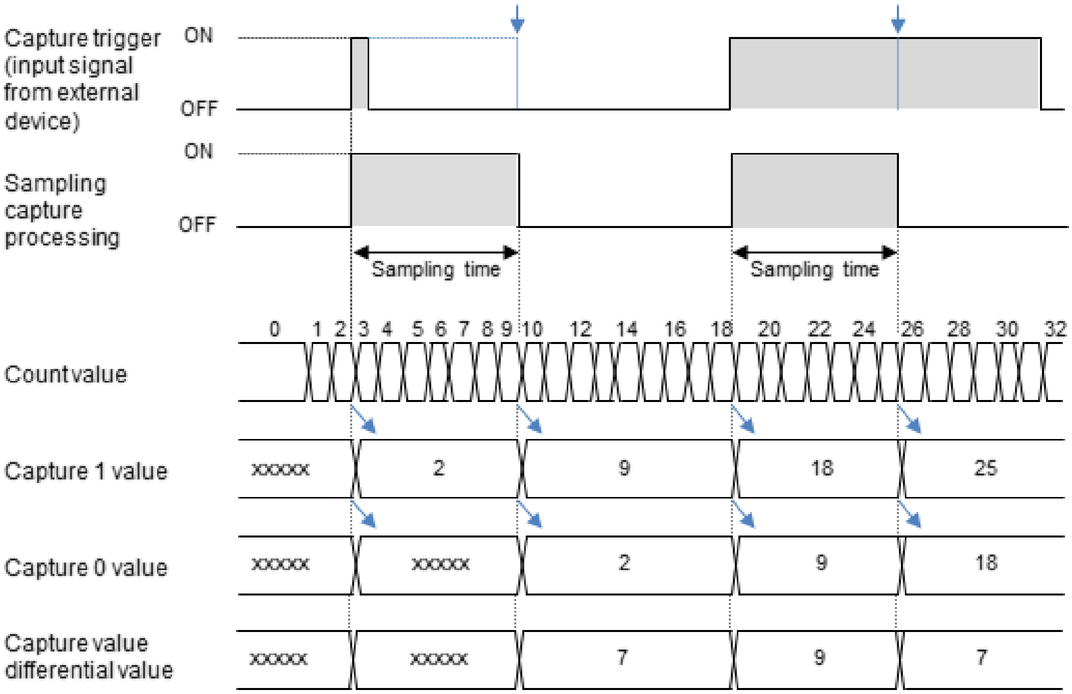
Capture function

- The count value at the point in time when the input signal from an external device changes is stored in the capture 0 value or capture 1 value register.


Capture enable request (turned ON by user program)
ON
OFF
Capture trigger (input signal from external device)
ON
OFF
0 1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24
Count value
Captured value
xxxxx
9

Sampling capture function

- The count value when the specified sampling time elapses after the input signal from an external device changes is stored in the capture 0 value or capture 1 value registers.
- The count value equivalent to the sampling time can be monitored by reading the capture differential value.



■ Comparison between capture function and sampling capture function

Available conditions differ between the functions.

Comparison between both functions

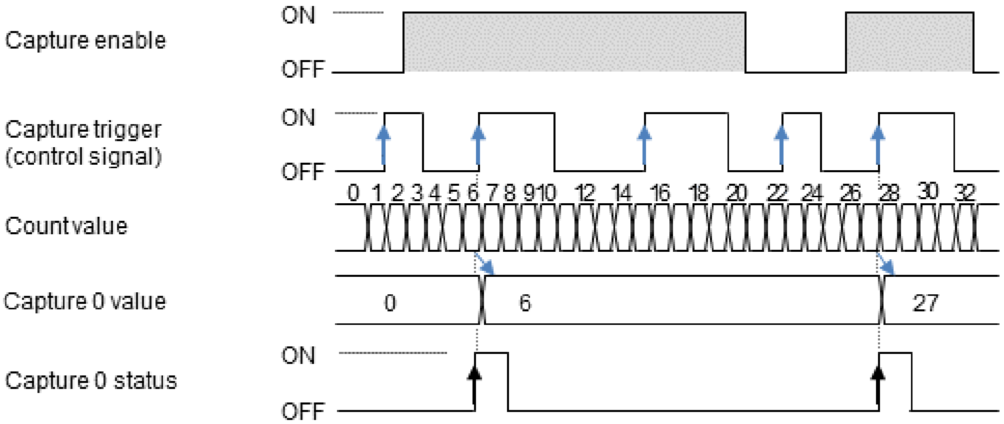
Item		Capture function	Sampling capture function
Number of points that can be used		Max. 2 points	Max. 1 point
		The capture function cannot be used when the sampling capture function is used.	
Registers used		Capture 0 value: 1 register (2 words) Capture 1 value: 1 register (2 words) These registers can be used individually.	Capture 0 value: 1 register (2 words) Capture 1 value: 1 register (2 words) These registers are used simultaneously.
		Capture differential value: 1 register (2 words)	
Enable condition		The capture function is enabled while the capture enable request bit is ON.	Always enabled
Capture flags		The Counter_Configuration parameter window is used to allocate capture flags.	
	Control 0 signal	Used as a trigger for capture 0 or capture 1.	Used as a trigger for the sampling capture function
	Control 1 signal	Used as a trigger for capture 0 or capture 1.	Not use
	Validity condition	Activated when either of the following conditions is met. At rising edge of control 0 / 1 signal At falling edge of control 0 / 1 signal	Activated when either of the following conditions is met. Control 0 signal (positive logic) Control 0 signal (negative logic)

Item		Capture function	Sampling capture function
		By allocating one of the control signals to the same capture number, the capture function can be allocated as the rising or falling edge of the signal.	Capture enable request bit (positive logic)
Clearing the capture * status bit		The capture * status bit is cleared automatically each time I/O refresh occurs.	

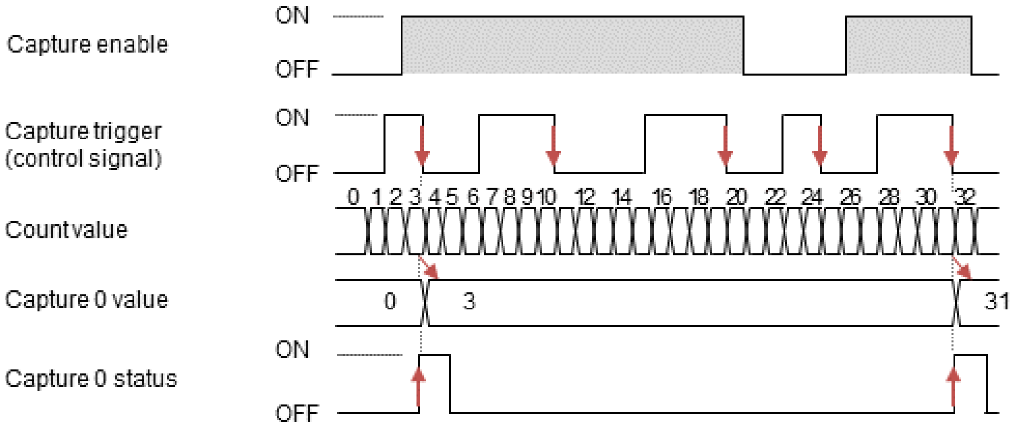
■ One operation

- When the capture enable request bit enables the capture function, capture is executed when the first capture flag becomes enabled.
- The behaviors differ according to the validity condition (rising edge or falling edge) of the capture flag to be enabled, as below.

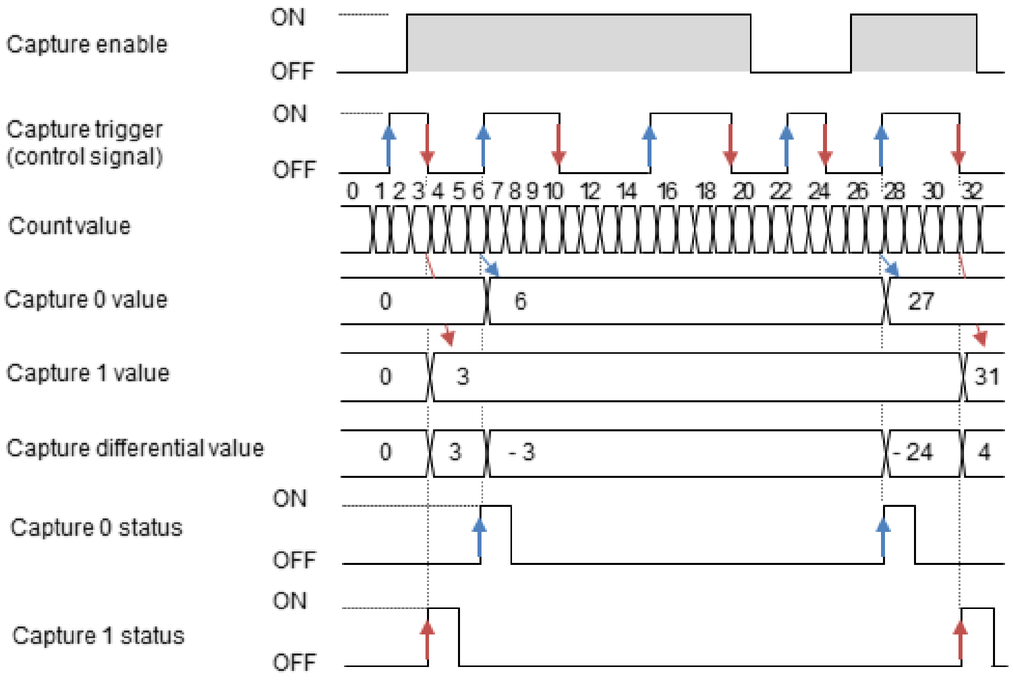
When "rising edge" is specified as the capture trigger condition



When "falling edge" is specified as the capture trigger condition



When "rising edge" and "falling edge" of the same signal are specified as the capture trigger conditions



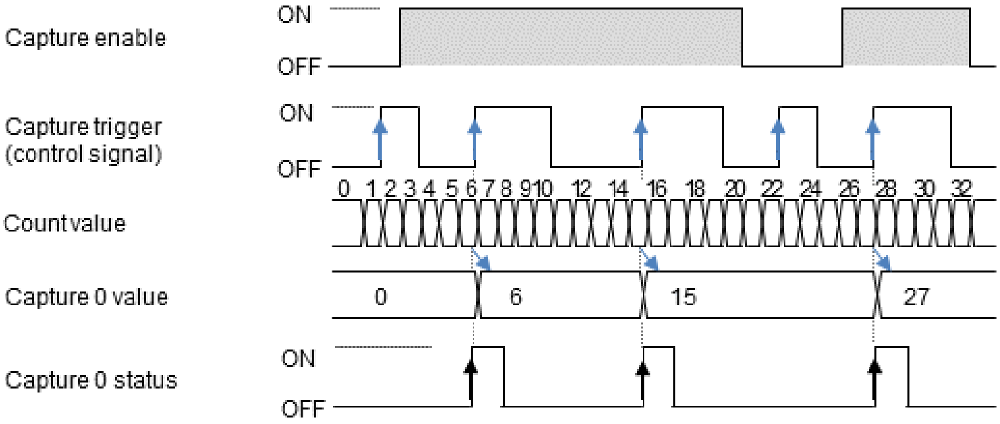
Info.

- The sign of the capture differential value changes according to the sequence of the capture enable request bit and capture flag.

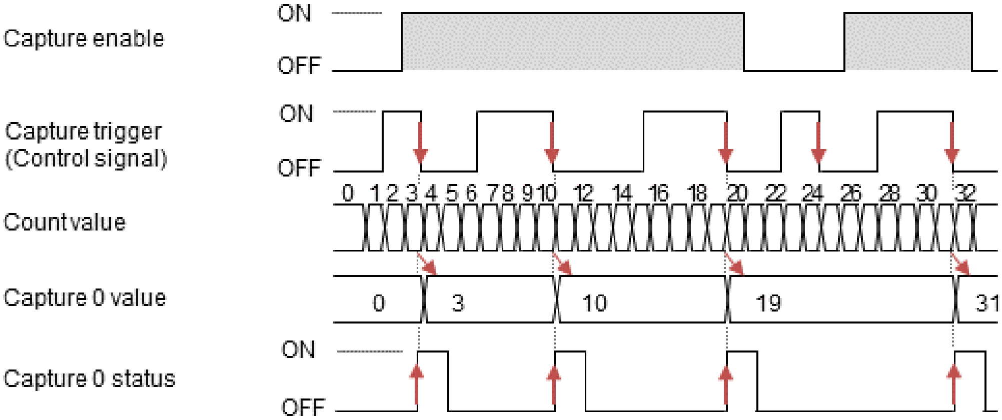
■ Continuous operation

- When the capture enable request bit enables the capture function, capture is executed every time a capture flag becomes enabled.
- The behaviors differ according to the validity condition (rising edge or falling edge) of the capture flag to be enabled, as below.

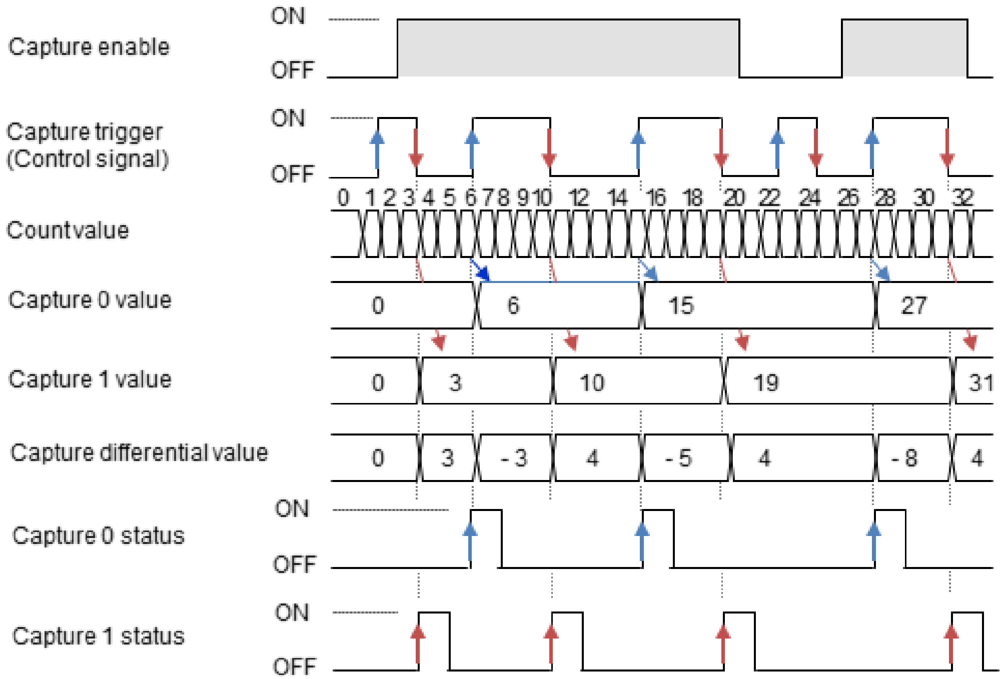
When "rising edge" is specified as the capture trigger condition



When "falling edge" is specified as the capture trigger condition



When "rising edge" and "falling edge" of the same signal are specified as the capture trigger conditions



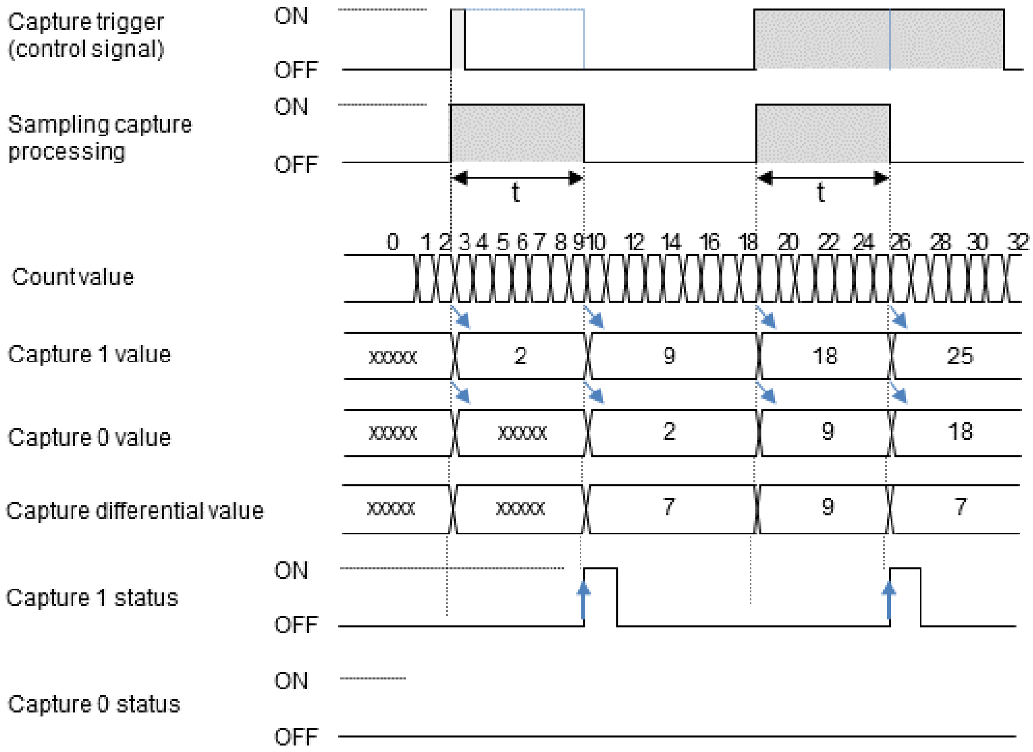
Info.

- The sign of the capture differential value changes according to the sequence of the capture enable request bit and capture flag.
- For continuous operation, capture 0 value, capture 1 value, and capture differential value are overwritten each time a capture operation is completed.

■ Sampling capture function (one operation)

- When the specified sampling time elapses after the capture flag turns ON or OFF, the count value is stored in the capture 0 value and capture 1 value registers and the differential value is stored in the capture differential value register.
- The sampling capture function is always executable when the control 0 signal is allocated to the sampling capture function.
- The trigger condition that starts sampling capture can be selected from control 0 signal (positive logic), control 1 signal (negative logic), and capture enable request bit (positive logic).
- For one operation, the capture 1 status bit is activated. Note that the capture 0 status bit is not activated.

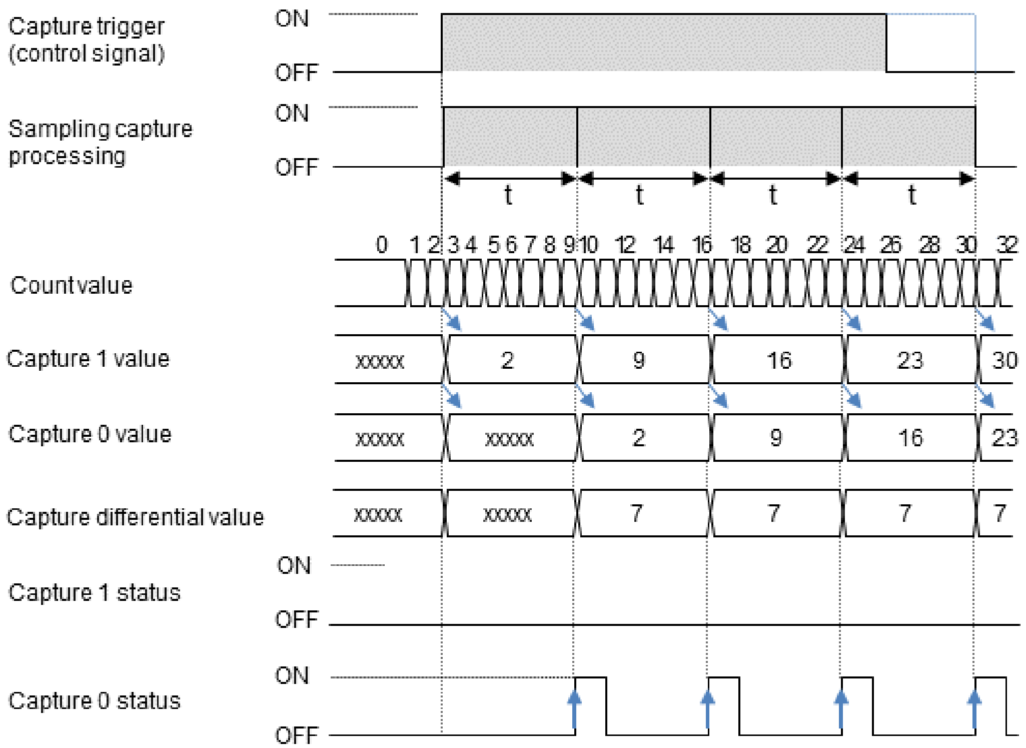
Timing chart



■ Sampling capture function (continuous operation)

- Each time the specified sampling time elapses after the capture flag turns ON or OFF, successively, the count value is stored in the capture 0 value and capture 1 value registers and the differential value is stored in the capture differential value register.
- The sampling capture function is always executable when the control 0 signal is allocated to the sampling capture function.
- The trigger condition that starts sampling capture can be selected from control 0 signal (positive logic), control 0 signal (negative logic), and capture enable request bit (positive logic).
- For continuous operation, the capture 0 status bit is activated. Note that the capture 1 status bit is not activated.

Timing chart



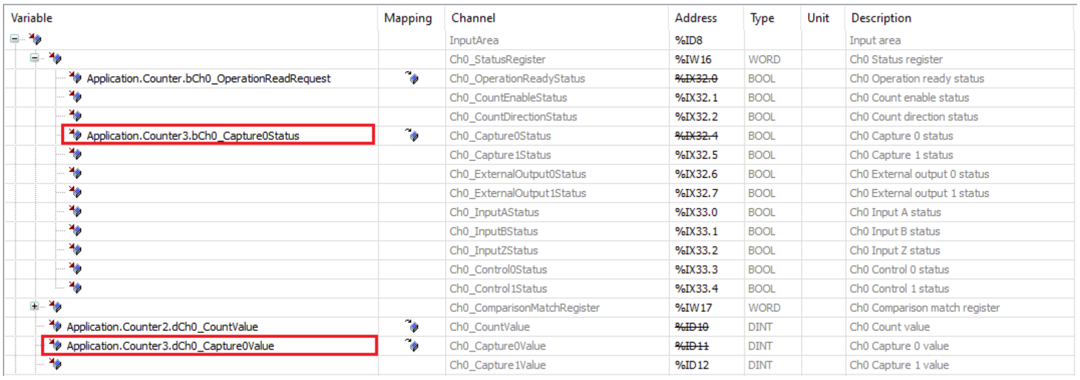
■ Reading captured data

Areas where captured data is stored

- The latest captured data is stored in the capture 0 value and capture 1 value registers.
- Captured data is stored as signed 32-bit data (-2,147,483,648 to 2,147,483,647).

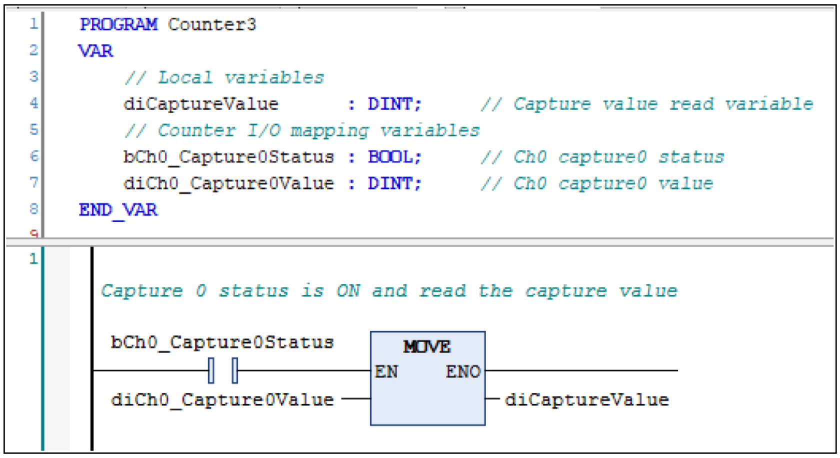
Sample program

To read and write channel data, variables are mapped to channels in the same way as in "13.6.5 Operation Ready Request". In this example, variables are mapped to the channel used for reading captured data for Counter Ch0 (the name of sample POU is created as "Counter3").



Variable	Mapping	Channel	Address	Type	Unit	Description
		InputArea	%ID8			Input area
		Ch0_StatusRegister	%IW16	WORD		Ch0 Status register
Application.Counter.bCh0_OperationReadRequest		Ch0_OperationReadyStatus	~~%IX32.0~~	BOOL		Ch0 Operation ready status
		Ch0_CountEnableStatus	%IX32.1	BOOL		Ch0 Count enable status
		Ch0_CountDirectionStatus	%IX32.2	BOOL		Ch0 Count direction status
Application.Counter3.bCh0_Capture0Status		Ch0_Capture0Status	~~%IX32.4~~	BOOL		Ch0 Capture 0 status
		Ch0_Capture1Status	%IX32.5	BOOL		Ch0 Capture 1 status
		Ch0_ExternalOutput0Status	%IX32.6	BOOL		Ch0 External output 0 status
		Ch0_ExternalOutput1Status	%IX32.7	BOOL		Ch0 External output 1 status
		Ch0_InputAStatus	%IX33.0	BOOL		Ch0 Input A status
		Ch0_InputBStatus	%IX33.1	BOOL		Ch0 Input B status
		Ch0_InputZStatus	%IX33.2	BOOL		Ch0 Input Z status
		Ch0_Control0Status	%IX33.3	BOOL		Ch0 Control 0 status
		Ch0_Control1Status	%IX33.4	BOOL		Ch0 Control 1 status
		Ch0_ComparisonMatchRegister	%IW17	WORD		Ch0 Comparison match register
Application.Counter2.dCh0_CountValue		Ch0_CountValue	~~%ID10~~	DINT		Ch0 Count value
Application.Counter3.dCh0_Capture0Value		Ch0_Capture0Value	~~%ID11~~	DINT		Ch0 Capture 0 value
		Ch0_Capture1Value	%ID12	DINT		Ch0 Capture 1 value

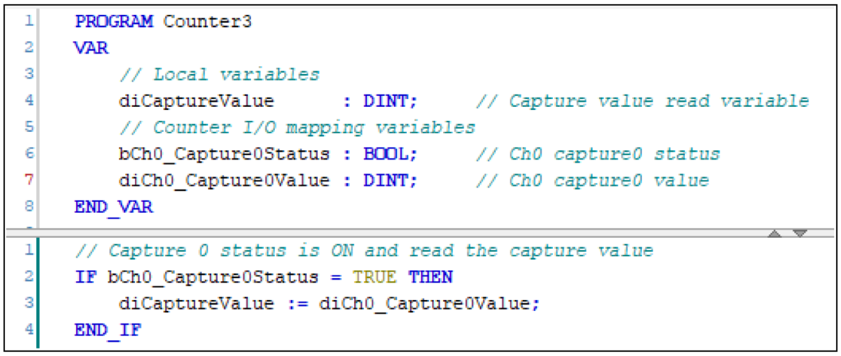
LD program



```
PROGRAM Counter3
VAR
    // Local variables
    diCaptureValue       : DINT;      // Capture value read variable
    // Counter I/O mapping variables
    bCh0_Capture0Status : BOOL;      // Ch0 capture0 status
    diCh0_Capture0Value : DINT;      // Ch0 capture0 value
END_VAR
```

Capture 0 status is ON and read the capture value

ST program



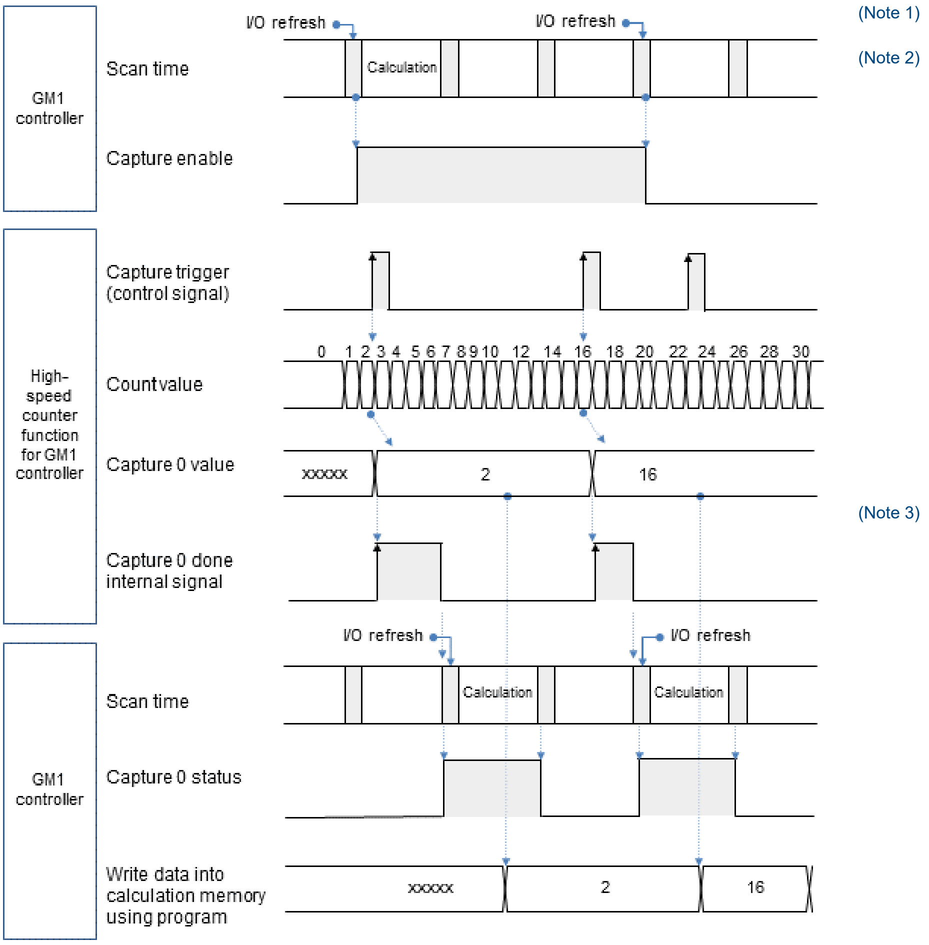
```
PROGRAM Counter3
VAR
    // Local variables
    diCaptureValue       : DINT;      // Capture value read variable
    // Counter I/O mapping variables
    bCh0_Capture0Status : BOOL;      // Ch0 capture0 status
    diCh0_Capture0Value : DINT;      // Ch0 capture0 value
END_VAR
```

```
// Capture 0 status is ON and read the capture value
IF bCh0_Capture0Status = TRUE THEN
    diCaptureValue := diCh0_Capture0Value;
END_IF
```

■ Behaviors and read operations of capture * status bits

Basic behaviors

- When capture is completed, it is reflected in the capture * status bit for each I/O refresh.
- For the capture function, the capture * status bit is reset when the capture enable request bit turns ON.

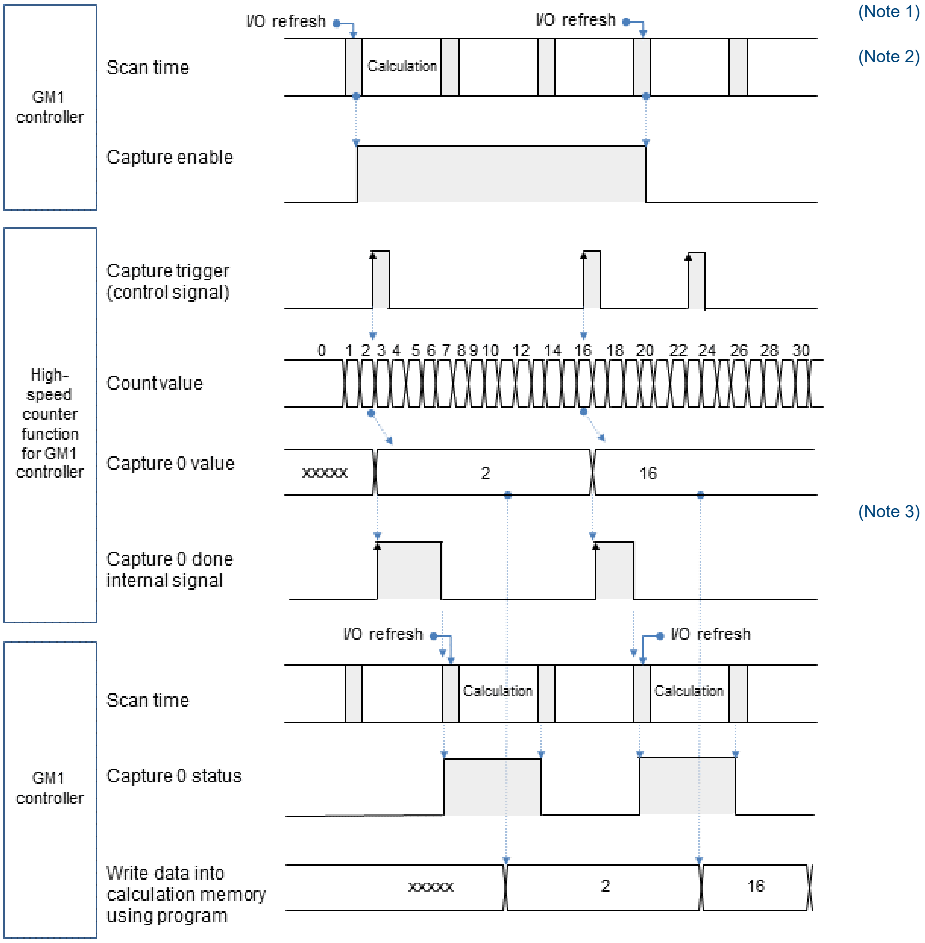


(Note 1) The capture enable request bit is turned ON by a user program.

(Note 2) Each time a capture flag turns ON, a capture operation is performed asynchronously with user program execution. The capture completion internal flag used by the system is reset each time an I/O refresh is performed.

(Note 3) Capture * status bits are reflected each time an I/O refresh is performed. Capture * status bits are used to read capture 0 value, capture 1 value, and capture differential value as arbitrary variables through user programs. These values are read at the time of relevant calculation processing.

Processing when capture trigger input occurs frequently



(Note 1) The capture enable request bit is turned ON by a user program.

(Note 2) Each time a capture flag turns ON, a capture operation is performed asynchronously with user program execution. The capture completion internal flag used by the system is reset each time an I/O refresh is performed.

(Note 3) Capture * status bits are reflected each time an I/O refresh is performed. If multiple capture operations are performed continuously, the capture 0 completion status bit or capture 1 completion status bit will remain ON.

Info.

- If control signals used as capture triggers are input frequently, the capture 0 completion status bit or capture 1 completion status bit will remain ON. Take care when reading multiple captured data items.

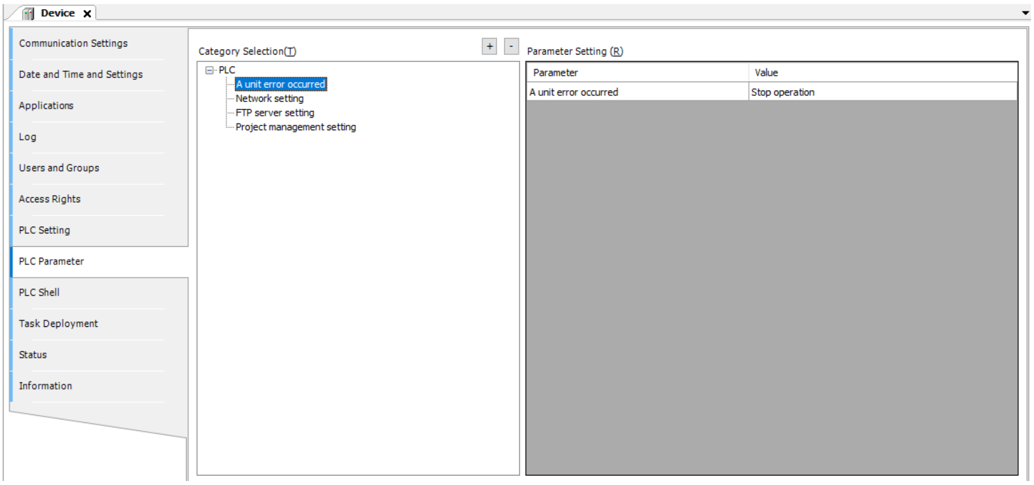
13.6.10 Unit Error

Overflow and underflow errors with the counter unit are operation stop errors.

To continue the operating status when an error occurs, change the value of PLC parameter "A unit error occurred" to "Continue operation".

1 2 Procedure

1. From "Device view" in the navigator pane, double-click the "Device" object.
2. Click the "PLC Parameters" tab in the Device window.
3. Change the value of "A unit error occurred" to "Continue operation".

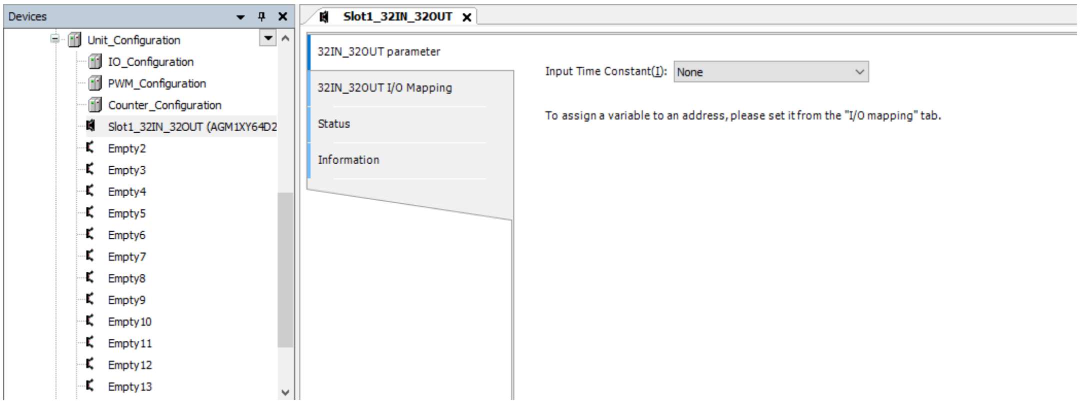


Parameter	Value
A unit error occurred	Stop operation

13.7 Settings of I/O Unit

Expansion I/O units are classified into 64-point input, 64-point output, and 32-point I/O units. This section explains 32-point I/O units as an example.

Setting Parameters with GM Programmer



Input Time Constant(I): None

To assign a variable to an address, please set it from the "I/O mapping" tab.

13.7.1 Parameter Settings

■ Parameter

Setting item	Settings	Default value	Description
Input time constant	None 0.1 ms 0.5 ms 1 ms 5 ms 10 ms 20 ms 70 ms	None	Input time constant

■ Accuracy of time constant

There is an error in the time constant to be set, so please check in advance before selecting the setting value.The precision of each time constant is shown in the table below.

Set value	Accuracy	
	Min.	Max.
No time constant setting	-	-
0.1ms	0.1ms	0.2ms
0.5ms	0.3ms	0.7ms
1ms	0.7ms	1.3ms
5ms	3.0ms	5.2ms
10ms	6.0ms	10.4ms

Set value	Accuracy	
	Min.	Max.
20ms	12.1ms	20.7ms
70ms	48.6ms	82.8ms

13.7.2 I/O Mapping for I/O Unit

Channel	Type	Description	Remarks
Ch0_In	WORD	Ch0_In	
Ch1_In	WORD	Ch1_In	
Ch0_Out	WORD	Ch0_Out	
Ch1_Out	WORD	Ch1_Out	

■ Ch*_In

* represents 0 or 1.

Channel	Type	Description	Remarks
Ch*_In00	BOOL	Ch*_In00	
Ch*_In01	BOOL	Ch*_In01	
Ch*_In02	BOOL	Ch*_In02	
Ch*_In03	BOOL	Ch*_In03	
Ch*_In04	BOOL	Ch*_In04	
Ch*_In05	BOOL	Ch*_In05	
Ch*_In06	BOOL	Ch*_In06	
Ch*_In07	BOOL	Ch*_In07	
Ch*_In08	BOOL	Ch*_In08	
Ch*_In09	BOOL	Ch*_In09	
Ch*_In10	BOOL	Ch*_In10	
Ch*_In11	BOOL	Ch*_In11	
Ch*_In12	BOOL	Ch*_In12	
Ch*_In13	BOOL	Ch*_In13	
Ch*_In14	BOOL	Ch*_In14	
Ch*_In15	BOOL	Ch*_In15	

■ Ch*_Out

* represents 0 or 1.

Channel	Type	Description	Remarks
Ch*_Out00	BOOL	Ch*_Out00	
Ch*_Out01	BOOL	Ch*_Out01	

Channel	Type	Description	Remarks
Ch*_Out02	BOOL	Ch*_Out02	
Ch*_Out03	BOOL	Ch*_Out03	
Ch*_Out04	BOOL	Ch*_Out04	
Ch*_Out05	BOOL	Ch*_Out05	
Ch*_Out06	BOOL	Ch*_Out06	
Ch*_Out07	BOOL	Ch*_Out07	
Ch*_Out08	BOOL	Ch*_Out08	
Ch*_Out09	BOOL	Ch*_Out09	
Ch*_Out10	BOOL	Ch*_Out10	
Ch*_Out11	BOOL	Ch*_Out11	
Ch*_Out12	BOOL	Ch*_Out12	
Ch*_Out13	BOOL	Ch*_Out13	
Ch*_Out14	BOOL	Ch*_Out14	
Ch*_Out15	BOOL	Ch*_Out15	

(MEMO)

14 Communication Function

14.1 Overview of Communication Function

The GM1 controller allows general-purpose communication and various protocols with external devices via COM and LAN ports.

The following table shows the functions supported by each port.

Port	Supported protocol	Maximum number of connections
COM port	General-purpose communication	1 port
	MODBUS-RTU(Master/Slave)	
LAN port 1/ port 2	General-purpose communication	
	Communication with the display （CodesysV3）	16 connections in total
	OPC UA （Server）	2 connections in total
	FTP （Server）	3 connections in total
	MQTT Client	3 connections in total
	MODBUS-TCP(Master/Slave)	port 1 : 16 connections in total port 2 : 32 connections in total
LAN port2	EtherNet/IP(Scanner/Adapter)	

Info.

- The IP address of the LAN port and FTP server setting can be changed using the "PLC Parameters" tab in the Device setting window.

  For details, refer to "5.2 Setting up the GM1 Controller" and "14.7.3 List of Setting Items".
- LAN port 2 has a limit on the total number of MODBUS-TCP and EtherNet / IP connections.

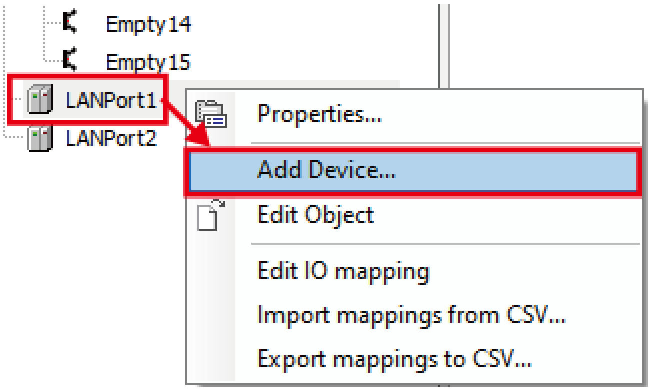
14.1.1 Adding Network Communication Devices

Add a communication device object to the device object of a LAN port.

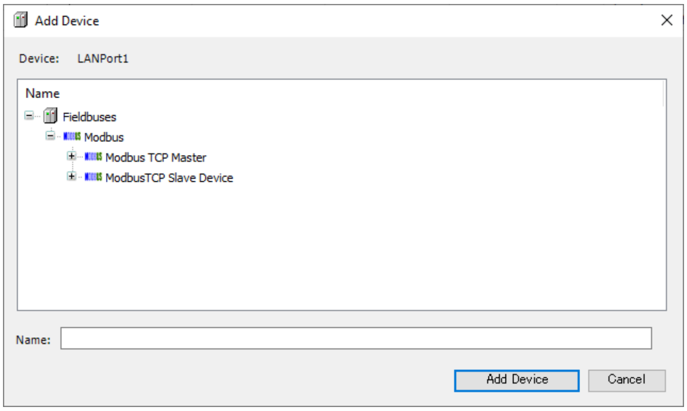
For example, use the following procedure to add "ModbusTCP Slave Device" to LAN port 1.

Procedure

1. Right-click the "LANPort1" object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.



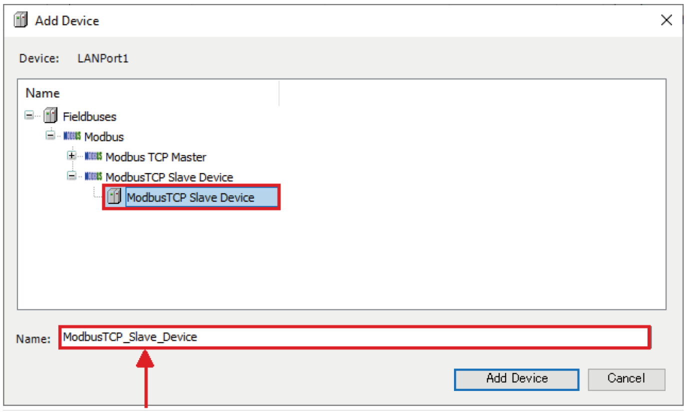
The "Add Device" dialog box will be displayed.



Device: LANPort1

Name:

2. Select "ModbusTCP Slave Device".

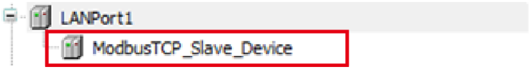


Device: LANPort1

Name: ModbusTCP_Slave_Device

Any device name can be assigned.

3. Click the [Add Device] button.
The selected "ModbusTCP_Slave_Device" object will be added to the navigator pane.

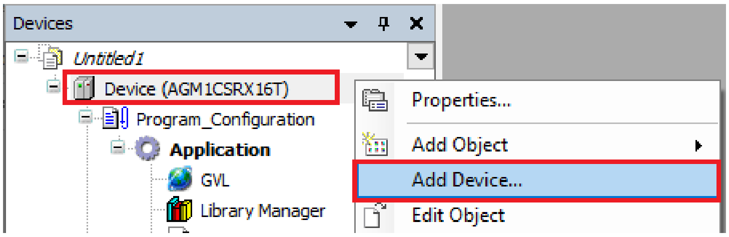


For details on how to set up Modbus TCP, refer to "14.3 MODBUS".

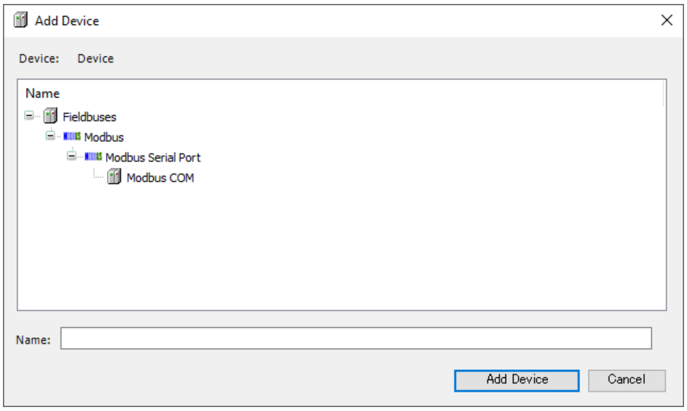
14.1.2 Adding Serial Communication Devices

1 2 Procedure

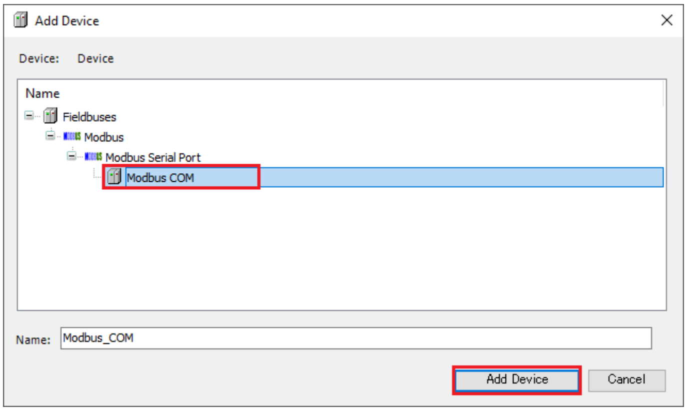
1. Right-click the "Device" object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.



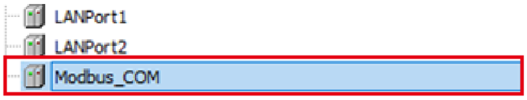
2. Select "Modbus COM".



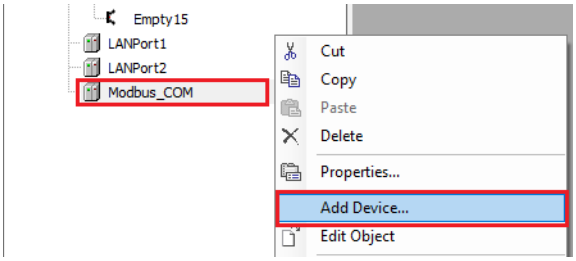
Name: Modbus_COM

3. Click the [Add Device] button.

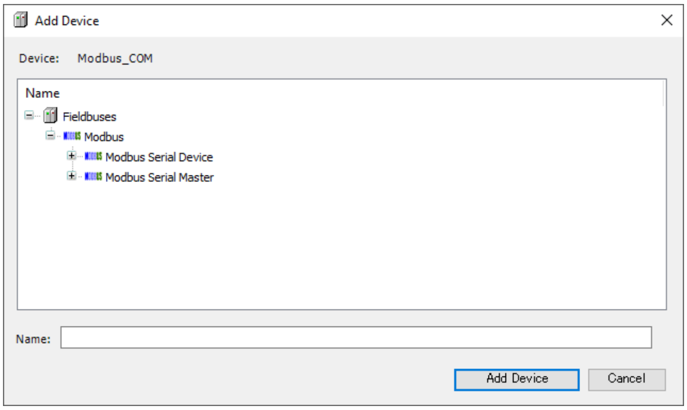
The selected "Modbus_COM" object will be added to the navigator pane.



4. Right-click the "Modbus_COM" object and then select "Add Device" from the context-sensitive menu that is displayed.



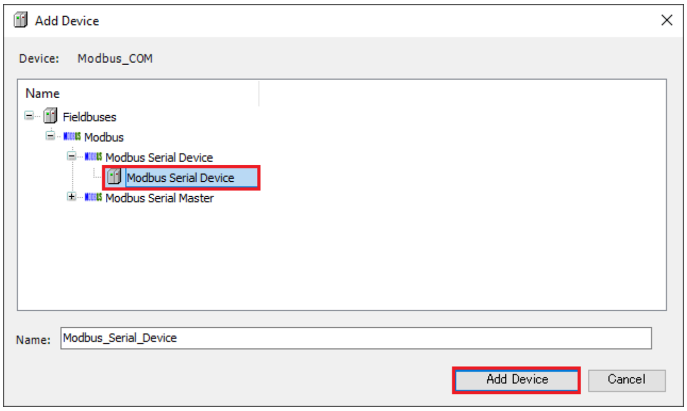
The "Add Device" dialog box will be displayed.



Device: Modbus_COM

Name:

5. Select "Modbus_Serial_Device".

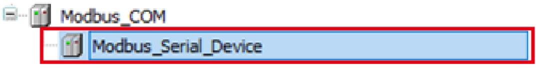


Device: Modbus_COM

Name: Modbus_Serial_Device

6. Click the [Add Device] button.

The selected "Modbus_Serial_Device" object will be added to the navigator pane.



For details on how to set up, refer to "14.3 MODBUS".

List of devices

Check the following list for devices that can be added.

Addition source	Added device
Device	Modbus COM
RTEX_Master	RTEX_A5N RTEX_A6N
EtherCAT_Master_SoftMotion	Various EtherCAT slaves can be selected.
SoftMotion General Axis Pool	SM_FreeEncoder SM_Drive_Virtual
Unit_Configuration or Empty1~15	AGM1X64D2 AGM1Y64T AGM1Y64P AGM1XY64D2T AGM1XY64D2P AGM1AD8 AGM1DA4 AGM1PG04T AGM1PG04L
LAN Port1	Modbus TCP Master ModbusTCP Slave Drive
LAN Port2	Modbus TCP Master ModbusTCP Slave Drive EtherNet/IP Scanner EtherNet/IP Adapter
Modbus TCP Master	Modbus TCP Slave
EtherNet/IP Scanner	Remote adapter for each device can be selected
EtherNet/IP Adapter	EtherNet/IP Module
Modbus COM	Modbus Serial Device Modbus Master, COM Port
Modbus Master, COM Port	Modbus Slave COM Port

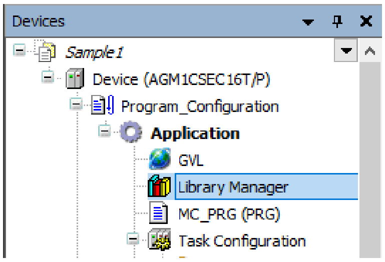
14.2 General-purpose Communication

14.2.1 General-purpose Communication (Ethernet)

This section explains how to use the CAA_NetBaseServices library, in the following order.

1. Library_Manager
2. TCP CLIENT processing example
3. TCP SERVER processing example
4. UDP processing example

Double-click Library_Manager in the navigator pane.



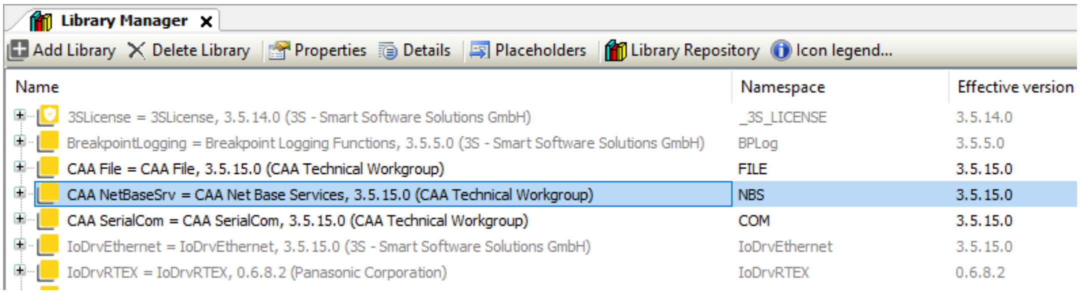
Devices

- Sample1
  - Device (AGM1CSEC16T/P)
    - Program_Configuration
      - Application
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Task Configuration

The Library_Manager setting window will be displayed.

■ Library_Manager

Check that the following CAA_NetBaseServices library is registered in Library_Manager.



Library Manager

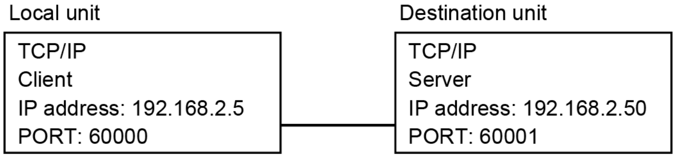
Add Library | Delete Library | Properties | Details | Placeholders | Library Repository | Icon legend...

Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CAA File = CAA File, 3.5.15.0 (CAA Technical Workgroup)	FILE	3.5.15.0
CAA NetBaseSrv = CAA Net Base Services, 3.5.15.0 (CAA Technical Workgroup)	NBS	3.5.15.0
CAA SerialCom = CAA SerialCom, 3.5.15.0 (CAA Technical Workgroup)	COM	3.5.15.0
IoDrvEthernet = IoDrvEthernet, 3.5.15.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.15.0
IoDrvRTEX = IoDrvRTEX, 0.6.8.2 (Panasonic Corporation)	IoDrvRTEX	0.6.8.2

■ TCP CLIENT processing example

The following is a processing example of data transmission / reception via TCP when the local unit is TCP CLIENT.

This processing example assumes the following operating environment.



Local unit

TCP/IP
Client
IP address: 192.168.2.5
PORT: 60000

Destination unit

TCP/IP
Server
IP address: 192.168.2.50
PORT: 60001

Processing for data transmission / reception

The processing for data transmission / reception is as follows:

- TCP client connection processing
- Reception start processing

- Transmission processing

Explanation of variables

Process

When the value is rewritten, the following processing is executed. After the execution is completed, the variable is set to 0 (invalid value).

1 = TCP client connection processing

2 = Reception start processing

3 = Transmission processing

ClientAddr

The IP address of the destination unit is set.

Port

The port number of the destination unit is set.

TimeOut

A connection timeout period is set.

SendData

Data to be sent is set.

RecvBuf

Received data is stored.

RecvCount

The number of receptions is stored.

RecvSize

The size of received data is stored.

Result

The result of processing execution is stored. (TRUE: Error occurrence, FALSE: Normal termination)

If the result of processing execution is abnormal, check the error code of each processing.

- NBS_ClientError: Result of TCP client connection processing
- NBS_WriteError: Result of transmission processing
- NBS_ReadError: Result of reception start processing

Operation example

The TCP client connects to the TCP server.

- The value of "Process" is changed to 1.

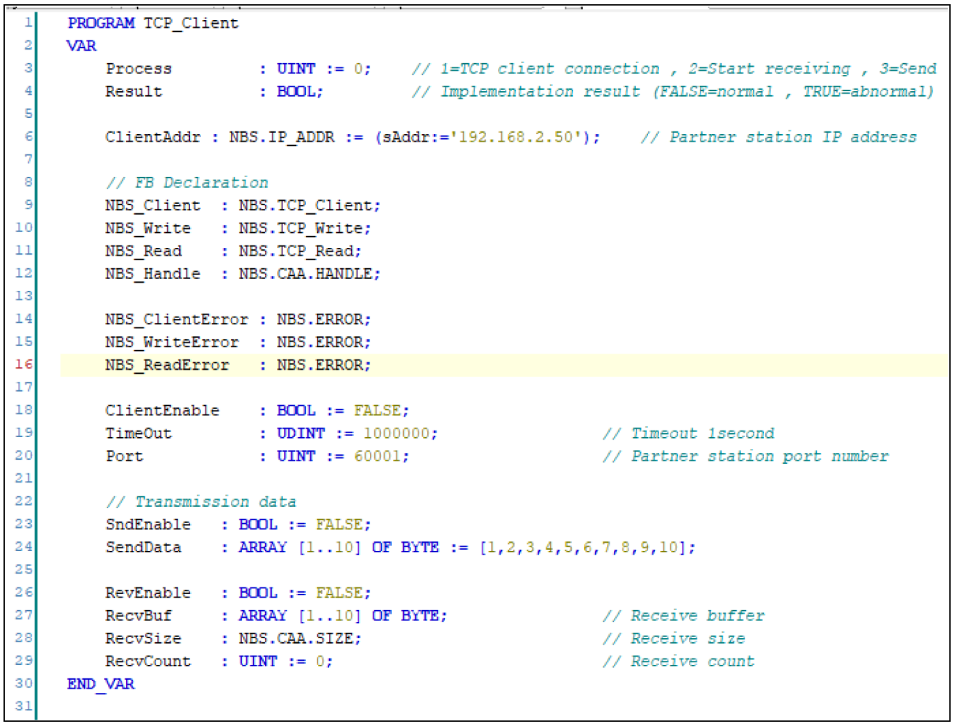
The local unit is ready to receive data. In this state, the local unit can receive data from the destination unit.

- The value of "Process" is changed to 2.

The local unit sends data to the destination unit. 10-byte data is sent to the destination unit.

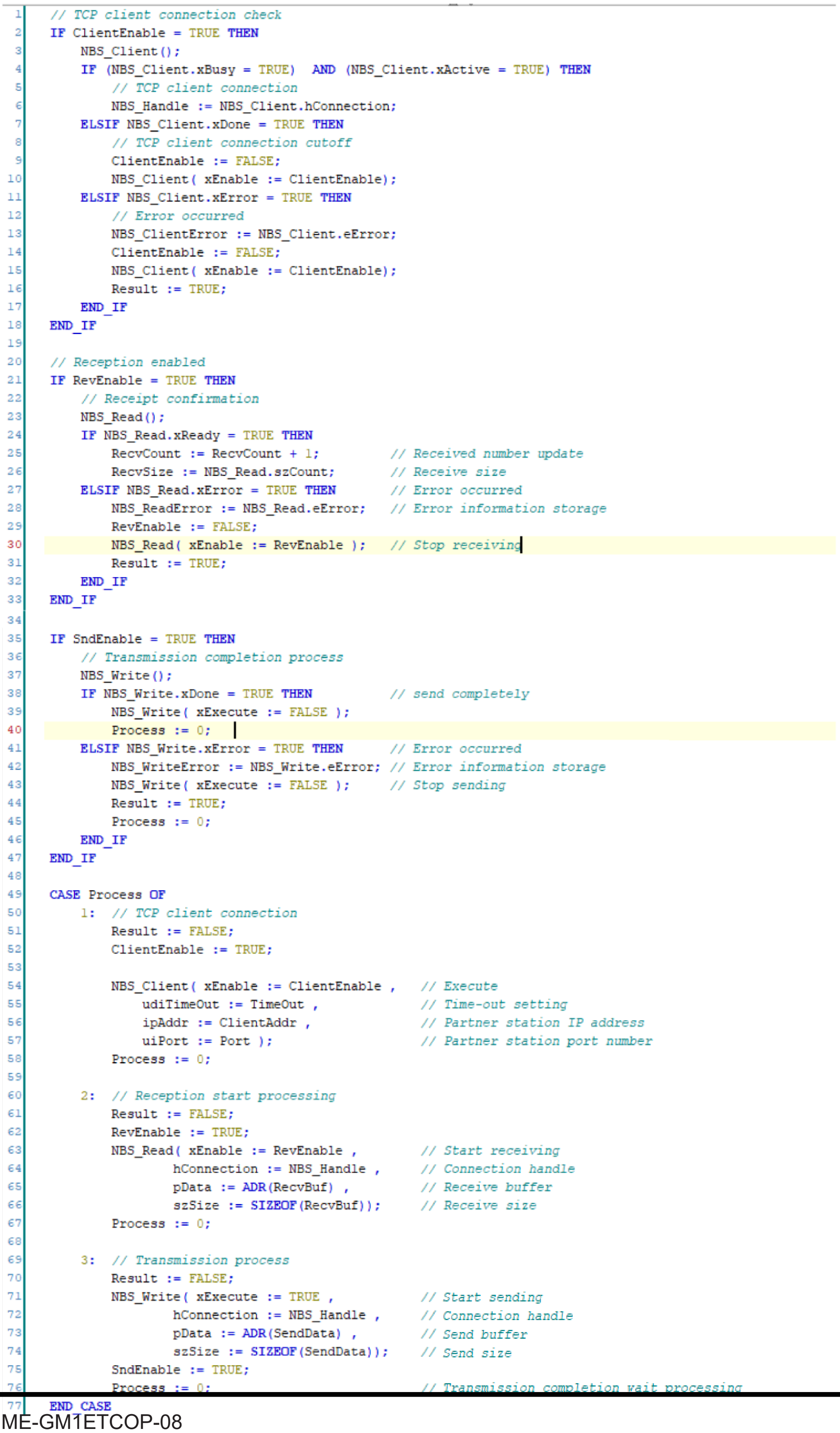
- The value of "Process" is changed to 3.

Declaration section (common to ST and LD programming languages)



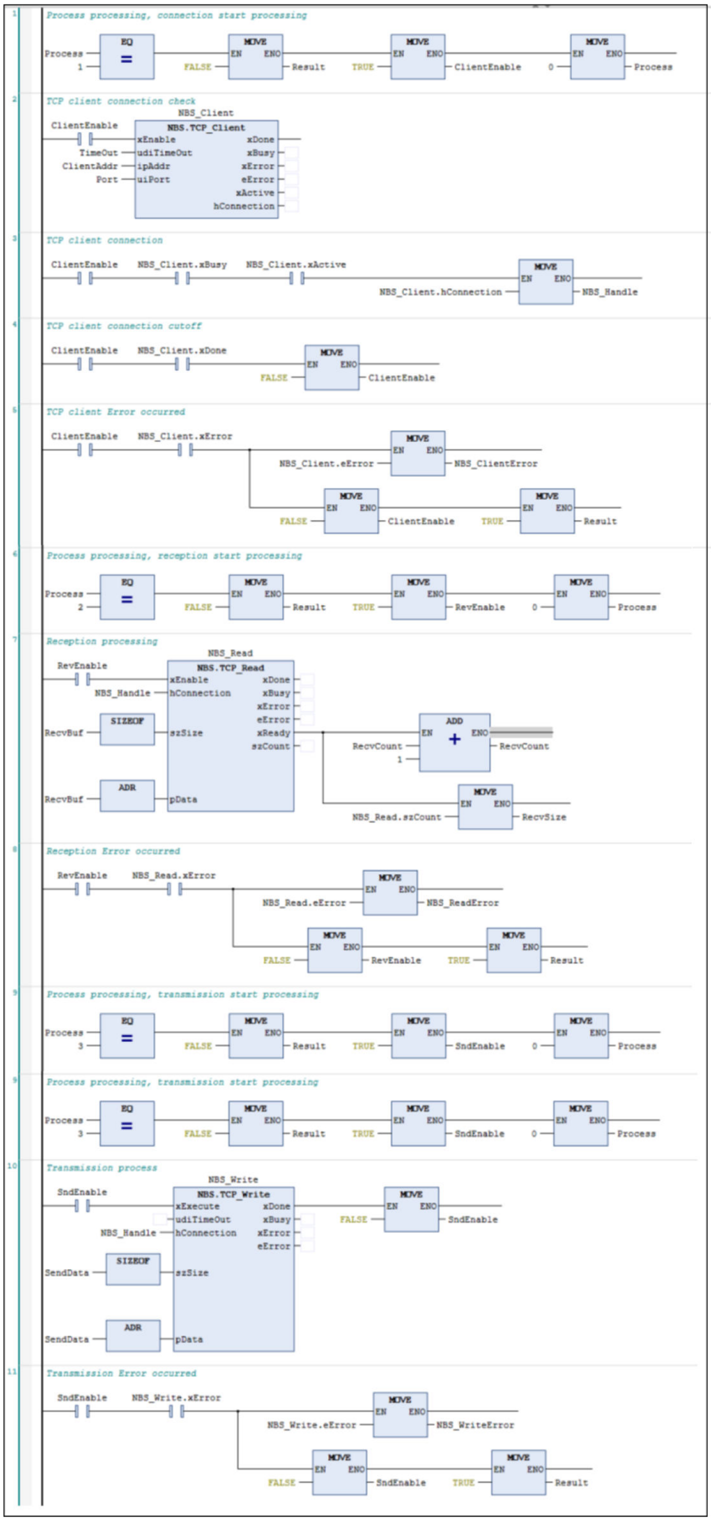
```
PROGRAM TCP_Client
VAR
    Process         : UINT := 0;    // 1=TCP client connection , 2=Start receiving , 3=Send
    Result          : BOOL;         // Implementation result (FALSE=normal , TRUE=abnormal)

    ClientAddr : NBS.IP_ADDR := (sAddr:='192.168.2.50');    // Partner station IP address

    // FB Declaration
    NBS_Client  : NBS.TCP_Client;
    NBS_Write   : NBS.TCP_Write;
    NBS_Read    : NBS.TCP_Read;
    NBS_Handle  : NBS.CAA.HANDLE;

    NBS_ClientError : NBS.ERROR;
    NBS_WriteError  : NBS.ERROR;
    NBS_ReadError   : NBS.ERROR;

    ClientEnable    : BOOL := FALSE;
    TimeOut         : UDINT := 1000000;                 // Timeout 1second
    Port            : UINT := 60001;                    // Partner station port number

    // Transmission data
    SndEnable   : BOOL := FALSE;
    SendData    : ARRAY [1..10] OF BYTE := [1,2,3,4,5,6,7,8,9,10];

    RevEnable   : BOOL := FALSE;
    RecvBuf     : ARRAY [1..10] OF BYTE;                // Receive buffer
    RecvSize    : NBS.CAA.SIZE;                         // Receive size
    RecvCount   : UINT := 0;                            // Receive count
END_VAR
```

Implementation section (ST programming language)



```
// TCP client connection check
IF ClientEnable = TRUE THEN
    NBS_Client();
    IF (NBS_Client.xBusy = TRUE)  AND (NBS_Client.xActive = TRUE) THEN
        // TCP client connection
        NBS_Handle := NBS_Client.hConnection;
    ELSIF NBS_Client.xDone = TRUE THEN
        // TCP client connection cutoff
        ClientEnable := FALSE;
        NBS_Client( xEnable := ClientEnable);
    ELSIF NBS_Client.xError = TRUE THEN
        // Error occurred
        NBS_ClientError := NBS_Client.eError;
        ClientEnable := FALSE;
        NBS_Client( xEnable := ClientEnable);
        Result := TRUE;
    END_IF
END_IF

// Reception enabled
IF RevEnable = TRUE THEN
    // Receipt confirmation
    NBS_Read();
    IF NBS_Read.xReady = TRUE THEN
        RecvCount := RecvCount + 1;         // Received number update
        RecvSize := NBS_Read.szCount;       // Receive size
    ELSIF NBS_Read.xError = TRUE THEN       // Error occurred
        NBS_ReadError := NBS_Read.eError;   // Error information storage
        RevEnable := FALSE;
        NBS_Read( xEnable := RevEnable );   // Stop receiving
        Result := TRUE;
    END_IF
END_IF

IF SndEnable = TRUE THEN
    // Transmission completion process
    NBS_Write();
    IF NBS_Write.xDone = TRUE THEN          // send completely
        NBS_Write( xExecute := FALSE );
        Process := 0;
    ELSIF NBS_Write.xError = TRUE THEN      // Error occurred
        NBS_WriteError := NBS_Write.eError; // Error information storage
        NBS_Write( xExecute := FALSE );     // Stop sending
        Result := TRUE;
        Process := 0;
    END_IF
END_IF

CASE Process OF
    1:  // TCP client connection
        Result := FALSE;
        ClientEnable := TRUE;

        NBS_Client( xEnable := ClientEnable ,   // Execute
            udiTimeOut := TimeOut ,              // Time-out setting
            ipAddr := ClientAddr ,               // Partner station IP address
            uiPort := Port );                    // Partner station port number
        Process := 0;

    2:  // Reception start processing
        Result := FALSE;
        RevEnable := TRUE;
        NBS_Read( xEnable := RevEnable ,         // Start receiving
                hConnection := NBS_Handle ,      // Connection handle
                pData := ADR(RecvBuf) ,          // Receive buffer
                szSize := SIZEOF(RecvBuf));      // Receive size
        Process := 0;

    3:  // Transmission process
        Result := FALSE;
        NBS_Write( xExecute := TRUE ,            // Start sending
                hConnection := NBS_Handle ,      // Connection handle
                pData := ADR(SendData) ,         // Send buffer
                szSize := SIZEOF(SendData));     // Send size
        SndEnable := TRUE;
        Process := 0;                            // Transmission completion wait processing
END_CASE
```

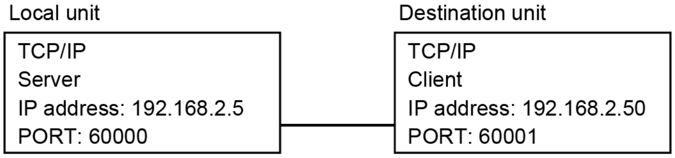
Implementation section (LD programming language)



■ TCP SERVER processing example

The following is a processing example of data transmission / reception via TCP when the local unit is TCP SERVER.

This processing example assumes the following operating environment.



Local unit

TCP/IP
Server
IP address: 192.168.2.5
PORT: 60000

Destination unit

TCP/IP
Client
IP address: 192.168.2.50
PORT: 60001

Processing for data transmission / reception

The processing for data transmission / reception is as follows:

- ● TCP server open processing
- ● TCP connection processing
- ● Reception start processing
- ● Transmission processing

Explanation of variables

Process

When the value is rewritten, the following processing is executed. After the execution is completed, the variable is set to 0 (invalid value).

1 = TCP server open processing

2 ＝ TCP connection processing

3 = Reception start processing

4 = Transmission processing

MyAddr

The IP address of the local unit is set.

MyPort

The port number of the local unit is set.

SendData

Data to be sent is set.

RecvBuf

Received data is stored.

RecvCount

The number of receptions is stored.

RecvSize

The size of received data is stored.

sClientAddr

The IP address of the connected data destination is stored.

Result

The result of processing execution is stored. (TRUE: Error occurrence, FALSE: Normal termination)

If the result of processing execution is abnormal, check the error code of each processing.

- ● NBS_ServError: Result of TCP server open processing
- ● NBS_ConErrorr: Result of TCP connection processing

- NBS_ReadError: Result of reception start processing
- NBS_WriteError: Result of transmission processing

Operation example

The TCP server is opened and connected to the TCP client.

- The value of "Process" is changed from 1 to 2.

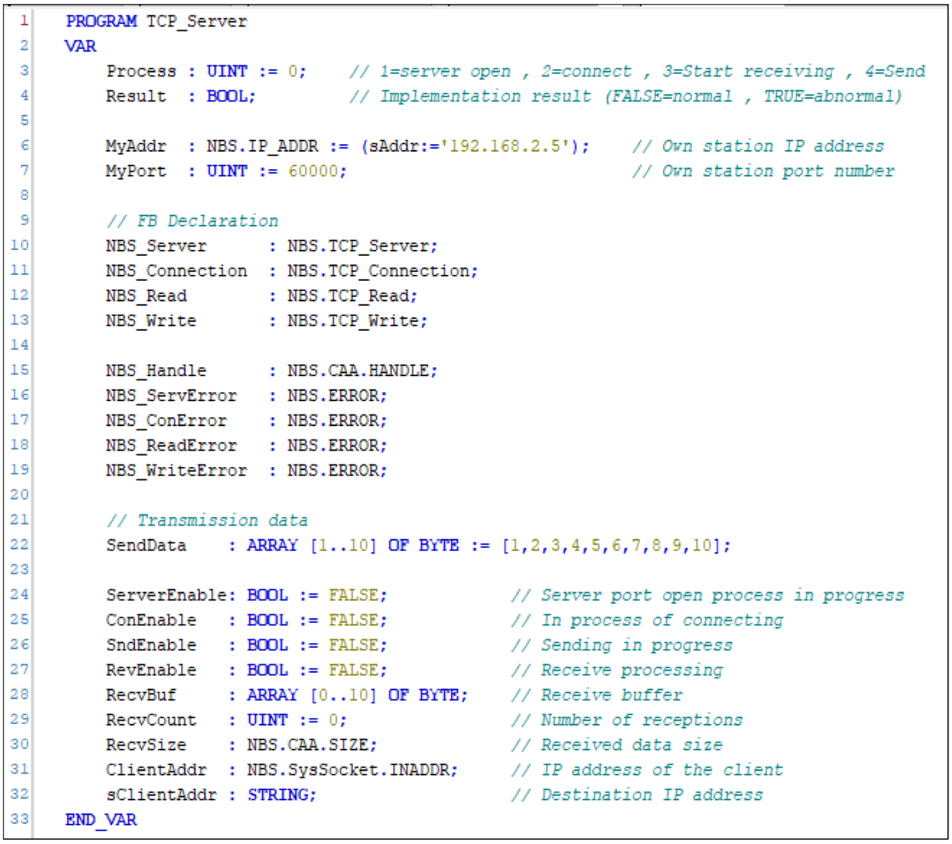
The local unit is ready to receive data. In this state, the local unit can receive data from the destination unit.

- The value of "Process" is changed to 3.

The local unit sends data to the destination unit. 10-byte data is sent to the destination unit.

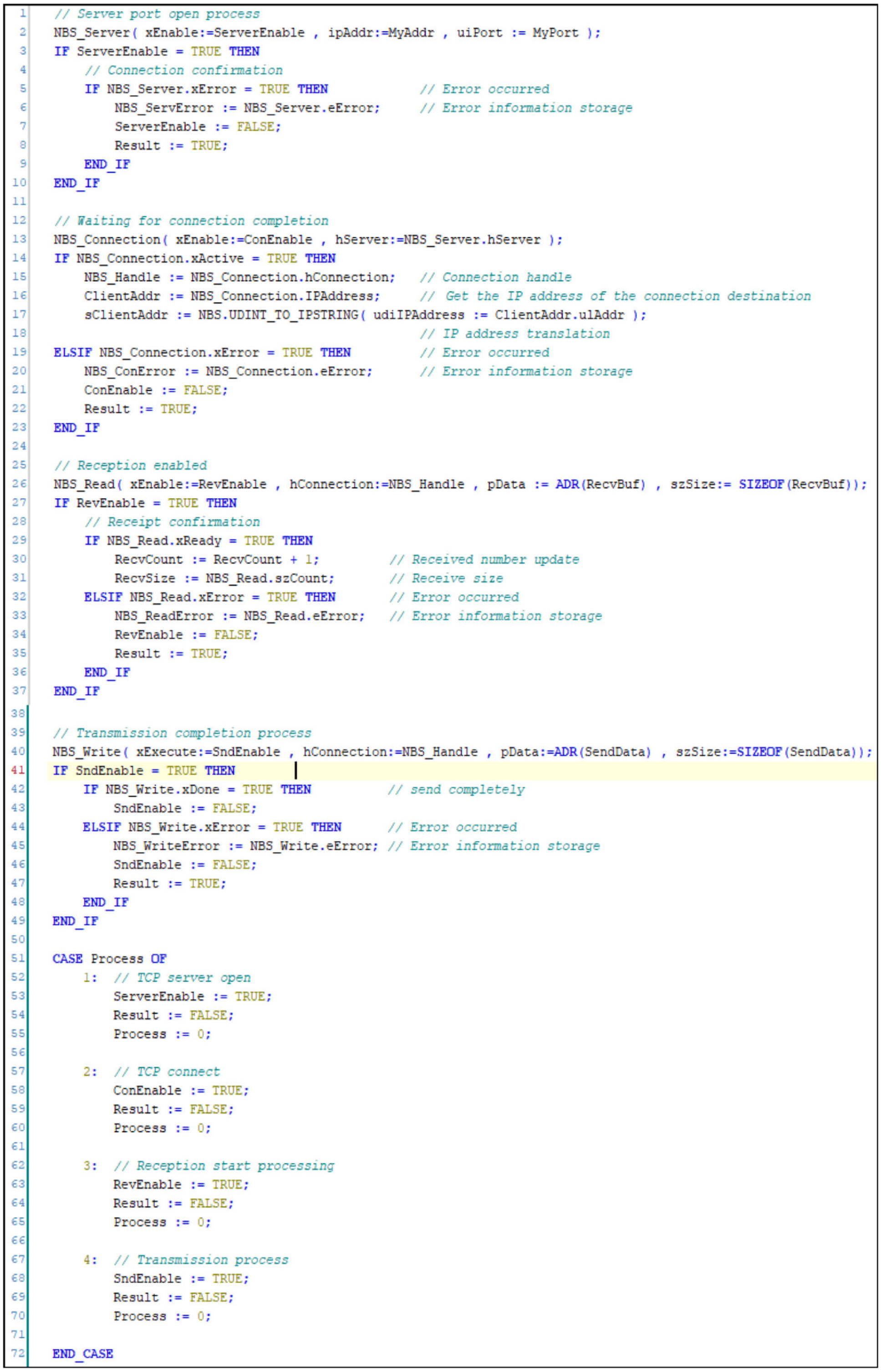
- The value of "Process" is changed to 4.

Declaration section (common to ST and LD programming languages)



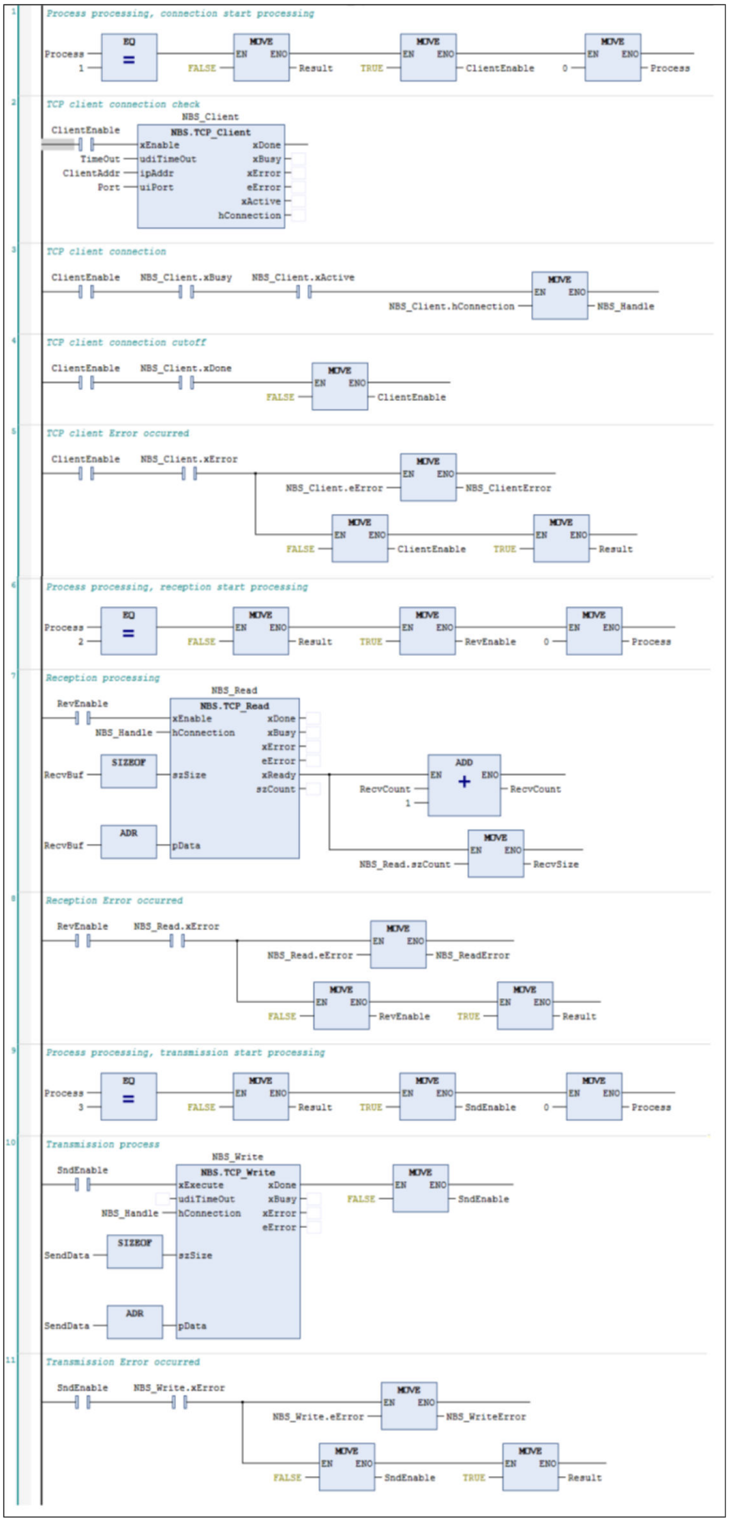
```
PROGRAM TCP_Server
VAR
    Process : UINT := 0;    // 1=server open , 2=connect , 3=Start receiving , 4=Send
    Result  : BOOL;         // Implementation result (FALSE=normal , TRUE=abnormal)

    MyAddr  : NBS.IP_ADDR := (sAddr:='192.168.2.5');    // Own station IP address
    MyPort  : UINT := 60000;                            // Own station port number

    // FB Declaration
    NBS_Server      : NBS.TCP_Server;
    NBS_Connection  : NBS.TCP_Connection;
    NBS_Read        : NBS.TCP_Read;
    NBS_Write       : NBS.TCP_Write;

    NBS_Handle      : NBS.CAA.HANDLE;
    NBS_ServError   : NBS.ERROR;
    NBS_ConError    : NBS.ERROR;
    NBS_ReadError   : NBS.ERROR;
    NBS_WriteError  : NBS.ERROR;

    // Transmission data
    SendData    : ARRAY [1..10] OF BYTE := [1,2,3,4,5,6,7,8,9,10];

    ServerEnable: BOOL := FALSE;            // Server port open process in progress
    ConEnable   : BOOL := FALSE;            // In process of connecting
    SndEnable   : BOOL := FALSE;            // Sending in progress
    RevEnable   : BOOL := FALSE;            // Receive processing
    RecvBuf     : ARRAY [0..10] OF BYTE;    // Receive buffer
    RecvCount   : UINT := 0;                // Number of receptions
    RecvSize    : NBS.CAA.SIZE;             // Received data size
    ClientAddr  : NBS.SysSocket.INADDR;     // IP address of the client
    sClientAddr : STRING;                   // Destination IP address
END_VAR
```

Implementation section (ST programming language)



```
// Server port open process
NBS_Server( xEnable:=ServerEnable , ipAddr:=MyAddr , uiPort := MyPort );
IF ServerEnable = TRUE THEN
    // Connection confirmation
    IF NBS_Server.xError = TRUE THEN            // Error occurred
        NBS_ServError := NBS_Server.eError;     // Error information storage
        ServerEnable := FALSE;
        Result := TRUE;
    END_IF
END_IF

// Waiting for connection completion
NBS_Connection( xEnable:=ConEnable , hServer:=NBS_Server.hServer );
IF NBS_Connection.xActive = TRUE THEN
    NBS_Handle := NBS_Connection.hConnection;   // Connection handle
    ClientAddr := NBS_Connection.IPAddress;     // Get the IP address of the connection destination
    sClientAddr := NBS.UDINT_TO_IPSTRING( udiIPAddress := ClientAddr.ulAddr );
                                                // IP address translation
ELSIF NBS_Connection.xError = TRUE THEN         // Error occurred
    NBS_ConError := NBS_Connection.eError;      // Error information storage
    ConEnable := FALSE;
    Result := TRUE;
END_IF

// Reception enabled
NBS_Read( xEnable:=RevEnable , hConnection:=NBS_Handle , pData := ADR(RecvBuf) , szSize:= SIZEOF(RecvBuf));
IF RevEnable = TRUE THEN
    // Receipt confirmation
    IF NBS_Read.xReady = TRUE THEN
        RecvCount := RecvCount + 1;         // Received number update
        RecvSize := NBS_Read.szCount;       // Receive size
    ELSIF NBS_Read.xError = TRUE THEN       // Error occurred
        NBS_ReadError := NBS_Read.eError;   // Error information storage
        RevEnable := FALSE;
        Result := TRUE;
    END_IF
END_IF

// Transmission completion process
NBS_Write( xExecute:=SndEnable , hConnection:=NBS_Handle , pData:=ADR(SendData) , szSize:=SIZEOF(SendData));
IF SndEnable = TRUE THEN
    IF NBS_Write.xDone = TRUE THEN          // send completely
        SndEnable := FALSE;
    ELSIF NBS_Write.xError = TRUE THEN      // Error occurred
        NBS_WriteError := NBS_Write.eError; // Error information storage
        SndEnable := FALSE;
        Result := TRUE;
    END_IF
END_IF

CASE Process OF
    1:  // TCP server open
        ServerEnable := TRUE;
        Result := FALSE;
        Process := 0;

    2:  // TCP connect
        ConEnable := TRUE;
        Result := FALSE;
        Process := 0;

    3:  // Reception start processing
        RevEnable := TRUE;
        Result := FALSE;
        Process := 0;

    4:  // Transmission process
        SndEnable := TRUE;
        Result := FALSE;
        Process := 0;

END_CASE
```

Implementation section (LD programming language)



When multiple clients are connected simultaneously to the same port, multiple TCP_Connection instances are created. The hServer handle acquired by one TCP_Server is set to the multiple TCP_Connection instances.

Example: When two clients are connected simultaneously to the same port

Declaration section

```
iServer: NBS.TCP_Server;// TCP_Server instance
iConnection: ARRAY [0..1] OF NBS.TCP_Connection;    // TCP_Connection instance (two
instances)
```

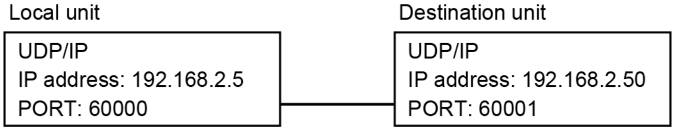
Implementation section

```
iServer( xEnable:=TRUE , ipAddr:=ipAddr , uiPort:=uiPort );    // Server opened
// Omitted (Waiting for TCP_Server completion)
iConnection[0]( xEnable := TRUE , hServer := iServer.hServer );    // For 1st client
iConnection[1]( xEnable := TRUE , hServer := iServer.hServer );    // For 2nd client
```

■ UDP processing example

An example of processing for data transmission / reception via UDP is as follows:

This processing example assumes the following operating environment.



Local unit

UDP/IP
IP address: 192.168.2.5
PORT: 60000

Destination unit

UDP/IP
IP address: 192.168.2.50
PORT: 60001

Processing for data transmission / reception

The processing for data transmission / reception is as follows:

- Port open processing
- Reception start processing
- Transmission processing

Explanation of variables

Process

When the value is rewritten, the following processing is executed. After the execution is completed, the variable is set to 0 (invalid value).

1 = Port open processing

2 = Reception start processing

3 = Transmission processing

MyipAddr

The IP address of the local unit is set.

MyPort

The port number of the local unit is set.

SendAddr

The IP address of the destination unit is set.

SendPort

The port number of the destination unit is set.

SendData

Data to be sent is set.

RecvBuf

Received data is stored.

RecvCount

The number of receptions is stored.

RecvPort

The port that received data is stored.

RecvSize

The size of received data is stored.

RecvIpAddr

The IP address of the received data destination is stored.

Result

The result of processing execution is stored. (TRUE: Error occurrence, FALSE: Normal termination)

If the result of processing execution is abnormal, check the error code of each processing.

- NBS_PeerError: Result of port open processing
- NBS_RecError: Result of reception start processing
- NBS_ReadError: Result of transmission processing

Operation example

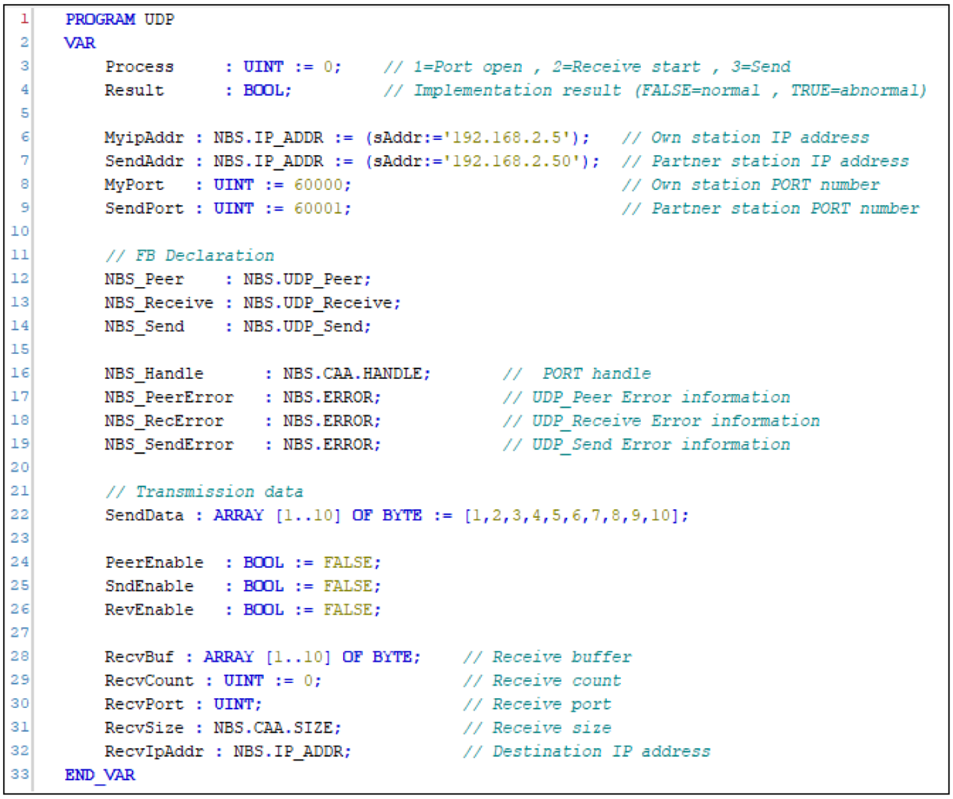
The port is opened and the local unit is ready to receive data. In this state, the local unit can receive data from the destination unit.

- The value of "Process" is changed from 1 to 2.

The local unit sends data to the destination unit. 10-byte data is sent to the destination unit.

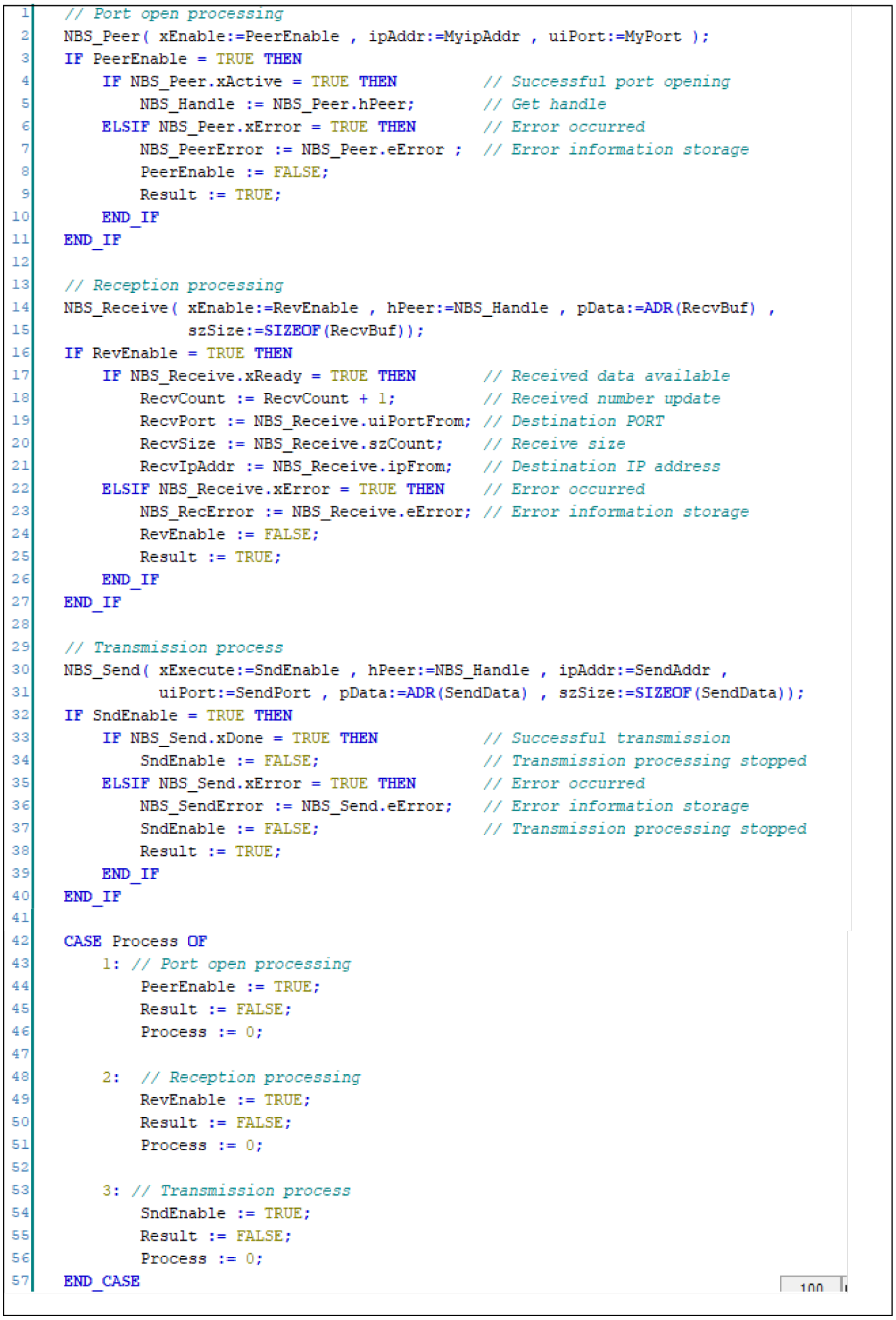
- The value of "Process" is changed to 3.

Declaration section (common to ST and LD programming languages)



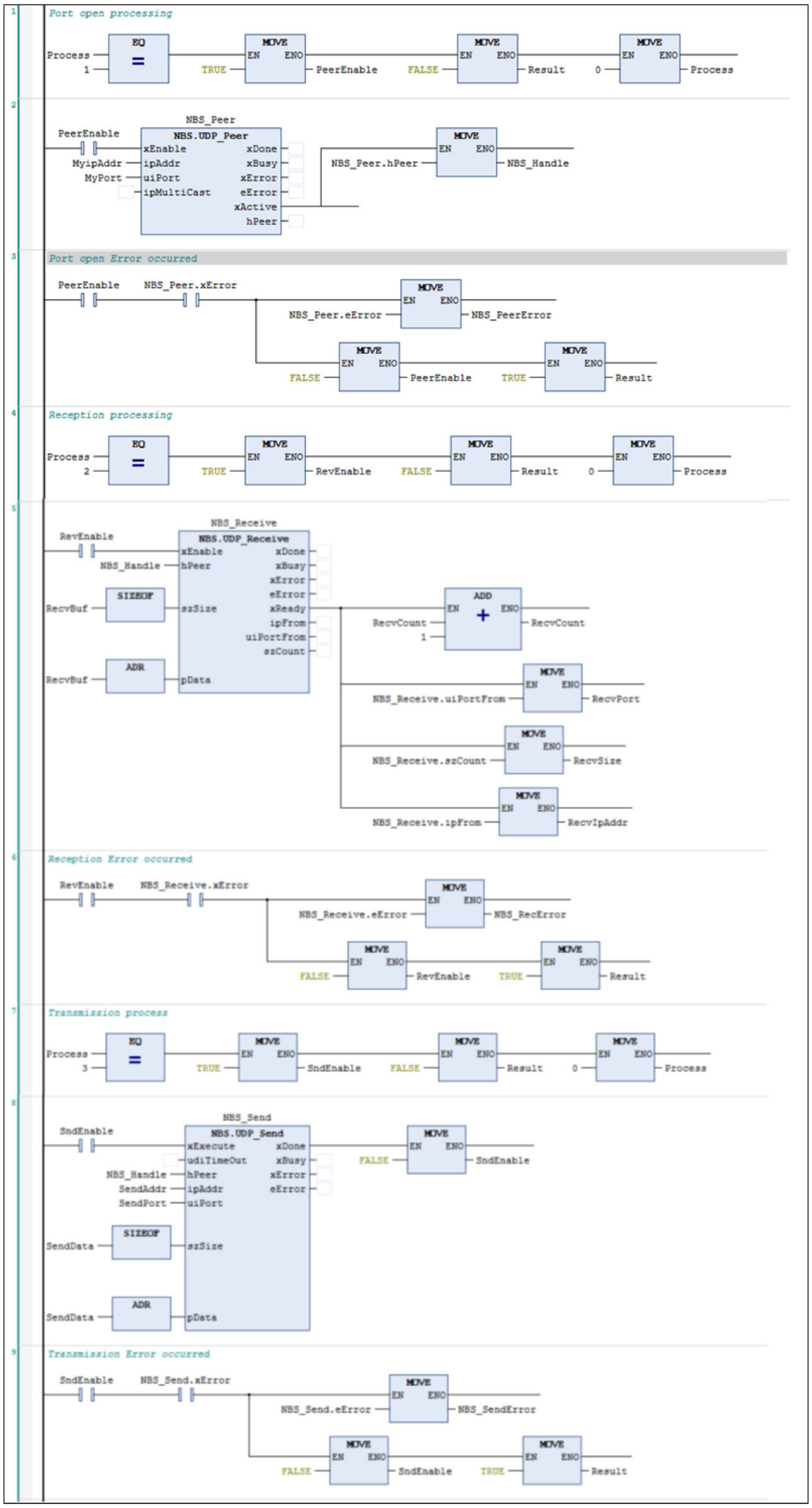
```
PROGRAM UDP
VAR
    Process     : UINT := 0;    // 1=Port open , 2=Receive start , 3=Send
    Result      : BOOL;         // Implementation result (FALSE=normal , TRUE=abnormal)

    MyipAddr : NBS.IP_ADDR := (sAddr:='192.168.2.5');   // Own station IP address
    SendAddr : NBS.IP_ADDR := (sAddr:='192.168.2.50');  // Partner station IP address
    MyPort   : UINT := 60000;                           // Own station PORT number
    SendPort : UINT := 60001;                           // Partner station PORT number

    // FB Declaration
    NBS_Peer    : NBS.UDP_Peer;
    NBS_Receive : NBS.UDP_Receive;
    NBS_Send    : NBS.UDP_Send;

    NBS_Handle      : NBS.CAA.HANDLE;       //  PORT handle
    NBS_PeerError   : NBS.ERROR;            // UDP_Peer Error information
    NBS_RecError    : NBS.ERROR;            // UDP_Receive Error information
    NBS_SendError   : NBS.ERROR;            // UDP_Send Error information

    // Transmission data
    SendData : ARRAY [1..10] OF BYTE := [1,2,3,4,5,6,7,8,9,10];

    PeerEnable  : BOOL := FALSE;
    SndEnable   : BOOL := FALSE;
    RevEnable   : BOOL := FALSE;

    RecvBuf : ARRAY [1..10] OF BYTE;    // Receive buffer
    RecvCount : UINT := 0;              // Receive count
    RecvPort : UINT;                    // Receive port
    RecvSize : NBS.CAA.SIZE;            // Receive size
    RecvIpAddr : NBS.IP_ADDR;           // Destination IP address
END_VAR
```

Implementation section (ST programming language)



```
// Port open processing
NBS_Peer( xEnable:=PeerEnable , ipAddr:=MyipAddr , uiPort:=MyPort );
IF PeerEnable = TRUE THEN
    IF NBS_Peer.xActive = TRUE THEN         // Successful port opening
        NBS_Handle := NBS_Peer.hPeer;       // Get handle
    ELSIF NBS_Peer.xError = TRUE THEN       // Error occurred
        NBS_PeerError := NBS_Peer.eError ;  // Error information storage
        PeerEnable := FALSE;
        Result := TRUE;
    END_IF
END_IF

// Reception processing
NBS_Receive( xEnable:=RevEnable , hPeer:=NBS_Handle , pData:=ADR(RecvBuf) ,
             szSize:=SIZEOF(RecvBuf));
IF RevEnable = TRUE THEN
    IF NBS_Receive.xReady = TRUE THEN       // Received data available
        RecvCount := RecvCount + 1;         // Received number update
        RecvPort := NBS_Receive.uiPortFrom; // Destination PORT
        RecvSize := NBS_Receive.szCount;    // Receive size
        RecvIpAddr := NBS_Receive.ipFrom;   // Destination IP address
    ELSIF NBS_Receive.xError = TRUE THEN    // Error occurred
        NBS_RecError := NBS_Receive.eError; // Error information storage
        RevEnable := FALSE;
        Result := TRUE;
    END_IF
END_IF

// Transmission process
NBS_Send( xExecute:=SndEnable , hPeer:=NBS_Handle , ipAddr:=SendAddr ,
          uiPort:=SendPort , pData:=ADR(SendData) , szSize:=SIZEOF(SendData));
IF SndEnable = TRUE THEN
    IF NBS_Send.xDone = TRUE THEN           // Successful transmission
        SndEnable := FALSE;                 // Transmission processing stopped
    ELSIF NBS_Send.xError = TRUE THEN       // Error occurred
        NBS_SendError := NBS_Send.eError;   // Error information storage
        SndEnable := FALSE;                 // Transmission processing stopped
        Result := TRUE;
    END_IF
END_IF

CASE Process OF
    1: // Port open processing
        PeerEnable := TRUE;
        Result := FALSE;
        Process := 0;

    2:  // Reception processing
        RevEnable := TRUE;
        Result := FALSE;
        Process := 0;

    3: // Transmission process
        SndEnable := TRUE;
        Result := FALSE;
        Process := 0;
END_CASE
```

Implementation section (LD programming language)



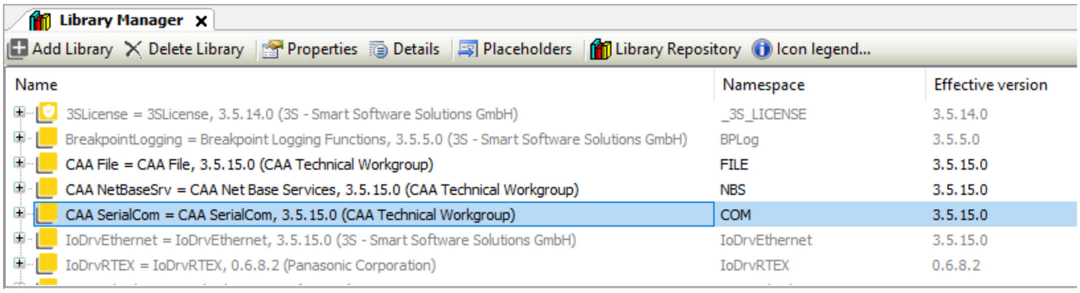
14.2.2 General-purpose Communication (Serial)

This section explains how to use the CAA SerialCom library, in the following order.

1. Library_Manager
2. COM transmission / reception processing example

■ Library_Manager

Check that the following CAA SerialCom library is registered in Library_Manager.



Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CAA File = CAA File, 3.5.15.0 (CAA Technical Workgroup)	FILE	3.5.15.0
CAA NetBaseSrv = CAA Net Base Services, 3.5.15.0 (CAA Technical Workgroup)	NBS	3.5.15.0
CAA SerialCom = CAA SerialCom, 3.5.15.0 (CAA Technical Workgroup)	COM	3.5.15.0
IoDrvEthernet = IoDrvEthernet, 3.5.15.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.15.0
IoDrvRTEX = IoDrvRTEX, 0.6.8.2 (Panasonic Corporation)	IoDrvRTEX	0.6.8.2

■ COM transmission / reception processing example

Send and receive data via SerialCom.
Specify communication settings as below.

COM number	1
Baud rate	115200 bps
Data bits	8
Parity bit	Odd
Stop bit	1

Processing for data transmission / reception

The processing for data transmission / reception is as follows:

- Serial port open processing
- Serial port close processing
- Reception processing
- Transmission processing

Explanation of variables

Process

When the value is rewritten, the following processing is executed. After the execution is completed, the variable is set to 0 (invalid value).

1 = Serial port open processing

2 = Reception processing

3 = Transmission processing

4 = Serial port close processing

SendBuf

Data to be sent is set.

SendBufLen

The length of data to be sent is set.

RecvBuf

The buffer to store received data is set.

ReadBufLen

The length of receiver buffer is set.

ReadSize

The size of received data is stored.

Result

The result of processing execution is stored. (TRUE: Error occurrence, FALSE: Normal termination)

If the result of processing execution is abnormal, check the following error code.

- ComErr: COM processing result

Operation example

Serial port is opened.

- The value of "Process" is changed to 1.

Received data is read.

- The value of "Process" is changed to 2.

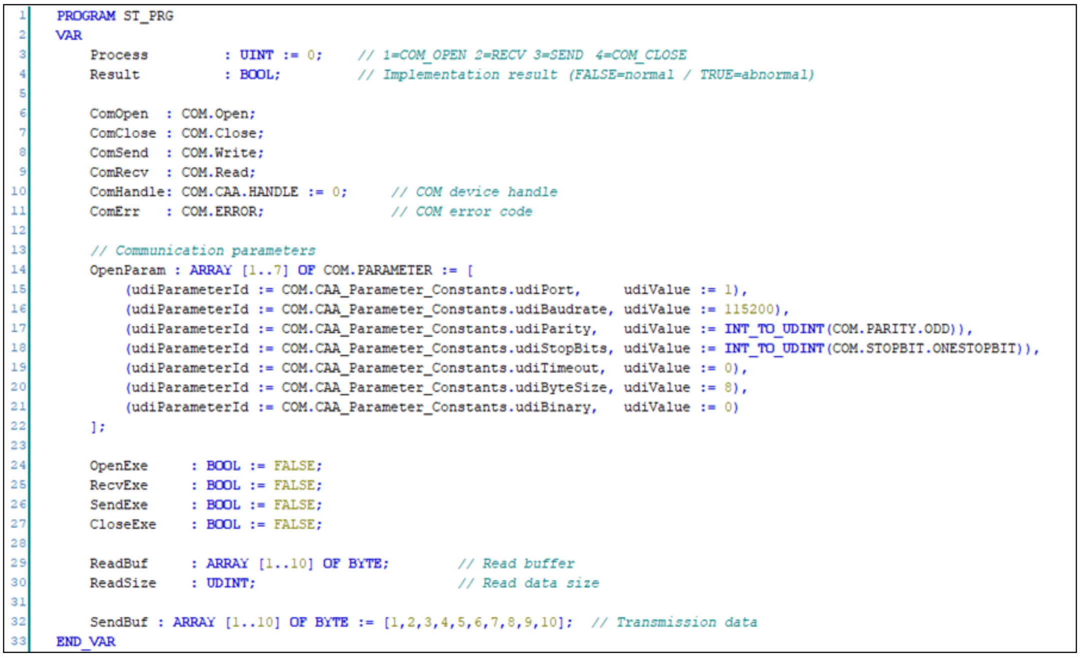
10-byte data is sent.

- The value of "Process" is changed to 3.

Serial port is closed.

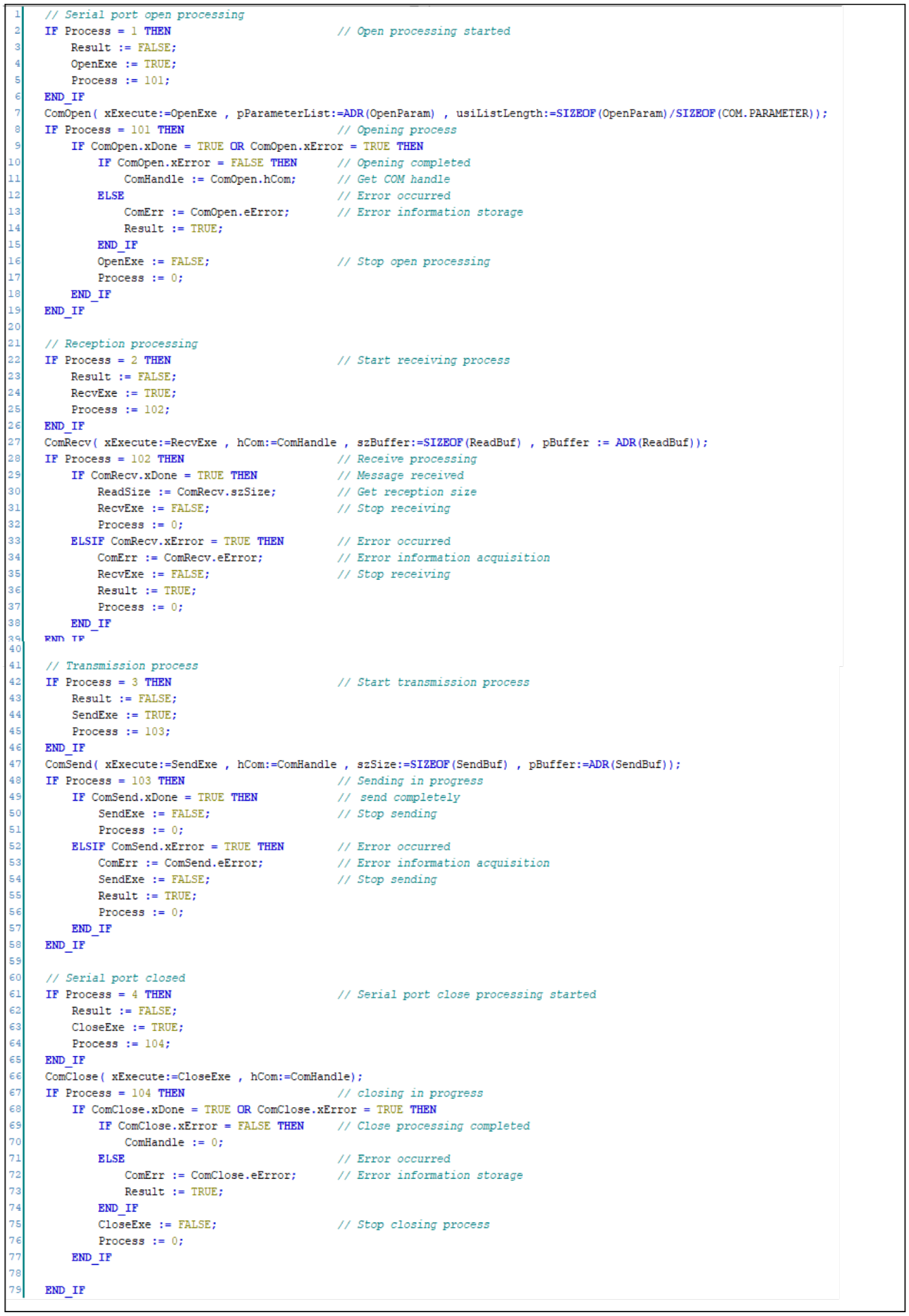
- The value of "Process" is changed to 4.

Declaration section (common to ST and LD programming languages)



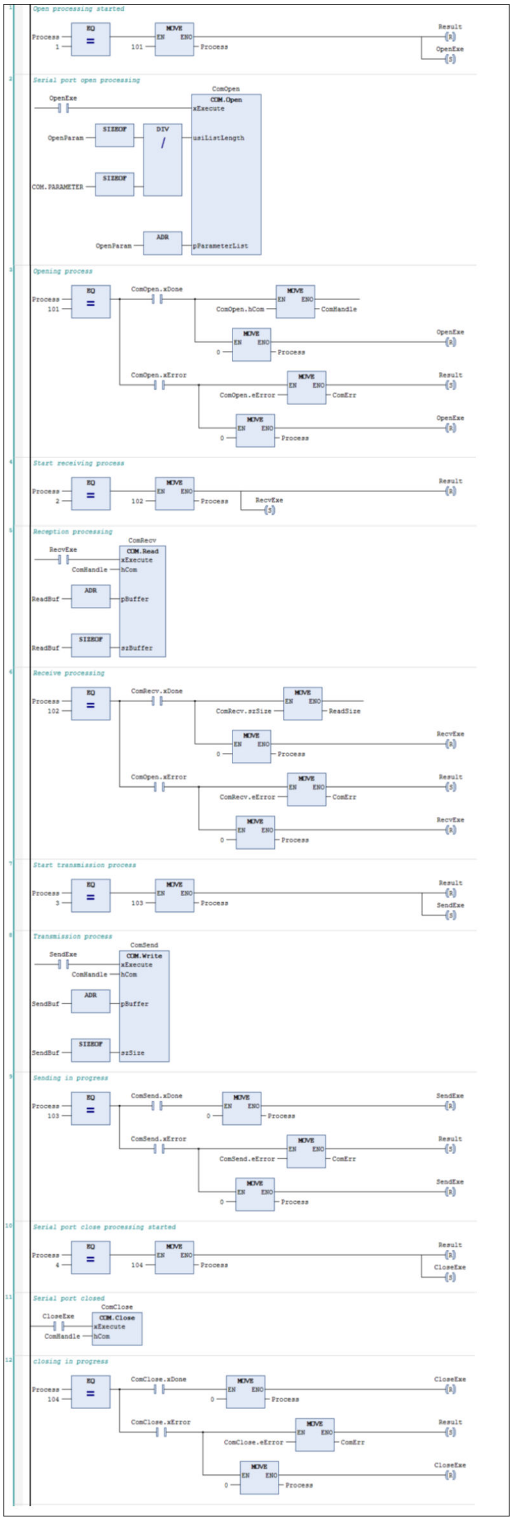
```
PROGRAM ST_PRG
VAR
    Process         : UINT := 0;    // 1=COM_OPEN 2=RECV 3=SEND  4=COM_CLOSE
    Result          : BOOL;         // Implementation result (FALSE=normal / TRUE=abnormal)

    ComOpen   : COM.Open;
    ComClose  : COM.Close;
    ComSend   : COM.Write;
    ComRecv   : COM.Read;
    ComHandle: COM.CAA.HANDLE := 0;     // COM device handle
    ComErr    : COM.ERROR;              // COM error code

    // Communication parameters
    OpenParam : ARRAY [1..7] OF COM.PARAMETER := [
        (udiParameterId := COM.CAA_Parameter_Constants.udiPort,     udiValue := 1),
        (udiParameterId := COM.CAA_Parameter_Constants.udiBaudrate, udiValue := 115200),
        (udiParameterId := COM.CAA_Parameter_Constants.udiParity,   udiValue := INT_TO_UDINT(COM.PARITY.ODD)),
        (udiParameterId := COM.CAA_Parameter_Constants.udiStopBits, udiValue := INT_TO_UDINT(COM.STOPBIT.ONESTOPBIT)),
        (udiParameterId := COM.CAA_Parameter_Constants.udiTimeout,  udiValue := 0),
        (udiParameterId := COM.CAA_Parameter_Constants.udiByteSize, udiValue := 8),
        (udiParameterId := COM.CAA_Parameter_Constants.udiBinary,   udiValue := 0)
    ];

    OpenExe     : BOOL := FALSE;
    RecvExe     : BOOL := FALSE;
    SendExe     : BOOL := FALSE;
    CloseExe    : BOOL := FALSE;

    ReadBuf     : ARRAY [1..10] OF BYTE;        // Read buffer
    ReadSize    : UDINT;                        // Read data size

    SendBuf : ARRAY [1..10] OF BYTE := [1,2,3,4,5,6,7,8,9,10];  // Transmission data
END_VAR
```

Implementation section (ST programming language)



```
// Serial port open processing
IF Process = 1 THEN                         // Open processing started
    Result := FALSE;
    OpenExe := TRUE;
    Process := 101;
END_IF
ComOpen( xExecute:=OpenExe , pParameterList:=ADR(OpenParam) , usiListLength:=SIZEOF(OpenParam)/SIZEOF(COM.PARAMETER));
IF Process = 101 THEN                       // Opening process
    IF ComOpen.xDone = TRUE OR ComOpen.xError = TRUE THEN
        IF ComOpen.xError = FALSE THEN      // Opening completed
            ComHandle := ComOpen.hCom;      // Get COM handle
        ELSE                                // Error occurred
            ComErr := ComOpen.eError;       // Error information storage
            Result := TRUE;
        END_IF
        OpenExe := FALSE;                   // Stop open processing
        Process := 0;
    END_IF
END_IF

// Reception processing
IF Process = 2 THEN                         // Start receiving process
    Result := FALSE;
    RecvExe := TRUE;
    Process := 102;
END_IF
ComRecv( xExecute:=RecvExe , hCom:=ComHandle , szBuffer:=SIZEOF(ReadBuf) , pBuffer := ADR(ReadBuf));
IF Process = 102 THEN                       // Receive processing
    IF ComRecv.xDone = TRUE THEN            // Message received
        ReadSize := ComRecv.szSize;         // Get reception size
        RecvExe := FALSE;                   // Stop receiving
        Process := 0;
    ELSIF ComRecv.xError = TRUE THEN        // Error occurred
        ComErr := ComRecv.eError;           // Error information acquisition
        RecvExe := FALSE;                   // Stop receiving
        Result := TRUE;
        Process := 0;
    END_IF
END_IF

// Transmission process
IF Process = 3 THEN                         // Start transmission process
    Result := FALSE;
    SendExe := TRUE;
    Process := 103;
END_IF
ComSend( xExecute:=SendExe , hCom:=ComHandle , szSize:=SIZEOF(SendBuf) , pBuffer:=ADR(SendBuf));
IF Process = 103 THEN                       // Sending in progress
    IF ComSend.xDone = TRUE THEN            //  send completely
        SendExe := FALSE;                   // Stop sending
        Process := 0;
    ELSIF ComSend.xError = TRUE THEN        // Error occurred
        ComErr := ComSend.eError;           // Error information acquisition
        SendExe := FALSE;                   // Stop sending
        Result := TRUE;
        Process := 0;
    END_IF
END_IF

// Serial port closed
IF Process = 4 THEN                         // Serial port close processing started
    Result := FALSE;
    CloseExe := TRUE;
    Process := 104;
END_IF
ComClose( xExecute:=CloseExe , hCom:=ComHandle);
IF Process = 104 THEN                       // closing in progress
    IF ComClose.xDone = TRUE OR ComClose.xError = TRUE THEN
        IF ComClose.xError = FALSE THEN     // Close processing completed
            ComHandle := 0;
        ELSE                                // Error occurred
            ComErr := ComClose.eError;      // Error information storage
            Result := TRUE;
        END_IF
        CloseExe := FALSE;                  // Stop closing process
        Process := 0;
    END_IF

END_IF
```

Implementation section (LD programming language)



14.3 MODBUS

14.3.1 What is Modbus TCP?

The GM1 controller can communicate with HMI and controllers via the Modbus protocol. Master communication or slave communication can be performed by adding a master or slave device for Modbus to the project file.

14.3.2 Modbus-TCP Master Communication

The Modbus-TCP master function can be used to send commands to slave devices in the following two ways.

1) When device object settings are used

- Transmission method based on slave initialization
- Transmission method
  - Cyclic
  - Rising edge
  - Application (ModbusChannel function block)

2) When device object settings are not used

- A method by which a user program (ModbusRequest function block) generates and sends commands

Info.

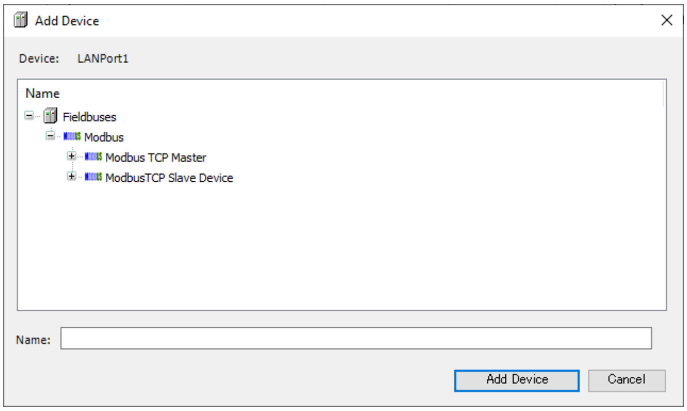
- For details on how to use ModbusRequest, refer to the *GM1 Series Reference Manual (Instruction)*.

The method in 1) is explained below.

Procedure

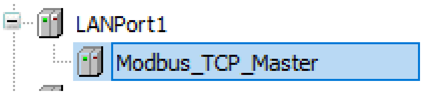
1. Right-click the "LANPort1" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.

The "Add Device" dialog box will be displayed.

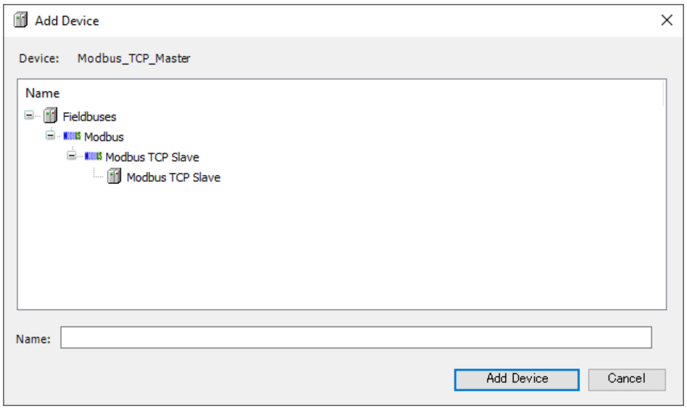


2. Select "Modbus TCP Master" under "Modbus" and click the [Add Device] button.
"Modbus_TCP_Master" will be added to the "LANPort1" object.

Example: When master communication is performed via LAN port1



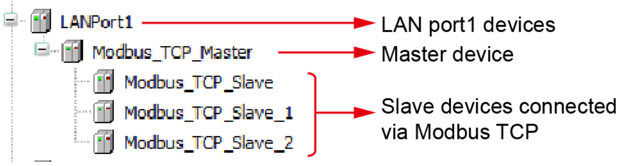
3. Right-click the "Modbus_TCP_Master" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.
The "Add Device" dialog box will be displayed.



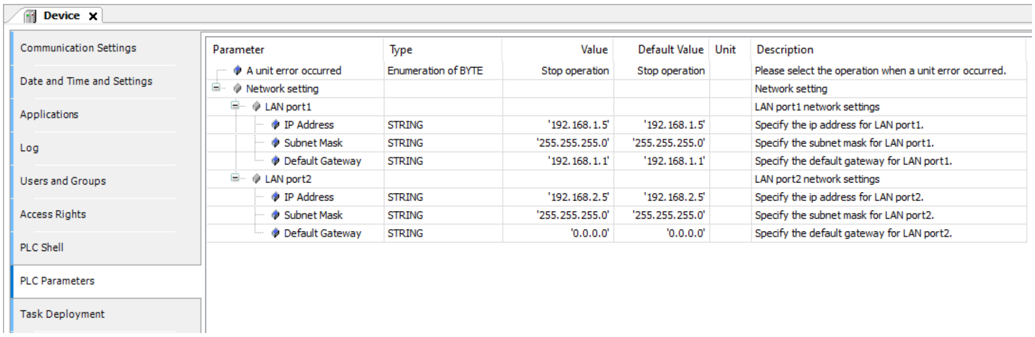
Device: Modbus_TCP_Master

4. Select "Modbus TCP Slave" under "Modbus TCP Slave" and click the [Add Device] button.
"Modbus_TCP_Slave" will be added to the "Modbus_TCP_Master" object.

Example: When master communication is performed via LAN port1

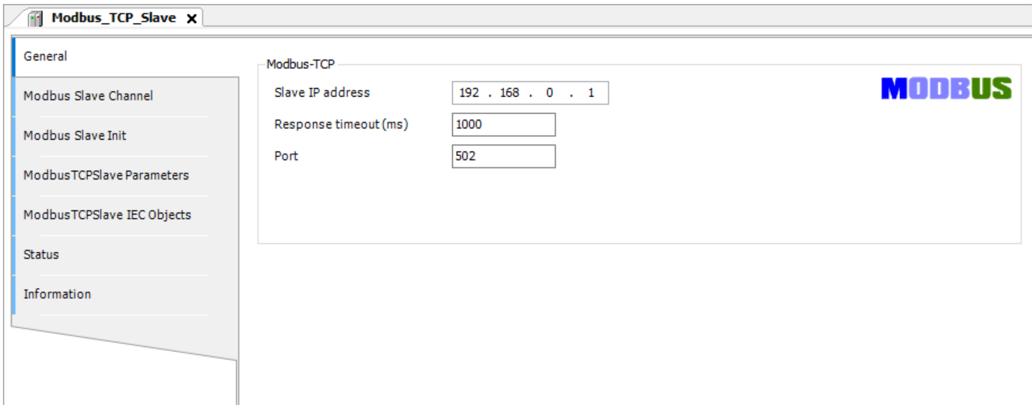


5. Open the device (AGM1CSR16T1) of the GM1 controller, select the "'PLC' Parameters" tab, and set the IP address of LAN port1.



Parameter	Type	Value	Default Value	Unit	Description
A unit error occurred	Enumeration of BYTE	Stop operation	Stop operation		Please select the operation when a unit error occurred.
Network setting					Network setting
LAN port1					LAN port1 network settings
IP Address	STRING	'192.168.1.5'	'192.168.1.5'		Specify the ip address for LAN port1.
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'		Specify the subnet mask for LAN port1.
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'		Specify the default gateway for LAN port1.
LAN port2					LAN port2 network settings
IP Address	STRING	'192.168.2.5'	'192.168.2.5'		Specify the ip address for LAN port2.
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'		Specify the subnet mask for LAN port2.
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'		Specify the default gateway for LAN port2.

6. Double-click "Modbus_TCP_Slave" in the navigator pane.
The "Modbus_TCP_Slave" object will be displayed.



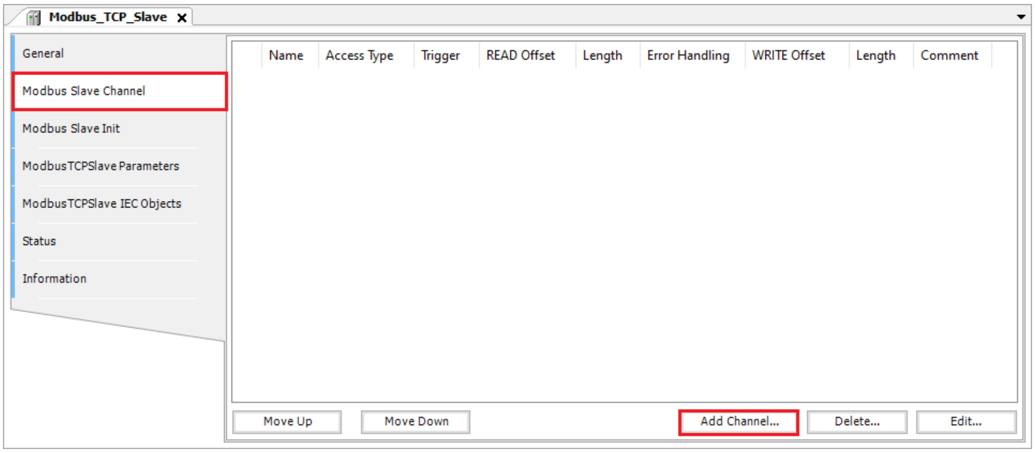
Modbus-TCP

Slave IP address 192 . 168 . 0 . 1

Response timeout (ms) 1000

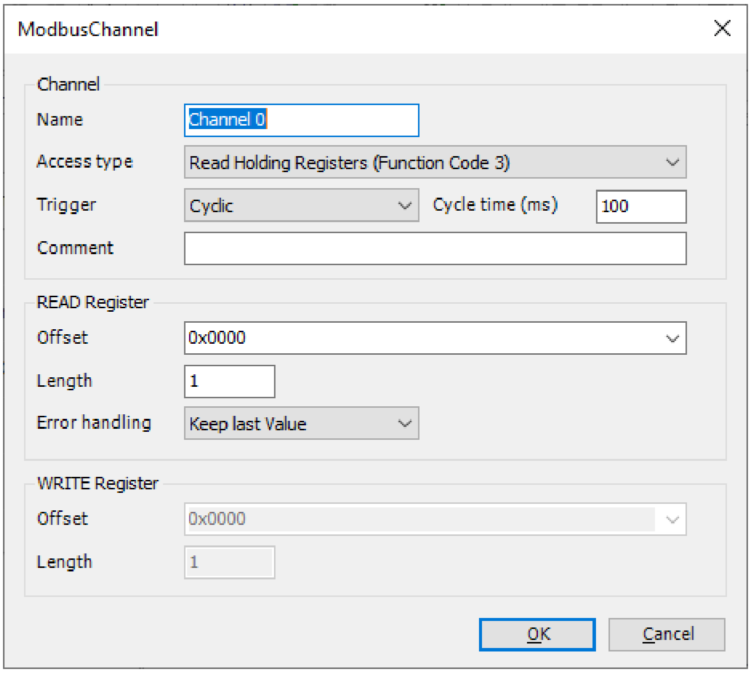
Port 502

7. Select the "Modbus Slave Channel" tab.



Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	Comment

8. Click the [Add Channel] button.
The "Modbus Channel" dialog box will be displayed.
Enter information for channels to be used. Up to 100 channels can be set.


ModbusChannel

Channel
- Name: Channel 0
- Access type: Read Holding Registers (Function Code 3)
- Trigger: Cyclic
- Cycle time (ms): 100
- Comment:

READ Register
- Offset: 0x0000
- Length: 1
- Error handling: Keep last Value

WRITE Register
- Offset: 0x0000
- Length: 1

Access type:

Select an access type (function code) and change the value of the READ or WRITE register parameter according to the selected access type.

Function code	Access type	Description
1	Read Coils	Reads from coils
2	Read Discrete Inputs	Reads from discrete inputs
3	Read Holding Registers	Reads from holding registers
4	Read Input Registers	Reads from input registers
5	Write Single Coil	Writes to single coil
6	Write Single Register	Writes to single register
15	Write Multiple Coils	Writes to multiple coils
16	Write Multiple Registers	Writes to multiple registers
23	Read / Write Multiple Registers	Reads from or writes to multiple registers

Trigger:

Select conditions for command transmission.

Access type	Description
Cyclic	Commands are sent periodically. Enter a transmission interval in the Cycle time field.
Rising edge	Commands are sent at the rising edge of a Boolean trigger variable. The trigger variable area is defined in the I/O Mapping tab.
Application	Commands are sent using the ModbusChannel function block in a user program.

READ register settings

Item	Description
Offset	Specifies the starting address from which read operation is to be started.
Length	Specifies the number of registers to be read from. The value of the parameter depends on the function code.
Error handling	Defines data that identifies communication errors. ● "Keep last value": Holds the last value that is read ● "Set to ZERO": Sets 0

WRITE register settings

Item	Description
Offset	Specifies the starting address from which write operation is to be started. For SP15, do not specify offset "65535".
Length	Specifies the number of registers to be written to. The value of the parameter depends on the function code.

This completes the device object settings for the method for sending commands based on slave initialization and the method for sending commands based on channel settings.

14.3.3 Modbus-TCP Slave Communication

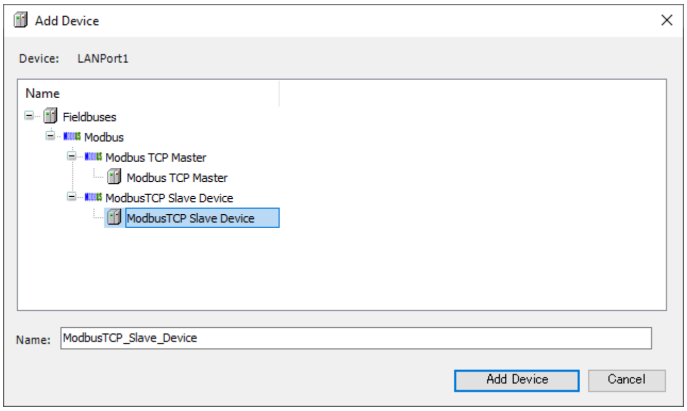
This section explains how to use the Modbus-TCP slave function.

The slave function is used by setting up the device object.

1 2 Procedure

1. Right-click the "LANPort1" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.

   The "Add Device" dialog box will be displayed.



Add Device

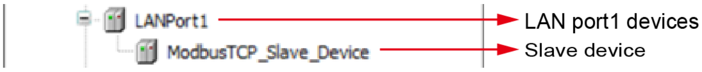
Device: LANPort1

Name
- Fieldbuses
  - Modbus
    - Modbus TCP Master
      - Modbus TCP Master
    - ModbusTCP Slave Device
      - ModbusTCP Slave Device

Name: ModbusTCP_Slave_Device

2. Select "ModbusTCP Slave Device" under "ModbusTCP Slave Device" and click the [Add Device] button.

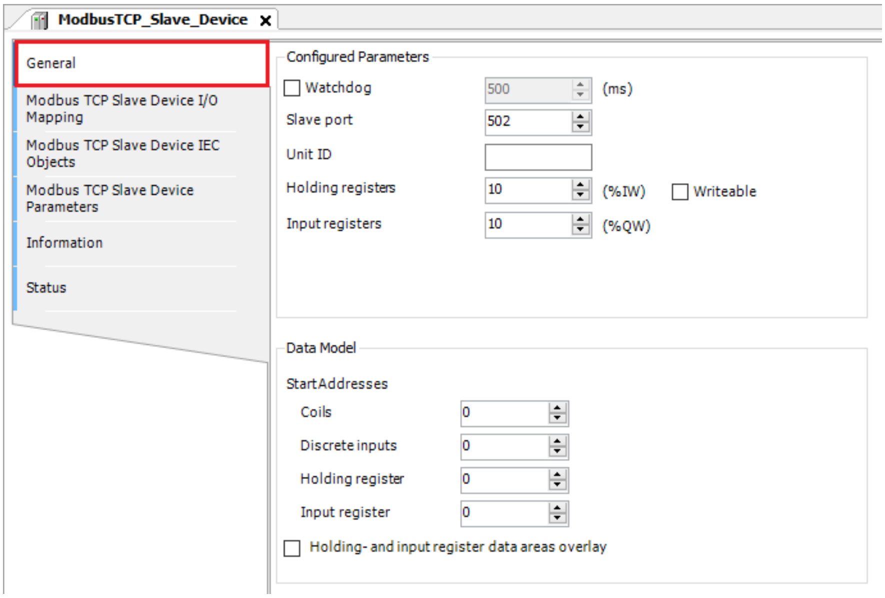
The "ModbusTCP_Slave_Device" object will be added to the "LANPort1" object.



LAN port1 devices

Slave device

3. Double-click "ModbusTCP_Slave_Device" in the navigator pane.
The "LANPort1" object will be displayed.
Select the "General" tab and set appropriate values for each item.


ModbusTCP_Slave_Device
General
Modbus TCP Slave Device I/O Mapping
Modbus TCP Slave Device IEC Objects
Modbus TCP Slave Device Parameters
Information
Status

Configured Parameters

□ Watchdog 500 (ms)

Slave port 502

Unit ID

Holding registers 10 (%IW) □ Writeable

Input registers 10 (%QW)

Data Model

StartAddresses

Coils 0

Discrete inputs 0

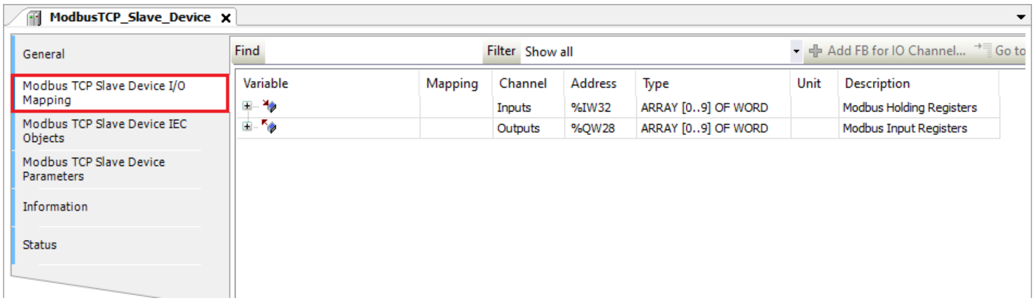
Holding register 0

Input register 0

□ Holding- and input register data areas overlay

Item	Description
Watchdog	If no valid command is received from the master during the preset time period, the holding register (%IW) will be set to zero. Settable value: 500 to 200000
Slave port	Port number used by the slave (local unit) Settable value: 1 to 65535
Unit ID	A station number is set. Settable value: 1 to 247
Holding registers (%IW)	The number of holding registers is set. Buffer size of holding register: 1 to 4096
Input registers (%QW)	The number of input registers is set. Buffer size of input register: 1 to 4096

4. Select the "Modbus TCP Slave Device I/O Mapping" tab.
You can allocate variables to holding registers and input registers.


ModbusTCP_Slave_Device
General
Modbus TCP Slave Device I/O Mapping
Modbus TCP Slave Device IEC Objects
Modbus TCP Slave Device Parameters
Information
Status
Find
Filter Show all
Add FB for IO Channel...
Go to

Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%IW32	ARRAY [0..9] OF WORD		Modbus Holding Registers
		Outputs	%QW28	ARRAY [0..9] OF WORD		Modbus Input Registers

Registers correspond to each access type (function code)

Function code	Access type	Register	
		When the check box is not selected(Note 1)	When the check box is selected(Note 1)
1	Read Coils	Holding register	Input register
2	Read Discrete Inputs	Input register	Input register
3	Read Holding Registers	Holding register	Input register
4	Read Input Registers	Input register	Input register
5	Write Single Coil	Holding register	Holding register
6	Write Single Register	Holding register	Holding register
15	Write Multiple Coils	Holding register	Holding register
16	Write Multiple Registers	Holding register	Holding register
23	Read / Write Multiple Registers	Holding register (Read/ Write)	Input register (Read) Holding register (Write)

(Note 1) The register to be used is changed according to whether the "Holding register data area overlay and input register data area overlay" check box is selected.

14.3.4 Modbus-RTU Master Communication

The Modbus-RTU master function can be used to send commands to slave devices in the following two ways.

1) When device object channel settings are used

- Transmission method based on slave initialization
- Transmission method based on channel settings
  - Cyclic
  - Rising edge
  - Application (ModbusChannel function block)

2) When device object channel settings are not used (Note 1)

- A method by which a user program (ModbusRequest function block) generates and sends commands

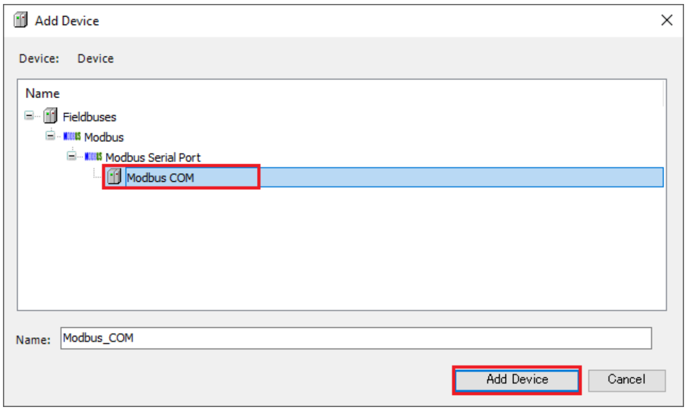
  (Note 1) Device object registration is required.

The method in 1) is explained below.

1 2 Procedure

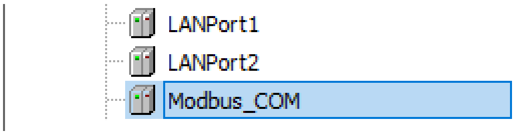
1. Right-click the "Device" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.

The "Add Device" dialog box will be displayed.



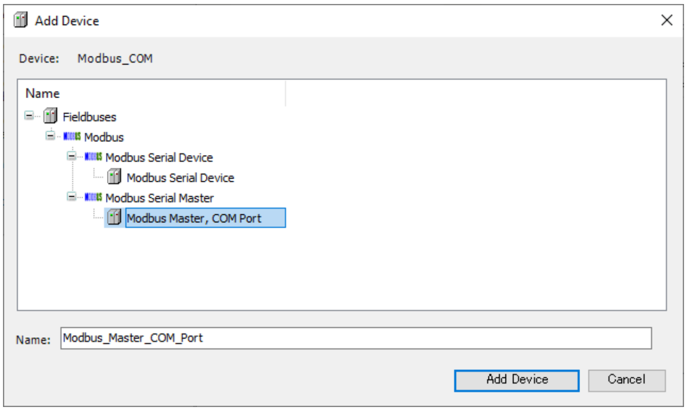
2. Select "Modbus COM" under "Modbus Serial Port" and click the [Add Device] button.

The "Modbus_COM" object will be added to the "Device" object.



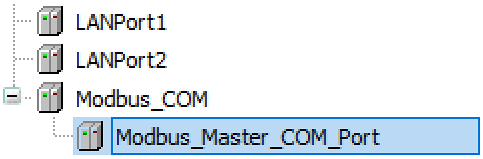
3. Right-click the "Modbus_COM" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.

The "Add Device" dialog box will be displayed.



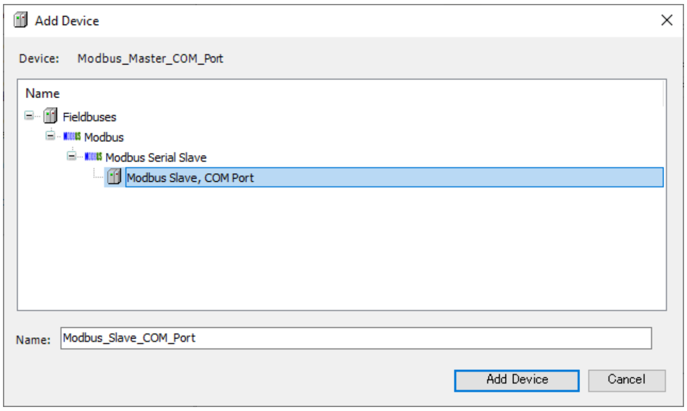
4. Select "Modbus Master, COM Port" under "Modbus Serial Master" and click the [Add Device] button.

The "Modbus_Master_COM_Port" object will be added below the "Modbus_COM" object.



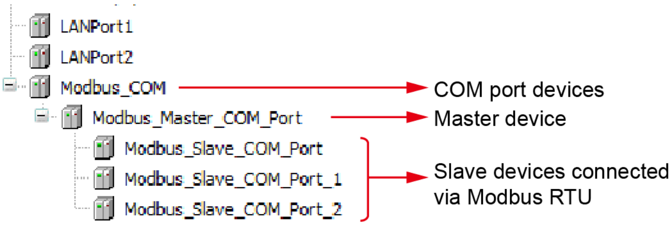
5. Right-click the "Modbus_Master_COM_Port" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.

The "Add Device" dialog box will be displayed.



6. Select "Modbus Slave, COM Port" under "Modbus Serial Slave" and click the [Add Device] button.

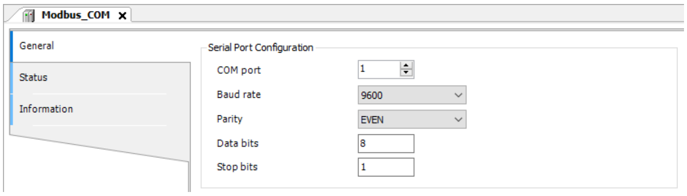
The "Modbus_Slave_COM_Port" object will be added below the "Modbus_Master_COM_Port" object.



7. Double-click "Modbus_COM" in the navigator pane.

"Modbus_COM"object will be displayed.

Select the"General"tab and set appropriate values for each item.



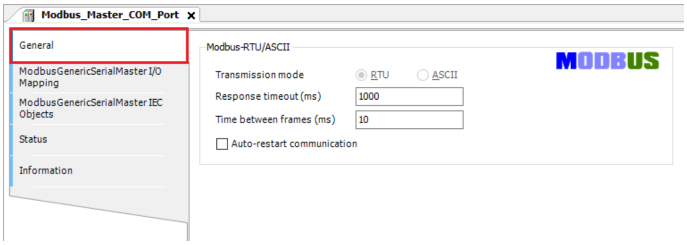
Serial Port Configuration

COM port: 1
Baud rate: 9600
Parity: EVEN
Data bits: 8
Stop bits: 1

Item	Description
COM port	Settable value: 1 to 99 The COM port of the GM1 controller is fixed at 1.

Item	Description
Baud rate	Can be selected from 9600, 19200, 38400, 57600, and 115200
Parity	Can be selected from EVEN, ODD, and NONE
Data bits	Data bit length between start bit and stop bit Settable value: 7 bits and 8 bits (Settable: 0 to 255) The default value of Modbus is 8 bits.
Stop bit	Settable value: 1 bit and 2 bits The default value of Modbus is 1 bit.

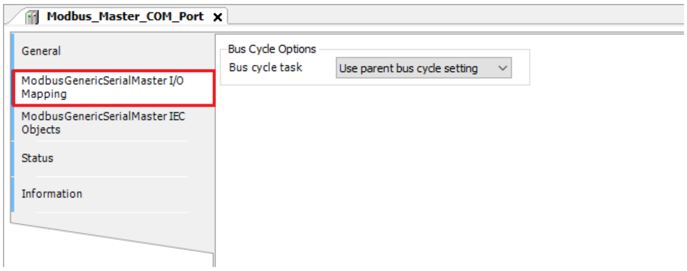
8. Double-click "Modbus Master, COM Port" in the navigator pane.
The "Modbus_Master_COM_Port"object will be displayed.
Select the "General"tab and set appropriate values for each item.


Modbus_Master_COM_Port
General
ModbusGenericSerialMaster I/O Mapping
ModbusGenericSerialMaster IEC Objects
Status
Information
Modbus-RTU/ASCII
MODBUS

Transmission mode ● RTU ○ ASCII
Response timeout (ms) 1000
Time between frames (ms) 10
□ Auto-restart communication

Item	Description
Transmission mode	RTU: Binary transmission ASCII: ASCII code transmission (Not supported)
Response timeout [ms] [0..65535]	Waiting time for response from slave * If a response timeout period is set in the slave device, the settings in the slave device will take effect.
Time between frames [ms] [0..65535]	Time period during which master transmission is paused from when the last response is received until the next command is sent.
Auto-restart communication	When the check box is selected: After a communication error occurs, the communication status is automatically checked. When the communication is restored, reconnection is performed. When the check box is not selected: After a communication error occurs, reconnection is not performed.

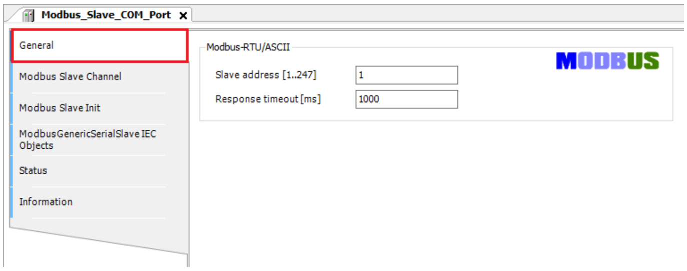
"ModbusGenericSerialMaster I/O Mapping"tab: Select a bus cycle task that performs Modbus communication.


Modbus_Master_COM_Port
General
ModbusGenericSerialMaster I/O Mapping
ModbusGenericSerialMaster IEC Objects
Status
Information
Bus Cycle Options

Bus cycle task: Use parent bus cycle setting

9. Double-click "Modbus Slave, COM Port" in the navigator pane.
The "Modbus_Slave_COM_Port"object will be displayed.

Select the "General"tab and set appropriate values for each item.

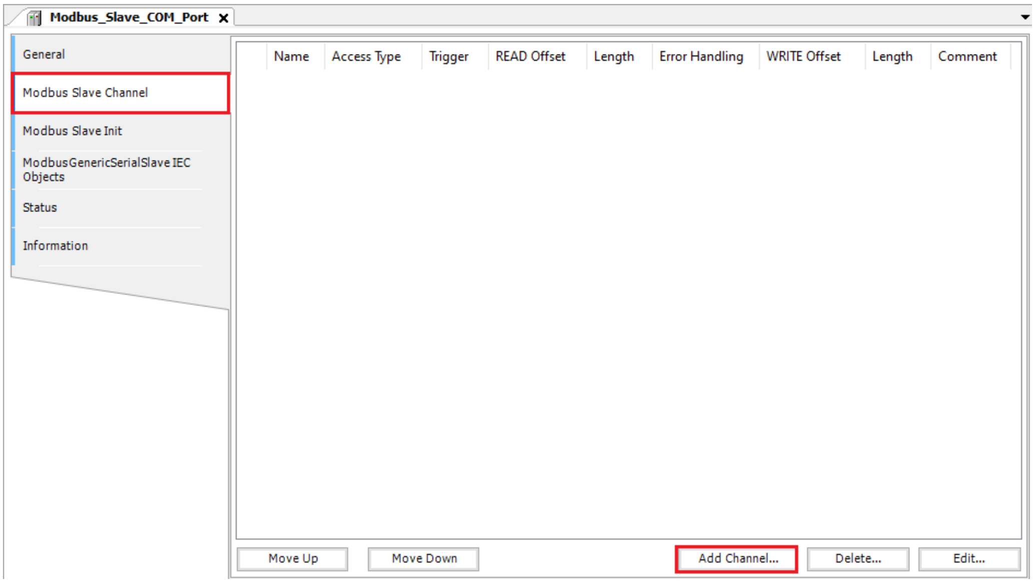


Slave address [1..247] 1

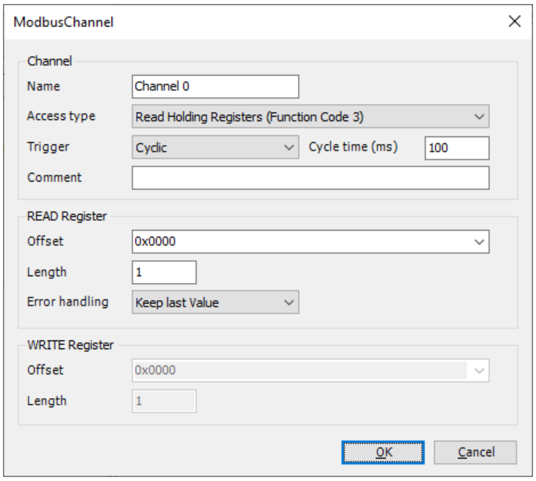
Response timeout [ms] 1000

Item	Description
Slave address [1..247]	Specifies the address (station number) of the slave device.
Response timeout (ms) [2..65535]	Waiting time for response from slave * The response timeout value for the master device is overwritten.

10. Select the "Modbus Slave Channel" tab.



11. Click the [Add Channel] button.
The "Modbus Channel"dialog box will be displayed.
Enter information for channels to be used. Up to 100 channels can be set.



ModbusChannel

Channel

Name: Channel 0

Access type: Read Holding Registers (Function Code 3)

Trigger: Cyclic   Cycle time (ms): 100

Comment:

READ Register

Offset: 0x0000

Length: 1

Error handling: Keep last Value

WRITE Register

Offset: 0x0000

Length: 1

Access type :

Select an access type (function code) and change the value of the READ or WRITE register parameter according to the selected access type.

Function code	Access type	Description
1	Read Coils	Reads from coils
2	Read Discrete Inputs	Reads from discrete inputs
3	Read Holding Registers	Reads from holding registers
4	Read Input Registers	Reads from input registers
5	Write Single Coil	Writes to single coil
6	Write Single Register	Writes to single register
15	Write Multiple Coils	Writes to multiple coils
16	Write Multiple Registers	Writes to multiple registers
23	Read/Write Multiple Registers	Reads from or writes to multiple registers

Trigger :

Select conditions for command transmission.

Access type	Description
Cyclic	Commands are sent periodically. Enter a transmission interval in the Cycle time field.
Rising edge	Commands are sent at the rising edge of a Boolean trigger variable. The trigger variable area is defined in the I/O Mapping tab.
Application	Commands are sent using the ModbusChannel function block in a user program.

READ register settings

Item	Description
Offset	Specifies the starting address from which read operation is to be started. For SP15, do not specify offset "65535".

Item	Description
Length	Specifies the number of registers to be read from. The value of the parameter depends on the function code.
Error handling	Defines data that identifies communication errors. ● "Keep last value": Holds the last value that is read ● "Set to ZERO": Sets 0

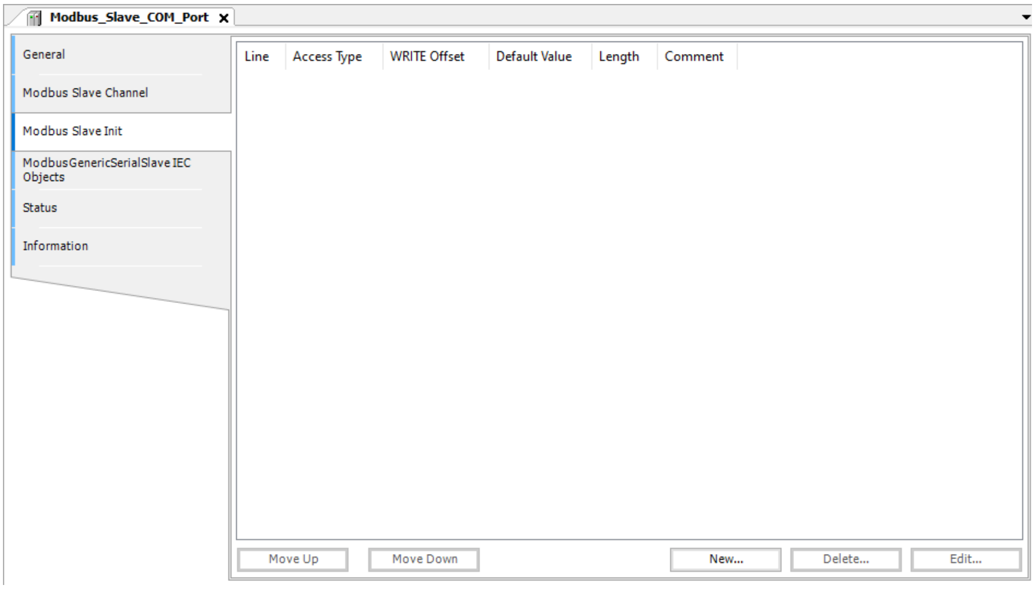
WRITE register settings

Item	Description
Offset	Specifies the starting address from which write operation is to be started.
Length	Specifies the number of registers to be written to. The value of the parameter depends on the function code.

12. Select the "Initialize Modbus Slave" tab.

Slave devices can be initialized.

Slave initialization is executed once when a slave is activated at the time of startup or RUN mode.Click the [New]button and enter information for channels to be used. Up to 20 commands can be added for each device.

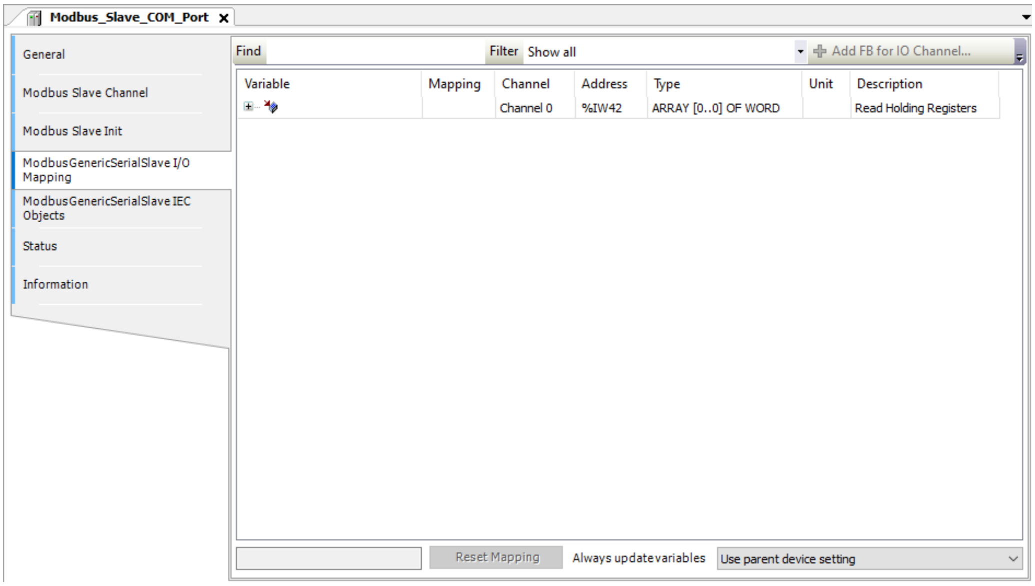


List of access types (function codes)

Code	Access type	Description
5	Write Single Coil	Writes to single coil
6	Write Single Register	Writes to single register
15	Write Multiple Coils	Writes to multiple coils
16	Write Multiple Registers	Writes to multiple registers

13. Select the "ModbusGenericSerialSlave I/O Mapping" tab.

Read areas, write areas, and trigger variable areas are defined according to the channel information created in Step 10. Allocate variables as necessary.

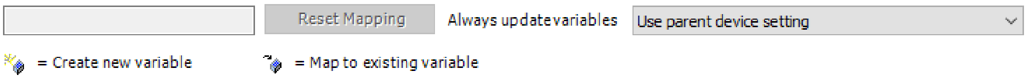


Filter Show all

Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%IW42	ARRAY [0..0] OF WORD		Read Holding Registers

Always update variables: Use parent device setting

Update settings for I/O variables



Always update variables: Use parent device setting

= Create new variable

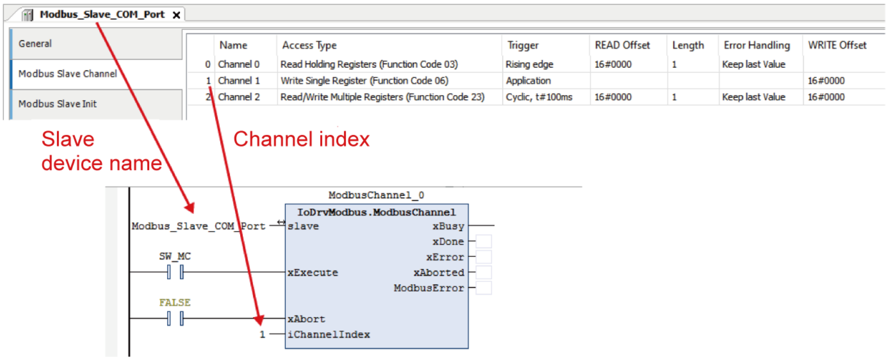
= Map to existing variable

Item	Description
Use parent device settings	Updates I/O variables according to the parent device settings
Enable 1 (Bus cycle task if not used by any tasks)	Updates I/O variables in the bus cycle if not used by any other task
Enable 2 (Always use bus cycle task)	Updates all I/O variables in each cycle of the bus cycle task

This completes the device object settings for the method for sending commands based on slave initialization and the method for sending commands based on channel settings.

The following is an example of creating an LD program that sends commands when a trigger is set as an "application" in channel settings.

The ModbusChannel function block is used for command transmission. The slave device added to the navigator pane is specified in the slave operand, and the index of the channel that has been added to the "Modbus Slave Channel" tab and that is used to send commands is specified in the iChannelIndex command.



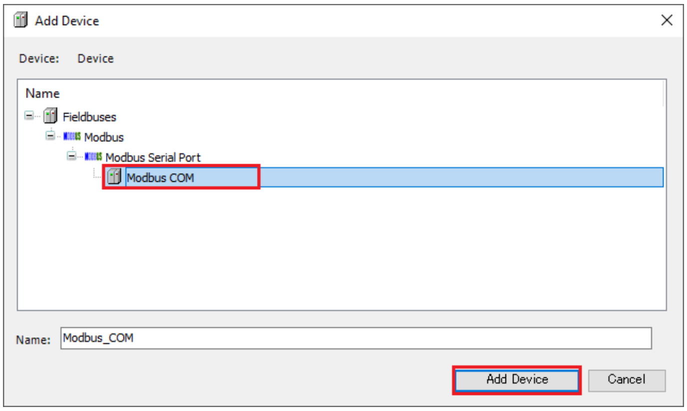
	Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset
0	Channel 0	Read Holding Registers (Function Code 03)	Rising edge	16#0000	1	Keep last Value	
1	Channel 1	Write Single Register (Function Code 06)	Application				16#0000
2	Channel 2	Read/Write Multiple Registers (Function Code 23)	Cyclic, t#100ms	16#0000	1	Keep last Value	16#0000

14.3.5 Modbus-RTU Slave Communication

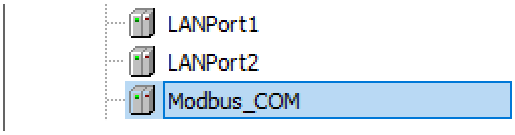
This section explains how to use the Modbus-RTU slave function.
The slave function is used by setting up the device object.

1 2 Procedure

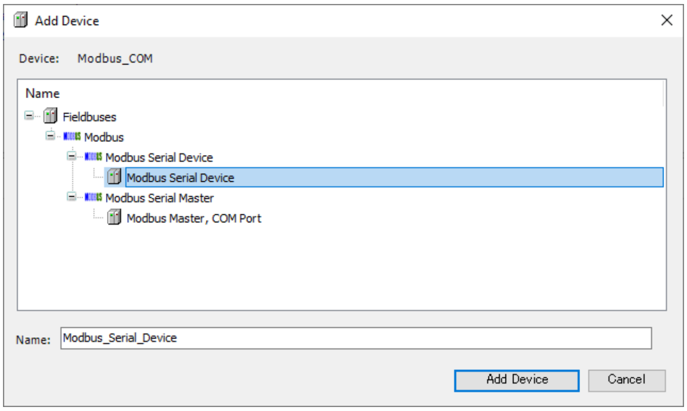
1. Right-click the "Device" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.
The "Add Device" dialog box will be displayed.



2. Select "Modbus COM" under "Modbus Serial Port" and click the [Add Device] button.
The "Modbus_COM" object will be added to the "Device" object.



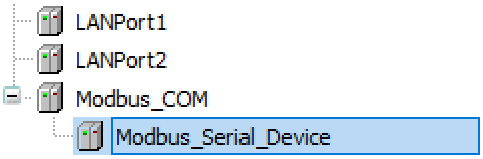
3. Right-click the "Modbus_COM" object in the navigator pane and then select Add Device from the context-sensitive menu that is displayed.
The "Add Device" dialog box will be displayed.



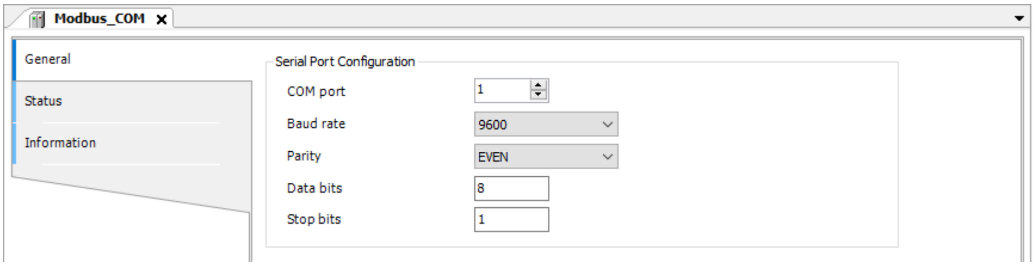
Device: Modbus_COM

Name: Modbus_Serial_Device

4. Select "Modbus Serial Device" under "Modbus Serial Device" and click the [Add Device] button.
The "Modbus_Serial_Device" object will be added below the "Modbus COM" object.



5. Double-click "Modbus_COM" in the navigator pane.
The "Modbus_COM"object will be displayed.
Select the "General"tab and set appropriate values for each item.



Serial Port Configuration

COM port: 1

Baud rate: 9600

Parity: EVEN

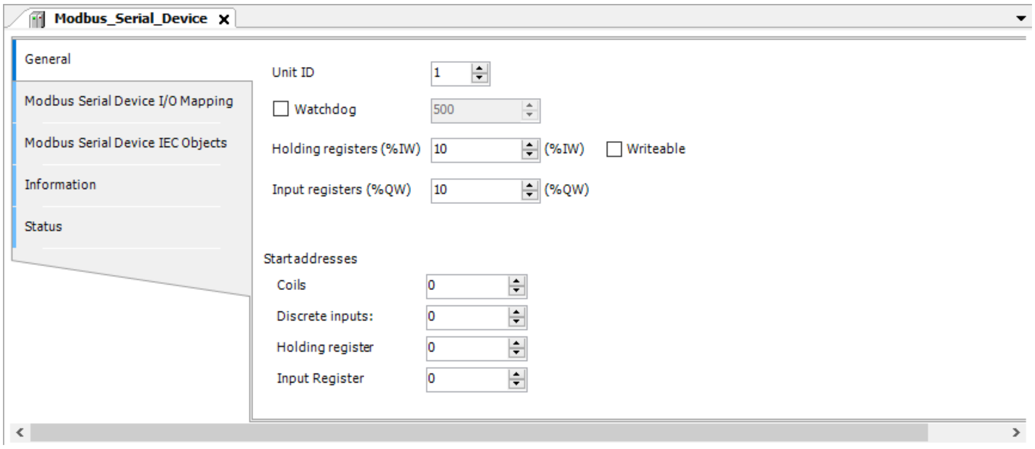
Data bits: 8

Stop bits: 1

Item	Description
COM port	Settable value: 1 to 99 The COM port of the GM1 controller is fixed at 1.
Baud rate	Can be selected from 9600, 19200, 38400, 57600, and 115200
Parity	Can be selected from EVEN, ODD, and NONE
Data bits	Data bit length between start bit and stop bit Settable value: 7 bits and 8 bits (Settable: 0 to 255) The default value of Modbus is 8 bits.
Stop bit	Settable value: 1 bit and 2 bits The default value of Modbus is 1 bit.

6. Double-click "Modbus_Serial_Device" in the navigator pane.
The "Modbus_Serial_Device"object will be displayed.

Select the "General"tab and set appropriate values for each item.



Modbus_Serial_Device

- General
- Modbus Serial Device I/O Mapping
- Modbus Serial Device IEC Objects
- Information
- Status

Unit ID: 1

□ Watchdog: 500

Holding registers (%IW): 10 (%IW) □ Writeable

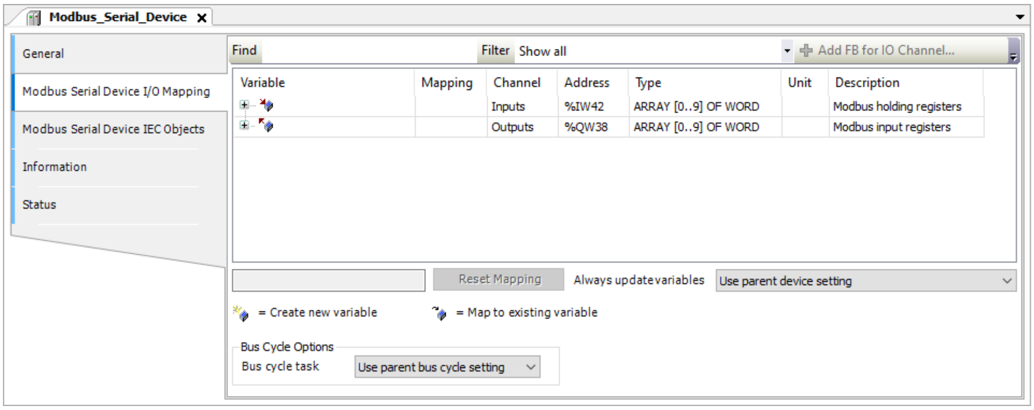
Input registers (%QW): 10 (%QW)

Startaddresses

Coils	0
Discrete inputs:	0
Holding register	0
Input Register	0

Item	Description
Watchdog	If no valid command is received from the master during the preset time period, the holding register (%IW) will be set to zero. Settable value: 500 to 200000
Unit ID	A station number is set. Settable value: 1 to 247
Holding registers (%IW)	The number of holding registers is set. Buffer size of holding register: 1 to 500
Input registers (%QW)	The number of input registers is set. Buffer size of input register: 1 to 500

7. Select the "Modbus Serial Device I/O Mapping" tab.
You can allocate variables to holding registers and input registers.



Modbus_Serial_Device

- General
- Modbus Serial Device I/O Mapping
- Modbus Serial Device IEC Objects
- Information
- Status

Find | Filter Show all | Add FB for IO Channel...

Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%IW42	ARRAY [0..9] OF WORD		Modbus holding registers
		Outputs	%QW38	ARRAY [0..9] OF WORD		Modbus input registers

Reset Mapping | Always update variables: Use parent device setting

= Create new variable | = Map to existing variable

Bus Cycle Options
Bus cycle task: Use parent bus cycle setting

Registers correspond to each access type (function code)

Function code	Access type	Description
1	Read Coils	Holding register
2	Read Discrete Inputs	Input register

Function code	Access type	Description
3	Read Holding Registers	Holding register
4	Read Input Registers	Input register
5	Write Single Coil	Holding register
6	Write Single Register	Holding register
15	Write Multiple Coils	Holding register
16	Write Multiple Registers	Holding register
23	Read/Write Multiple Registers	Holding register (Read/Write)

14.4 EtherNet/IP

14.4.1 What is EtherNet/IP?

EtherNet/IP (Ethernet Industrial Protocol) is an industrial multi-vendor real-time Ethernet system that executes a communication protocol for controlling the Common Industrial Protocol (CIP) in the application layer implemented over standard Ethernet.

For details on CIP, refer to ODVA documentation.

14.4.2 Cyclic Communication Function

After the scanner device connects to an adapter device and a connection is established, the cyclic communication function allows them to send data mutually at the requested packet interval (RPI).

- Scanner device: Controllers such as PLC
- Adapter device: Robot controllers, encoders, I/O devices, etc.

The EtherNet/IP function of the GM1 controller consists of the scanner function and adapter function.

Supplementary note: About GM1 controllers used as adapter devices

- Adapter devices are classified into the following two types: Adapter devices connected under the control of the GM1 controller used as a scanner device and adapter devices that are the GM1 controller itself. To avoid confusion, the following two different terms are used in this manual.
  - Local adapter

    Adapter device that is the GM1 controller itself
  - Remote adapter

    Adapter device connected to the GM1 controller used as a scanner

14.4.3 EtherNet/IP Scanner Function

The EtherNet/IP scanner function allows the GM1 controller to communicate with EtherNet/IP adapter devices.

Communication settings for an adapter device to be connected can be configured by loading the EDS file of the adapter device. Multiple adapter devices can be connected to a scanner device.

14.4.4 Setting up the EtherNet/IP Scanner Function

This section explains how to set up the EtherNet/IP scanner function.

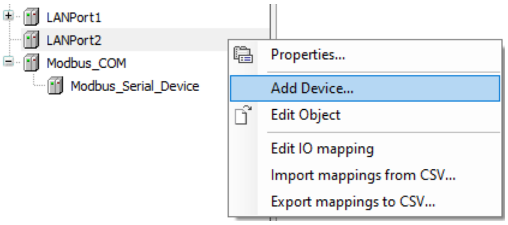
Adding devices

Add an EtherNet/IP scanner device and remote adapter device to the Device tree, as described below.

1 2 Procedure

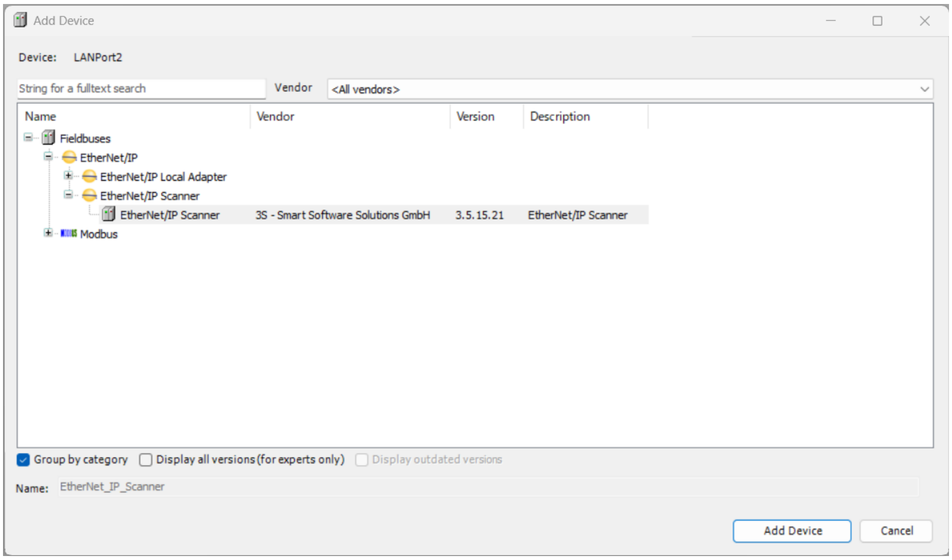
1. Add an EtherNet/IP scanner device.

1-1 Right-click the "LANPort2" object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.

1-2 Select "EtherNet_IP_Scanner" and click the [Add Device] button.



Device: LANPort2

String for a fulltext search Vendor <All vendors>

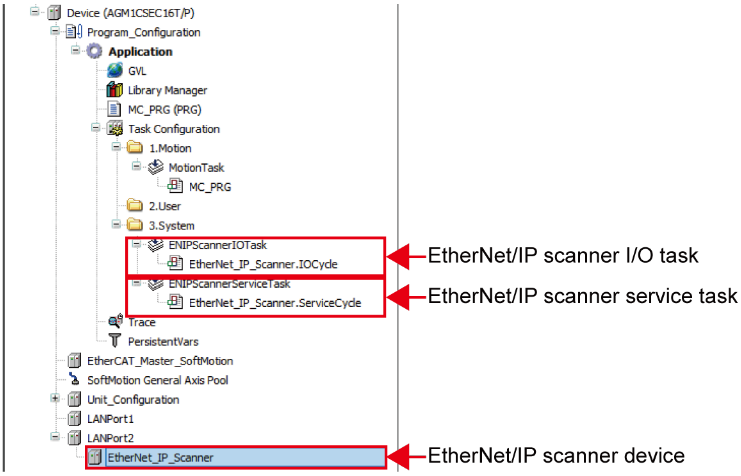
Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Local Adapter			
EtherNet/IP Scanner			
EtherNet/IP Scanner	3S - Smart Software Solutions GmbH	3.5.15.21	EtherNet/IP Scanner
Modbus			

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name: EtherNet_IP_Scanner

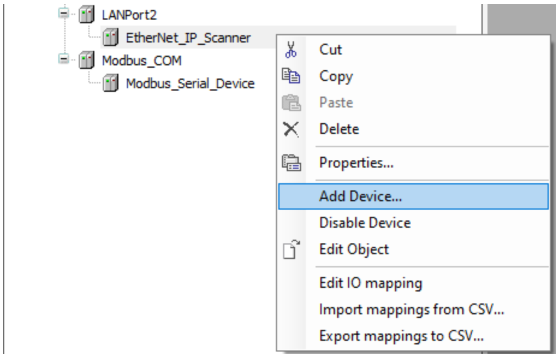
Image of added device and tasks

After an EtherNet/IP scanner device has been added, a device and tasks are added to the Device tree, as shown below.



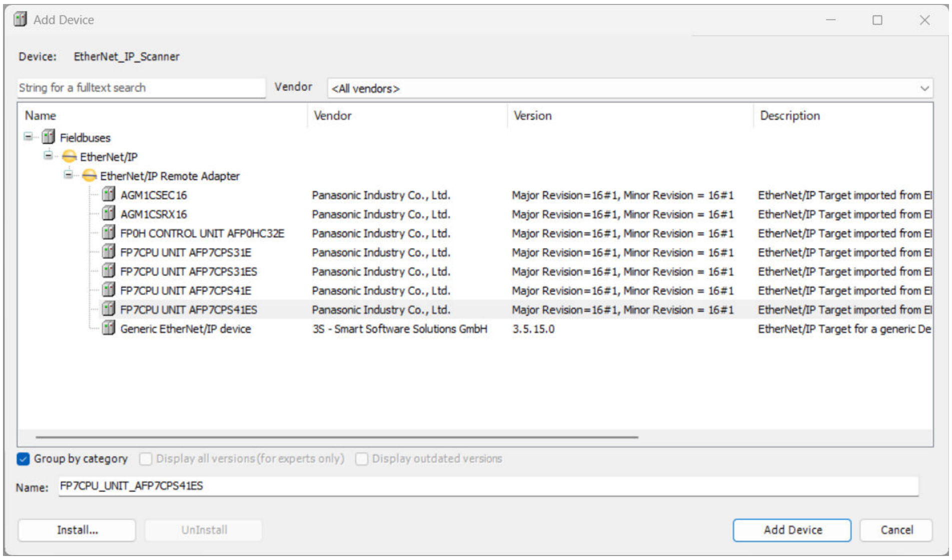
2. Add a remote adapter device.

2-1 Right-click the "EtherNet/IP scanner device" object added in "Step 1" and select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.

2-2 Select a remote adapter device to be added and click the [Add Device] button.



Add Device

Device: EtherNet_IP_Scanner

String for a fulltext search Vendor <All vendors>

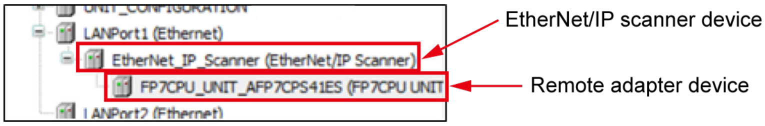
Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Remote Adapter			
AGM1CSEC16	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
AGM1CSRX16	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
FP0H CONTROL UNIT AFP0HC32E	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
FP7CPU UNIT AFP7CPS31E	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
FP7CPU UNIT AFP7CPS31ES	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
FP7CPU UNIT AFP7CPS41E	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
FP7CPU UNIT AFP7CPS41ES	Panasonic Industry Co., Ltd.	Major Revision=16#1, Minor Revision = 16#1	EtherNet/IP Target imported from El
Generic EtherNet/IP device	3S - Smart Software Solutions GmbH	3.5.15.0	EtherNet/IP Target for a generic De

☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name: FP7CPU_UNIT_AFP7CPS41ES

A new remote adapter device can also be added by selecting an EDS file. Click the [Install Device] button and select a desired EDS file.

Device tree after devices are added



UNIT_CONFIGURATION
LANPort1 (Ethernet)
EtherNet_IP_Scanner (EtherNet/IP Scanner)
FP7CPU_UNIT_AFP7CPS41ES (FP7CPU UNIT
LANPort2 (Ethernet)

EtherNet/IP scanner device

Remote adapter device

Setting up an EtherNet/IP scanner device

Set up an EtherNet/IP scanner device as below.

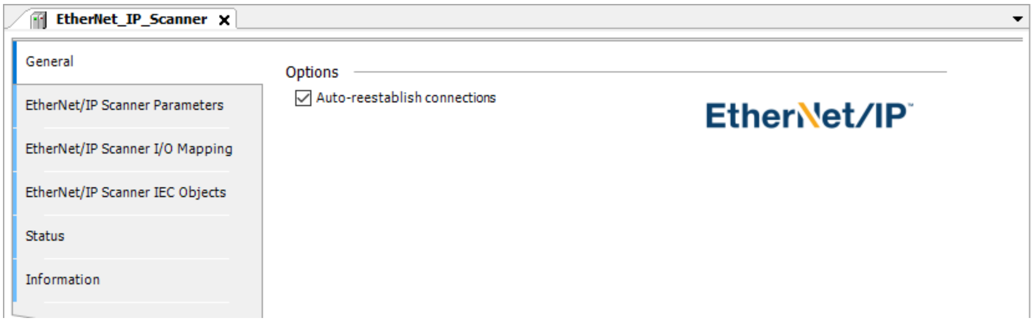
1 2 Procedure

1. Double-click "EtherNet_IP_Scanner" in the navigator pane.

2. In the "General" tab, select the "Auto-reestablish connections" check box.

When the check box is not selected: The device is stopped in the event of a communication error.

When the check box is selected: The device is reconnected automatically in the event of a communication error.


EtherNet_IP_Scanner
General
EtherNet/IP Scanner Parameters
EtherNet/IP Scanner I/O Mapping
EtherNet/IP Scanner IEC Objects
Status
Information
EtherNet/IP

Options

☑ Auto-reestablish connections

Setting up a remote adapter device

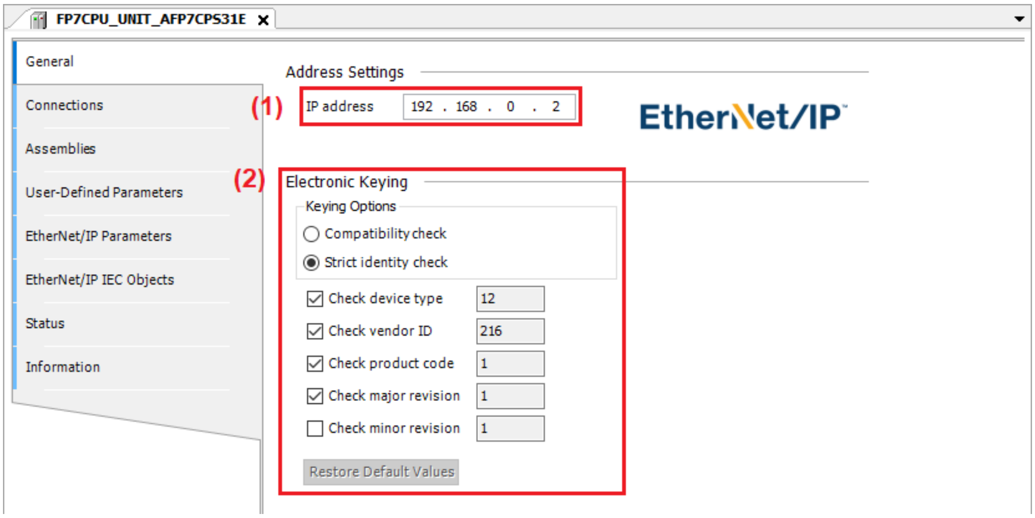
Set up a remote adapter device, as below.

Setting items for remote adapter devices differ according to the EDS file. The following procedure is explained, using Panasonic "AFP7CPS31E" as an example.

1 2 Procedure

1. Double-click "FP7CPU_UNIT_AFP7CPS31E" in the navigator pane.

2. In the "General" tab, set an IP address and items to be checked at the time of connection.



Address Settings

IP address 192 . 168 . 0 . 2

Electronic Keying

Keying Options

○ Compatibility check

● Strict identity check

☑ Check device type 12

☑ Check vendor ID 216

☑ Check product code 1

☑ Check major revision 1

□ Check minor revision 1

(1) IP address

Set the IP address of the adapter device.

(2) Electronic Keying

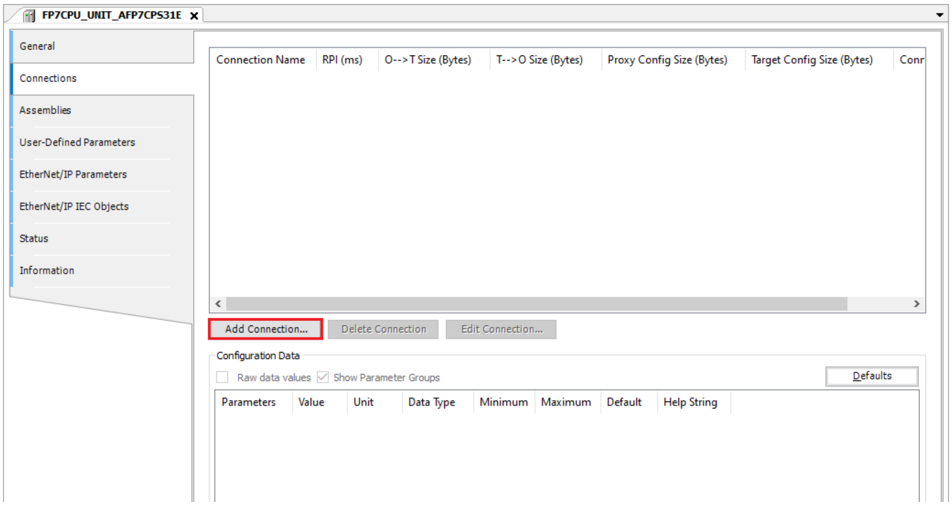
Select items to be checked at the time of connection.

- Compatibility check
  The adapter device executes its own compatibility check.
  In this case, the user can select only the "Check device type" check box.
- Strict identity check
  The user specifies check items individually.

Normally, it is OK to use the default values.

3. Set a connection point.

3-1 In the "Connection" tab, click the [Add Connection] button.



Connection Name	RPI (ms)	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)	Conr

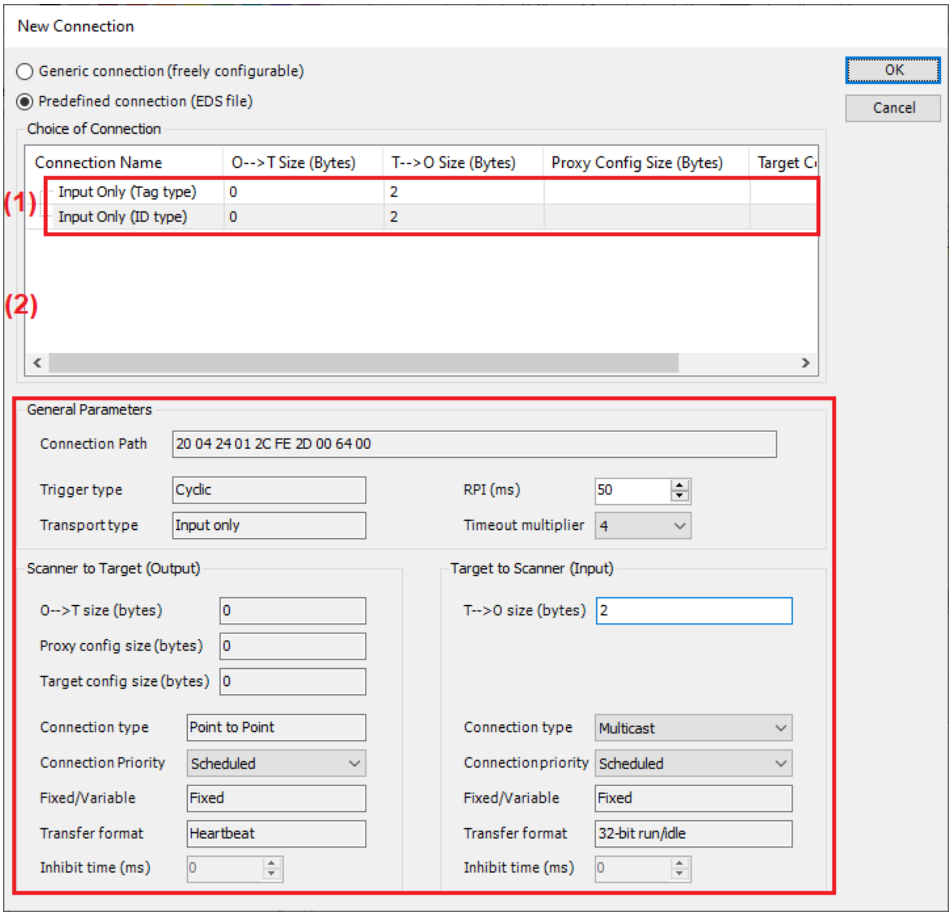
Configuration Data

□ Raw data values ☑ Show Parameter Groups

Parameters	Value	Unit	Data Type	Minimum	Maximum	Default	Help String

The "New Connection" window will be displayed.

3-2 Set up parameters required for connection.



New Connection

○ Generic connection (freely configurable)
● Predefined connection (EDS file)

Choice of Connection

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target C
Input Only (Tag type)	0	2		
Input Only (ID type)	0	2		

General Parameters

Connection Path: 20 04 24 01 2C FE 2D 00 64 00

Trigger type: Cyclic
RPI (ms): 50
Transport type: Input only
Timeout multiplier: 4

Scanner to Target (Output)

O-->T size (bytes): 0
Proxy config size (bytes): 0
Target config size (bytes): 0
Connection type: Point to Point
Connection Priority: Scheduled
Fixed/Variable: Fixed
Transfer format: Heartbeat
Inhibit time (ms): 0

Target to Scanner (Input)

T-->O size (bytes): 2
Connection type: Multicast
Connection priority: Scheduled
Fixed/Variable: Fixed
Transfer format: 32-bit run/idle
Inhibit time (ms): 0

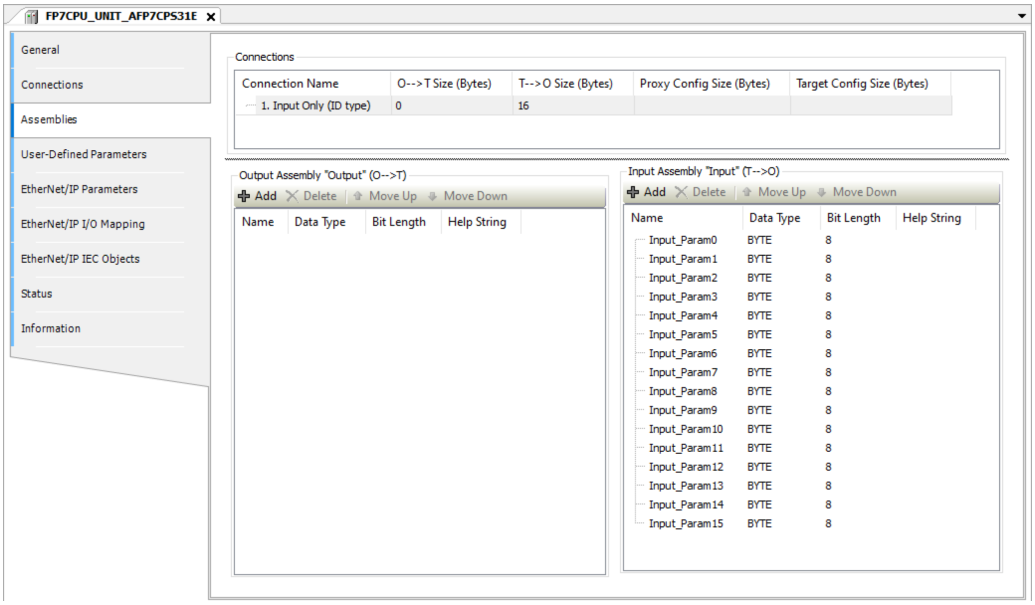
(1) This section displays the connection points that are supported by the adapter device according to the EDS file.
Select a connection point to be used.
Example: Select "Input Only (ID type)".

(2) The parameters in the "General parameters" section differ according to the selected connection point.
Example: Set "RPI" to 10 (ms) and "T→O size" to 16 (bytes).

Info.

- For tag connection, uppercase English letters cannot be used for tag names.

4. In the "Assemblies" tab, set up a data configuration.
If "T→O size" in the "General parameters" section is set to 16 bytes, the default data configuration will be as shown below.



Connections

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)
1. Input Only (ID type)	0	16		

Output Assembly "Output" (O-->T)

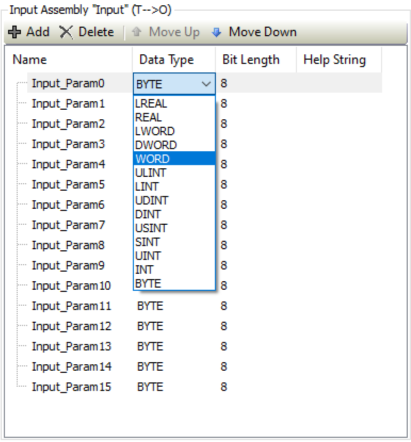
Name	Data Type	Bit Length	Help String

Input Assembly "Input" (T-->O)

Name	Data Type	Bit Length	Help String
Input_Param0	BYTE	8	
Input_Param1	BYTE	8	
Input_Param2	BYTE	8	
Input_Param3	BYTE	8	
Input_Param4	BYTE	8	
Input_Param5	BYTE	8	
Input_Param6	BYTE	8	
Input_Param7	BYTE	8	
Input_Param8	BYTE	8	
Input_Param9	BYTE	8	
Input_Param10	BYTE	8	
Input_Param11	BYTE	8	
Input_Param12	BYTE	8	
Input_Param13	BYTE	8	
Input_Param14	BYTE	8	
Input_Param15	BYTE	8	

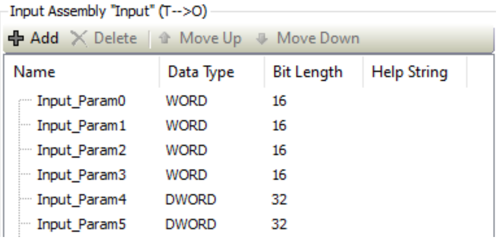
To change the data structure, click a relevant data type and select a desired data type.

In the following example, the BYTE type (16-byte data structure) is changed to the WORD type (4-word data structure) and the DWORD type (2-word data structure).



Input Assembly "Input" (T-->O)

Name	Data Type	Bit Length	Help String
Input_Param0	BYTE	8	
Input_Param1		8	
Input_Param2		8	
Input_Param3		8	
Input_Param4		8	
Input_Param5		8	
Input_Param6		8	
Input_Param7		8	
Input_Param8		8	
Input_Param9		8	
Input_Param10		8	
Input_Param11	BYTE	8	
Input_Param12	BYTE	8	
Input_Param13	BYTE	8	
Input_Param14	BYTE	8	
Input_Param15	BYTE	8	

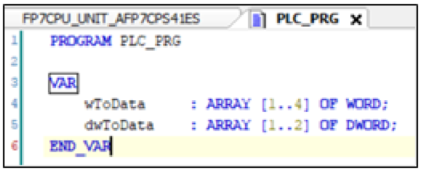


Input Assembly "Input" (T-->O)

Name	Data Type	Bit Length	Help String
Input_Param0	WORD	16	
Input_Param1	WORD	16	
Input_Param2	WORD	16	
Input_Param3	WORD	16	
Input_Param4	DWORD	32	
Input_Param5	DWORD	32	

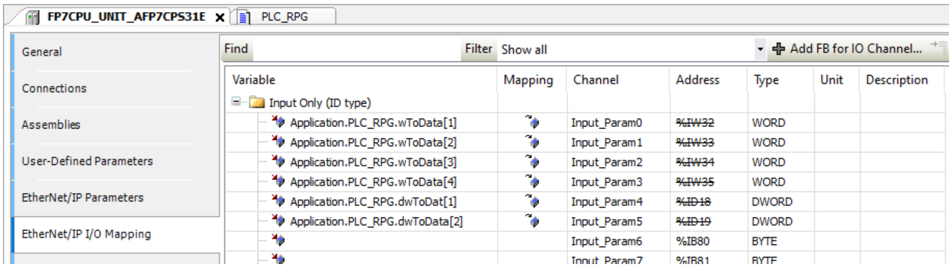
5. In the "EtherNet/IP I/O Mapping" tab, map data to variables in the application POU, as below.

5-1 In the POU, create variables to which data is to be mapped.



```
PROGRAM PLC_PRG

VAR
    wToData     : ARRAY [1..4] OF WORD;
    dwToData    : ARRAY [1..2] OF DWORD;
END_VAR
```

5-2 Map I/O data to each variable that has been created.

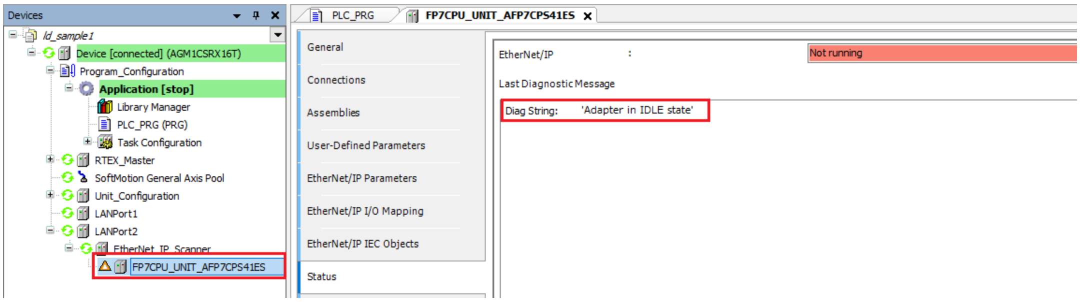


Variable	Mapping	Channel	Address	Type	Unit	Description
Input Only (ID type)						
Application.PLC_RPG.wToData[1]		Input_Param0	~~%IW32~~	WORD		
Application.PLC_RPG.wToData[2]		Input_Param1	~~%IW33~~	WORD		
Application.PLC_RPG.wToData[3]		Input_Param2	~~%IW34~~	WORD		
Application.PLC_RPG.wToData[4]		Input_Param3	~~%IW35~~	WORD		
Application.PLC_RPG.dwToDat[1]		Input_Param4	~~%ID18~~	DWORD		
Application.PLC_RPG.dwToData[2]		Input_Param5	~~%ID19~~	DWORD		
		Input_Param6	%IB80	BYTE		
		Input_Param7	%IB81	BYTE		

14.4.5 EtherNet/IP Scanner Operation

When a project in which EtherNet/IP scanner settings have been configured is downloaded to the GM1 controller and then an adapter is connected, cyclic communication is started, regardless of whether the GM1 controller is set to RUN or STOP mode.

When the GM1 controller is set to STOP mode, the remote adapter is placed in "Adapter in IDLE state" and displayed as △ in the Device tree.

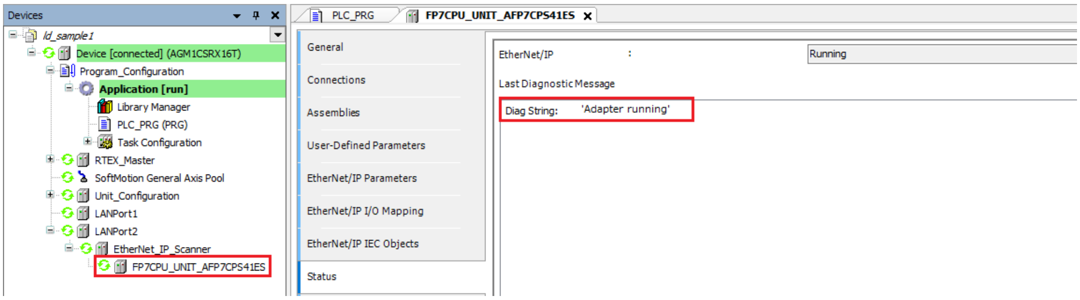


EtherNet/IP : Not running

Last Diagnostic Message

Diag String: 'Adapter in IDLE state'

When the GM1 controller is set to RUN mode, the remote adapter is placed in "Adapter running" and displayed as ○ in the Device tree.



EtherNet/IP : Running

Last Diagnostic Message

Diag String: 'Adapter running'

To perform a status check or reset using a program, use I/O of the remote adapter device.

The following is an example of adapter operation using device I/O.

- Declaration section

```
PROGRAM PLC_PRG
VAR
    eState :
    IoDrvEtherNetIP.AdapterState;// Remote adapter status
    xDiagnosticAvailable : BOOL; // TRUE if there is diagnostic information
    sDiagString : STRING; // Diagnostic string
    xAcknowledge : BOOL := FALSE; // Approve diagnostic information
    xReset : BOOL := FALSE; // Remote adapter reset
END_VAR
```

- Implementation section

```
eState := FP7CPU_UNIT_AFP7CPS41ES.eState;
xDiagnosticAvailable := FP7CPU_UNIT_AFP7CPS41ES.xDiagnosticAvailable;
sDiagString := FP7CPU_UNIT_AFP7CPS41ES.sDiagString;
FP7CPU_UNIT_AFP7CPS41ES.xAcknowledge := xAcknowledge;
FP7CPU_UNIT_AFP7CPS41ES.xReset := xReset;
```

The current state of the remote adapter is stored in eState.

Example	When line is connected normally:	RUNNING
	When line is disconnected:	ENCAPSULATION_CONFIG

If an error occurs, xDiagnosticAvailable will be set to TRUE and a message will be found in sDiagString.

If xAcknowledge is set to TRUE, xDiagnosticAvailable will return to FALSE.

If xReset is set to TRUE, the line will be closed temporarily and then reconnected.

(To reset all remote adapters simultaneously, use xReaset of the EtherNet/IP scanner device.)

14.4.6 EtherNet/IP Adapter Function

The EtherNet/IP adapter function allows the GM1 controller to communicate with EtherNet/IP scanner devices.

14.4.7 Setting up the EtherNet/IP Adapter Function

This section explains how to set up the EtherNet/IP adapter function.

Adding devices

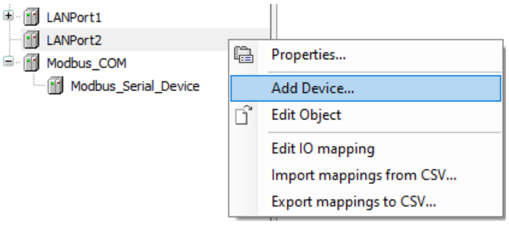
Add a local adapter device and module device to the Device tree, as described below.

1 2 Procedure

1. Add a local adapter device.

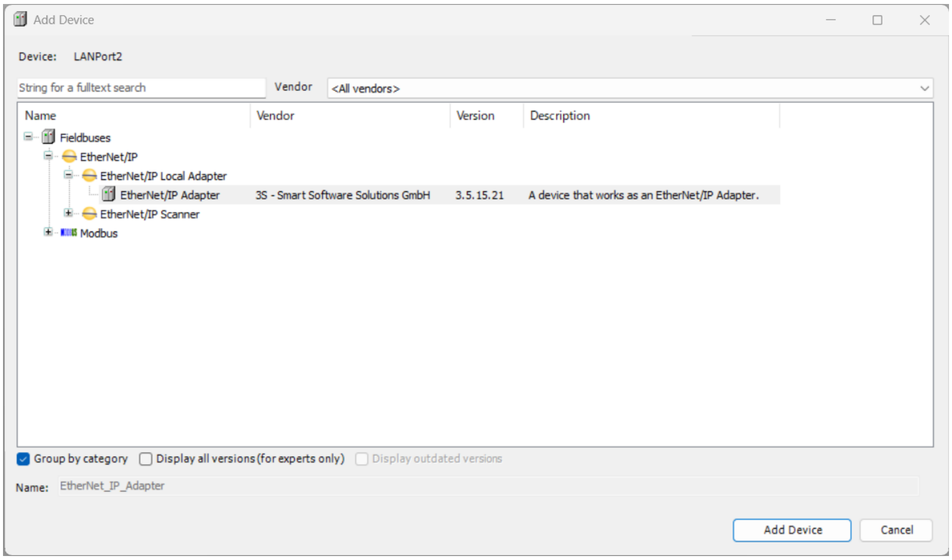
   A local adapter device serves as a connection point to which the scanner device connects.

   1-1 Right-click the "LANPort2" object in the navigator pane and then select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.

1-2 Select "EtherNet_IP_Adapter" and click the [Add Device] button.



Device: LANPort2

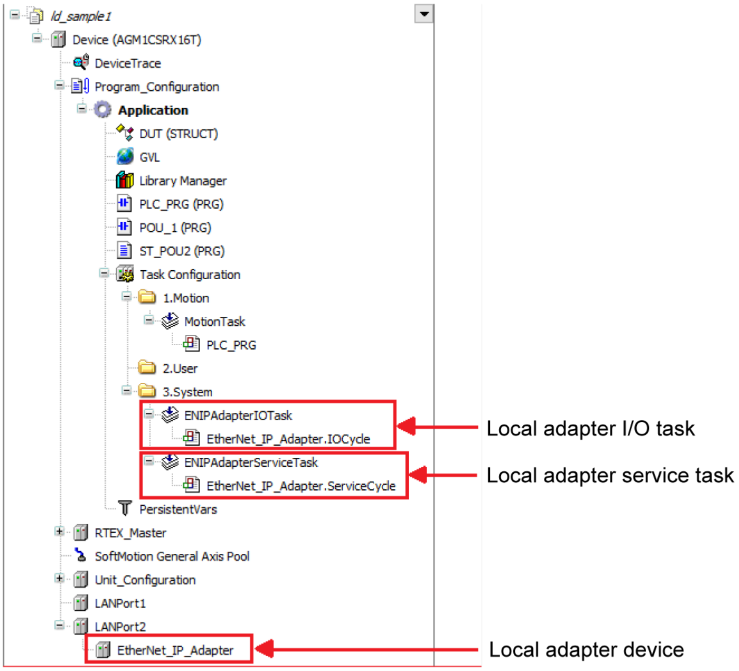
Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Local Adapter			
EtherNet/IP Adapter	3S - Smart Software Solutions GmbH	3.5.15.21	A device that works as an EtherNet/IP Adapter.
EtherNet/IP Scanner			
Modbus			

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: EtherNet_IP_Adapter

Image of added device and tasks

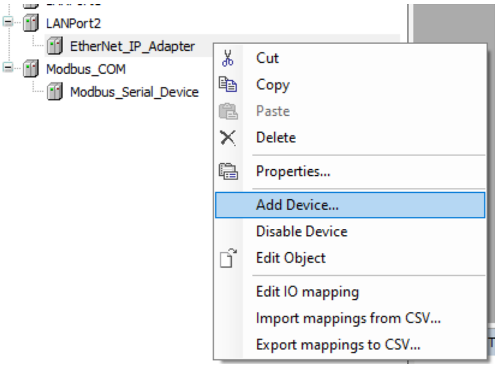
After a local adapter device has been added, a device and tasks are added to the Device tree, as shown below.



2. Add a module device.

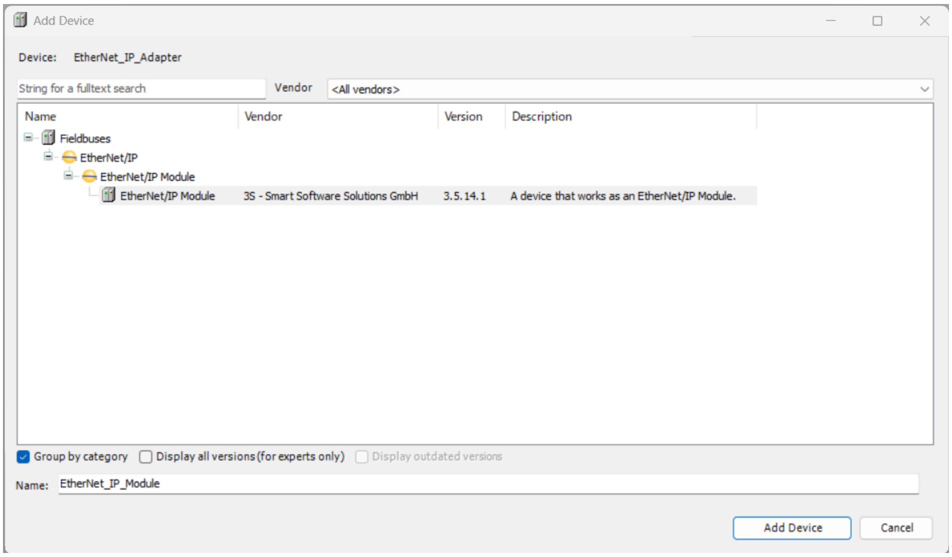
A module device defines data to be transferred via cyclic communication.

2-1 Right-click the "Local adapter device" object added in "Step 1" and select "Add Device" from the context-sensitive menu that is displayed.



The "Add Device" dialog box will be displayed.

2-2 Select the "EtherNet/IP Module" object to be added and click the [Add Device] button.



Add Device

Device: EtherNet_IP_Adapter

String for a fulltext search Vendor <All vendors>

Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Module			
EtherNet/IP Module	3S - Smart Software Solutions GmbH	3.5.14.1	A device that works as an EtherNet/IP Module.

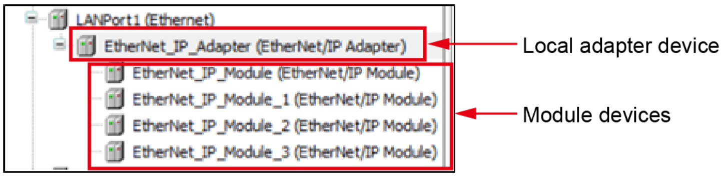
☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: EtherNet_IP_Module

Image of added devices

Multiple module devices can be added within a local adapter device.

The following is an example of four module devices added to a local adapter device.



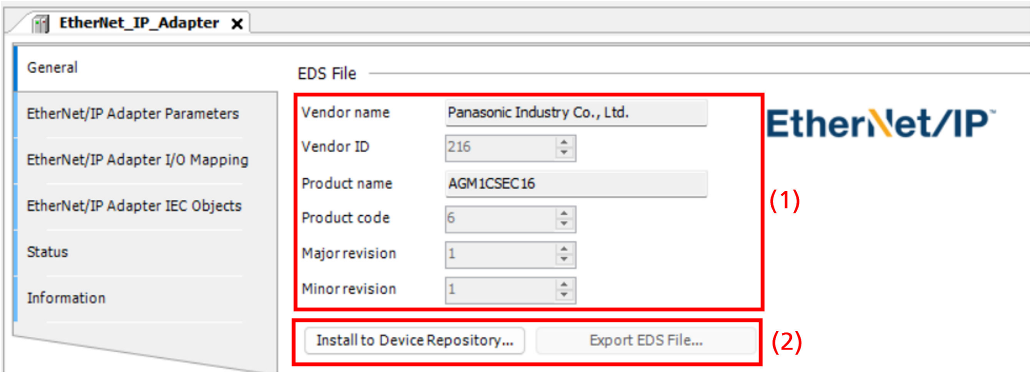
LANPort1 (Ethernet)
- EtherNet_IP_Adapter (EtherNet/IP Adapter)
  - EtherNet_IP_Module (EtherNet/IP Module)
  - EtherNet_IP_Module_1 (EtherNet/IP Module)
  - EtherNet_IP_Module_2 (EtherNet/IP Module)
  - EtherNet_IP_Module_3 (EtherNet/IP Module)

Settings of local adapter device

Check the settings of the local adapter device.

1 2 Procedure

1. Double-click "EtherNet_IP_Adapter" in the navigator pane.
2. In the "General" tab, check the settings of the local adapter device.


EtherNet_IP_Adapter
General
EtherNet/IP Adapter Parameters
EtherNet/IP Adapter I/O Mapping
EtherNet/IP Adapter IEC Objects
Status
Information
EDS File
Install to Device Repository...
Export EDS File...
EtherNet/IP
(1)
(2)

Vendor name: Panasonic Industry Co., Ltd.
Vendor ID: 216
Product name: AGM1CSEC16
Product code: 6
Major revision: 1
Minor revision: 1

(1) The settings of the local adapter device are shown below. However, the settings cannot be changed.

Vendor name: Panasonic Industry Co., Ltd.
Vendor ID: 216
Product name: AGM1CSEC16
Product code: 6
Major revision: 1
Minor revision: 1

(2) For the following items, the settings cannot be changed.

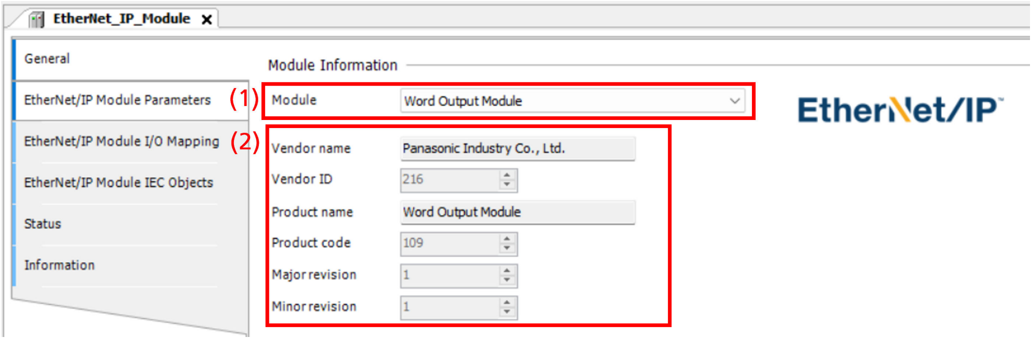
- Install to Device Repository
  The remote adapter device specified here is registered as a device in CODESYS.
- Export EDS File
  The EDS file specified here is output.
  For scanner device settings, use the EDS file provided by Panasonic ("panasonicgm1csec16_0006_0101.eds").

Setting up a module device

Set up a module device, as below.

Procedure

1. Double-click "EtherNet_IP_Module" in the navigator pane.
2. In the "General" tab, set module information.


EtherNet_IP_Module
General
EtherNet/IP Module Parameters
EtherNet/IP Module I/O Mapping
EtherNet/IP Module IEC Objects
Status
Information
Module Information
EtherNet/IP
(1)
(2)

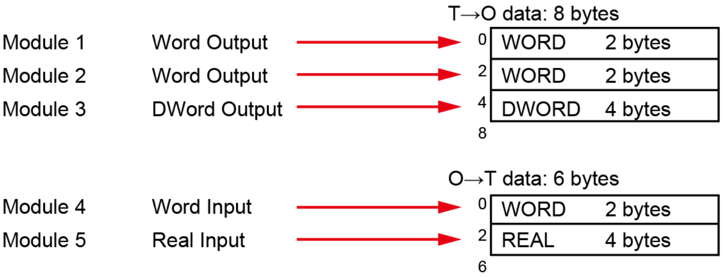
Module: Word Output Module
Vendor name: Panasonic Industry Co., Ltd.
Vendor ID: 216
Product name: Word Output Module
Product code: 109
Major revision: 1
Minor revision: 1

2-1 (1) Select a desired module type from the following 10 types.

Module type	Size	Direction
Byte Input	1 byte	O→T
Byte Output	1 byte	T→O
Word Input	1 word (2 bytes)	O→T
Word Output	1 word (2 bytes)	T→O
DWord Input	1 double-word (4 bytes)	O→T
DWord Output	1 double-word (4 bytes)	T→O
Real Input	1 single-precision real number (4 bytes)	O→T
Real Output	1 single-precision real number (4 bytes)	T→O
Big Input	505 bytes	O→T
Big Output	509 bytes	T→O

By generating multiple module devices and setting module types, data structure can be created within cyclic data.

Example) T→O data: 8 bytes, O→T data: 6 bytes



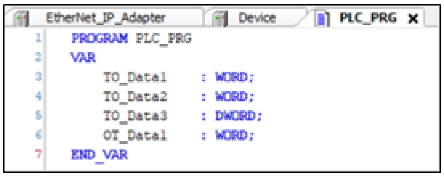
Note

- The maximum data length within a single connection point is as follows:
  O→T data: 505 bytes
  T→O data: 509 bytes

2-2 There is no need to set all items in (2), as values cannot be entered.

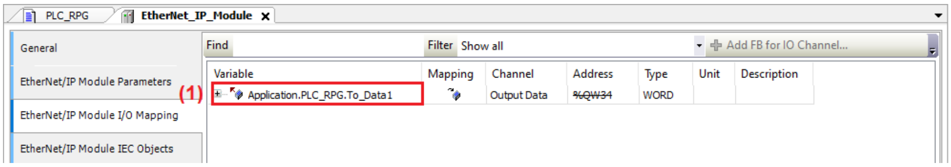
3. In the "EtherNet/IP Module I/O Mapping" tab, map data to variables in the application POU, as below.

3-1 In the POU, create variables to which data is to be mapped.



```
PROGRAM PLC_PRG
VAR
    TO_Data1    : WORD;
    TO_Data2    : WORD;
    TO_Data3    : DWORD;
    OT_Data1    : WORD;
END_VAR
```

3-2 Map I/O data to each variable that has been created. Double-click the section indicated by (1) and select a variable to which I/O data is to be mapped.

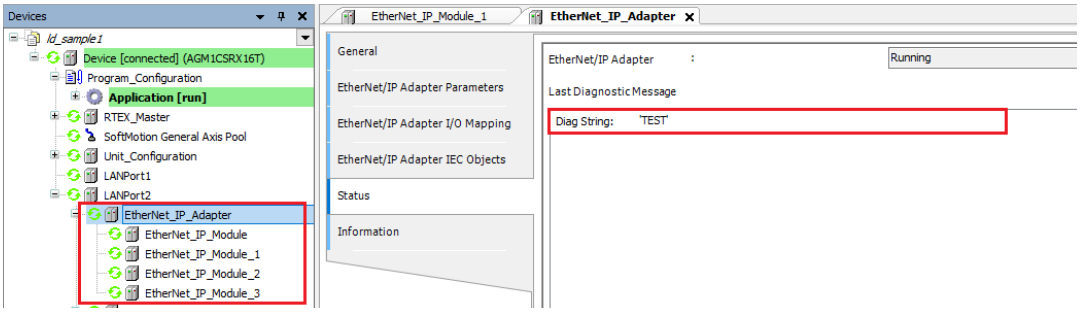

PLC_RPG
EtherNet_IP_Module
General
EtherNet/IP Module Parameters
EtherNet/IP Module I/O Mapping
EtherNet/IP Module IEC Objects
Find
Filter Show all
Add FB for IO Channel...
(1)

Variable	Mapping	Channel	Address	Type	Unit	Description
Application.PLC_RPG.To_Data1		Output Data	~~%QW34~~	WORD		

14.4.8 EtherNet/IP Adapter Operation

When a project in which EtherNet/IP adapter settings have been configured is downloaded to the GM1 controller and then RUN mode is invoked, the adapter device responds to a ForwordOpen packet from the scanner device and cyclic communication is started.

The following shows the respective states of the local adapter device and module devices during normal operation.



EtherNet/IP Adapter : Running

Last Diagnostic Message

Diag String: 'TEST'

The local adapter device and module devices are displayed with "○" symbol in the Device tree and the status of the local adapter device is displayed as "Adapter running".

The local adapter device can be connected from multiple scanner devices.

Note

- ExclusiveOwner connection (using O→T data) is allowed for only one scanner device. This is to prevent the same variable from being overwritten with input data from multiple scanner devices.

14.5 Communicating with Display Units

Variables that communicate with the display can be set in the GM Programmer's symbol configuration.

When code generation is executed in the set project, the symbol definition XML file (.xml) is generated in the same folder as the project file (.project), so you can import it into the display and share the variable definition.

14.5.1 Symbol Configuration

This section describes the symbol configuration function.

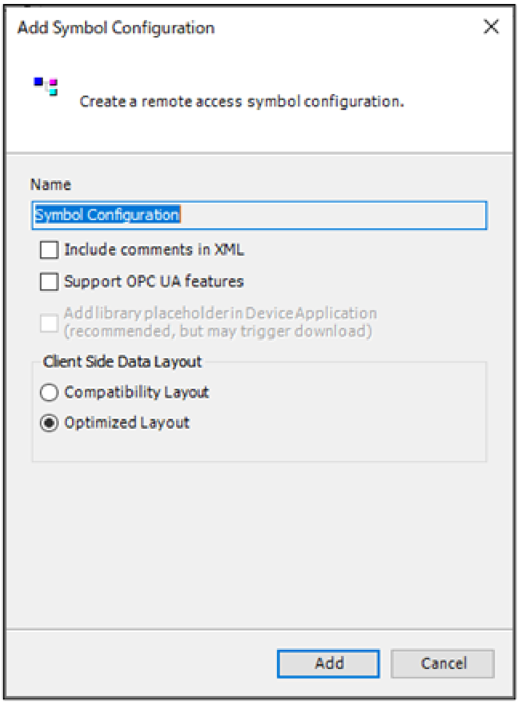
With this function, a symbol configuration object can be added and used.

You can set variables communicating with the display unit as well as variables open to the OPC UA server.

■ Creating symbol configuration object

1. Right-click the [Application] object in the navigation pane and select **Add Object**>**Symbol Configuration** from the context-sensitive menu that is displayed.

   The "Add Symbol Configuration" dialog box will be displayed.



Add Symbol Configuration

Create a remote access symbol configuration.

Name

Symbol Configuration

□ Include comments in XML

□ Support OPC UA features

□ Add library placeholder in DeviceApplication (recommended, but may trigger download)

Client Side Data Layout

○ Compatibility Layout

● Optimized Layout

2. Enter a name, select check boxes for necessary items, and click the [Add] button.

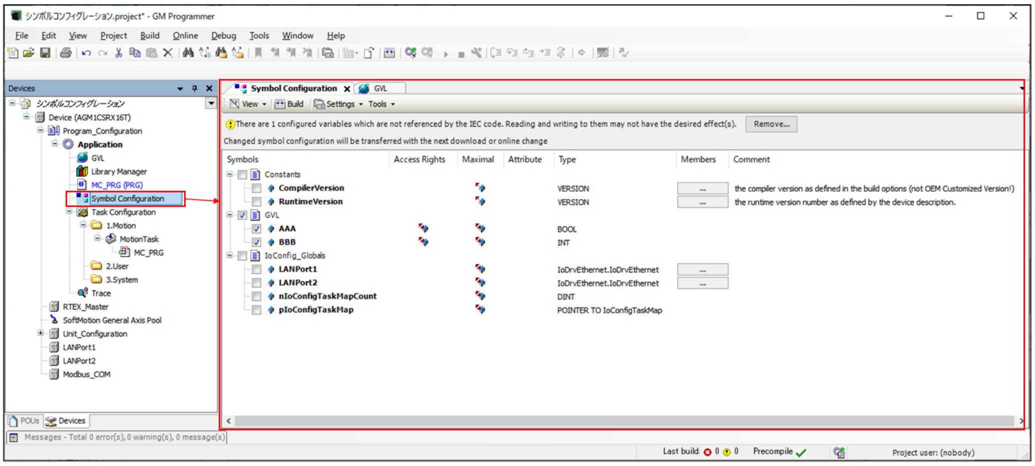
   A symbol configuration object will be added under the [Application] object.

List of setting items

Item		Description
Include comments in XML		Includes comments on variables in an XML file when the symbol configuration is exported to the XML file.
Support OPC UA features		Enables the symbol configuration OPC UA Server function.
Client Side Data Layout		Specifies a method for generating a client side data layout.
	Compatibility Layout	Generates a data layout compatible with older version. Do not use this setting because of possible trouble occurrence.
	Optimized Layout	Use this setting.

■ Editing symbol configuration

1. Double-click the [Symbol Configuration] object in the navigator pane.
   The setting pane will be displayed in the main pane.



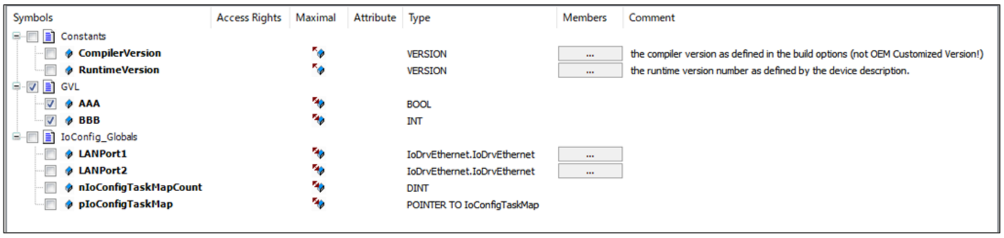
There are 1 configured variables which are not referenced by the IEC code. Reading and writing to them may not have the desired effect(s).

Changed symbol configuration will be transferred with the next download or online change

Messages - Total 0 error(s), 0 warning(s), 0 message(s)

Press [Build] to display the declared variables list in the window.
In the setting pane, the symbols (variables) table and menu bar for editing are displayed.

- Symbols table



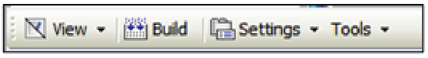
Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
⊟ □ Constants						
□ CompilerVersion				VERSION	...	the compiler version as defined in the build options (not OEM Customized Version!)
□ RuntimeVersion				VERSION	...	the runtime version number as defined by the device description.
⊟ ☑ GVL						
☑ AAA				BOOL		
☑ BBB				INT		
⊟ □ IoConfig_Globals						
□ LANPort1				IoDrvEthernet.IoDrvEthernet	...	
□ LANPort2				IoDrvEthernet.IoDrvEthernet	...	
□ nIoConfigTaskMapCount				DINT		
□ pIoConfigTaskMap				POINTER TO IoConfigTaskMap		

The symbols table shows the following components:

Item name	Description
Symbols	Displays a list of variables. If the check boxes are selected, values can be read or written from display units and OPC UA devices.

Item name	Description
Access Rights	If a symbol in the [Access Rights] column is clicked, the right of access to the symbol can be changed. Access right icons Read only Write only Read / Write With these rights, functions, function blocks, methods, and programs can be accessed.
Maximal	Maximum access rights.
Attribute	For symbols set by Attribute ‘symbol’ pragma, set access rights are displayed.
Type	Displays the data type of the symbol.
Members	A elision mark button is displayed only for structure-type symbols. By clicking the elision mark button, you can set the access right member by member in the structure. Note: This selection is applied to all instances of this data type for which the symbol is exported. Note: If members in the structure cannot be selected, an asterisk is displayed at the member check box, indicating that all the members that can be read or written will be exported.
Comment	Displays a description of the symbol.

- Menu bar



The menu bar displays the following menus:

Item name		Description
View		When the following items used in the symbol configuration are clicked, symbol display is filtered.
	Project, not yet configured	Symbols that are not added to the symbol configuration but are provided in projects.
	Library, not yet configured	Symbols that are not added to the symbol configuration but are provided in libraries.
	Symbols exported on attribute	Symbols set by Attribute ‘symbol’ pragma.
Build		This item is used to compile a project. Note: Build needs to be executed to display variables in the symbol table.
Settings	Support OPC UA features	Enables the symbol configuration OPC UA function.
	Include comments in XML	Includes comments on variables in an XML file when the symbol configuration is exported to the XML file.
	Include node flags in XML	Includes namespace node flags in an XML file when the symbol configuration is exported to the XML file.

Item name		Description
	Include comments and attributes	About comments and attributes, you can set details of the symbol configuration and contents required to be included in the XML file. XML symbol file contents can be configured to include "namespace node flag", "comment", and "attribute". Do not use this setting for symbol table contents because the OPC UA function does not support it.
	Configure sync with IEC task	You can set a function with which to access variables in sync with an IEC task. Do not use.
	Compatibility Layout	Generates a data layout compatible with older version. Do not use this setting because of possible trouble occurrence.
	Optimized Layout	Generates a data layout independent of the internal layout. Use this setting.
	Use empty namespace by default	Do not use.
	Enable direct I/O access	Do not use.
	Enable symbol set	Do not use.
Tools	Save XML schema file	Saves a symbol file in XSD format to use it in external programs.

14.6 OPC UA Server

14.6.1 What is OPC UA?

OPC Unified Architecture (OPC UA) is an interoperability standard for the secure and reliable exchange of data in the industrial automation field and other industries.

Abstracting PLC specific protocols (such as Modbus) into a standardized interface, OPC UA is platform-independent and ensures the seamless flow of information among many devices.

As a result, users with HMI (display unit) or SCADA systems can build systems that interface with and seamlessly interact with devices via OPC.

14.6.2 OPC UA Server Settings

Procedure

1. With reference to section "14.5.1 Symbol Configuration", select variables to be opened.
2. Select the "Support OPC UA features" check box to enable the server function.
3. After downloading the project to the GM1 controller, turn the controller OFF and then ON to get the OPC UA Server function enabled.

Note

- Variables that are opened are used in common with settings used for communication with the display unit.
- Please use the number of items that can be monitored at one time within the range of "14.6.3 OPC UA Server Specifications".

14.6.3 OPC UA Server Specifications

OPC UA server specifications with the GM1 controller are described below:

Item	Details
Available ports	Both LAN ports 1 and 2
Supported Profiles	Micro Embed Device Server Profile
Server URL	Port 1: opc.tcp://192.168.1.5:4840 (default) Port 2: opc.tcp://192.168.2.5:4840 (default) * IP addresses depend on settings
Maximum number of sessions	2
Number of all monitored items on server	1000 Monitoring interval: Minimum 300 ms Client notification interval: Minimum 500 ms

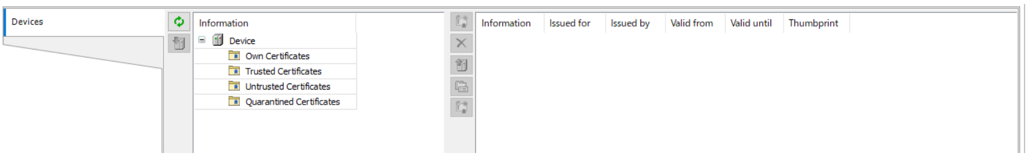
Item	Details
Variable types not permitted to be opened to server	● Real number (LREAL: permitted, REAL: not permitted) ● Character string (WSTRING: permitted, STRING: not permitted) ● Time (only DATE_AND_TIME(DT): permitted, others: not permitted) ● Constant (const), Interface, Properties ● Union, Pointer
Limit on the number of variable name characters	● Up to 255 characters ● Half-width characters (alphanumeric characters, symbols)
Security settings	Security mode: ● None: Neither signature nor encryption required ● Sign: Only signature required ● Sing＆Encrypt: Both signature and encryption required Security policy: ● Basic256 Sha256
Certificates	Classified as below (Each 32 units can be registered) ● Own Certificates: own certificates ● Trusted Certificates: certificates that are trusted ● UnTrusted Certificates: certificates that are not trusted ● Quarantined Certificates: certificates that are quarantined
User authentication	● Username&Password (user name and password) ● Anonymous (anonymous user) * Do not use anonymous user from a security perspective

14.6.4 Security Settings

You can set security settings for secure communication with the OPC UA server

1 2 Procedure

1. Perform a network scan and connect to the GM1 controller.
2. Click the display security screen of menu bar **View**>**Security Screen**.
3. Click the ⟳ button to display the certificate information in the device.

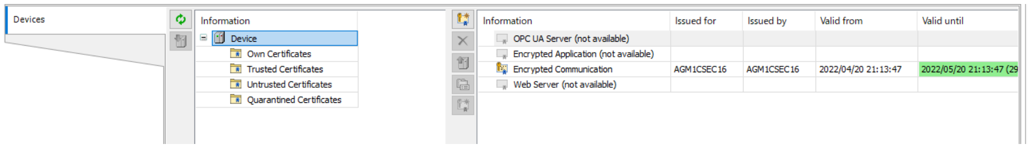


Devices

Information
- Device
  - Own Certificates
  - Trusted Certificates
  - Untrusted Certificates
  - Quarantined Certificates

Information	Issued for	Issued by	Valid from	Valid until	Thumbprint

4. Select "Device" to display the certificate status.

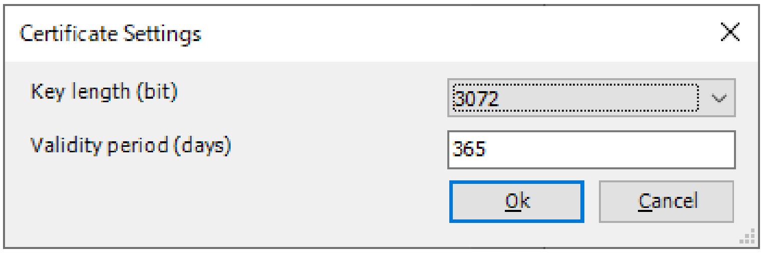


Devices

Information
- Device
  - Own Certificates
  - Trusted Certificates
  - Untrusted Certificates
  - Quarantined Certificates

Information	Issued for	Issued by	Valid from	Valid until
OPC UA Server (not available)				
Encrypted Application (not available)				
Encrypted Communication	AGM1CSEC16	AGM1CSEC16	2022/04/20 21:13:47	2022/05/20 21:13:47 (29
Web Server (not available)				

5. Select "OPC UA Server" and click the button on the certificate icon.
6. The screen for setting the length and validity period of the certificate key will be displayed. Set it and click the "OK" button.

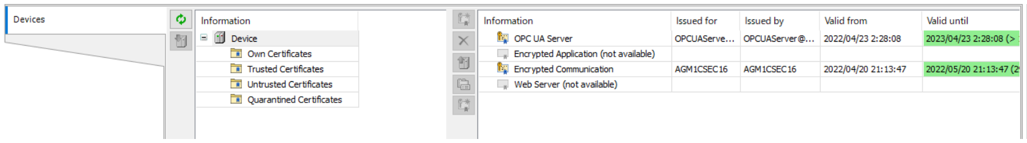


Certificate Settings

Key length (bit): 3072

Validity period (days): 365

7. Once the OPC UA server certificate has been created, it becomes active.



Devices

Information
- Device
  - Own Certificates
  - Trusted Certificates
  - Untrusted Certificates
  - Quarantined Certificates

Information	Issued for	Issued by	Valid from	Valid until
OPC UA Server	OPCUAServe...	OPCUAServer@...	2022/04/23 2:28:08	2023/04/23 2:28:08 (>
Encrypted Application (not available)				
Encrypted Communication	AGM1CSEC16	AGM1CSEC16	2022/04/20 21:13:47	2022/05/20 21:13:47 (2
Web Server (not available)				

14.7 FTP Server Function

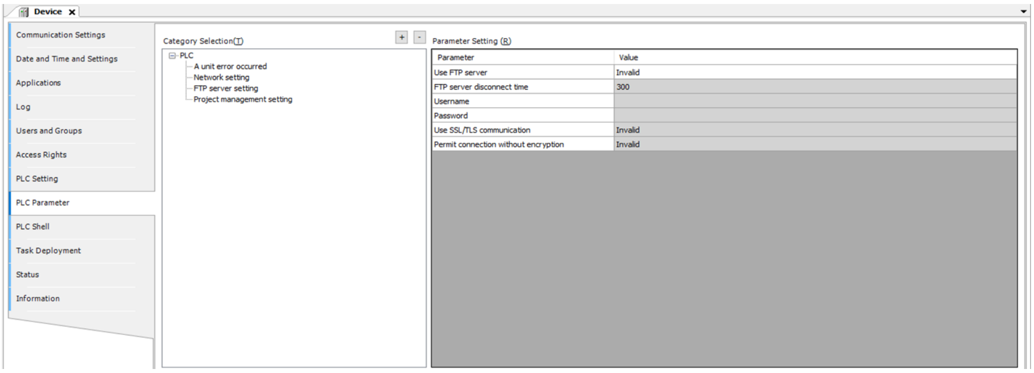
14.7.1 Overview of FTP Server Function

The FTP server function allows the FTP client to read files from the SD card mounted on the GM1 controller and write files into the SD card.

14.7.2 Setup Procedure

1 2 Procedure

1. Double-click the [Device] object in the navigator pane to display the Device setting window.
2. Click the "PLC parameters" tab in the Device setting window.
3. Click "FTP server settings" in the "Category selection".
4. When the "Use FTP server" setting is changed to "Valid" in "Parameter settings", the settings in items take effect. For the contents of setting items, refer to "14.7.3 List of Setting Items".



Parameter	Value
Use FTP server	Invalid
FTP server disconnect time	300
Username	
Password	
Use SSL/TLS communication	Invalid
Permit connection without encryption	Invalid

Info.

- The FTP server function is supported by the unit firmware Ver. 1.3 or later.
- Up to three FTP clients can be connected at the same time.

14.7.3 List of Setting Items

List of setting Items (FTP server)

Setting item	Default	Description
Use FTP server	Not use	Select whether to use the FTP server function.
FTP server disconnection time	300	Setting unit: 1 (second) Setting range: 300 to 4,294,967,295 Enter an inactive communication monitoring time for FTP connection. When the time for which the FTP server is in an inactive communication state has exceeded the specified time, the server is automatically disconnected.
User name	—	Enter a user name for logging in to the FTP server. Number of characters: 4 to 32 Allowed characters: lowercase and uppercase alphabetic characters, numerals, and an underscore (_)
Password	—	Enter a password for logging in to the FTP server. Number of characters: 8 to 32 Allowed characters: lowercase and uppercase alphabetic characters, numerals, and symbols (space (SP), !, #, $, %, &, (,), *, +, -, ., /, :, ;, <, =, >, ?, @, [,], ^, _, {, |, }, ~) Combine three different types out of lowercase and uppercase alphabetic characters, numerals, and symbols.
Use SSL/TLS communication	Not use	Set whether to use SSL/TLS communication (Explicit mode) under encrypted connection. Only SSL/TLS version TLS1.2 can be used.
Permit connection without encryption	Not permit	Select whether to allow communication even under insecure state by configuring settings without encryption or to allow only secure communication using the SSL/TLS (encryption). This item is operative only when "Use SSL/TLS communication" is set to "Valid". If set to "Valid", match the SSL/TLS (encryption) communication setting on the FTP client side to the setting on this side.

Info.

- If you forget your password, you cannot log in to the GM1 controller. In this case, execute device reset from the GM1 controller.

  （For details, refer to the "9.5.3 Executing Device Reset from GM1 Controller"）

- The root directory of the FTP server is the root directory of the SD card on the GM1 Controller.

14.7.4 FTP Server Standards

The FTP server provides FTPS functions according to the following SSL/TLS specifications.

Item	Description
Protocol	TLS1.2
Cipher suite	Available cipher suites differ depending on the cipher system of the certificate used. "When RSA certificate is used" ● TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ● TLS_RSA_WITH_AES_128_GCM_SHA256 ● TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ● TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ● TLS_RSA_WITH_AES_256_CBC_SHA256 ● TLS_RSA_WITH_AES_128_CBC_SHA256 ● TLS_RSA_WITH_AES_256_CBC_SHA "When ECDSA certificate is used" ● TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ● TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ● TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA * The suites are listed in order of precedence. * Key length: ECDHE/ECDSA ＝ 256bit、RSA ＝ 2048bit * The RSA certificate is built in by default.
Server certificate	The server certificate is a default certificate, which is used when the user certificate is not set, or is a user certificate, which is used when the user certificate is set. Default certificate: A 2048-bit self-signed RSA certificate that is set by factory default. User certificate: A certificate set by the user. * This can be set as either a RSA certificate or ECDSA certificate.
Connection method	Explicit connection. * Implicit connection is not supported.

Info.

- For the method of writing a user certificate, refer to "14.7.5 SSL/TLS Certificate Settings".
- A desired certificate / secret key can be set for use in the SSL/TLS communication. * When this is not set, the default certificate is used.
- Do not perform the following operations while the operation mode is in RUN mode. Please change to STOP mode and execute.
  - Insert the SD card with "Use FTP server" enabled.
  - Registering and deleting certificates with "14.7.5 SSL/TLS Certificate Settings".

14.7.5 SSL/TLS Certificate Settings

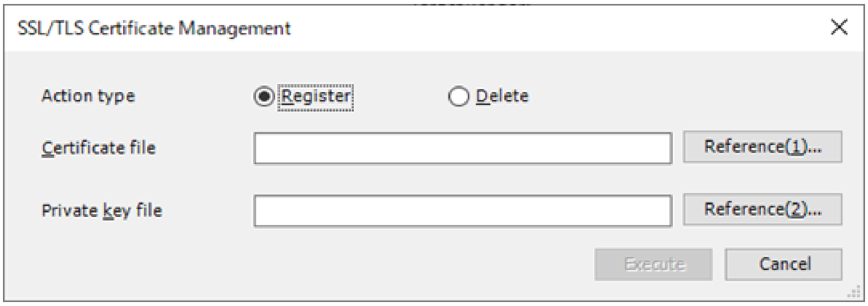
Set this when using an SSL/TLS certificate / secret key prepared by the user for the SSL/TLS communication of the FTP server function.

* When this is not set, the default self-signed certificate / secret key is used.

Perform the procedure in "8.4 Communication Setting" in advance to connect GM Programmer to the GM1 controller.

1 2 Procedure

1. From the menu bar, select **Online**>**SSL/TLS Certificate Management**.
   The "SSL/TLS Certificate Management" screen is displayed.



SSL/TLS Certificate Management

Action type ● Register ○ Delete

Certificate file [ ] Reference(1)...

Private key file [ ] Reference(2)...

Execute Cancel

2. Select "Register" under "Operation type".
   Select "Delete" to delete the certificate and follow the following step 5.

3. Press the [Refer (1)] button and select a certificate file.

Certificate designation file

Item	Description
Extension	pem, cer, crt
File format	ASN.1 (Base64)
Data size	8,192 bytes or less

4. Press the [Refer (2)] button and select the secret key file that corresponds to the certificate selected in step 3.

Secret key designation file

Item	Description
Extension	pem, key
File format	ASN.1 (Base64)
Data size	8,192 bytes or less

5. Press the [Execute] button.
   The designated certificate / secret key is transferred to the GM1 controller and an attempt is made to restart the FTP server. When the FTP server is successfully restarted, the

transferred certificate / secret key is reflected. The same reflection timing applies when the certificate is deleted.

Info.

- Checking user certificate storage status

  The storage status of the user certificate can be checked by the SYS_GetFTPCertState function. For details, refer to the *GM1 Series Reference Manual (Instruction Edition)*.
- The user certificate is not transferred when a project is downloaded.

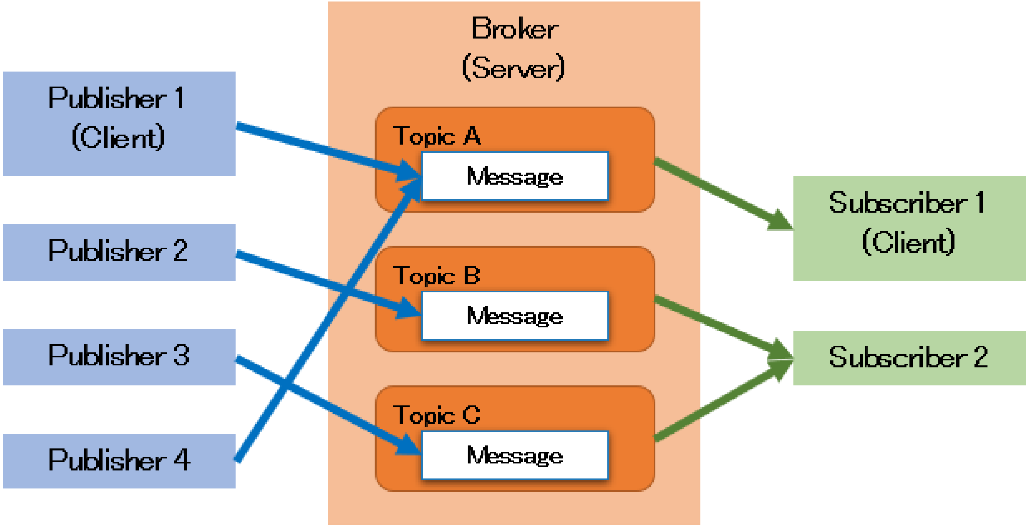
14.8 MQTT Client

14.8.1 What is MQTT?

MQTT stands for Message Queuing Telemetry Transport. It is a simple and lightweight publish/ subscribe messaging protocol.

This protocol allows asynchronous many-to-many communication by a mechanism called “topic” designed to identify messages. Messages are sent and received through an intermediary called a broker server, and thus MQTT enables a device to communicate with another device on the opposite side without being conscious of the opposite device. In addition, since the intermediary is responsible for most of message management, the number of connected client devices can be readily increased. Another feature is that client devices can be freely connected to and disconnected from the broker server. Generally, the amount of information necessary for exchange is small, and this helps to reduce the burden on CPUs and power consumption.

Because of these features, MQTT is widely adopted for IoT applications and is a protocol effective in a system for control among a large number of devices, as well as data logging, traceability, and other communication with a host system.



14.8.2 MQTT Client Specifications

MQTT client specifications with the GM1 controller are described below.

- MQTT Client Specifications

Item	Details
Usable port	LAN ports 1, 2
MQTT protocol version	Version 3.1.1 Version 5.0

Item	Details
Data size	Max. 6000 bytes per packet（payload part Max. 4096 Bytes）(Note 1)
Topic	Topic name：Max.1024 characters Topic level：Max.10
Communication constraints	Max. 20 publishes/subscribes per connection Max. 3 connections
Supported QoS	● QoS0 (publish at most once) ● QoS1 (publish at least once) ● QoS2 (publish exactly once)

(Note 1) This applies to the MQTT protocol packet size out of the total packet.

- List of supported functions

Type	Function	Overview	MQTT Version	
			3.1.1	5.0
Connection	KeepAlive	Specifies an interval during which connection closing is judged	○	○
	Will Message	Specifies a message that is sent when connection is closed	○	○
	Will Retain	Specifies if the Will Message is to be retained	○	○
	Will QoS	Specifies the QoS level for the Will Message	○	○
	Clean Session	Specifies a session used for connection with the broker server	○	○
	User authentication	Connection using a user name and password	○	○
	Client ID	Specifies a client identifier	○	○
	Ping Interval	Specifies an interval at which a ping request (existance check) is sent	○	○
	TLS connection	Connection encrypted by TLS	×	×
	WebSocket connection	Connection using WebSocket	×	×
	Reason Code	An output value of the detailed result of an operation	×	○
	Session Expiry Interval	Specifies how long to retain the session after a disconnect	×	○
	Enhanced authentication	Using other forms of authentication	×	×
	Request Problem Information	Specifies the way an operation result is received	×	○
	Request Response Information	Requests the server to return Response Information (runs on request/response format)	×	×
	Receive Maximum	Specifies the number of messages that the client can process concurrently	×	○
	Topic Alias Maximum	Specifies the number of Topic Aliases that the client can receive	×	×

Type	Function	Overview	MQTT Version	
			3.1.1	5.0
	Maximum Packet Size	Specifies a Maximum Packet Size value	×	○
	Payload Format Indicator	Specifies a format for the Will Message	×	○
	Message Expiry Interval	Specifies an interval for the expiry of the message	×	○
	Content Type	Specifies a type of the content of the Will Message	×	×
	Response Topic	The topic name for a response message (runs on request/response format)	×	○
	Correlation Data	Specifies correlation data (runs on request/response format)	×	○
	Will Delay Interval	Specifies a delay that occurs before the Will Message is sent	×	○
	User Property	User-defined properties	×	○
Publish	Re Delivery	Specifies the re-delivery flag (DUP Flag)	○	○
	Retain	Specifies a message store setting	○	○
	Payload Format Indicator	Specifies a format for the message	×	○
	Message Expiry Interval	Specifies an interval for the expiry of the message	×	○
	Content Type	Specifies a type of the content of the message	×	×
	Response Topic	The topic name for a response message (runs on request/response format)	×	○
	Correlation Data	Specifies correlation data (runs on request/response format)	×	○
	Subscription ID	The identifier of the subscription (for the broker)	×	×
	Topic Alias	Specifies a Topic Alias value	×	○
	User Property	User-defined properties	×	○
Subscribe	Subscription ID	Specifies the identifier of the subscription	×	○
	Correlation Data	Reception of correlation data (runs on request/response format)	×	○
	No Local Option	The setting of reception of messages from the same client	×	○
	Retain As Published	The setting of the Retain flag in a forwarded message	×	×
	RetainHandling	Setting of whether or not to receive retained messages at the time of subscribing	×	○
	User Property	User-defined properties	×	○

14.9 DNS client

14.9.1 What is DNS?

DNS stands for the Domain Name System and refers to a system that manages a mapping between the name of a domain or a host on the network and its IP address. A DNS server has information about mappings between domain names and IP addresses. In response to a query containing a “host name” as a key from a DNS client, the DNS server sends back a corresponding “IP address”.

The GM1 controller can obtain an IP address corresponding to a domain name from a DNS server through an FB of a DNS client.

14.10 NTP client

14.10.1 What is NTP?

NTP stands for the Network Time Protocol and is a communication protocol used to synchronize time between systems in a server/client model. A mechanism for time synchronization operates by allowing an NTP client to send a query about time to a NTP server and obtain a response. Handled time information is defined as the number of seconds that have elapsed since 00:00:00 UTC on January 1 in 1970, and the NTP mechanism allows the synchronization of time in units of milliseconds.

The GM1 controller supports the Simple Network Time Protocol (SNTP), a simplified version of NTP. The GM1 controller can obtain time information from an SNTP server through an FB of SNTP client communication.

15 Other Controller Functions

15.1 SD Card Access Function

15.1.1 Overview of SD Card Access Function

The GM1 controller allows directories and files to be written to and read from the SD card via the CAA File library.

■ Preparation for SD card access

1. Before starting operation using the CAA File library, always use the following function of the Panasonic_GM_System library to check whether the SD card can be accessed.
   a) SYS_GetSDAccessRdy: Reads the mount state of the SD card

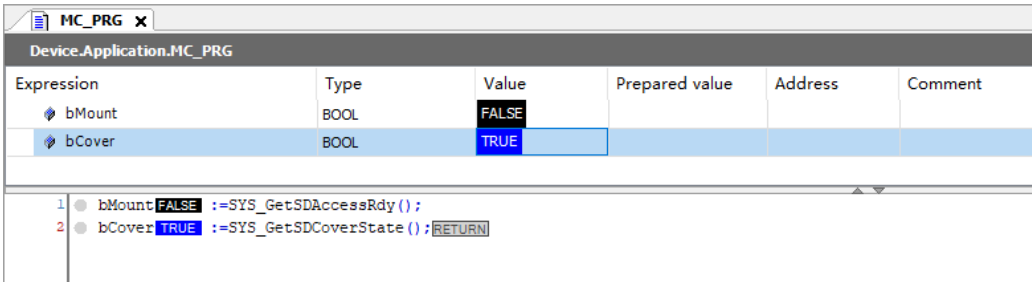
      TRUE: SD card is accessible. An SD card has been inserted.

      FALSE: SD card is inaccessible. No SD card has been inserted.
2. Considering situations such as removal of the SD card during operation, use the following function to check whether the SD card cover is open or closed.
   a) SYS_GetSDCoverState: Reads the open / closed state of the SD card slot cover

      TRUE: The SD card slot cover is closed.

      FALSE: The SD card slot cover is open.

   When the SD card slot cover is open, the SD card can be removed safely by stopping processing such as writing or reading directories or files using the CAA File library.



Device.Application.MC_PRG

Expression	Type	Value	Prepared value	Address	Comment
bMount	BOOL	FALSE			
bCover	BOOL	TRUE			

```
bMount FALSE :=SYS_GetSDAccessRdy();
bCover TRUE  :=SYS_GetSDCoverState();RETURN
```

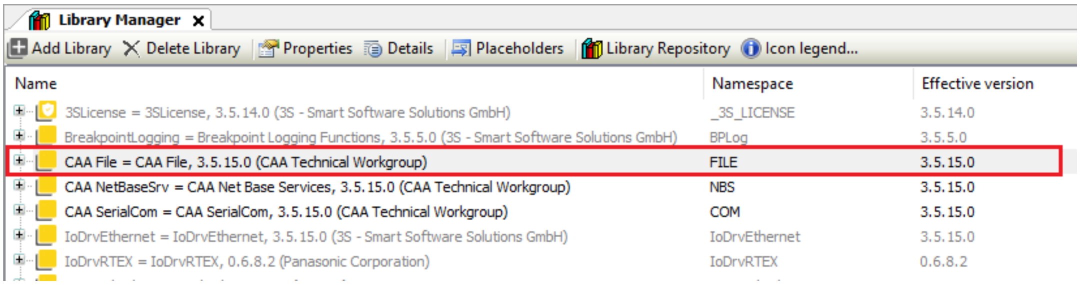
15.1.2 File Manipulations Using the CAA File Library

This section explains how to use the CAA File library to access files on the SD card, in the following order.

1. Library_Manager
2. Example of file write processing
3. Example of file read processing

1. Library_Manager

Check that the following CAA File library is registered in Library_Manager.



Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CAA File = CAA File, 3.5.15.0 (CAA Technical Workgroup)	FILE	3.5.15.0
CAA NetBaseSrv = CAA Net Base Services, 3.5.15.0 (CAA Technical Workgroup)	NBS	3.5.15.0
CAA SerialCom = CAA SerialCom, 3.5.15.0 (CAA Technical Workgroup)	COM	3.5.15.0
IoDrvEthernet = IoDrvEthernet, 3.5.15.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.15.0
IoDrvRTEX = IoDrvRTEX, 0.6.8.2 (Panasonic Corporation)	IoDrvRTEX	0.6.8.2

2. Example of file write processing

SampleDir/SampleFile.txt is created in the SD card and specified data is written to the file.

■ File write processing sequence

The file write processing sequence is shown below.

- File open processing (overwrite mode, insert mode)
  - Overwrite mode: For existing files, the contents of the file are cleared.
  - Insert mode: For existing files, the contents of the file are not cleared.
  - When write is executed, data is written following the end of the previous data.
- File write processing
- File close processing

■ Explanation of variables

uiProcess:

Executes processing when file open processing is set to 1 (overwrite mode) or 2 (insert mode). After the execution is completed, the variable is set to 0 (invalid value).

sFileName:

Specifies a directory or file name.

sWriteData:

Sets data to be written.

bResult:

Substitutes the result of processing execution. (TRUE: Error occurrence, FALSE: Normal termination)

If the result of processing execution is abnormal, check the error code of each processing.

- eOpenResult: Result of file open processing
- eWriteResult: Result of file write processing
- eCloseResult: Result of file close processing

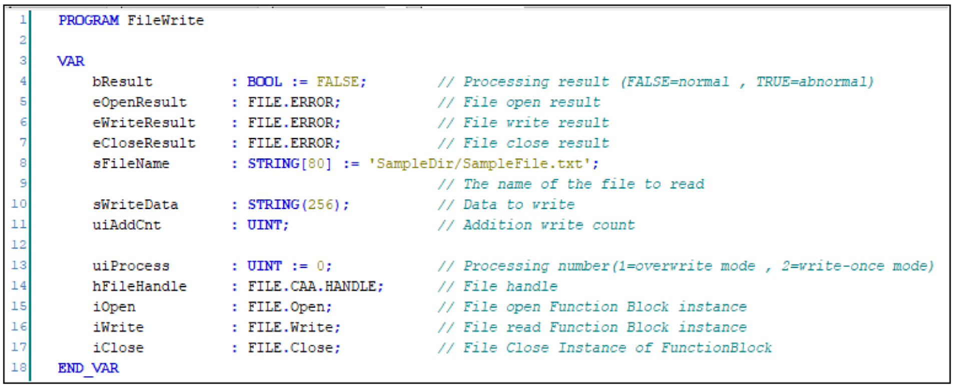
■ Operation example

- In this example, operations are performed by setting the value of uiProcess to 1, 2, and 2 in this order.
- The contents of SampleFile.txt which is output are as follows:

```
NEW_DATA
ADD_DATA1
ADD_DATA2
```

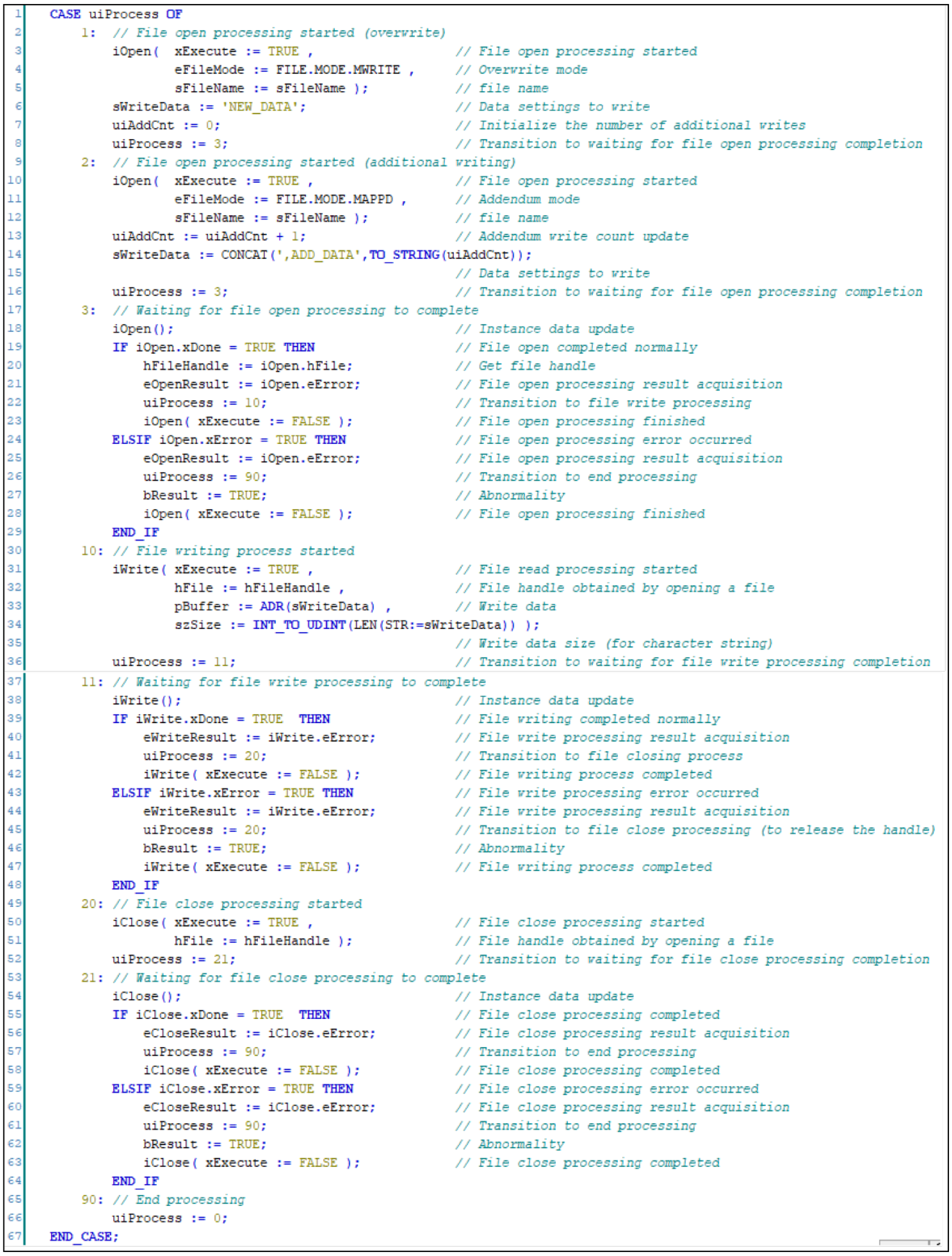
- Sample program

Declaration section (common to ST and LD programming languages)



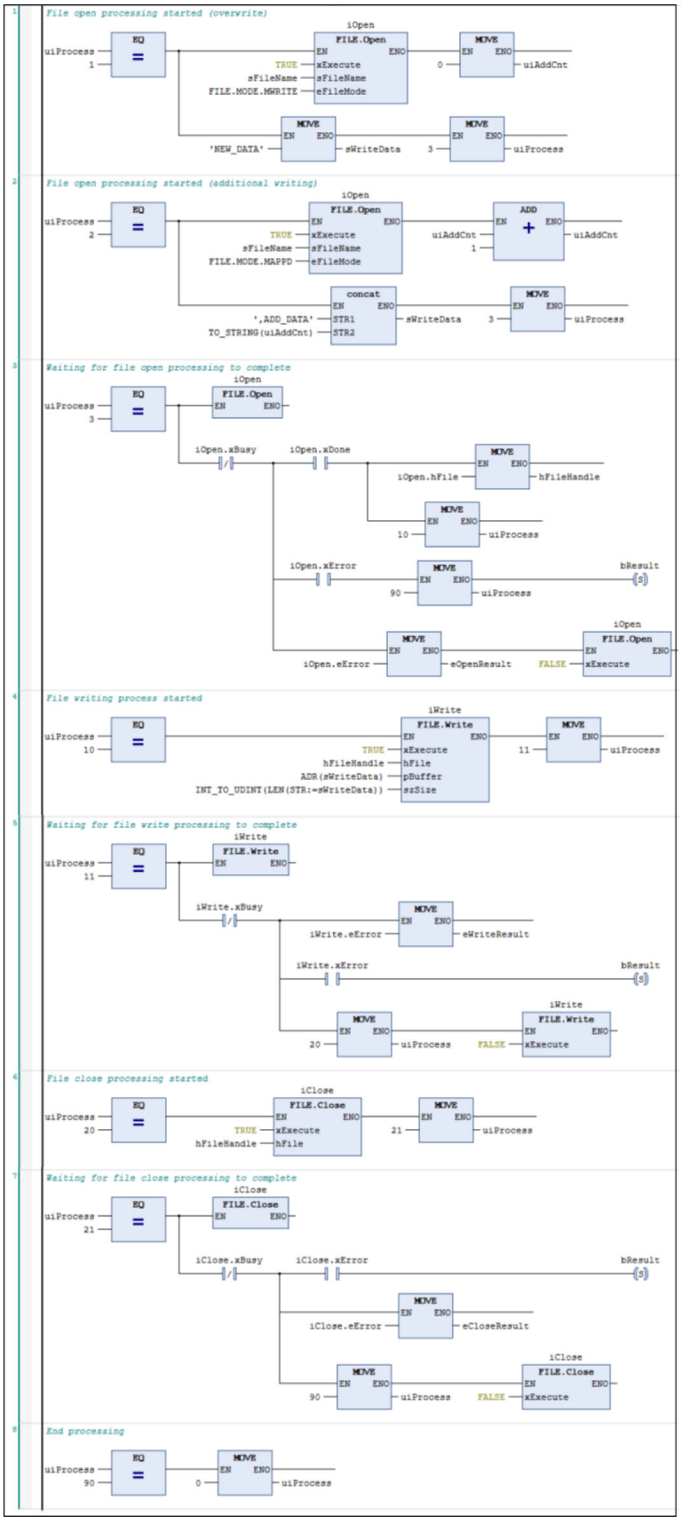
```
PROGRAM FileWrite

VAR
    bResult         : BOOL := FALSE;        // Processing result (FALSE=normal , TRUE=abnormal)
    eOpenResult     : FILE.ERROR;           // File open result
    eWriteResult    : FILE.ERROR;           // File write result
    eCloseResult    : FILE.ERROR;           // File close result
    sFileName       : STRING[80] := 'SampleDir/SampleFile.txt';
                                            // The name of the file to read
    sWriteData      : STRING(256);          // Data to write
    uiAddCnt        : UINT;                 // Addition write count

    uiProcess       : UINT := 0;            // Processing number(1=overwrite mode , 2=write-once mode)
    hFileHandle     : FILE.CAA.HANDLE;      // File handle
    iOpen           : FILE.Open;            // File open Function Block instance
    iWrite          : FILE.Write;           // File read Function Block instance
    iClose          : FILE.Close;           // File Close Instance of FunctionBlock
END_VAR
```

Implementation section (ST programming language)



```
CASE uiProcess OF
    1:  // File open processing started (overwrite)
        iOpen(  xExecute := TRUE ,                  // File open processing started
                eFileMode := FILE.MODE.MWRITE ,     // Overwrite mode
                sFileName := sFileName );           // file name
        sWriteData := 'NEW_DATA';                   // Data settings to write
        uiAddCnt := 0;                              // Initialize the number of additional writes
        uiProcess := 3;                             // Transition to waiting for file open processing completion
    2:  // File open processing started (additional writing)
        iOpen(  xExecute := TRUE ,                  // File open processing started
                eFileMode := FILE.MODE.MAPPD ,      // Addendum mode
                sFileName := sFileName );           // file name
        uiAddCnt := uiAddCnt + 1;                   // Addendum write count update
        sWriteData := CONCAT(',ADD_DATA',TO_STRING(uiAddCnt));
                                                    // Data settings to write
        uiProcess := 3;                             // Transition to waiting for file open processing completion
    3:  // Waiting for file open processing to complete
        iOpen();                                    // Instance data update
        IF iOpen.xDone = TRUE THEN                  // File open completed normally
            hFileHandle := iOpen.hFile;             // Get file handle
            eOpenResult := iOpen.eError;            // File open processing result acquisition
            uiProcess := 10;                        // Transition to file write processing
            iOpen( xExecute := FALSE );             // File open processing finished
        ELSIF iOpen.xError = TRUE THEN              // File open processing error occurred
            eOpenResult := iOpen.eError;            // File open processing result acquisition
            uiProcess := 90;                        // Transition to end processing
            bResult := TRUE;                        // Abnormality
            iOpen( xExecute := FALSE );             // File open processing finished
        END_IF
    10: // File writing process started
        iWrite( xExecute := TRUE ,                  // File read processing started
                hFile := hFileHandle ,              // File handle obtained by opening a file
                pBuffer := ADR(sWriteData) ,        // Write data
                szSize := INT_TO_UDINT(LEN(STR:=sWriteData)) );
                                                    // Write data size (for character string)
        uiProcess := 11;                            // Transition to waiting for file write processing completion
    11: // Waiting for file write processing to complete
        iWrite();                                   // Instance data update
        IF iWrite.xDone = TRUE  THEN                // File writing completed normally
            eWriteResult := iWrite.eError;          // File write processing result acquisition
            uiProcess := 20;                        // Transition to file closing process
            iWrite( xExecute := FALSE );            // File writing process completed
        ELSIF iWrite.xError = TRUE THEN             // File write processing error occurred
            eWriteResult := iWrite.eError;          // File write processing result acquisition
            uiProcess := 20;                        // Transition to file close processing (to release the handle)
            bResult := TRUE;                        // Abnormality
            iWrite( xExecute := FALSE );            // File writing process completed
        END_IF
    20: // File close processing started
        iClose( xExecute := TRUE ,                  // File close processing started
                hFile := hFileHandle );             // File handle obtained by opening a file
        uiProcess := 21;                            // Transition to waiting for file close processing completion
    21: // Waiting for file close processing to complete
        iClose();                                   // Instance data update
        IF iClose.xDone = TRUE  THEN                // File close processing completed
            eCloseResult := iClose.eError;          // File close processing result acquisition
            uiProcess := 90;                        // Transition to end processing
            iClose( xExecute := FALSE );            // File close processing completed
        ELSIF iClose.xError = TRUE THEN             // File close processing error occurred
            eCloseResult := iClose.eError;          // File close processing result acquisition
            uiProcess := 90;                        // Transition to end processing
            bResult := TRUE;                        // Abnormality
            iClose( xExecute := FALSE );            // File close processing completed
        END_IF
    90: // End processing
        uiProcess := 0;
END_CASE;
```

Implementation section (LD programming language)



3. Example of file read processing

Data in SampleDir/SampleFile.txt in the SD card is read into the buffer.

The effective range of data read into the buffer is judged from the data size information that is output after read processing.

■ File read processing sequence

The file read processing sequence is shown below.

- File open processing (read mode)
- File read processing
- File close processing

■ Explanation of variables

uiProcess:

Executes processing when the variable is set to 1 (read mode).

sFileName:

Specifies a directory or file name.

sReadData:

Sets a buffer into which data is to be read.

szReadSize:

Stores the size of read data after read processing.

bResult:

Substitutes the result of processing execution. (TRUE: Error occurrence, FALSE: Normal termination)

If the result of processing execution is abnormal, check the error code of each processing.

- eOpenResult: Result of file open processing
- eReadResult: Result of file read processing
- eCloseResult: Result of file close processing

■ Operation example

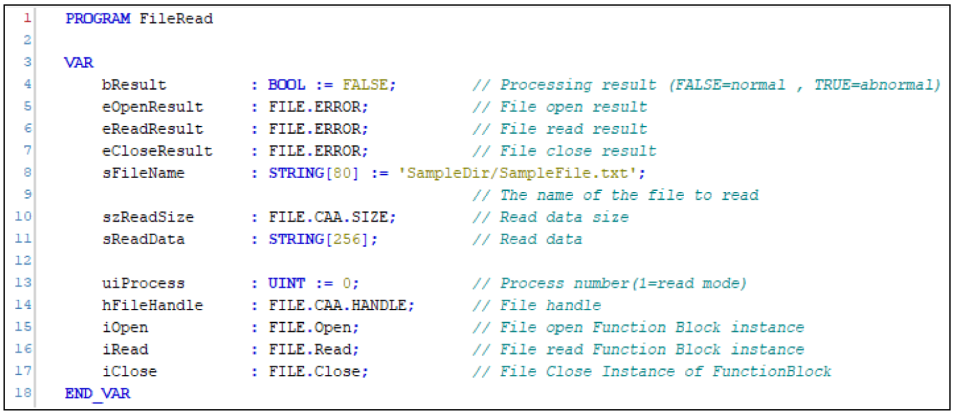
- In this example, operations are performed according to the following contents of SampleFile.txt.

  NEW_DATA
  ADD_DATA1
  ADD_DATA2
- Read data and data size are as below.

  Data (STRING type): 'NEW_DATA$R$NADD_DATA1$R$NADD_DATA2$R$NADD_DATA3'

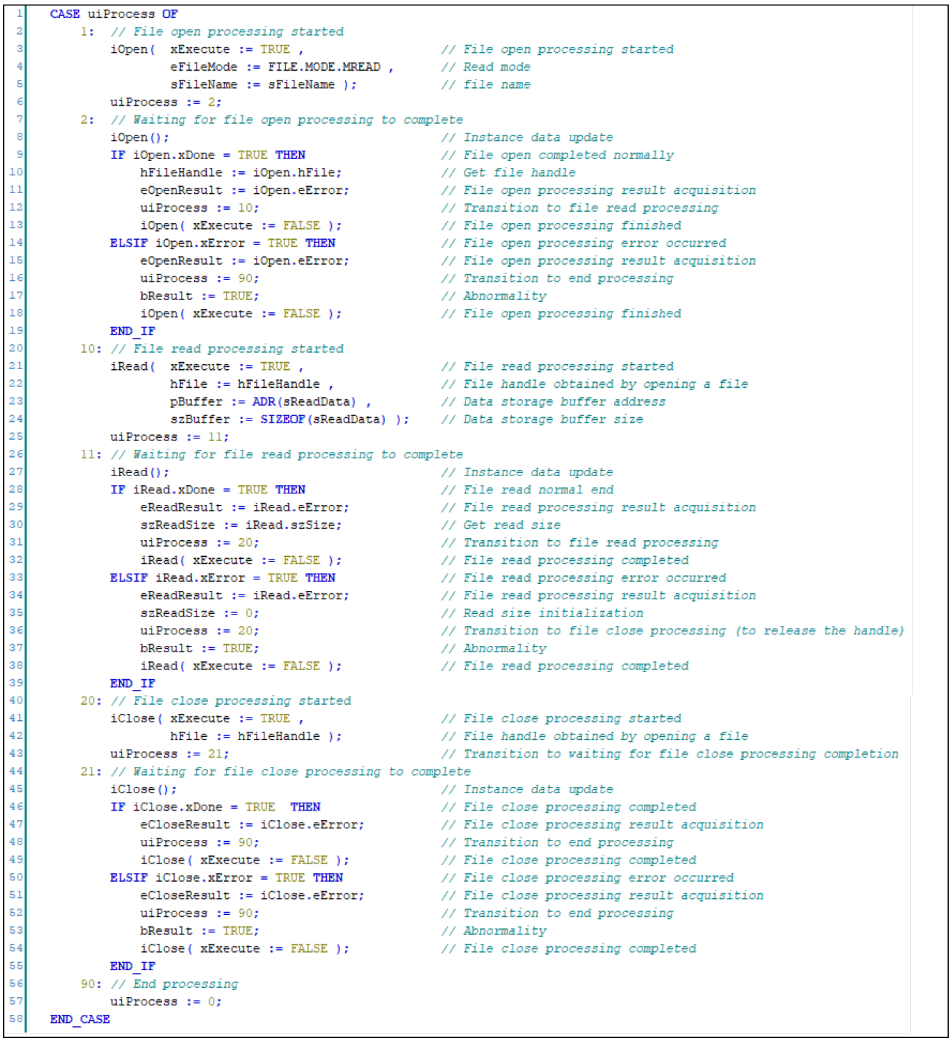
  Data size: 41
- Sample program

Declaration section (common to ST and LD programming languages)



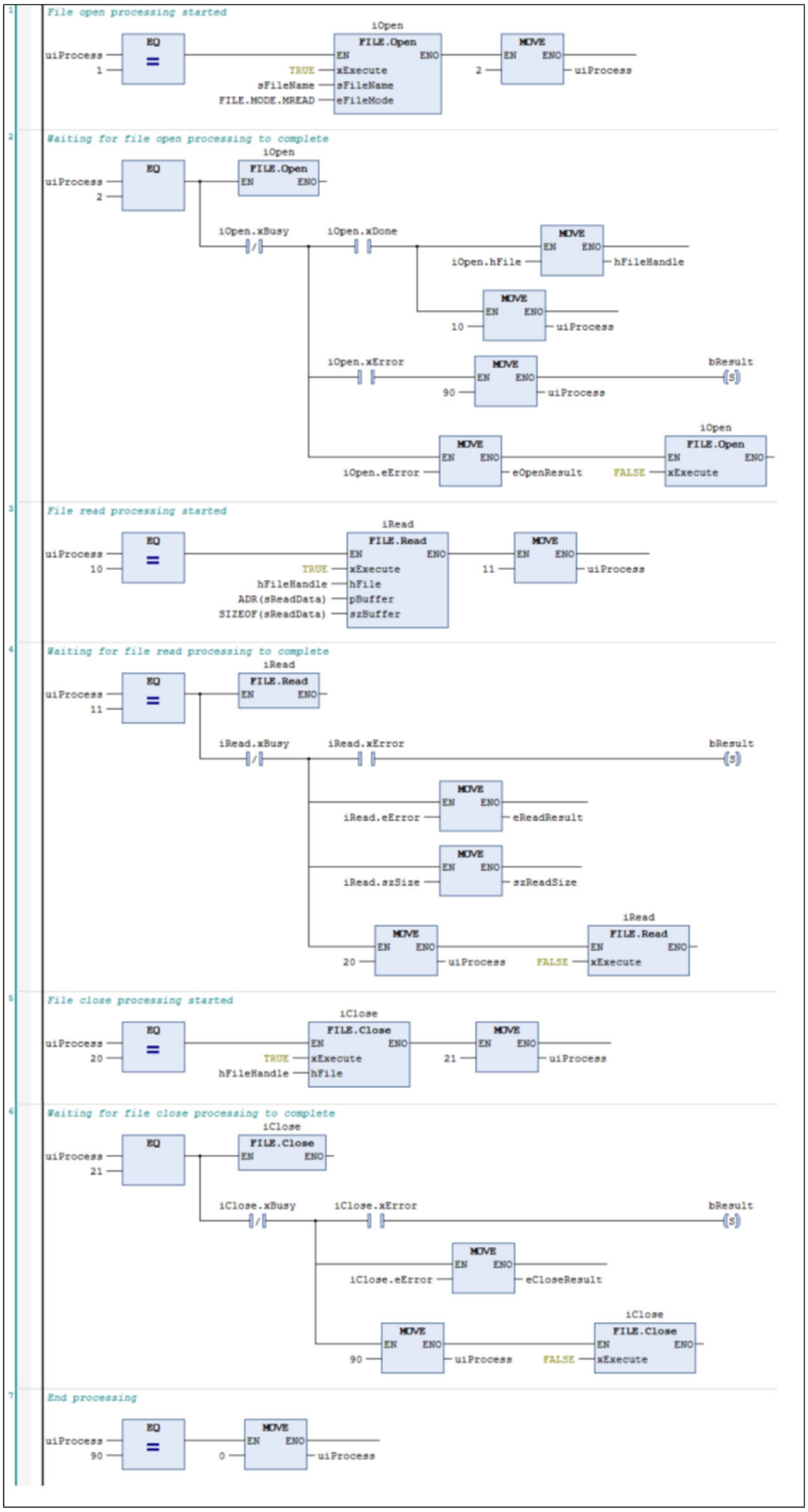
```
PROGRAM FileRead

VAR
    bResult         : BOOL := FALSE;        // Processing result (FALSE=normal , TRUE=abnormal)
    eOpenResult     : FILE.ERROR;           // File open result
    eReadResult     : FILE.ERROR;           // File read result
    eCloseResult    : FILE.ERROR;           // File close result
    sFileName       : STRING[80] := 'SampleDir/SampleFile.txt';
                                            // The name of the file to read
    szReadSize      : FILE.CAA.SIZE;        // Read data size
    sReadData       : STRING[256];          // Read data

    uiProcess       : UINT := 0;            // Process number(1=read mode)
    hFileHandle     : FILE.CAA.HANDLE;      // File handle
    iOpen           : FILE.Open;            // File open Function Block instance
    iRead           : FILE.Read;            // File read Function Block instance
    iClose          : FILE.Close;           // File Close Instance of FunctionBlock
END_VAR
```

Implementation section (ST programming language)



```
CASE uiProcess OF
    1:  // File open processing started
        iOpen(  xExecute := TRUE ,                      // File open processing started
                eFileMode := FILE.MODE.MREAD ,          // Read mode
                sFileName := sFileName );               // file name
        uiProcess := 2;
    2:  // Waiting for file open processing to complete
        iOpen();                                        // Instance data update
        IF iOpen.xDone = TRUE THEN                      // File open completed normally
            hFileHandle := iOpen.hFile;                 // Get file handle
            eOpenResult := iOpen.eError;                // File open processing result acquisition
            uiProcess := 10;                            // Transition to file read processing
            iOpen( xExecute := FALSE );                 // File open processing finished
        ELSIF iOpen.xError = TRUE THEN                  // File open processing error occurred
            eOpenResult := iOpen.eError;                // File open processing result acquisition
            uiProcess := 90;                            // Transition to end processing
            bResult := TRUE;                            // Abnormality
            iOpen( xExecute := FALSE );                 // File open processing finished
        END_IF
    10: // File read processing started
        iRead(  xExecute := TRUE ,                      // File read processing started
                hFile := hFileHandle ,                  // File handle obtained by opening a file
                pBuffer := ADR(sReadData) ,             // Data storage buffer address
                szBuffer := SIZEOF(sReadData) );        // Data storage buffer size
        uiProcess := 11;
    11: // Waiting for file read processing to complete
        iRead();                                        // Instance data update
        IF iRead.xDone = TRUE THEN                      // File read normal end
            eReadResult := iRead.eError;                // File read processing result acquisition
            szReadSize := iRead.szSize;                 // Get read size
            uiProcess := 20;                            // Transition to file read processing
            iRead( xExecute := FALSE );                 // File read processing completed
        ELSIF iRead.xError = TRUE THEN                  // File read processing error occurred
            eReadResult := iRead.eError;                // File read processing result acquisition
            szReadSize := 0;                            // Read size initialization
            uiProcess := 20;                            // Transition to file close processing (to release the handle)
            bResult := TRUE;                            // Abnormality
            iRead( xExecute := FALSE );                 // File read processing completed
        END_IF
    20: // File close processing started
        iClose( xExecute := TRUE ,                      // File close processing started
                hFile := hFileHandle );                 // File handle obtained by opening a file
        uiProcess := 21;                                // Transition to waiting for file close processing completion
    21: // Waiting for file close processing to complete
        iClose();                                       // Instance data update
        IF iClose.xDone = TRUE  THEN                    // File close processing completed
            eCloseResult := iClose.eError;              // File close processing result acquisition
            uiProcess := 90;                            // Transition to end processing
            iClose( xExecute := FALSE );                // File close processing completed
        ELSIF iClose.xError = TRUE THEN                 // File close processing error occurred
            eCloseResult := iClose.eError;              // File close processing result acquisition
            uiProcess := 90;                            // Transition to end processing
            bResult := TRUE;                            // Abnormality
            iClose( xExecute := FALSE );                // File close processing completed
        END_IF
    90: // End processing
        uiProcess := 0;
END_CASE
```

Implementation section (LD programming language)



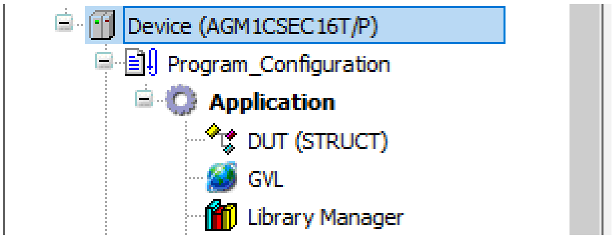
15.2 Time Function

15.2.1 Overview of Time Function

This section explains the time function that uses date and time settings in GM Programmer and function blocks.

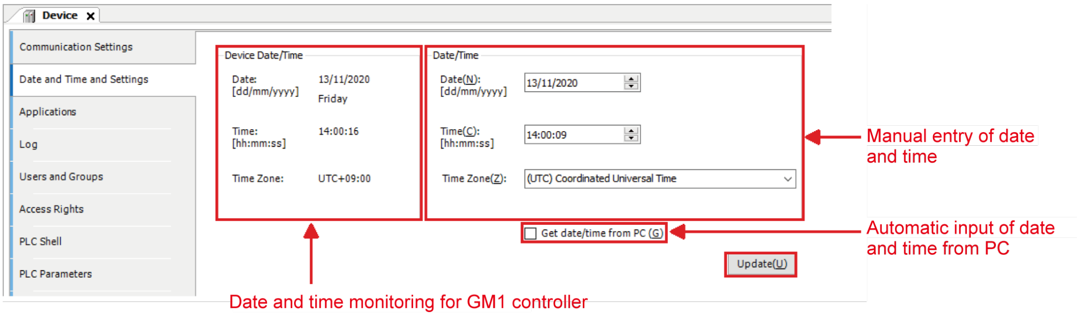
15.2.2 Settings Based on GM Programmer

Date and time settings and monitoring for the GM1 controller can be performed using the "Date and Time Settings" window of GM Programmer. Connect to the GM1 controller and select the "Date and Time Settings" tab of the "Device" window.



Device (AGM1CSEC16T/P)
Program_Configuration
Application
DUT (STRUCT)
GVL
Library Manager

Date and Time Settings pane



Device ×

Communication Settings
Date and Time and Settings
Applications
Log
Users and Groups
Access Rights
PLC Shell
PLC Parameters

Device Date/Time
Date: [dd/mm/yyyy] 13/11/2020 Friday
Time: [hh:mm:ss] 14:00:16
Time Zone: UTC+09:00

Date/Time
Date(N): [dd/mm/yyyy] 13/11/2020
Time(C): [hh:mm:ss] 14:00:09
Time Zone(Z): (UTC) Coordinated Universal Time

□ Get date/time from PC (G)

Update(U)

Manual entry of date and time

Automatic input of date and time from PC

Date and time monitoring for GM1 controller

The "Device Date and Time" frame on the left side of the pane displays the date / time information of the GM1 controller. To set date and time, enter date and time in the "Date and Time" frame on the right side of the pane and click the [Update] button. The entered date and time will be set in the GM1 controller.

If the "Get date / time from PC" check box is selected, the date/time information of the PC will be automatically set in the GM1 controller. In this case, the time zone of the PC will be automatically set in the GM1 controller.

15.2.3 Settings Based on Function Blocks

With the GM1 controller, you can obtain and set the current time, obtain and set a time zone, and obtain the SNTP time.

For details on how to use this function, refer to the *GM1 Series Reference Manual (Instruction Edition)*.

15.3 Trace Function

The trace function allows variable value changes in the GM1 controller over time to be displayed in graph form and be checked. Thus, you can check a motion operation locus and log whether a process or communication computed at high speeds is properly dealt with. You can save obtained trace data in an external file and check such data by an external tool. Saved files can also be read offline.

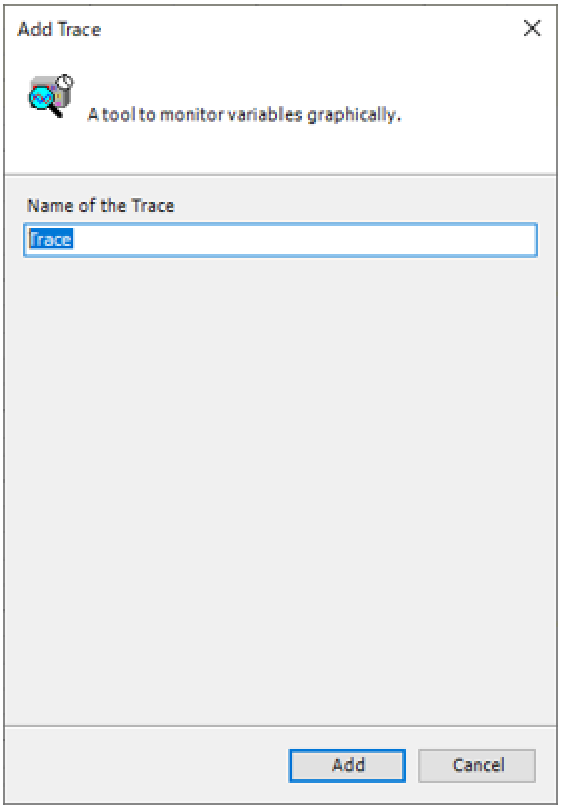
15.3.1 Setting up Trace

To use the trace function, after adding a trace object, you must register variables to be traced. This section describes a procedure for setting up a trace.

1 2 Procedure

1. Right-click Application in the navigator pane and then select “Add Object>Trace” from the menu that is displayed.

   The "Add Trace" dialog box will be displayed.



Add Trace

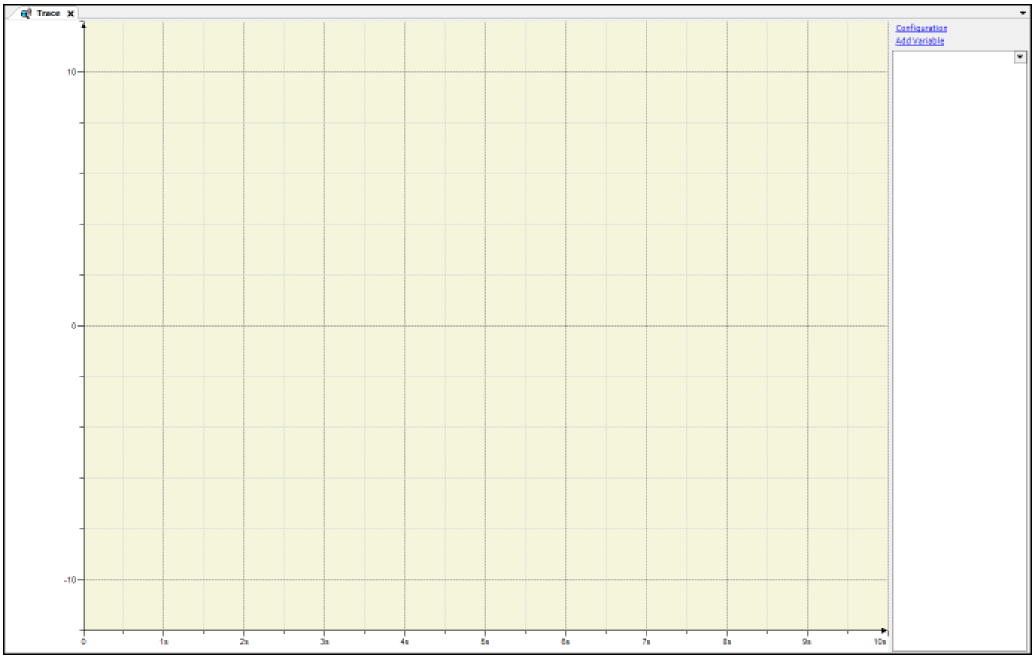
A tool to monitor variables graphically.

Name of the Trace

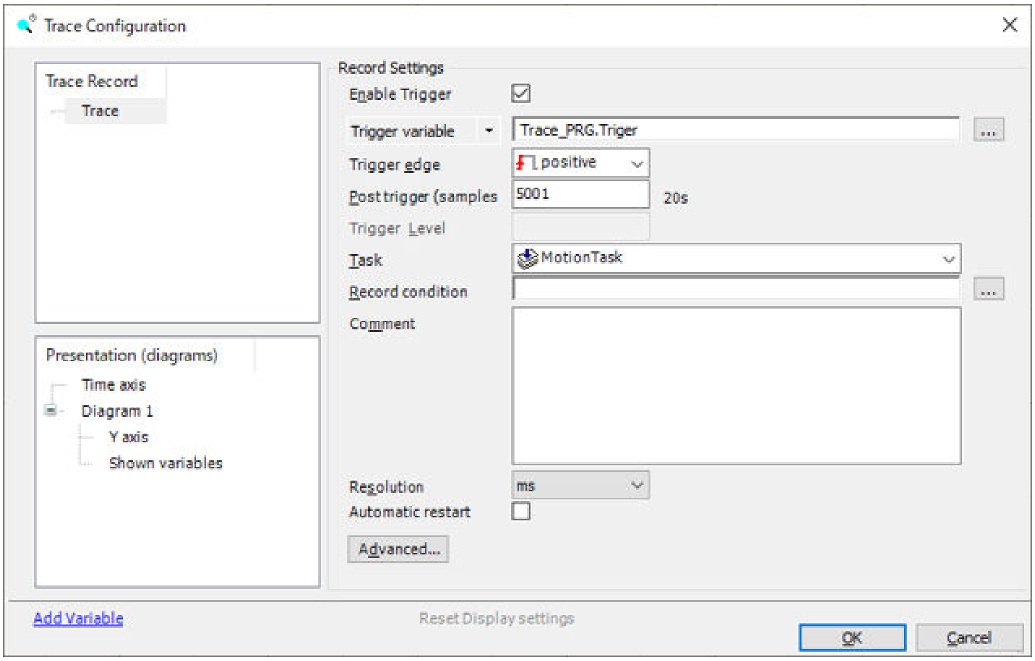
Trace

2. Enter a desired trace name and click the “Add” button.

   When the trace is added, a trace object screen appears.



3. Configure trace settings.
Click "Configuration" at the upper right of the object screen to display the settings pane.
Configure record settings shown in the table below and after that, click the OK button.



Trace Configuration

Trace Record
- Trace

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables

Record Settings

Enable Trigger ☑

Trigger variable: Trace_PRG.Triger

Trigger edge: positive

Post trigger (samples: 5001 20s

Trigger Level:

Task: MotionTask

Record condition:

Comment:

Resolution: ms

Automatic restart □

- Record settings

Trigger and record time settings can be primarily configured.

Setting items	Overview
Enable Trigger	Trigger is enabled. If trigger is not enabled, data is always sampled after the start of a trace, and old data is deleted depending on the number of samples.
Trigger variable / parameter	Set a variable that constitutes a trigger condition. When the variable is an integer or a real number other than Booleans, you must set a trigger level.
Trigger edge	Set a trigger detection edge. ● positive : Detect a rising edge ● negative : Detect a falling edge ● both : Detect both rising / falling edges
Posttrigger (samples)	Set the number of samples recored after detection of a trigger. Record time depending on the number of samples set at the right side is displayed.
Trigger Level	Set a value at which an edge is detected according to the variable set in Trigger edge.
Task (Note 1)	Select a task from MotionTask and a created UserTask. The cycle of the selected task is a cycle of the trace.
Record condition	Set a condition for executing a trace. A trace is executed only when the set condition is met. Description example: Trace_Record (Boolean type) = TRUE, Axis.Velocity (Integer type) = 100
Comment	Write a comment. If data is saved in a trace file and a trace dump file, details of the comment are also output.
Resolution	Set trace time resolution. You can select us or ms. (Note 2)
Automatic restart	Do not use.
Advanced settings	Details will be described later.

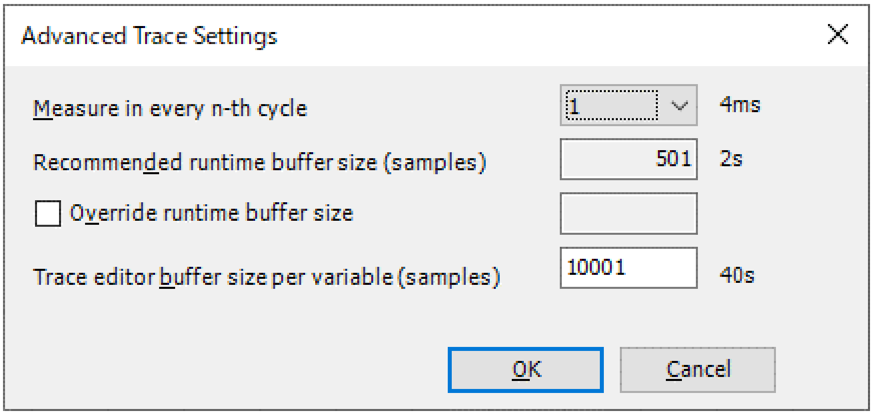
(Note 1) To start the trace function, you must set a task.

(Note 2) You must set the trace resolution that is smaller than the task cycle. If the task cycle is 1 ms or less, you must set the trace resolution to us.

- Advanced trace settings

The number of measurement cycles and a buffer size can be specified. Click the [Advanced...] button to display the setting window.

<EtherCAT>



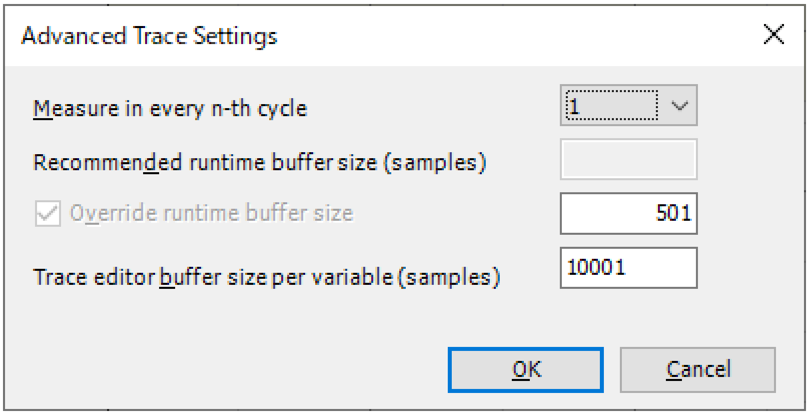
Measure in every n-th cycle: 1 4ms

Recommended runtime buffer size (samples): 501 2s

□ Override runtime buffer size:

Trace editor buffer size per variable (samples): 10001 40s

<RTEX>



Measure in every n-th cycle: 1

Recommended runtime buffer size (samples):

☑ Override runtime buffer size: 501

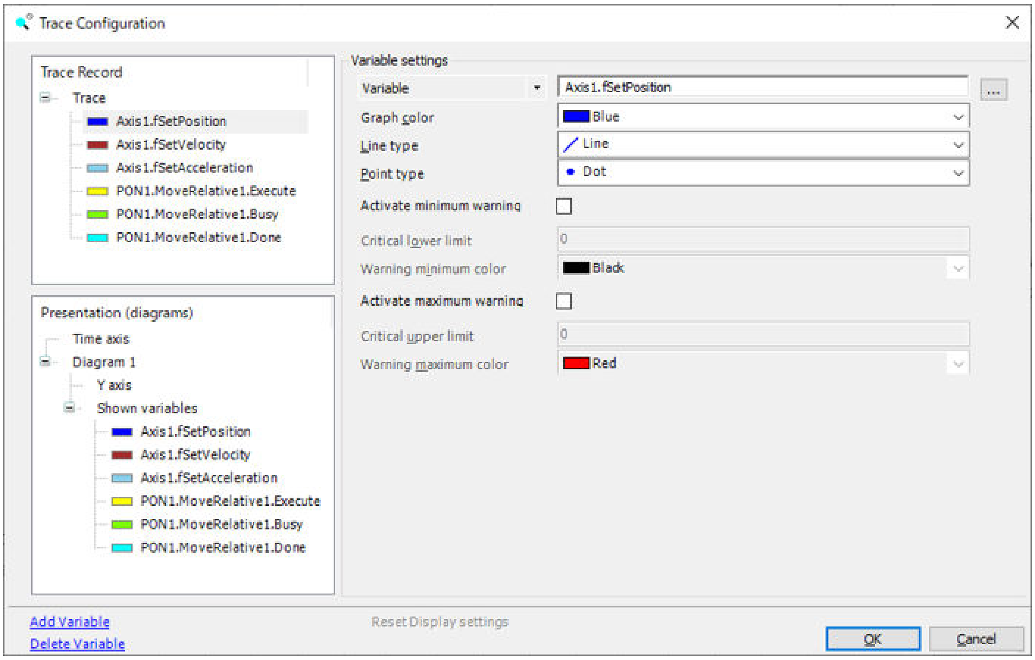
Trace editor buffer size per variable (samples): 10001

Setting items	Overview
Measurement frequency (number of cycles)	Set a sampling interval. (Note 3) Sampling interval = Number of cycles × task cycle Settable number of cycles: 1 to 20, 100, 200, 500, 1000
Recommended runtime buffer size (sample)	A value is automatically specified to suit the set measurement frequency (number of cycles). This setting is not displayed in RTEX. (Note 3)
Override runtime buffer size	You can change the runtime buffer size. When the check box is not selected, the recommended runtime buffer size is adapted. (Note 3)
Trace editor buffer size for each variable (sample)	Set the maximum sampling number for each variable. When the maximum sampling number is exceeded, pieces of the sampling data are deleted in the order of their occurrence. (Note 3) (Note 4)

(Note 1) In RTEX, record time depending on the set number of samples is not displayed.

(Note 2) You must set this buffer size to at least twice the runtime buffer size.

4. Next, register a variable to be traced.
   Click "Add variable" at the upper right of the object screen to display the Variable settings pane.



Trace Configuration

Trace Record
- Trace
  - 🟦 Axis1.fSetPosition
  - 🟥 Axis1.fSetVelocity
  - 🟦 Axis1.fSetAcceleration
  - 🟨 PON1.MoveRelative1.Execute
  - 🟩 PON1.MoveRelative1.Busy
  - 🟦 PON1.MoveRelative1.Done

Presentation (diagrams)
- Time axis
- Diagram 1
  - Y axis
  - Shown variables
    - 🟦 Axis1.fSetPosition
    - 🟥 Axis1.fSetVelocity
    - 🟦 Axis1.fSetAcceleration
    - 🟨 PON1.MoveRelative1.Execute
    - 🟩 PON1.MoveRelative1.Busy
    - 🟦 PON1.MoveRelative1.Done

Variable settings

Variable	Axis1.fSetPosition
Graph color	Blue
Line type	Line
Point type	Dot
Activate minimum warning	□
Critical lower limit	0
Warning minimum color	Black
Activate maximum warning	□
Critical upper limit	0
Warning maximum color	Red

5. Enter a variable you want to trace and set a line color, type, and other items.

Either directly enter a variable or click the ... button to select a desired variable from the Input Assistant window. The registered variables are displayed in the Trace Record pane at the upper left.

- Variable settings

You can set a graph color and line/point types for the variable.

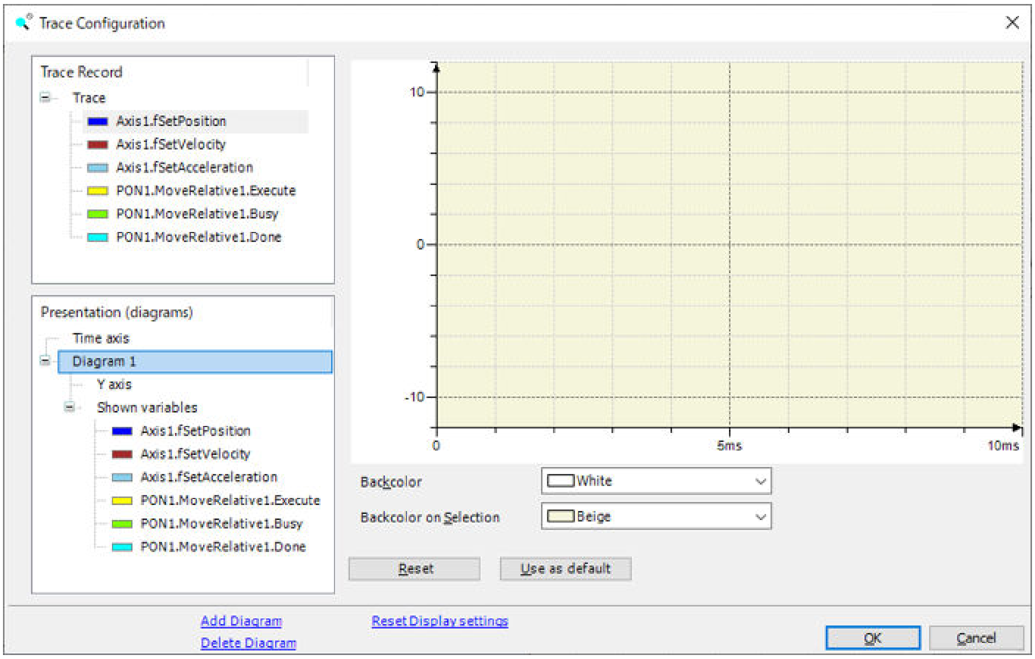
Setting items	Overview
Variable / Parameter	Specify a variable or parameter to be traced.
Graph color	Set the line color of the graph.
Line type	Set the line type of the graph.
Point type	Set the point type of the graph.
Activate minimum warning	When the trace value is smaller than the critical lower limit, the graph is displayed in warning minimum color.
Critical lower limit	Set the lower limit for minimum warning.
Warning minimum color	Set the color of the graph for minimum warning.
Activate maximum warning	When the trace value is larger than the critical upper limit, the graph is displayed in warning maximum color.
Critical upper limit	Set the upper limit for maximum warning.
Warning maximum color	Set the color of the graph for maximum warning.

6. Configure diagram settings.

From the Presentation (diagrams) pane at the lower left, you can configure diagram settings. Configure settings to add a diagram or assign a variable to be traced.

- Diagram settings

You can add a diagram and set a background color. Click “Diagram” to display the settings pane on the right side. For two or more diagrams, settings of each diagram can be configured.



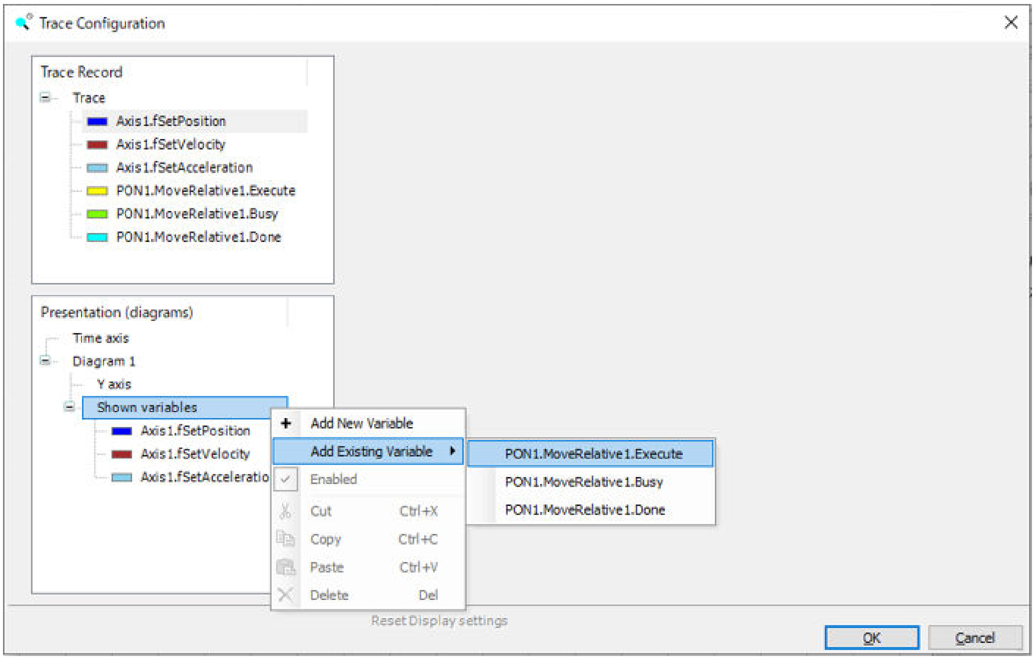
Backcolor: White

Backcolor on Selection: Beige

Setting items	Overview
Add Diagram	Add one diagram you want to display on the trace screen.
Delete Diagram	Delete the specified diagram.
Reset Display settings	All display settings return to default values.
Backcolor	Set the background color of the diagram.
Backcolor on Selection	Set the background color of the diagram that is selected (active).
Reset	Display settings of the diagram return to default values.
Use as default	Default display settings of diagrams change to the current settings.

- Setting of assigning variable to be displayed on diagram

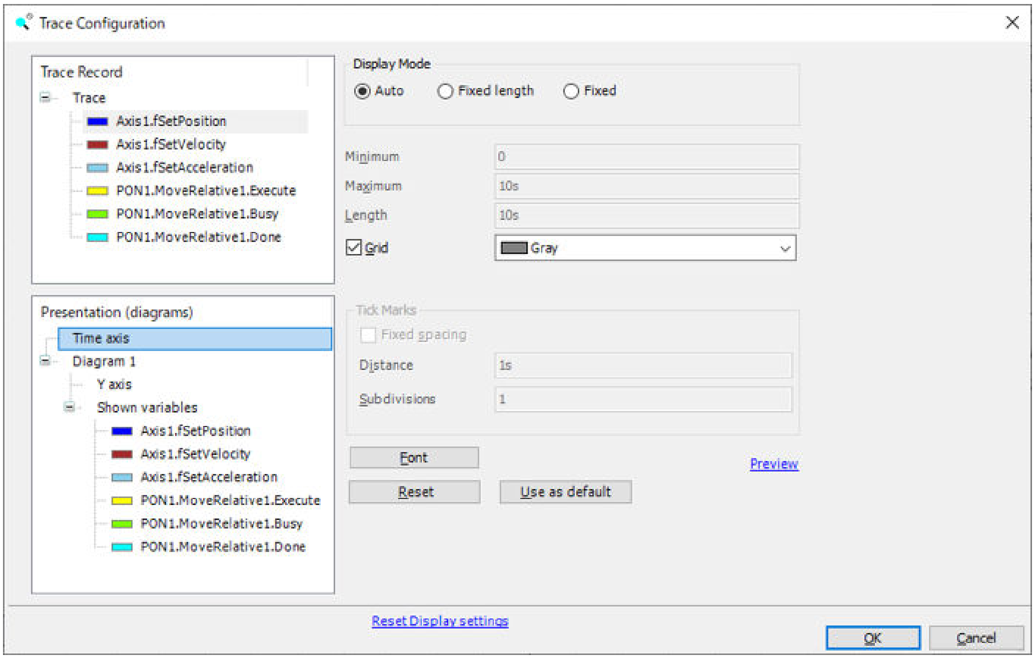
Assign a variable to be displayed on the diagram. Right-click “Shown variables” to display a menu for variable assignment.



Setting items	Overview
Add new variable	A new variable to be traced is registered and added to the diagram.
Add existing variable	Select a registered variable to add it to the diagram.

- Time axis settings

Click “Time axis” in the Presentation pane to display the settings pane on the right side.



Display Mode

● Auto ○ Fixed length ○ Fixed

Minimum 0

Maximum 10s

Length 10s

☑ Grid Gray

Tick Marks

□ Fixed spacing

Distance 1s

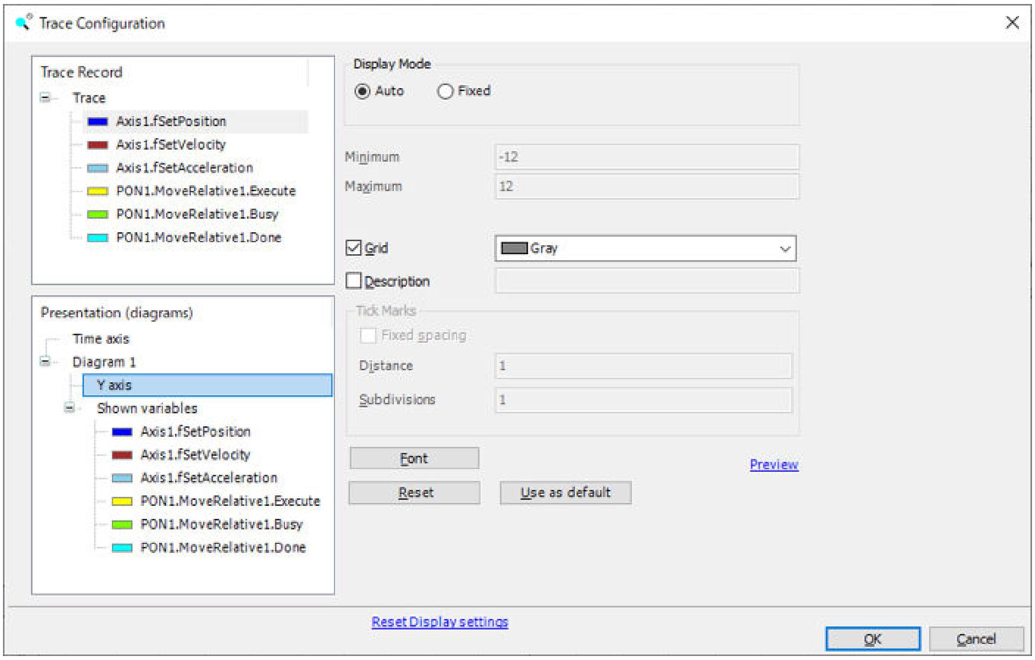
Subdivisions 1

Setting items		Overview
Display Mode		Set the time axis display mode. ● Auto: Time axis tick marks suitable for the trace are automatically displayed. ● Fixed length: The display width of the time axis is fixed to a value set in “Length” and the time axis is displayed by automatic scrolling. ● Fixed: The time axis is displayed in a range between points that are fixed to values set in “Minimum” and “Maximum”.
Grid		A grid that fits tick marks is displayed. Also set the color of the grid.
Tick Marks	Fixed spacing	The space between the tick mark and auxiliary tick mark can be set. (Note 5)
	Distance	Set the space between tick marks in units of time.
	Subdivisions	Sets the number of auxiliary tick mark.
Font		Set the display font of the time axis.
Preview		Display a preview.
Result		Time axis settings return to default values.
Use as default		Default time axis settings change to the current settings.

(Note 1) The auto display mode cannot be specified.

- Y axis settings

Click “Y axis” to display the settings pane on the right side.



Trace Configuration

Trace Record

Trace

- Axis1.fSetPosition
- Axis1.fSetVelocity
- Axis1.fSetAcceleration
- PON1.MoveRelative1.Execute
- PON1.MoveRelative1.Busy
- PON1.MoveRelative1.Done

Presentation (diagrams)

Time axis

Diagram 1

Y axis

Shown variables

- Axis1.fSetPosition
- Axis1.fSetVelocity
- Axis1.fSetAcceleration
- PON1.MoveRelative1.Execute
- PON1.MoveRelative1.Busy
- PON1.MoveRelative1.Done

Display Mode: ● Auto ○ Fixed

Minimum: -12

Maximum: 12

☑ Grid: Gray

□ Description

Tick Marks

□ Fixed spacing

Distance: 1

Subdivisions: 1

Setting items	Overview
Display mode	Set the Y axis display mode. ● Auto: Y axis tick marks suitable for the trace are automatically displayed.

Setting items		Overview
		● Fixed: The Y axis is displayed in a range between points that are fixed to values set in “Minimum” and “Maximum”.
Grid		A grid that fits tick marks is displayed. Also set the color of the grid.
Description		Display a label on the Y axis.
Tick Marks	Fixed spacing	The space between the tick mark and auxiliary tick mark can be set. (Note 6)
	Distance	Set the space between tick marks in units of time.
	Subdivisions	Sets the number of auxiliary tick mark.
Font		Set the display font of the Y axis.
Preview		Display a preview.
Result		Y axis settings return to default values.
Use as default		Default Y axis settings change to the current settings.

(Note 1) The auto display mode cannot be specified.

7. After the settings are configured, click the “OK” button.

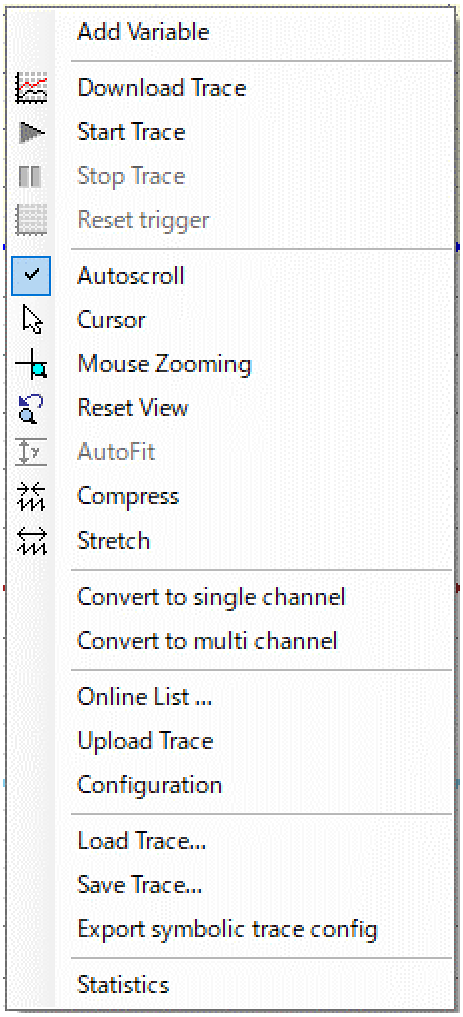
This completes the procedure for registering variables to be traced and configuring trace settings.

15.3.2 Trace Menu

The trace menu allows you to perform actions, such as operating a trace or the display screen and saving a file, on the trace object.

■ Trace menu

To display the trace menu, select Trace from the menu bar or right-click the trace screen with the mouse cursor put on it.



Setting items	Overview
Add Variable	The Variable settings pane (Trace Configuration) appears.
Download Trace	A trace is downloaded and starts.
Start Trace	The stopped trace starts again.
Stop Trace	Stop the active trace.

Setting items	Overview
Reset trigger	Reset the trigger in progress and put the trace in a state of waiting for trigger again.
Autoscroll	The trace screen automatically scrolls in response to the latest trace data.
Cursor	A maximum of two cursors can be displayed. A trace value at the cursor place is displayed.
Mouse Zooming	The mouse pointer changes to a zoom cursor to allow a selected trace part to be enlarged.
Reset View	Trace object display settings return to default values.
AutoFit	The Y axis tick marks of the diagram are automatically optimized.
Compress	Shorten the time (X) axis of the diagram.
Stretch	Lengthen the time (X) axis of the diagram.
Convert to single channel	Display all variables registered in Trace Record on one diagram.
Convert to multi channel	Display all variables registered in Trace Record on separate diagrams.
Online List	Display a list of variables saved in the runtime buffer.
Upload Trace	Display trace data saved in the runtime buffer.
Configuration	The Trace Configuration window appears.
Load Trace	A trace file (.trace / .trace.csv) is imported and displayed.
Save Trace	The trace is saved as a trace file (.trace / .txt / .trace.csv).
Export symbolic trace config	Not supported. Do not use.
Statistics	Display trace statistics. Display the upper limit, lower limit, mean value, median, root-mean-square (RMS), and suchlike of trace data by variable.

Info.

● Diagram reduction/enlargement

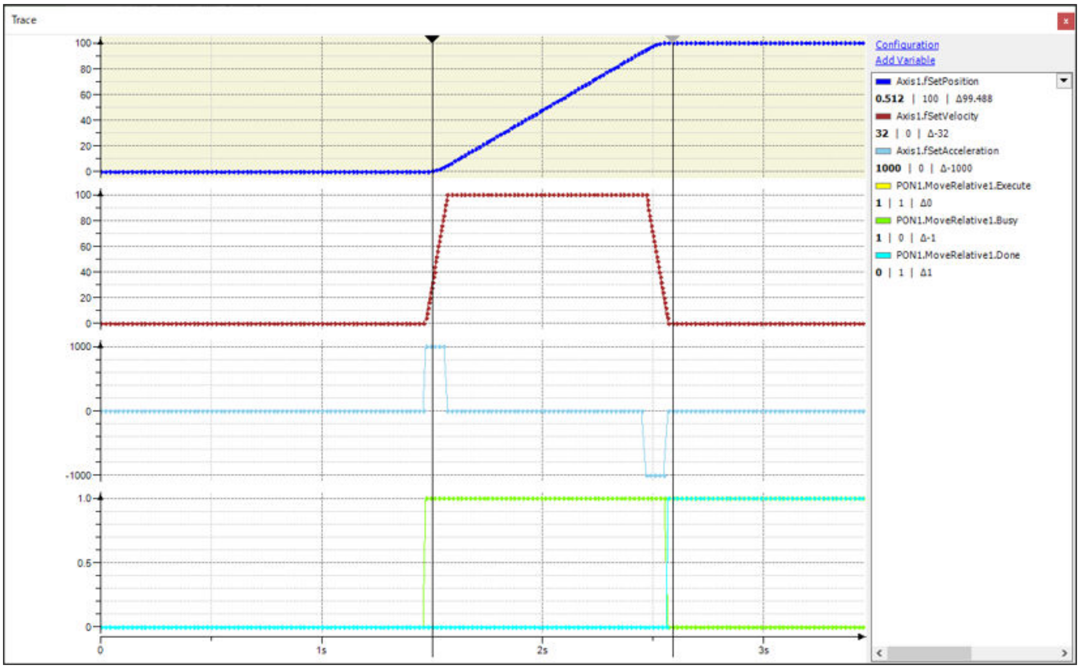
You can reduce or enlarge a diagram by operating a mouse or a keyboard. The operation method is shown below.

Operation items	Mouse operation	Keyboard operation
Shorten time (X) axis	Rotate the mouse wheel rearward.	“－”
Lengthen time (X) axis	Rotate the mouse wheel forward.	“＋”
Shorten Y-axis	With “Ctrl” held depressed, rotate the mouse wheel rearward.	“Ctrl” + “-”
Lengthen Y-axis	With “Ctrl” held depressed, rotate the mouse wheel forward.	“Ctrl” + “+”

■ Cursor Function

A maximum of two cursors can be displayed on the trace screen, and a value at the cursor place can be displayed. Whenever you select Trace>Cursor from the menu bar, the cursor display toggles through “one cursor display > two cursors display > no cursor display”. Values are displayed under the name of each variable on the right side of the trace object, as shown below.

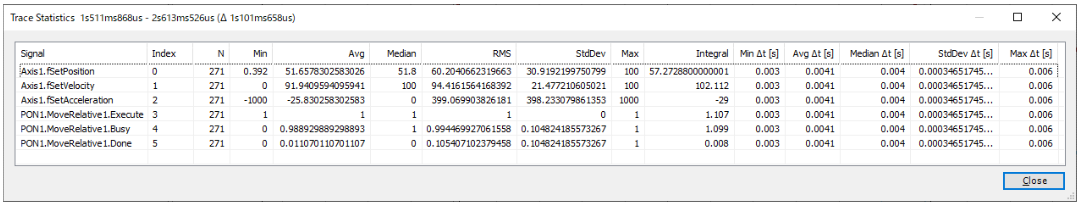
A value at the left cursor place | A value at the right cursor place | A Δ value (the value at the right cursor place - the value at the left cursor place)



Axis1.fSetPosition
0.512 | 100 | Δ99.488
Axis1.fSetVelocity
32 | 0 | Δ-32
Axis1.fSetAcceleration
1000 | 0 | Δ-1000
PON1.MoveRelative1.Execute
1 | 1 | Δ0
PON1.MoveRelative1.Busy
1 | 0 | Δ-1
PON1.MoveRelative1.Done
0 | 1 | Δ1

■ Trace Statistics Function

The trace statistics function is used to display the upper limit, lower limit, mean value, median, root-mean-square (RMS), and suchlike of trace data by variable. When you select **Trace**>**Statistics** from the menu bar, trace statistics are displayed as shown below. For two cursors display, statistics of trace data between the cursors are displayed.

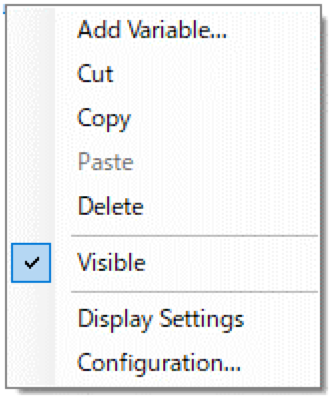


Trace Statistics 1s511ms868us - 2s613ms526us (Δ 1s101ms658us)

Signal	Index	N	Min	Avg	Median	RMS	StdDev	Max	Integral	Min Δt [s]	Avg Δt [s]	Median Δt [s]	StdDev Δt [s]	Max Δt [s]
Axis1.fSetPosition	0	271	0.392	51.6578302583026	51.8	60.2040662319663	30.9192199750799	100	57.2728800000001	0.003	0.0041	0.004	0.00034651745...	0.006
Axis1.fSetVelocity	1	271	0	91.9409594095941	100	94.4161564168392	21.477210605021	100	102.112	0.003	0.0041	0.004	0.00034651745...	0.006
Axis1.fSetAcceleration	2	271	-1000	-25.830258302583	0	399.069903826181	398.233079861353	1000	-29	0.003	0.0041	0.004	0.00034651745...	0.006
PON1.MoveRelative1.Execute	3	271	1	1	1	1	0	1	1.107	0.003	0.0041	0.004	0.00034651745...	0.006
PON1.MoveRelative1.Busy	4	271	0	0.988929889298893	1	0.994469927061558	0.104824185573267	1	1.099	0.003	0.0041	0.004	0.00034651745...	0.006
PON1.MoveRelative1.Done	5	271	0	0.011070110701107	0	0.105407102379458	0.104824185573267	1	0.008	0.003	0.0041	0.004	0.00034651745...	0.006

■ Variable Menu

To display the variable menu, right-click a registered variable on the right side of the trace screen with the mouse cursor put on it. The selected variable can be operated.



Setting items	Overview
Add Variable	The Variable settings pane (Trace Configuration) appears.
Cut	Cut the selected variable.
Copy	Copy the selected variable.
Paste	Paste the copied variable.
Delete	Delete the selected variable.
Visible	The diagram display for the selected variable switches. With check box selected: Display, With check box not selected: No display
Display settings	The Display settings pane (Trace Configuration) appears.
Configuration	The Trace Configuration window appears.

15.3.3 Executing Trace

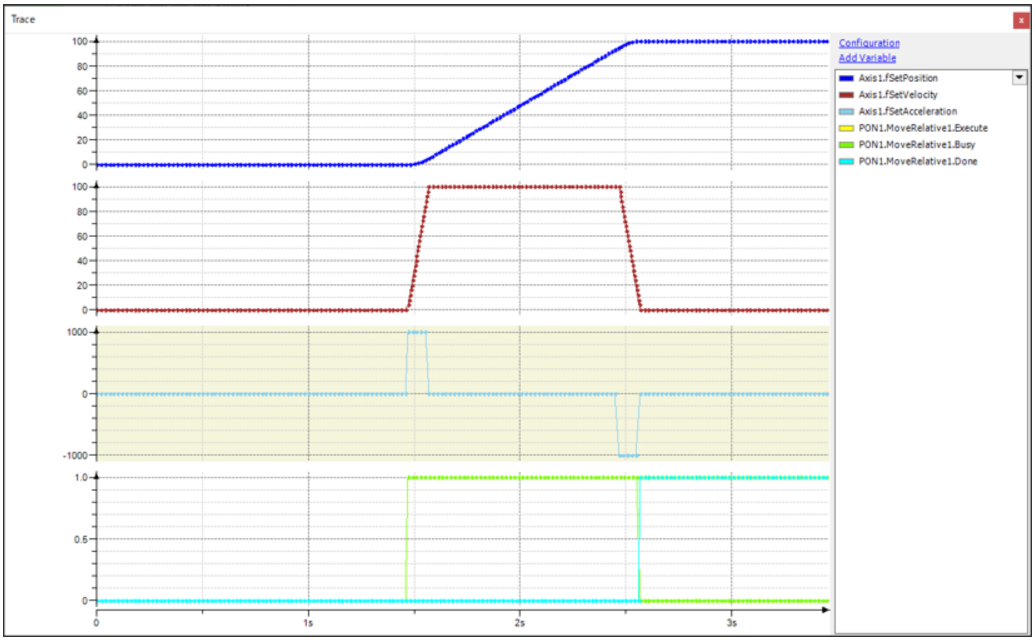
By executing a trace, you can check values of the variables registered in the object.

Procedure

1. Connect the PC where GM Programmer is installed and the GM1 controller and log in to the GM1 controller.

2. From the menu bar, select “Trace > Download Trace”.

   A trace will be started. The values of the variables assigned to each diagram are recorded, and trace data is displayed in graph form.



3. From the menu bar, select “Trace > Stop Trace”.

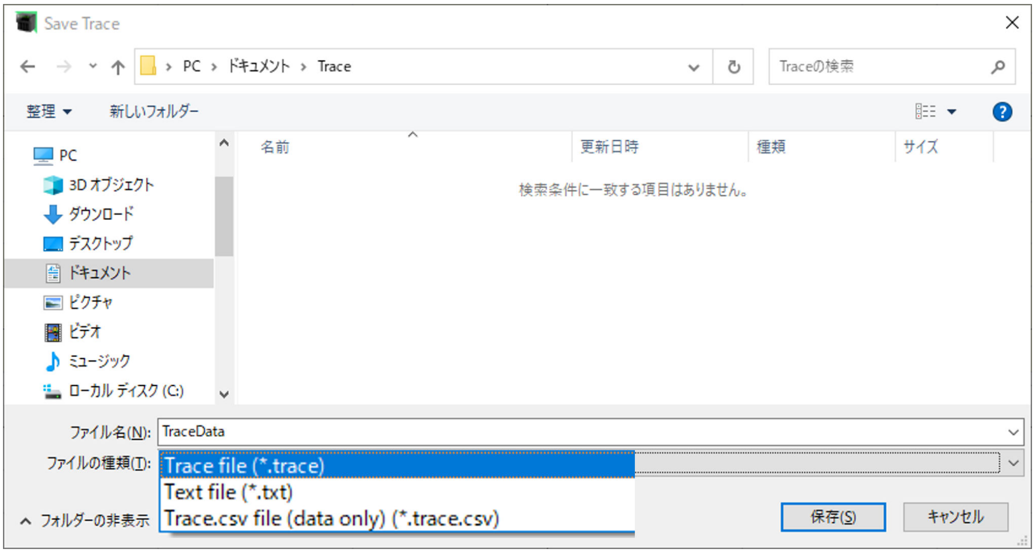
   The trace will be stopped. To automatically stop the trace, enable Trigger in the trace settings.

15.3.4 Saving Trace

Data recorded through execution of the trace can be saved.

Procedure

1. From the menu bar, select “Trace > Save Trace”.
2. The Save Trace pop-up window appears. Specify a destination where the trace data is saved, as well as a file name. From the File Type menu, select a file format in which the trace data is saved.



ファイル名(N): TraceData

ファイルの種類(T): Trace file (*.trace)

3. After the settings are configured, click the “Save” button.
4. If the data is saved in trace file (*.trace) format, you can open it on the trace object screen by selecting “Trace > Load Trace” from the menu bar.

Trace File Types

There are three trace file formats.

Through the trace file (.trace) format, all of recorded variable values (hereinafter referred to as trace data) and trace configuration data can be saved. By importing a file saved in this format, you can display a trace waveform identical to that shown when the file was saved. A file in trace file format can be imported offline to change the save format to trace dump file or text file.

Thus, when you use such data, you are recommended to save it in trace file (.trace) format.

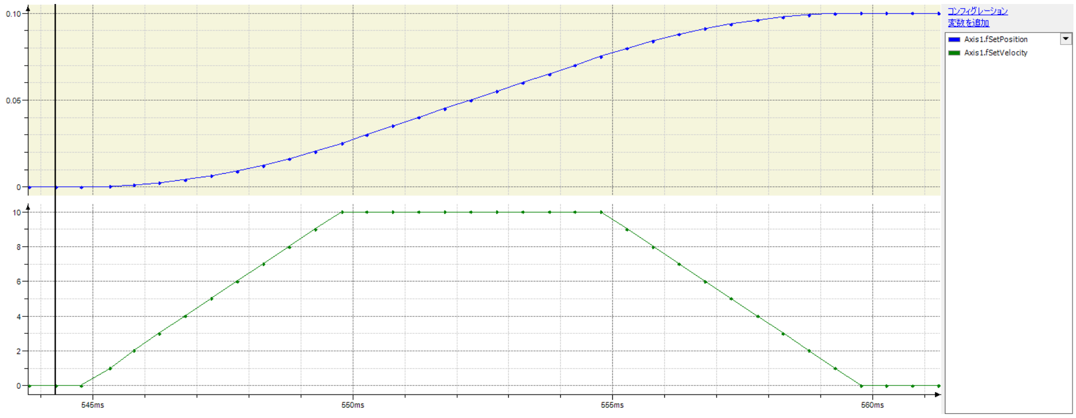
Setting items	Extension	Overview
Trace file	*.trace	Trace data and trace configuration data are output in XML format. You can import output trace files and analyze trace data offline.
Trace dump file	*.trace.csv	Trace data and part of trace configuration data are output in CSV format. You can import output trace files and analyze trace data offline. (Note 1)
Text file	*.txt	Trace data is output in text (ASCII) format. You can edit such a file using an external tool. However, text files cannot be imported.

(Note 1) To output a trace dump file, do not set any record condition in the record settings.

Examples of Output Trace Files

Examples in which trace object data as shown below is output into trace file.

Variable0 = Axis1.fSetPosition, Variable1 = Axis1.fSetVelocity



Example of output trace file (.trace)

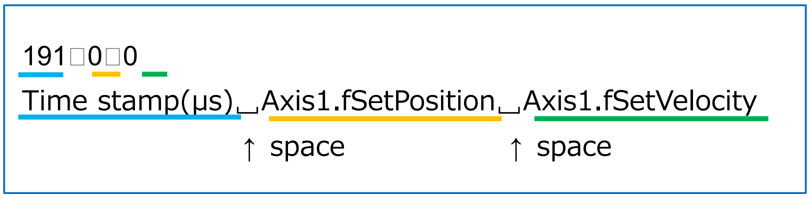
```
<Trace>
 <TraceConfiguration>
  <Single xml:space="preserve" Type="{f7aa3620-8073-4c91-b6ec-86ed9eb60303}" Method="IArchivable">
 <Dictionary Type="System.Collections.Hashtable" Name="TraceDataConfig" />
 <Dictionary Type="System.Collections.Hashtable" Name="TraceOutputConfig" />
    :            Configuration Data
    :            Trigger setting Data
    :            Variable setting Data
 </TraceConfiguration>
 <TraceData Version="1.0.0.0">
  <TraceRecord>
   <TriggerState>3</TriggerState>
   <StartTime>10534190988</StartTime>
   <TriggerStartDate>2022/11/11 1:11:11</TriggerStartDate>
   <TriggerTimeStamp>10534609669</TriggerTimeStamp>
   <TriggerHistory>418681</TriggerHistory>
   <TraceVariable VarName="Axis1.fSetPosition" VariableIndex="0" Type="System.Double">
    <Values>0,0.00025,0.001,0.00225,0.004,0.00625,0.00901,0.01225,0.016,0.02025,0.02505,0.03006
            0.035,0.04008,0.04505,0.05,0.05507,0.06005,0.065,0.07,,,</Values>  Trace Data (Axis1.fSetPosition)
  <Timestamps>419187,419734,420175,420676,421205,421678,422176,422676,423188,423677,424182,424681,
            425210,425675,426175,426681,427196,427676,428177,428676,,,</Timestamps>
   </TraceVariable>
   <TraceVariable VarName="Axis1.fSetVelocity" VariableIndex="0" Type="System.Double">
    <Values>0,1,2,3,4,5,6,7,8,9.018,10,10,10,
            10,10,10,10,10,10,10,10,,,,</Values>                 Trace Data (Axis1.fSetVelocity)
    <Timestamps>419187,419734,420175,420676,421205,421678,422176,422676,423188,423677,424182,424681,
            425210,425675,426175,426681,427196,427676,428177,428676,,,</Timestamps>
   </TraceVariable>
  </TraceRecord>
 </TraceData>
</Trace>
```

Example of output trace dump file (.trace.csv)

```
[key]; [value]
Version; 0x03050000
Name; Application.Trace
ApplicationName; Application
ApplicationDataGuid; 00000000-0000-0000-0000-000000000000
IecTaskName; MotionTask
Comment; Trace_Coment
       :     Configuration Data
       :     Trigger setting Data
       :     Variable0 setting Data
       :
0.Data;
; 419187; 0             Variable0 Trace Data
; 419734; 0.00025        ; Time Stamp(μs); Axis1.fSetPosition
; 420175; 0.001
; 420676; 0.00225
; 421205; 0.004
; 421678; 0.00625
; 422176; 0.00901
       :
1.Variable; Axis1.fSetVelocity
       :     Variable1 setting Data
       :
1.Data;
; 419187; 0             Variable1 Trace Data
; 419734; 1              ; Time Stamp(μs); Axis1.fSetVelocity
; 420175; 2
; 420676; 3
; 421205; 4
; 421678; 5
; 422176; 6
       :
```

Example of output text file (.txt)

```
GM Programmer V** Trace: Trace
E:¥GMP_Trace.project
Timestamp(μs) Axis1.fSetPosition Axis1.fSetVelocity
191 0 0
687 0 0
1198 0 0
1686 0 0
2186 0 0
2684 0 0
3184 0 0
     :
419187 0 0
419734 0.00025 1
420175 0.001 2
420676 0.00225 3
421205 0.004 4
421678 0.00625 5
422176 0.00901 6
422676 0.01225 7
423188 0.016 8
423677 0.02025 9.0018
424182 0.02505 10
424681 0.03006 10
425210 0.035 10
425675 0.04008 10
426175 0.04505 10
426681 0.05 10
427196 0.05507 10
427676 0.06005 10
428177 0.065 10
428676 0.07 10
     :
```



191□0□0

Time stamp(μs)␣Axis1.fSetPosition␣Axis1.fSetVelocity

↑ space ↑ space

15.4 Recipe Manager Functions

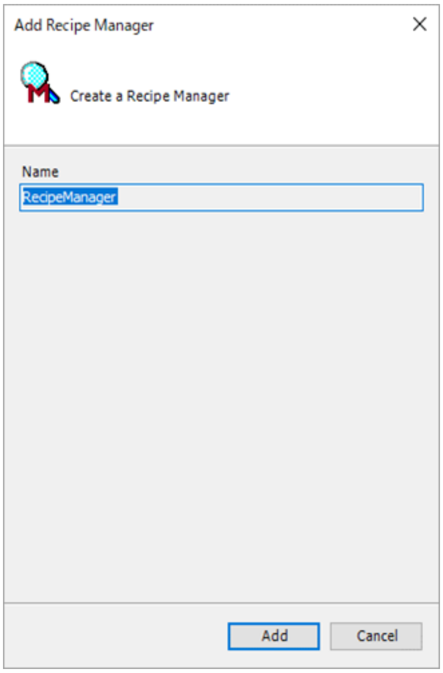
With the Recipe Manager, you can add recipes and alos switch and control the recipe data.

15.4.1 Setting the Recipe Manager

This section explains how to add and set Recipe Manager objects.

1 2 Procedure

1. Right-click the “Application” object in the navigation pane and select **Add Object** >**Recipe Manager** from the context-sensitive menu that is displayed.
   The “Add Recipe Manager” dialog box will be displayed.

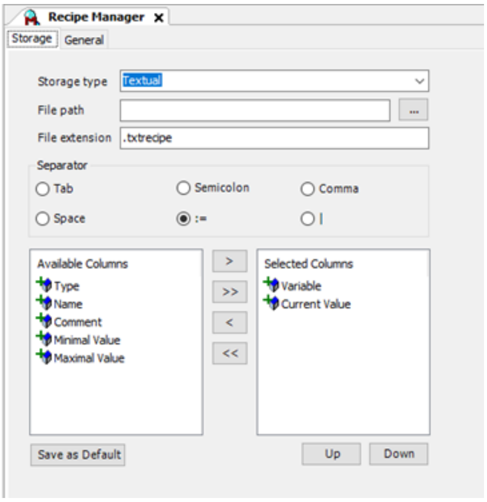


Add Recipe Manager

Create a Recipe Manager

Name: RecipeManager

2. Click the [Add]button.
3. The Recipe Manager setting window will be displayed.
   The default settings can be used as is. You can also make settings if necessary.
   ■Strage
   It is displayed when you select **Recipe Manager** >**Storage** tab.



Recipe Manager ✕

Storage | General

Storage type: Textual

File path:

File extension: .txtrecipe

Separator

○ Tab ○ Semicolon ○ Comma

○ Space ● := ○ |

Available Columns: Type, Name, Comment, Minimal Value, Maximal Value

Selected Columns: Variable, Current Value

Setting items		Overview
Storage type	Textual	The recipe file is written in text format.
	Binary	The recipe file is written in binary format. Please do not use this setting.
File path		Set the path where you want to save the recipe file. The path that can be specified is under the root directory of the SD card. Examples Recipe\(Note 1)
File extension		Specify the extension of the recipe file. Possible to set a desired name with up to 10 characters.
Separator	Tab/Semicolon/ Comma/Space/ ":=" / "|"	Specify the delimiter within the recipe file. If you do not set the same settings as when saving, it will not be loaded correctly. Ex. Description in the recipe file when ": =" is selected AAA:=1 (Variable := Current Value) BBB:=12 (Variable := Current Value)
Available Columns	Variable Current Value Type Name Comment Minimal Value Maximal Value	It is a list of information that can be described in the recipe file. The information registered in this list is not saved in the recipe file.
Selected digits(Note 2)(Note 3)(Note 4)		Specifies the information and the order in which it is stored in the recipe file. It will not be loaded correctly unless the settings are the same as when saving. The "Current Value" is always registered.
Save as Defalt		Please do not use this.

(Note 1) Enter backslash "\" at end of folder.

(Note 2) "Selected Columns" must need "Variable" and "Current Value".

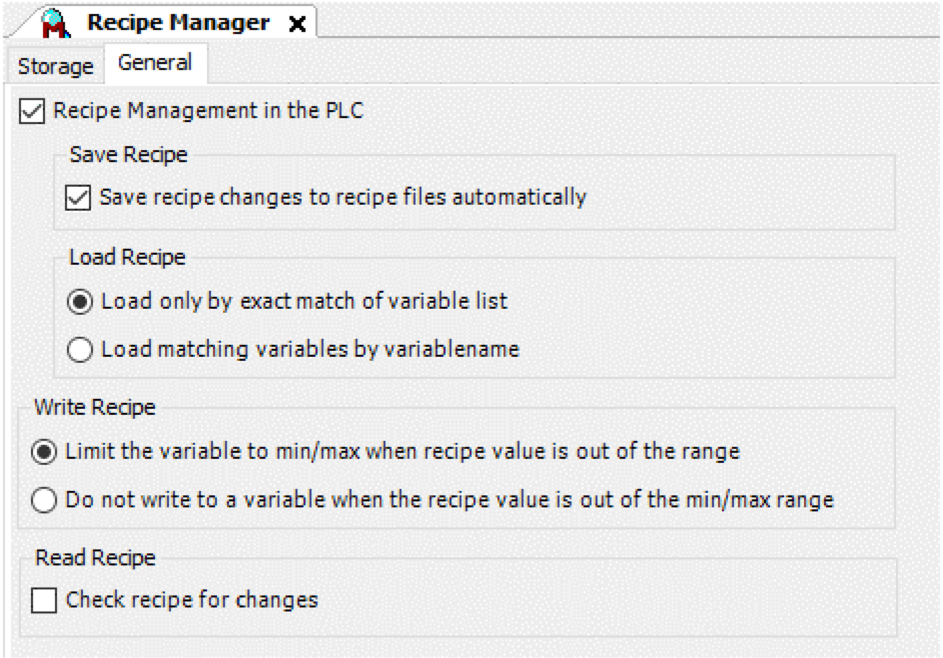
(Note 3) When loading a recipe from the Recipe Manager menu, all items written to the recipe file in the "Selected Columns" setting are loaded into the recipe definition.

(Note 4) When using "Minimal Value" and "Maximal Value", be sure to register both in the "Selected Columns".

■General

It is displayed when you select **Recipe Manager** >**General** tab.

For details, refer to the GM1 Series Reference Manual Instruction Edition.



☑ Recipe Management in the PLC

Save Recipe

☑ Save recipe changes to recipe files automatically

Load Recipe

● Load only by exact match of variable list

○ Load matching variables by variablename

Write Recipe

● Limit the variable to min/max when recipe value is out of the range

○ Do not write to a variable when the recipe value is out of the min/max range

Read Recipe

□ Check recipe for changes

Setting items		Overview
Recipe Management in the PLC		"Save Recipe" setting, "Load Recipe" setting, and instructions by POU are possible.
Save Recipe	Save recipe changes to recipe files automatically	Each recipe set in the recipe definition is automatically saved in the recipe file when you log in to the GM1.
Load Recipe	Load only by exact match of variable list	The recipe value is loaded only if the recipe file contains all the variables in the variable list of the recipe definition and they are listed in the same order as the "Selected Columns".
	Liad matching variables by variablename	Only variables with the same name in the recipe file and the recipe definition will be loaded with the recipe value.
Write Recipe	Limit the variable to min/max when recipe value is out of range	When writing the recipe value to the "Current Value", if the value is outside the range of "Minimal Value" / "Maximal Value", the minimal value or maximal value is written.
	Do not write to a variable when the recipe value is out of the min/max range	When writing the recipe value to the "Current Value", if the value is outside the range of the "Minimal Value" / "Maximal Value", the value is not written and the "Current Value" is retained.
Read Recipe	Check recipe for changes	A file check is performed when the ReadAndSaveRecipe method is called. It also enables the

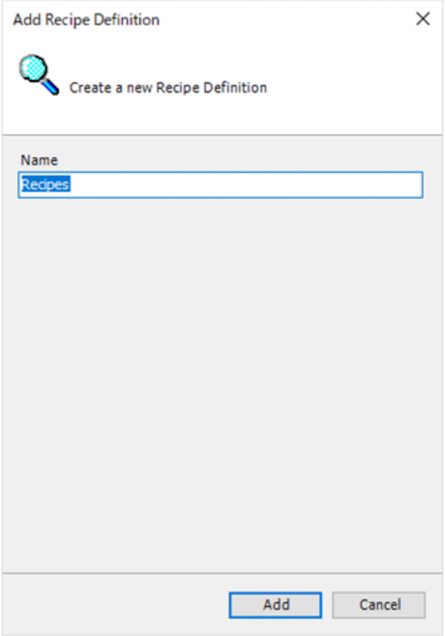
Setting items		Overview
		RecipeManCommands.prvCompareRecipe method.(Note 5)

(Note 1) For details, refer to the GM1 Series Reference Manual (Instruction).

15.4.2 Setting the Recipe Definition

Procedure

1. Right-click the “Recipe Manager” object and in the navigation pane and select **Add Object**>**Recipe Definition** from the context-sensitive menu that is displayed.
   The “Add Recipe Definition” dialog box will be displayed.



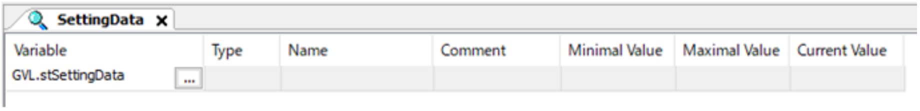
Add Recipe Definition

Create a new Recipe Definition

Name: Recipes

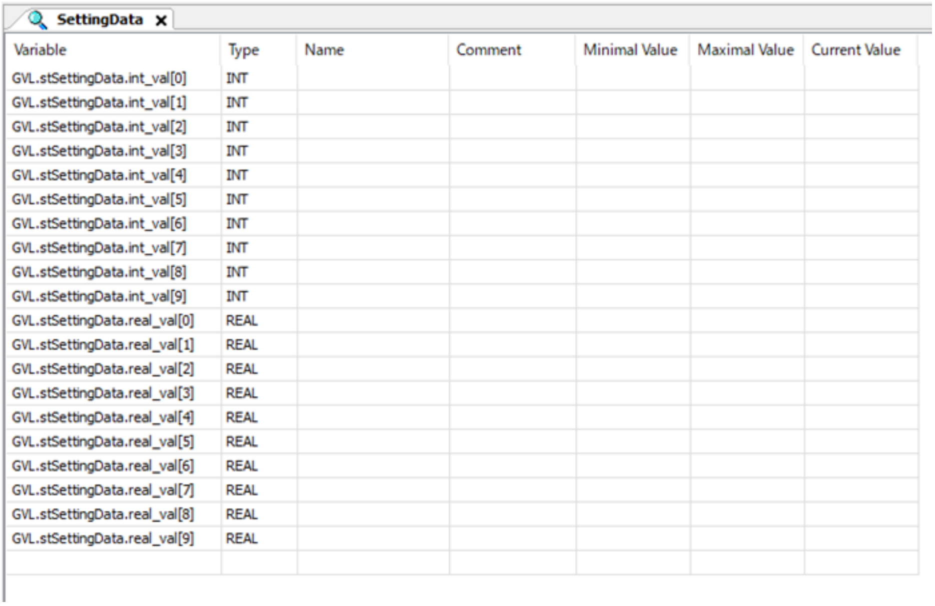
2. Enter a recipe definition name and click the [Add] button.
   Possible to set a desired name with up to 35 characters.

3. Move the cursor to below the variable, enter a variable name you want to add to the recipe definition.



Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value
GVL.stSettingData						

If you enter an array or structure, a list of developed variables is automatically registered.
However, since it takes time for the development if the number of elements is large, implement this only after saving the project in advance.



SettingData

Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value
GVL.stSettingData.int_val[0]	INT					
GVL.stSettingData.int_val[1]	INT					
GVL.stSettingData.int_val[2]	INT					
GVL.stSettingData.int_val[3]	INT					
GVL.stSettingData.int_val[4]	INT					
GVL.stSettingData.int_val[5]	INT					
GVL.stSettingData.int_val[6]	INT					
GVL.stSettingData.int_val[7]	INT					
GVL.stSettingData.int_val[8]	INT					
GVL.stSettingData.int_val[9]	INT					
GVL.stSettingData.real_val[0]	REAL					
GVL.stSettingData.real_val[1]	REAL					
GVL.stSettingData.real_val[2]	REAL					
GVL.stSettingData.real_val[3]	REAL					
GVL.stSettingData.real_val[4]	REAL					
GVL.stSettingData.real_val[5]	REAL					
GVL.stSettingData.real_val[6]	REAL					
GVL.stSettingData.real_val[7]	REAL					
GVL.stSettingData.real_val[8]	REAL					
GVL.stSettingData.real_val[9]	REAL					

Setting items	Overview
Variable	The variable names registered in the recipe definition are displayed. To add a new variable, double-click the blank field at the bottom and specify the name of the variable you want to register. You can enter variables using the Input Assistant from the […] button.
Type	The variable type name is displayed. It is automatically entered when the variable is registered.
Name	You can write the name of each variable separately from "Variable".
Comment	You can optionally include additional information about the variable.
Minimal Value / Maximal Value	Specify the minimum and maximum values used in "15.4.1 Setting the Recipe Manager". When using, be sure to set all variables for possible variables.If there is an unconfigured variable, it will not function correctly.
Current Value	The current variable value is displayed.

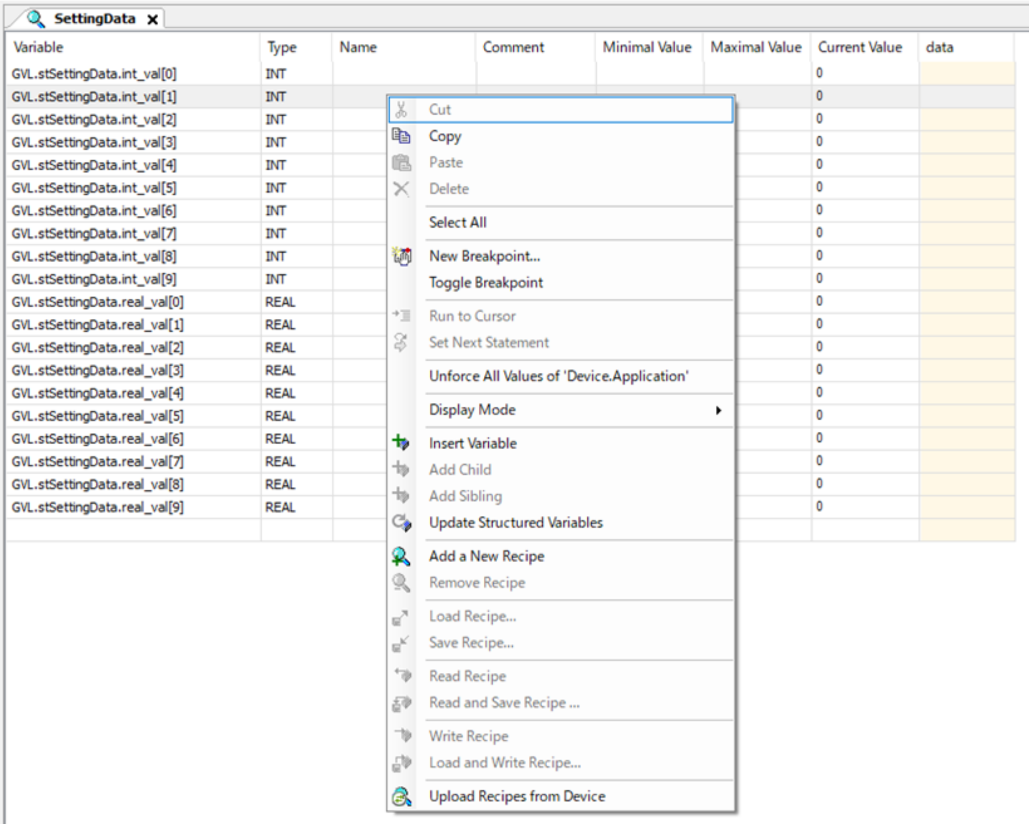
15.4.3 Recipe Operation Using the GM Programmer

1 2 Procedure

1. After setting the recipe definition in "15.4.2 Setting the Recipe Definition", log in to the GM1 unit and download the settings.

2. In the login state, the current value of the variable registered is displayed in the current value.

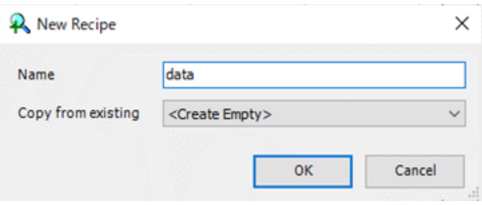
3. In the login state, move the cursor to the recipe definition and select “Add New Recipe”. The “New Recipe” dialog box will be displayed.



SettingData

Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	data
GVL.stSettingData.int_val[0]	INT					0	
GVL.stSettingData.int_val[1]	INT					0	
GVL.stSettingData.int_val[2]	INT					0	
GVL.stSettingData.int_val[3]	INT					0	
GVL.stSettingData.int_val[4]	INT					0	
GVL.stSettingData.int_val[5]	INT					0	
GVL.stSettingData.int_val[6]	INT					0	
GVL.stSettingData.int_val[7]	INT					0	
GVL.stSettingData.int_val[8]	INT					0	
GVL.stSettingData.int_val[9]	INT					0	
GVL.stSettingData.real_val[0]	REAL					0	
GVL.stSettingData.real_val[1]	REAL					0	
GVL.stSettingData.real_val[2]	REAL					0	
GVL.stSettingData.real_val[3]	REAL					0	
GVL.stSettingData.real_val[4]	REAL					0	
GVL.stSettingData.real_val[5]	REAL					0	
GVL.stSettingData.real_val[6]	REAL					0	
GVL.stSettingData.real_val[7]	REAL					0	
GVL.stSettingData.real_val[8]	REAL					0	
GVL.stSettingData.real_val[9]	REAL					0	

4. Enter a name for the new recipe and select OK to add the recipe as show in the above figure.
Possible to set a desired name with up to 35 characters.

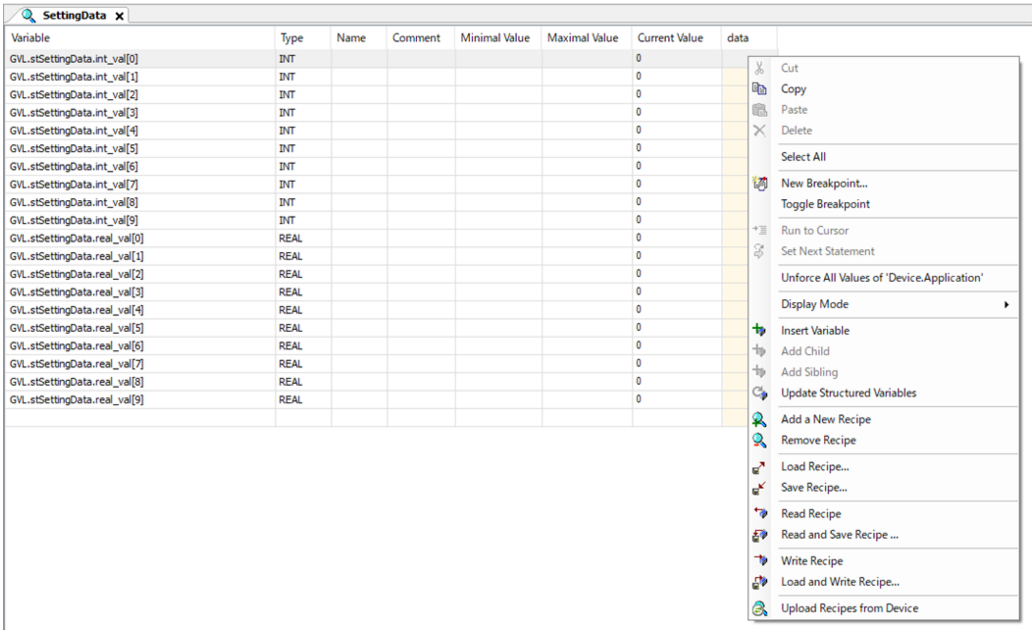


New Recipe

Name: data

Copy from existing: <Create Empty>

5. Move the cursor to the added recipe and right-click to switch and control the recipe data.



Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	data
GVL.stSettingData.int_val[0]	INT					0	
GVL.stSettingData.int_val[1]	INT					0	
GVL.stSettingData.int_val[2]	INT					0	
GVL.stSettingData.int_val[3]	INT					0	
GVL.stSettingData.int_val[4]	INT					0	
GVL.stSettingData.int_val[5]	INT					0	
GVL.stSettingData.int_val[6]	INT					0	
GVL.stSettingData.int_val[7]	INT					0	
GVL.stSettingData.int_val[8]	INT					0	
GVL.stSettingData.int_val[9]	INT					0	
GVL.stSettingData.real_val[0]	REAL					0	
GVL.stSettingData.real_val[1]	REAL					0	
GVL.stSettingData.real_val[2]	REAL					0	
GVL.stSettingData.real_val[3]	REAL					0	
GVL.stSettingData.real_val[4]	REAL					0	
GVL.stSettingData.real_val[5]	REAL					0	
GVL.stSettingData.real_val[6]	REAL					0	
GVL.stSettingData.real_val[7]	REAL					0	
GVL.stSettingData.real_val[8]	REAL					0	
GVL.stSettingData.real_val[9]	REAL					0	

Info.

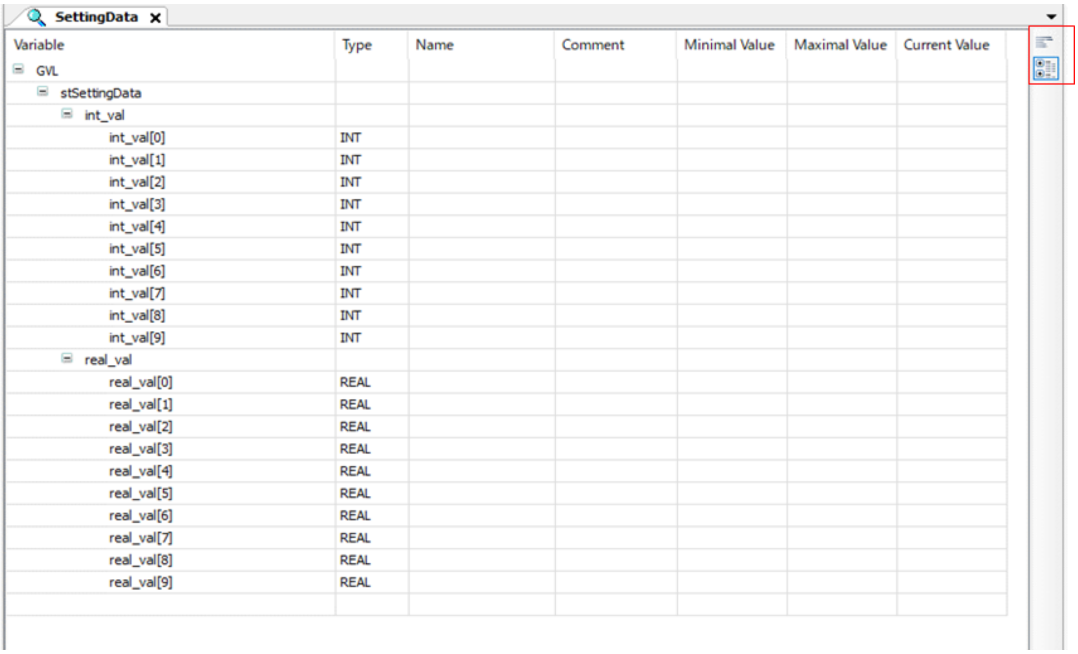
- The following operations can be performed using the GM Programmer.Items that can be operated are different depending on whether the mode is offline or online. GM Programmer

Item	Offline	Online	Function
Insert Variable	○	○	Inerts a new variable in the recipe definition.
Add Child	○		You can add a child array that can be used in the structured view and that has not been registered in the recipe definition.
Add Sibling	○		You can add a brother array that can be used in the structured view and that has not been registered in the recipe definition.
Update Structured variable	○	○	If there are any items that are not registered in structure, they are reflected on the variable list.
Add a New Recipe	○	○	Adds a new recipe to the recipe definition.
Remove Recipe	○	○	Deletes the selected recipe.
Load Recipe	○	○	Loads the recipe file in the operating PC and write to the selected recipe.
Save Recipe(Note 1)	○	○	Saves the selected recipe in the operating PC as a recipe file.
Read Recipe		○	Reads the current value to the recipe data.
Read and Save Recipe		○	Reads the current value to the recipe (within the tool) and saves the recipe file in the operating PC.
Write Recipe		○	Writes the current value to the recipe data.

Item	Offline	Online	Function
Load and Write Recipe		○	Loads data from the recipe file to the recipe (within the tool) and write it in the current value.
Upload Recipes from device		○	Reads the recipe within the controller device and reflects it on the recipe within the tool.
Display mode (Decimal/ Hexadecimal)		○	-

(Note 1) This function cannot be used because REAL data of 7 or more digits and LREAL data of 16 or more digits may not be saved correctly.

- Switch to list view/structured view



Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value
⊟ GVL						
⊟ stSettingData						
⊟ int_val						
int_val[0]	INT					
int_val[1]	INT					
int_val[2]	INT					
int_val[3]	INT					
int_val[4]	INT					
int_val[5]	INT					
int_val[6]	INT					
int_val[7]	INT					
int_val[8]	INT					
int_val[9]	INT					
⊟ real_val						
real_val[0]	REAL					
real_val[1]	REAL					
real_val[2]	REAL					
real_val[3]	REAL					
real_val[4]	REAL					
real_val[5]	REAL					
real_val[6]	REAL					
real_val[7]	REAL					
real_val[8]	REAL					
real_val[9]	REAL					

- When you update the structure or global variable list registered in the recipe definition, you can use the "Update Structured Variables" function to update the variable list in the recipe definition.

The following describes how to update a recipe definition when you declare a new variable for a global variable.

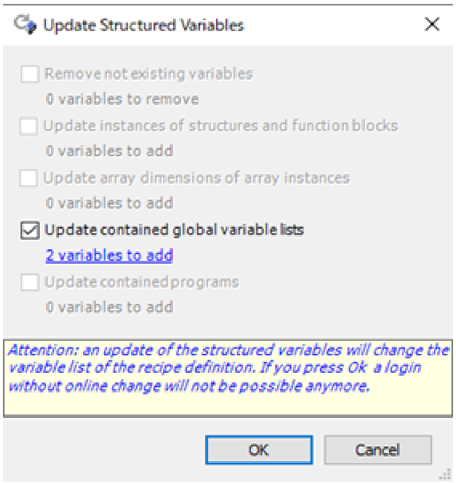
* The same operation is performed when the structure is changed or the variable deleted from the declaration is deleted from the recipe definition.

1. Add add_variables to the global variable.

```
{attribute 'qualified_only'}
VAR_GLOBAL
    variables : ARRAY [0..4] OF INT;
    add_variables : ARRAY [0..1] OF LREAL;
END_VAR
```

2. From the Recipe Definition menu, select “Update Structured Variables” and a window appears.

   Items are displayed in black only if there is a variable that can be added or removed.

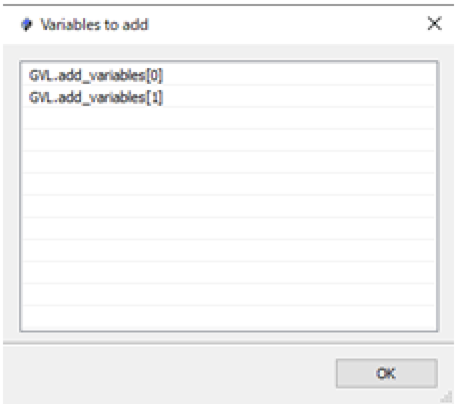


Update Structured Variables

□ Remove not existing variables
0 variables to remove
□ Update instances of structures and function blocks
0 variables to add
□ Update array dimensions of array instances
0 variables to add
☑ Update contained global variable lists
2 variables to add
□ Update contained programs
0 variables to add

Attention: an update of the structured variables will change the variable list of the recipe definition. If you press Ok a login without online change will not be possible anymore.

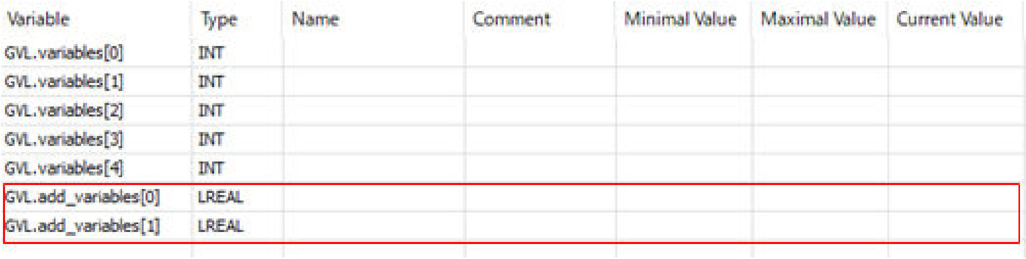
3. Select “2 variables to add” to see the list of variables that will be added to the recipe definition in the update.
   All variables in the list are subject to the update and cannot be selected individually.



Variables to add

GVL.add_variables[0]
GVL.add_variables[1]

4. Variables are automatically added to the recipe definition by pressing OK.



Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value
GVL.variables[0]	INT					
GVL.variables[1]	INT					
GVL.variables[2]	INT					
GVL.variables[3]	INT					
GVL.variables[4]	INT					
GVL.add_variables[0]	LREAL					
GVL.add_variables[1]	LREAL					

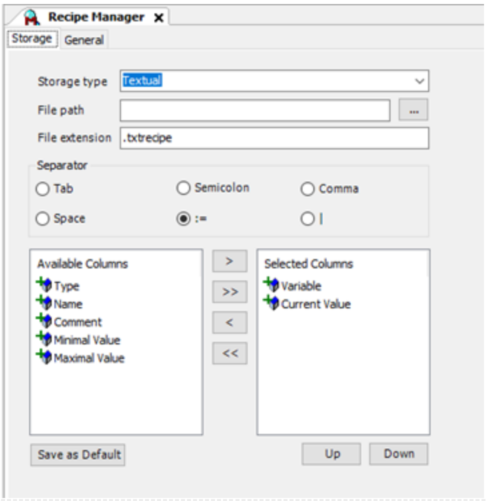
15.4.4 Recipe operation using instructions in the POU

It is also possible to operate a recipe from the recipe method described in the POU without operating the Recipe Manager.For details of each method, refer to the GM1 Series Reference Manual (Instruction Edition).

■Recipe Manager recommended settings

The following are the recommended settings for the Recipe Manager when using instructions in the POU to perform recipe operations.

For details on the effect of changing and executing the settings, refer to the Instructions section.



Storage type: Textual

File path:

File extension: .txtrecipe

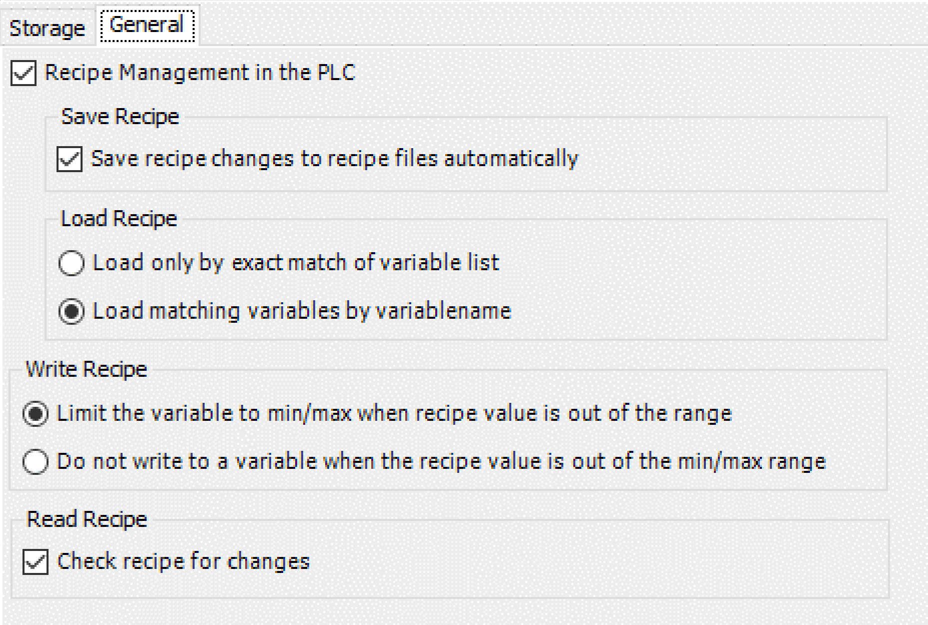
Separator

○ Tab ○ Semicolon ○ Comma

○ Space ● := ○ |

Available Columns: Type, Name, Comment, Minimal Value, Maximal Value

Selected Columns: Variable, Current Value



☑ Recipe Management in the PLC

Save Recipe

☑ Save recipe changes to recipe files automatically

Load Recipe

○ Load only by exact match of variable list

● Load matching variables by variablename

Write Recipe

● Limit the variable to min/max when recipe value is out of the range

○ Do not write to a variable when the recipe value is out of the min/max range

Read Recipe

☑ Check recipe for changes

■Recipe command description example

Example 1 Create a new recipe.

Declaration section

```
PROGRAM sample
VAR
    //FB Instance
    RecipeManCommands_0  : RecipeManCommands;
    //Varable
    ERR  : BOOL;//ErrorFlag
    iRecipeCnt  : INT;//Number of recipe
    GetRcpNames: DWORD;//Return Value of GetRecipeNames
    CreateRcp  : DWORD;//Return Value of CreateRecipe
    RECIPE_DATA_DEF  : STRING := 'Recipes';//Name of Recipe Difinition
    RECIPE_NAME  : STRING := 'Rcp01';//New Recipe Name
    RecipeNames  : ARRAY[0..9] OF STRING;//List of getting Recipe
    i : INT := 0;//FOR loop counter
    iSequence : INT;//SquenceNo.
END_VAR
```

Implement section

```
CASE Process OF
    1:
        //Get the number of recipe files in the SD card
        iRecipeCnt := RecipeManCommands_0.GetRecipeCount(RECIPE_DATA_DEF);
        iSequence := 2;

    2:
        //Get the recipe name if there are recipe files
        IF iRecipeCnt <> 0 THEN
            GetRcpNames:=RecipeManCommands_0.GetRecipeNames(
             RECIPE_DATA_DEF, ADR(RecipeNames), iRecipeCnt, 0);
            iSequence := 3;
        ELSE
            iSequence := 4;
        END_IF

    3:
        //Check if the recipe name is the same as the existing recipe
        FOR i := 0 TO 9 DO
            IF RecipeNames[i] = RECIPE_NAME THEN
                ERR := TRUE;
                RETURN;
            END_IF
        END_FOR
        iSequence := 4;

    4:
        //Create the new recipe
        IF NOT ERR THEN
            CreateRcp := RecipeManCommands_0.CreateRecipe(
             RECIPE_DATA_DEF, RECIPE_NAME);
            iSequence := 0;
        END_IF
END_CASE
```

Example 2 Read the recipe value from the recipe file and write it in the "Current Value"

Declaration section

```
PROGRAM sample
VAR
    //FB Instance
    RecipeManCommands_0     : RecipeManCommands;
    //Varable
    ERR                  : BOOL;//ErrorFlag
    iRecipeCnt            : INT;//Number of recipe
    GetRcpNames             : DWORD;//Return Value of GetRecipeNames
    LoadFromAndWtRcp        : DWORD;//Return Value of LoadFromAndWriteRecipe
    RECIPE_DATA_DEF         : STRING := 'Recipes';//Name of Recipe Difinitio
n
    RECIPE_NAME             : STRING := 'Rcp01';//Reading Recipe Name
     RECIPE_FILENAME         : STRING;//Reading Recipe file Name
     EXTENTION               : STRING := '.txtrecipe';//Extention
    RecipeNames             : ARRAY[0..9] OF STRING;//List of getting Recipe
    i                       : INT := 0;//FOR loop counter
    iSequence               : INT;//SquenceNo.
END_VAR
```

Implement section

```
CASE Process OF
    1:
        //Get the number of recipe files in the SD card

        iRecipeCnt := RecipeManCommands_0.GetRecipeCount(RECIPE_DATA_DEF);
        iSequence := 2;

    2:
        //Get the recipe name if there are recipe files
        IF iRecipeCnt <> 0 THEN
            GetRcpNames:=RecipeManCommands_0.GetRecipeNames(
             RECIPE_DATA_DEF, ADR(RecipeNames), iRecipeCnt, 0);
            iSequence := 3;
        ELSE
            ERR := TRUE;
        END_IF

    3:
        //Check if the recipe name is the same as the existing recipe
        FOR i := 0 TO 9 DO
            IF RecipeNames[i] = RECIPE_NAME THEN
                iSequence := 4;
                EXIT;
            END_IF
        END_FOR
        IF iSequence = 3 THEN
            ERR := TRUE;
        END_IF

    4:
        //Read the recipe file and Write the recipe value to current variabl
e value.
```

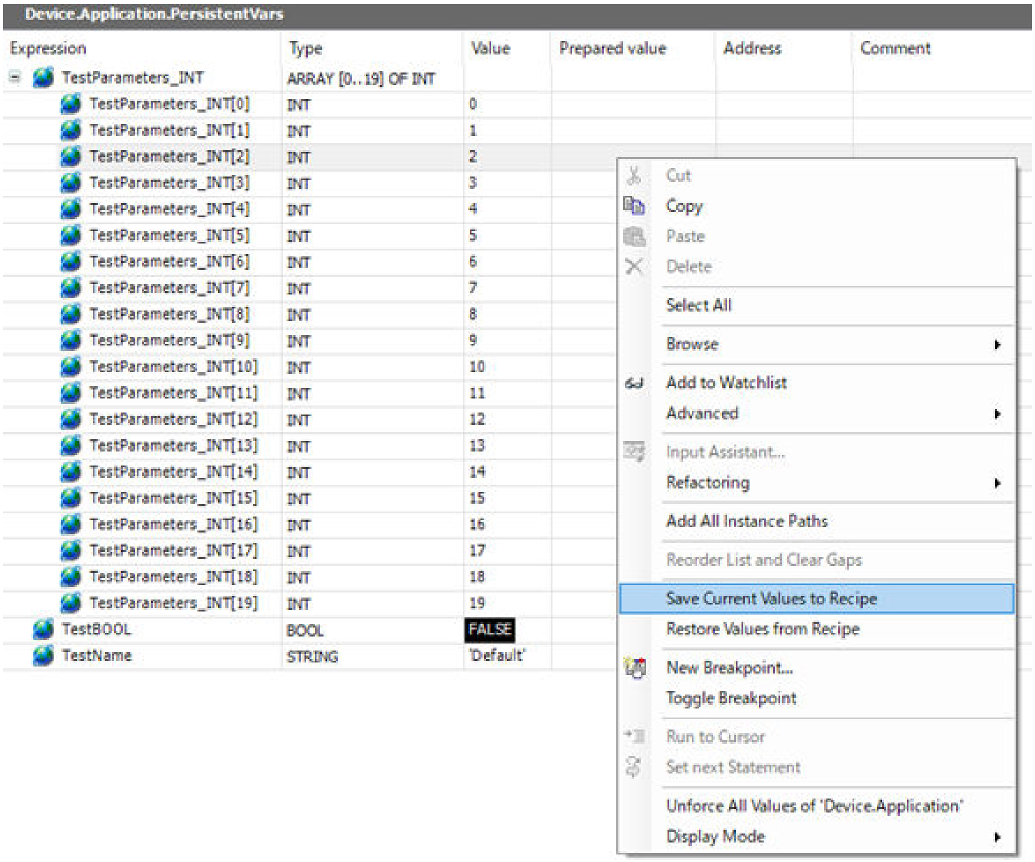
```
       RECIPE_FILENAME := CONCAT(CONCAT(CONCAT(RECIPE_NAME, '.'),
         RECIPE_DATA_DEF),EXTENTION);
        IF NOT ERR THEN
            LoadFromAndWtRcp := RecipeManCommands_0.LoadFromAndWriteRecipe(
             RECIPE_DATA_DEF, RECIPE_NAME, RECIPE_FILENAME);
        END_IF
        iSequence := 0;

END_CASE
```

15.4.5 Saving and restoring persistent variables

You can save and restore data by automatically creating recipe definitions and recipes from persistent variable lists. If you update a variable that has been registered as a persistent variable, the data held by the persistent variable will be cleared when you download it.

Procedure

1. After setting the recipe definition in "15.4.2 Setting the Recipe Definition", log in to the GM1 unit and download the settings.
2. While logged in, place the cursor on a variable in the persistent variable list, right-click, and select [Save Current Value to Recipe] from the menu that appears.

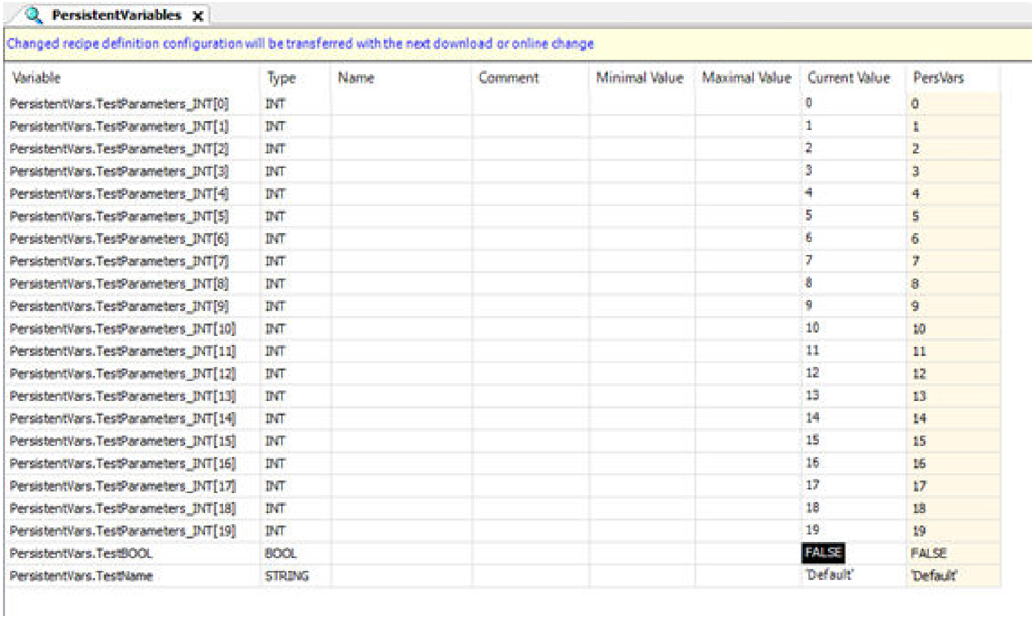


Device.Application.PersistentVars

Expression	Type	Value	Prepared value	Address	Comment
TestParameters_INT	ARRAY [0..19] OF INT				
TestParameters_INT[0]	INT	0			
TestParameters_INT[1]	INT	1			
TestParameters_INT[2]	INT	2			
TestParameters_INT[3]	INT	3			
TestParameters_INT[4]	INT	4			
TestParameters_INT[5]	INT	5			
TestParameters_INT[6]	INT	6			
TestParameters_INT[7]	INT	7			
TestParameters_INT[8]	INT	8			
TestParameters_INT[9]	INT	9			
TestParameters_INT[10]	INT	10			
TestParameters_INT[11]	INT	11			
TestParameters_INT[12]	INT	12			
TestParameters_INT[13]	INT	13			
TestParameters_INT[14]	INT	14			
TestParameters_INT[15]	INT	15			
TestParameters_INT[16]	INT	16			
TestParameters_INT[17]	INT	17			
TestParameters_INT[18]	INT	18			
TestParameters_INT[19]	INT	19			
TestBOOL	BOOL	FALSE			
TestName	STRING	'Default'			

3. Recipe Definition PersistentVariables is created and all variables in the persistent variable list are set.

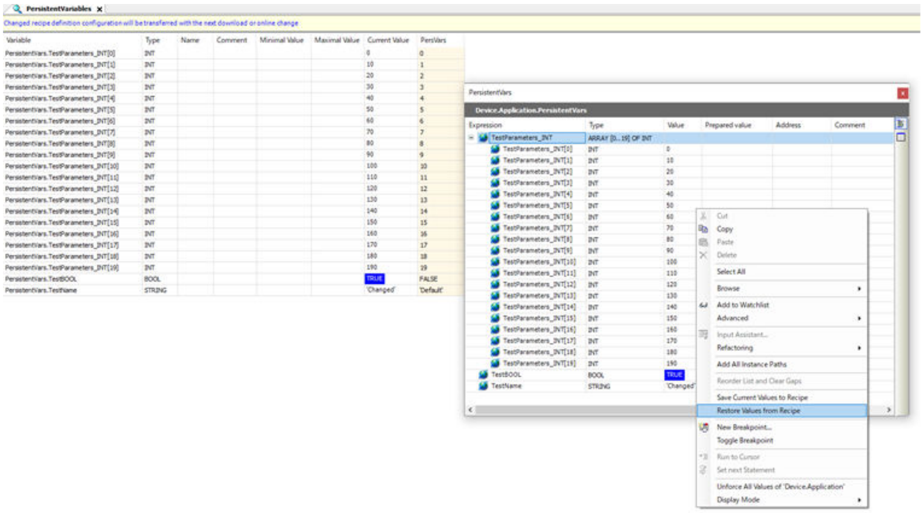
   Also, the recipe "PersVars" is created from the current values.



Changed recipe definition configuration will be transferred with the next download or online change

Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	PersVars
PersistentVars.TestParameters_INT[0]	INT					0	0
PersistentVars.TestParameters_INT[1]	INT					1	1
PersistentVars.TestParameters_INT[2]	INT					2	2
PersistentVars.TestParameters_INT[3]	INT					3	3
PersistentVars.TestParameters_INT[4]	INT					4	4
PersistentVars.TestParameters_INT[5]	INT					5	5
PersistentVars.TestParameters_INT[6]	INT					6	6
PersistentVars.TestParameters_INT[7]	INT					7	7
PersistentVars.TestParameters_INT[8]	INT					8	8
PersistentVars.TestParameters_INT[9]	INT					9	9
PersistentVars.TestParameters_INT[10]	INT					10	10
PersistentVars.TestParameters_INT[11]	INT					11	11
PersistentVars.TestParameters_INT[12]	INT					12	12
PersistentVars.TestParameters_INT[13]	INT					13	13
PersistentVars.TestParameters_INT[14]	INT					14	14
PersistentVars.TestParameters_INT[15]	INT					15	15
PersistentVars.TestParameters_INT[16]	INT					16	16
PersistentVars.TestParameters_INT[17]	INT					17	17
PersistentVars.TestParameters_INT[18]	INT					18	18
PersistentVars.TestParameters_INT[19]	INT					19	19
PersistentVars.TestBOOL	BOOL					FALSE	FALSE
PersistentVars.TestName	STRING					'Default'	'Default'

4. If you place the cursor on a variable in the Persistent Variables list, right-click, and select [Restore Values from Recipe] from the menu that appears, the values currently set for Recipe PersVars are reflected in the Persistent Variables list.



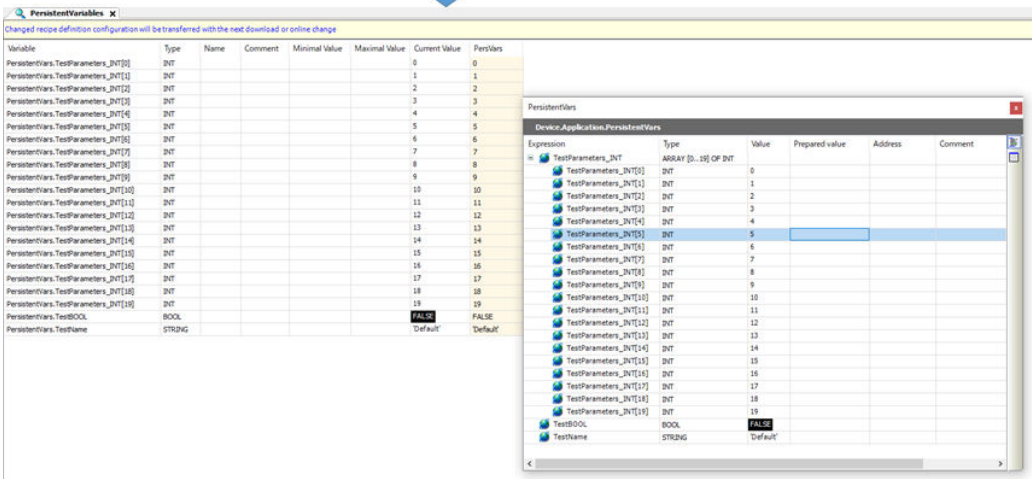
Changed recipe definition configuration will be transferred with the next download or online change

Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	PersVars
PersistentVars.TestParameters_INT[0]	INT					0	0
PersistentVars.TestParameters_INT[1]	INT					10	1
PersistentVars.TestParameters_INT[2]	INT					20	2
PersistentVars.TestParameters_INT[3]	INT					30	3
PersistentVars.TestParameters_INT[4]	INT					40	4
PersistentVars.TestParameters_INT[5]	INT					50	5
PersistentVars.TestParameters_INT[6]	INT					60	6
PersistentVars.TestParameters_INT[7]	INT					70	7
PersistentVars.TestParameters_INT[8]	INT					80	8
PersistentVars.TestParameters_INT[9]	INT					90	9
PersistentVars.TestParameters_INT[10]	INT					100	10
PersistentVars.TestParameters_INT[11]	INT					110	11
PersistentVars.TestParameters_INT[12]	INT					120	12
PersistentVars.TestParameters_INT[13]	INT					130	13
PersistentVars.TestParameters_INT[14]	INT					140	14
PersistentVars.TestParameters_INT[15]	INT					150	15
PersistentVars.TestParameters_INT[16]	INT					160	16
PersistentVars.TestParameters_INT[17]	INT					170	17
PersistentVars.TestParameters_INT[18]	INT					180	18
PersistentVars.TestParameters_INT[19]	INT					190	19
PersistentVars.TestBOOL	BOOL					TRUE	FALSE
PersistentVars.TestName	STRING					'Changed'	'Default'

PersistentVars

Device.Application.PersistentVars

Expression	Type	Value	Prepared value	Address	Comment
TestParameters_INT	ARRAY [0..19] OF INT				
TestParameters_INT[0]	INT	0			
TestParameters_INT[1]	INT	10			
TestParameters_INT[2]	INT	20			
TestParameters_INT[3]	INT	30			
TestParameters_INT[4]	INT	40			
TestParameters_INT[5]	INT	50			
TestParameters_INT[6]	INT	60			
TestParameters_INT[7]	INT	70			
TestParameters_INT[8]	INT	80			
TestParameters_INT[9]	INT	90			
TestParameters_INT[10]	INT	100			
TestParameters_INT[11]	INT	110			
TestParameters_INT[12]	INT	120			
TestParameters_INT[13]	INT	130			
TestParameters_INT[14]	INT	140			
TestParameters_INT[15]	INT	150			
TestParameters_INT[16]	INT	160			
TestParameters_INT[17]	INT	170			
TestParameters_INT[18]	INT	180			
TestParameters_INT[19]	INT	190			
TestBOOL	BOOL	TRUE			
TestName	STRING	'Changed'			



Changed recipe definition configuration will be transferred with the next download or online change

Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	PersVars
PersistentVars.TestParameters_INT[0]	INT					0	0
PersistentVars.TestParameters_INT[1]	INT					1	1
PersistentVars.TestParameters_INT[2]	INT					2	2
PersistentVars.TestParameters_INT[3]	INT					3	3
PersistentVars.TestParameters_INT[4]	INT					4	4
PersistentVars.TestParameters_INT[5]	INT					5	5
PersistentVars.TestParameters_INT[6]	INT					6	6
PersistentVars.TestParameters_INT[7]	INT					7	7
PersistentVars.TestParameters_INT[8]	INT					8	8
PersistentVars.TestParameters_INT[9]	INT					9	9
PersistentVars.TestParameters_INT[10]	INT					10	10
PersistentVars.TestParameters_INT[11]	INT					11	11
PersistentVars.TestParameters_INT[12]	INT					12	12
PersistentVars.TestParameters_INT[13]	INT					13	13
PersistentVars.TestParameters_INT[14]	INT					14	14
PersistentVars.TestParameters_INT[15]	INT					15	15
PersistentVars.TestParameters_INT[16]	INT					16	16
PersistentVars.TestParameters_INT[17]	INT					17	17
PersistentVars.TestParameters_INT[18]	INT					18	18
PersistentVars.TestParameters_INT[19]	INT					19	19
PersistentVars.TestBOOL	BOOL					FALSE	FALSE
PersistentVars.TestName	STRING					'Default'	'Default'

PersistentVars

Device.Application.PersistentVars

Expression	Type	Value	Prepared value	Address	Comment
TestParameters_INT	ARRAY [0..19] OF INT				
TestParameters_INT[0]	INT	0			
TestParameters_INT[1]	INT	1			
TestParameters_INT[2]	INT	2			
TestParameters_INT[3]	INT	3			
TestParameters_INT[4]	INT	4			
TestParameters_INT[5]	INT	5			
TestParameters_INT[6]	INT	6			
TestParameters_INT[7]	INT	7			
TestParameters_INT[8]	INT	8			
TestParameters_INT[9]	INT	9			
TestParameters_INT[10]	INT	10			
TestParameters_INT[11]	INT	11			
TestParameters_INT[12]	INT	12			
TestParameters_INT[13]	INT	13			
TestParameters_INT[14]	INT	14			
TestParameters_INT[15]	INT	15			
TestParameters_INT[16]	INT	16			
TestParameters_INT[17]	INT	17			
TestParameters_INT[18]	INT	18			
TestParameters_INT[19]	INT	19			
TestBOOL	BOOL	FALSE			
TestName	STRING	'Default'			

Info.

- When the recipe definition "PersistentVariables" was created, it was not downloaded into the GM1 itself.

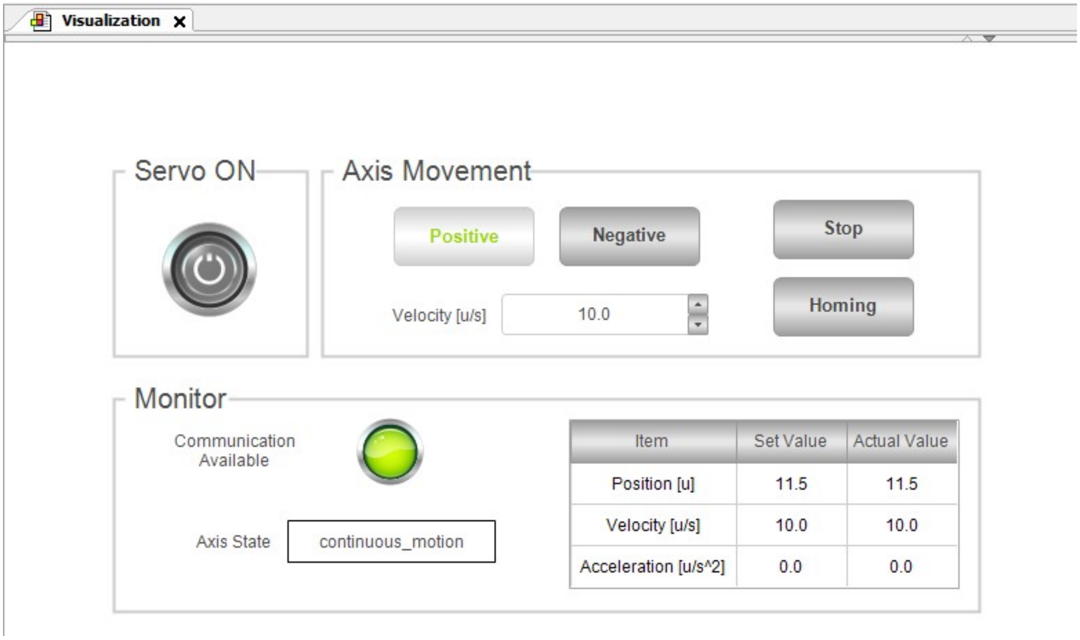
  To use the Recipe Manager function block or to automatically create a recipe file on the SD card, log out, log back in to the GM1 unit, and download the project.
- The function to restore values from a recipe is not limited to the persistent variable list, but covers all set values of the recipe “PersVars”.

  An error occurs if there is no recipe “PersVars” in the recipe definition “PersistentVariables”.

15.5 Visualization Function

By using the visualization function, POU variables can be read or written from the tool elements such as switches and check boxes that are placed on the visualization object. It can be used when performing a debug or other application.

You can create a visualization object that can be used to check parameters during axis operation as shown below.



Velocity [u/s] 10.0

Axis State continuous_motion

Item	Set Value	Actual Value
Position [u]	11.5	11.5
Velocity [u/s]	10.0	10.0
Acceleration [u/s^2]	0.0	0.0

15.5.1 Setting of Visualization

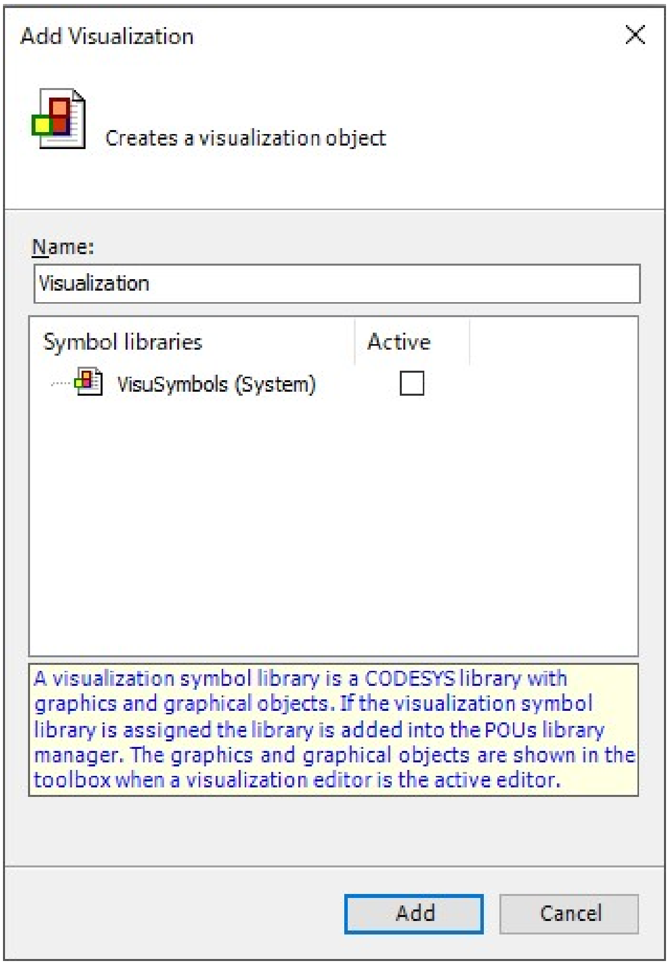
This section explains how to add and set a Visualization object.

1 2 Procedure

1. Right-click the “Application” object in the navigation pane and then select **Add Object**>**Visualization** from the context-sensitive menu that is displayed.

   The “Add Visualization” dialog box will be displayed.

   - First addition



Add Visualization

Creates a visualization object

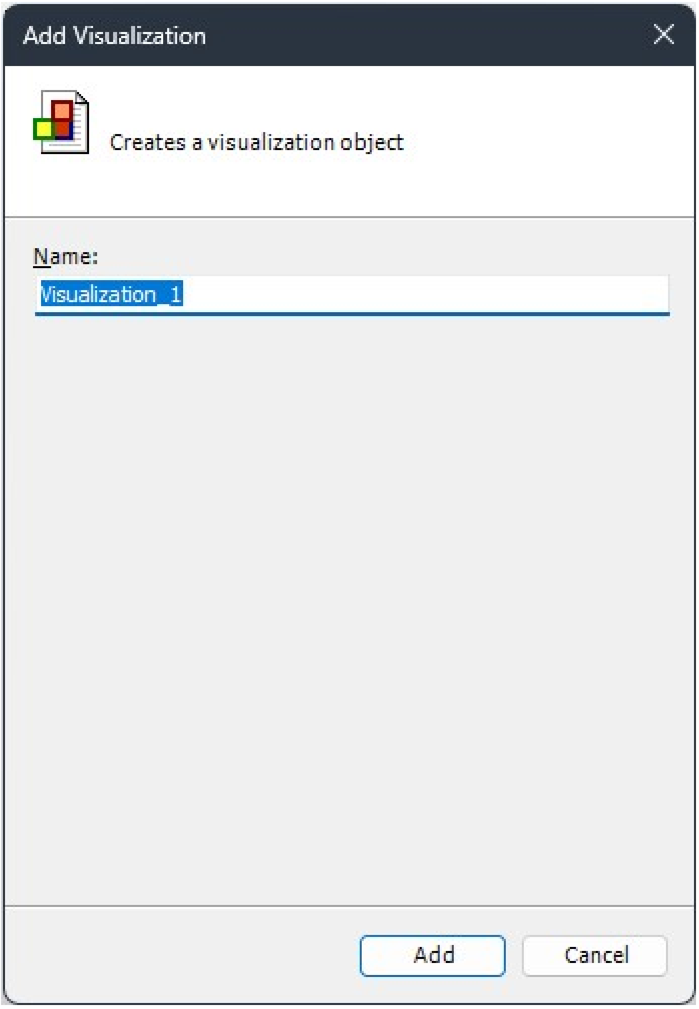
Name:

Visualization

Symbol libraries	Active
VisuSymbols (System)	□

A visualization symbol library is a CODESYS library with graphics and graphical objects. If the visualization symbol library is assigned the library is added into the POUs library manager. The graphics and graphical objects are shown in the toolbox when a visualization editor is the active editor.

   - After the Second addition

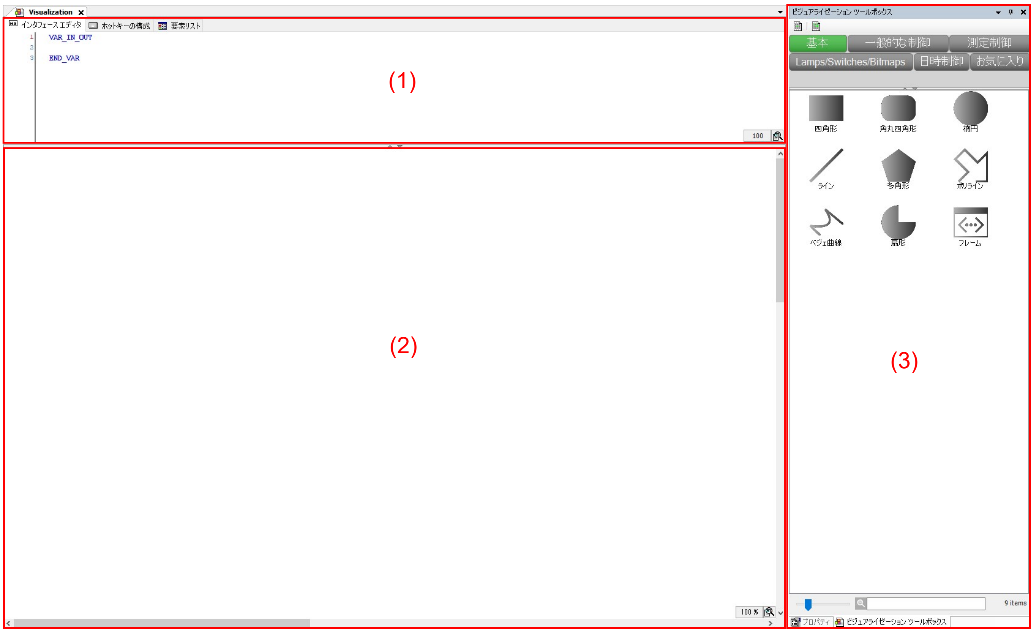


Add Visualization

Creates a visualization object

Name: Visualization_1

2. Enter a desired visualization name and then click the [Add] button.
   When the addition is completed, the Visualization pane will be displayed.



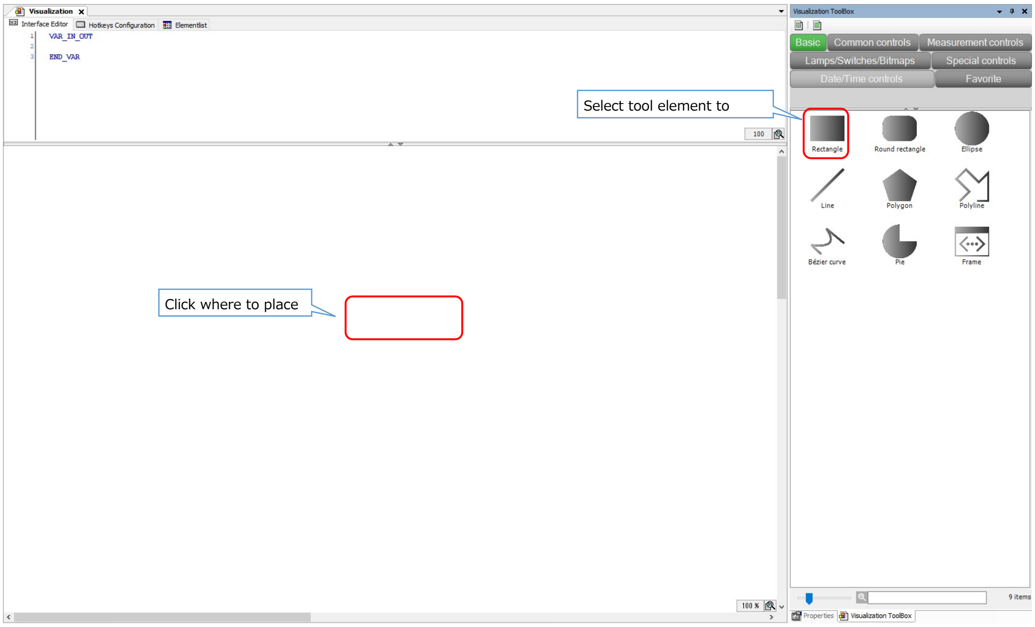
```
VAR_IN_OUT

END_VAR
```

The Visualization pane displays the following:

No.	Name	Function
1	Declaration section	Declare variables for using the visualization as a dialog. Refer to "About editor of the Visualization".
2	Implementation section	Place the tool element.
3	Visualization tool box or properties	Displays the tool element and the properties of the tool element that you want to place in the visualization. The available tool elements are described below.

3. For placing the tool element, select a tool element in the visualization tool box, and then click a desired point in the visualization section.

For the list of tool elements that can be used by GM Programmer, refer to "15.5.3 List of Available Tool Elements".



```
VAR_IN_OUT

END_VAR
```

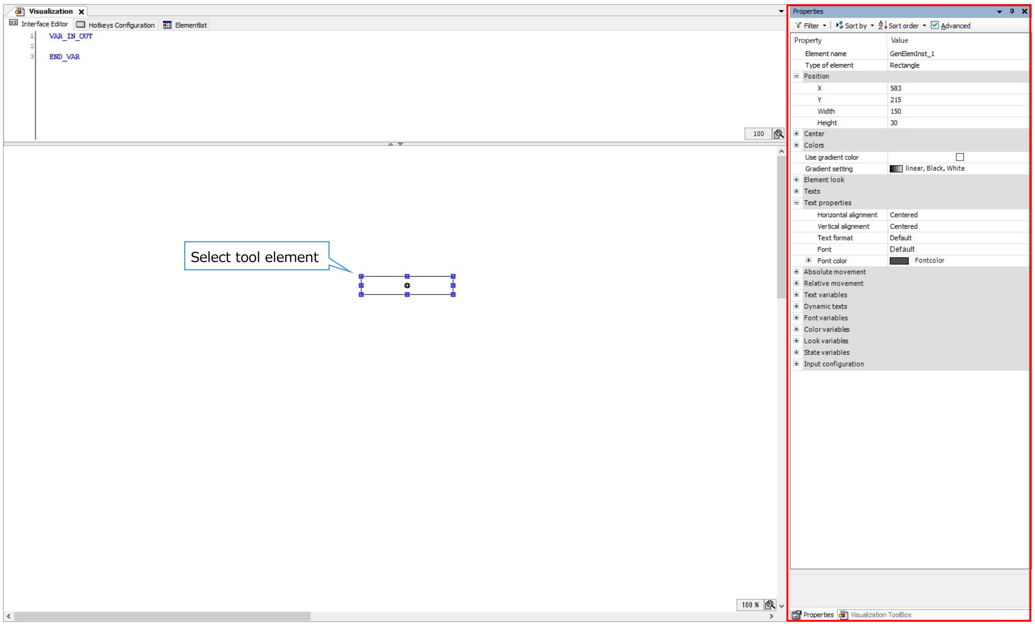
Select tool element to

Click where to place

4. Set the properties of the tool element.

[Properties] will be displayed. On the displayed[Properties], set POU object variables,[Input configuration] that can be used to switch variables by mouse operation.

For the list of [Properties] of tool elements, refer to "15.5.4 List of Properties of Tool Elements".



```
VAR_IN_OUT

END_VAR
```

Select tool element

Property	Value
Element name	GenElemInst_1
Type of element	Rectangle
⊟ Position	
X	583
Y	215
Width	150
Height	30
⊞ Center	
⊞ Colors	
Use gradient color	□
Gradient setting	linear, Black, White
⊞ Element look	
⊞ Texts	
⊟ Text properties	
Horizontal alignment	Centered
Vertical alignment	Centered
Text format	Default
Font	Default
⊞ Font color	Fontcolor
⊞ Absolute movement	
⊞ Relative movement	
⊞ Text variables	
⊞ Dynamic texts	
⊞ Font variables	
⊞ Color variables	
⊞ Look variables	
⊞ State variables	
⊞ Input configuration	

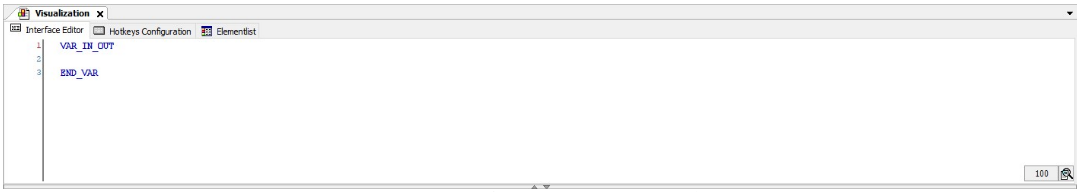
This completes the Visualization setting procedure.

About editor of the Visualization

There are three editors (tabs) in the declaration section of a visualization object

Editor	Function
InterFace Edeitor （InterFace Edeitor Tab）	Declares variables for when the visualization is used as a dialog
HotKeys Edeitor （HotKeys Configuration Tab）	Do not use.
Elementlist Edeitor （Elementlist Tab）	You can change the position and size of placed tool elements.

■ InterFace Edeitor（InterFace Edeitor Tab）



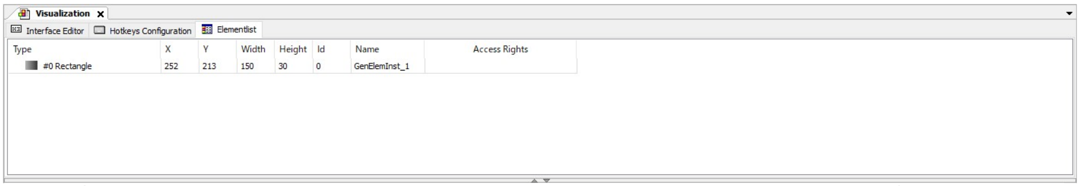
```
VAR_IN_OUT

END_VAR
```

Variables can be declared as in a normal POU. However, the following restrictions apply.

- You can use VAR_IN_OUT, VAR_OUTPUT, and VAR_INPUT.Use them according to the input/output variables, output variables, and input variables used in the dialog.
- They can only be used when the visualization is used as a dialog.

  If you declare variables in the interface editor when using it as a visualization, it will not work as a visualization.
- Initial values cannot be set for declared variables. （NG 例：iVariable : INT := 5;）

■ Elementlist Edeitor（Elementlist Tab）



Type	X	Y	Width	Height	Id	Name	Access Rights
#0 Rectangle	252	213	150	30	0	GenElemInst_1	

A list of tool elements placed in the visualization and their positions and other information are displayed.

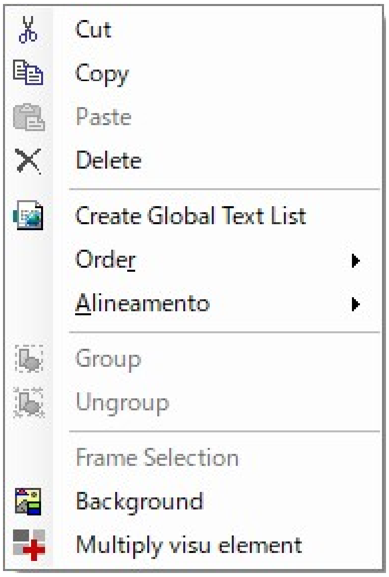
items	Function
Type	The object name of the placed tool element is displayed. The number after # indicates the order in which the tool elements are placed. #0 is the tool element that resides farthest to the rear.
X	X position of tool element Can be changed on the visual element list editor.
Y	Y position of tool element Can be changed on the visual element list editor.
Width	Width of the tool element Can be changed on the visual element list editor.
Height	Height of the tool element Can be changed on the visual element list editor.
ID	Internally set identification number of the tool element
Name	Element name of the tool element Can be changed on the visual element list editor.
Access Rights	Do not use.

Visualization Menu

The Visualization Menu allows for the alignment and grouping of tool elements.

■ Visualization Menu

Select "Visualization" from the Menu bar, or right-click on the object screen in the Visualization area to display it.



Items	Overview
Create Global Text List	Create a GlobalTextList from the [Text] currently used in each Visualization.
InterFace Edeitor	Open the InterFace editor of the Visualization.
HotKeys Edeitor	Do not use.
Elementlist Edeitor	Open the Elementlist of the Visualization.
Order	Change the order in which selected tool elements overlap with other tool elements.
Allineamento	Align multiple selected tool elements.
Group	group multiple selected tool elements. Grouped tool elements are treated as a single object and cannot be selected and edited individually.
Ungroup	Ungroups the selected tool elements.
Frame Selection	Open the "Frame Configuration" dialog for the selected frame object or tab control object.
Background	Set the Background in Visualization. ● User color: Fill the background with the selected color. ● Use image：Do not use.
Multiply visu element	Expand the selected tool element and text variables to the specified number of rows and columns.

■ About Matrix Expansion of Visualization Elements

To use this function, you need to set the properties of the tool element according to the conditions.

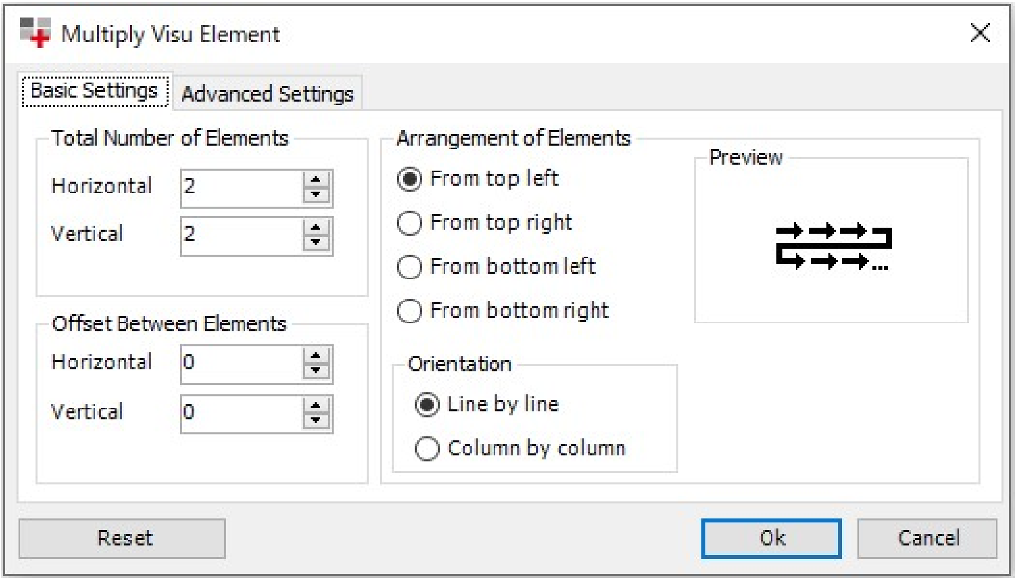
- Conditions
  - Array type variables are set to [text] or [Text variable] of the tool element.
  - The format of the variables to be set for the property is as follows
    - 1st dimension array:"variable name"[$FIRSTDIM$]

  - 2nd dimension array:"variable name"[$FIRSTDIM$, $SECONDDIM$]
- When expanding Array type variables of 3rd dimension or higher, set fixed values except for the 2 indices that set $FIRSTDIM$ and $SECONDDIM$.

- Expansion settings

  Select a tool element that can be expanded and choose "Multiply visu element" from the menu to open the configuration window for expansion. After setting the necessary parameters and pressing the "OK" button, the tool element is copied according to the description and each element of the Array type variables is assigned.

  - Basic Settings tab


Multiply Visu Element

Basic Settings | Advanced Settings

Total Number of Elements
Horizontal: 2
Vertical: 2

Offset Between Elements
Horizontal: 0
Vertical: 0

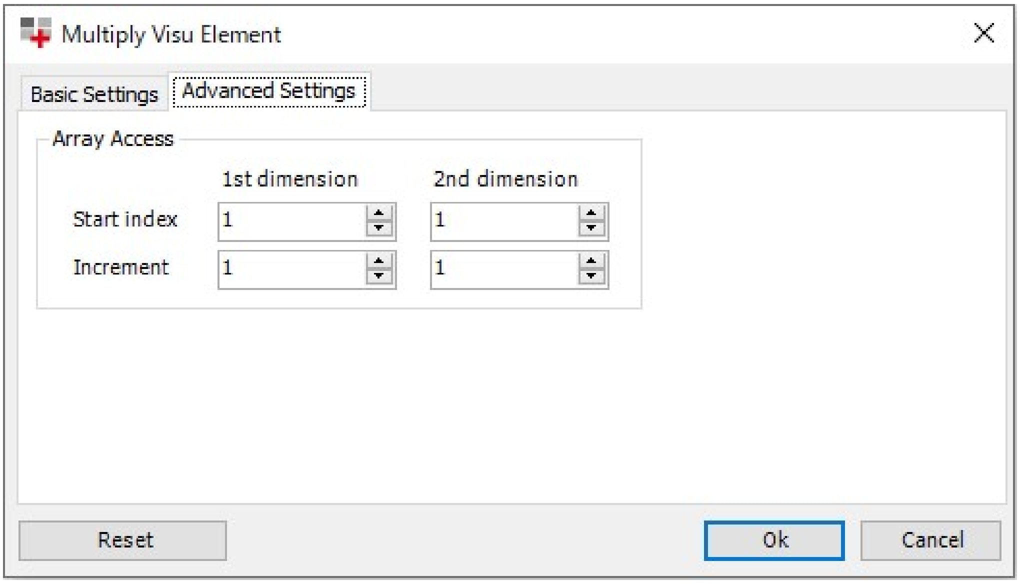
Arrangement of Elements
● From top left
○ From top right
○ From bottom left
○ From bottom right

Orientation
● Line by line
○ Column by column

Items		Overview	
Basic Settings	Total Number of Elements	Horizontal	Number of elements in each row As an initial value, the number of indices corresponding to $FIRSTDIM$ of the set Array type variables is set.
		Vertical	Number of rows required to place all elements The number of indexes corresponding to $SECONDDIM$ (or $FIRSTDIM$) of the set Array type variables is set as the initial value.
	Offset Between Elements	Distance (pixels) between tool elements created when expanded ● 0: The tool elements will be expanded with their borders overlapping by 1 pixel. ● 1: The tool elements will be expanded so that they touch each other. ● n: The tool elements are expanded so that they are separated by n - 1 pixels.	
		Horizontal	Distance between tool elements in row direction (pixels)
		Vertical	Distance between tool elements in columns (pixels)
	Arrangement of Elements	Specify the direction of placement after expansion ● From top left,From top Right,From bottom left,From bottom right	

Items		Overview	
		● From left,From right ● From top,From bottom	
		Orientation	Specification of the direction of $FIRSTDIM$ increment after expansion ● Line by line,Column by column

- Advanced Settings tab



Multiply Visu Element

Array Access

	1st dimension	2nd dimension
Start index	1	1
Increment	1	1

Items		Overview	
Advanced Settings	Array Access	Start index	Specify Start index of Array type variables to be expanded
		Increment	Specify the Increment of the index of the Array type variables to be expanded
		1st dimension	Set for $FIRSTDIM$.
		2nd dimension	Set for $SECONDDIM$

● Example: A 5 × 10 STRING type array declared in GVL is expanded with a Rectangleobject.

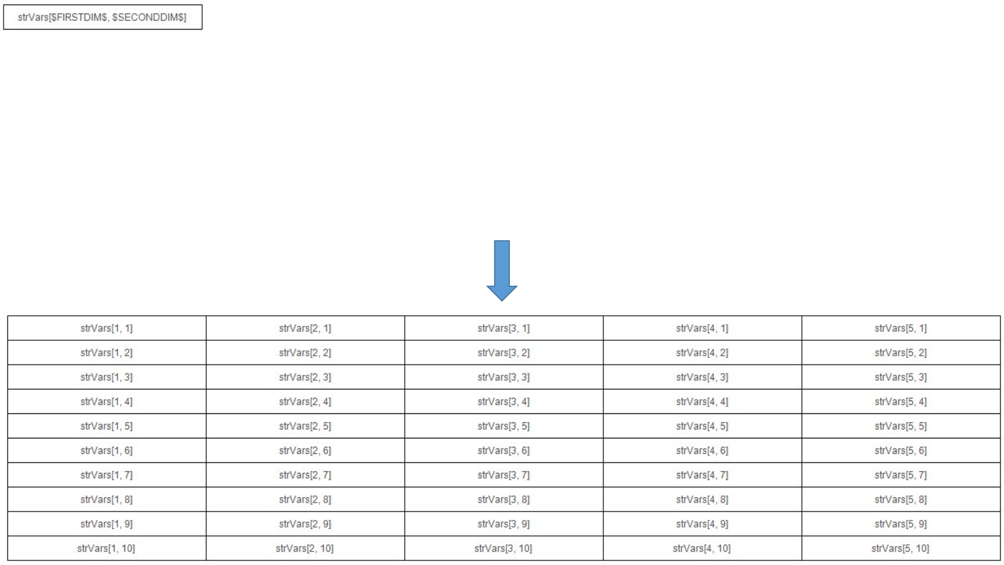
- Declaration（GVL）

```
strVars    : ARRAY [1..5, 1..10] OF STRING;
```

- Setting Properties on Rectangle Objects

Item1	Item2	Description
Text	Text	strVars[$FIRSTDIM$, $SECONDDIM$]

- The result of matrix expansion with default settings



strVars[$FIRSTDIM$, $SECONDDIM$]

strVars[1, 1]	strVars[2, 1]	strVars[3, 1]	strVars[4, 1]	strVars[5, 1]
strVars[1, 2]	strVars[2, 2]	strVars[3, 2]	strVars[4, 2]	strVars[5, 2]
strVars[1, 3]	strVars[2, 3]	strVars[3, 3]	strVars[4, 3]	strVars[5, 3]
strVars[1, 4]	strVars[2, 4]	strVars[3, 4]	strVars[4, 4]	strVars[5, 4]
strVars[1, 5]	strVars[2, 5]	strVars[3, 5]	strVars[4, 5]	strVars[5, 5]
strVars[1, 6]	strVars[2, 6]	strVars[3, 6]	strVars[4, 6]	strVars[5, 6]
strVars[1, 7]	strVars[2, 7]	strVars[3, 7]	strVars[4, 7]	strVars[5, 7]
strVars[1, 8]	strVars[2, 8]	strVars[3, 8]	strVars[4, 8]	strVars[5, 8]
strVars[1, 9]	strVars[2, 9]	strVars[3, 9]	strVars[4, 9]	strVars[5, 9]
strVars[1, 10]	strVars[2, 10]	strVars[3, 10]	strVars[4, 10]	strVars[5, 10]

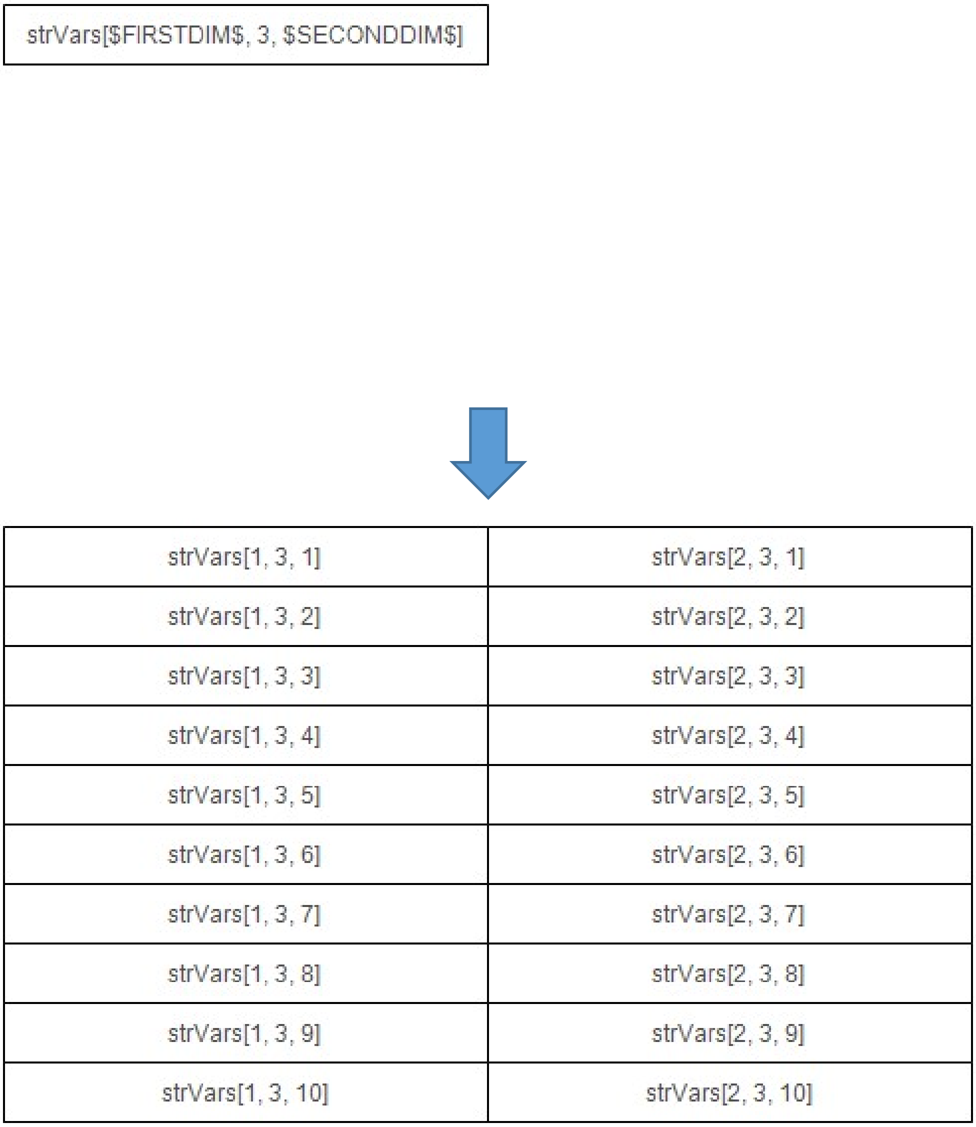
- Example: A2 × 5 × 10 STRING type array declared in GVL is expanded with a Rectangleobject.
  - Declaration（GVL）

```
strVars    : ARRAY [1..2, 1..5, 1..10] OF STRING;
```

  - Setting Properties on Rectangle Objects

Item1	Item2	Description
Text	Txet	strVars[$FIRSTDIM$, 3, $SECONDDIM$]

  - The result of matrix expansion with default settings



strVars[$FIRSTDIM$, 3, $SECONDDIM$]

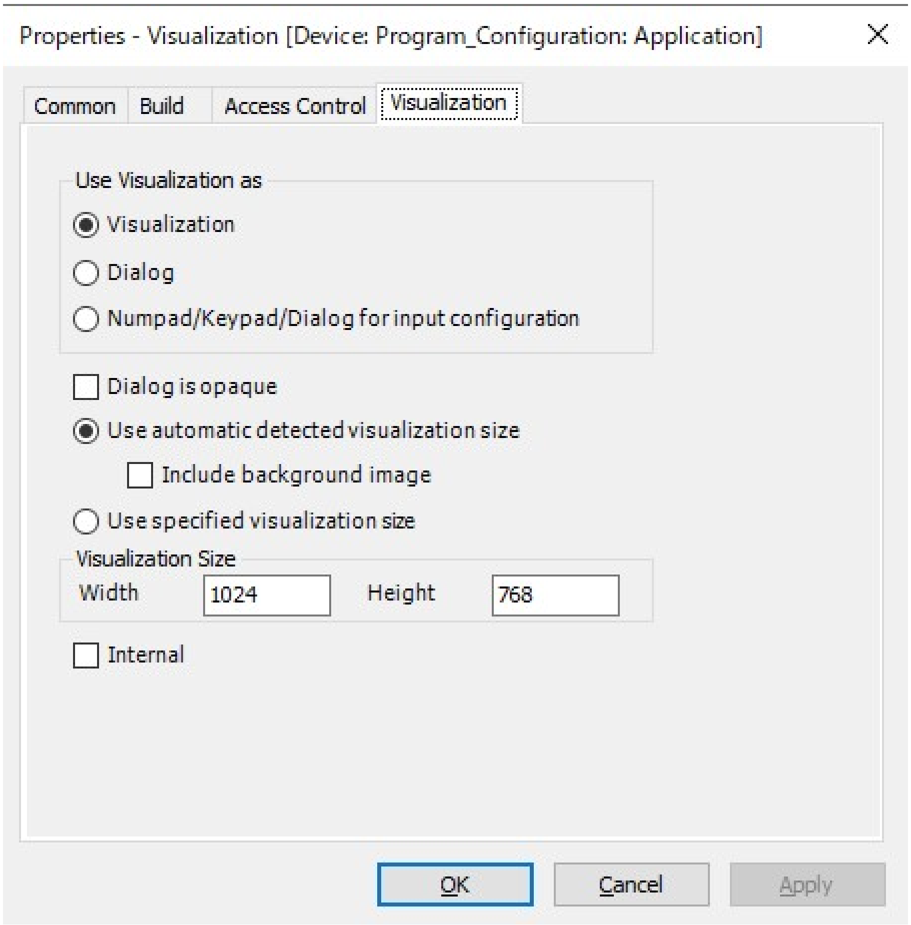
strVars[1, 3, 1]	strVars[2, 3, 1]
strVars[1, 3, 2]	strVars[2, 3, 2]
strVars[1, 3, 3]	strVars[2, 3, 3]
strVars[1, 3, 4]	strVars[2, 3, 4]
strVars[1, 3, 5]	strVars[2, 3, 5]
strVars[1, 3, 6]	strVars[2, 3, 6]
strVars[1, 3, 7]	strVars[2, 3, 7]
strVars[1, 3, 8]	strVars[2, 3, 8]
strVars[1, 3, 9]	strVars[2, 3, 9]
strVars[1, 3, 10]	strVars[2, 3, 10]

Visualization Properties

Visualization Properties can be set to use the created Visualization as a dialog.

■ Properties List

Right-click on the Visualization object in the Navigator window and select "Properties" from the menu that appears. Select the "Visualization" tab in the Properties window.



Properties - Visualization [Device: Program_Configuration: Application]

Use Visualization as

● Visualization

○ Dialog

○ Numpad/Keypad/Dialog for input configuration

□ Dialog is opaque

● Use automatic detected visualization size

□ Include background image

○ Use specified visualization size

Visualization Size

Width 1024 Height 768

□ Internal

Item		Overview
Use Visualization as	Visualization	Use as normal window.
	Dialog	Use as a Pop-up window.
	Numpad/keypad/Dialog for input configuration	Do not use.
Dialog is opaque		Do not use.
Use automatic detected visualization size		All tool elements will be sized and displayed as they appear.
Include backgraund image		Do not use.
Use specified visualization size		The visualization will be displayed in the size specified in the "Height" and "Width" fields.
Internal		Do not use.

Visualization Option

The Visualization Option allows you to configure grid display and other settings.

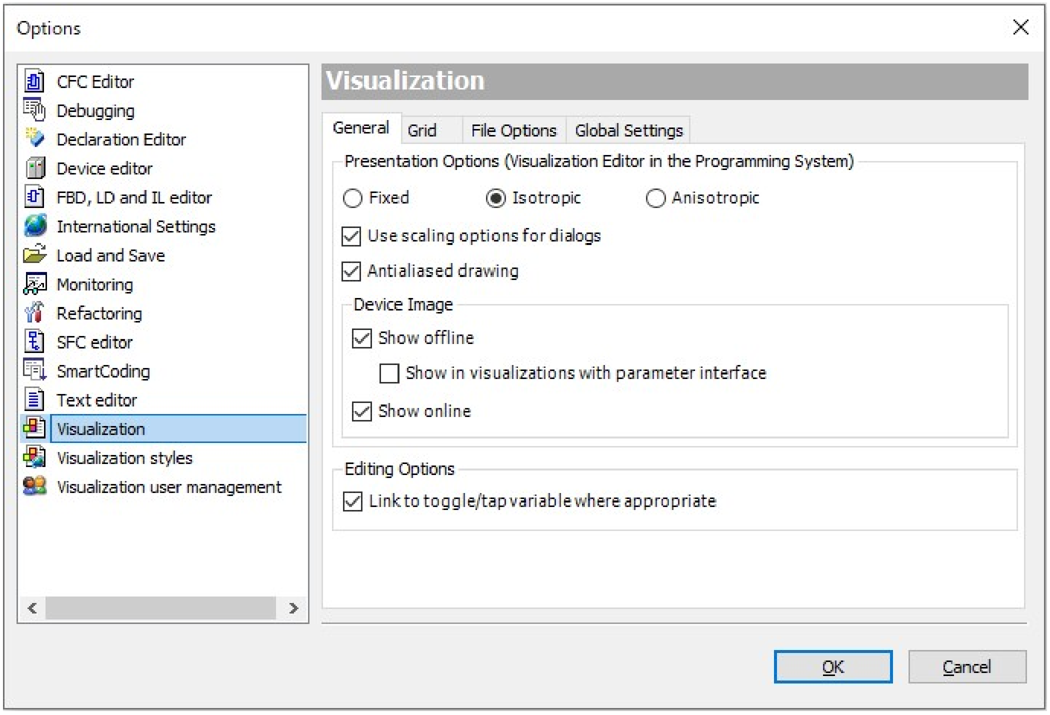
■ Option list

It is displayed by selecting **Tool**>**Option** from the menu bar.

Option Category

Category name	Function
Visualization styles	Do not use.
Visualization user management	
Visualization	Configure settings such as the display of Visualization.

- Category:Visualization
  General Tab



Presentation Options (Visualization Editor in the Programming System)

○ Fixed ● Isotropic ○ Anisotropic

☑ Use scaling options for dialogs

☑ Antialiased drawing

Device Image

☑ Show offline

□ Show in visualizations with parameter interface

☑ Show online

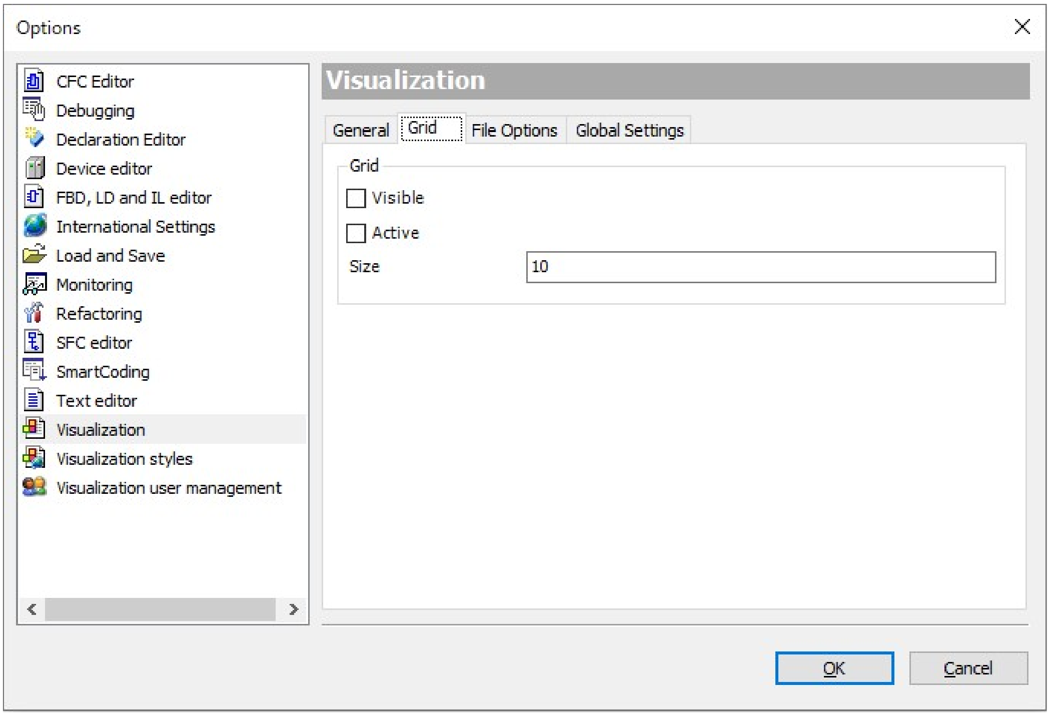
Editing Options

☑ Link to toggle/tap variable where appropriate

Setting items		Overview
Presentation Options	Fixed	Visualization is displayed in its original size.
	Isotropic	Visualization will be scaled to fit on the screen while maintaining the aspect ratio.
	Anisotropic	Visualization will be scaled to fit within the screen, but the aspect ratio will not be preserved.
	Use scaling options for dialogs	If checked, the Isotropic / Anisotropic setting will also be applied to dialogs displayed by the input configuration.
Device Image	Show offline	Do not use.
	Show in visualizations with parameter interface	
	Show online	

Setting items		Overview
Editing Options	Link to toggle/tap variable where approproate	If checked, the variables set in [Input configuration] - [Tap] or [Toggle] will be automatically applied to the tool element property [Color variables] - [Toggle color]. The variable set to [Tap] will be used first.

Grid Tab



Grid

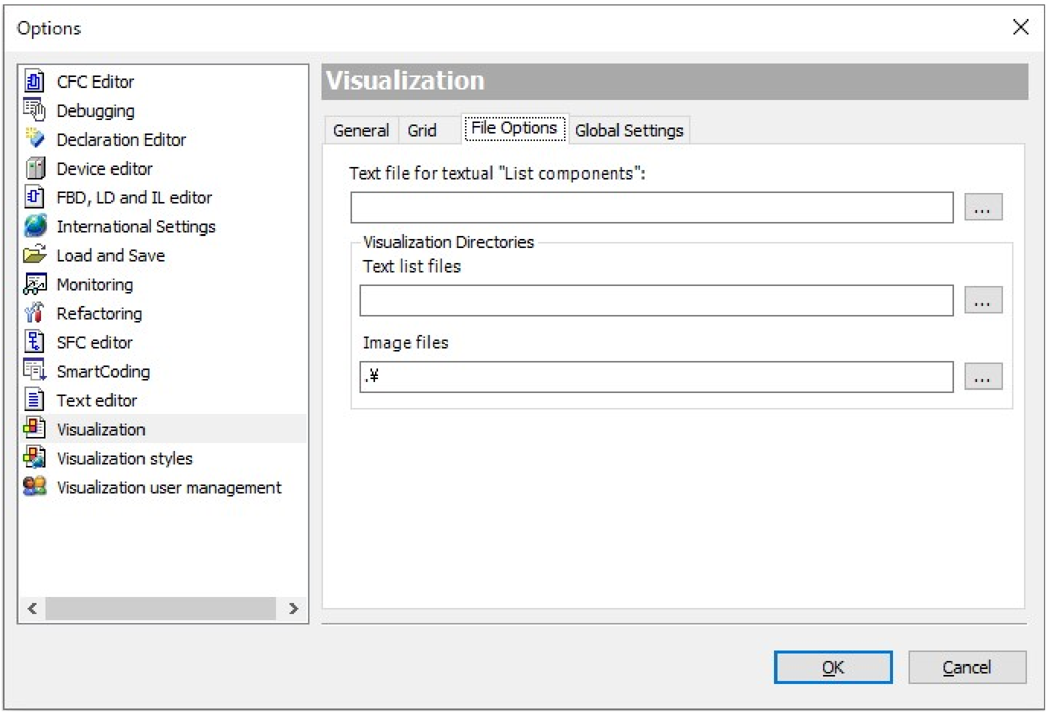
□ Visible

□ Active

Size 10

Setting items	Overview
Visible	If checked, grid points are displayed in Visualization.
Active	If checked, placement of tool elements in Visualization is restricted to grid points.
Size	Specify the distance between grid points in pixels.

File Options Tab



Text file for textual "List components":

Visualization Directories

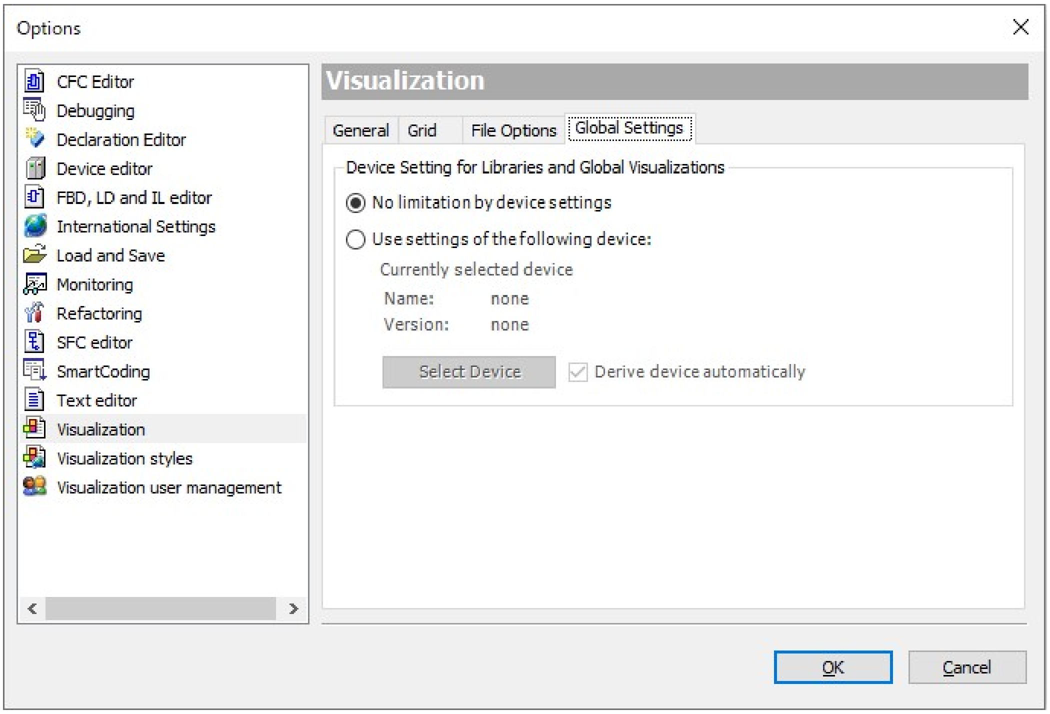
Text list files

Image files

.¥

Setting items	Overview
Text list files for textual “List components”	Set up a .csv file in text list format. When entering text for tool element properties, the text in the text list file will be displayed as input candidates.
Text list files	Do not use.
Image files	

Global Settings Tab



Options

Visualization

General | Grid | File Options | Global Settings

Device Setting for Libraries and Global Visualizations

● No limitation by device settings

○ Use settings of the following device:

Currently selected device

Name: none

Version: none

☑ Derive device automatically

Setting items	Overview
No limitation by device settings	Fixed to selected state
Currently selected device	Do not use.
Select Device	
Derive device automatically	

15.5.2 Setting of TextList

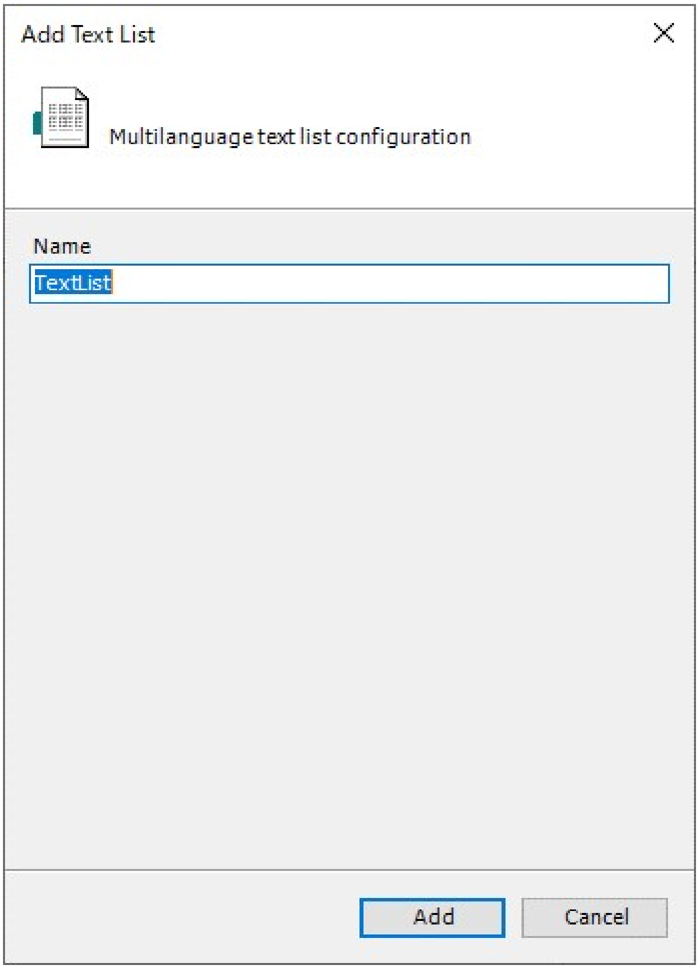
TextList is an object that lists texts identified with ID numbers. When using TextList to create visualization, add a TextList object.

The following section explains how to add and set a TextList object.

1 2 Procedure

1. Right-click the “Application” object in the navigation pane and then select **Add Object**>**TextList** from the context-sensitive menu that is displayed.

   The “Add TextList” dialog box will be displayed.



Add Text List

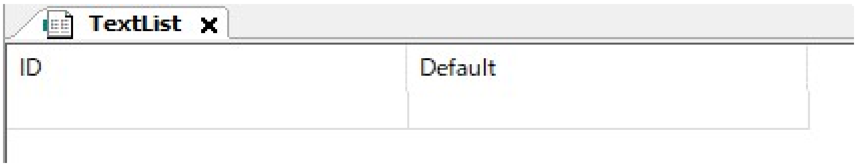
Multilanguage text list configuration

Name

TextList

2. Enter a desired TextList name and click the [Add] button.

   When the addition is completed, the TextList pane will be displayed.

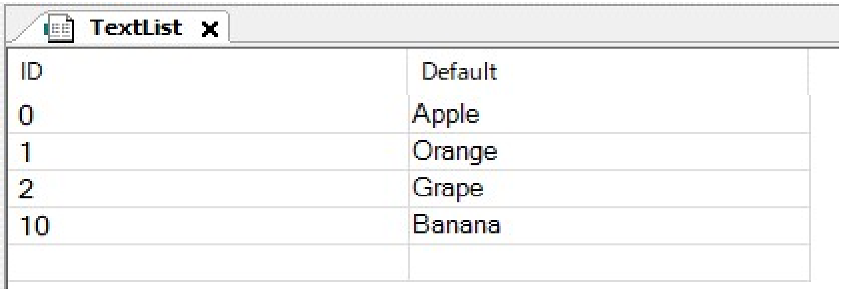


ID	Default

The TextList pane displays the following contents.

Name	Function
ID	Set the ID number for the text described in “Default”. The same ID number cannot be set.
Default	Set the text to be listed. If the text is set with “ID” left blank, an ID is automatically set.

3. Set the TextList object.
   Shown below is an example of the display when "Apple", "Orange", and "Grape” are described in this order in the “Default” field, 10 is assigned as an ID number, and “Banana” is described in the “Default” field.

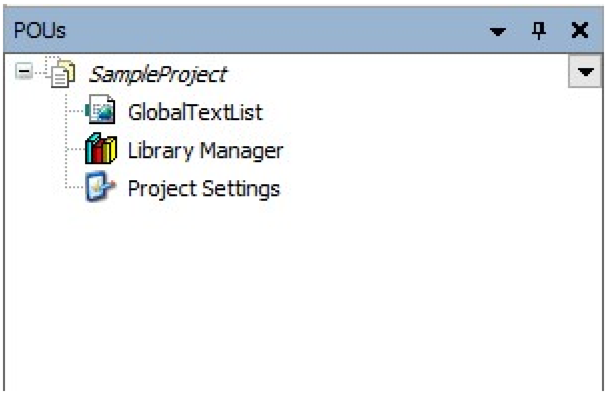


ID	Default
0	Apple
1	Orange
2	Grape
10	Banana

This completes the TextList setting procedure.

GlobalTextList

GlobalTextList is automatically added to the POU tree when you set the [Text] property or other properties of a tool element.



Name	Function
ID	● When the tool element [Text] - [Text] is set: The [TextID] value is automatically set. ● When the tool element [Text] - [ToolTip] is set: The [HintID] value is automatically set.
Default	When the tool element [Text] - [Text] and [Text] - [Tooltip] are set: The text is automatically described.

Info.

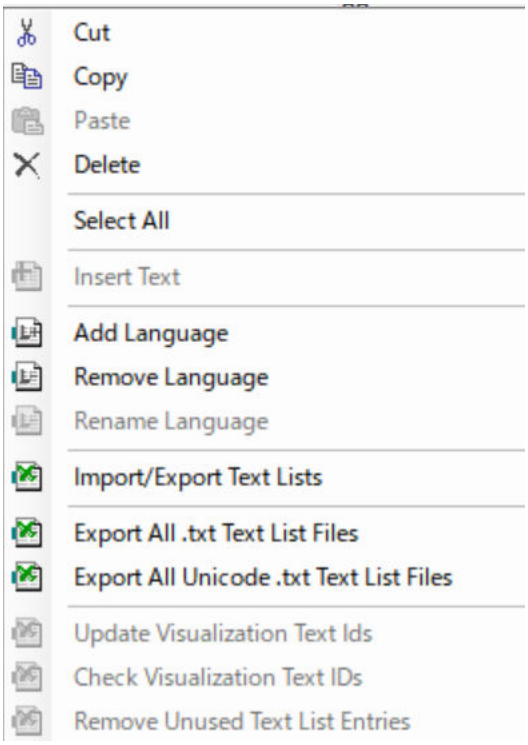
- The contents of the GlobalTextList cannot be edited.
- By using the "Import Replace File" described below, you can replace the text of all tool elements associated with GlobalTextList at once.

TextList Menu

The TextList Menu allows you to add other languages and import and export TextLists.

■ TextList Menu

Select "TextList" from the menu bar or right-click on the TextList object screen.


Cut
Copy
Paste
Delete
Select All
Insert Text
Add Language
Remove Language
Rename Language
Import/Export Text Lists
Export All .txt Text List Files
Export All Unicode .txt Text List Files
Update Visualization Text Ids
Check Visualization Text IDs
Remove Unused Text List Entries

Items	Overview
Cut	Cut the selected text.
Copy	Copy the selected text.

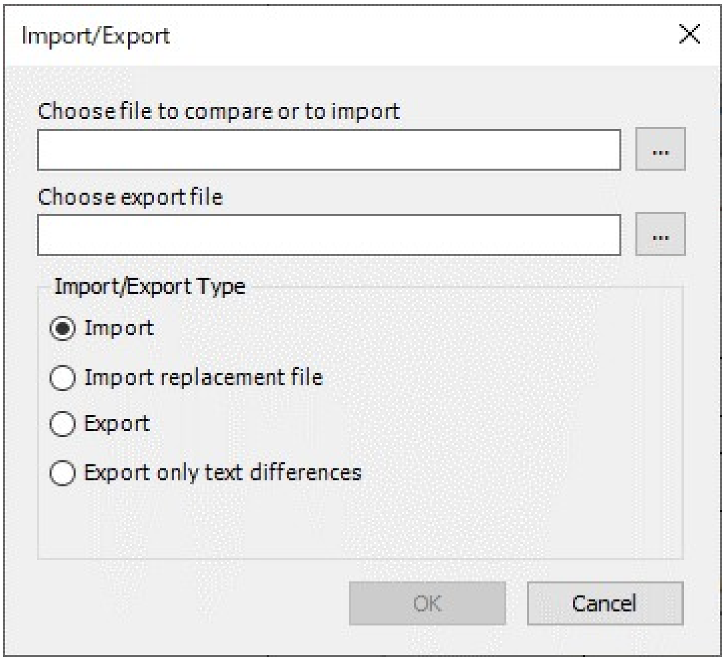
Items	Overview
Paste	Paste the copied text.
Delete	Delete selected text.
Select All	ID, selects all text.
Browse	Do not use.
Insert Text	Insert text. The "ID" will be set automatically.
Add Language	Add a new language column.
Remove Language	Delete selected language. You cannot delete the "default" language.
Rename Language	Rename the selected language. "Default" cannot be renamed.
Import/Export Text Lists	Import/Export TextList. See below for details.
Export All .txt Text List Files	Do not use.
Export All Unicode.txt Text List Files	
Update Visualization Text Ids	Add the text currently used in the visualization to the GlobalTextList. Can be used on a GlobalTextList.
Chek Visualization Text IDs	Do not use.
Remove Unused TexT List Entries	Remove text from the GlobalTextList that is not currently used in the visualization. Can be used on a GlobalTextList.

i Info.

- By adding or writing other languages to TextList, languages other than the default language can be displayed in the visualization.
- You can change the language displayed in the visualization by setting the "Language switch" in the "Input configuration" property of the tool element.

■ About Import/Export

TextList can be imported and exported for saving and reuse. In the "Import/Export TextList" menu of the TextList Menu, set and execute the following items



Import/Export

Choose file to compare or to import

Choose export file

Import/Export Type

● Import

○ Import replacement file

○ Export

○ Export only text differences

Setting Item		Overview
Choose file to compare or to import		Specify a tab-delimited CSV file to import. ● The contents of the CSV file to be imported will be reflected in all TextLists in the project. ● If a CSV file contains a TextList name that does not exist, a TextList will be generated in the device tree.
Choose export file		Specify CSV file to export TextList. ● All TextList and GlobalTextList in the project are exported. ● If the specified file does not exist, a new tab-delimited CSV file is created.
Import/Export type	Import	Import the specified tab-delimited CSV file into TextList Incorrect CSV files, such as using the same ID number or the same language name, will not be imported correctly.
	Import repacement file	Replaces the contents of GlobalTextList based on the contents of the specified replacement file. The replacement file must follow the specified format, and an error will occur if a file with a different format is specified. For details, please refer to the example below.
	Export	Export TextList to a specified tab-delimited CSV file.
	Export only text differences	Do not use.

Info.

- When "Import" is executed on GlobalTextList, the new TextList by TextList file is imported into the POU tree.

■ Example 1: "Export" is executed

The following TextList exists in the project, and in the visualization object there is a button object with the text 'ON'.
The "Export" is performed on one TextList object.

- TextList

TextList name	ID	Default	ja
TextList1	0	Apple	りんご
	1	Orange	みかん
	2	Grape	ぶどう
TextList2	0	Lemon	れもん
	1	Cherry	さくらんぼ
	2	Peach	もも

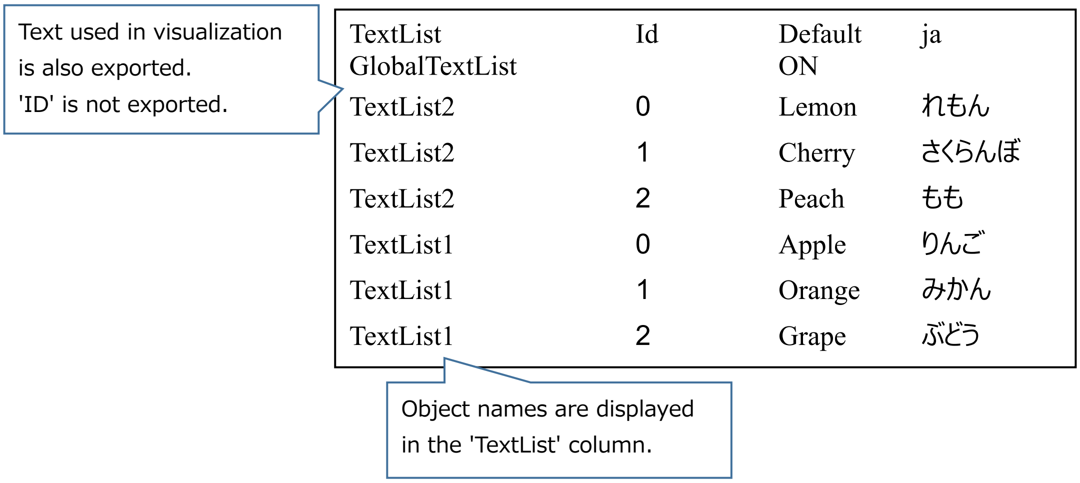
- Visualization

Tool element	Item1	Item2	Description
Button	Text	Text	ON

- GlobalTextList

TextList name	ID	Default	ja
GlobalTextList	21	ON	

The result of "Export" is as follows



```
TextList          Id        Default    ja
GlobalTextList              ON
TextList2         0         Lemon      れもん
TextList2         1         Cherry     さくらんぼ
TextList2         2         Peach      もも
TextList1         0         Apple      りんご
TextList1         1         Orange     みかん
TextList1         2         Grape      ぶどう
```

Text used in visualization is also exported.
'ID' is not exported.

Object names are displayed in the 'TextList' column.

Example 2: "Importing a replacement file

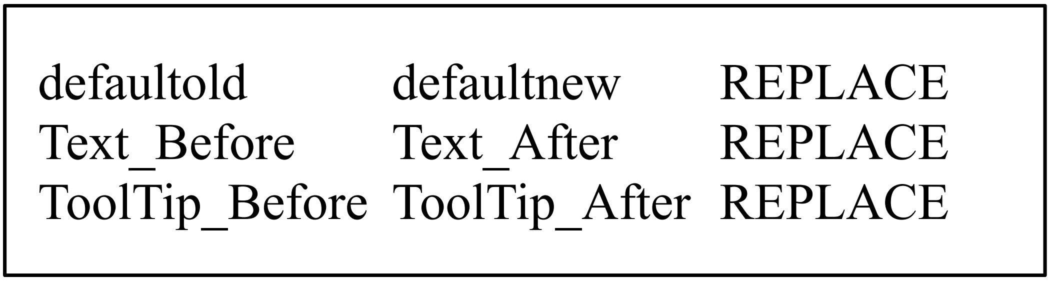
Change the contents of GlobalTextList using a replacement file.

- Before replacement

ID	Default
836	Text_Before
124	ToolTip_Before

- Replacement files

  Each is created as a tab-delimited file.

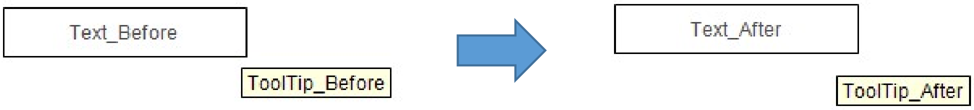


```
defaultold	defaultnew	REPLACE
Text_Before	Text_After	REPLACE
ToolTip_Before	ToolTip_After	REPLACE
```

- After replacement

ID	Default
836	Text_After
124	ToolTip_After

- Visualization



15.5.3 List of Available Tool Elements

■ Tool element

The following tool elements can be used with the GM Programmer.

Class	Name	Description
Basic	Rectangle	Displays a rectangle.
	Rounded rectangle	Displays a rounded rectangle
	Ellipse	Displays an ellipse.
	Line	Displays a straight line. Place a line by clicking two points as the both ends of the straight line.
	Polygon	Displays a polygon.
	Polyline	Displays a polyline.
	Bezier curve	Displays a Bezier curve.
	Pie	Displays a pie.
	Frame	The visualization created in the project will be displayed as a frame. By setting the frame switching variable, you can switch between multiple visualizations.
Common controls	Label	Displays a text. The text cannot be rewritten in the online mode.
	Tab control	The visualization created in the project will be displayed as a frame that can be switched with tabs.
	Button	Displays a button. The button can be used to switch between TRUE and FALSE of the BOOL type variable.
	Group box	Displays a box for grouping tool elements. Group name can be set.
	Table	Data array is displayed as a table. Data such as one-dimensional array, two-dimensional array, array of arrays, structures, or function blocks can be set.
	Text field	Displays a text box. The text can be rewritten even in the online mode.
	Scroll bar	Displays a scroll bar. Variable value can be set according to the scroll bar position.
	Slider	Displays a slider bar. Variable value can be set according to the slider position.
	Spin control	Displays a spin control. The variable value can be increased or decreased by a specified numerical width each time one of the arrow buttons is pressed.
	Invisible input	A rectangle object that is not displayed in the online mode can be added.
	Check box	Displays a check box.

Class	Name	Description
		Depending on whether it is checked or not, the BOOL type variable is switched between TRUE and FALSE
	Radio button	Displays radio buttons grouped together. The number of the selected radio button is identified by an integer value.
	Progress bar	Displays a variable value in the progress bar.
Measurement controls	Bar display	Displays a variable value in the bar.
	Meter 90°	Displays a variable value with a meter.
	Meter 180°	
	Meter	
	Potentiometer	Displays a variable value with a potentiometer. The value can be changed by moving the pointer position.
	Histogram	One-dimensional array data is displayed as a histogram.
Lamps/ Switches/ Bitmaps	Lamp	Displays a lamp. Displays a BOOL type variable value with the lamp turned ON or OFF. TRUE: Lamp is ON., FALSE: Lamp is OFF
	DIP switch	Displays a DIP switch. It can be used to switch between TRUE and FALSE of the BOOL type variable.
	Power switch	
	Push switch	
	Push switch LED	
	Rocker switch	
	Rotary switch	
Date and time control	Analog clock	Displays an analog clock.

15.5.4 List of Properties of Tool Elements

The following properties can be used with the GM Programmer. If the same property set values exist in more than one tool element items, repeated descriptions are omitted.

■ Class: Basic

Setting item		Overview	Tool element
Element name		Name of the tool element. The name is assigned automatically.	Common
Type of element		The object name of the tool element is displayed.	
Center		Set the coordinates of the center point of the tool element.	
Color	Normal state Alarm state	Set the frame color and fill color for the normal state and alarm state and also set the transparency of each one.	
Use gradient color		If this item is checked, the object is filled with the contents of [Gradient setting].	

Setting item		Overview	Tool element
Gradient setting		Set the fill gradation of the tool element. On the "Gradient editor" dialog box, specify the gradation type, color, etc.	
Element look	Line width	Set the frame line width.	
	Fill attributes	Set the fill format of the tool element. Filled, unfilled	
	Line style	Set the line style. Solid line, dashed line, dotted line, dash-dotted line, dash-dotted-dotted line, outline line	
Texts	Text	Set the text to be displayed on the tool element. When using text variables, specify the numerical format.	
	Tooltip	Set the text to be displayed on the tooltip of the tool element. When using tooltip variables, specify the numerical format.	
Text properties	Horizontal alignment Vertical alignment	Set the position where the text is displayed.	
	Text format	Set the display method when the text protrudes from the tool element.	
	Font	Set the font and size of the text.	
	Font color	Set the font color and transparency of the text.	
Absolute motion	Movement	Specify the distance from the current coordinates in the online mode.	
	Rotation	Set the angle to rotate around the center point in the online mode.	
	Scaling	Set the reduce/enlarge ratio from the center point in the online mode. The original size is 1000.	
	Use REAL values	If this item is checked, the [Absolute motion] value is set as a REAL type value. Therefore, the movement becomes smoother than when the item is not checked.	
Relative move	Movement top-left Movement bottom-right	Specify the distance that the Movement top-left or the Movement bottom-right of the tool element moves in the online mode. Since it is a relative move, the size of the tool element changes.	
Text variable	Text variable	Set the variable of the text to be displayed on the tool element.	
	Tooltip variable	Set the text variable to be displayed when a mouse cursor is placed over the tool element.	
Dynamic text	TextList	Set the TextList name to be used.	
	Text index	Set the TextList ID. The content of the corresponding “ID” is displayed in the text.	

Setting item		Overview	Tool element
	Tooltip index	Set the TextList ID. The content of the corresponding "ID" is displayed on the tooltip.	
Font variables	Font name	Set a STRING type variable.	
	Size	Set the numerical data type variable. Units: <pt> = point, <px> = pixel	
	Flags	Set a DWORD type variable. The following display formats can be used and their combined use is possible. ● 2#0001: Italic ● 2#0010: Bold ● 2#0100: Underline ● 2#1000: Strikethrough	
	Character set	Do not use.	
	Color	Set a DWORD type variable. 16#[Transparent_8bit][R_8bit][G_8bit][B_8bit]	
	Flags for text alignment	Set a DWORD type variable. The following display formats can be used and their combined use is possible. ● 2#0000: Align top and left ● 2#0001: Center horizontally ● 2#0010: Align right ● 2#0100: Center vertically ● 2#1000: Align bottom	
Color variables	Toggle color	Set a BOOL type variable for switching colors. In the initial state, the variable set in [Tap] or [Switching] for [Input configuration] is linked. ● TRUE: Alarm state settings are used. ● FALSE: Normal state settings are used.	
	Normal state Alarm state	Set the frame color and fill color for the normal state and alarm state using variables.	
Look variables	Line width	Set the frame line width using variables.	
	Fill attributes	Set the fill format of the tool element using variables. ● 0: Fill ● 1: Unfill	
	Line style	Set the line style using variables. ● 1: Solid line ● 2: Dashed line ● 3: Dash-dotted line ● 4: Dash-dotted-dotted line ● 5: Outline line	
Status variables	Invisible	Set the BOOL type variable that switches between Show and Hide of the tool element in the online mode. When set to TRUE, the tool element is hidden.	
	Deactivate inputs	Set the BOOL type variable that switches between Enable and Disable of the character input or mouse operation in the online mode.	

Setting item		Overview	Tool element
		When set to TRUE, mouse operation is disabled.	
Input configuration	OnDialogClosed	When the dialog is closed, the event is raised.	
	OnMouseClick	An event is fired when the mouse button is pressed and released on a tool element.	
	OnMouseDown	An event occurs when the mouse button is pressed on a tool element.	
	OnMouseEnter	An event occurs when the mouse pointer enters the tool element.	
	OnMouseLeave	An event occurs when the mouse pointer leaves the tool element.	
	OnMouseMove	Do not use.	
	OnMouseUp	An event occurs when the mouse button is released on the tool element (regardless of where it is pressed) or when the mouse button is pressed and released on the tool element (regardless of where it is released).	
	Tap	● Variable: Set the BOOL type variable that switches between TRUE and FALSE while the mouse remains clicked on the tool element. ● If “Change to FALSE using Tap” is checked, the set variable switches from TRUE to FALSE while the mouse remains clicked. ● If “Tap enter if captured” is checked, the tool element is considered to be tapped even when the mouse is placed over the tool element while the mouse remains clicked after it is clicked in the area outside the tool element.	
	Toggle	● Variable: Set the BOOL type variable that switches between TRUE and FALSE when the mouse is clicked on the tool element. ● If “Tap up arrow if captured” is checked, the variable changes even when the mouse click is released in an area outside the tool element while the mouse remains clicked after it is clicked on the tool element.	
	Hot key	Do not use.	
Position		Set the coordinates, width, or height of the tool element. Operate the tool element using the mouse to change the coordinates, width, or height.	Common
		Set the coordinates of each vertex of the tool element. Operate the tool element using the mouse to change the vertex coordinates.	Polyline Bezier curve Polygon
Radius setting	Radius	Set the style of the corner rounding radius. ● From the style: The corner is rounded to the radius that matches the visualization style. ● Proportional to the element size: The corner is rounded to the radius that matches the size of a rounded corner rectangle object. ● Explicit: The corner is rounded to the set radius.	Rounded rectangle
Line width variable	Integral value	Set the line width using variables.	Line

Setting item		Overview	Tool element
Line style variable	Integral value	Set the line style using variables.	
Begin End		Set the begin and end angles of the pie.	Pie
Variable for begin Variable for end		Set the variables that specify begin and end angles of a sector.	
Only show cirde line		If this item is checked, only the portion of an arc is displayed.	
Dynamic points	Array of points	Do not use.	Polyline Bezier curve Polygon
	Number of points	Do not use.	
Clipping		If you check the box, only the visualization within the frame of the tool element will be displayed.	Frame
Show frame		You can set the visibility of the frame of the tool element.	
Scaling type		You can set the type of zoom for the image: ● Isotropic: The visualization will be scaled while maintaining the aspect ratio of the image. ● Anisotropic: The visualization will be scaled without maintaining the aspect ratio of the image. ● Fixed: Only the size of the frame of the tool element will be changed. The visualization will be displayed in its original size. ● Fixed and Scrollable: Only the part of the visualization within the frame of the tool element will be displayed. You can scroll to view different parts using scrollbars.。	
Scroll bar settings	Left-Right Scroll Position Variable Vertical Scroll Position Variable	Do not use.	
Deactivate background drawing		Do not use.	
Reference		Set the visualization to display on the frame object.	
Switch frame variable	variable	Sets the index number of the visualization to display on the frame object. Integer type variables can be set.	

■ Class: General control

Setting item		Overview	Tool element
Value, variable, data array		Set the variable, array, etc.used by the tool element.	Common
Tab Width		Set the tab width.	Tab controll
Tab height		Set the tab height	
References	Header	Set the heading text to be displayed on the tab.	
	ImageID	Do not use.	
Bitmap info	Static ID	Do not use.	Button
Bitmap ID variable			

Setting item		Overview	Tool element
Columns	Show row header	Set the title of the show row header.	Table
	Show colum header	Set the width of the column.	
	Image column	Do not use.	
	Image configuration	Do not use.	
	Alignment of headline text	Set the position of the headline text.	
	Use template	If this item is checked, [Templates] is displayed. In the online mode, cells are drawn in the style according to the settings set in [Templates].	
	Alignment of template headline text	If this item is checked, the headline is drawn in the style according to the settings set in [Templates] in the online mode.	
Maximum array index		Set the maximum number of items of each data array to be displayed.	
Row height		Set the row height.	
Scroll bar size		Set the scroll bar size.	
Display the row header Display the column header		If this item is checked, the row header and column header are displayed.	
Row header width		Set the width of the row header.	
Selection	selection color	Set the display color of the selected cell.	
	Selection font color	Set the font color of the selected cell.	
	Selection type	Set the type of the cell selection.	
	Frame around selected cells.	If this item is checked, the border of the selected cell becomes thicker.	
	Variable for selected column Variable for selected row	Set the variable that indicates the number of the selected columns or rows.	
	Variable for valid column selection Variable for row valid selection	Set the BOOL type variable that reflects the selected state of column or row.	
Shadow type		Set the shadow type of the border.	Text field
Selection and caret configuration	Caret position	Set the variable that indicates the position of the caret (that displays the insertion point).	
	Selection start Selection end	Set the variable that indicates the selection start position and selection end position when multiple texts are selected.	
	All selected	Set the BOOL type variable for selecting all texts. When set to TRUE, all texts are selected.	
Minimum value Maximum value		Set the maximum value and the minimum value of the bar.	Scroll bar Progress bar

Setting item		Overview	Tool element
Page size		Set the movement amount of a value when the scroll area is clicked.	Scroll bar Slider
Move to click		If this item is checked, the tab moves to the position (value) where the mouse is clicked in the scroll area.	
Bar	Diagram type (slider only)	Set the position where the scale is displayed.	
	Orientation	Set the orientation of the scale.	
	Running direction	Set the direction in which the scale values increase.	
Scale	Show scale	If this item is checked, the graduations are displayed.	Slider
	Scale start Scale end	Set the maximum value and the minimum value of the scale.	
	Main scale	Set the interval between the main scale graduations.	
	Sub scale	Set the interval between the sub scale graduations. If set to 0, the sub scale is disabled.	
	Scale format	Set the numerical value format.	
	Scale proportion	Set the ratio (%) occupied by the scale to the entire object size. The default is 0.	
Number format		Set the numerical value format.	Spin control
Interval		Set the interval in which the scale value increases or decreases when the arrow button is pressed.	
Value range	Minimum value Maximum value	Set the maximum value and the minimum value.	
Text properties	Usage of	If this item is checked, select whether to use the default for the text style or to individually set it.	Check box Radio button
Number of columns		Set the number of columns for displaying radio buttons.	Radio button
Radio button order		Set the radio button placement order. ● From left to right: Radio buttons are placed in order from the left top to the right direction for the number of set columns. ● From top to bottom: Radio buttons are placed from the left top to the bottom direction equally for the number of set columns.	
Radio button settings	Radio button	● New creation of buttons: Newly create radio buttons. ● Area: Individually set the text or tooltip to be displayed on the radio button. Set the offset value that shifts the radio button downward for the line height (pixel). ● Delete button: Used to delete the selected radio button.	

■ Class: Measurement control

Setting item		Overview	Tool element
Background	Image color	Set the image color of the tool element.	Bar display Meter Meter 90° Meter 180° Potentiometer
	Own Image	Set the background image. ● Image: Use in the initial state (< default >). ● Transparent color: Set a color that makes the background transparent.	
	Optimized drawing	Please use the initial state (with the checkbox checked).	
Scale	Refer also to scale settings under “Class: General control”.		
	Sub scale position	Set the position where the sub scale is displayed.	
	Scale type	Set the type of graduations indicated on the scale. Line, dot, square	
	Scale line width	Set the width of the scale graduation line.	
	Scale color	Set the color of the scale graduation line.	
	scale in 3D	If this item is checked, a gray shadow is added to each graduation of the scale.	
	Element frame	If this item is checked, a frame line is displayed outside the tool element.	
	Inside frame Outside frame	If this item is checked, a frame line is displayed inside and outside of the scale.	
	Base line	Set the baseline value for the histogram to be displayed.	Histogram
Bar	Refer also to bar settings under “Class: General control”.		Bar display
	Optimum size for bar	If this item is checked, the bar width is expanded to match the element size.	
Positioning	Horizontal offset Vertical offset	The scale position is offset by the specified value from the original position. If a positive value is specified, the position is offset to the right direction and downward direction.	
	Horizontal scaling Vertical scaling	The scale size is enlarged or reduced by the specified value from th original size. If a positive value is specified, the size in the horizontal direction is reduced to the center direction and the one in the vertical direction is reduced to the downward direction.	
Arrow	Arrow type	Set the type of the arrow mark.	Meter 90° Meter 180° Meter Potentiometer
	Color	Set the color of the arrow mark.	
	Angle range	Set the direction of the meter.	
	Aditional arrow	If this item is checked, an arrow mark is added to the scale portion of the meter.	
Subrange array	Use subrange	Do not use.	Histogram
	Start Index End Index		
Display type		Set the type of the histogram.	

Setting item		Overview	Tool element
		Bar, line, curve	
Line width		Set the line width (pixel) of the histogram to be displayed.	
Relative bar width		Set the bar width of the histogram relative to the graph area.	
Color - Color areas	Dorable Color areas	If checked, the colors within the range set in the color areas will always be displayed in the tool element. If neither "Persistent Color areas" nor "Use Color areas on Scale" is checked, the tool element will display the set color when the value in the "Value" field is within the range of the color range.	Bar display Meter 90° Meter 180° Meter Potentiometer Histogram
	Use color areas for scale	If checked, the colors within the range set in the color areas will be applied to the scale and number parts of the tool element. If unchecked, the color areas settings will be applied to the background of the scale when the checkbox is not selected.	
Color - color areas -Areas	Begin of area	Set the starting value of the color area.	
	End of areas	Set the ending value of the color area.	
	Color	Set the display color of the color area.	

■ Class: Lamps/Switches/Bitmaps

Setting item		Overview	Tool element
Image setting	Transparent Transparent color	If [Transparent] is checked, the color set in [Transparent color] becomes transparent.	Dip switch Power switch Push switch LED switch Rocker switch Rotary switch
	Isotropic type	You can set the type of zoom: ● Isotropic: The image will be scaled while maintaining the aspect ratio of the width and height. ● Anisotropic: The image will be scaled without maintaining the aspect ratio of the width and height. ● Fixed: Only the size of the frame will be changed. The elements inside will remain at a fixed size.	
	Horizontal alignment Vertical alignment	The display will appear when the "Isotropic" option is selected in the Isotropic Type. Set the position of the elements within the frame.	
Element behavior		Set the switch operation to “Toggle” or “Tap”. ● Image toggler: Toggles between TRUE and FALSE of the set “Variable” every time the tool element is pressed. ● Image tapper: The set “Variable” remains TRUE while the tool element is pressed.	

■ Class: Date and time control

Setting item		Overview	Tool element
Display time	Use system time	If this item is checked, the system time is displayed.	Analog clock

Setting item		Overview	Tool element
		If this item is not checked, the time of the TOD type variable set in the displayed [Variable] is displayed.	

■ Numerical value format

The following table shows numerical value formats that can be set in properties.

Numerical data type	Numerical value format	Overview
Integer type	%d	Decimal notation
	%o	Unsigned integer in octal notation
	%x	32-bit unsigned integer in hexadecimal notation (with the letters a to f in lower case)
	%X	32-bit unsigned integer in hexadecimal notation (with the letters A to F in upper case)
	%u	Unsigned integer in decimal notation
Float	%f	Decimal notation with decimal point
	%e	Exponent notation
Time	%t[d] %t[dd]	Date display ● %t[d]: 1 to 31 ● %t[dd]: 01 to 31 (with 0 prefix if single digit)
	%t[H] %t[HH] %t[h] %t[hh]	Hour display ● %t[H]: 1 to 24 ● %t[HH]: 01 to 24 (with 0 prefix if single digit) ● %t[h]: 0 to 12 ● %t[hh]: 00 to 12 (with 0 prefix if single digit)
	%t[m] %t[mm]	Minute display ● %t[m]: 0 to 59 ● %t[mm]: 00 to 59 (with 0 prefix if single digit)
	%t[s] %t[ss]	Second display ● %t[s]: 0 to 59 ● %t[ss]: 00 to 59 (with 0 prefix if single digit)
	%t[ms] %t[us] %t[ns]	Milli/micro/nano-second display ● %t[ms]: 0 to 999 ● %t[us]: 0 to 999 ● %t[ns]: 0 to 999
	%t[yyyy] %t[yy] %t[y]	Year display ● %t[yyyy]: The year is displayed in four digits. ● %t[yy]: The year is displayed in two digits (00 to 99) ● %t[y]: The year is displayed in one or two digits (0 to 99)
	%t[MMMM] %t[MMM] %t[MM] %t[M]	Month display ● %t[MMMM]: The month is displayed in English (Example: January) ● %t[MMM]: The month is displayed in abbreviated English (Example: Jan) ● %t[MM]: The month is displayed in two digits (01 to 12)

Numerical data type	Numerical value format	Overview
		● %t[M]: The month is displayed in one or two digits (1 to 12)
	%t[ddddd] %t[dddd] %t[ddd] %t[dd] %t[d]	Day display ● %t[ddddd]: The day of the week is displayed in numerals (1 = Monday, …, 7 = Sunday) ● %t[dddd]: The day of the week is displayed in English (Example: Monday) ● %t[ddd]: The day of the week is displayed in abbreviated English (Example: Mon) ● %t[dd]: The date is displayed in two digits (01 to 31) ● %t[d]: The date is displayed in one or two digits (1 to 31)
	%t[jjj]	Day of year display %t[jjj] = 001 to 366
String	%s	The character string is displayed in STRING type or WSTRING type.
Other	%%	The percent symbol is displayed. The percent notation can be displayed by adding another numerical value format to the beginning. (Example: Specifying %d %% will display "decimal number" %.)
	%c	Hexadecimal notation character is displayed in ASCII. Example: "16#41" is displayed as “A” in the tool element.

■ Regarding the main property settings of the tool elements

Below are the main properties and their corresponding settings for each tool element.

Tool Element	Property	Setting Value
Lamp	Variable	BOOL type variable
DIP Switch Power Switch Push switch Push switch LED Rocker Switch Rotary switch		
Checkbox		
scroll bar Bar display Meter 90° Meter 180° Meter	Value	Numeric variables
Slider Spin Control Potentiometer Radio button Progress bar	Variable	
Analog Clock	Variable	TOD type variables
Table	Data arrays	Array type variables

Tool Element	Property	Setting Value
Histogram		
Rectangle Round rectangle Ellipse Pie Button	Input configuration	Set in each Input configuration

Info.

- About the display priority of input text

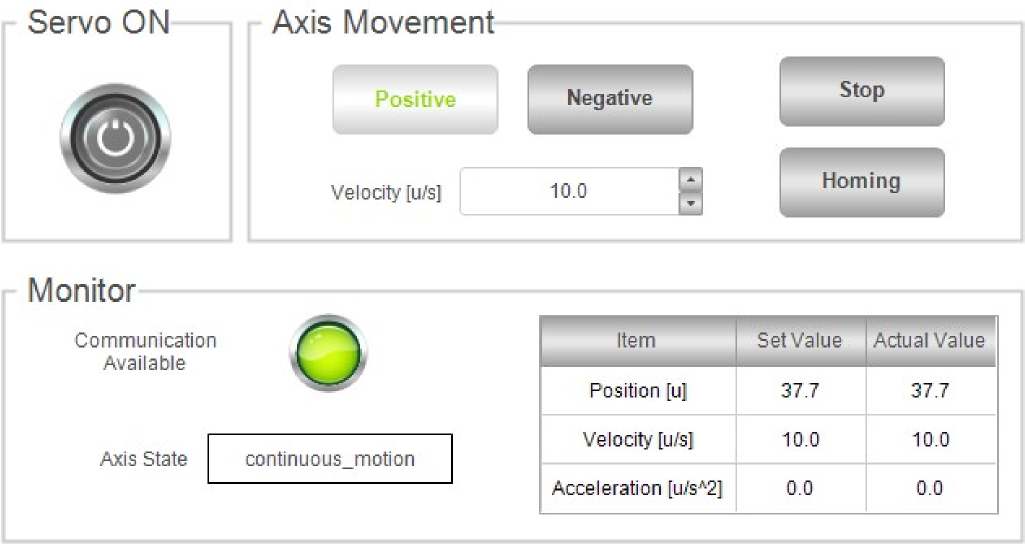
  Tool elements, such as Rectangle, have several display settings: Text, Text Variable, Dynamic Text (text index) and Tooltip, Tooltip Variable, and Dynamic Text (tooltip index). If you set them at the same time, the display order is fixed.

Setting item 1	Setting item 2	Items to display
Texts	Text variable	Texts
Tooltip	Tooltip variable	Tooltip
Texts	Dynamic text（Text index）	Dynamic text
Tooltip	Dynamic text（Tooltip index）	Dynamic text
Text variable	Dynamic text（Text index）	Dynamic text
Tooltip variable	Dynamic text（Tooltip index）	Dynamic text

15.5.5 Visualization examples

Visualization Example1：axis movement and Moniter window

Create a monitor window to execute a jog or axis stop and to retrieve each parameter of the axis.



Item	Set Value	Actual Value
Position [u]	37.7	37.7
Velocity [u/s]	10.0	10.0
Acceleration [u/s^2]	0.0	0.0

1 2 Procedure

1. Place the tool elements required for visualization.
   The tool elements used for this visualization are as follows.

Tool element	volume
Rectangle	1
Label	3
Button	4
Group box	3
Table	1
Spin control	1
Lamp	1
Power switch	1

2. Set "Text" for each Tool element.

Tool element	Setting item1	Setting item2	Description
Rectangle	Text	Text	%s
Label1	Text	Text	Velocity [u/s]
Label2	Text	Text	Communication Available
Label3	Text	Text	Axis State
Button1	Text	Text	Positive
Button2	Text	Text	Negative
Button3	Text	Text	Stop
Button4	Text	Text	Homing

Tool element	Setting item1	Setting item2	Description
Group box1	Text	Text	Servo ON
Group box2	Text	Text	Axis Movement
Group box3	Text	Text	Monitor

3. Write a program that works with Visualization.
Add a POU object to the project named 'MC_PRG' and include the following

- Declaration section

```
// Motion
MC_Power_0    : MC_Power;
bRegON        : BOOL;

MC_Jog_0    : MC_Jog;
bForward    : BOOL;
bBackward    : BOOL;
lrVel        : LREAL:=10;
lrAcc        : LREAL:=100;
lrDec        : LREAL:=200;

MC_Stop_0    : MC_Stop;
bStop        : BOOL;

ETC_CO_SdoWrite_0    : ETC_CO_SdoWrite;
bExe_SdoWrite        : BOOL;
wIndexW                : WORD;
bySubIndexW            : BYTE;
diWriteData            : DINT;
eWriteSize            : CAA.SIZE;

MC_Home_0    : MC_Home;
bExe_Home    : BOOL;
bHome        : BOOL;
iStep        : INT;

// Monitor
alrParameters    : ARRAY [1..3] OF ARRAY [1..3] OF LREAL:=[[0,1,2],[0,0
,0],[0,0,0]];
```

- Implementation section

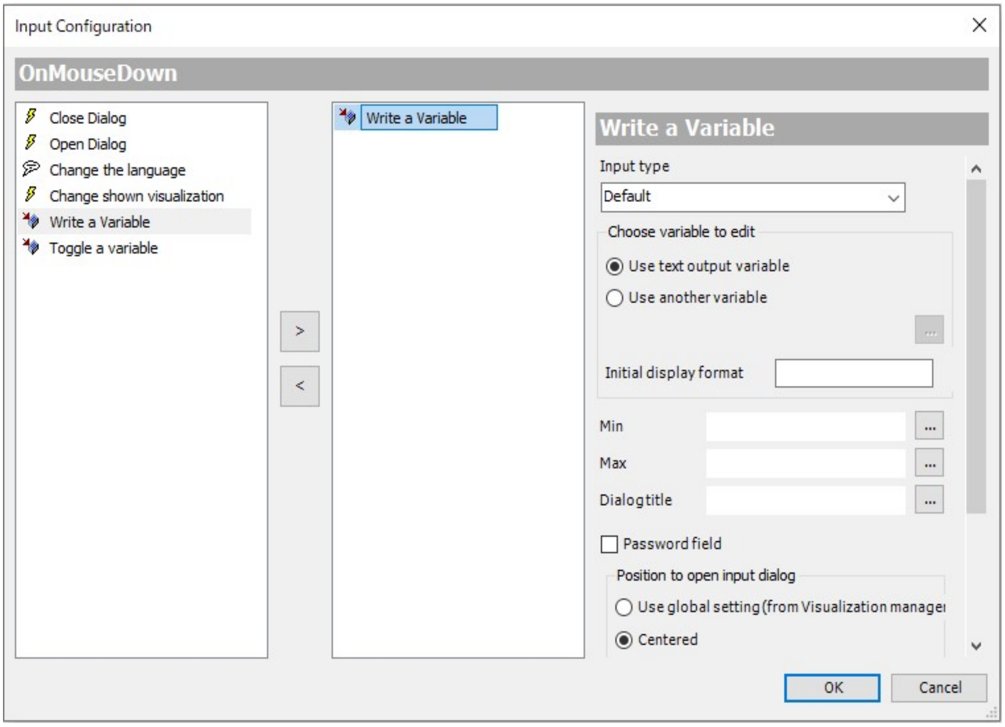
```
MC_Power_0(
    Axis:=Axis1,
    Enable:=TRUE,
    bRegulatorOn:=bRegON,
    bDriveStart:=TRUE,
);

MC_Jog_0(
    Axis:=Axis1,
    JogForward:=bForward,
    JogBackward:=bBackward,
    Velocity:=lrVel,
    Acceleration:=lrAcc,
    Deceleration:=lrDec,
);

MC_Stop_0(
    Axis:=Axis1,
    Execute:=bStop,
    Deceleration:=10000,
);

ETC_CO_SdoWrite_0(
    xExecute:=bExe_SdoWrite,
    uiDevice:=1001,
    wIndex:=wIndexW,
    bySubIndex:=bySubIndexW,
    udiTimeOut:=5000,
    pBuffer:=ADR(diWriteData),
    szSize:=eWriteSize
);

MC_Home_0(
    Axis:=Axis1,
    Execute:=bExe_Home,
    Position:=0,
);

// MC_Home Sequence
CASE iStep OF
    0:
        diWriteData:=16#21;
        wIndexW:=16#6098;
        eWriteSize:=1;
        IF bHome = TRUE THEN
            bExe_SdoWrite:=TRUE;
        END_IF
        IF ETC_CO_SdoWrite_0.xDone = TRUE THEN
            bExe_SdoWrite:=FALSE;
            iStep:=1;
        END_IF
    1:
        diWriteData:=16#D55550;
        wIndexW:=16#6099;
        bySubIndexW:=1;
        eWriteSize:=4;
        bExe_SdoWrite:=TRUE;
        IF ETC_CO_SdoWrite_0.xDone = TRUE THEN
            bExe_SdoWrite:=FALSE;
            iStep:=2;
        END_IF
    2:
        diWriteData:=16#155550;
        wIndexW:=16#6099;
        bySubIndexW:=2;
        eWriteSize:=4;
        bExe_SdoWrite:=TRUE;
        IF ETC_CO_SdoWrite_0.xDone = TRUE THEN
            bExe_SdoWrite:=FALSE;
            iStep:=3;
        END_IF
    3:
        diWriteData:=16#F42400;
        wIndexW:=16#609A;
        bySubIndexW:=0;
        eWriteSize:=4;
        bExe_SdoWrite:=TRUE;
        IF ETC_CO_SdoWrite_0.xDone = TRUE THEN
            bExe_SdoWrite:=FALSE;
            bExe_Home:=TRUE;
            iStep:=4;
        END_IF
    4:
        IF MC_Home_0.Busy = FALSE THEN
            bHome:=FALSE;
            bExe_Home:=FALSE;
            iStep:=0;
        END_IF
END_CASE

// Monitor Table
alrParameters[2][1]:=Axis1.fSetPosition;
alrParameters[2][2]:=Axis1.fSetVelocity;
alrParameters[2][3]:=Axis1.fSetAcceleration;
alrParameters[3][1]:=Axis1.fActPosition;
alrParameters[3][2]:=Axis1.fActVelocity;
alrParameters[3][3]:=Axis1.fActAcceleration;
```

4. Set the variable required for each Tool element to work.

Tool element	Setting item1	Setting item2	Setting item3	Description
Rectangle	Text variable	Text variable		Axis1.nAxisState
Button1	Input configuration	Toggle	variable	MC_PRG.bForward
Button2	Input configuration	Toggle	variable	MC_PRG.bBackward
Button3	Input configuration	Toggle	variable	MC_PRG.bStop
Button4	Input configuration	Toggle	variable	MC_PRG.bHome
Table	Data arrays			MC_PRG.alrParameters
Spin control	variable			MC_PRG.lrVel
	Number format			%3.1f
	Interval			1
	Value range	Minimum value		1
		Maximum value		500
	Input configuration	OnMouseDown		Refer to the next image for details.
Lamp	variable			Axis1.bCommunication

Tool element	Setting item1	Setting item2	Setting item3	Description
Power switch	variable			MC_PRG.bRegON

- Input configuration of Spin control



Input Configuration

OnMouseDown

- Close Dialog
- Open Dialog
- Change the language
- Change shown visualization
- Write a Variable
- Toggle a variable

Write a Variable

Write a Variable

Input type: Default

Choose variable to edit

● Use text output variable

○ Use another variable

Initial display format

Min

Max

Dialogtitle

□ Password field

Position to open input dialog

○ Use global setting (from Visualization managei

● Centered

5. Create a TextList.

Add the TextList object to the project as 'TextList' and describe the following The contents described here will be used in the Table object.

ID	Default
0	Position [u]
1	Velocity [u/s]
2	Acceleration [u/s^2]

6. Set the properties of the Table.

Set the header of the item to be displayed in the [Show row header] of each [Columns] in the table.

Setting item1	Setting item2	Setting item3	Setting item4	Description
Columns	Columns	[0]	Show row header	Item
		[1]	Show row header	Set Value
		[2]	Show row header	Actual Value

Next, uncheck [Display the row header] and check [Columns] - [Use template] for each column. In the [template] field that appears, set the contents to be displayed in each column of the Table.

- Creation of 'item' column

Setting item1	Setting item2	Setting item3	Setting item4	Description
[0]	template	Dynamic text	TextList	'TextList'
			TextIndex	MC_PRG.alrParameters[1][INDEX]

- Creation of 'Set Value' column

Setting item1	Setting item2	Setting item3	Setting item4	Description
[1]	template	Text	Text	%3.1f

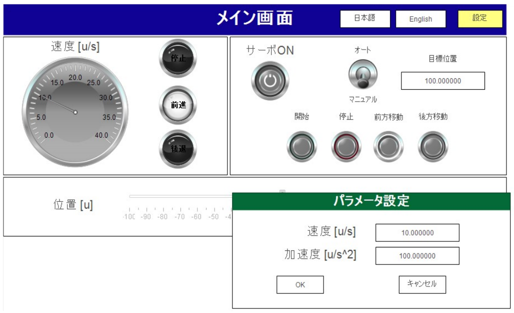
- Creation of 'Actual Value' column

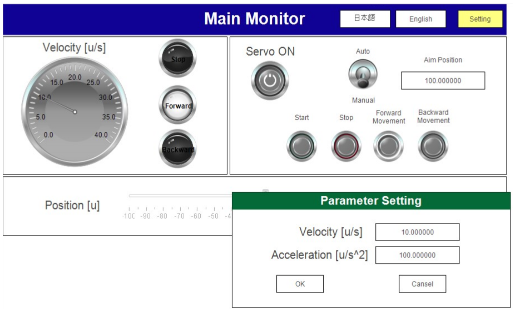
Setting item1	Setting item2	Setting item3	Setting item4	Description
[2]	template	Text	Text	%3.1f

Adjust the alignment, size, etc. of each Tool element.

Visualization Example2：Language Change and Dialog Window

Use the textList language addition to change the language displayed on the Visualization window or to create a parameter setting window using a dialog box. In this example, the language is changed from Japanese to English.

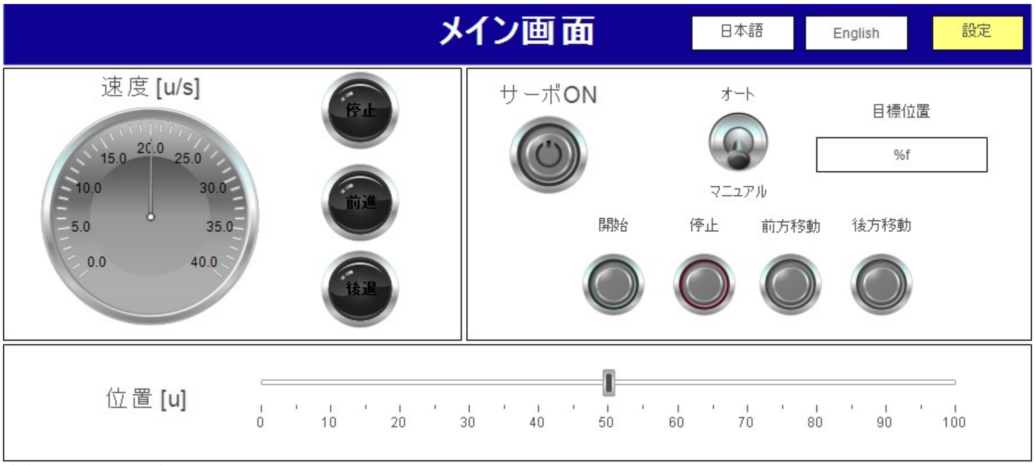




1 2 Procedure

1. Create a Visualization that will serve as the main window.

Add the Visualization to the project and place the necessary Tool elements. Then, set each property (text, variable, etc.).



Of these, the following tool elements are used to change the language and display dialogs.

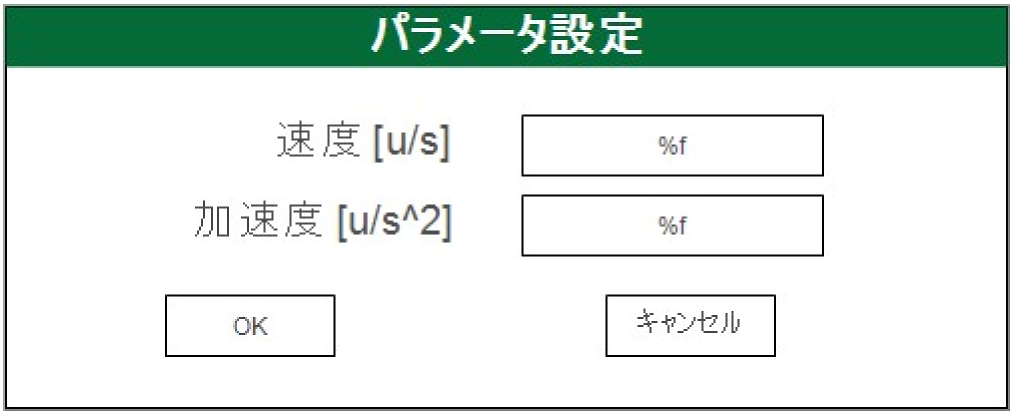
Tool element	Setting item1	Setting item2	Description	purpose
Rectangle	Text	Text	日本語	Language Change
Rectangle	Text	Text	English	Language Change
Rectangle	Text	Text	Setting	Displaying Dialogs

2. Create a Visualization for the dialog window.

Add Visualization to the project and declare the variables to be passed in the dialog in InterFace Edeitor.

```
VAR_IN_OUT
    Velocity     : LREAL;
    Acceleration     : LREAL;
END_VAR
```

Next, place the required Tool element and set its properties.



Descriptions of the main properties are shown below.
The "Text variable" set in Rectangle1 and Rectangle2 refers to the variable declared in the InterFace Edeitor.

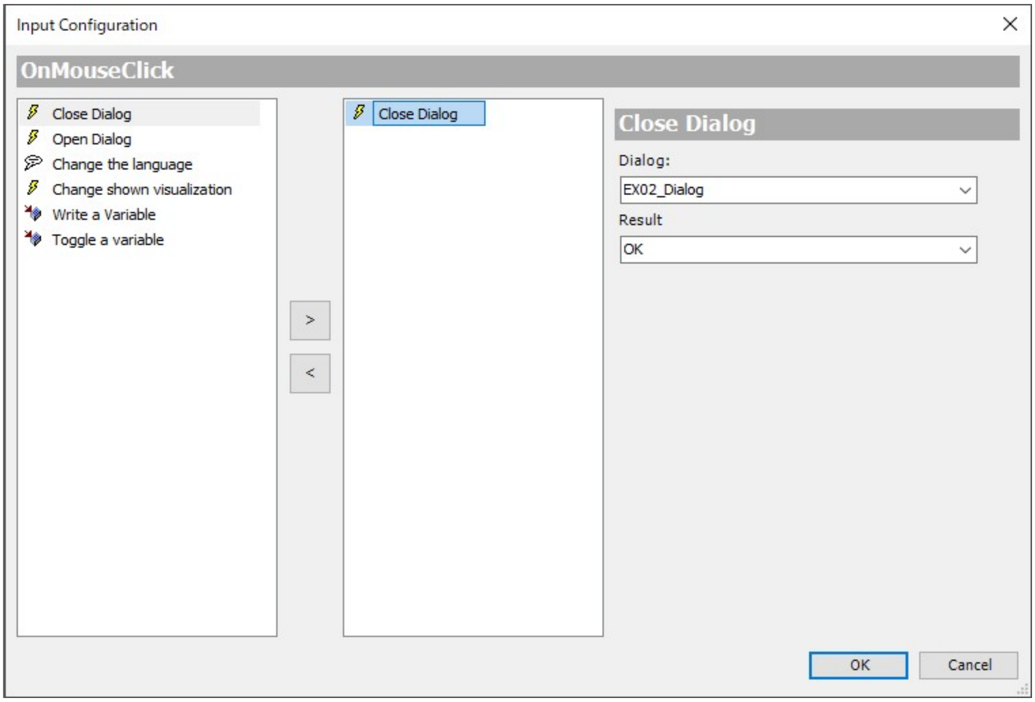
Item	Setting item1	Setting item2	Description
Rectangle1	Text	Text	%f
	Text variable	Text variable	Velocity
	Input configuration	OnMouseClick	Writing Variables
Rectangle2	Text	Text	%f
	Text variable	Text variable	Acceleration
	Input configuration	OnMouseClick	Writing Variables
Rectangle3	Text	Text	OK
	Input configuration	OnMouseClick	dismiss a dialog
Rectangle4	Text	Text	Cansel
	Input configuration	OnMouseClick	dismiss a dialog

Setting background and Visualization properties

Item	Setting item1	Setting item2	Description
Background	Color setting	User color	Enable
			White
Visualization property	Visualization tab	Use Visualization as follows	Dialog
		Use the specified Visualization size	Enable
		Visualization size	Width：500 Height：200

For the Input configuration of Rectangle 3 and Rectangle 4, set "dismiss a dialog".

- For "Dialog," select the Visualization for the created dialog screen. (In this example, the Visualization name for the Dialog is "EX02_Dialog".)
- In the "Result" section, select the text that is to be included in each Rectangle object.



Close Dialog

Dialog: EX02_Dialog

Result: OK

3. Add the language to be changed to TextList.

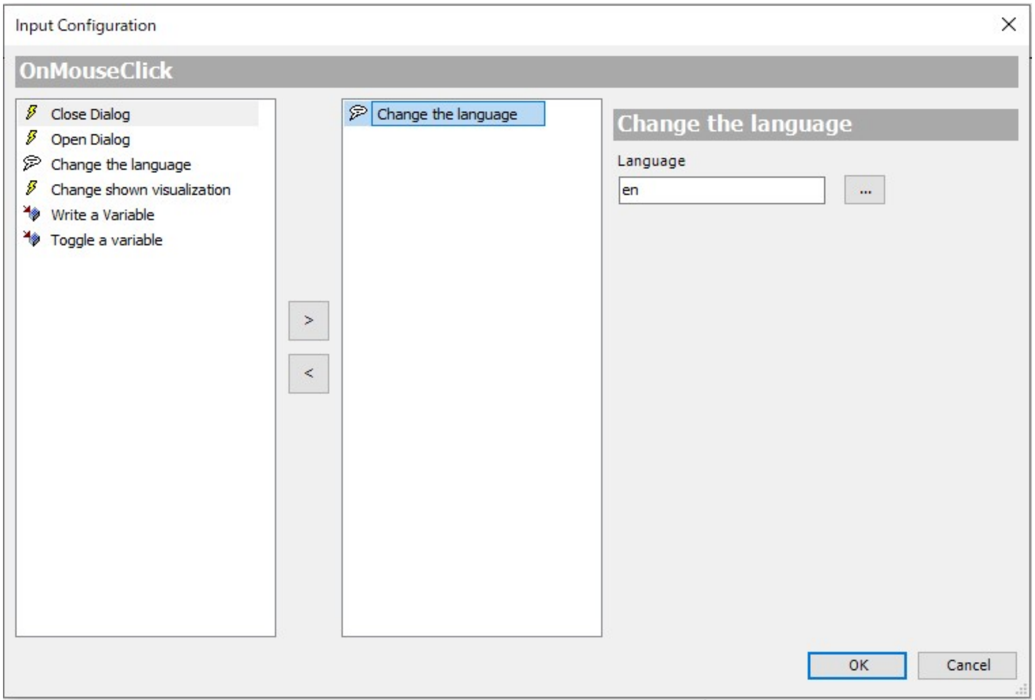
   Add "en" column to GlobalTextList in the device tree. Put 日本語 in the "Default" column and the English translation for the "Default" column in the "en" column.A partial example is shown below.

   - GlobalTextList

ID	Default	en
943	English	
365	キャンセル	Cansel
108	サーボ ON	Servo ON
588	日本語	
238	速度 [u/s]	Velocity [u/s]

4. Set [Input configuration] for language change.

   In the properties of each Rectangle object for language change on the main screen, set the "Change language" action to [Input configuration] - [OnMouseClick].



Change the language

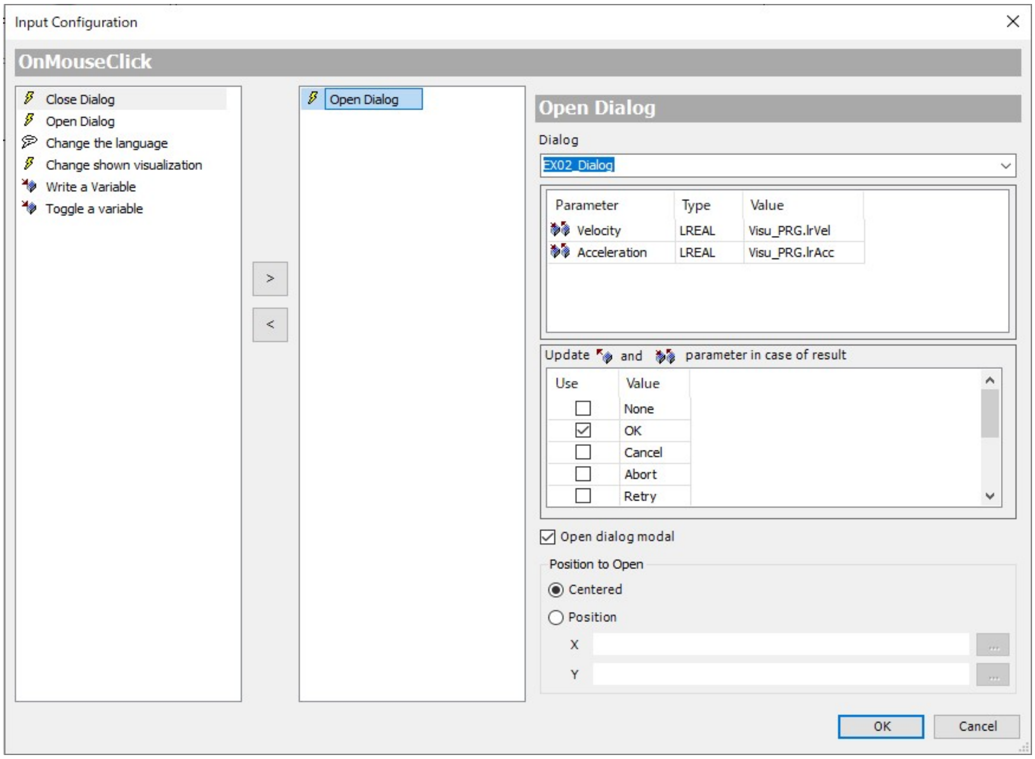
Language: en

Set the language of each Rectangle object as follows. For "日本語," the "Language" field in [Input configuration] should be left blank, since the content is listed in the "Default" column of the TextList.

Text	Language
日本語	
English	en

5. Set [Input configuration] to display a dialog.

   Set the "Open Dialog" action to [Input configuration] - [OnMouseClick] in the properties of the Rectangle object for displaying the main window language.



Input Configuration

OnMouseClick

- Close Dialog
- Open Dialog
- Change the language
- Change shown visualization
- Write a Variable
- Toggle a variable

- Open Dialog

Open Dialog

Dialog

EX02_Dialog

Parameter	Type	Value
Velocity	LREAL	Visu_PRG.lrVel
Acceleration	LREAL	Visu_PRG.lrAcc

Update and parameter in case of result

Use	Value
□	None
☑	OK
□	Cancel
□	Abort
□	Retry

☑ Open dialog modal

Position to Open

● Centered

○ Position

X

Y

- Set the "Dialog" to the Visualization name created in step 2.
- Set "Value" to the variable on the POU that reflects each "Parameter" entered in the dialog.
- For the "Update" part, check the "Value" is "OK" checkbox to reflect the value in the POU only when the "OK" button in the dialog is clicked.
- The other settings are to be used as they are in the initial state.

This completes the settings.

15.6 Firmware Version Upgrade Function

15.6.1 Checking Unit Configuration

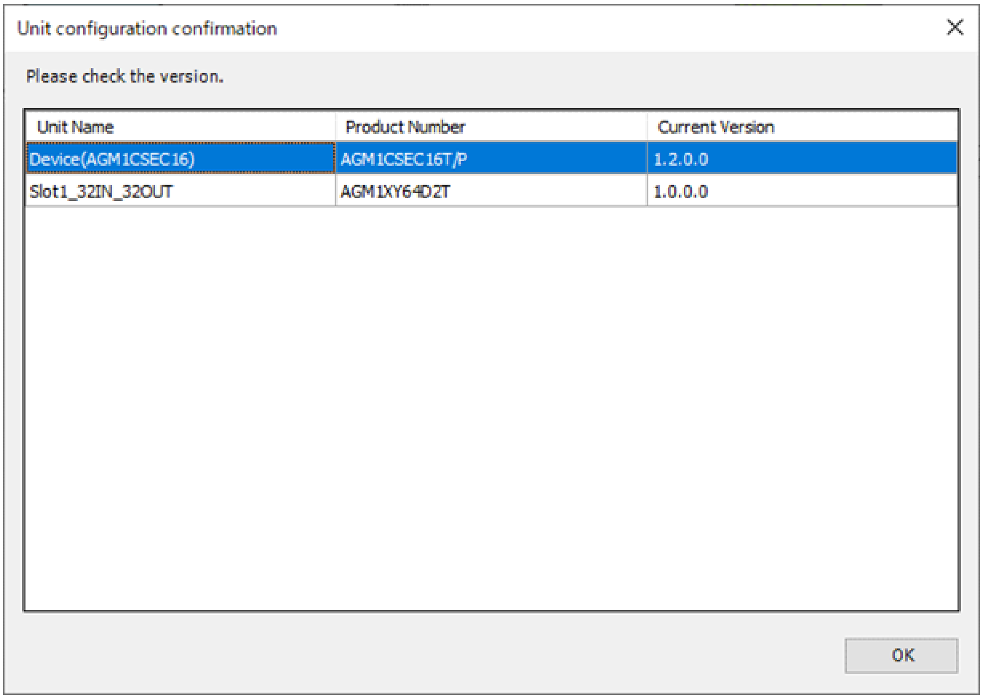
You can check the product numbers and current firmware versions of the GM1 controller and expansion units connected to the controller.

Procedure

1. From the menu bar, select **Online**>**Unit configuration confirmation**.
   The “Select Device” dialog box appears to show a GM1 controller that you can connect.

2. "Unit configuration confirmation"The "Unit configuration confirmation” dialog box appears and displays the product numbers and current versions of the GM1 controller and expansion units connected to the controller.

3. Click the[OK]button.
   The"Unit configuration confirmation"dialog box will be closed.



Unit configuration confirmation

Please check the version.

Unit Name	Product Number	Current Version
Device(AGM1CSEC16)	AGM1CSEC16T/P	1.2.0.0
Slot1_32IN_32OUT	AGM1XY64D2T	1.0.0.0

Note

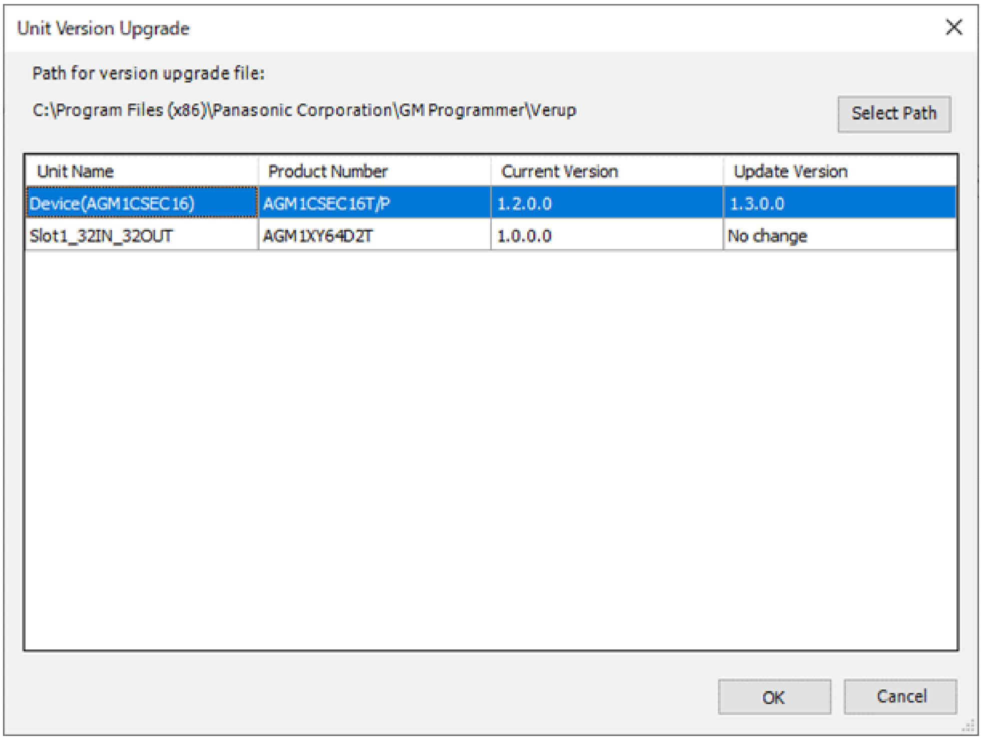
- The configuration of units can also be confirmed by selecting "Basic Operations" on the start page and Unit "Configuration Confirmation".

15.6.2 Unit Version Upgrade

You can upgrade the firmware versions of the GM1 controller and expansion units connected to the controller. The versions of the expansion units as well as the GM1 controller can be upgraded by batch.

Procedure

1. From the menu bar, select **Online**>**Unit Version Upgrade**.
   The “Select Device” dialog box appears to show a GM1 controller that you can connect.
2. Select Device of the GM1 controller and click the [OK] button.
3. "Unit Version Upgrade"The " Unit Version Upgrade” dialog box appears and displays current firmware versions in the Current Version column.
   Latest versions are displayed by default in the Update Version column.



Path for version upgrade file:

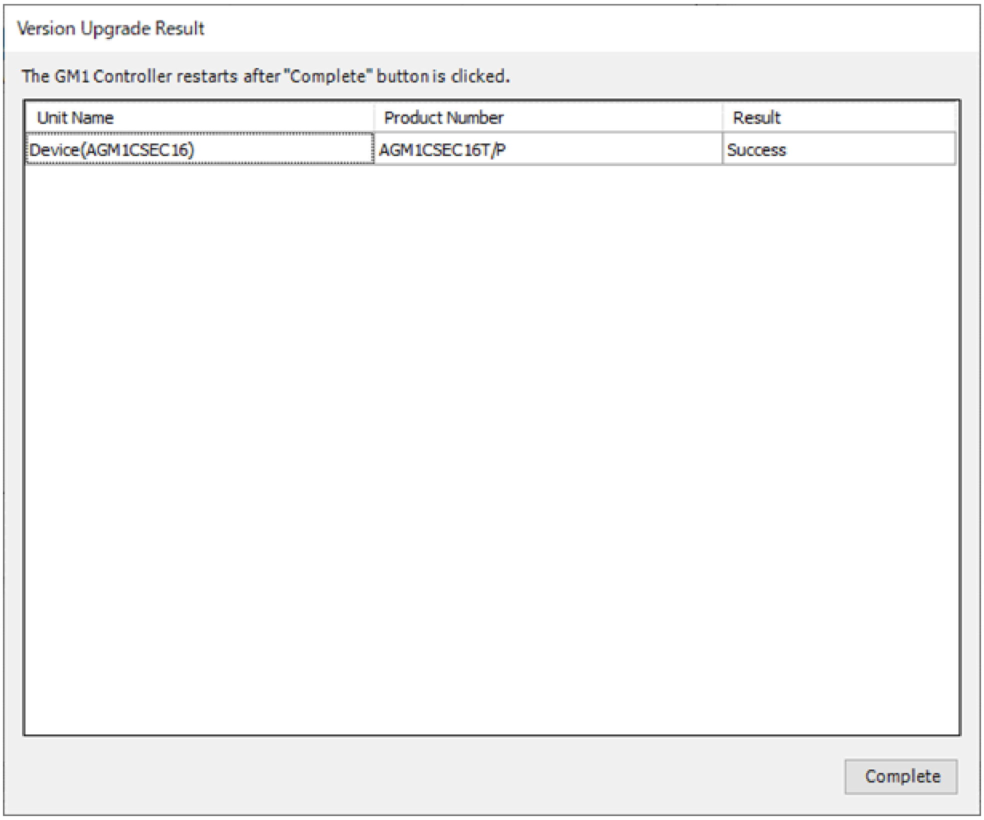
C:\Program Files (x86)\Panasonic Corporation\GM Programmer\Verup

Unit Name	Product Number	Current Version	Update Version
Device(AGM1CSEC16)	AGM1CSEC16T/P	1.2.0.0	1.3.0.0
Slot1_32IN_32OUT	AGM1XY64D2T	1.0.0.0	No change

4. Select the version in the Update Version column.
   To change to a version other than the latest version, select the desired version.
   If the version is not to be upgraded, select"Not change".
5. Set the RUN/STOP switch on the GM1 Controller to STOP.
6. Click the[OK]button.
   A confirmation dialog box will be displayed. Proceed with version upgrading by following the prompts in the dialog box.

After the version upgrading is completed, the"Version Upgrade Result"dialog box appears and displays upgrade results.

Check that every result is"Success"in the Result column.



Version Upgrade Result

The GM1 Controller restarts after "Complete" button is clicked.

Unit Name	Product Number	Result
Device(AGM1CSEC16)	AGM1CSEC16T/P	Success

7. Click the[Complete]button.

   Restart of the GM1 controller will start automatically. When the restart ends, the version upgrading is completed.

Info.

- When the firmware version of the GM1 controller is upgraded, Reset Origin is executed.
- The designation of the path for the version upgrade file can be changed. Click the"Select Path"button and specify a folder where the version upgrade file (.hex) is saved.
- The versions of units can also be upgraded by selecting Basic Operations on the start page and Unit Version Upgrade.

15.7 Project Management Function

15.7.1 What is Project Management Function?

The backup and restore functions for data of each type in the GM1 controller are called project management functions.

■ Backup function

This is a function used to collectively save data of every type in the GM1 controller to an SD memory card. File data to be saved is called a backup file. This function can be executed by any of two methods: operation by the controller and a function block in a program.

■ Restore function

This is a function used to transfer a backup file in an SD memory card to the GM1 controller. Data of every type in the GM1 controller will be replaced by a backup file. This function can be executed by any of two methods: operation by the controller and a function block in a program.

■ Backup file and restoration configuration file

When the backup function is performed, a backup file and a restoration configuration file are created in a predetermined directory on the SD memory card. These files have roles as shown below.

File name	Description	Backup	Restore
Backup file AUTOEXEC.GM1	Data of every type in the GM1 controller is stored in this file.	Creation	Reference
Restoration configuration file AUTOEXEC.INI	This file is used to specify data subject to restoration. This is text data and thus is editable.	Creation	Reference

■ Each type of settings data subject to backup and restoration

Target data	Backup	Restore
Source file	○	○
External file	○	Selectable(Note 1)
Bootstrap application	○	○
RETAIN variable	○	Selectable(Note 1)
Network setting	○	Selectable(Note 1)
Time zone	○	Selectable(Note 1)
Account (device user)	○	○
Certificates	×	×

(Note 1) Whether or not to specify it for a target of restoration can be selected.

■ Method for writing specifications in restoration configuration file

The restoration configuration file is a text file used to specify each type of settings data for a target of restoration when data is restored. The following is an example of specifications written in a restoration configuration file.

```
[AUTOEXEC_OPTION]
                Application_File=yes
                External_File=no
                Retain_File=no
                Network_File=no
                Timezone_File=no
```

Method for specifying each type of settings data

Writing	Description
Application_File	Source file, bootstrap application, and account (device user)
External_File	External file
Retain_File	RETAIN variable
Network_File	Network setting
Timezone_File	Time zone

Method for specifying data for a target of restoration.

Writing	Description
yes	Specify it for a target of restoration
no	Not specify it for a target of restoration

Info.

- Application_File cannot be excluded from the target of restoration. If it is excluded from the target of restoration, an error occurs.
- In writing, do not use spaces around the equals "=" sign. If there is any single- or double-width space around the equals "=" sign, the type of data is regarded as (no), being excluded from the target of restoration.

15.7.2 Backup (Operation by the controller)

This is a function used to collectively save every type of settings data in the GM1 controller to an SD memory card by operating the controller.

To enable this function, open the PLC parameter screen and change project backup of project management setting from "Invalid"→ "Valid". (Refer to "5.2 Setting up the GM1 Controller" for PLC parameter screens)

1 2 Procedure

1. Set the mode switch to STOP.
2. Turn OFF the power to the GM1 controller.
3. Create "AUTO" folder in the root directory of the SD card and empty the folder.
4. Open the cover and insert the SD card into the slot.
5. With the cover left open, turn ON the power to the GM1 controller.
6. When the backup process starts, the STOP lamp starts flashing.
7. When the process is completed normally, the STOP lamp is lit and the ERROR is unlit.
   If an error occurs, the STOP lamp goes off and the ERROR lamp blinks.
8. Close the cover, set the mode switch to RUN, and turn the controller OFF and then ON, so that the controller returns to the original state.

i Info.

- If there is no "AUTO" empty folder, the bag-up process will not start.
- After the backup process is completed, "AUTOEXEC.GM1" project data and the "AUTOEXEC.INI" configuration file are created in the "AUTO" folder.
- If the "AUTOEXEC.INI" configuration file is already contained in the "AUTO" folder at the time of starting the backup process, the configuration file will remain unchanged.

15.7.3 Restore (Operation by the controller)

This is a function used to transfer a backup file in an SD memory card to the GM1 Controller by operating the controller.

To enable this function, open the PLC parameter screen, select project management setting, and change the project backup and project restore setting from "Invalid"→ "Valid". (Refer to "5.2 Setting up the GM1 Controller" for PLC parameter screens)

1 2 Procedure

1. Set the mode switch to STOP.
2. Turn OFF the power to the GM1 controller.
3. Confirm that the project data "AUTOEXEC.GM1" and the setting file "AUTOEXEC.INI" generated by the backup process are contained in the "AUTO" folder in the SD card.
4. To edit targets to be restored, configure settings by editing the "AUTOEXEC.INI" configuration file.
5. Open the cover and insert the SD card into the slot.
6. With the cover left open, turn ON the power to the GM1 controller.
7. When the restore process starts, the STOP lamp will start blinking.
8. When the process is completed normally, the STOP lamp is lit and the ERROR is unlit.
   If an error occurs, the STOP lamp goes off and the ERROR lamp blinks.
9. Close the cover, set the mode switch to RUN, and turn the controller OFF and then ON, so that the controller switches to the restored project.

i Info.

- Application_File in the “AUTOEXEC.INI” restoration configuration file cannot be excluded from the target of restoration. If it is excluded from the target of restoration, an error occurs.

15.7.4 Backup and Restoration (Function Blocks)

You can back up or restore data from a program using function blocks. In combination with the FTP server function, a maintenance function can be implemented by remotely upgrading versions and regularly backing up data. The function block types are as shown below.

Item	Description
SYS_PRJBackup	This function block is used to collectively save every type of settings data in the GM1 controller to an SD memory card.
SYS_PRJRestore	This function block is used to transfer a backup file in an SD memory card to the GM1 Controller.
SYS_GetPRJRestoreResult	This function block is used to check results of execution of SYS_PRJRestore.

Info.

- For details on the function blocks, refer to to the Instruction Edition.

(MEMO)

16 Overview of PANATERM Lite for GM

16.1 System Requirements

16.1.1 Operating Environment of PANATERM Lite for GM

Programming software

Product name	Applicable language
PANATERM Lite for GM	Japanese / English / Chinese

(Note 1) When GM Programmer is installed, MINAS setup support software"PANATERM Lite for GM"is installed at the same time.

Software operating environment

Item	Description
OS	Microsoft(R) Windows(R) 10: 32bit / 64bit
PC	PC with the following software installed: ● Microsoft.NET Framework 4.6.1 or later ● Microsoft Visual C++ 2010 SP1 Redistributable Package (x86) ● Microsoft Visual C++ 2010 SP1 Redistributable Package (x64) ● Microsoft Visual C++ 2013 Redistributable Package (x86) ● Microsoft Visual C++ 2013 Redistributable Package (x64) ● Microsoft Visual C++ 2015 Update 3 Redistributable Package (x86) ● Microsoft Visual C++ 2015 Update 3 Redistributable Package (x64)
HDD	At least 4 GB of free space
Memory	At least 8 GB
Communication port	LAN port (for Ethernet connection) USB 2.0 port (for USB connection)

16.2 Installation and Uninstallation

16.2.1 Installing PANATERM Lite for GM

When GM Programmer is installed, PANATERM Lite for GM is also installed at the same time.

16.2.2 Uninstalling PANATERM Lite for GM

When GM Programmer is uninstalled, PANATERM Lite for GM is also uninstalled at the same time.

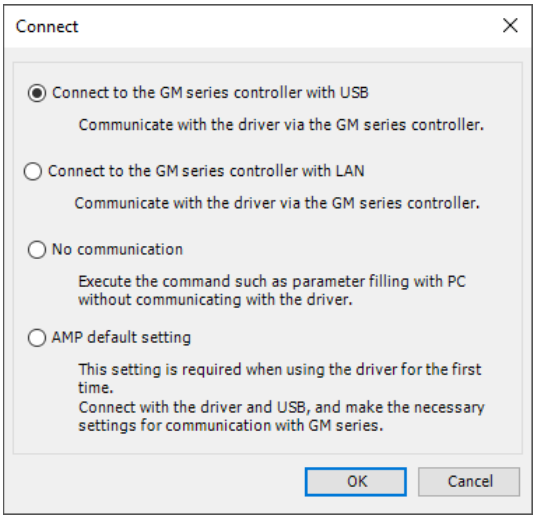
16.3 Basic Operations

This section explains how to start and exit PANATERM Lite for GM.

16.3.1 How to Start

1 2 Procedure

1. Click the [Start] button in the Windows task bar and select **Panasonic Corporation**>**PANATERM Lite for GM**.
2. The "Connect" dialog box will be displayed.
   Select a communication setting option and click [OK].



Connect

● Connect to the GM series controller with USB

Communicate with the driver via the GM series controller.

○ Connect to the GM series controller with LAN

Communicate with the driver via the GM series controller.

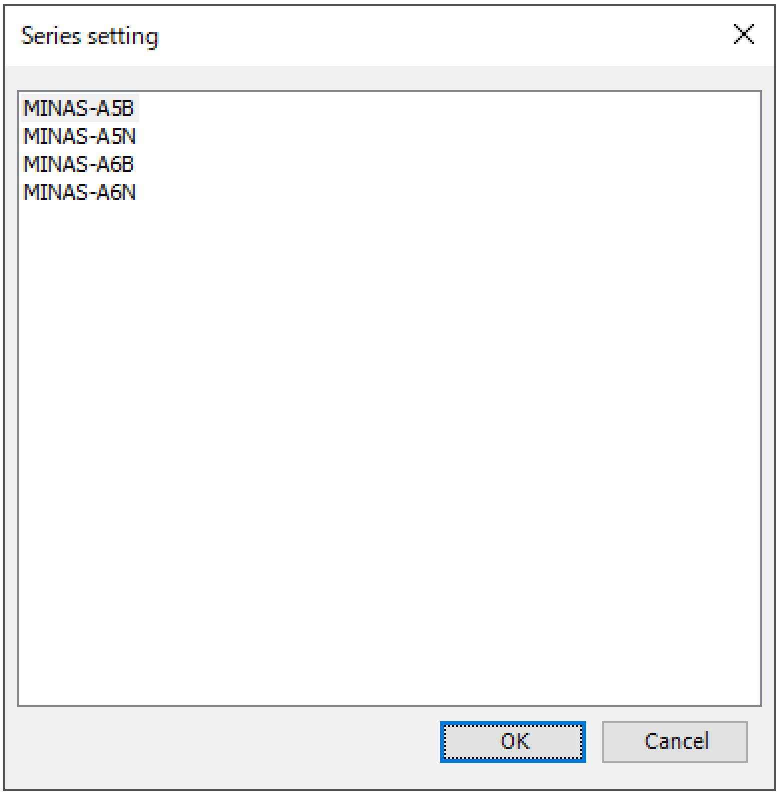
○ No communication

Execute the command such as parameter filling with PC
without communicating with the driver.

○ AMP default setting

This setting is required when using the driver for the first time.
Connect with the driver and USB, and make the necessary settings for communication with GM series.

3. The "Series Setting" dialog box will be displayed.

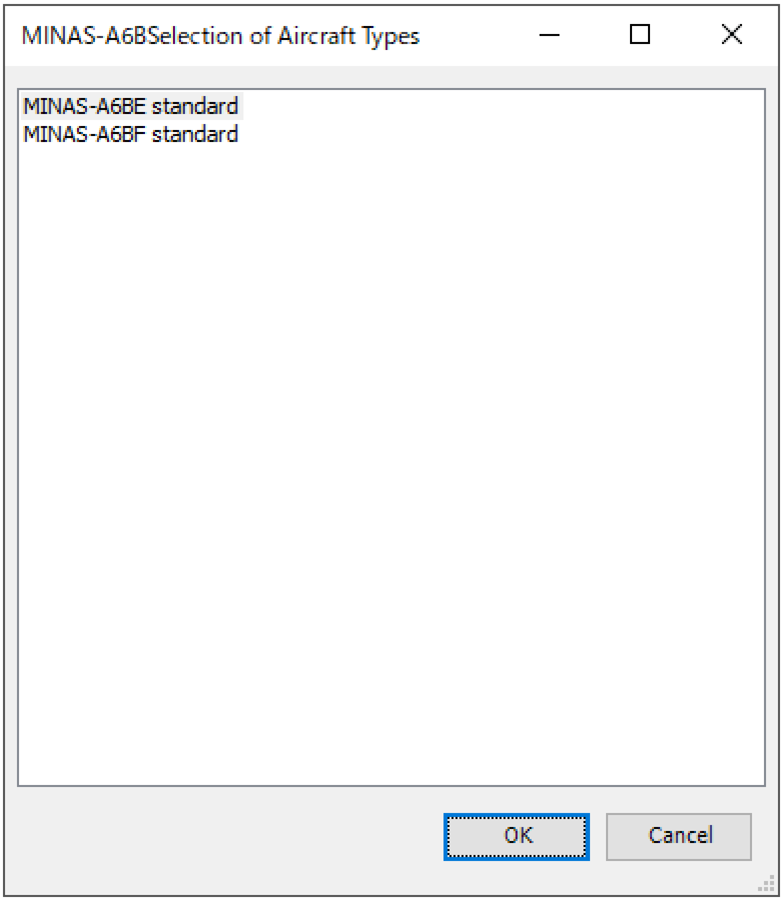


Series setting

MINAS-A5B
MINAS-A5N
MINAS-A6B
MINAS-A6N

Note

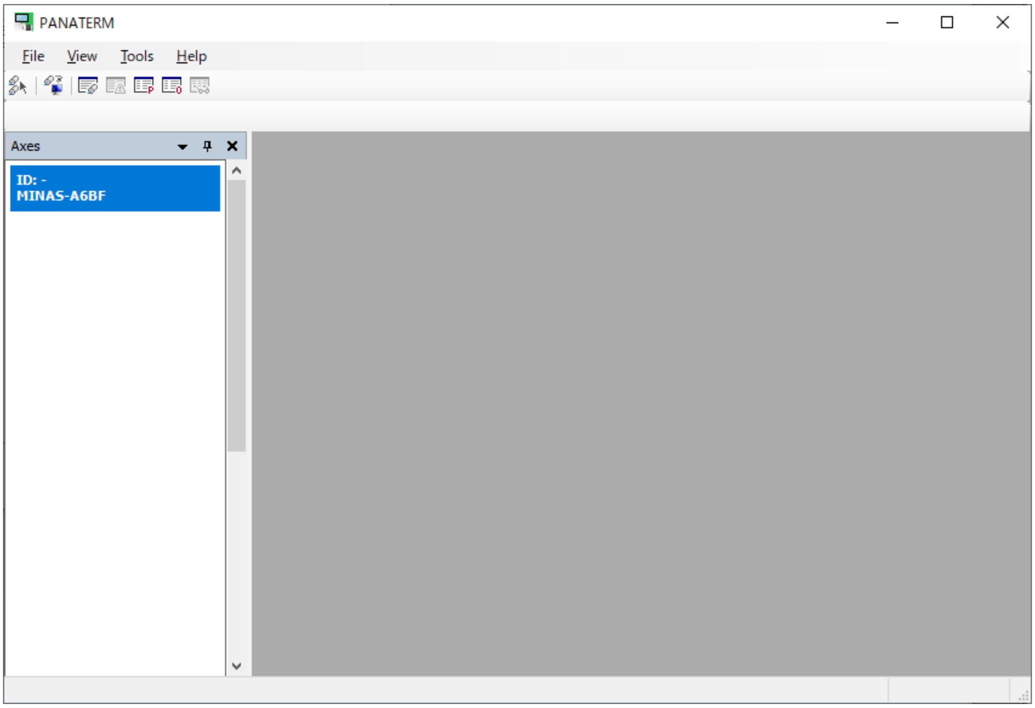
When the "Selection of Aircraft Types" dialog box is displayed, select a model and click the "OK" button.



MINAS-A6BSelection of Aircraft Types

MINAS-A6BE standard
MINAS-A6BF standard

4. PANATERM Lite for GM will be started.



PANATERM
File View Tools Help
Axes
ID: -
MINAS-A6BF

16.3.2 How to Exit

!

- Note that all information will be lost if you close the program without saving settings, collected data, or other information.

1 2 Procedure

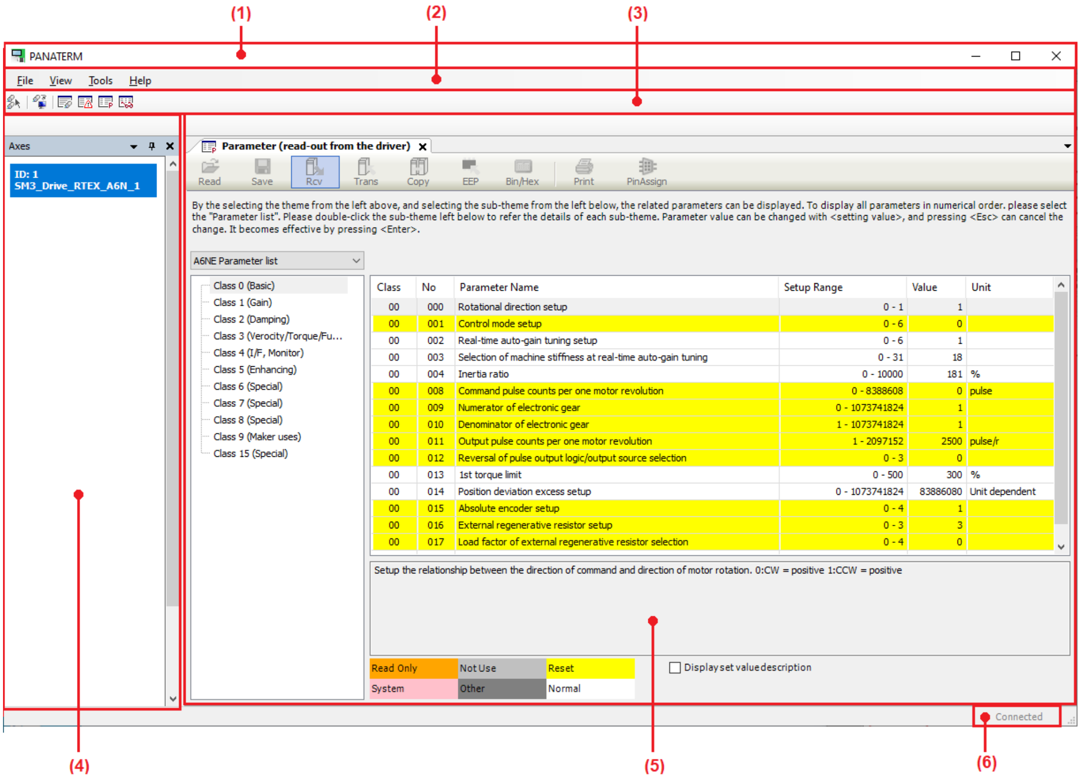
1. From the menu bar, select **File**>**Exit**.
   PANATERM Lite for GM will be closed.

i Info.

- You can also close PANATERM Lite for GM by clicking the [×] button on the title bar.

16.4 Component Names

This section explains the components and displays of PANATERM Lite for GM.



By the selecting the theme from the left above, and selecting the sub-theme from the left below, the related parameters can be displayed. To display all parameters in numerical order. please select the "Parameter list". Please double-click the sub-theme left below to refer the details of each sub-theme. Parameter value can be changed with <setting value>, and pressing <Esc> can cancel the change. It becomes effective by pressing <Enter>.

Class	No	Parameter Name	Setup Range	Value	Unit
00	000	Rotational direction setup	0 - 1	1	
00	001	Control mode setup	0 - 6	0	
00	002	Real-time auto-gain tuning setup	0 - 6	1	
00	003	Selection of machine stiffness at real-time auto-gain tuning	0 - 31	18	
00	004	Inertia ratio	0 - 10000	181	%
00	008	Command pulse counts per one motor revolution	0 - 8388608	0	pulse
00	009	Numerator of electronic gear	0 - 1073741824	1	
00	010	Denominator of electronic gear	1 - 1073741824	1	
00	011	Output pulse counts per one motor revolution	1 - 2097152	2500	pulse/r
00	012	Reversal of pulse output logic/output source selection	0 - 3	0	
00	013	1st torque limit	0 - 500	300	%
00	014	Position deviation excess setup	0 - 1073741824	83886080	Unit dependent
00	015	Absolute encoder setup	0 - 4	1	
00	016	External regenerative resistor setup	0 - 3	3	
00	017	Load factor of external regenerative resistor selection	0 - 4	0	

Setup the relationship between the direction of command and direction of motor rotation. 0:CW = positive 1:CCW = positive

No.	Name	Description
(1)	Title bar	The title bar displays the project file name, [minimize] button, [maximize] button, and [close] button.
(2)	Menu bar	The menu bar displays the menu commands for each purpose in list format.
(3)	Toolbar	The toolbar displays each command as an icon.
(4)	Navigator pane	This pane displays a list of axes.
(5)	Main pane	This pane displays the Parameter window, Monitor window, Alarm window, and other windows. The window can be switched by selecting a desired tab.
(6)	Status field	This field displays the status of connection to the GM1 controller.

16.4.1 Menu Bar

The menu bar displays the following menus:

File

Item	Function
Settings	Used to set up an amplifier with the amplifier connected or with a model selected. Select Model: Select an amplifier to be connected. Connect Amplifier: Select either the connection via PC or the direct connection for connecting the GM1 controller and the amplifier order to set up an amplifier.
Exit	Closes PANATERM Lite for GM

View

Item	Function
Axes	Displays a list of axes
Parameter	Displays the Parameter window
Monitor	Displays the Monitor window
Alarm	Displays the Alarm window

Tools

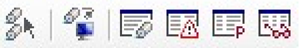
Item	Function
English	Switches the display language of GM Programmer to English
日本語	Switches the display language of GM Programmer to Japanese
中文(簡体)旧版	Switches the display language of GM Programmer to Chinese

Help

Item	Function
About	Displays version information
PANATERM Lite for GM Help	Displays the manual

16.4.2 Toolbar

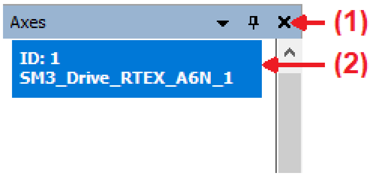
The toolbar displays the following icons:



Name	Icon	Function
Select the drive series		Sets up an amplifier with the amplifier unconnected
Opens the connrct dialog		Select either the connection via PC or the direct connection for connecting the GM1 controller and the amplifier order to set up an amplifier.
Opens the Axis view		Displays a list of axes
Opens the Alarm view		Displays the Alarm view
Opens the Parameter view		Displays the Parameter view
Opens the Monitor view		Displays the Monitor view

16.4.3 Navigation Pane

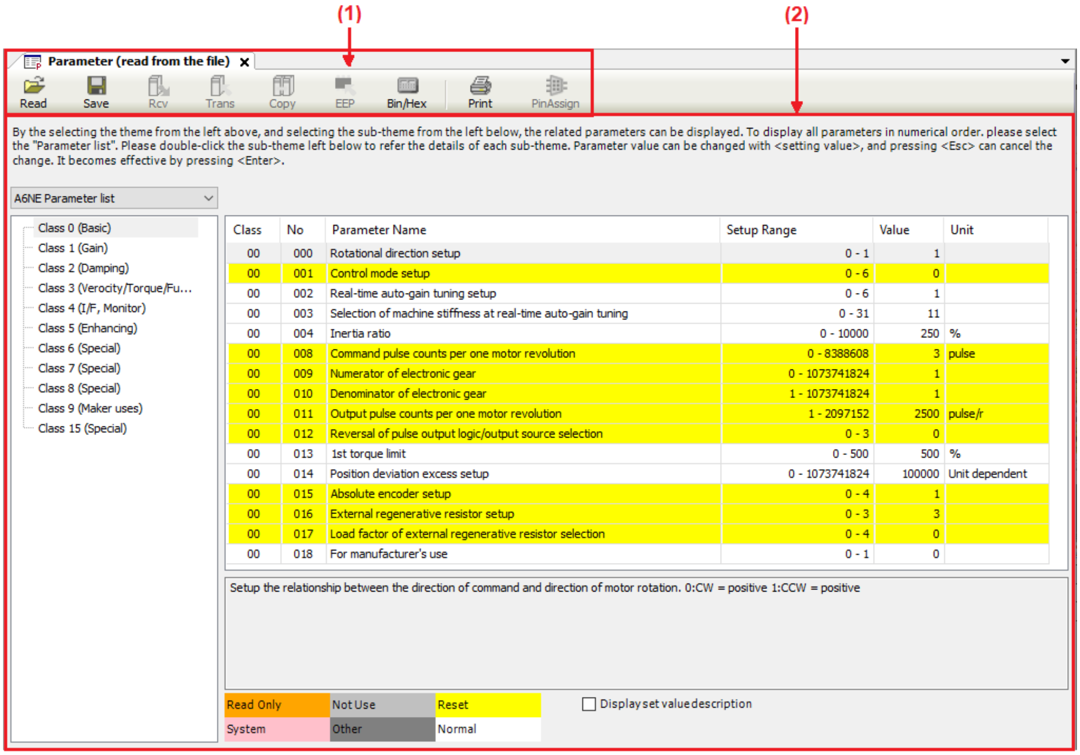
The navigation pane displays the following tree:



No.	Name	Icon	Function
(1)	Auto Hide		Always shows the navigation pane.
			Minimizes and hides the navigation pane.
	Close	✕	Closes the navigation pane.
(2)	Axes		Displays a list of axes downloaded to the GM1 controller

16.4.4 Main Pane

The main pane displays the following sub-panes:



By the selecting the theme from the left above, and selecting the sub-theme from the left below, the related parameters can be displayed. To display all parameters in numerical order. please select the "Parameter list". Please double-click the sub-theme left below to refer the details of each sub-theme. Parameter value can be changed with <setting value>, and pressing <Esc> can cancel the change. It becomes effective by pressing <Enter>.

A6NE Parameter list

- Class 0 (Basic)
- Class 1 (Gain)
- Class 2 (Damping)
- Class 3 (Verocity/Torque/Fu...
- Class 4 (I/F, Monitor)
- Class 5 (Enhancing)
- Class 6 (Special)
- Class 7 (Special)
- Class 8 (Special)
- Class 9 (Maker uses)
- Class 15 (Special)

Class	No	Parameter Name	Setup Range	Value	Unit
00	000	Rotational direction setup	0 - 1	1	
00	001	Control mode setup	0 - 6	0	
00	002	Real-time auto-gain tuning setup	0 - 6	1	
00	003	Selection of machine stiffness at real-time auto-gain tuning	0 - 31	11	
00	004	Inertia ratio	0 - 10000	250	%
00	008	Command pulse counts per one motor revolution	0 - 8388608	3	pulse
00	009	Numerator of electronic gear	0 - 1073741824	1	
00	010	Denominator of electronic gear	1 - 1073741824	1	
00	011	Output pulse counts per one motor revolution	1 - 2097152	2500	pulse/r
00	012	Reversal of pulse output logic/output source selection	0 - 3	0	
00	013	1st torque limit	0 - 500	500	%
00	014	Position deviation excess setup	0 - 1073741824	100000	Unit dependent
00	015	Absolute encoder setup	0 - 4	1	
00	016	External regenerative resistor setup	0 - 3	3	
00	017	Load factor of external regenerative resistor selection	0 - 4	0	
00	018	For manufacturer's use	0 - 1	0	

Setup the relationship between the direction of command and direction of motor rotation. 0:CW = positive 1:CCW = positive

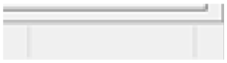
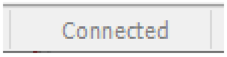
Read Only	Not Use	Reset
System	Other	Normal

□ Display set value description

No.	Name	
(1)	Toolbar	In each window, the toolbar displays commands as icons.
(2)	Main view	The main view displays the Parameter window, Alarm window, Monitor window, and other windows.

16.4.5 Status Field

The status field displays the current communication status.

Display	Description
	Indicates that there is no communication with the GM1 controller.
 Connected	Indicates that there are communications with the GM1 controller.

16.5 Window Operations

This section explains common window operations for PANATERM Lite for GM.

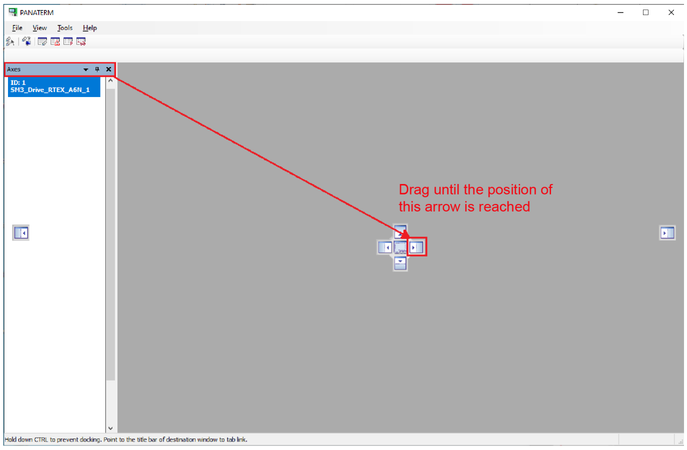
16.5.1 Moving the Pane Location

You can freely change the layout of each pane of PANATERM Lite for GM.

Example: Moving the navigator pane from the left edge to the right edge of the window

1 2 Procedure

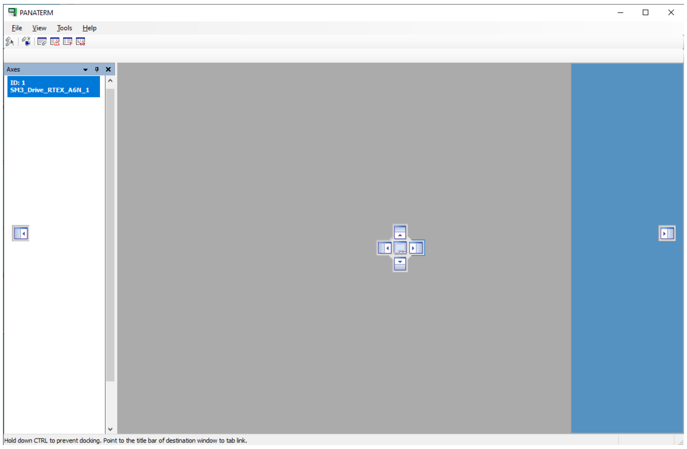
1. Click the title bar of the navigator pane and then drag it to the main pane.
   The navigator pane will stay in a floating state and arrows indicating movable directions will be displayed.



Drag until the position of this arrow is reached

Hold down CTRL to prevent docking. Point to the title bar of destination window to tab link.

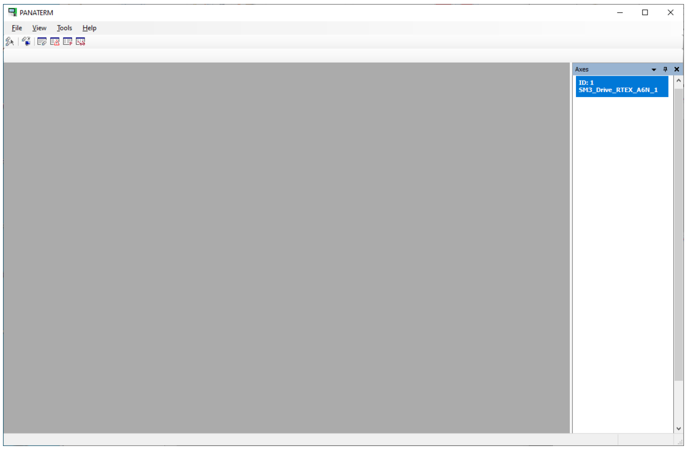
2. Drag the navigator pane in the direction in which you want to move it.
   The relocation destination will be displayed in light blue.



Hold down CTRL to prevent docking. Point to the title bar of destination window to tab link.

3. Release the left mouse button.

The navigator pane will be docked into the existing pane and the relocation will be completed.

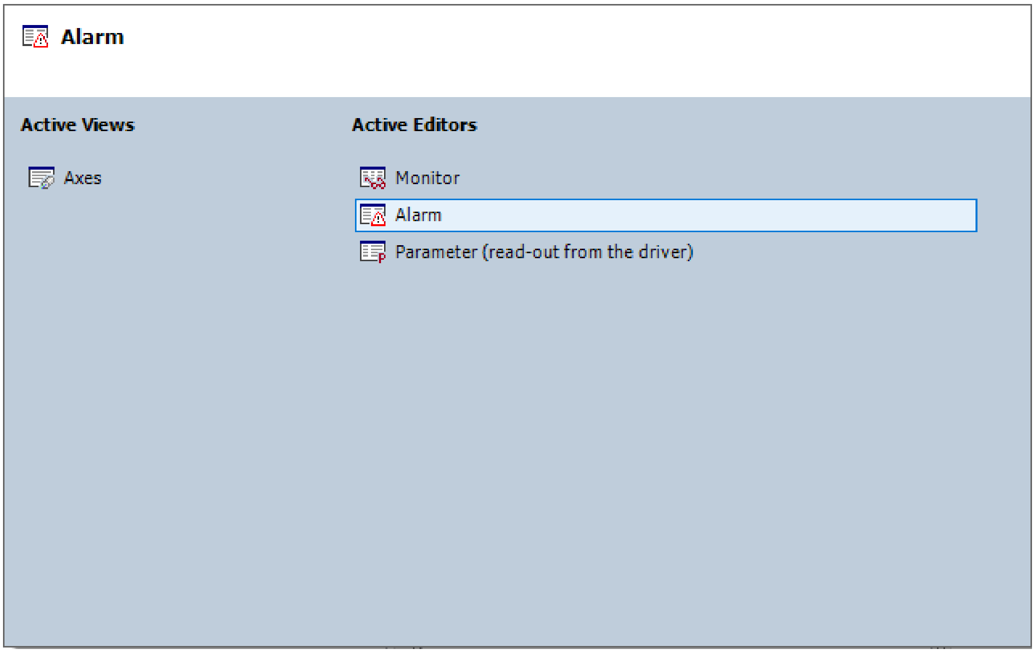


16.5.2 Switching the Tab of the Main Pane

You can switch the tab of the main pane.

1 2 Procedure

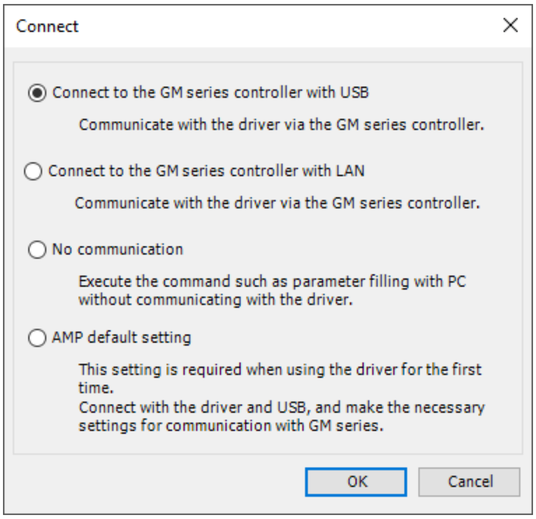
1. Press the <Ctrl> key + <Tab> key simultaneously.
   The window for switching the tab of the main pane will be displayed.



2. While holding down the <Ctrl> key, press the <Tab> key until the desired tab is selected.
3. Release the <Ctrl> key.
   The window corresponding to the selected tab will be displayed.

16.6 Selecting the Device to Connect

After you start PANATERM Lite for GM or when you select **File**>**Settings**>**Connect Amplifier**, a dialog box for selecting the device to be connected will be displayed.



● Connect to the GM series controller with USB

Communicate with the driver via the GM series controller.

○ Connect to the GM series controller with LAN

Communicate with the driver via the GM series controller.

○ No communication

Execute the command such as parameter filling with PC without communicating with the driver.

○ AMP default setting

This setting is required when using the driver for the first time.
Connect with the driver and USB, and make the necessary settings for communication with GM series.

The following options will be displayed.

- Connect to the GM series controller with USB
- Connect to the GM series controller with LAN
- No communication
- AMP default setting

The following section describes operations when each option is selected.

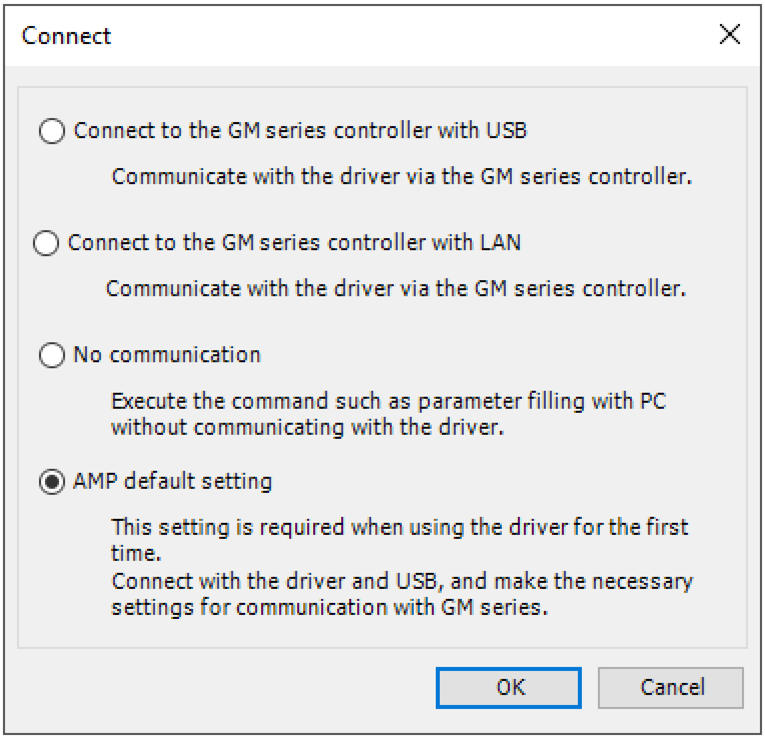
When you use the servo amplifier for the first time after it was purchased, you must establish a communication between the GM1 Controller and the servo amplifier. Connect the PC and the servo amplifier with a USB cable and execute "Configure amplifier communication settings".

Then, with the servo amplifier and GM1 Controller connected, set up the servo amplifier.

16.6.1 Configuring Servo Amplifier Communication Settings

1 2 Procedure

1. Start PANATERM Lite for GM.
   The "Connect" dialog box will be displayed.



Connect

○ Connect to the GM series controller with USB

Communicate with the driver via the GM series controller.

○ Connect to the GM series controller with LAN

Communicate with the driver via the GM series controller.

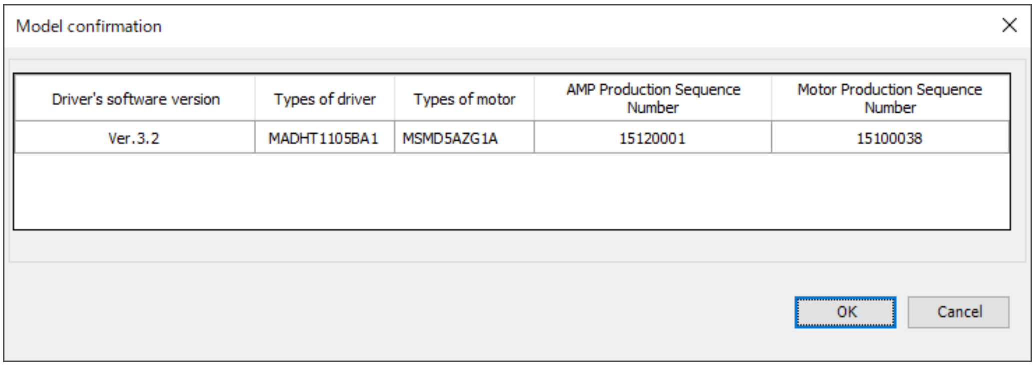
○ No communication

Execute the command such as parameter filling with PC without communicating with the driver.

● AMP default setting

This setting is required when using the driver for the first time.
Connect with the driver and USB, and make the necessary settings for communication with GM series.

2. Select "AMP default setting" and click [OK].
   The "Model confirmation" dialog box will be displayed.

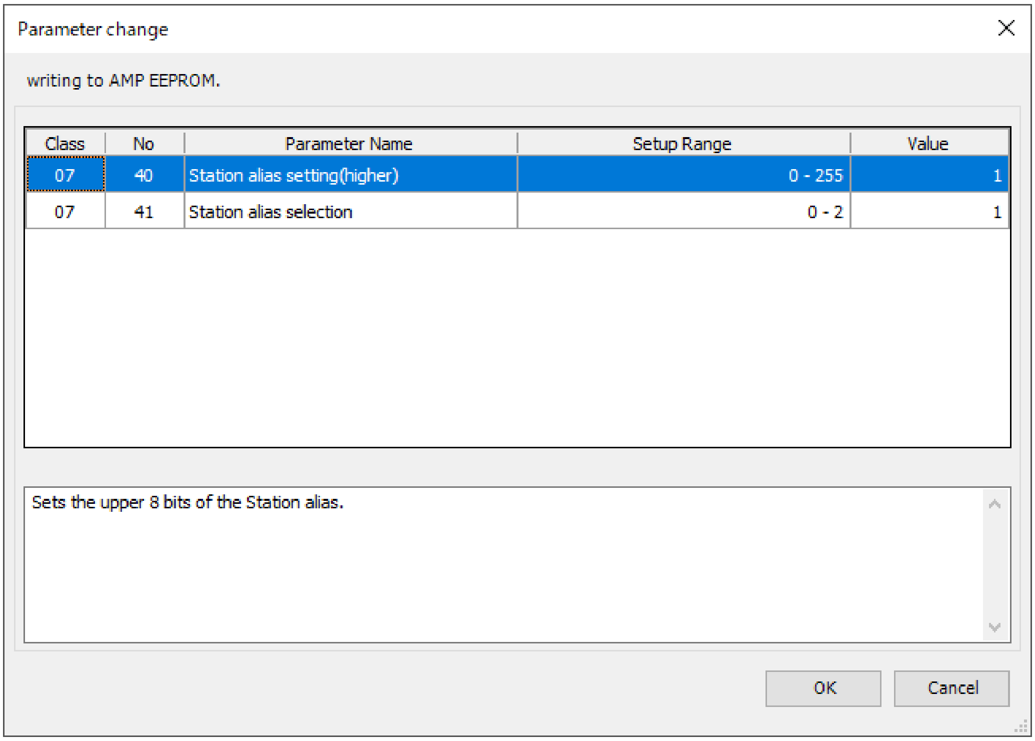


Model confirmation

Driver's software version	Types of driver	Types of motor	AMP Production Sequence Number	Motor Production Sequence Number
Ver.3.2	MADHT1105BA1	MSMD5AZG1A	15120001	15100038

3. Check the software version of the servo amplifier for which settings are to be changed and then click [OK].

The "Parameter change" dialog box will be displayed.



Parameter change

writing to AMP EEPROM.

Class	No	Parameter Name	Setup Range	Value
07	40	Station alias setting(higher)	0 - 255	1
07	41	Station alias selection	0 - 2	1

Sets the upper 8 bits of the Station alias.

When Pr7.41 is set to 0, set value of the rotary switch on the front panel of the servo amplifier and the set value of Pr7.40 are set as “Station alias”.

Station alias	
Higher 8 bits	**Lower 8 bits**
Set value of 3740h	Set value of the rotary switch

When Pr7.41 is set to 1, the value of the SII area (0004h) is regarded as Station alias.

The setting of the SII area (0004h) can be made using EtherCAT slave device object of the GM Programmer.

4. Click the [OK] button.
   The "Setting Complete" dialog box will be displayed.

5. Click the [OK] button.
   The main pane will be displayed. Start the servo amplifier.

16.6.2 Connection settings Servo Amplifier to the GM1 Controller

This section explains how to set up MINAS series servo amplifiers connected to the GM1 Controller.

For other devices, refer to their instruction manuals when setting up them.

The PC communicates with the servo amplifier connected to the GM1 Controller.

Connect the PC and GM1 Controller with an Ethernet cable or USB cable. With the GM1 Controller and servo amplifier connected with a Cat5e shielded cable, set up the servo amplifier.

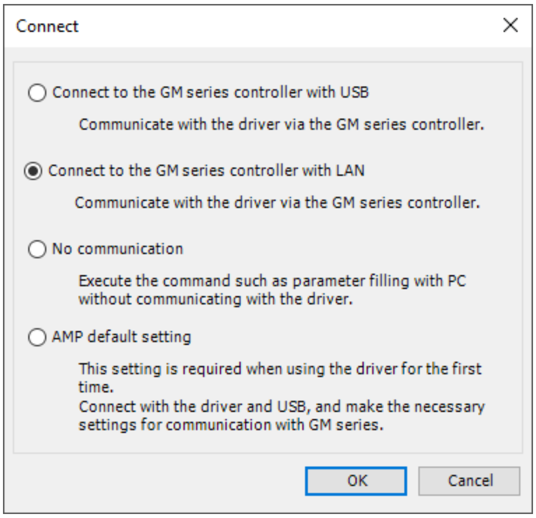
> **!**
> - Perform this setup work only after the connection between the GM1 Controller and the servo amplifier has been established.

If Connected with Ethernet Cables

If connected with Ethernet cables, use the following procedure.

1 2 Procedure

1. Start PANATERM Lite for GM.
The "Connect" dialog box will be displayed.



Connect

○ Connect to the GM series controller with USB

Communicate with the driver via the GM series controller.

● Connect to the GM series controller with LAN

Communicate with the driver via the GM series controller.

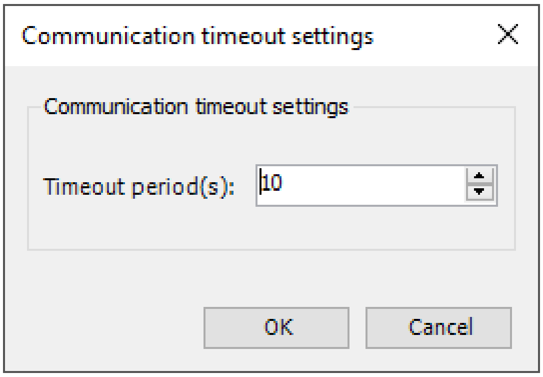
○ No communication

Execute the command such as parameter filling with PC without communicating with the driver.

○ AMP default setting

This setting is required when using the driver for the first time.
Connect with the driver and USB, and make the necessary settings for communication with GM series.

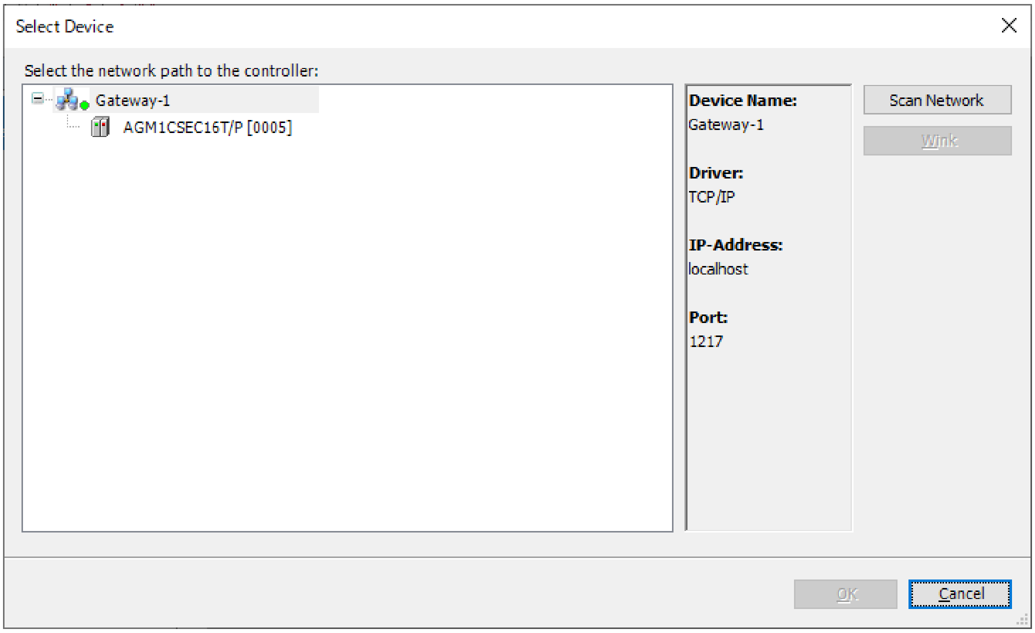
2. Select "Connect to the GM series controller with LAN" and click the [OK] button.
The "Communication timeout settings" dialog box will be displayed.



Communication timeout settings

Timeout period(s): 10

3. Change the timeout period and click the [OK] button.
   The "Select Device" dialog box will be displayed.



Select Device

Select the network path to the controller:

Gateway-1
AGM1CSEC16T/P [0005]

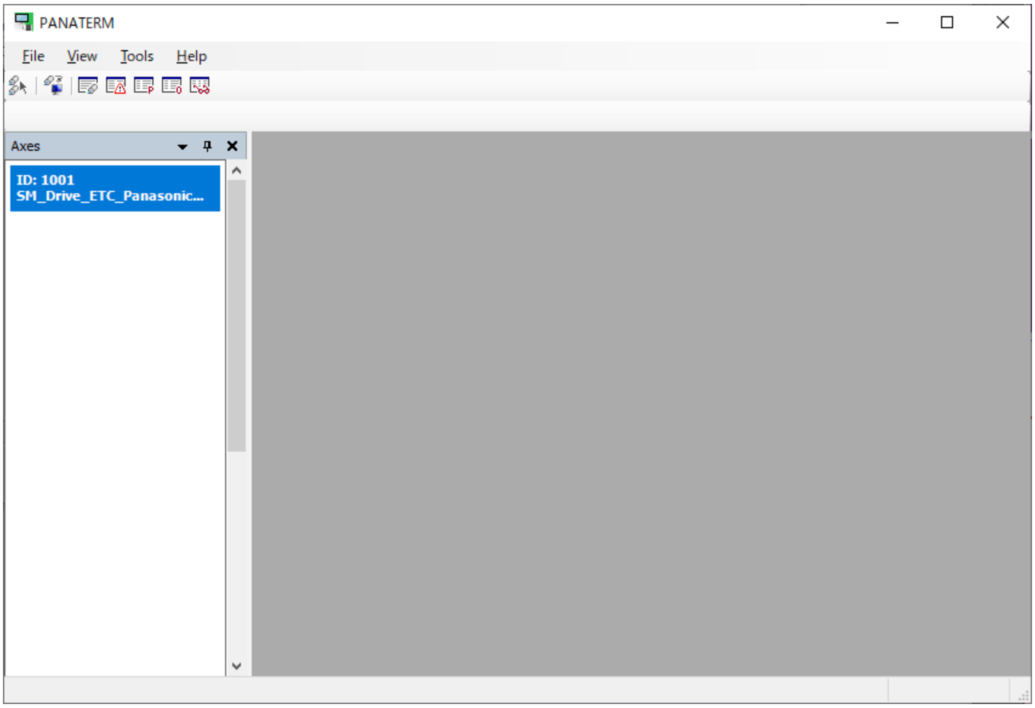
Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

4. Click the [Search Network] button, select the GM1 Controller, and click the [OK] button.
   The main pane will be displayed.

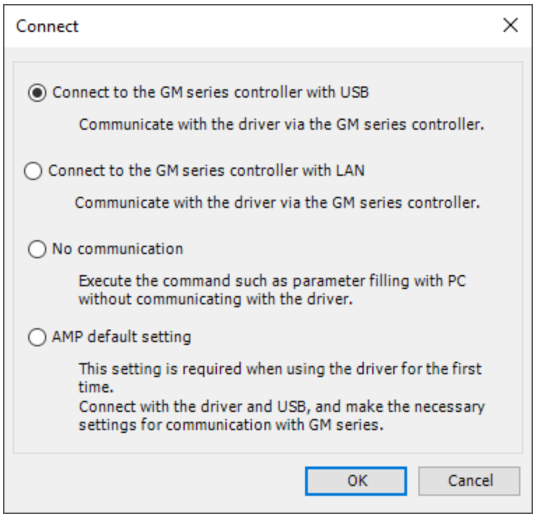

PANATERM
File View Tools Help
Axes
ID: 1001
SM_Drive_ETC_Panasonic...

If Connected with USB Cables

If connected with USB cables, use the following procedure.

1 2 Procedure

1. Start PANATERM Lite for GM.
   The "Connect" dialog box will be displayed.



Connect

● Connect to the GM series controller with USB

Communicate with the driver via the GM series controller.

○ Connect to the GM series controller with LAN

Communicate with the driver via the GM series controller.

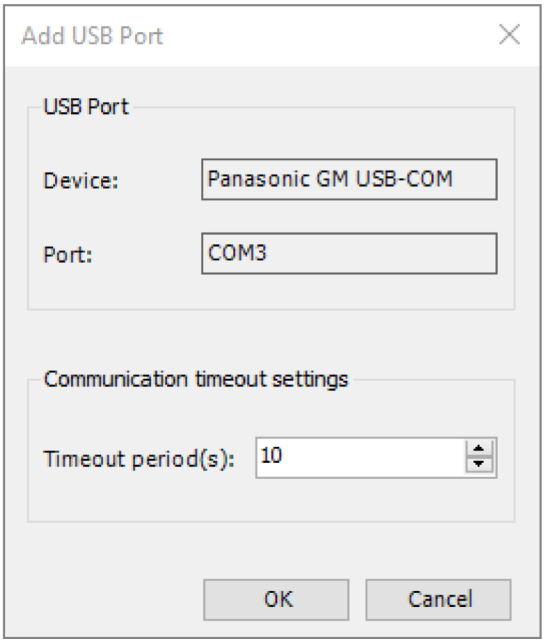
○ No communication

Execute the command such as parameter filling with PC without communicating with the driver.

○ AMP default setting

This setting is required when using the driver for the first time.
Connect with the driver and USB, and make the necessary settings for communication with GM series.

2. Select "Connect to the GM series controller with USB" and click the [OK] button.
   The "Add USB Port" dialog box will be displayed.



Add USB Port

USB Port

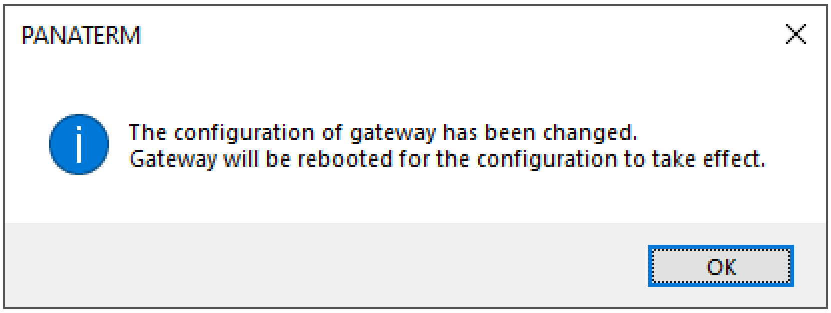
Device: Panasonic GM USB-COM

Port: COM3

Communication timeout settings

Timeout period(s): 10

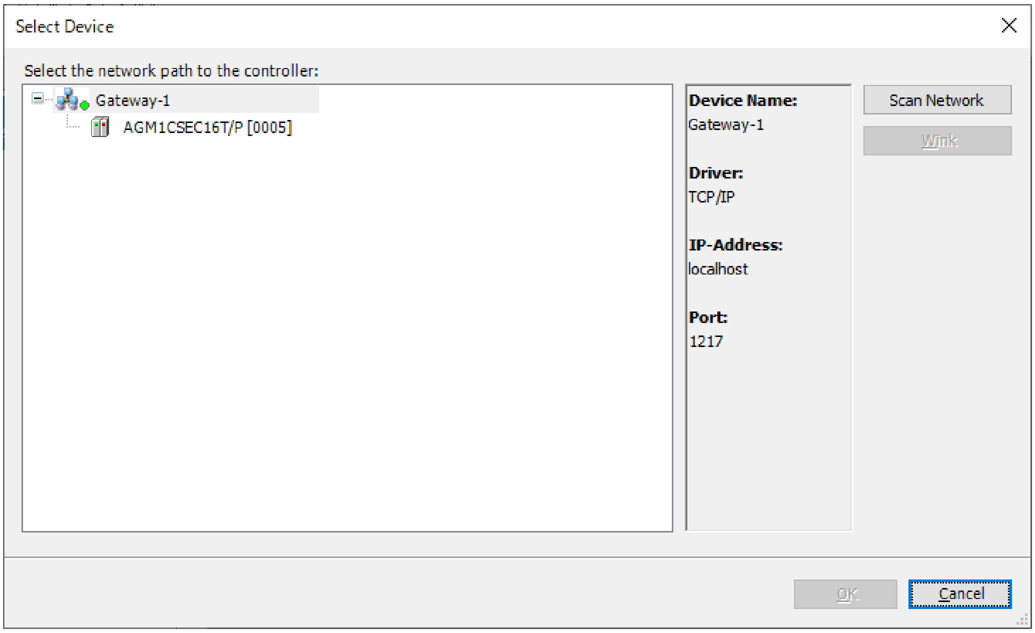
3. Change the timeout period and click the [OK] button.
A message window will be displayed, asking whether to add a USB port and restart the Gateway.



PANATERM

The configuration of gateway has been changed.
Gateway will be rebooted for the configuration to take effect.

4. Click the [OK] button.
The "Select Device" dialog box will be displayed.



Select Device

Select the network path to the controller:

Gateway-1
AGM1CSEC16T/P [0005]

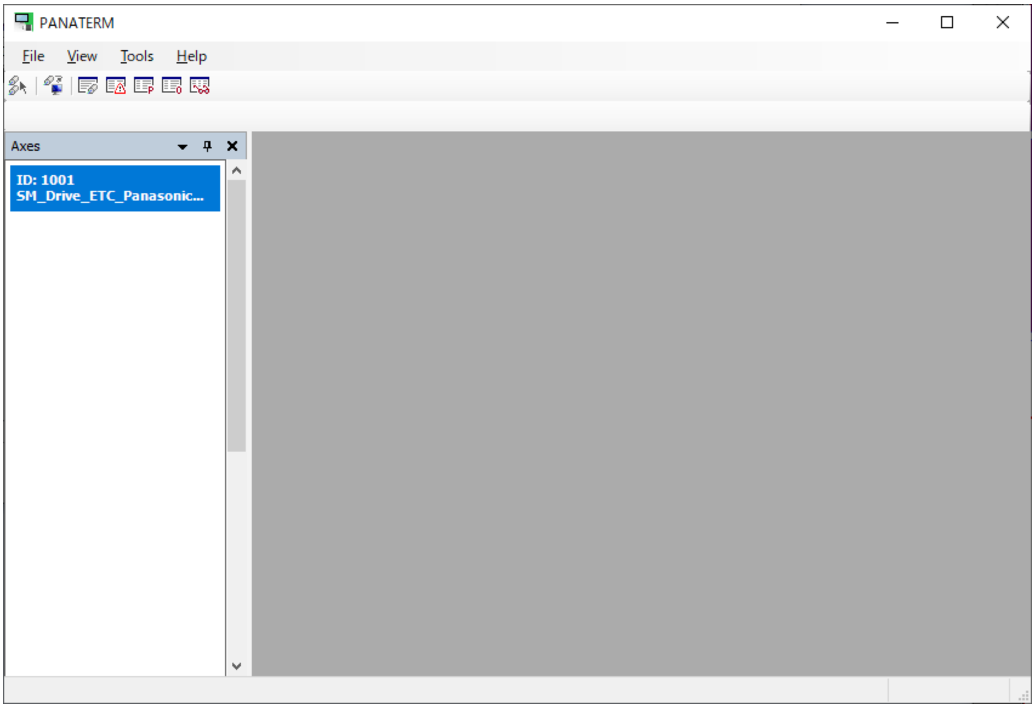
Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

5. Click the [Search Network] button, select the GM1 Controller, and click the [OK] button.
The main pane will be displayed.

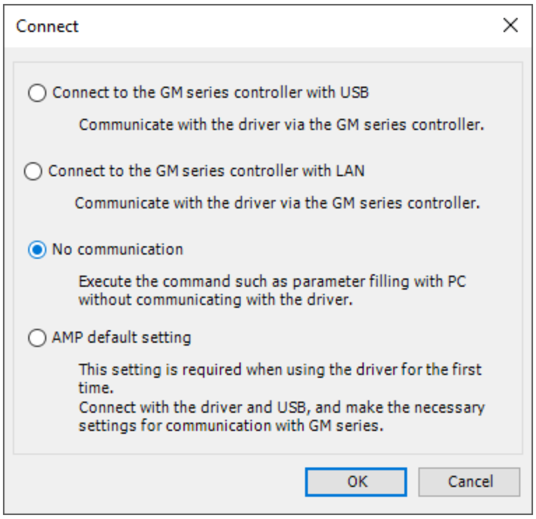

PANATERM
File
View
Tools
Help
Axes
ID: 1001
SM_Drive_ETC_Panasonic...

16.6.3 Editing Settings without Connecting to the GM1 Controller

You can freely edit parameters and other data saved in files without connecting to the GM1 controller. The edited contents are not written to the servo amplifier.

1 2 Procedure

1. Start PANATERM Lite for GM.
   The "Connect" dialog box will be displayed.



Connect

○ Connect to the GM series controller with USB

Communicate with the driver via the GM series controller.

○ Connect to the GM series controller with LAN

Communicate with the driver via the GM series controller.

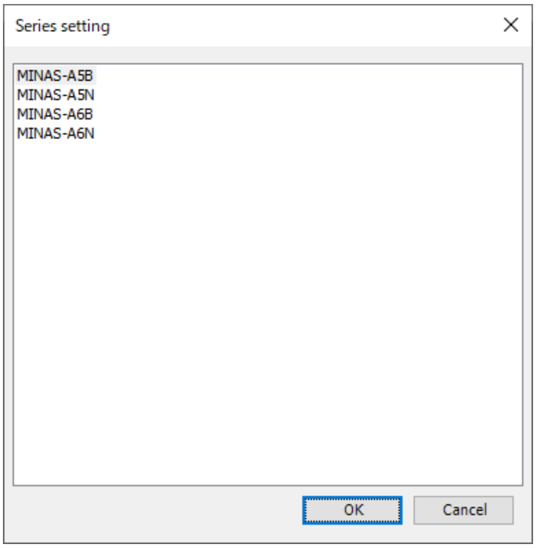
● No communication

Execute the command such as parameter filling with PC without communicating with the driver.

○ AMP default setting

This setting is required when using the driver for the first time.
Connect with the driver and USB, and make the necessary settings for communication with GM series.

2. Select "No communication" and click the [OK] button.
   The "Select Series " dialog box will be displayed.



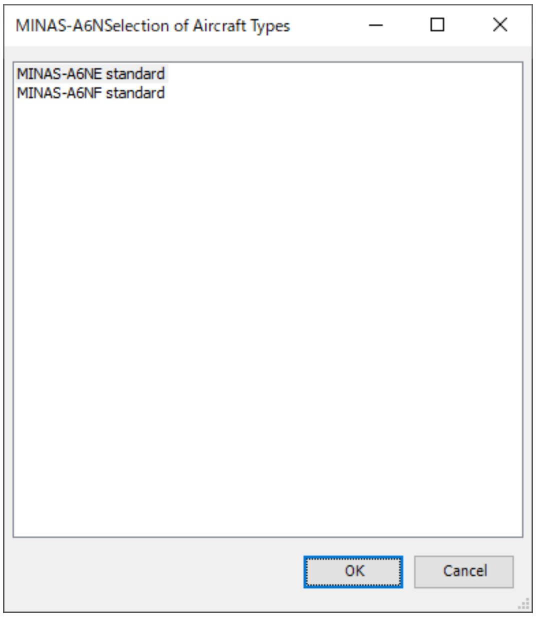
Series setting

MINAS-A5B
MINAS-A5N
MINAS-A6B
MINAS-A6N

3. Select a servo amplifier to be connected and click the [OK] button.
   The main pane will be displayed.

Note

When the "Select Model" dialog box is displayed, select a model and click the "OK" button.



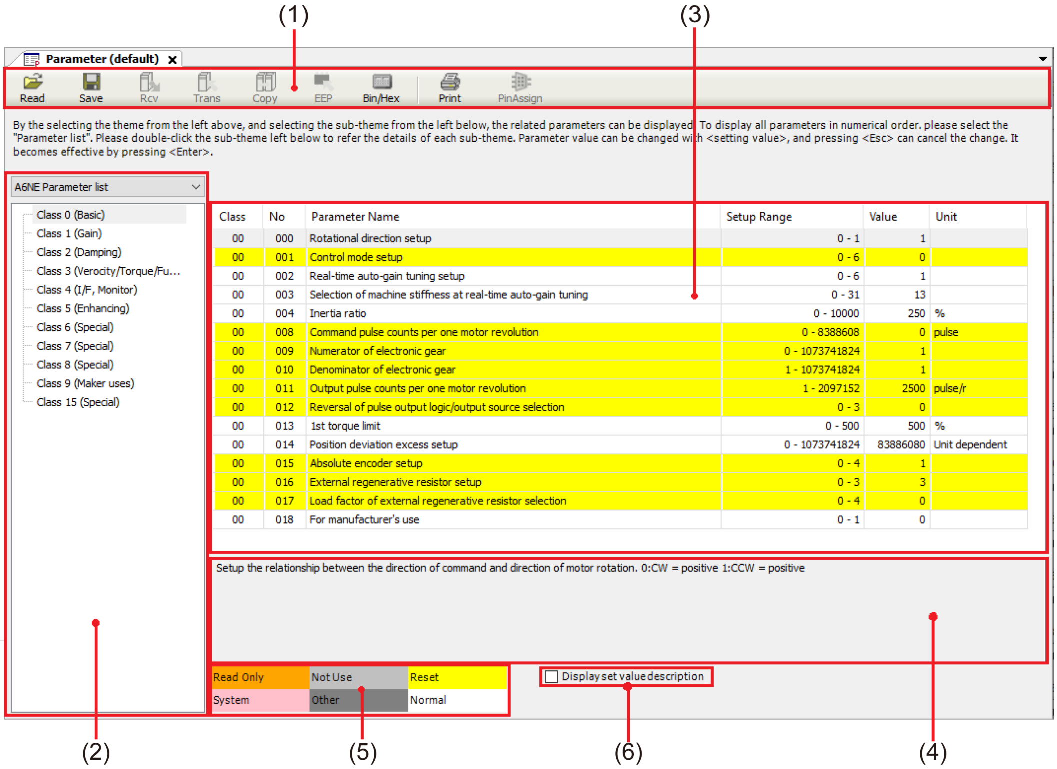
MINAS-A6NSelection of Aircraft Types

MINAS-A6NE standard
MINAS-A6NF standard

16.7 Parameter Window

The Parameter window allows the user to check and rewrite the values of servo amplifier parameters, save them to parameter files, and perform other parameter-related operations.

16.7.1 Configuration of Parameters Window

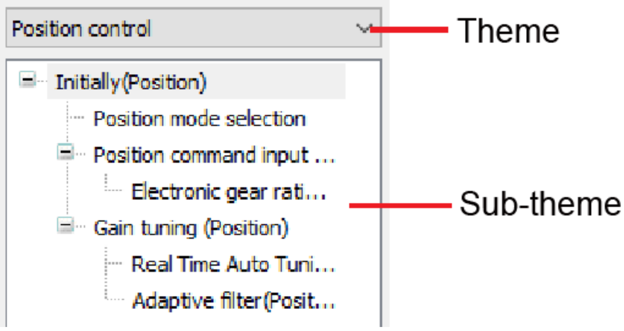


By the selecting the theme from the left above, and selecting the sub-theme from the left below, the related parameters can be displayed. To display all parameters in numerical order. please select the "Parameter list". Please double-click the sub-theme left below to refer the details of each sub-theme. Parameter value can be changed with <setting value>, and pressing <Esc> can cancel the change. It becomes effective by pressing <Enter>.

Class	No	Parameter Name	Setup Range	Value	Unit
00	000	Rotational direction setup	0 - 1	1	
00	001	Control mode setup	0 - 6	0	
00	002	Real-time auto-gain tuning setup	0 - 6	1	
00	003	Selection of machine stiffness at real-time auto-gain tuning	0 - 31	13	
00	004	Inertia ratio	0 - 10000	250	%
00	008	Command pulse counts per one motor revolution	0 - 8388608	0	pulse
00	009	Numerator of electronic gear	0 - 1073741824	1	
00	010	Denominator of electronic gear	1 - 1073741824	1	
00	011	Output pulse counts per one motor revolution	1 - 2097152	2500	pulse/r
00	012	Reversal of pulse output logic/output source selection	0 - 3	0	
00	013	1st torque limit	0 - 500	500	%
00	014	Position deviation excess setup	0 - 1073741824	83886080	Unit dependent
00	015	Absolute encoder setup	0 - 4	1	
00	016	External regenerative resistor setup	0 - 3	3	
00	017	Load factor of external regenerative resistor selection	0 - 4	0	
00	018	For manufacturer's use	0 - 1	0	

Setup the relationship between the direction of command and direction of motor rotation. 0:CW = positive 1:CCW = positive

<table>
<tr><th>No.</th><th>Name</th><th colspan="3">Function</th></tr>
<tr><td rowspan="6">(1)</td><td rowspan="6">Toolbar</td><td colspan="3">The toolbar consists of basic operation commands related to parameters, such as save and read.</td></tr>
<tr><th>Icon</th><th>Name</th><th>Function</th></tr>
<tr><td>Read</td><td>Read</td><td>Reads parameters from file".prm5"</td></tr>
<tr><td>Save</td><td>Save</td><td>Writes parameters to file".prm5"</td></tr>
<tr><td>Rcv</td><td>Rcv</td><td>Receives parameters from the servo amplifier.</td></tr>
<tr><td>Trans</td><td>Trans</td><td>Transmits parameters to the servo amplifier.</td></tr>
</table>

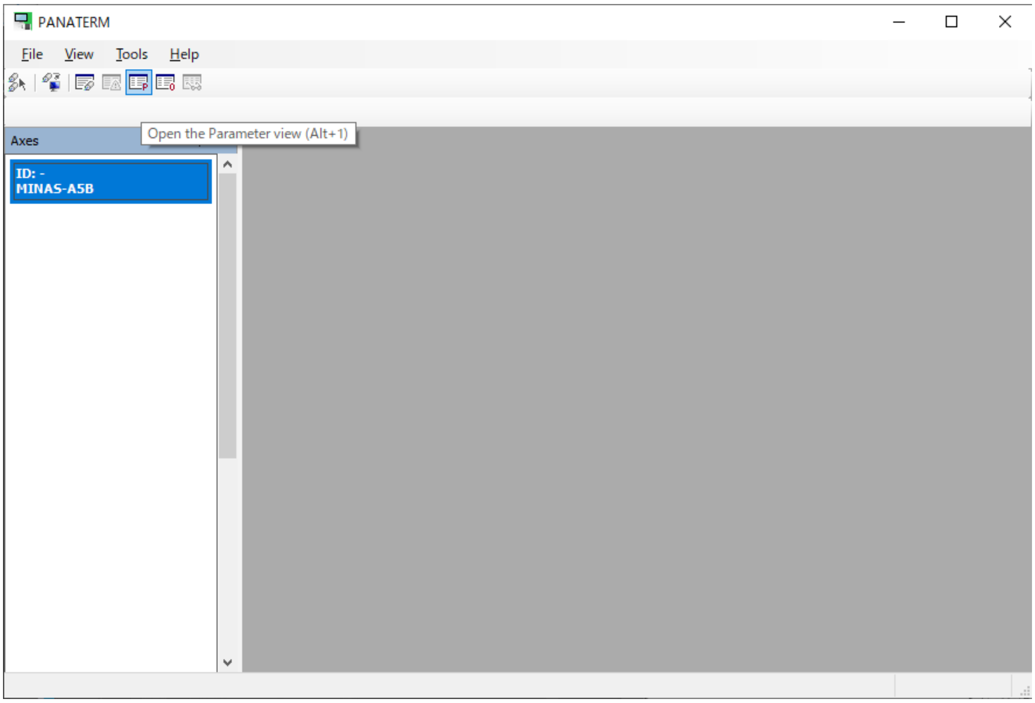
<table>
<tr><th>No.</th><th>Name</th><th>Function</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Icon</th><th>Name</th><th>Function</th></tr>
<tr><td>Copy</td><td>Copy</td><td>Copies the parameters of a servo amplifier to servo amplifiers for other axes.</td></tr>
<tr><td>EEP</td><td>EEP</td><td>Writes parameters to EEPROM of the servo amplifier.</td></tr>
<tr><td>Bin/Hex</td><td>Bin / Hex</td><td>Inputs the selected settings in binary or hexadecimal format.</td></tr>
<tr><td>Print</td><td>Print</td><td>Prints parameters.</td></tr>
<tr><td>PinAssign</td><td>Pin assignment setting</td><td>Sets I/O pin assignment.</td></tr>
</table>
</td></tr>
<tr><td>(2)</td><td>Theme selection pane</td><td>After a theme is selected, if a parameter category is selected from a sub-theme, related parameters will be displayed in the parameter setting area.

For details on each parameter, refer to the instruction manual and other technical references for the servo amplifier.</td></tr>
<tr><td>(3)</td><td>Parameter setting area</td><td>Allows the user to set or edit parameters.
<table>
<tr><th>Name</th><th>Function</th></tr>
<tr><td>Class</td><td>Displays parameter categories</td></tr>
<tr><td>No.</td><td>Displays parameter numbers</td></tr>
<tr><td>Parameter Name</td><td>Displays parameter names</td></tr>
<tr><td>Setup Range</td><td>Displays the maximum and minimum allowable values of parameter settings</td></tr>
<tr><td>Value</td><td>Displays parameter values. Values can be changed. For parameters provided with a ▼ button beside the set value, a desired value can be selected from the combo box. After selecting a value from the combo box, press the <Enter> key. For parameters without a ▼ button beside the set value, either directly enter a value using <numerical> keys or click "▲""▼" to edit the value by increasing or decreasing it. To set a value, press the <Enter> key. To return a value to its original value, press the <Esc> key.</td></tr>
<tr><td>Unit</td><td>Displays the unit of parameter settings.</td></tr>
</table>
</td></tr>
</table>

No.	Name	Function
(4)	Text display area	Displays a description related to the selected parameter.
(5)	Parameter attribute description area	Displays a description of parameter attributes. The background color of each parameter in the parameter setting area represents an attribute.
(6)	"Display-set value description" check box	Selecting the check box displays combo boxes and decimal points in the "Value" column of the parameter setting area. To display parameter set values in an easy-to-understand manner, select the check box.

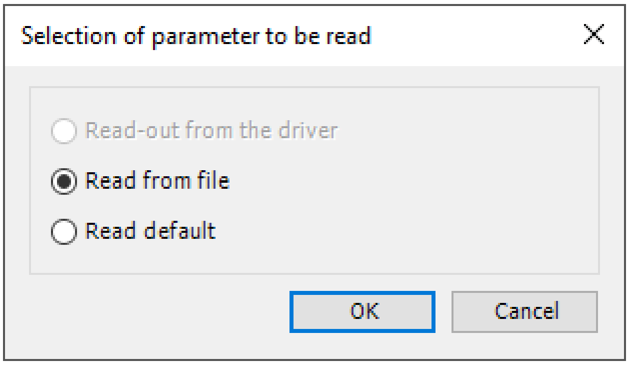
16.7.2 Setting Parameters

1 2 Procedure

1. From the menu bar on the main pane, select **View**>**Parameter**. Alternatively, on the toolbar, click the "Open the Parameter view" icon.



The "Selection of parameter to be read" dialog box will be displayed.



Selection of parameter to be read

○ Read-out from the driver

● Read from file

○ Read default

□Read-out from the driver

Communicates with the connected servo amplifier and reads the parameter settings from the servo amplifier. If this mode is selected, parameter values will be reflected in the servo amplifier as soon as they are changed.

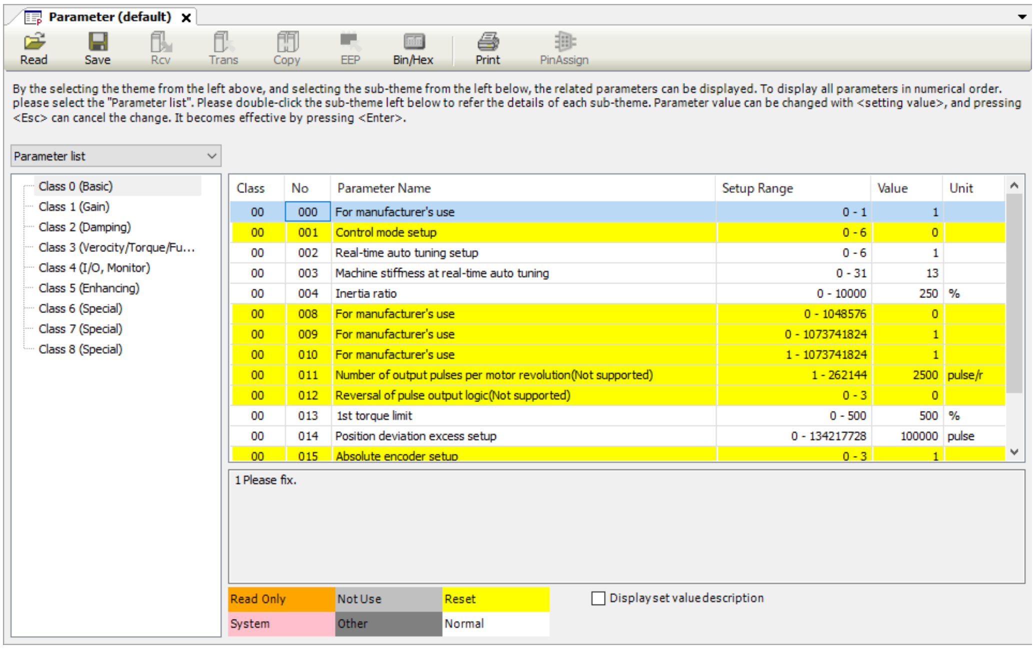
□Read from file

Reads the parameter file (".prm5") that was edited previously. If communication is performed with the servo amplifier, parameter values will be reflected in the servo amplifier as soon as they are changed.

□Read default

Reads the factory default settings of the servo amplifier that were saved during installation. If communication is performed with the servo amplifier, parameter values will be reflected in the servo amplifier as soon as they are changed.

2. Select one of the three options above and click the [OK] button.
   The Parameter window will be displayed.



By the selecting the theme from the left above, and selecting the sub-theme from the left below, the related parameters can be displayed. To display all parameters in numerical order. please select the "Parameter list". Please double-click the sub-theme left below to refer the details of each sub-theme. Parameter value can be changed with <setting value>, and pressing <Esc> can cancel the change. It becomes effective by pressing <Enter>.

Parameter list

- Class 0 (Basic)
- Class 1 (Gain)
- Class 2 (Damping)
- Class 3 (Verocity/Torque/Fu...
- Class 4 (I/O, Monitor)
- Class 5 (Enhancing)
- Class 6 (Special)
- Class 7 (Special)
- Class 8 (Special)

Class	No	Parameter Name	Setup Range	Value	Unit
00	000	For manufacturer's use	0 - 1	1	
00	001	Control mode setup	0 - 6	0	
00	002	Real-time auto tuning setup	0 - 6	1	
00	003	Machine stiffness at real-time auto tuning	0 - 31	13	
00	004	Inertia ratio	0 - 10000	250	%
00	008	For manufacturer's use	0 - 1048576	0	
00	009	For manufacturer's use	0 - 1073741824	1	
00	010	For manufacturer's use	1 - 1073741824	1	
00	011	Number of output pulses per motor revolution(Not supported)	1 - 262144	2500	pulse/r
00	012	Reversal of pulse output logic(Not supported)	0 - 3	0	
00	013	1st torque limit	0 - 500	500	%
00	014	Position deviation excess setup	0 - 134217728	100000	pulse
00	015	Absolute encoder setup	0 - 3	1	

1 Please fix.

Read Only	Not Use	Reset
System	Other	Normal

□ Display set value description

3. After changing the parameter settings, click the [EEP] button to write the parameter settings to the EEPROM of the servo amplifier.

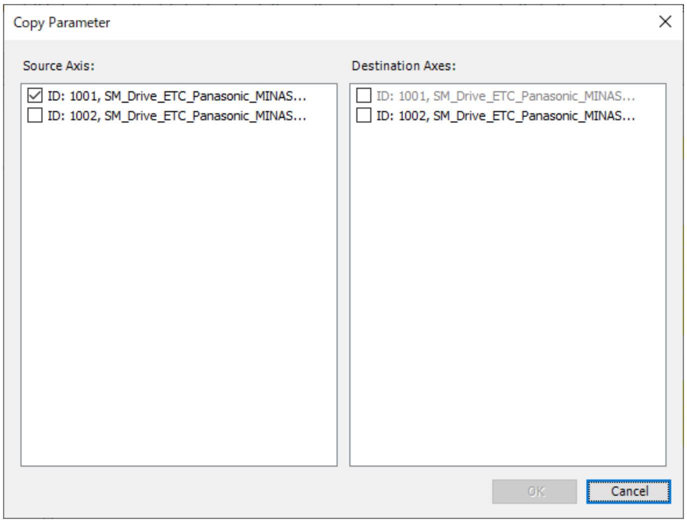
4. Click the [×] button on the Parameter window to close the Parameter window.

16.7.3 Copying Parameters

The parameters of a servo amplifier can be copied to servo amplifiers for other axes.
It is not possible to copy from A6B type to A5B type or from A5B type to A6B type.

Procedure

1. Click the "Copy" icon on the toolbar.
The Copy Parameter window will be displayed.



Copy Parameter

Source Axis:

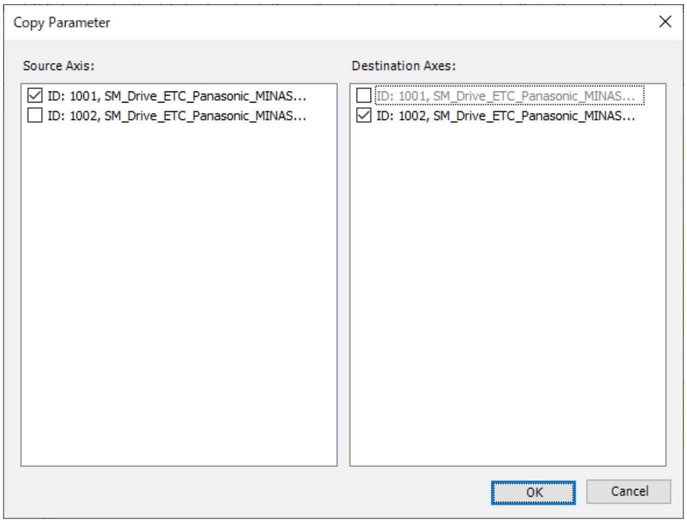
☑ ID: 1001, SM_Drive_ETC_Panasonic_MINAS...
☐ ID: 1002, SM_Drive_ETC_Panasonic_MINAS...

Destination Axes:

☐ ID: 1001, SM_Drive_ETC_Panasonic_MINAS...
☐ ID: 1002, SM_Drive_ETC_Panasonic_MINAS...

2. In the "Copy source" area, select an axis from which parameters are to be copied.

3. In the "Copy destination" area, select an axis to which parameters are to be copied.


Copy Parameter
Source Axis:
☑ ID: 1001, SM_Drive_ETC_Panasonic_MINAS...
□ ID: 1002, SM_Drive_ETC_Panasonic_MINAS...
Destination Axes:
□ ID: 1001, SM_Drive_ETC_Panasonic_MINAS...
☑ ID: 1002, SM_Drive_ETC_Panasonic_MINAS...
OK
Cancel

4. Click the [OK] button.

5. The Parameter window for the copy destination axis will be displayed.

6. Click the "Trans" icon on the toolbar.
   The parameters will be written to the servo amplifier.

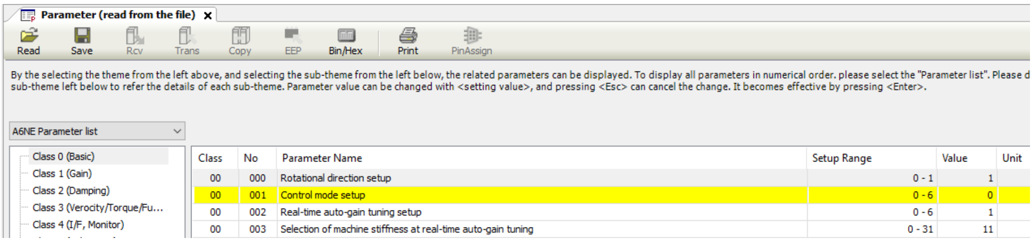
7. Click the [EEP] icon on the toolbar.
   The parameters will be written to the EEPROM of the servo amplifier.

16.7.4 Switching the Input Format of Parameter Values

Selected parameter values can be entered in binary or hexadecimal format.

Procedure

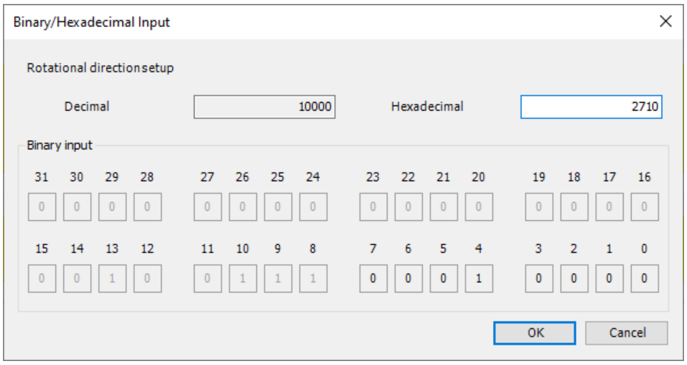
1. Click the "Bin / Hex" icon on the toolbar.
   The Binary / Hexadecimal Input window will be displayed.



By the selecting the theme from the left above, and selecting the sub-theme from the left below, the related parameters can be displayed. To display all parameters in numerical order. please select the "Parameter list". Please d
sub-theme left below to refer the details of each sub-theme. Parameter value can be changed with <setting value>, and pressing <Esc> can cancel the change. It becomes effective by pressing <Enter>.

Class	No	Parameter Name	Setup Range	Value	Unit
00	000	Rotational direction setup	0 - 1	1	
00	001	Control mode setup	0 - 6	0	
00	002	Real-time auto-gain tuning setup	0 - 6	1	
00	003	Selection of machine stiffness at real-time auto-gain tuning	0 - 31	11	

2. To enter parameter values in hexadecimal format, enter a value and then press the <Enter> key. To enter parameter values in binary format, click the button corresponding to each bit to switch between "0" and "1".


Binary/Hexadecimal Input

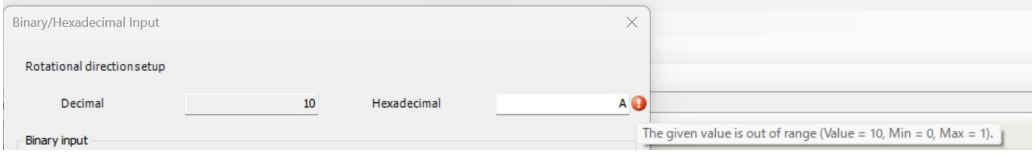
Rotational directionsetup

Decimal 10000 Hexadecimal 2710

Binary input

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0

Note: If the entered value is outside the setting range of the parameter, the allowable range will be displayed below the decimal display section.


Binary/Hexadecimal Input

Rotational directionsetup

Decimal 10 Hexadecimal A

Binary input

The given value is out of range (Value = 10, Min = 0, Max = 1).

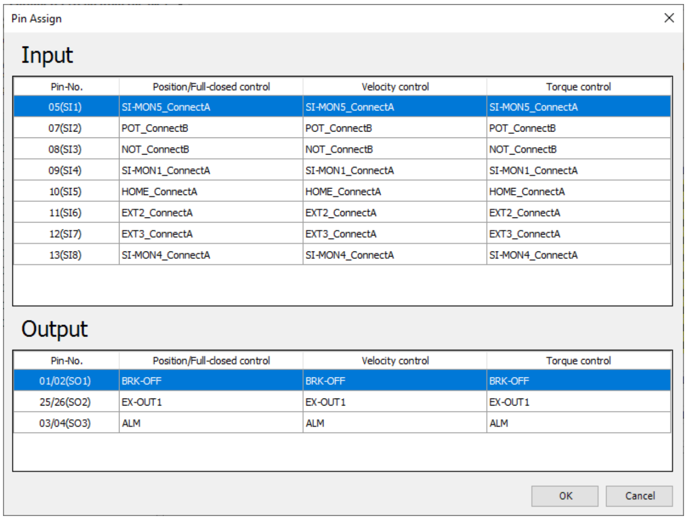
3. After the above input operation is complete, click the [OK]button.

16.7.5 Setting I/O Pin Assignment

I/O pin assignment can be set.
The assignment can be set for the input and output related parameter "Class 4 (I/F, monitor)".

Procedure

1. Click the "Pin assignment " icon on the toolbar.
The pin assignment setting input window will be displayed.


Pin Assign

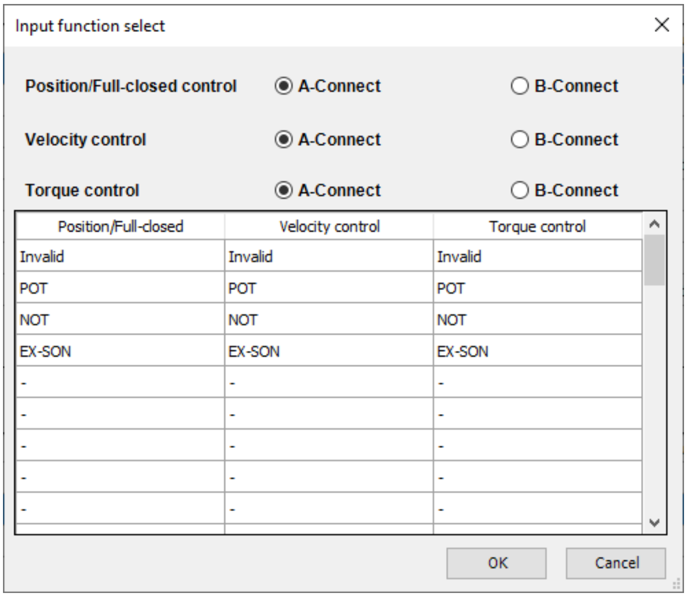
Input

Pin-No.	Position/Full-closed control	Velocity control	Torque control
05(SI1)	SI-MON5_ConnectA	SI-MON5_ConnectA	SI-MON5_ConnectA
07(SI2)	POT_ConnectB	POT_ConnectB	POT_ConnectB
08(SI3)	NOT_ConnectB	NOT_ConnectB	NOT_ConnectB
09(SI4)	SI-MON1_ConnectA	SI-MON1_ConnectA	SI-MON1_ConnectA
10(SI5)	HOME_ConnectA	HOME_ConnectA	HOME_ConnectA
11(SI6)	EXT2_ConnectA	EXT2_ConnectA	EXT2_ConnectA
12(SI7)	EXT3_ConnectA	EXT3_ConnectA	EXT3_ConnectA
13(SI8)	SI-MON4_ConnectA	SI-MON4_ConnectA	SI-MON4_ConnectA

Output

Pin-No.	Position/Full-closed control	Velocity control	Torque control
01/02(SO1)	BRK-OFF	BRK-OFF	BRK-OFF
25/26(SO2)	EX-OUT1	EX-OUT1	EX-OUT1
03/04(SO3)	ALM	ALM	ALM

2. Double-click the row of the pin number to be set.
   The function selection window will be displayed.



Input function select

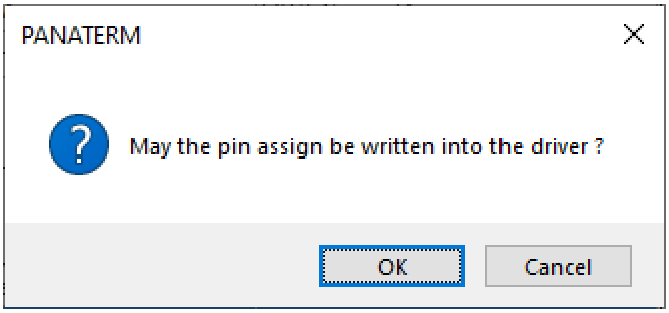
Position/Full-closed control	● A-Connect	○ B-Connect
Velocity control	● A-Connect	○ B-Connect
Torque control	● A-Connect	○ B-Connect

Position/Full-closed	Velocity control	Torque control
Invalid	Invalid	Invalid
POT	POT	POT
NOT	NOT	NOT
EX-SON	EX-SON	EX-SON
-	-	-
-	-	-
-	-	-
-	-	-
-	-	-

3. Select a function to be assigned to the pin for each control mode and a contact method (only when an input function is selected).

4. In the function selection window, click the [OK] button.
   The display will be returned to the pin assignment setting input window.

5. Click the [OK] button.
   A confirmation window will be displayed, asking whether to write to the servo amplifier.



PANATERM

May the pin assign be written into the driver ?

Click the [OK] button to write the parameter settings to the EEPROM of the servo amplifier. Click the [Cancel] button to close the window without writing the parameter settings to the EEPROM of the servo amplifier.

Note: Pin assignment settings do not take effect until the servo amplifier is restarted.

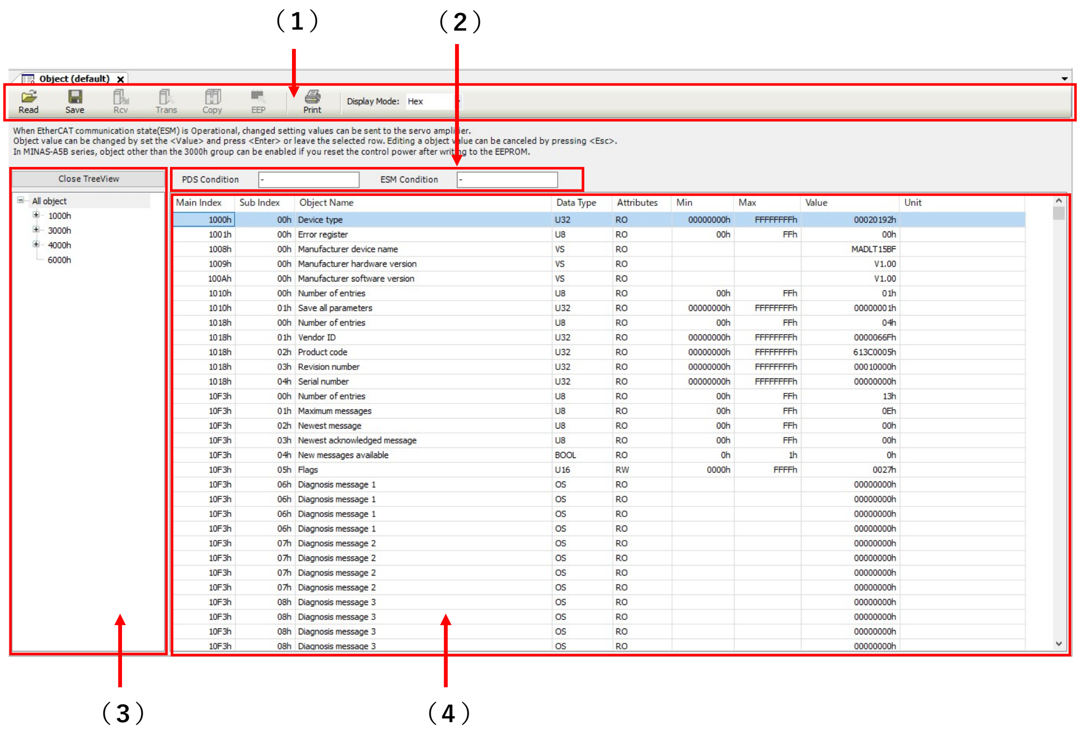
16.8 Object Window

Troubleshooting can be performed more easily by displaying and editing the list of objects on the amplifier side without using a host device.

Info.

- Object Window is for EtherCAT connection only.
- Some objects affect the behaviors of servo amplifiers or motors. Therefore, before changing the objects, carefully read the instruction manual and other technical references for the servo amplifier and pay careful attention when changing them.

16.8.1 Configuration of Object Window



Display Mode: Hex

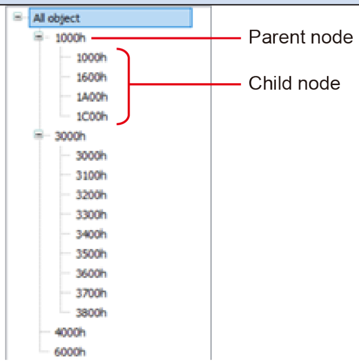
When EtherCAT communication state(ESM) is Operational, changed setting values can be sent to the servo amplifier.
Object value can be changed by set the <Value> and press <Enter> or leave the selected row. Editing a object value can be canceled by pressing <Esc>.
In MINAS-A5B series, object other than the 3000h group can be enabled if you reset the control power after writing to the EEPROM.

PDS Condition: -
ESM Condition: -

Main Index	Sub Index	Object Name	Data Type	Attributes	Min	Max	Value	Unit
1000h	00h	Device type	U32	RO	00000000h	FFFFFFFFh	00020192h	
1001h	00h	Error register	U8	RO	00h	FFh	00h	
1008h	00h	Manufacturer device name	VS	RO			MADLT15BF	
1009h	00h	Manufacturer hardware version	VS	RO			V1.00	
100Ah	00h	Manufacturer software version	VS	RO			V1.00	
1010h	00h	Number of entries	U8	RO	00h	FFh	01h	
1010h	01h	Save all parameters	U32	RO	00000000h	FFFFFFFFh	00000001h	
1018h	00h	Number of entries	U8	RO	00h	FFh	04h	
1018h	01h	Vendor ID	U32	RO	00000000h	FFFFFFFFh	0000066Fh	
1018h	02h	Product code	U32	RO	00000000h	FFFFFFFFh	613C0005h	
1018h	03h	Revision number	U32	RO	00000000h	FFFFFFFFh	00010000h	
1018h	04h	Serial number	U32	RO	00000000h	FFFFFFFFh	00000000h	
10F3h	00h	Number of entries	U8	RO	00h	FFh	13h	
10F3h	01h	Maximum messages	U8	RO	00h	FFh	0Eh	
10F3h	02h	Newest message	U8	RO	00h	FFh	00h	
10F3h	03h	Newest acknowledged message	U8	RO	00h	FFh	00h	
10F3h	04h	New messages available	BOOL	RO	0h	1h	0h	
10F3h	05h	Flags	U16	RW	0000h	FFFFh	0027h	
10F3h	06h	Diagnosis message 1	OS	RO			00000000h	
10F3h	06h	Diagnosis message 1	OS	RO			00000000h	
10F3h	06h	Diagnosis message 1	OS	RO			00000000h	
10F3h	06h	Diagnosis message 1	OS	RO			00000000h	
10F3h	07h	Diagnosis message 2	OS	RO			00000000h	
10F3h	07h	Diagnosis message 2	OS	RO			00000000h	
10F3h	07h	Diagnosis message 2	OS	RO			00000000h	
10F3h	07h	Diagnosis message 2	OS	RO			00000000h	
10F3h	08h	Diagnosis message 3	OS	RO			00000000h	
10F3h	08h	Diagnosis message 3	OS	RO			00000000h	
10F3h	08h	Diagnosis message 3	OS	RO			00000000h	
10F3h	08h	Diagnosis message 3	OS	RO			00000000h	

<table>
<tr><th>No.</th><th>Name</th><th colspan="3">Function</th></tr>
<tr><td rowspan="4">(1)</td><td rowspan="4">Toolbar</td><td colspan="3">The toolbar consists of basic operation commands related to objects, such as save and read.</td></tr>
<tr><td>Icon</td><td>Name</td><td>Function</td></tr>
<tr><td>Read</td><td>Read</td><td>Reads objects from the file (.obj5).</td></tr>
<tr><td>Save</td><td>Save</td><td>Writes objects to the file (.obj5).</td></tr>
</table>

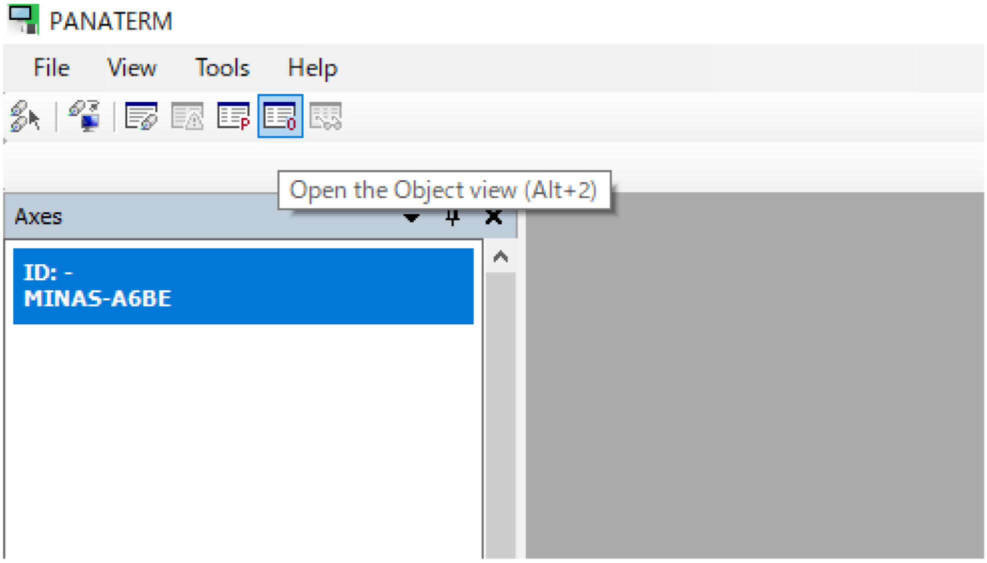
No.	Name	Function		
		Rcv	Rcv	Receives objects from the servo amplifier.
		Trans	Trans	Transmits objects to the servo amplifier. If you turn OFF the power supply to the servo amplifier without writing to EEPROM, the object will be reverted back to the value before the change was made.
		Copy	Copy	Copies the objects of a servo amplifier to servo amplifiers for other axes.
		EEP	EEP	Writes objects to EEPROM of the servo amplifier. Do not turn OFF the power supply to the servo amplifier and the PC while data is being written to EEPROM. If the power supply is cut off while data is being written, the data content is not warranted.
		Print	Print	Prints objects.
		Hex ▾	Display mode	Changes the numerical display of the object being displayed. Hex: Displayed in hexadecimal number and “h” is placed at the end of the number. Dec: Displayed in decimal number and a sign is set. Bin: Displayed in binary number and “b” is placed at the end of the number. The values in the Min-Max column are displayed in hexadecimal number.
(2)	Condition monitor	**PDS Condition** Displays the PDS condition on the servo amplifier side. It varies depending on the object value of 6041h-00h.		
		ESM Condition Displays the condition that indicates whether the object can be rewritten on the servo amplifier side. Rewriting is possible when “Operational” is displayed.		
(3)	Object tree	When a node is selected from the object tree, related objects are displayed in the object setting field. Operate **Close TreeView** or **Open TreeView** to hide or display the object tree. Select “All object” to display all objects. When a parent node is selected, all child node objects under the selected node are displayed. When a child node is selected, objects of the selected node are displayed. For details of each object, refer to the instruction manual and other technical references for the servo amplifier.		

<table>
<tr><th>No.</th><th>Name</th><th colspan="2">Function</th></tr>
<tr><td></td><td></td><td colspan="2">
</td></tr>
<tr><td rowspan="12">(4)</td><td rowspan="12">Object setting field</td><td colspan="2">Allows the user to edit and set an object.</td></tr>
<tr><td>Name</td><td>Function</td></tr>
<tr><td><b>Main Index</b></td><td>Displays the Main Index of an object.</td></tr>
<tr><td><b>Main Index</b></td><td>Displays the Sub Index of an object.</td></tr>
<tr><td><b>Object Name</b></td><td>Displays the name of an object.</td></tr>
<tr><td><b>Data Type</b></td><td>Displays the data type of an object.
I8: Integer 8 type
I16: Integer 16 type
I32: Integer 32 type
U8: Unsigned 8 type
U16: Unsigned 16 type
U32: Unsigned 32 type
Bool: Boolean type
OS: Octet String type
VS: Visible String type</td></tr>
<tr><td><b>Attributes</b></td><td>Displays the attributes of an object.
RO: Read-only attribute, exclusively for reading.
RW: Read/write attribute, possible to read or write.</td></tr>
<tr><td><b>Min</b></td><td rowspan="2">Displays the setting range of an object.
The setting range is not displayed for an object whose Data Type is “OS” or “VS”.</td></tr>
<tr><td><b>Max</b></td></tr>
<tr><td><b>Value</b></td><td>Displays the value of an object. The value can be changed if the object attribute is “RW” and the set value is a numerical value.
Note that there are input restrictions for each Display mode setting. The following values can be input.
Hex: 0 to 9, A to F (After editing, “"h"” is automatically placed at the end of the number.)
Dec: 0 to 9 and minus sign “-”
Bin: 0 and 1 (“"b"” is automatically placed at the end of the number.)
After changing the value, press the <ENTER> key or click the [Trans] button.
To return a value to its original value press the <ESC> key.</td></tr>
<tr><td><b>Unit</b></td><td>Displays the unit of an object set value.</td></tr>
</table>

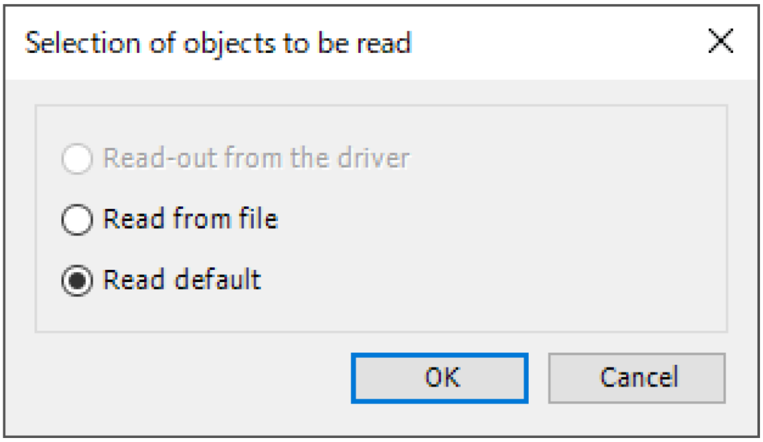
16.8.2 Setting Objects

1 2 Procedure

1. From the menu bar on the main pane, select **View**>**Object**. Alternatively, on the toolbar, click the "Open the Object view" icon.



The "Selection of objects to be read" dialog box will be displayed.



Selection of objects to be read

○ Read-out from the driver

○ Read from file

● Read default

OK | Cancel

□**Read-out from the amplifier.**

Communicates with the connected servo amplifier and reads objects set in the servo amplifier.

If this mode is selected, object values will be reflected in the servo amplifier as soon as they are changed.

□Read from file

Reads the parameter file ("obj5") that was edited previously. If communication is performed with the servo amplifier, object values will be reflected in the servo amplifier as soon as they are changed.

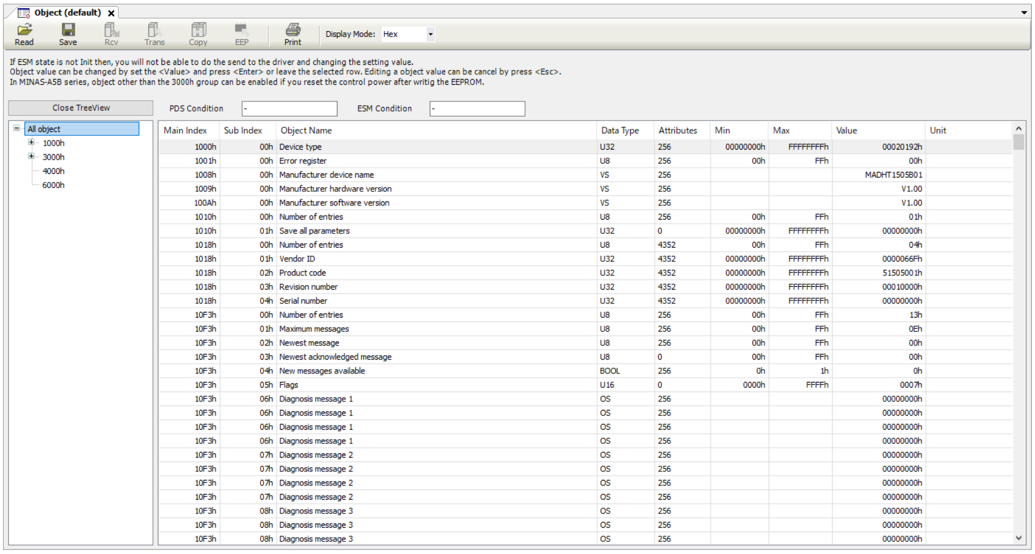
□Read default

Reads the standard default settings of the servo amplifier that were saved during installation.

If communication is performed with the servo amplifier, object values will be reflected in the servo amplifier as soon as they are changed.

2. Select one of the three options above for reading the object settings and click the [OK] button.

   The Object window will be displayed.



Display Mode: Hex

If ESM state is not Init then, you will not be able to do the send to the driver and changing the setting value.
Object value can be changed by set the <Value> and press <Enter> or leave the selected row. Editing a object value can be cancel by press <Esc>.
In MINAS-A5B series, object other than the 3000h group can be enabled if you reset the control power after writig the EEPROM.

PDS Condition: - ESM Condition: -

Main Index	Sub Index	Object Name	Data Type	Attributes	Min	Max	Value	Unit
1000h	00h	Device type	U32	256	00000000h	FFFFFFFFh	00020192h	
1001h	00h	Error register	U8	256	00h	FFh	00h	
1008h	00h	Manufacturer device name	VS	256			MADHT1505B01	
1009h	00h	Manufacturer hardware version	VS	256			V1.00	
100Ah	00h	Manufacturer software version	VS	256			V1.00	
1010h	00h	Number of entries	U8	256	00h	FFh	01h	
1010h	01h	Save all parameters	U32	0	00000000h	FFFFFFFFh	00000000h	
1018h	00h	Number of entries	U8	4352	00h	FFh	04h	
1018h	01h	Vendor ID	U32	4352	00000000h	FFFFFFFFh	0000066Fh	
1018h	02h	Product code	U32	4352	00000000h	FFFFFFFFh	51505001h	
1018h	03h	Revision number	U32	4352	00000000h	FFFFFFFFh	00010000h	
1018h	04h	Serial number	U32	4352	00000000h	FFFFFFFFh	00000000h	
10F3h	00h	Number of entries	U8	256	00h	FFh	13h	
10F3h	01h	Maximum messages	U8	256	00h	FFh	0Eh	
10F3h	02h	Newest message	U8	256	00h	FFh	00h	
10F3h	03h	Newest acknowledged message	U8	0	00h	FFh	00h	
10F3h	04h	New messages available	BOOL	256	0h	1h	0h	
10F3h	05h	Flags	U16	0	0000h	FFFFh	0007h	
10F3h	06h	Diagnosis message 1	OS	256			00000000h	
10F3h	06h	Diagnosis message 1	OS	256			00000000h	
10F3h	06h	Diagnosis message 1	OS	256			00000000h	
10F3h	06h	Diagnosis message 1	OS	256			00000000h	
10F3h	07h	Diagnosis message 2	OS	256			00000000h	
10F3h	07h	Diagnosis message 2	OS	256			00000000h	
10F3h	07h	Diagnosis message 2	OS	256			00000000h	
10F3h	07h	Diagnosis message 2	OS	256			00000000h	
10F3h	08h	Diagnosis message 3	OS	256			00000000h	
10F3h	08h	Diagnosis message 3	OS	256			00000000h	
10F3h	08h	Diagnosis message 3	OS	256			00000000h	

3. After changing the object settings, click the [EEP] button to write the object settings to the EEPROM of the servo amplifier.

4. Click the [×] button on the Object window to close the Object window.

16.9 MINAS Parameters for the GM1 Controller

Some parameters for servo amplifiers on the A5B / A6B side affect the behaviors of the EtherCAT-compatible GM1 controller. Use the following parameters.

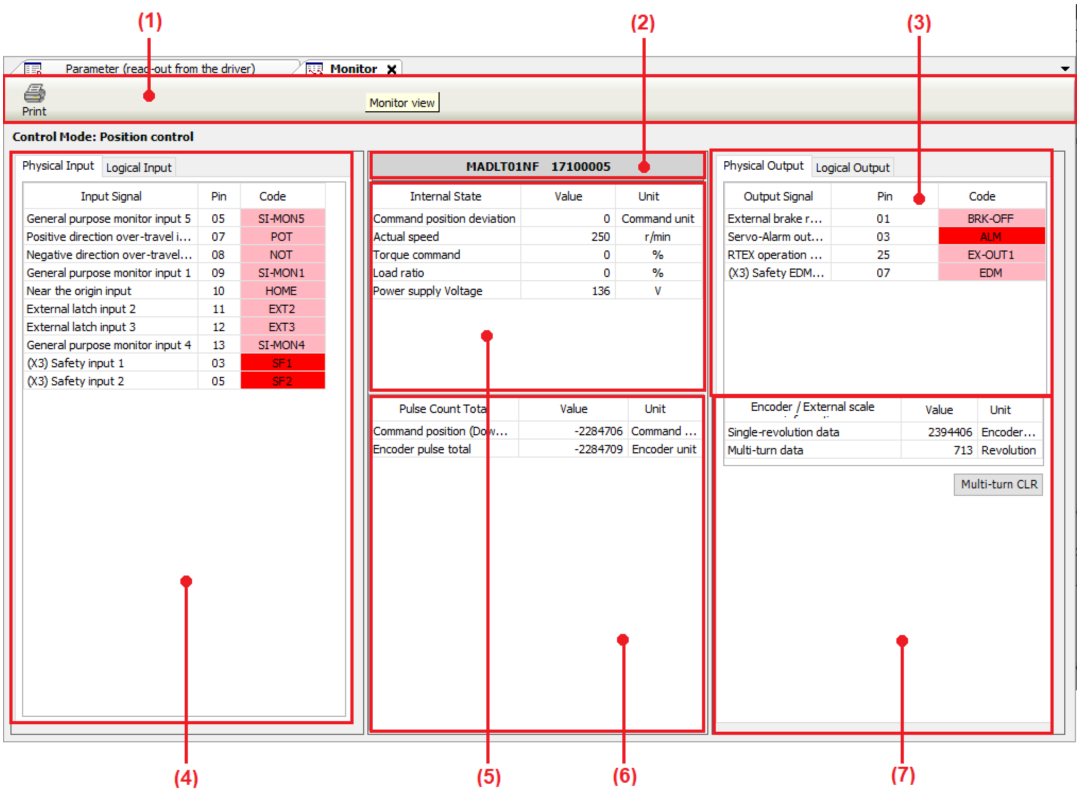
No.	Name	Settings	Standard factory default setting
Pr5.04	Over-travel inhibit input setup	Use setting value 1. (Recommended)	1(Note 1)

(Note 1) We recommend that the set value should not be changed judging from the characteristics of the GM1 and MINAS.

16.10 Monitor Window

The Monitor window displays the operating states of servo amplifiers and motors, I/O signals, internal statuses, and other information and also allows the user to check them.

16.10.1 Configuration of Monitor Window



Control Mode: Position control

Input Signal	Pin	Code
General purpose monitor input 5	05	SI-MON5
Positive direction over-travel i...	07	POT
Negative direction over-travel...	08	NOT
General purpose monitor input 1	09	SI-MON1
Near the origin input	10	HOME
External latch input 2	11	EXT2
External latch input 3	12	EXT3
General purpose monitor input 4	13	SI-MON4
(X3) Safety input 1	03	SF1
(X3) Safety input 2	05	SF2

MADLT01NF 17100005

Internal State	Value	Unit
Command position deviation	0	Command unit
Actual speed	250	r/min
Torque command	0	%
Load ratio	0	%
Power supply Voltage	136	V

Pulse Count Tota	Value	Unit
Command position (Dow...	-2284706	Command ...
Encoder pulse total	-2284709	Encoder unit

Output Signal	Pin	Code
External brake r...	01	BRK-OFF
Servo-Alarm out...	03	ALM
RTEX operation ...	25	EX-OUT1
(X3) Safety EDM...	07	EDM

Encoder / External scale	Value	Unit
Single-revolution data	2394406	Encoder...
Multi-turn data	713	Revolution

NO.	Name	Description
(1)	Toolbar	The toolbar consists of basic operation commands related to parameters. **Icon** / **Name** / **Function** Print / Print / Prints the contents of the Monitor window.
(2)	Amplifier model name and serial number	Displays the model name and serial number of the servo amplifier.
(3)	Output signal status monitor	Displays the status of each output signal. The tab can be switched between "Physical Output" and "Logical Output". Physical Output – Displays the status of output signals from the servo amplifier. Red: Indicates that output transistor is ON Pink: Indicates that output transistor is OFF

<table>
<tr><th>NO.</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td>Logical Output – Displays the status of signals within the servo amplifier.
Red: Indicates that signal status is active
Pink: Indicates that signal status is inactive</td></tr>
<tr><td>(4)</td><td>Input signal status monitor</td><td>Displays the status of input signals. The tab can be switched between "Physical Input" and "Logical Input".
Physical Input – Displays the status of input signals to the servo amplifier.
Red: Indicates that COM- is connected
Pink: Indicates that signal status is open
Logical Input – Displays the status of signals within the servo amplifier.
Red: Indicates that signal status is active
Pink: Indicates that signal status is inactive</td></tr>
<tr><td>(5)</td><td>Internal status monitor</td><td>Displays the internal status of the servo amplifier.
<table>
<tr><th>Name</th><th>Function</th></tr>
<tr><td>Commanded position deviation</td><td>Displays the position deviation of a command unit.</td></tr>
<tr><td>Actual speed</td><td>Displays the monitor speed</td></tr>
<tr><td>Torque command</td><td>Displays the torque command.</td></tr>
<tr><td>Load factor</td><td>Displays the ratio relative to the rated load. Adjust the operation pattern so that 100% is not exceeded.</td></tr>
<tr><td>Power supply voltage value</td><td>Displays the voltage (voltage between the P and N terminals) of power supply to the servo amplifier.</td></tr>
</table></td></tr>
<tr><td>(6)</td><td>Pulse sum monitor</td><td>Displays the sum of command and encoder pulses received by the servo amplifier.</td></tr>
<tr><td>(7)</td><td>Encoder information monitor</td><td>Displays encoder information.
<table>
<tr><td>Single-turn data</td><td>Displays an absolute position when the motor makes no more than a single turn.</td></tr>
<tr><td>Multi-turn data</td><td>Displays how many turns the motor made after "Clear" operation.</td></tr>
</table>
Clicking "Clear Multi-turn" resets the multi-turn data stored in the encoder to "0" and clears all encoder errors.
Note: Before using "Clear Multi-turn", check the precautions on use. To clear encoder errors, you may need to restart the servo amplifier.</td></tr>
</table>

(Note 1) Because Ethernet communication is used to transfer data between the servo amplifier and PC, there is a difference or delay between the value displayed on the screen and the actual value of the servo amplifier.

(Note 2) When the polarity is "+", symbol "+" is not displayed.

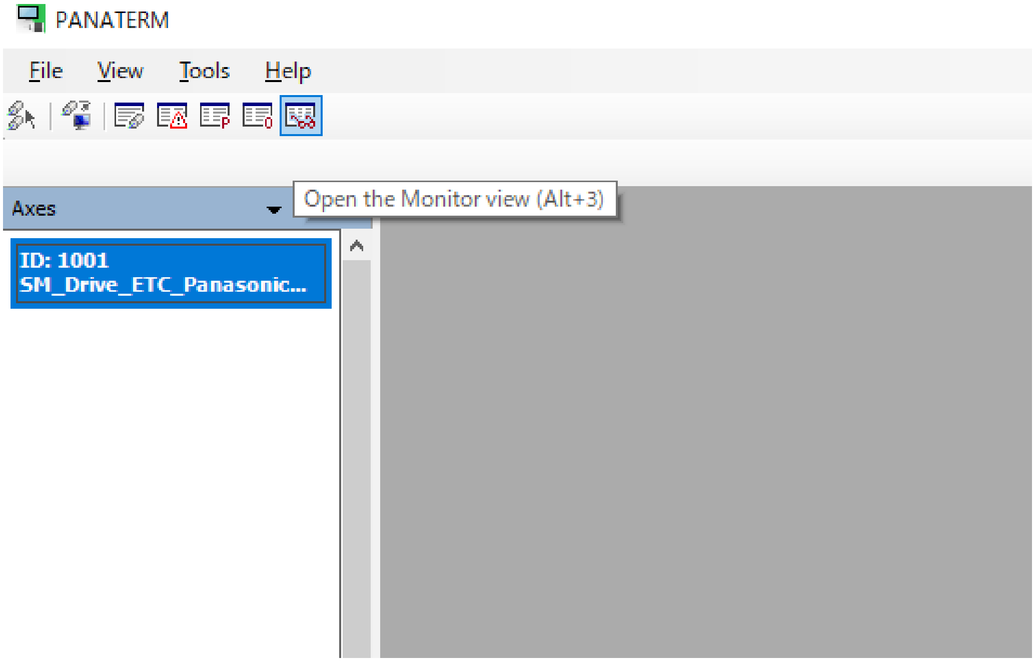
(Note 3) The monitor function is not a measuring instrument. Use the values displayed in the Monitor window as a guide.

(Note 4) If the servo amplifier outputs "Error 40.0　Error protection from absolute system failure" or "Error 42.0　Error protection from absolute overspeed", execute "Clear Multi-turn". Unless the absolute encoder is reset, the alarm cannot be cleared.

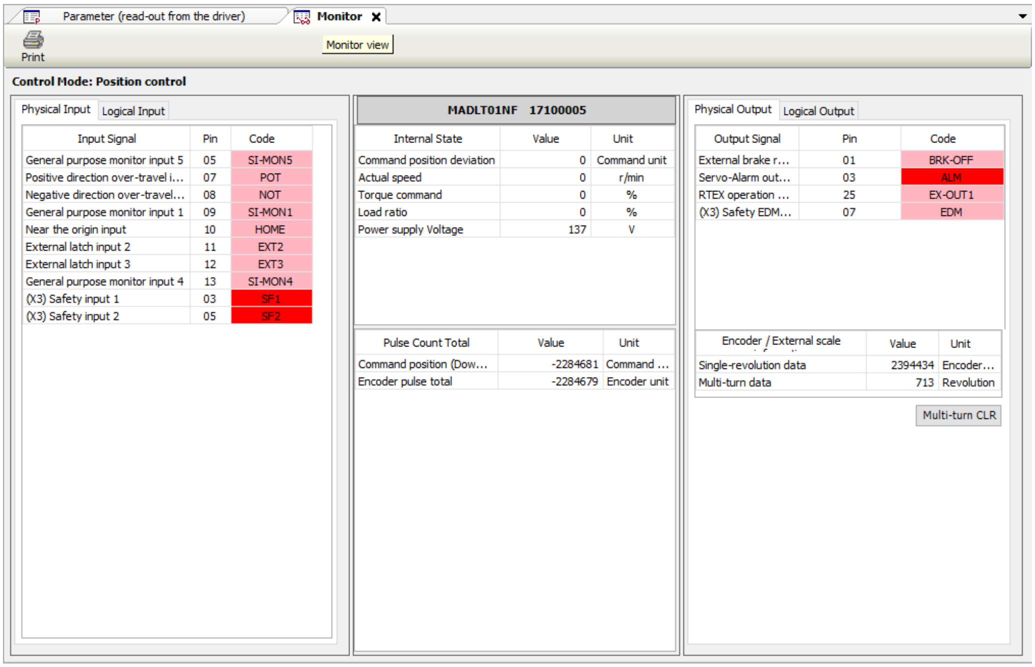
16.10.2 Checking the Monitor Window

Procedure

1. From the menu bar on the main pane, select **View**>**Monitor**. Alternatively, on the toolbar, click the "Open the Monitor view" icon.



The Monitor window will be displayed.



Control Mode: Position control

Input Signal	Pin	Code
General purpose monitor input 5	05	SI-MON5
Positive direction over-travel i...	07	POT
Negative direction over-travel...	08	NOT
General purpose monitor input 1	09	SI-MON1
Near the origin input	10	HOME
External latch input 2	11	EXT2
External latch input 3	12	EXT3
General purpose monitor input 4	13	SI-MON4
(X3) Safety input 1	03	SF1
(X3) Safety input 2	05	SF2

MADLT01NF 17100005

Internal State	Value	Unit
Command position deviation	0	Command unit
Actual speed	0	r/min
Torque command	0	%
Load ratio	0	%
Power supply Voltage	137	V

Pulse Count Total	Value	Unit
Command position (Dow...	-2284681	Command ...
Encoder pulse total	-2284679	Encoder unit

Output Signal	Pin	Code
External brake r...	01	BRK-OFF
Servo-Alarm out...	03	ALM
RTEX operation ...	25	EX-OUT1
(X3) Safety EDM...	07	EDM

Encoder / External scale	Value	Unit
Single-revolution data	2394434	Encoder...
Multi-turn data	713	Revolution

2. Check each item.
   Check the input signal state, output signal state, and the internal status of the servo amplifier.

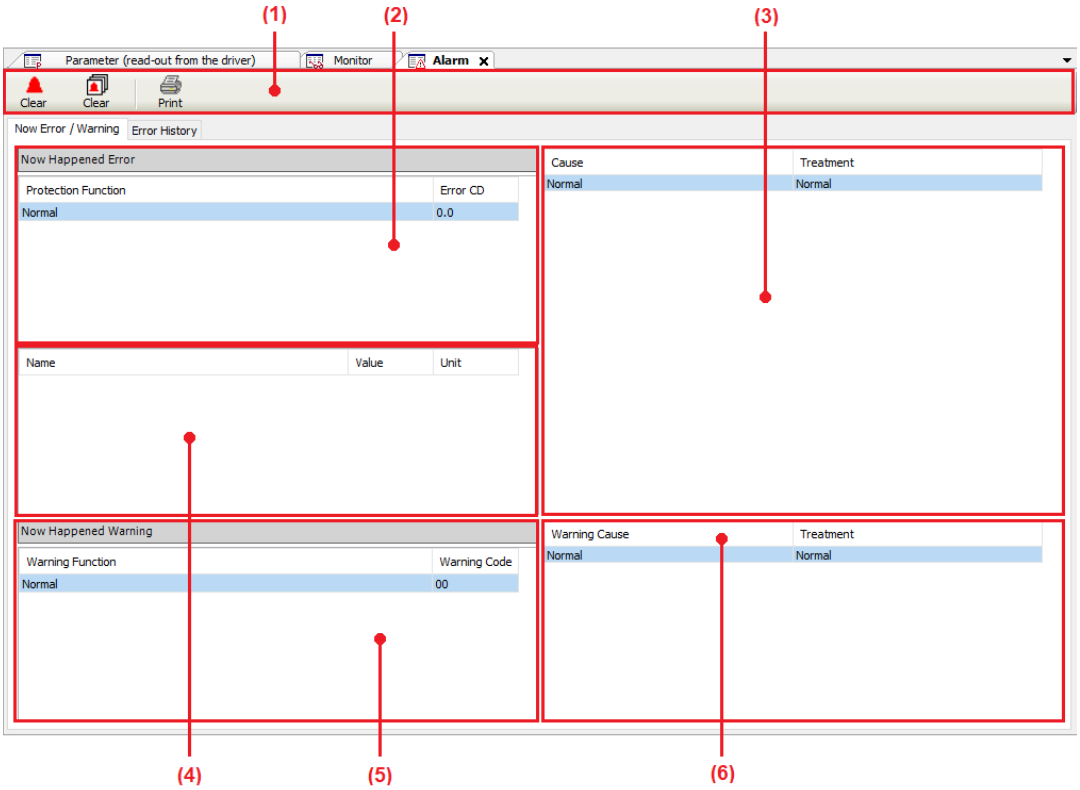
3. Click the [×] button on the Monitor window.
   The Monitor window will be closed.

16.11 Alarm Window

If the front panel of the servo amplifier is blinking, such as when the motor is not working, you can check the alarm status.Please note that the alarm cannot be cleared.

16.11.1 Configuration of Alarm Window

Display of the current errors and warnings (only during communication with servo amplifier)



Now Happened Error

Protection Function	Error CD
Normal	0.0

Cause	Treatment
Normal	Normal

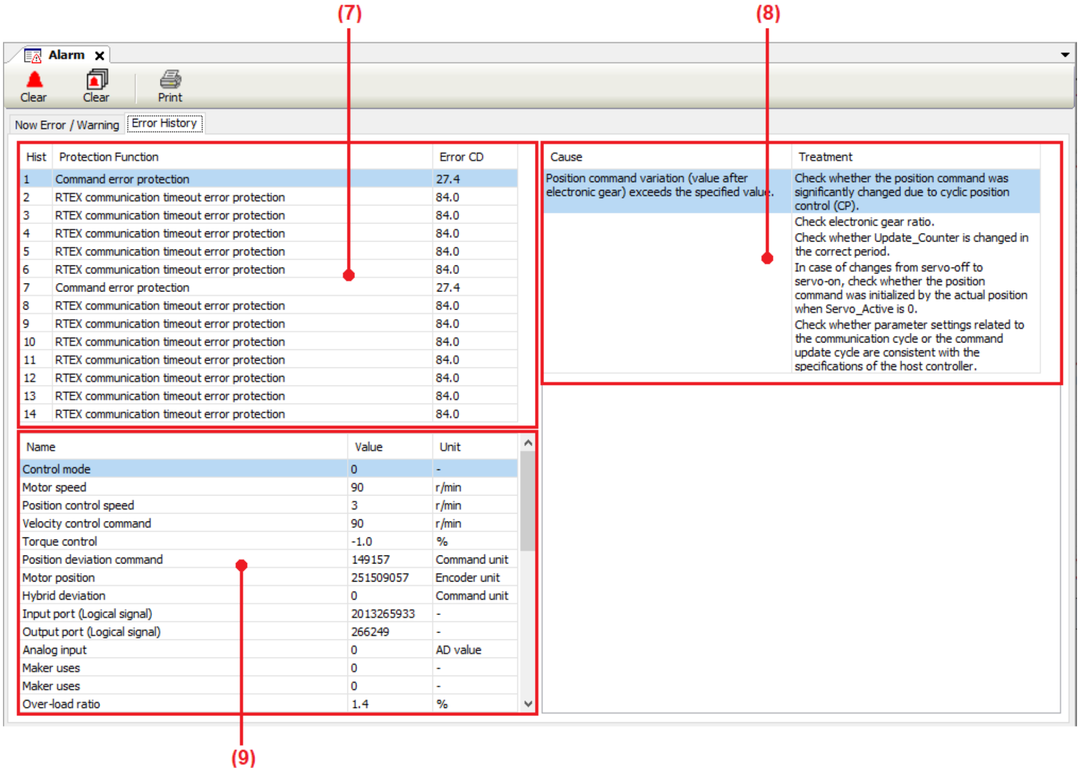
Name	Value	Unit

Now Happened Warning

Warning Function	Warning Code
Normal	00

Warning Cause	Treatment
Normal	Normal

Display of error histories



Hist	Protection Function	Error CD
1	Command error protection	27.4
2	RTEX communication timeout error protection	84.0
3	RTEX communication timeout error protection	84.0
4	RTEX communication timeout error protection	84.0
5	RTEX communication timeout error protection	84.0
6	RTEX communication timeout error protection	84.0
7	Command error protection	27.4
8	RTEX communication timeout error protection	84.0
9	RTEX communication timeout error protection	84.0
10	RTEX communication timeout error protection	84.0
11	RTEX communication timeout error protection	84.0
12	RTEX communication timeout error protection	84.0
13	RTEX communication timeout error protection	84.0
14	RTEX communication timeout error protection	84.0

Cause	Treatment
Position command variation (value after electronic gear) exceeds the specified value.	Check whether the position command was significantly changed due to cyclic position control (CP). Check electronic gear ratio. Check whether Update_Counter is changed in the correct period. In case of changes from servo-off to servo-on, check whether the position command was initialized by the actual position when Servo_Active is 0. Check whether parameter settings related to the communication cycle or the command update cycle are consistent with the specifications of the host controller.

Name	Value	Unit
Control mode	0	-
Motor speed	90	r/min
Position control speed	3	r/min
Velocity control command	90	r/min
Torque control	-1.0	%
Position deviation command	149157	Command unit
Motor position	251509057	Encoder unit
Hybrid deviation	0	Command unit
Input port (Logical signal)	2013265933	-
Output port (Logical signal)	266249	-
Analog input	0	AD value
Maker uses	0	-
Maker uses	0	-
Over-load ratio	1.4	%

<table>
<tr><th>No.</th><th>Name</th><th>Description</th></tr>
<tr><td>(1)</td><td>Toolbar</td><td>
<table>
<tr><th>Icon</th><th>Name</th><th>Function</th></tr>
<tr><td>Clear</td><td>Clear</td><td>Allows the user to clear the current alarm history. If you click this button after eliminating the cause of the alarm, the current alarm will be cleared and normal operation will be performed. However, you cannot clear any alarms that cannot be cleared by alarm clear input signals of servo amplifiers. In such a case, turn off the servo amplifier, eliminate the cause of the alarm, and then turn the power on again.</td></tr>
<tr><td>Clear</td><td>Clear</td><td>Allows the user to clear error histories.</td></tr>
<tr><td>Print</td><td>Print</td><td>Prints error-related information.</td></tr>
</table>
</td></tr>
<tr><td>(2)</td><td>Current error display area</td><td>Displays the alarm numbers and names of all errors that are currently occurring.
The alarm displayed on the top of the list is the alarm displayed on the front panel of the servo amplifier.</td></tr>
</table>

No.	Name	Description
(3)	Error cause / treatment display area	Displays the cause and treatment of the selected error.
(4)	Motor internal status display area	Displays the motor internal status in the event of an alarm.
(5)	Current warning display area	Displays the warning numbers and names of all warnings that are currently occurring.
(6)	Warning cause / treatment display area	Displays the cause and treatment of the selected warning.
(7)	Error history display area	Displays the order of error histories, alarm numbers, and error names.
(8)	Error cause / treatment display area	Displays the cause and treatment of the selected error.
(9)	Motor internal status display area	Displays the motor internal status in the event of an alarm.

(Note 1) Some alarms cause tripping as errors but are not recorded in error histories. For alarms that are not recorded in error histories, refer to the instruction manual of the servo amplifier.

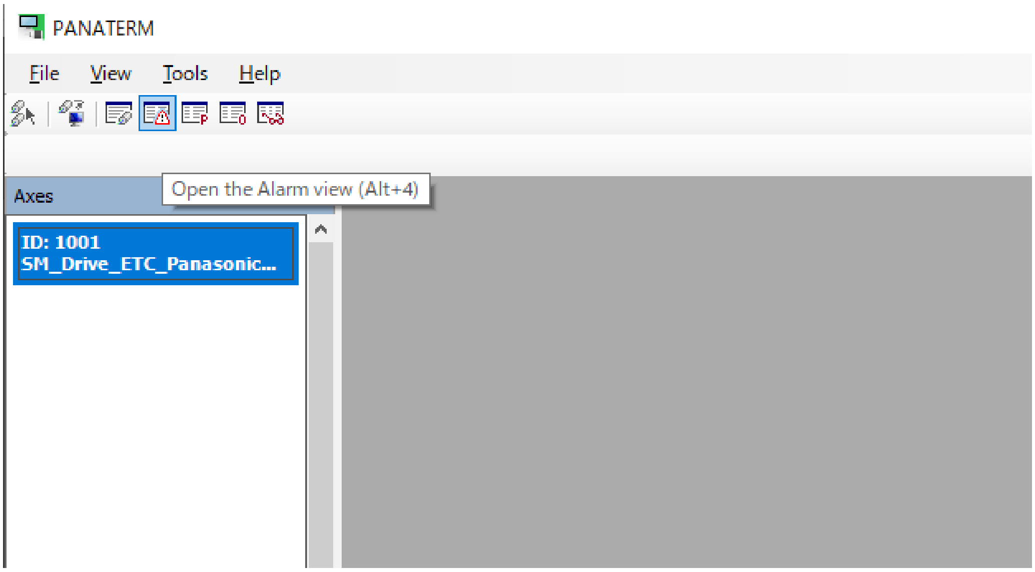
(Note 2) Up to 14 error histories are stored. When more than 14 errors occur, error histories are erased in chronological order (the oldest error history is erased first).

(Note 3) Up to three histories of motor internal status in the event of an alarm are stored. If an alarm occurs immediately after the power is turned on, motor internal status may not be captured normally.

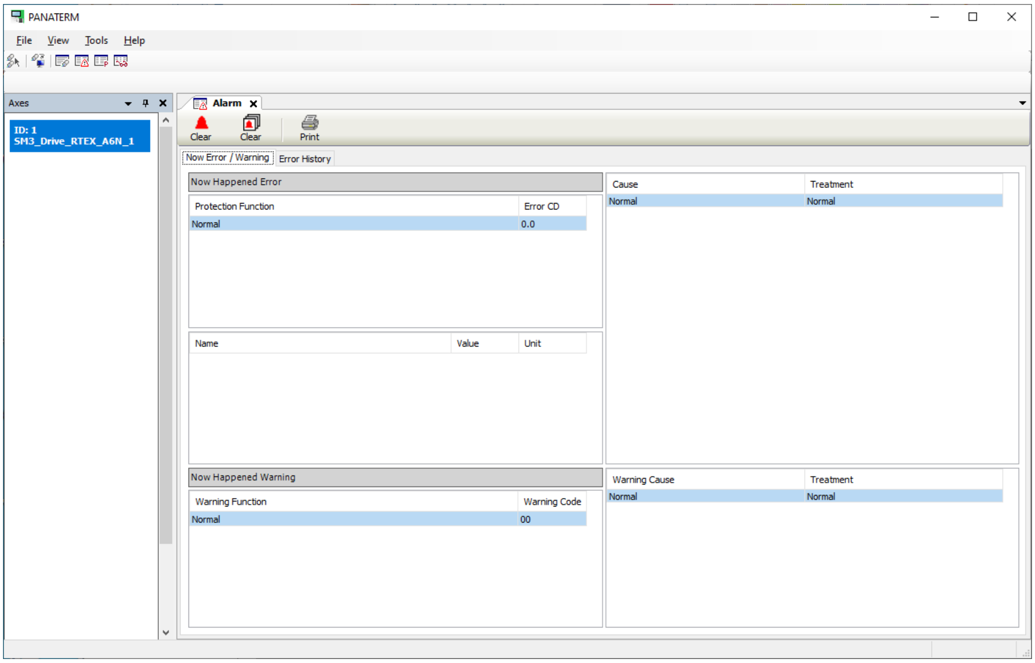
16.11.2 Checking Alarms

1 2 Procedure

1. From the menu bar on the main pane, select **View**>**Alarm**. Alternatively, on the toolbar, click the "Open the Alarm view" icon.



The Alarm window will be displayed.



Now Happened Error

Protection Function	Error CD
Normal	0.0

Cause	Treatment
Normal	Normal

Name	Value	Unit

Now Happened Warning

Warning Function	Warning Code
Normal	00

Warning Cause	Treatment
Normal	Normal

2. Check for any errors that are currently occurring.
   Click the "Now Error / Warning" tab and check for any errors that are currently occurring.

3. Check for any errors that occurred in the past.
   Click the "Error History" tab and check for any errors that occurred in the past.

4. Click the [×] button on the Alarm window.
   The Alarm window will be closed.

16.12 Other Functions

16.12.1 Language Setting Function

This function allows the user to set the display language of PANATERM Lite for GM.
The default setting is the same language as the one set in GM Programmer.

1 2 Procedure

1. Select a language from the menu bar tool.
   The language set in PANATERM Lite for GM will be switched.

Info.

- The display language setting of PANATERM Lite for GM is linked with that of the GM Programmer.

16.12.2 Help Function

While performing operation in PANATERM Lite for GM, you can start the Help function to check information such as operating methods.

1 2 Procedure

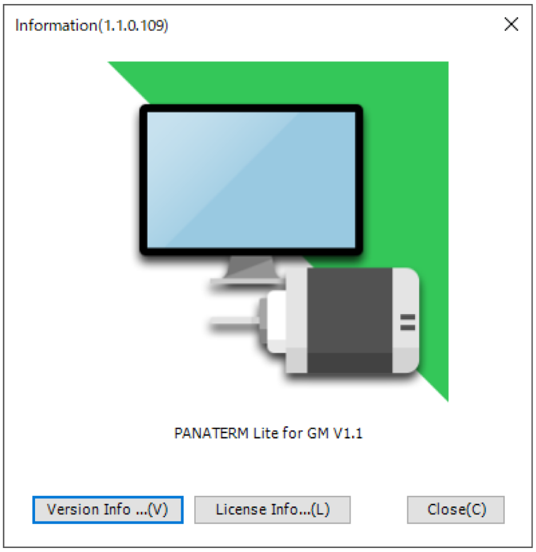
1. From the menu bar, select **Help**>**PANATERM Lite for GM Help**.
   "PANATERM Lite for GM Operation Guide" will be started.

16.12.3 Version Display Function

This function allows the user to check the version, license, and other information for PANATERM Lite for GM.

1 2 Procedure

1. From the menu bar, select **Help**>**About**.



Information(1.1.0.109)

PANATERM Lite for GM V1.1

2. Click a desired button at the bottom of the window.

Button	Description
Version Info	Displays information about the plug-ins that have been applied and the operating system of the PC that is used.
License Info	Displays license information for the software used by PANATERM Lite for GM.

16.13 Troubleshooting for Servo Amplifiers and Motors

This section explains how to resolve problems.

16.13.1 I Cannot Set up

Symptom	Action method
I cannot set up	● Refer to the chapter related to the system configuration that you need, and check that the PC in which you install the software satisfies the necessary conditions. In particular, note that all the necessary service packs for the operating system must have been applied. ● The installer may have been damaged due to download failure. Clear the browser cache and then download again.

16.13.2 I Cannot Communicate

Symptom	Solution
After PANATERM Lite for GM is started, the servo amplifier is not recognized in the window for connecting to the servo amplifier.	● Check that the power to the control circuit of the servo amplifier is turned on. ● Check the USB communication cable for loose connections or breakage or check whether a correct cable is used. ● Check whether the USB port on the PC is functioning normally. (Refer to the instruction manual of the PC.) ● Check whether the USB driver is installed correctly.
After PANATERM Lite for GM is started, the GM1 controller is not recognized in the window for connecting to the GM1 controller.	● Check that the power to the GM1 controller is turned on. ● Check the USB communication cable or Ethernet cable for loose connections or breakage or check whether a correct cable is used. ● Check whether the USB port on the PC is functioning normally. (Refer to the instruction manual of the PC.) ● Check whether the LAN adapter on the PC is functioning normally. (Refer to the instruction manual of the PC.) ● Check whether the USB driver is installed correctly. ● Check whether Gateway is running.

16.13.3 I Cannot Print

Symptom	Action method
I cannot print	● Check whether the printer is connected correctly. Check that the printer driver is operating normally, by printing a test page, for example. ● You may be unable to print any comment that consists of too many characters in a single line. Split such a comment into multiple lines so that the number of characters per line falls within the printable range.

16.13.4 I Cannot Set up Axes

Symptom	Solution
The number of servo amplifiers connected does not match the number of servo amplifiers checked by performing a search.	● Check whether the axis name (ID) of the servo amplifier connected to the PC is 0, the respective axis names (IDs) of other servo amplifiers are 1 to 15, and there are any duplicate axis names (IDs). ● Check the communication cable for loose connections or breakage or check whether a correct cable is used.

16.13.5 PANATERM Lite for GM Does Not Behave Normally

Symptom	Solution
PANATERM Lite for GM responds or acts slowly	● Close any windows that are not used. All windows communicate with the servo amplifier in certain cycles even if they are hidden under other windows. ● If USB devices other than servo amplifiers are connected, reduce USB communication load by pausing their operations or taking some other action.
The window does not open or icons appear garbled	● The PC is running out of memory. Temporarily close PANATERM Lite for GM and other applications that are not used. Alternatively, turn the PC off and then on, and start PANATERM Lite for GM again.
PANATERM Lite for GM does not respond	● Press the <Ctrl> key + <Alt> key + <Delete> key simultaneously to invoke the window for forcibly terminating programs, and terminate PANATERM Lite for GM.
PANATERM Lite for GM suddenly terminates	● Start PANATERM Lite for GM again.
PANATERM Lite for GM does not start	● Microsoft .NET Framework Ver. 4.6.1 may have failed to be installed. Refer to the website of Microsoft and install Microsoft .NET Framework Ver. 4.6.1 directly on the PC that you use. After installing Microsoft .NET Framework Ver. 4.6.1, run the PANATERM Lite for GM installer again.

16.13.6 The Parameter Window Does Not Behave Normally

Symptom	Solution
The changed parameter value returns to its original value	● After changing the parameter value, press the <Enter> key or click the [Trans] button. If you move to another parameter or make changes in the window without performing either of these operations, any change to the parameter value will be canceled. ● When the values read from a file are displayed in the window, changed parameter values are not sent to the servo amplifier. To send changed parameter values to the servo amplifier, click the [Trans] button.

16.13.7 The Object Window Does Not Behave Normally

Symptom	Solution
The object window does not open.	● The object editor window can be displayed only when a series that uses the EtherCAT communication is selected. (Example: MINAS-A5B, MINAS-A6B)
The object set value cannot be changed or transmitted.	● Check if the ESM Condition is set to “Operational”.
The object set value returns to its original value.	● After changing the object value, press the <Enter> key or click the [Trans] button. If you move to another object or make changes in the window without performing either of these operations, any change to the object value will be canceled. ● When the values read from a file are displayed in the window, changed object values are not sent to the servo amplifier. To send changed object values to the servo amplifier, click the [Trans] button.

16.13.8 The Monitor Window Does Not Behave Normally

Symptom	Solution
Monitor display does not change	● There is a possibility that communication with the servo amplifier has been disrupted and the PC is in an offline state. Check the connection status of the servo amplifier.

16.13.9 The Alarm Window Does Not Behave Normally

Symptom	Action method
Error histories are not displayed	● If no error has occurred before or error histories have been cleared, error histories are not displayed. ● Supplementary information about errors is displayed only when the most recent, the second most recent, or the third most recent error history is selected. Select an error history number again. ● Errors that are not recorded in error histories are not displayed in error histories even if they occur.

Appendix Warranty / Cautions for Proper Use

Warranty

Warranty Period

The warranty period of the Product shall be 12 months from the ex-factory date or 18 months from the date of manufacturing unless otherwise specified between both parties.

Warranty Scope

Panasonic warrants the replacement of the defected parts of the Product or repair of them when the defects of the Product occur during the Warranty Period, and when the defects are under Panasonic responsibility. This Warranty only covers the Product itself and does not cover any damage to your company and the third party incurred by the Product, such as damage that is induced by an object machined or produced using the Product or by the defects of the Product. This Warranty shall be exempted in the following cases,

1. Defects resulting from misuse and/or repair or modification by the customer.
2. Defects resulting from drop of the Product or damage during transportation.
3. Defects resulting from improper usage of the Product beyond the Specifications.
4. Defects resulting from fire, earthquake, lightening, flood, damage from salt, abnormal voltage or other Act of God, or other disaster.
5. Defects resulting from the intrusion of foreign material to the Product, such as water, oil or metallic particles.
6. Parts exceeding their standard lifetime specified in this document.
7. The machines are not assembled in accordance with the instructions or precautions noted in this specification.
8. When the machine does not match the Product assembled in the machine.
9. When the machine condition is not caused by Panasonic reasons.
10. Defects that Panasonic could not foresee at the time of delivery of the Prodcuct.

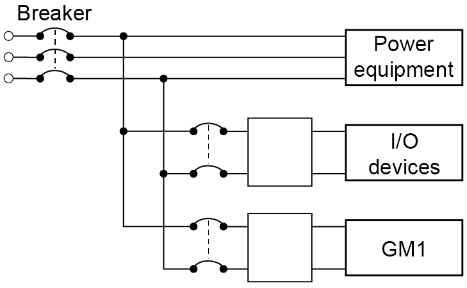
Cautions for Proper Use

■ Selection of a power supply

- Use a low noise power supply.
- The inherent noise resistance is sufficient for the noise superimposed on the power wires, however, the noise can be attenuated further by using the isolation transformer / insulated power supply.

■ Isolation of power supply systems

- Wiring to the units, I/O devices, and other power devices should have separate wiring systems.



Isolation transformer
Insulated DC power supply

■ Power supply sequence

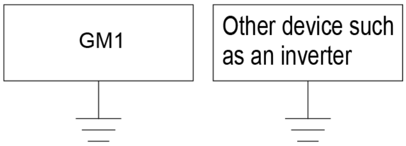
- Start the GM1 controller only after I/O devices and power devices are energized.

  In case of stopping the operation of the GM1 controller, have the I/O devices or power devices turned OFF after the GM1 controller has stopped operating.

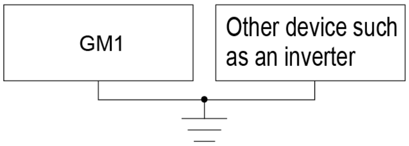
■ Grounding

- The grounding connection should have a resistance of 100 Ω or less.
- The point of grounding should be as close to the GM1 controller as possible. The ground wire should be as short as possible.
- Sharing the ground with another device may have an adverse effect. Therefore, be sure that grounding is dedicated.

○ Possible



🚫 Not possible



Conversely, depending on your environment, grounding may cause a problem. Do not ground the function earth when grounding a plus (+) terminal of the power.

■ Wiring

- Turn OFF the power supply when carry out wiring or connecting the GM1 controller to expansion units.
- Noise resistance measures such as attaching a noise filter, a surge absorber or a ferrite core may be necessary in some cases, depending on the usage environment.

■ Installation of an interlock circuit

- When controlling conflicting operations such as the motor rotation in clockwise or counter-clockwise direction, provide an interlock circuit external to the GM1 controller.

■ Installation of an emergency stop circuit

- Provide an emergency stop circuit external to the GM1 controller to turn OFF the power supply of the output device.

■ Installation environment

Do not use it in the following environments.

- Direct sunlight
- Sudden temperature changes causing condensation.
- Inflammable or corrosive gas.
- Excessive airborne dust, metal particles or saline matter.
- Benzine, paint thinner, alcohol or other organic solvents or strong alkaline solutions such as ammonia or caustic soda.
- Direct vibration, shock or direct drop of water.
- Influence from power transmission lines, high voltage equipment, power cables, power equipment, radio transmitters, or any other equipment that would generate high switching surges. (100 mm or more)

■ Handling instructions

- Before touching the unit, always touch a grounded piece of metal in order to discharge static electricity.
- Always rid yourself of any static electricity before handling this product.
- Do not connect a unit other than our GM1 series to the side connector on the unit.
- Use copper wires with a temperature rating of 90°C or higher.

Revision History

The manual code is shown at the bottom of the cover page.

Date of issue	Manual code	Revision details
August 2021	WUME-GM1ETCOP-01	First edition
February 2022	WUME-GM1ETCOP-02	2nd edition ● Updated the CNC control (G codes, instructions). ● Newly added the Recipe Manager.
April 2022	WUME-GM1ETCOP-03	3rd edition ● Changed the Company name
June 2022	WUME-GM1ETCOP-04	4th Edition ● New OPC UA server function added ● New communication with display ● New FTP server function added ● New project management function added ● Update of CNC control (G code, instruction) ● Addition of recipe manager function ● New project archive function added ● New customization function added ● Added whether to write IP address to parameter
September 2022	WUME-GM1ETCOP-05	5th Edition ● Added restrictions on the FTP server function.
August 2023	WUME-GM1ETCOP-06	6th Edition ● Newly added position lag supervision function. ● Updated cam synchronous control. ● Updated CNC control. ● Updated trace function.
November 2023	WUME-GM1ETCOP-07	7th Edition ● Added a description of version system and project conversion. ● Updated OPC UA server specifications ● Newly added MQTT client. ● Newly added DNS client ● Newly added NTP client ● Updated project management function. ● Newly added visualization function. ● Updated cam synchronous control function. ● Windows(R) 11 ：64bit support

Date of issue	Manual code	Revision details
April 2024	WUME-GM1ETCOP-08	8th Edition ● Changed some configurations ● Modified the description of the speed ramp type Sin^{2}for EtherCAT axis settings ● Updated OPC UA server specifications ● Updated MQTT client specifications ● Updated visualization function server specifications ● Updated PANATERM Lite for GM function ● Changed the Company name

(MEMO)

Please contact ....

Industrial Device Business Division,
Panasonic Industry Co., Ltd.

7-1-1 Morofuku, Daito City, Osaka, 574-0044, Japan
industrial.panasonic.com/ac/e/