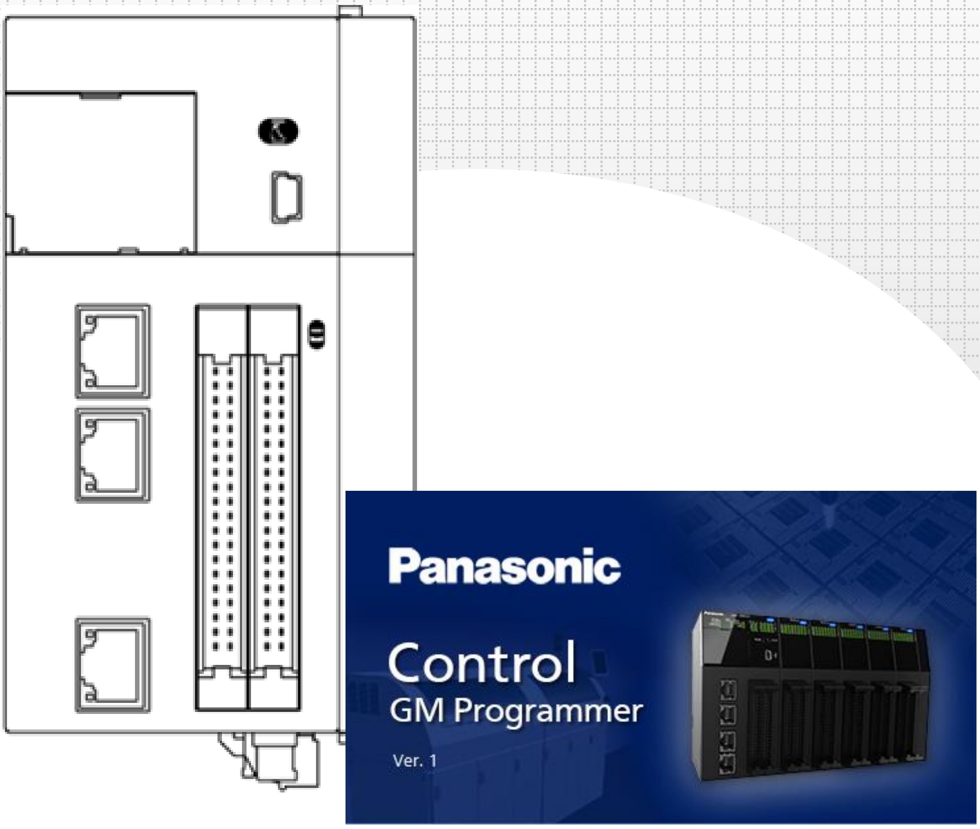
Panasonic®

Hello! GM1 Controller Communication Edition



memo

Copyright and Trademarks

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.

❗ ・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.

❗ ・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 factory use only.

GM1 Controller Communication Edition

0 Preparation

Installing tool software

Install GM Programmer from the following website:
GM Programmer: https://industrial.panasonic.com/ac/j/motor/motion-controller/mc/gm1/index.jsp

INFO

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

・GM Programmer: This is a setup tool for the GM1 controller. Using GM Programmer makes it possible to set positioning data and various positioning parameters, and perform various monitoring operations.
・PANATERM Lite for GM1 (not used in this textbook): This is 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 a PC.

Before installing GM Programmer on a PC, log on to the PC with Administrator privileges.
If other applications are running, be sure to close all the applications before installing GM Programmer.

In this textbook, one RTEX type GM1 controller and one EtherCAT type GM1 controller are used, but there is no difference in communication specifications between them.
Applicable models: AGM1CSRX16T, AGM1CSEC16T, and AGM1CSEC16P

Ethernet Communication: Modbus TCP Master/Slave

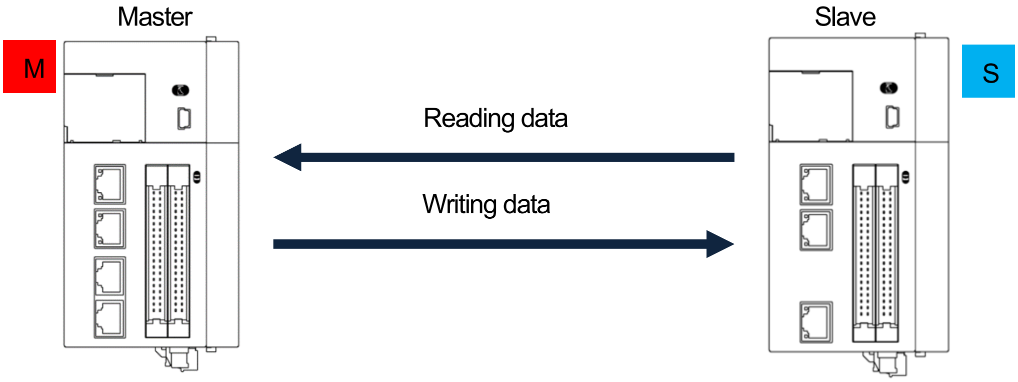
1 Basic Setup

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 methods
  ･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

In this textbook, device object settings are used to send commands to slave devices as described in 1) above.

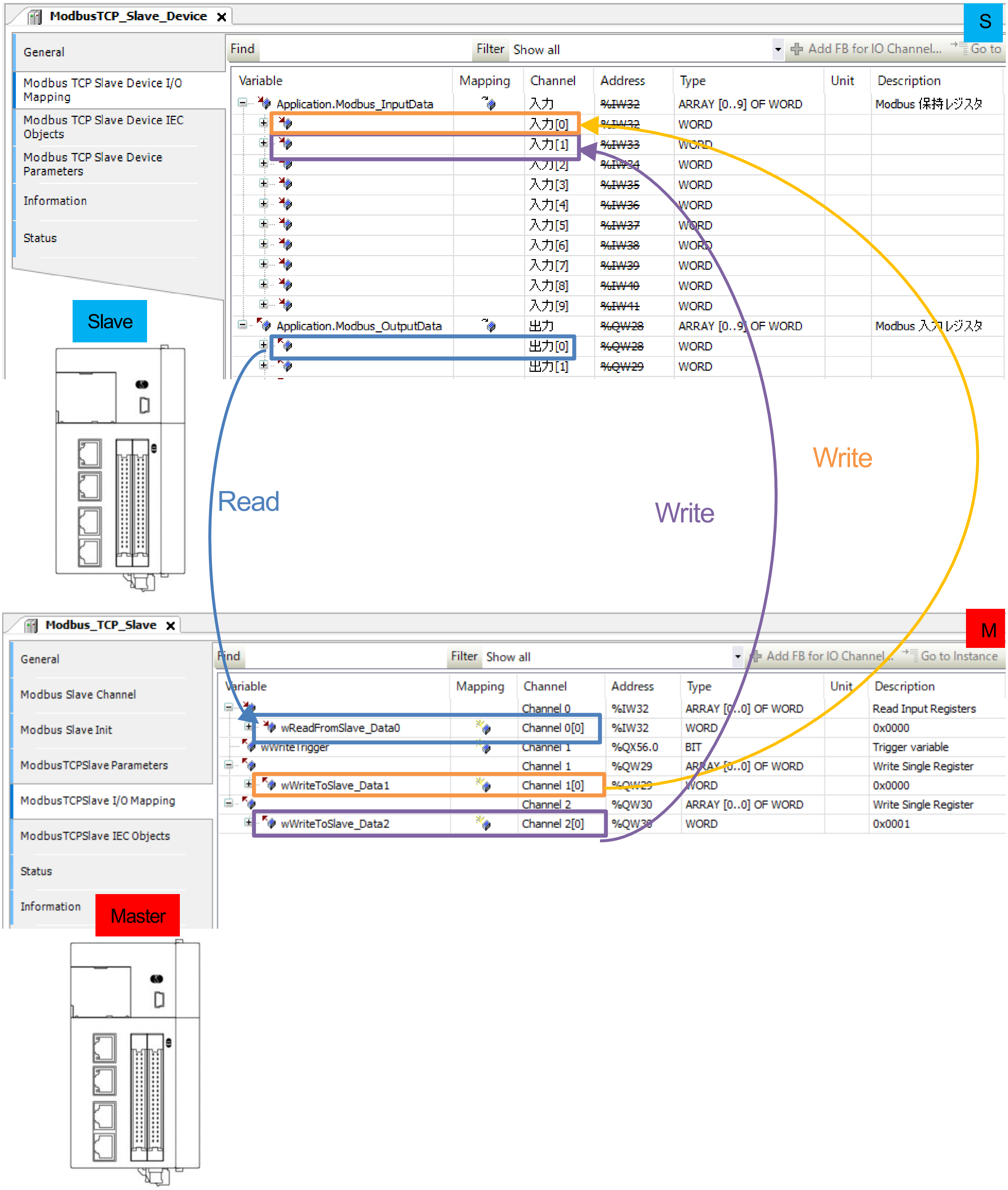
1.1 Behavior Overview

Modbus TCP
Two GM1 controllers are used as a master and slave.
LAN Port 2 on each device is used to perform Modbus TCP master communication.



* Because two copies of GM Programmer are installed on the master and slave, to identify each copy easily, M or S is shown at the top right corner of the GM Programmer window on the master or slave device, as defined below.
M : Master
S : Slave

The variables set in the **ModbusTCP_Slave_Device** window on the slave GM1 controller and in the **Modbus_TCP_Slave** window on the master GM1 controller are used to write and read data.



Variable	Mapping	Channel	Address	Type	Unit	Description
Application.Modbus_InputData		入力	~~%IW32~~	ARRAY [0..9] OF WORD		Modbus 保持レジスタ
		入力[0]	~~%IW32~~	WORD		
		入力[1]	~~%IW33~~	WORD		
		入力[2]	~~%IW34~~	WORD		
		入力[3]	~~%IW35~~	WORD		
		入力[4]	~~%IW36~~	WORD		
		入力[5]	~~%IW37~~	WORD		
		入力[6]	~~%IW38~~	WORD		
		入力[7]	~~%IW39~~	WORD		
		入力[8]	~~%IW40~~	WORD		
		入力[9]	~~%IW41~~	WORD		
Application.Modbus_OutputData		出力	~~%QW28~~	ARRAY [0..9] OF WORD		Modbus 入力レジスタ
		出力[0]	~~%QW28~~	WORD		
		出力[1]	~~%QW29~~	WORD		

Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%IW32	ARRAY [0..0] OF WORD		Read Input Registers
wReadFromSlave_Data0		Channel 0[0]	%IW32	WORD		0x0000
wWriteTrigger		Channel 1	%QX56.0	BIT		Trigger variable
		Channel 1	%QW29	ARRAY [0..0] OF WORD		Write Single Register
wWriteToSlave_Data1		Channel 1[0]	%QW29	WORD		0x0000
		Channel 2	%QW30	ARRAY [0..0] OF WORD		Write Single Register
wWriteToSlave_Data2		Channel 2[0]	%QW30	WORD		0x0001

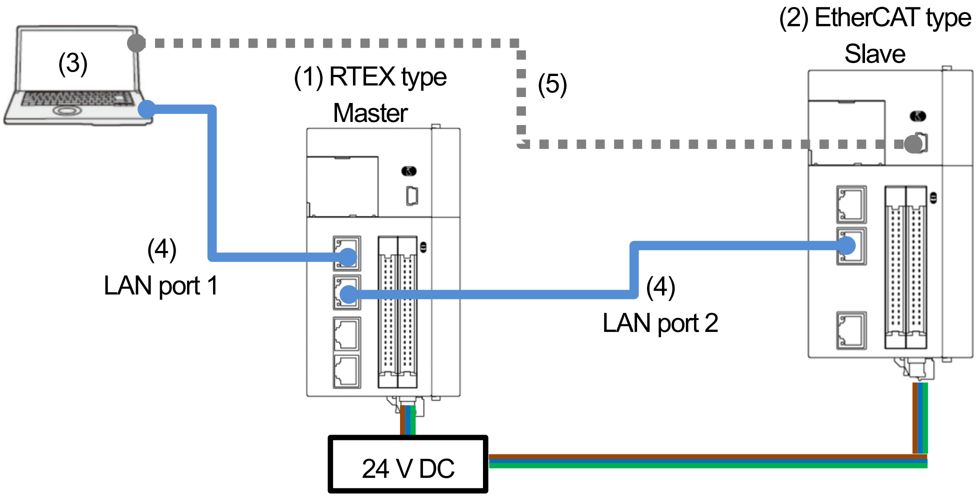
1.2 Preparing and Wiring the Required Devices

Prepare the following devices.

No.	Name	
(1)	GM1 controller (RTEX type) x 1: Master	(In this textbook, one RTEX type GM1 controller and one EtherCAT type GM1 controller are used.)
(2)	GM1 controller (EtherCAT type) x 1: Slave	
(3)	PC (with GM Programmer installed)	
(4)	LAN cable: x 2	
(5)	USB cable (Mini-B)	

* In this textbook, one RTEX type GM1 controller and one EtherCAT type GM1 controller are used, but there is no difference in communication specifications between them.

Wire each device as shown below.

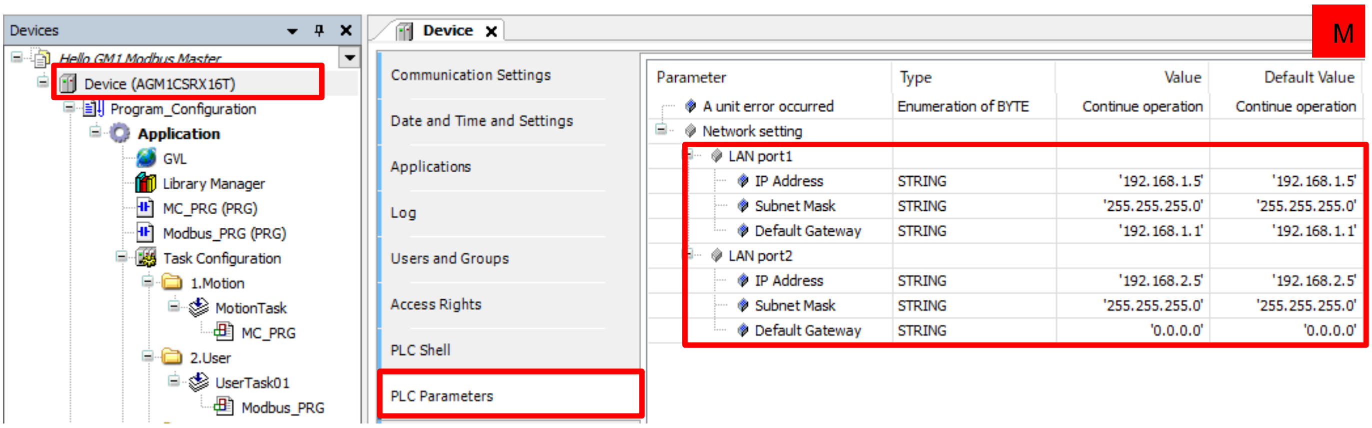


1.3 RTEX Type: Master IP Address Setting to Network Scanning

Step 1

Open GM Programmer and double-click **Device**.

Select **PLC Parameters** and check the respective IP addresses of LAN port 1 and LAN port 2.



Devices

- Hello GM1 Modbus Master
  - Device (AGM1CSRX16T)
    - Program_Configuration
      - Application
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Modbus_PRG (PRG)
        - Task Configuration
          - 1.Motion
            - MotionTask
              - MC_PRG
          - 2.User
            - UserTask01
              - Modbus_PRG

Device

- Communication Settings
- Date and Time and Settings
- Applications
- Log
- Users and Groups
- Access Rights
- PLC Shell
- PLC Parameters

Parameter	Type	Value	Default Value
A unit error occurred	Enumeration of BYTE	Continue operation	Continue operation
Network setting			
LAN port1			
IP Address	STRING	'192.168.1.5'	'192.168.1.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'
LAN port2			
IP Address	STRING	'192.168.2.5'	'192.168.2.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'

LAN port 1 (default value)

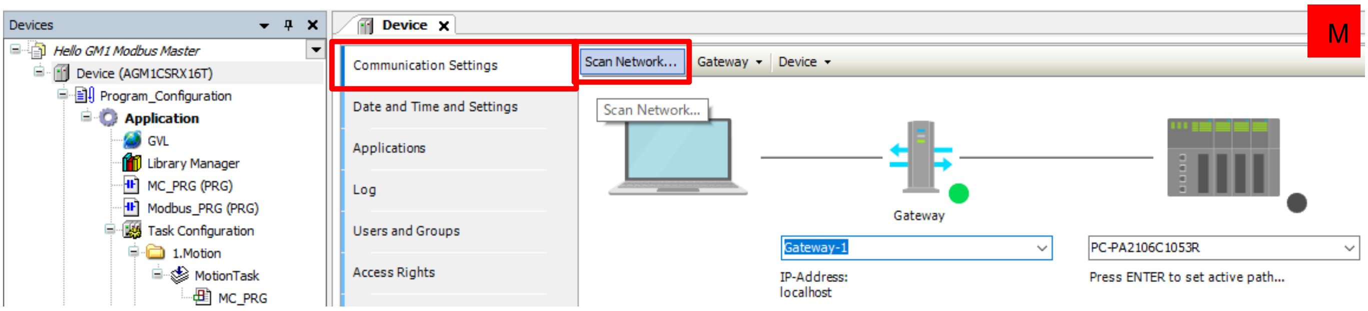
IP address	192.168.1.5
Subnet mask	255.255.255.0
Default gateway	192.168.1.1

LAN port 2 (default value)

IP address	192.168.2.5
Subnet mask	255.255.255.0
Default gateway	0.0.0.0

Step 2

Select **Communication Settings** and click **Scan Network**.



Devices

- Hello GM1 Modbus Master
  - Device (AGM1CSRX16T)
    - Program_Configuration
      - Application
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Modbus_PRG (PRG)
        - Task Configuration
          - 1.Motion
            - MotionTask
              - MC_PRG

Device

- Communication Settings
- Date and Time and Settings
- Applications
- Log
- Users and Groups
- Access Rights

Gateway-1

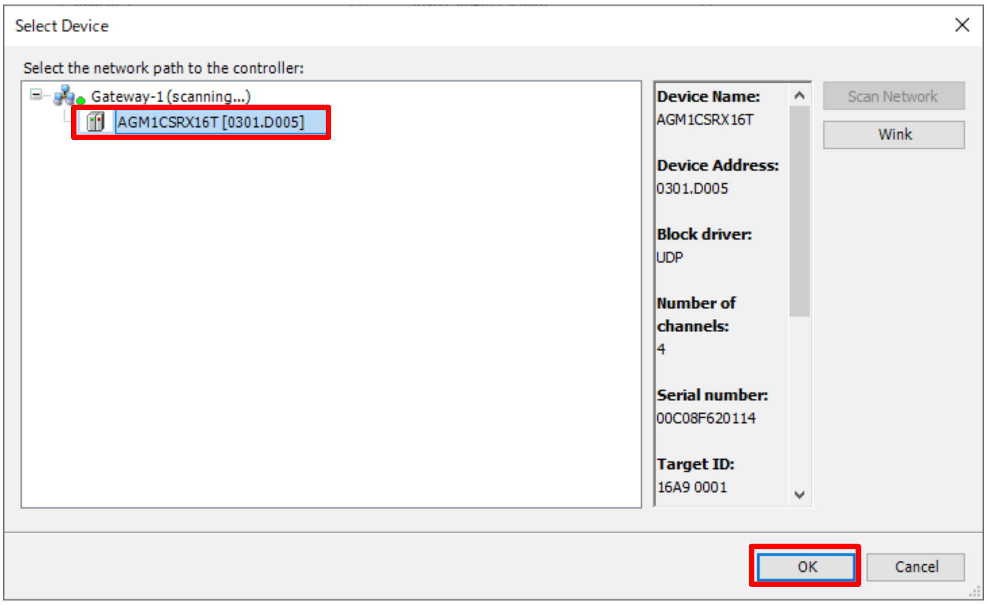
IP-Address:
localhost

PC-PA2106C1053R

Press ENTER to set active path...

Step 3

Select a device to be connected and click **OK**.



Select Device

Select the network path to the controller:

- Gateway-1 (scanning...)
  - AGM1CSRX16T [0301.D005]

Device Name:
AGM1CSRX16T

Device Address:
0301.D005

Block driver:
UDP

Number of channels:
4

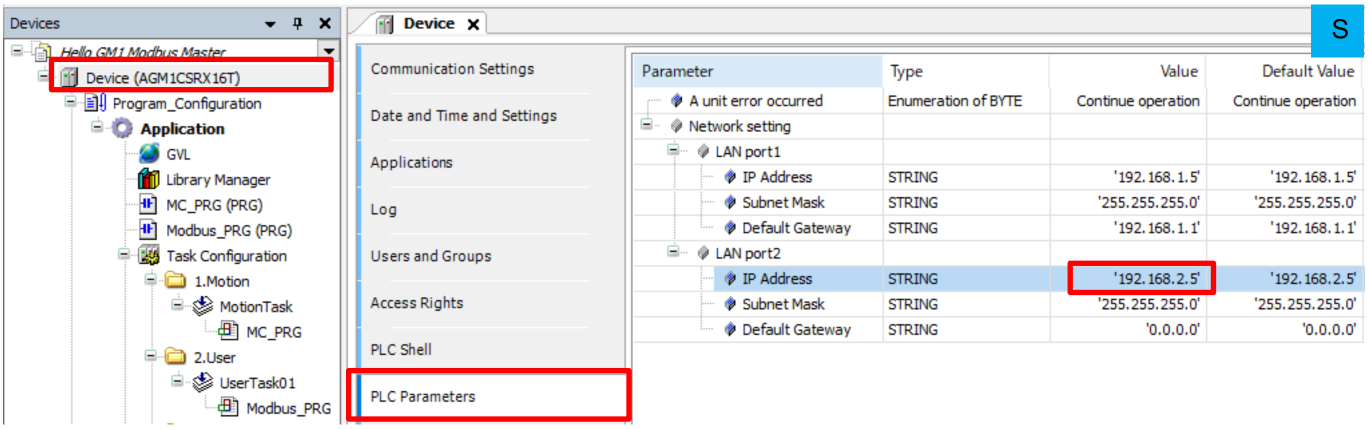
Serial number:
00C08F620114

Target ID:
16A9 0001

1.4 EtherCAT Type: Slave IP Address Setting to USB Addition

Step 1

Open GM Programmer and double-click **Device**.
Select **PLC Parameters** and change the IP address of LAN port 2 to 192.168.2.6.



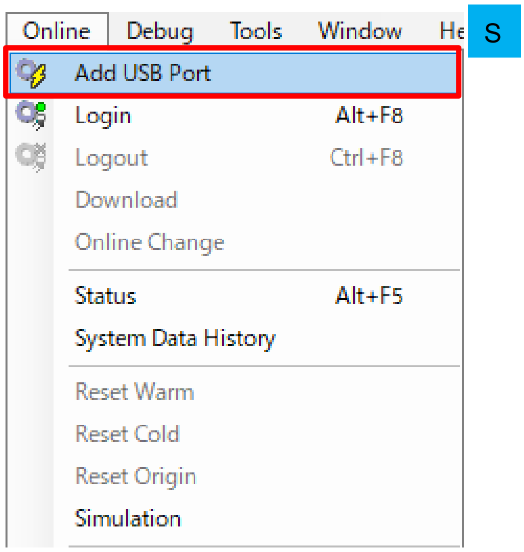
Parameter	Type	Value	Default Value
A unit error occurred	Enumeration of BYTE	Continue operation	Continue operation
Network setting			
LAN port1			
IP Address	STRING	'192.168.1.5'	'192.168.1.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'
LAN port2			
IP Address	STRING	'192.168.2.5'	'192.168.2.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'

LAN port 2

IP address	192.168.2.6
Subnet mask	255.255.255.0
Default gateway	0.0.0.0

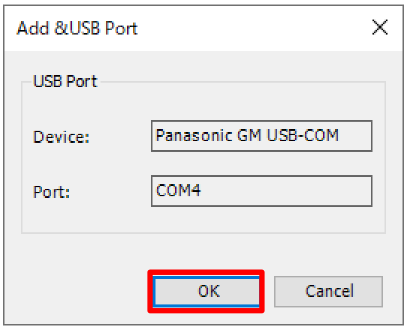
Step 2

Select **Add USB Port** from the **Online** menu.



Step 3

The **Add USB Port** dialog box will be displayed. Check the device and the port to be used and click **OK**.

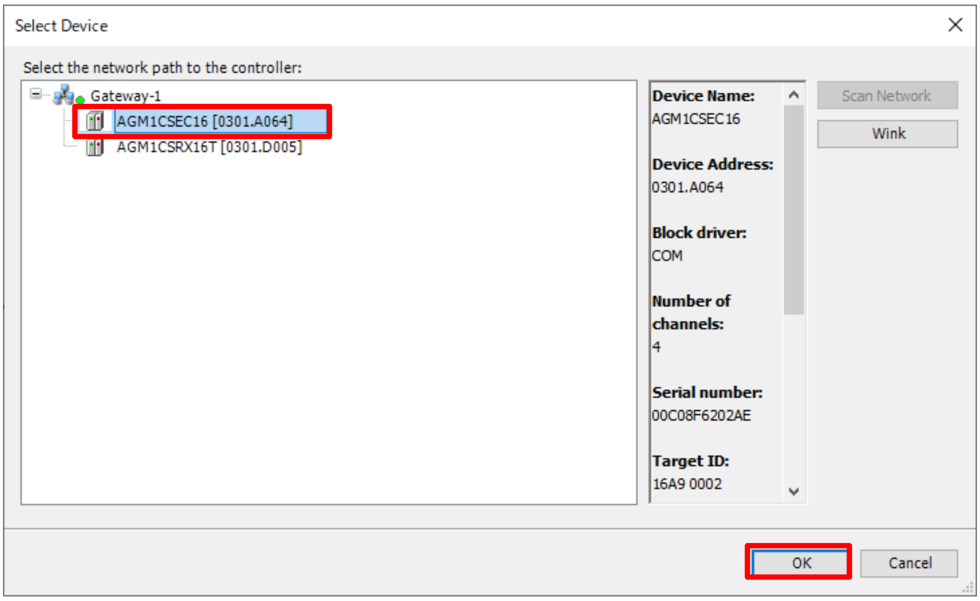


Device: Panasonic GM USB-COM

Port: COM4

Step 4

The **Select Device** dialog box will be displayed.
Select a device to be connected and click **OK**.



Select Device

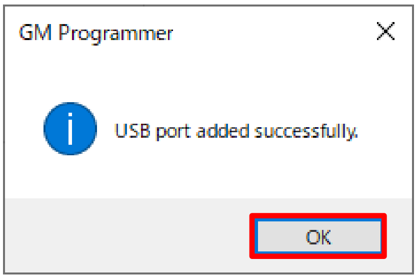
Select the network path to the controller:

Gateway-1
AGM1CSEC16 [0301.A064]
AGM1CSRX16T [0301.D005]

Device Name: AGM1CSEC16
Device Address: 0301.A064
Block driver: COM
Number of channels: 4
Serial number: 00C08F6202AE
Target ID: 16A9 0002

Step 5

When a connection is established, the USB port is added as a communication interface between the PC and GM1 controller.



GM Programmer

USB port added successfully.

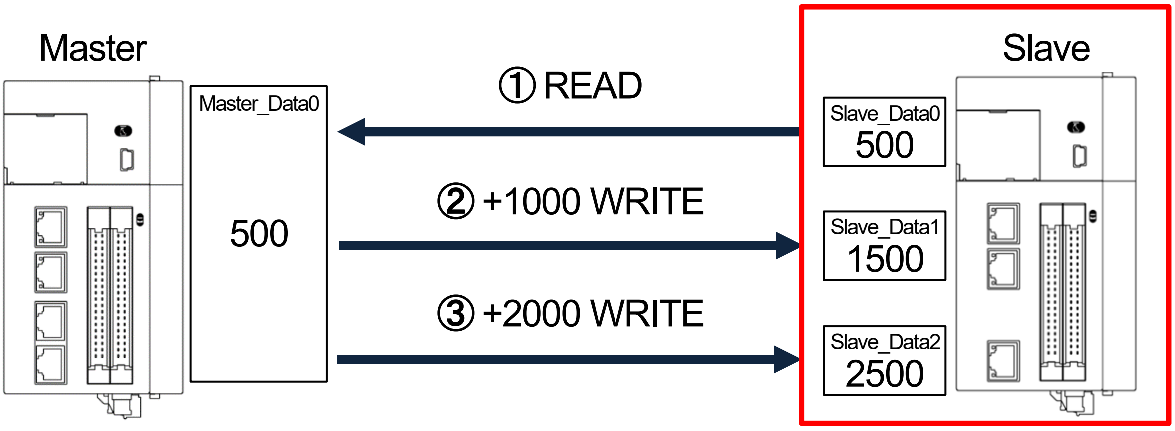
Column (1): GM1 controller-compatible function codes

Function code	Access type	Description	Address
1	Read Coils	Reads from coils	%IX
2	Read Discrete Inputs	Reads from discrete inputs	%QX
3	Read Holding Registers	Reads from holding registers	%IW
4	Read Input Registers	Reads from input registers	%QW
5	Write Single Coil	Writes to single coil	%IX
6	Write Single Register	Writes to single register	%IW
15	Write Multiple Coils	Writes to multiple coils	%IX
16	Write Multiple Registers	Writes to multiple registers	%IW
23	Read/Write Multiple Registers	Reads from or writes to multiple registers	%QW／%IW

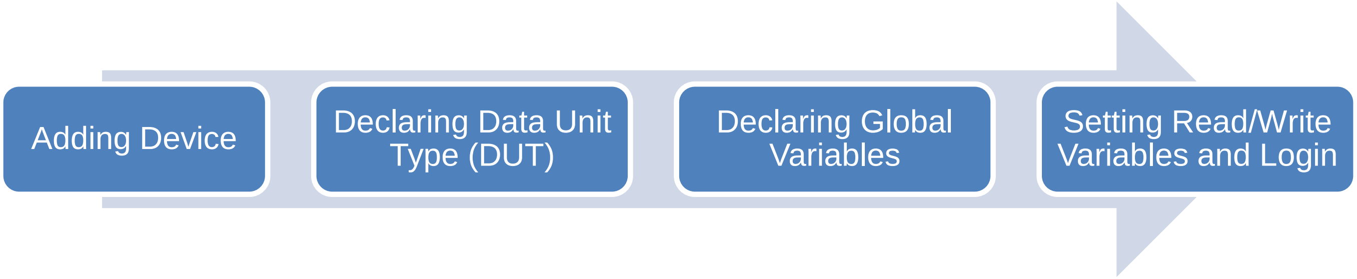
In this textbook, function code 4 and function code 6 are used.

2 Slave-side Setup

① Reads the value of Slave_Data0 from the slave. (Function code: 4)
② Writes the value plus 1,000 to Slave_Data1 on the slave (trigger: rising edge). (Function code: 6)
③ Writes the value plus 2,000 to Slave_Data2 on the slave (trigger: application). (Function code: 6)



Start setup on the slave side in the following order.



Adding Device

Declaring Data Unit Type (DUT)

Declaring Global Variables

Setting Read/Write Variables and Login

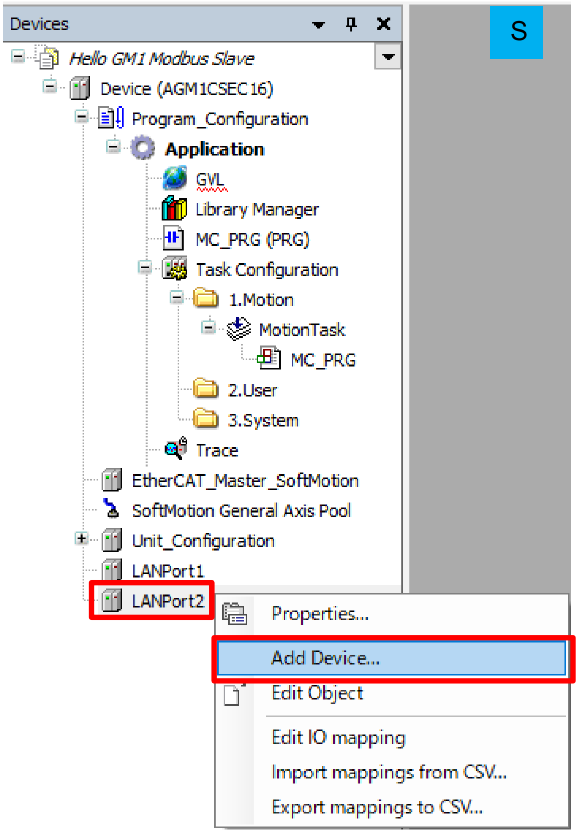
2.1 Adding Device

Add a master device which reads data from a slave unit.

Step 1

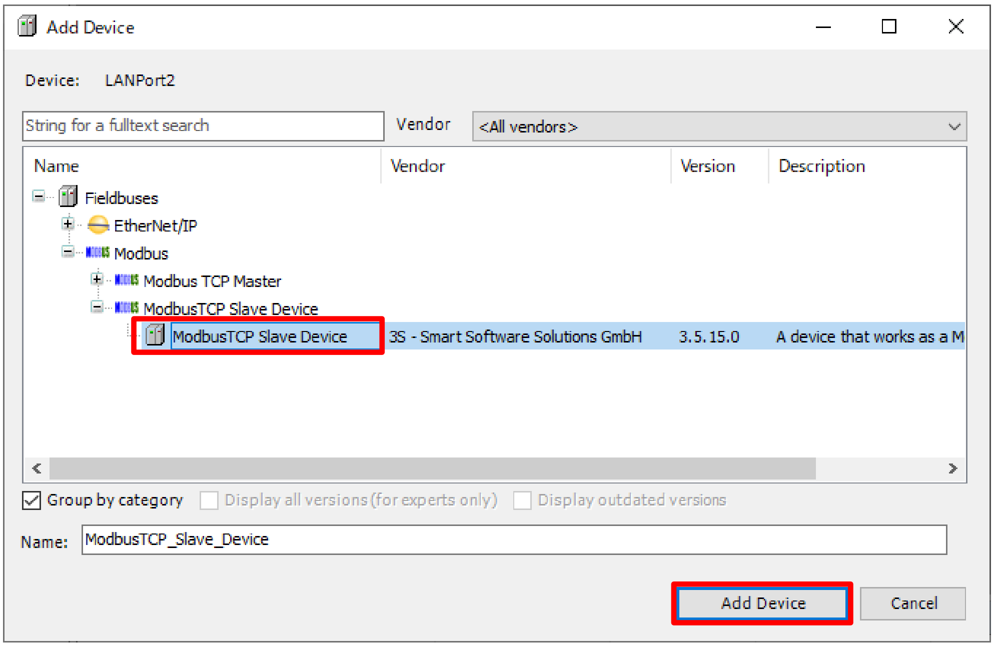
Add a Modbus TCP master device to LAN Port 2.

In the navigation pane, right-click **LANPort2** and then select **Add Device**.



Step 2

In the dialog box that is displayed, select **Modbus**, **ModbusTCP Slave Device**, and then **ModbusTCP Slave Device** and click **Add Device**.



Device: LANPort2

Name	Vendor	Version	Description
ModbusTCP Slave Device	3S - Smart Software Solutions GmbH	3.5.15.0	A device that works as a M

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

Name: ModbusTCP_Slave_Device

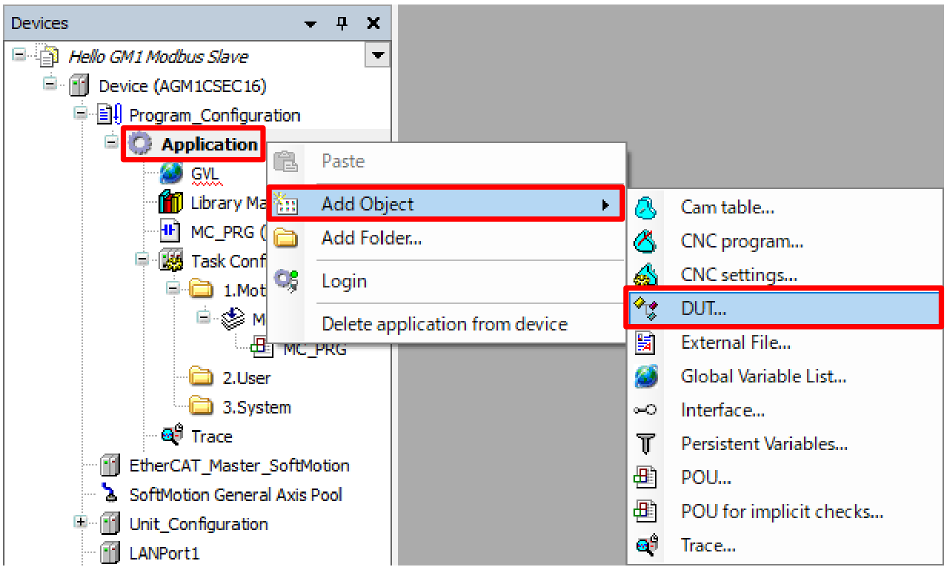
2.2 Declaring Data Unit Type (DUT)

INFO

DUT stands for Data Unit Type, which consists of multiple variables with different data types.
Define a DUT first and then use it in a global variable list or POU header in the same way as standard data types such as BOOL and INT.

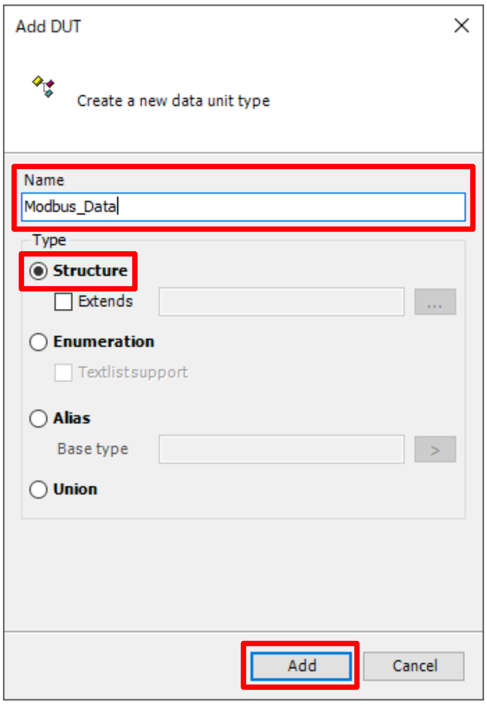
Step 1

Right-click **Application** and select **Add Object** and then **DUT**.



Step 2

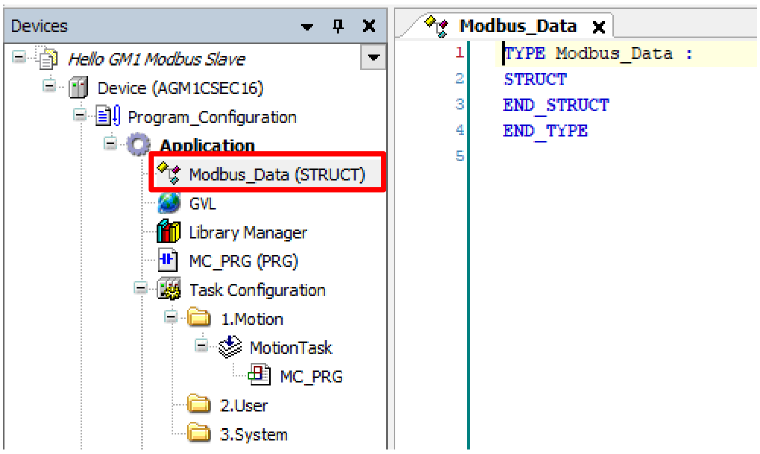
The **Add DUT** dialog box will be displayed. Assign any desired name.
Type: Select the **Structure** check box and click **Add**.



Name: Modbus_Data

Type:
● Structure
□ Extends
○ Enumeration
□ Textlistsupport
○ Alias
Base type
○ Union

Modbus_Data (STRUCT) will be added.

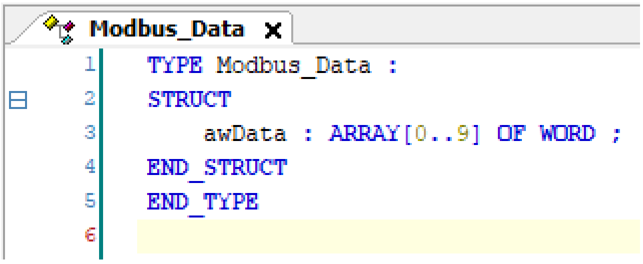


```
TYPE Modbus_Data :
STRUCT
END_STRUCT
END_TYPE
```

Step 3

Configure a DUT as shown in the figure below.

Variable name	Data type
awData	ARRAY[0..9]OF WORD

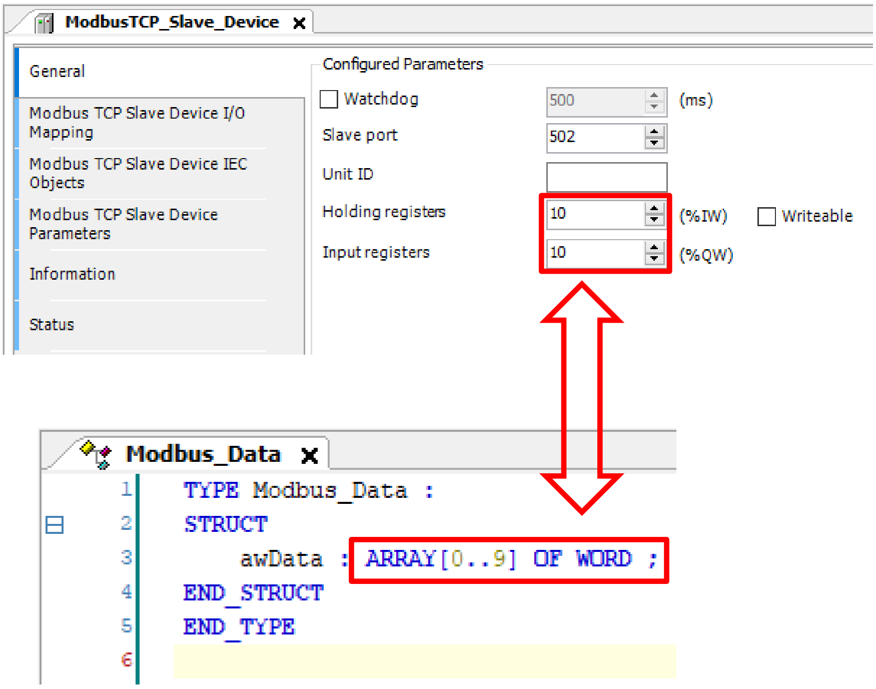


Modbus_Data

```
TYPE Modbus_Data :
STRUCT
    awData : ARRAY[0..9] OF WORD ;
END_STRUCT
END_TYPE
```

INFO

Set the respective values of **Holding registers** and **Input registers** to 10 (words) in "2.4 Setting Read/Write Variables and Login". These values must match the number of words in the DUT.



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)

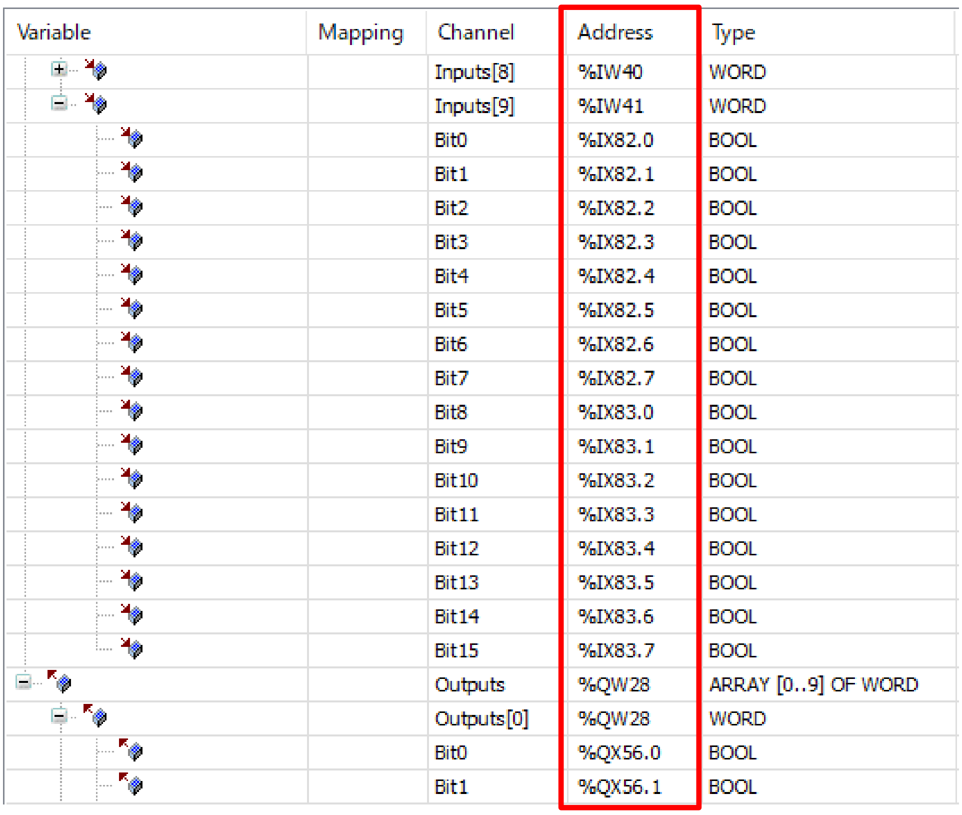
Modbus_Data

```
TYPE Modbus_Data :
STRUCT
    awData : ARRAY[0..9] OF WORD ;
END_STRUCT
END_TYPE
```

This completes the declaration of DUT.

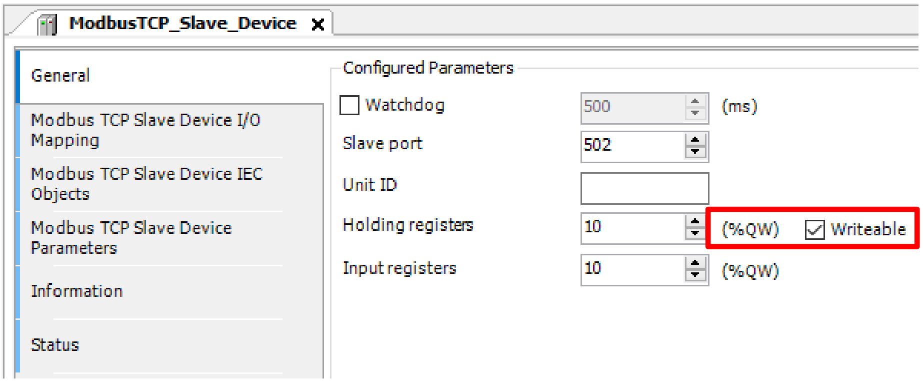
Column (2): Relationship between register and address

Register	Address
Holding register (WORD)	%IW
Holding register (BOOL)	%IX
Input register (WORD)	%QW
Input register (BOOL)	%QX



Variable	Mapping	Channel	Address	Type
		Inputs[8]	%IW40	WORD
		Inputs[9]	%IW41	WORD
		Bit0	%IX82.0	BOOL
		Bit1	%IX82.1	BOOL
		Bit2	%IX82.2	BOOL
		Bit3	%IX82.3	BOOL
		Bit4	%IX82.4	BOOL
		Bit5	%IX82.5	BOOL
		Bit6	%IX82.6	BOOL
		Bit7	%IX82.7	BOOL
		Bit8	%IX83.0	BOOL
		Bit9	%IX83.1	BOOL
		Bit10	%IX83.2	BOOL
		Bit11	%IX83.3	BOOL
		Bit12	%IX83.4	BOOL
		Bit13	%IX83.5	BOOL
		Bit14	%IX83.6	BOOL
		Bit15	%IX83.7	BOOL
		Outputs	%QW28	ARRAY [0..9] OF WORD
		Outputs[0]	%QW28	WORD
		Bit0	%QX56.0	BOOL
		Bit1	%QX56.1	BOOL

If you select the **Writeable** check box in the **General** tab of the **ModbusTCP_Slave_Device** window, the registers will be write-enabled and the address will be changed to "%QW".



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	(%QW) ☑ Writeable
Input registers	10	(%QW)

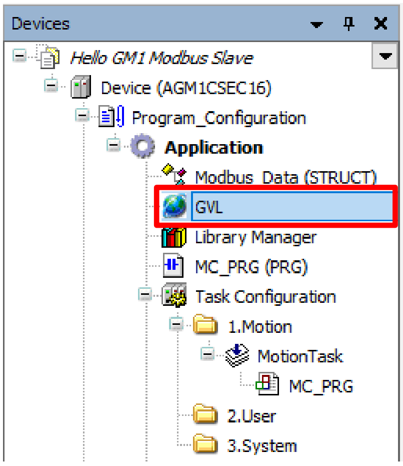
2.3 Declaring Global Variables

INFO

Global variables can be used throughout the entire project.

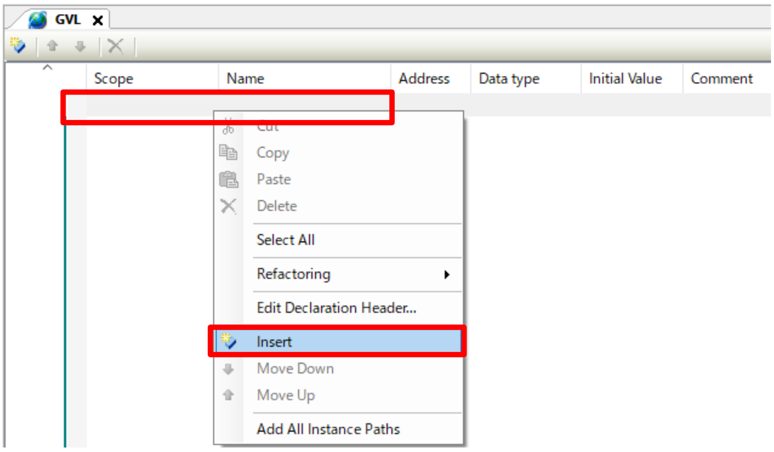
Step 1

Double-click **GVL**.



Step 2

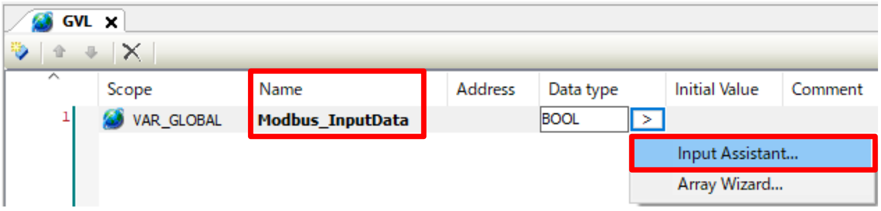
Right-click in the gray area and select **Insert**.



Step 3

Enter "Modbus_InputData" in the **Name** column.

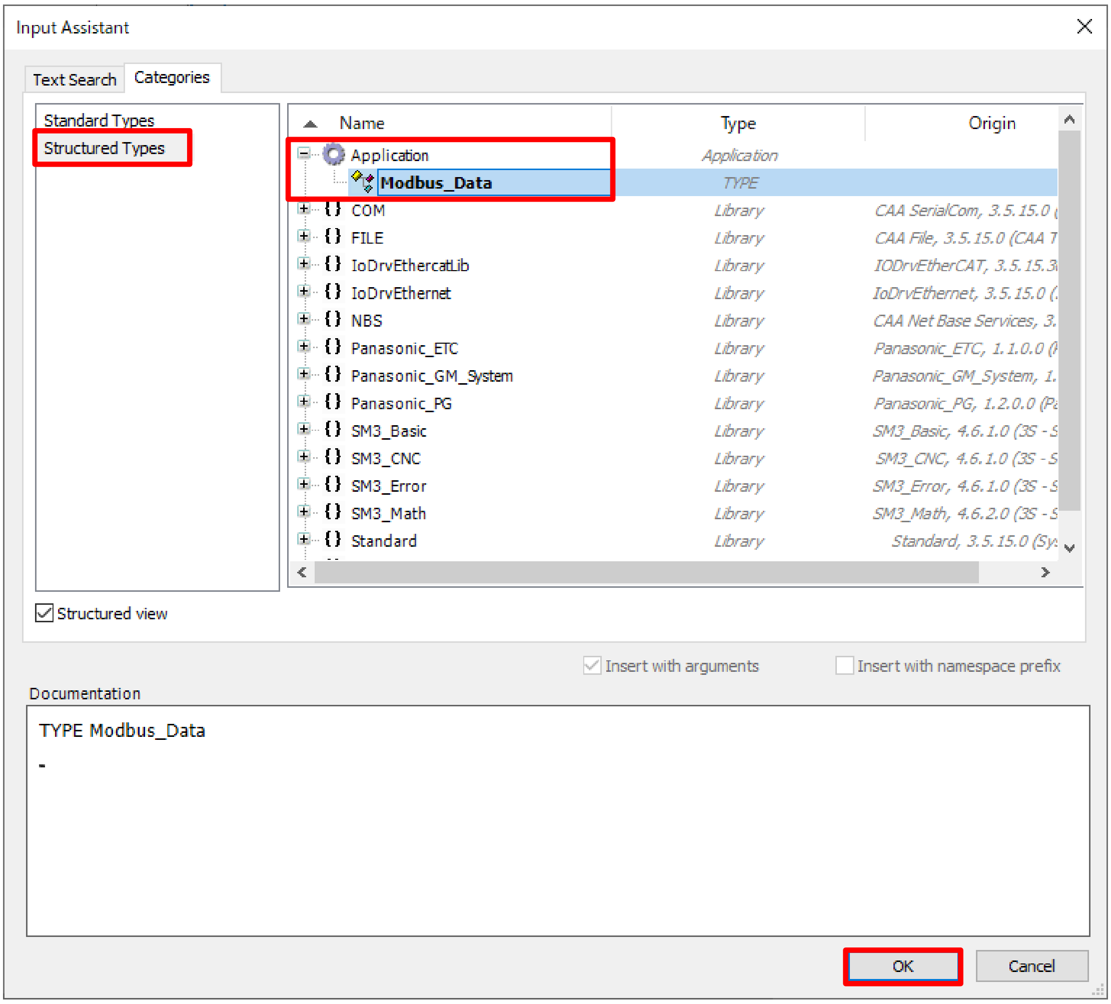
Double-click in the **Data type** column and select **Input Assistant**.



Scope	Name	Address	Data type	Initial Value	Comment
VAR_GLOBAL	Modbus_InputData		BOOL		

Step 4

The **Input Assistant** dialog box will be displayed. Select **Structured Types**, **Application**, and then **Modbus_Data** and click **OK**.



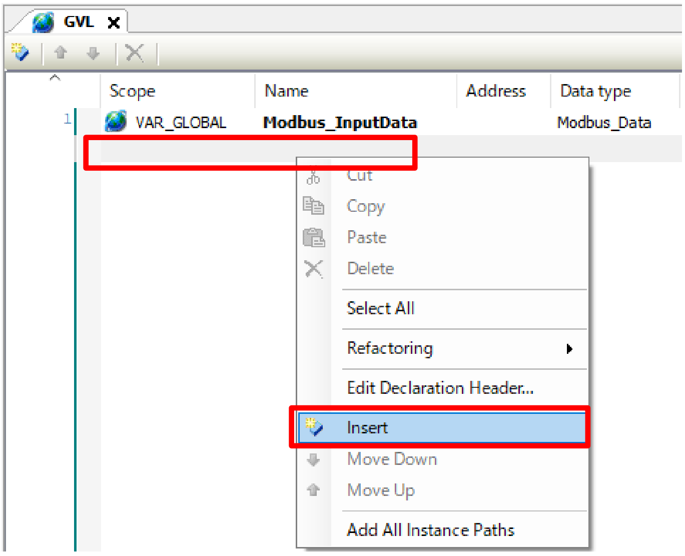
Name	Type	Origin
Application	Application	
Modbus_Data	TYPE	
COM	Library	CAA SerialCom, 3.5.15.0 (
FILE	Library	CAA File, 3.5.15.0 (CAA T
IoDrvEthercatLib	Library	IODrvEtherCAT, 3.5.15.3
IoDrvEthernet	Library	IoDrvEthernet, 3.5.15.0 (
NBS	Library	CAA Net Base Services, 3.
Panasonic_ETC	Library	Panasonic_ETC, 1.1.0.0 (
Panasonic_GM_System	Library	Panasonic_GM_System, 1.
Panasonic_PG	Library	Panasonic_PG, 1.2.0.0 (P
SM3_Basic	Library	SM3_Basic, 4.6.1.0 (3S - S
SM3_CNC	Library	SM3_CNC, 4.6.1.0 (3S - S
SM3_Error	Library	SM3_Error, 4.6.1.0 (3S - S
SM3_Math	Library	SM3_Math, 4.6.2.0 (3S - S
Standard	Library	Standard, 3.5.15.0 (Sy

TYPE Modbus_Data

-

Step 5

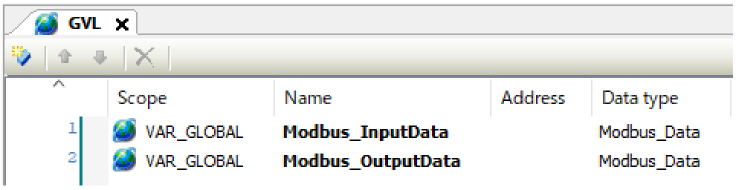
Right-click in the area below **Modbus_InputData**, which has been inserted, and select **Insert**.



Scope	Name	Address	Data type
VAR_GLOBAL	Modbus_InputData		Modbus_Data

Step 6

Enter "Modbus_OutputData" in the **Name** column.



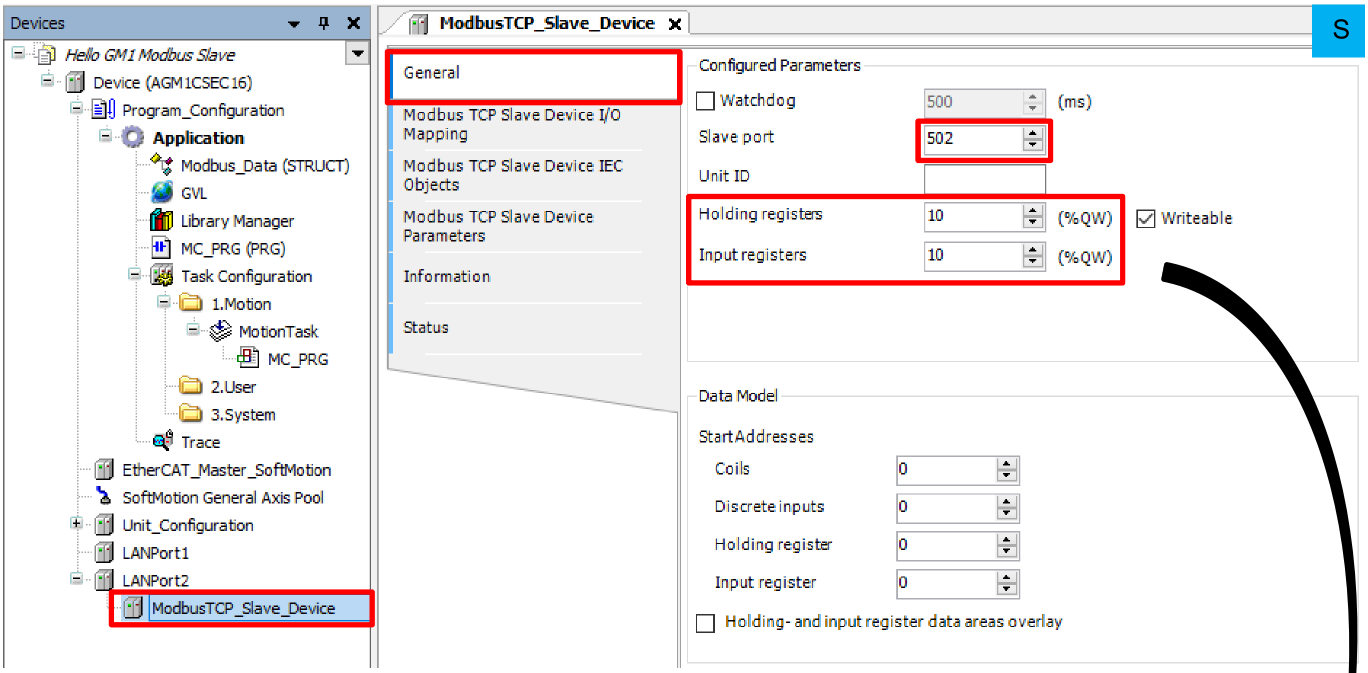
	Scope	Name	Address	Data type
1	VAR_GLOBAL	**Modbus_InputData**		Modbus_Data
2	VAR_GLOBAL	**Modbus_OutputData**		Modbus_Data

This completes the declaration of global variables.

2.4 Setting Read/Write Variables and Login

Step 1

Double-click **ModbusTCP_Slave_Device**, which has been added, and select the **General** tab.
Set **Slave port** to 502 (default).



Devices

- Hello GM1 Modbus Slave
  - Device (AGM1CSEC16)
    - Program_Configuration
      - Application
        - Modbus_Data (STRUCT)
        - 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
    - LANPort1
    - LANPort2
      - ModbusTCP_Slave_Device

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 (%QW) ☑ Writeable
- Input registers: 10 (%QW)

Data Model

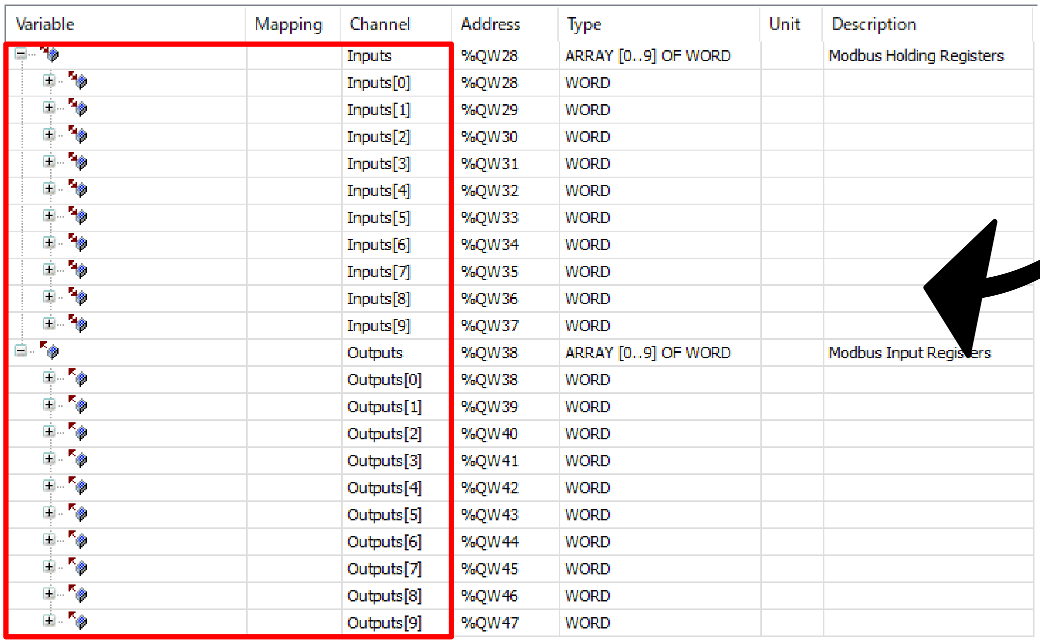
Start Addresses

- Coils: 0
- Discrete inputs: 0
- Holding register: 0
- Input register: 0

□ Holding- and input register data areas overlay

INFO

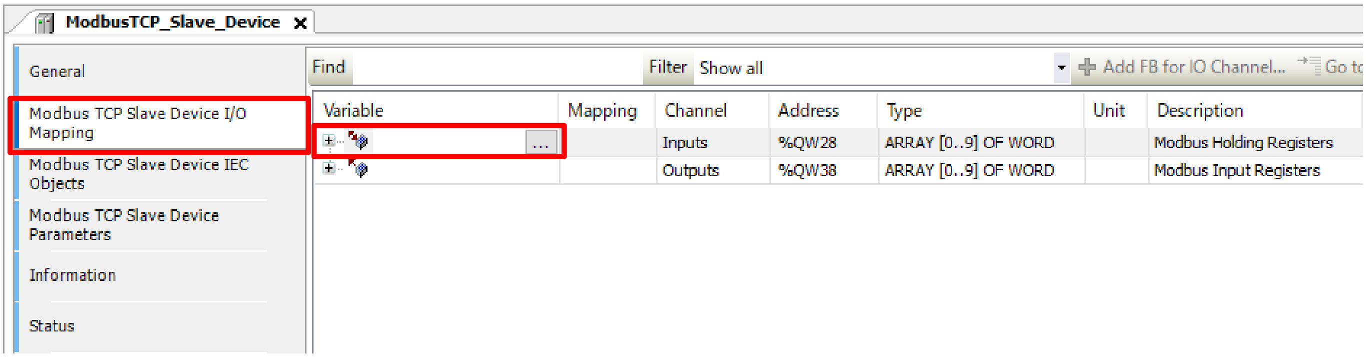
The respective values of **Holding registers** and **Input registers** in the **General** tab of the **ModbusTCP_Slave_Device** pane are set to "10", as shown above.
Therefore, 10 holding registers from Inputs[0] to Inputs[9] and 10 input registers from Outputs[0] to Outputs[9] in the figure below are reserved as an area.



Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%QW28	ARRAY [0..9] OF WORD		Modbus Holding Registers
		Inputs[0]	%QW28	WORD		
		Inputs[1]	%QW29	WORD		
		Inputs[2]	%QW30	WORD		
		Inputs[3]	%QW31	WORD		
		Inputs[4]	%QW32	WORD		
		Inputs[5]	%QW33	WORD		
		Inputs[6]	%QW34	WORD		
		Inputs[7]	%QW35	WORD		
		Inputs[8]	%QW36	WORD		
		Inputs[9]	%QW37	WORD		
		Outputs	%QW38	ARRAY [0..9] OF WORD		Modbus Input Registers
		Outputs[0]	%QW38	WORD		
		Outputs[1]	%QW39	WORD		
		Outputs[2]	%QW40	WORD		
		Outputs[3]	%QW41	WORD		
		Outputs[4]	%QW42	WORD		
		Outputs[5]	%QW43	WORD		
		Outputs[6]	%QW44	WORD		
		Outputs[7]	%QW45	WORD		
		Outputs[8]	%QW46	WORD		
		Outputs[9]	%QW47	WORD		

Step 2

Select **ModbusTCP_Slave_Device I/O Mapping**, double-click the inside of the red frame in the **Variable** column, and click ... .



ModbusTCP_Slave_Device

General

Modbus TCP Slave Device I/O Mapping

Modbus TCP Slave Device IEC Objects

Modbus TCP Slave Device Parameters

Information

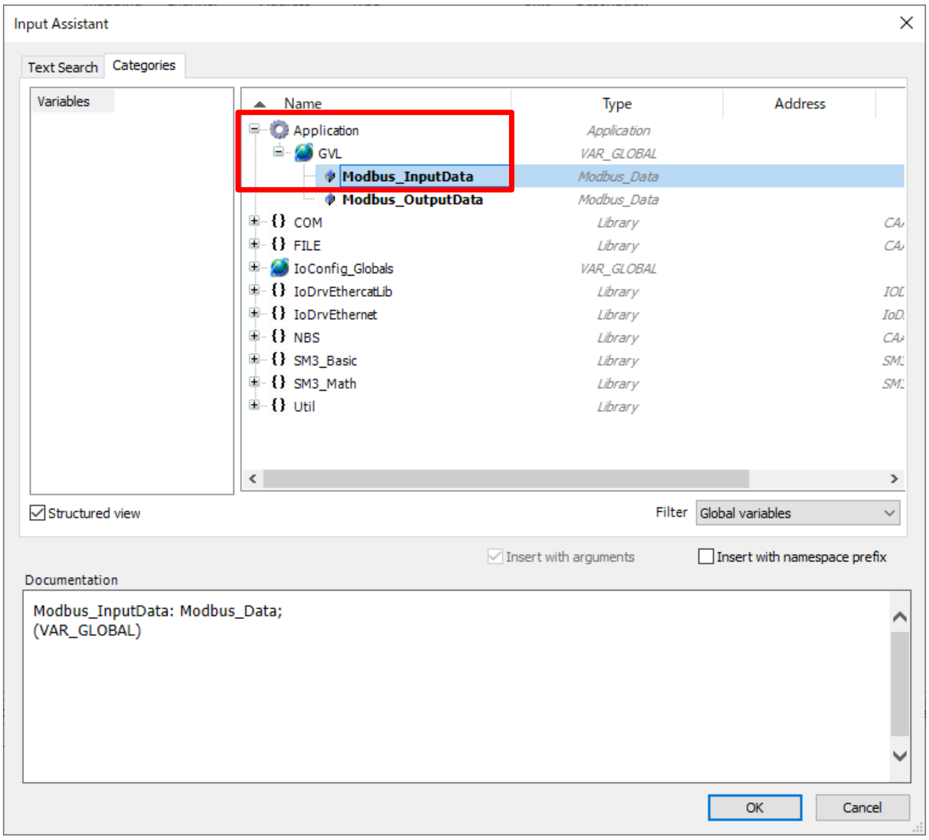
Status

Find | Filter Show all | Add FB for IO Channel... | Go to

Variable	Mapping	Channel	Address	Type	Unit	Description
...		Inputs	%QW28	ARRAY [0..9] OF WORD		Modbus Holding Registers
		Outputs	%QW38	ARRAY [0..9] OF WORD		Modbus Input Registers

Step 3

The **Input Assistant** dialog box will be displayed. Select **Application**, **GVL**, and then **Modbus_InputData** and click **OK**.



Input Assistant

Text Search | Categories

Variables

Name	Type	Address
Application	Application	
GVL	VAR_GLOBAL	
Modbus_InputData	Modbus_Data	
Modbus_OutputData	Modbus_Data	
COM	Library	CA
FILE	Library	CA
IoConfig_Globals	VAR_GLOBAL	
IoDrvEthercatLib	Library	IOL
IoDrvEthernet	Library	IoD
NBS	Library	CA
SM3_Basic	Library	SM
SM3_Math	Library	SM
Util	Library	

☑ Structured view

Filter Global variables

☑ Insert with arguments ☐ Insert with namespace prefix

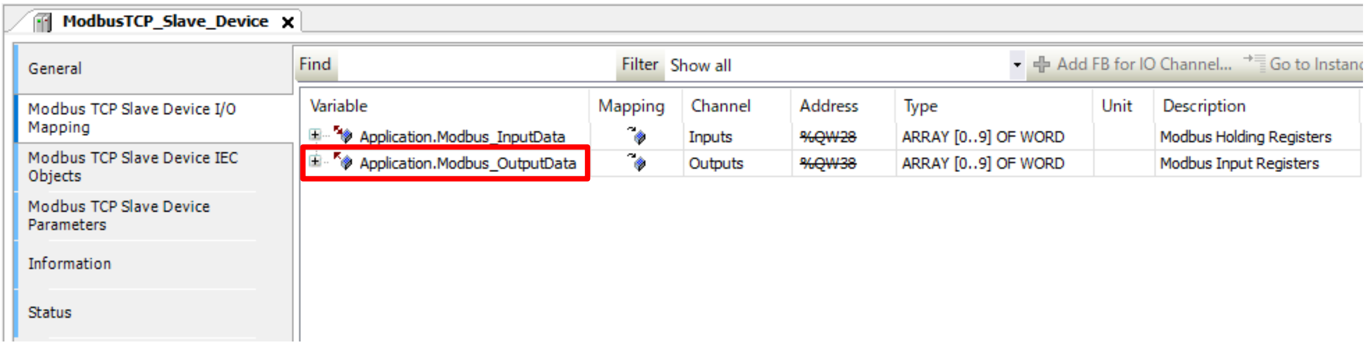
Documentation

Modbus_InputData: Modbus_Data;
(VAR_GLOBAL)

OK | Cancel

Step 4

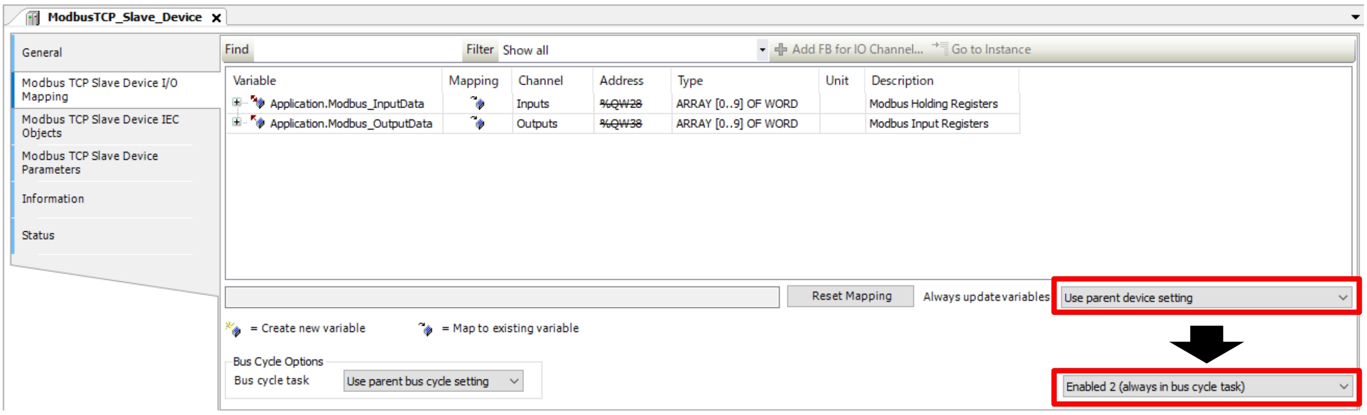
Select **Modbus_OutputData** inside the red frame in the same way as Steps 2 and 3.


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 Instanc

Variable	Mapping	Channel	Address	Type	Unit	Description
Application.Modbus_InputData		Inputs	~~%QW28~~	ARRAY [0..9] OF WORD		Modbus Holding Registers
Application.Modbus_OutputData		Outputs	~~%QW38~~	ARRAY [0..9] OF WORD		Modbus Input Registers

Step 5

Select **Use parent device setting** from the drop-down list on the right side of **Always update variables** and then select **Enabled 2 (always in bus cycle task)** from the drop-down list below the above one.



Variable	Mapping	Channel	Address	Type	Unit	Description
Application.Modbus_InputData		Inputs	~~%QW28~~	ARRAY [0..9] OF WORD		Modbus Holding Registers
Application.Modbus_OutputData		Outputs	~~%QW38~~	ARRAY [0..9] OF WORD		Modbus Input Registers

Always update variables: Use parent device setting

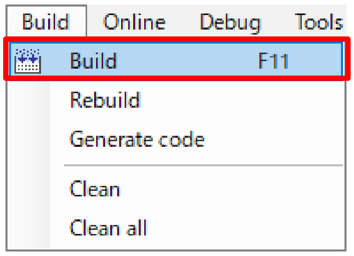
Enabled 2 (always in bus cycle task)

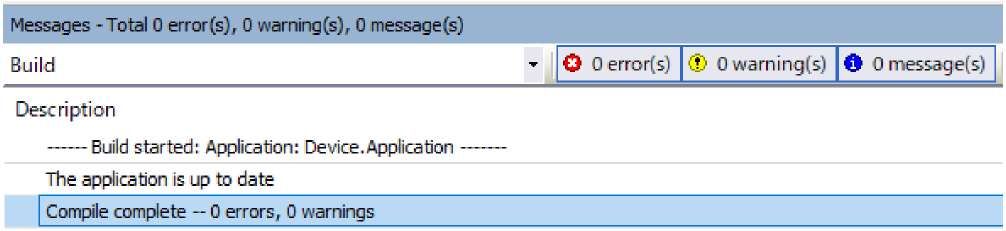
Bus Cycle Options

Bus cycle task: Use parent bus cycle setting

Step 6

Select **Build** from the **Build** menu and check that no error occurs.





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

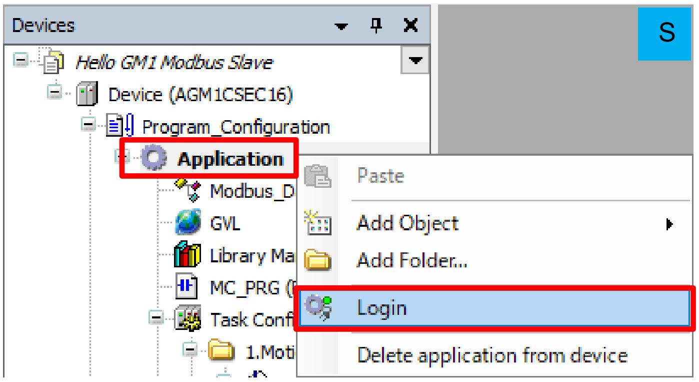
------ Build started: Application: Device.Application -------

The application is up to date

Compile complete -- 0 errors, 0 warnings

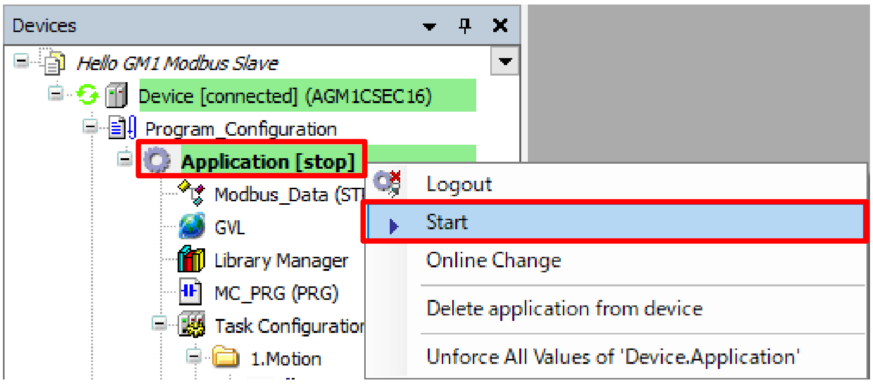
Step 7

Right-click **Application** and then select **Login** to write to the GM1 controller.



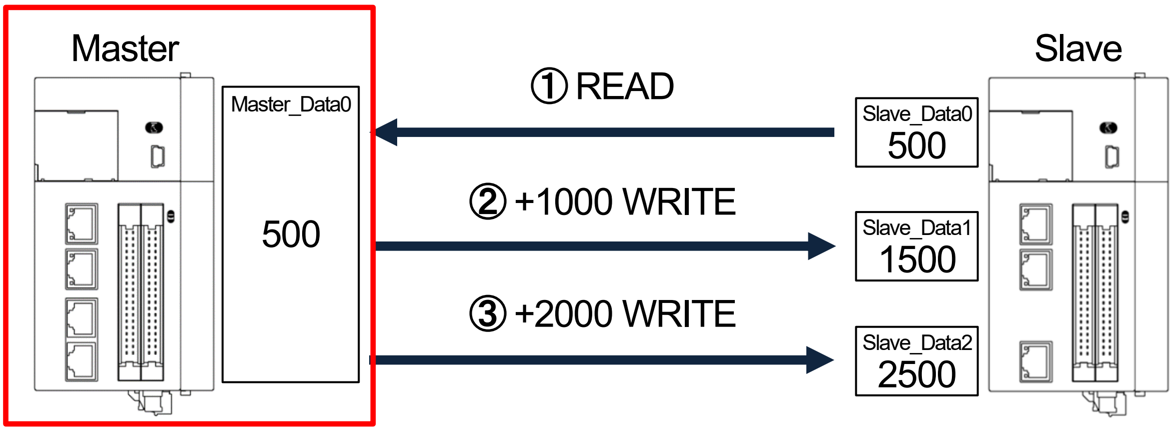
Step 8

Right-click **Application [stop]** and then select **Start** to switch the state from STOP to RUN.

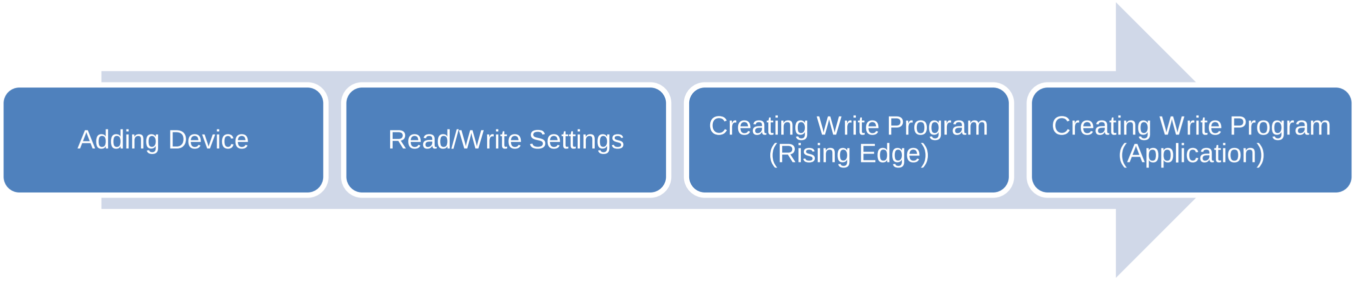


え

3 Master-side Setup to Programming



Start setup on the master side in the following order.



Adding Device

Read/Write Settings

Creating Write Program (Rising Edge)

Creating Write Program (Application)

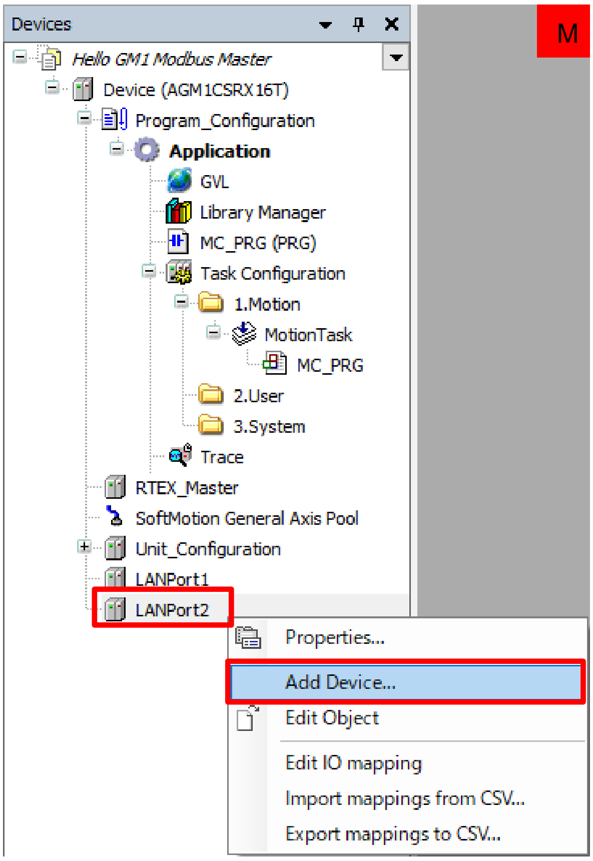
3.1 Adding Device

Add a slave device from which data is to be read.

Step 1

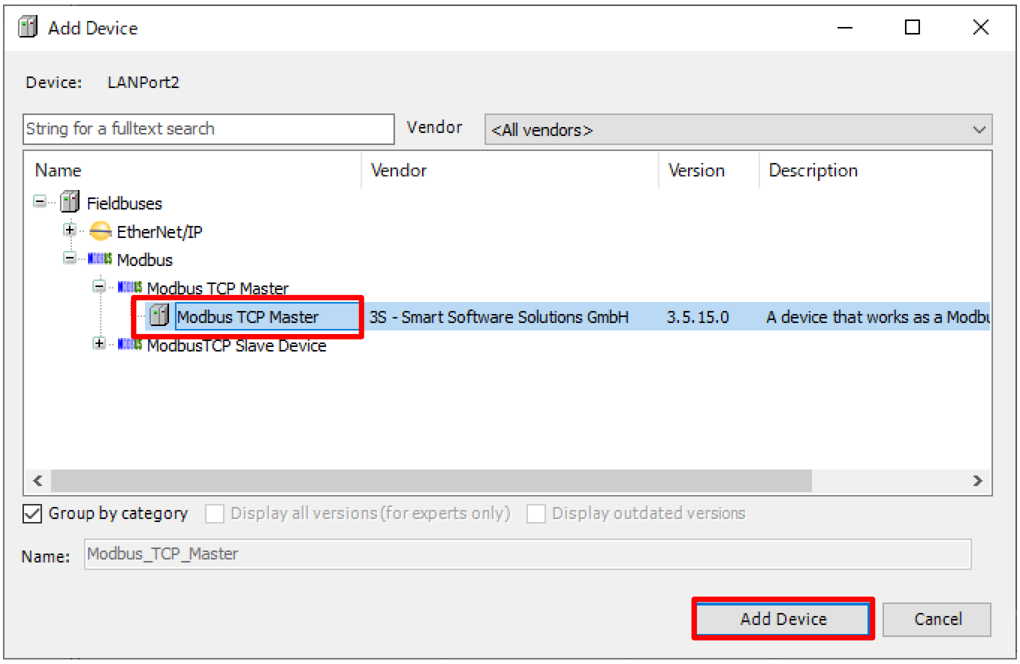
Add a Modbus TCP master device to LAN Port 2.

In the navigation pane, right-click **LANPort2** and then select **Add Device**.



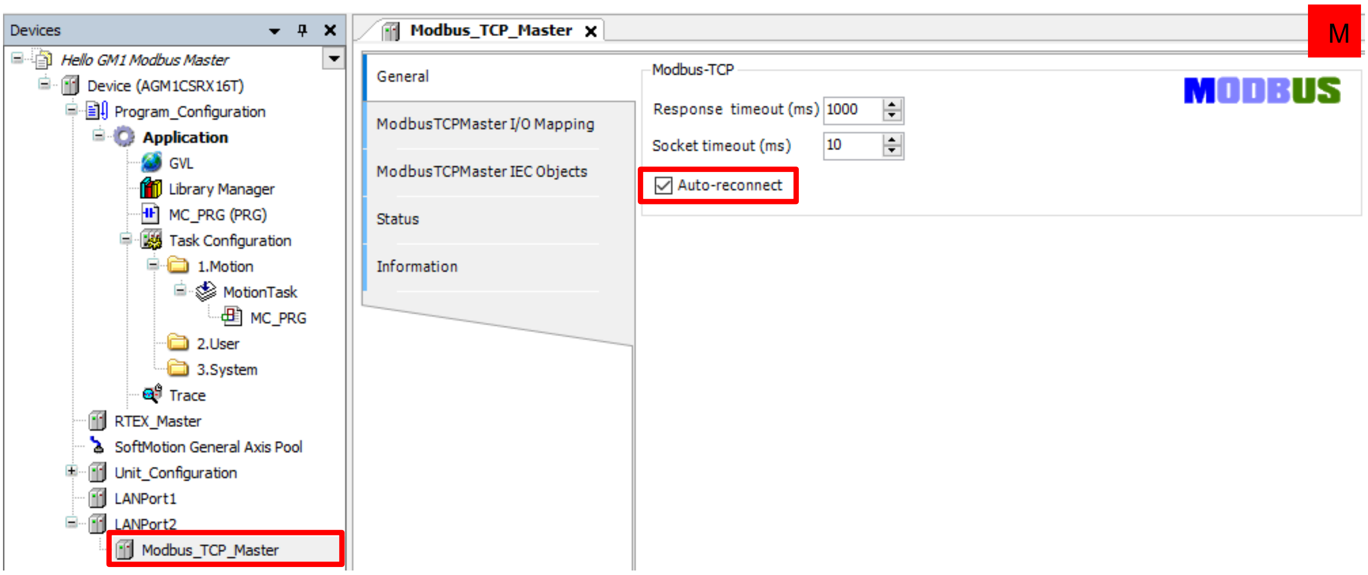
Step 2

In the dialog box that is displayed, select **Modbus**, **Modbus TCP Master**, and then **Modbus TCP Master** and click **Add Device**.



Step 3

Double-click **Modbus_TCP_Master**, which has been added, and select **General**.
In the **General** tab, select the **Auto-reconnect** check box.



Modbus-TCP

Response timeout (ms) 1000

Socket timeout (ms) 10

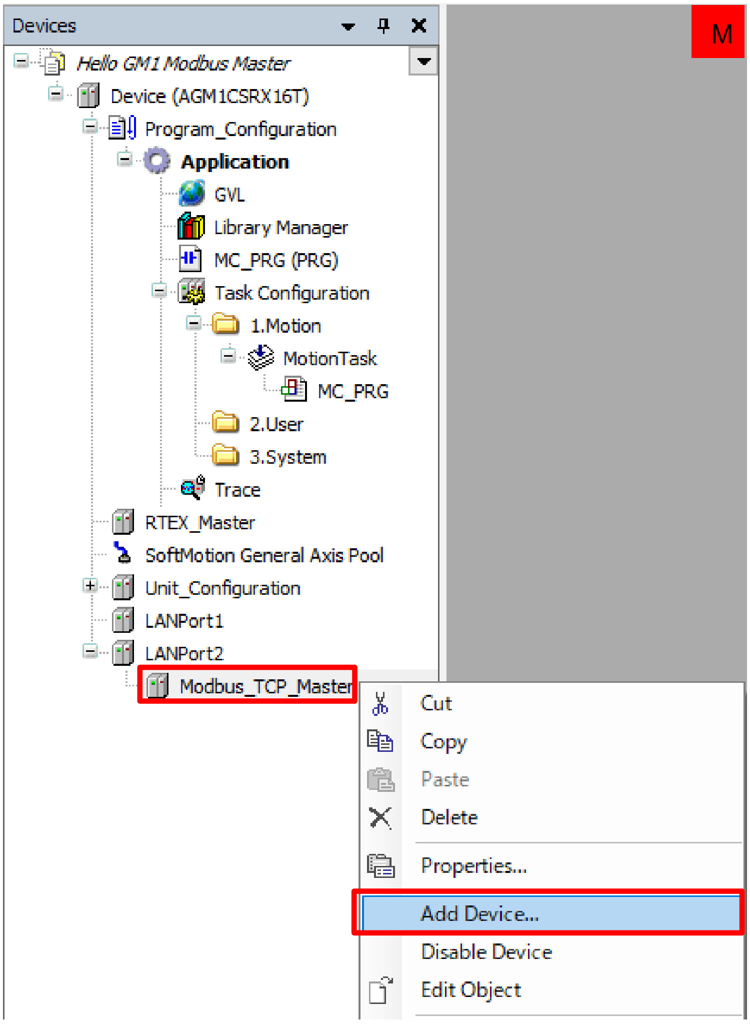
☑ Auto-reconnect

INFO

When the GM1 controller is used as a master, it attempts to open a connection. If the **Auto-reconnect** check box is not selected, the GM1 controller will not attempt a retry when a connection fails due to timeout or for some other reason.

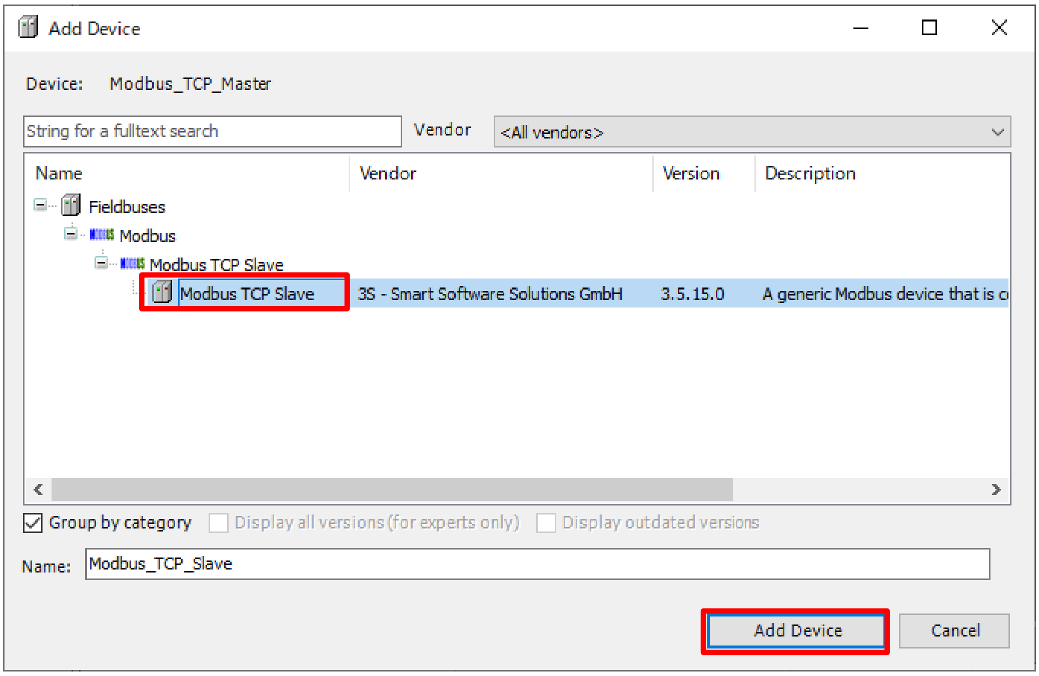
Step 4

Right-click **Modbus_TCP_Master** and select **Add Device**.



Step 5

In the dialog box that is displayed, select **Modbus TCP Slave** and then **Modbus TCP Slave** and click **Add Device**.



Device: Modbus_TCP_Master

Name	Vendor	Version	Description
Modbus TCP Slave	3S - Smart Software Solutions GmbH	3.5.15.0	A generic Modbus device that is c

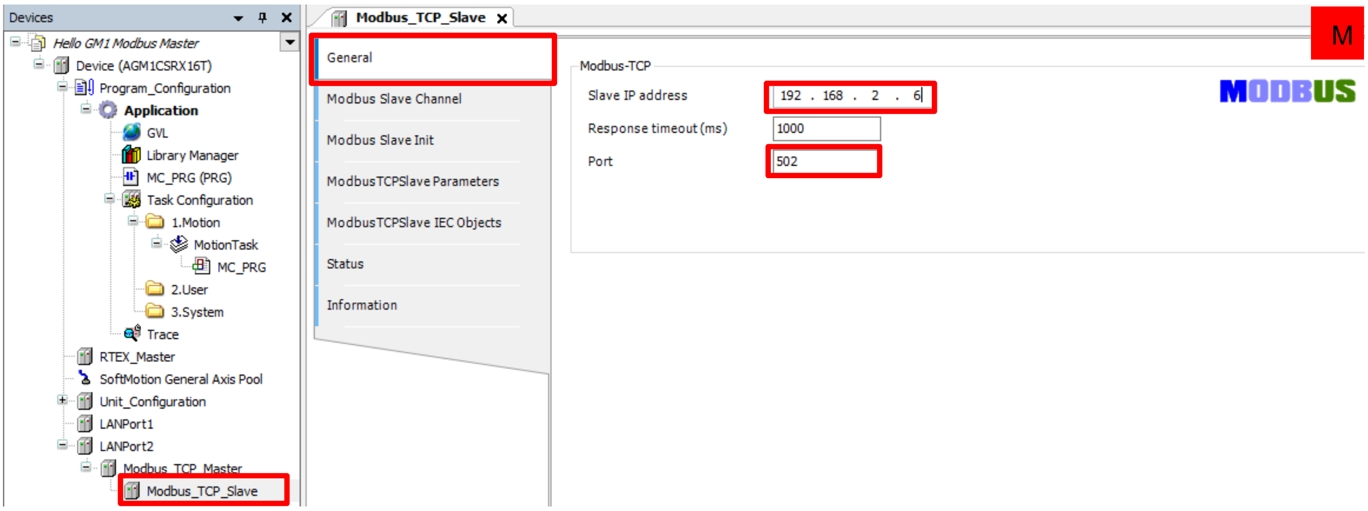
☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: Modbus_TCP_Slave

Step 6

Modbus_TCP_Slave will be added under **Modbus TCP Master**.
Double-click **Modbus_TCP_Slave** and select the **General** tab.

Set an IP address and port number.
Because the IP address (192.168.2.6) and port number (502) of the slave GM1 controller are used, set an IP address and port number as shown in the following figure.



Modbus-TCP

Slave IP address: 192 . 168 . 2 . 6

Response timeout (ms): 1000

Port: 502

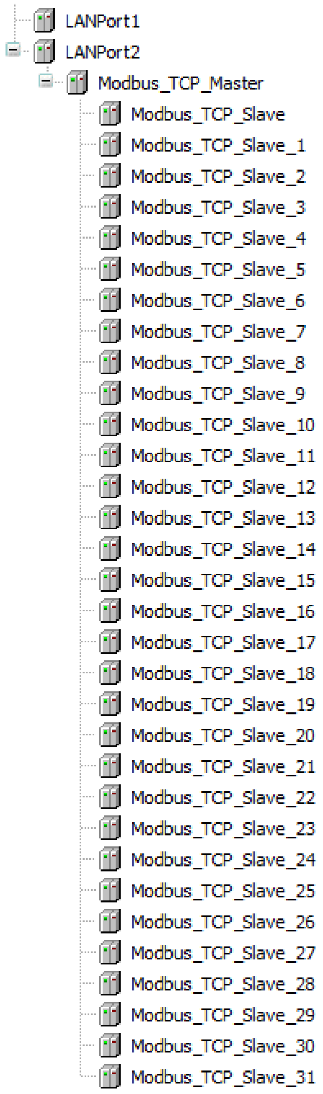
This completes registration of the slave GM1 controller.

Column (3): Number of communication ports

For LAN port 1, 16 connections can be allocated as a communication port. For LAN port 2, 32 connections can be allocated as a communication port.
"Modbus TCP Slave", which has been added just before, is applicable to a connection.

If LAN port 2 is used as an example, 32 instances of "Modbus TCP Slave" can be registered as shown in the figure below. They can be used as different connections by assigning a different IP address or port number to each instance of "Modbus TCP Slave".

If an attempt is made to register more than 32 instances of "Modbus TCP Slave", **Add Device** in the context-sensitive menu will be grayed out and no more instance can be added.



LANPort1
LANPort2
Modbus_TCP_Master
Modbus_TCP_Slave
Modbus_TCP_Slave_1
Modbus_TCP_Slave_2
Modbus_TCP_Slave_3
Modbus_TCP_Slave_4
Modbus_TCP_Slave_5
Modbus_TCP_Slave_6
Modbus_TCP_Slave_7
Modbus_TCP_Slave_8
Modbus_TCP_Slave_9
Modbus_TCP_Slave_10
Modbus_TCP_Slave_11
Modbus_TCP_Slave_12
Modbus_TCP_Slave_13
Modbus_TCP_Slave_14
Modbus_TCP_Slave_15
Modbus_TCP_Slave_16
Modbus_TCP_Slave_17
Modbus_TCP_Slave_18
Modbus_TCP_Slave_19
Modbus_TCP_Slave_20
Modbus_TCP_Slave_21
Modbus_TCP_Slave_22
Modbus_TCP_Slave_23
Modbus_TCP_Slave_24
Modbus_TCP_Slave_25
Modbus_TCP_Slave_26
Modbus_TCP_Slave_27
Modbus_TCP_Slave_28
Modbus_TCP_Slave_29
Modbus_TCP_Slave_30
Modbus_TCP_Slave_31

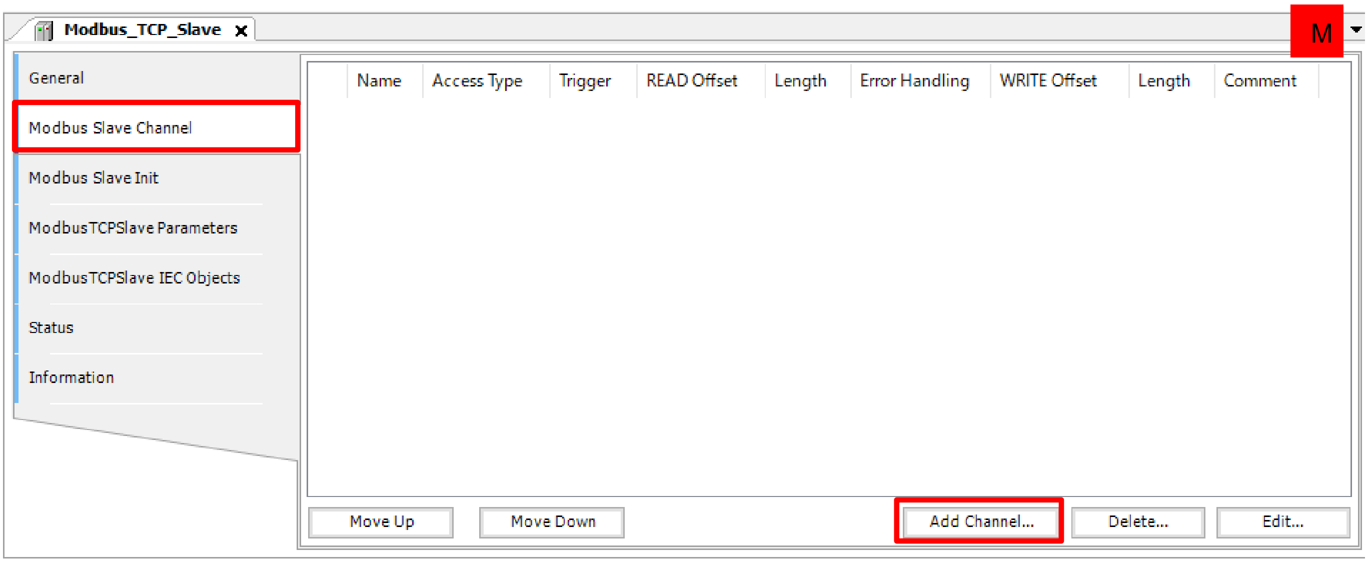
3.2 Read/Write Settings

Add a channel to "Modbus TCP Slave" for each read/write setting.

Step 1

Configure settings for "① Read from Slave_Data0".

In the **Modbus_TCP_Slave** window, select the **Modbus Slave Channel** tab and click **Add Channel**.



INFO

The GM1 controller provides three triggers that start command transmission for Modbus TCP master communication.

Cyclic	Command transmission is automatically executed according to the value specified in **Cycle time**. This trigger is used when values (such as elapsed values) of the slave device must be read periodically.
Rising edge	Command transmission is executed at the rising edge of the trigger bit. Flags indicating execution in progress and completion of execution are not provided.
Application	Command transmission is executed with the ModbusChannel instruction. Flags indicating execution in progress and completion of execution, such as "Busy", "Done", and "Error", are provided for out of instruction function blocks. Therefore, use this method when the user program manages command execution states.

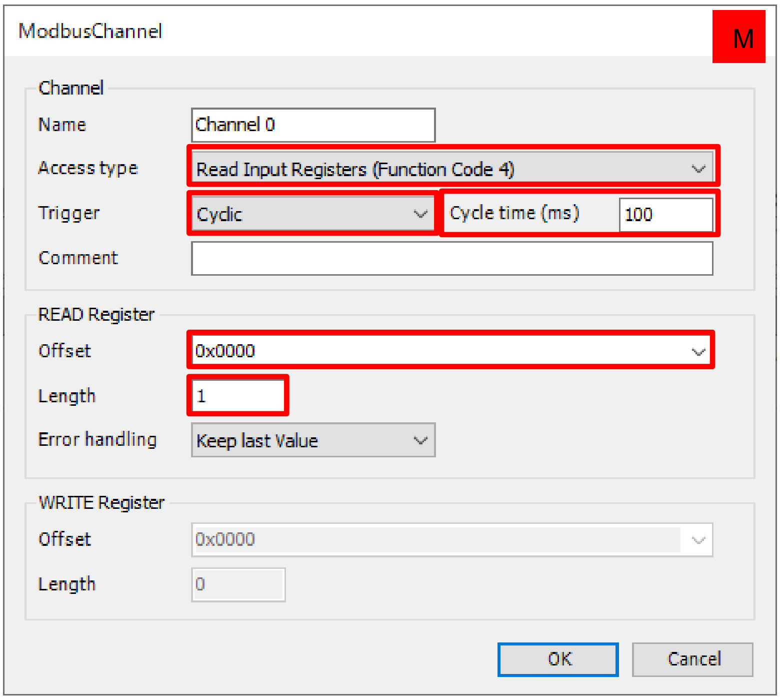
Step 2

The ModbusChannel setting window will open. Set each item as shown below.

Read Modbus input register output [0].

Configure settings to send commands automatically from the master every 100 ms.

Access type	Read Input Resisters (Function code 4)
Trigger	Cyclic
Cycle time	100
Offset	0×0000
Length	1 (Word)



ModbusChannel

Channel

Name: Channel 0

Access type: Read Input Registers (Function Code 4)

Trigger: Cyclic

Cycle time (ms): 100

Comment:

READ Register

Offset: 0x0000

Length: 1

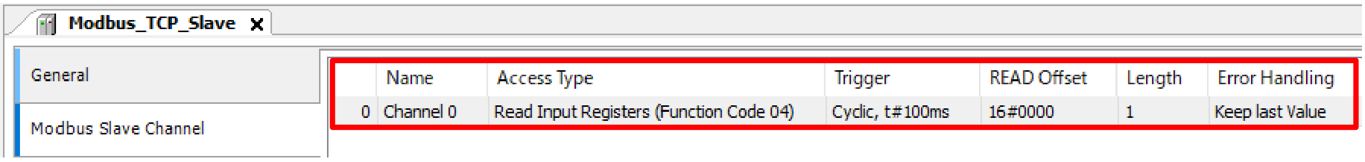
Error handling: Keep last Value

WRITE Register

Offset: 0x0000

Length: 0

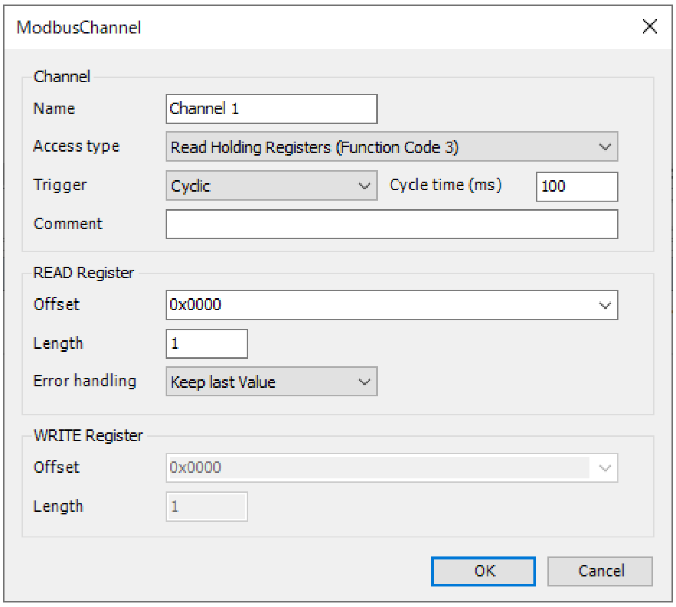
A channel will be added.



	Name	Access Type	Trigger	READ Offset	Length	Error Handling
0	Channel 0	Read Input Registers (Function Code 04)	Cyclic, t#100ms	16#0000	1	Keep last Value

Column (4): Explanation of Modbus channel settings

In the **ModbusChannel** window, **Access type** (function code), **Offset** (transmission/reception destination address), **Length** (sent/received data size), and command transmission timing are set individually.
These settings can specified only once for each channel and up to 100 channels can be registered for each instance of Modbus TCP Slave.



ModbusChannel

Channel

Name: Channel 1

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

Channel	
Name	Name of each channel
Access type	Setting of Modbus function code
Trigger	Selection of transmission trigger type
Cycle time	Enabled when **Cyclic** is selected for **Trigger**
Comment	Allows any comment to be entered for the channel

READ Register	
Offset	Starting address of read destination
Length	Length of data to be read
Error handling	Method of handling when an error occurs

WRITE Register	
Offset	Starting address of write destination
Length	Length of data to be written

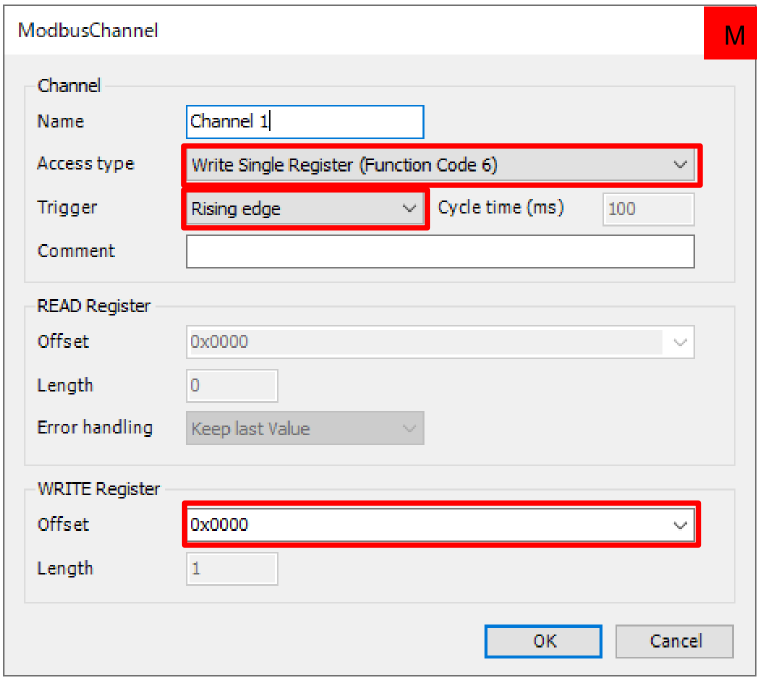
Either **READ Register** or **WRITE Register** is enabled, depending on the function code selected in **Access type**.
In the above example, **READ Register** is enabled because **Read Holding Registers (Function Code 3)** is selected.

Step 3

Configure settings for "② Write to Slave_Data1".

Click **Add Channel** and specify each item as below.

Access type	Write Single Resister (Function Code 6)
Trigger	Rising edge
Offset	0×0000
Length	1 (Because Function Code 6 writes a single word, the length is fixed at "1".)


ModbusChannel

M

Channel

Name: Channel 1

Access type: Write Single Register (Function Code 6)

Trigger: Rising edge

Cycle time (ms): 100

Comment:

READ Register

Offset: 0x0000

Length: 0

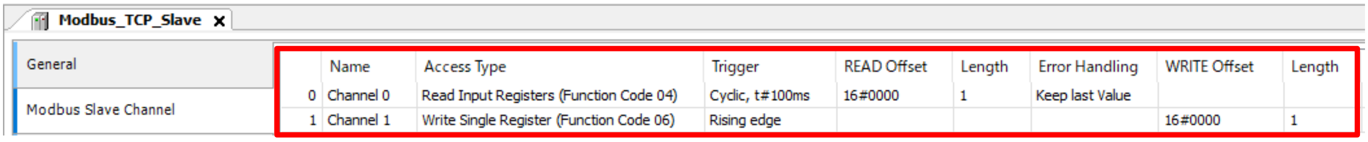
Error handling: Keep last Value

WRITE Register

Offset: 0x0000

Length: 1

A channel will be added.



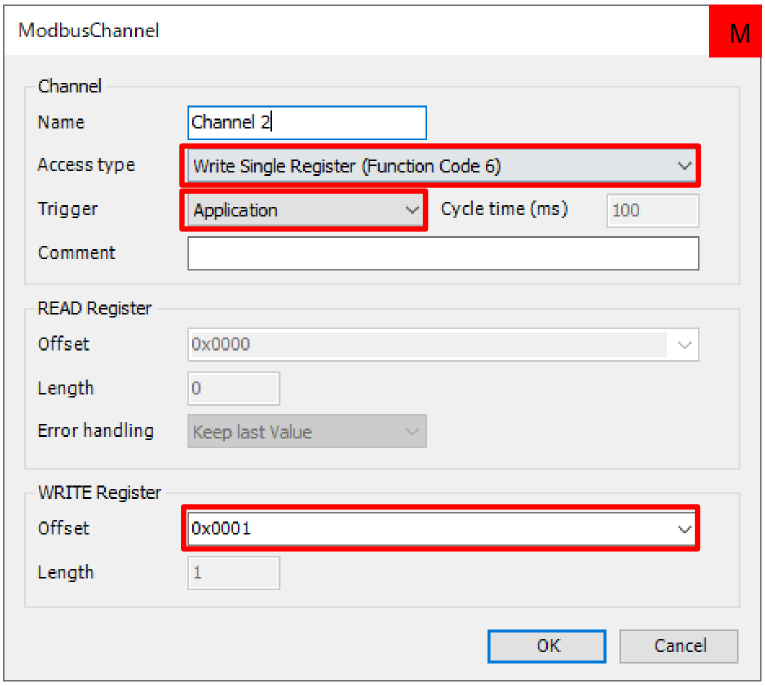
	Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length
0	Channel 0	Read Input Registers (Function Code 04)	Cyclic, t#100ms	16#0000	1	Keep last Value		
1	Channel 1	Write Single Register (Function Code 06)	Rising edge				16#0000	1

Step 5

Configure settings for "③ Write to Slave_Data2".

Click **Add Channel** and specify each item as below.

Access type	Write Single Resister (Function Code 6)
Trigger	Application
Offset	0×0001
Length	1 (Because Function Code 6 writes a single word, the length is fixed at "1".)

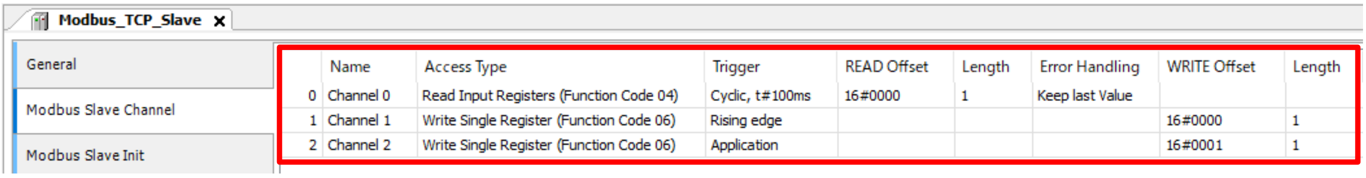


Channel
Name: Channel 2
Access type: Write Single Register (Function Code 6)
Trigger: Application
Cycle time (ms): 100
Comment:

READ Register
Offset: 0x0000
Length: 0
Error handling: Keep last Value

WRITE Register
Offset: 0x0001
Length: 1

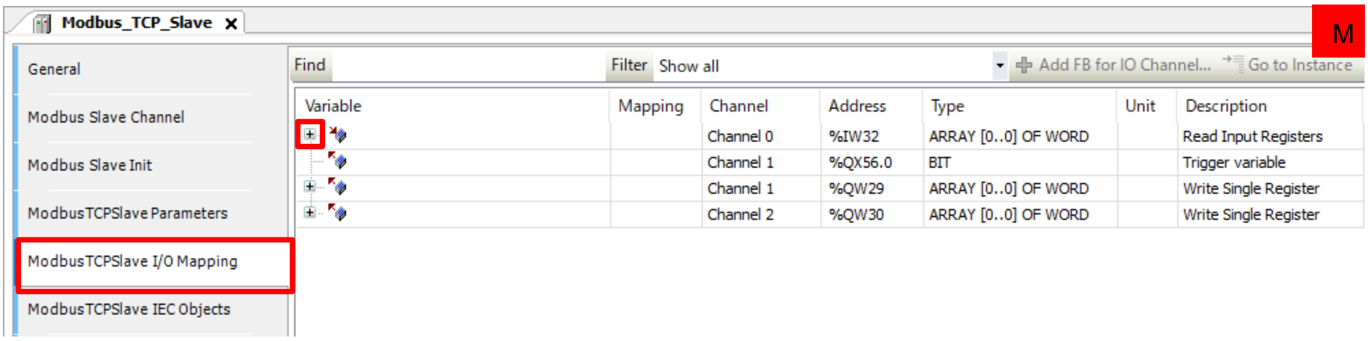
A channel will be added.



	Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length
0	Channel 0	Read Input Registers (Function Code 04)	Cyclic, t#100ms	16#0000	1	Keep last Value		
1	Channel 1	Write Single Register (Function Code 06)	Rising edge				16#0000	1
2	Channel 2	Write Single Register (Function Code 06)	Application				16#0001	1

Step 6

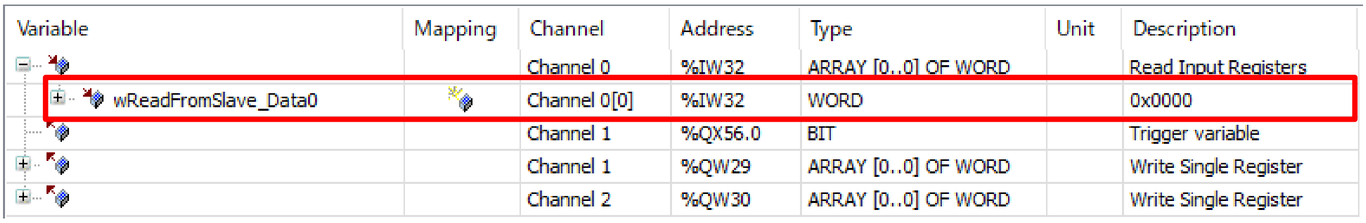
Assign variables to the data area for each channel that has been set.
Select the **Modbus TCP Slave I/O Mapping** tab and click ⊞ on the left side of **Channel 0**.



Variable	Mapping	Channel	Address	Type	Unit	Description
⊞		Channel 0	%IW32	ARRAY [0..0] OF WORD		Read Input Registers
		Channel 1	%QX56.0	BIT		Trigger variable
⊞		Channel 1	%QW29	ARRAY [0..0] OF WORD		Write Single Register
⊞		Channel 2	%QW30	ARRAY [0..0] OF WORD		Write Single Register

Step 7

In the **Variable** column corresponding to **Channel 0[0]**, register a variable to be used.
The location indicated by the offset value (0x0000) of the read destination in the **Description** column is the read destination on the I/O map.
Because Channel 0 is specified as the read destination for Slave_Data0, register wReadFromSlave_Data0 in the **Variable** column.

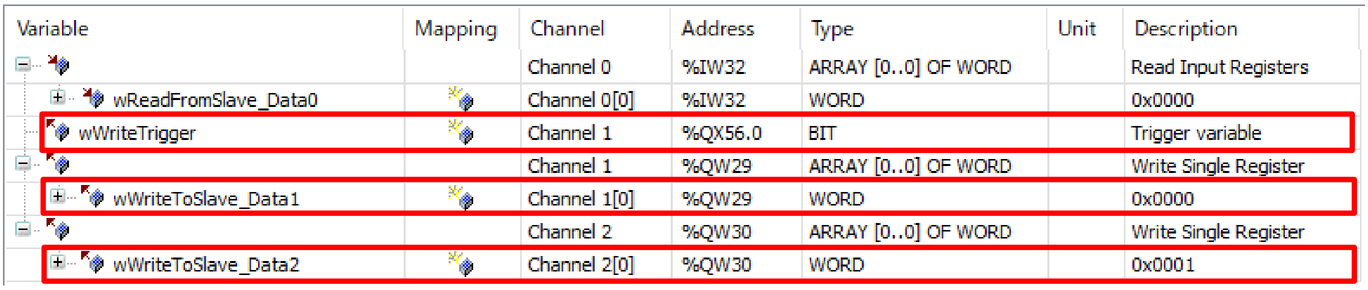


Variable	Mapping	Channel	Address	Type	Unit	Description
⊟		Channel 0	%IW32	ARRAY [0..0] OF WORD		Read Input Registers
⊞ wReadFromSlave_Data0		Channel 0[0]	%IW32	WORD		0x0000
		Channel 1	%QX56.0	BIT		Trigger variable
⊞		Channel 1	%QW29	ARRAY [0..0] OF WORD		Write Single Register
⊞		Channel 2	%QW30	ARRAY [0..0] OF WORD		Write Single Register

Step 8

Click ⊞ on the left side of **Channel 1** and **Channel 2** separately, and register variables to be used in the respective **Variable** columns corresponding to **Channel 1[0]** and **Channel 2[0]**.

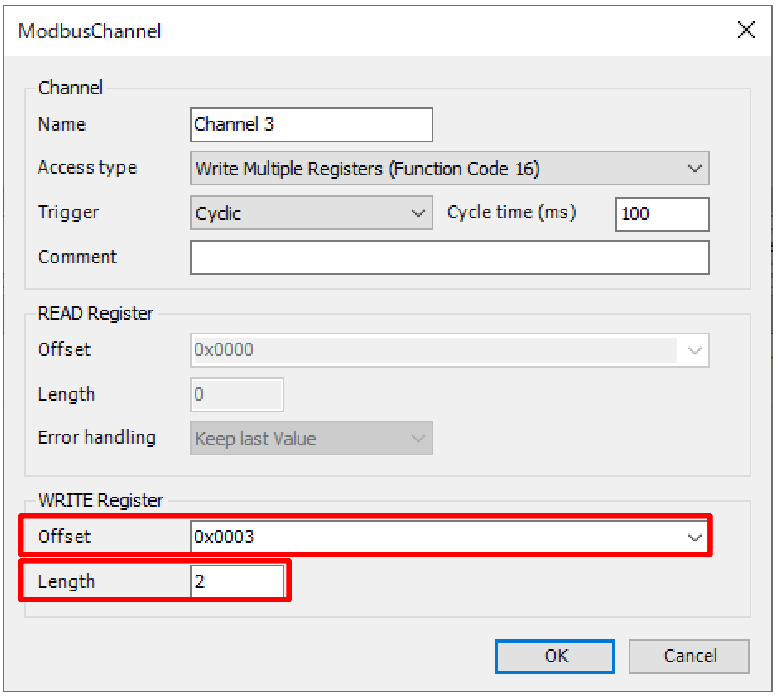
Channel	Variable
Channel1	xWriteTrigger
Channel1[0]	wWriteToSlave_Data1
Channel2[0]	wWriteToSlave_Data2



Variable	Mapping	Channel	Address	Type	Unit	Description
⊟		Channel 0	%IW32	ARRAY [0..0] OF WORD		Read Input Registers
⊞ wReadFromSlave_Data0		Channel 0[0]	%IW32	WORD		0x0000
wWriteTrigger		Channel 1	%QX56.0	BIT		Trigger variable
⊟		Channel 1	%QW29	ARRAY [0..0] OF WORD		Write Single Register
⊞ wWriteToSlave_Data1		Channel 1[0]	%QW29	WORD		0x0000
⊟		Channel 2	%QW30	ARRAY [0..0] OF WORD		Write Single Register
⊞ wWriteToSlave_Data2		Channel 2[0]	%QW30	WORD		0x0001

This completes Modbus master communication setup for the GM1 controller.

Column (5): Offset and length of Modbus channel



Channel

Name: Channel 3

Access type: Write Multiple Registers (Function Code 16)

Trigger: Cyclic Cycle time (ms): 100

Comment:

READ Register

Offset: 0x0000

Length: 0

Error handling: Keep last Value

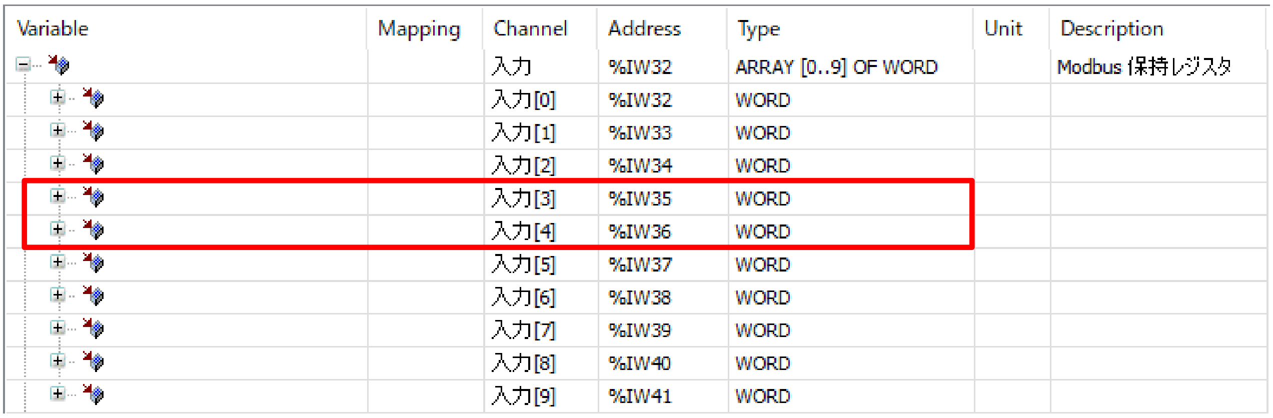
WRITE Register

Offset: 0x0003

Length: 2

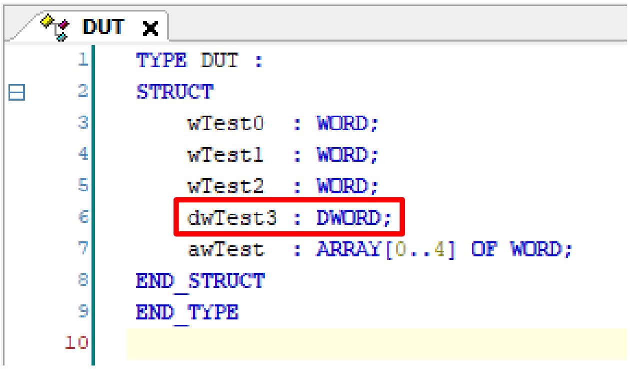
Because **Offset** is set to 0×0003, Input [3] is used.

Because **Length** is set to 2, Input [3] plus two words, which means that Input [3] and Input [4] are applicable.



Variable	Mapping	Channel	Address	Type	Unit	Description
		入力	%IW32	ARRAY [0..9] OF WORD		Modbus 保持レジスタ
		入力[0]	%IW32	WORD		
		入力[1]	%IW33	WORD		
		入力[2]	%IW34	WORD		
		入力[3]	%IW35	WORD		
		入力[4]	%IW36	WORD		
		入力[5]	%IW37	WORD		
		入力[6]	%IW38	WORD		
		入力[7]	%IW39	WORD		
		入力[8]	%IW40	WORD		
		入力[9]	%IW41	WORD		

If a variable containing two or more words is used, it can be handled by declaring a variable whose Data Unit Type (DUT) can handle the number of words (length).



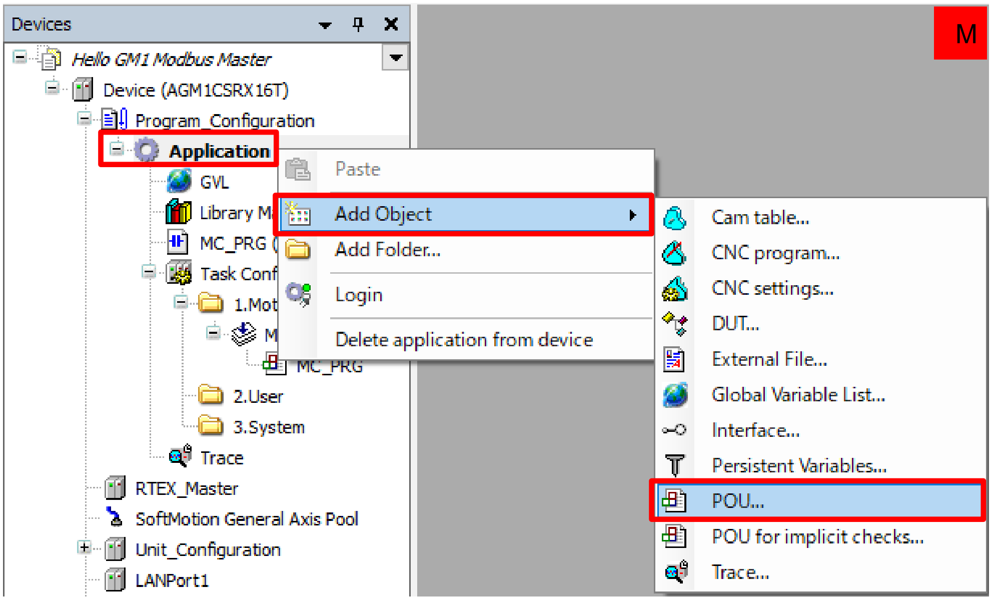
```
TYPE DUT :
STRUCT
    wTest0  : WORD;
    wTest1  : WORD;
    wTest2  : WORD;
    dwTest3 : DWORD;
    awTest  : ARRAY[0..4] OF WORD;
END_STRUCT
END_TYPE
```

3.3 Creating Write Program (Trigger: Rising Edge)

Create a program that writes the value read from Slave_Data0 on the slave GM1 controller plus 1000 to Slave_Data1.

Step 1

Right-click **Application** and select **Add Object** and then **POU** to create a new POU.



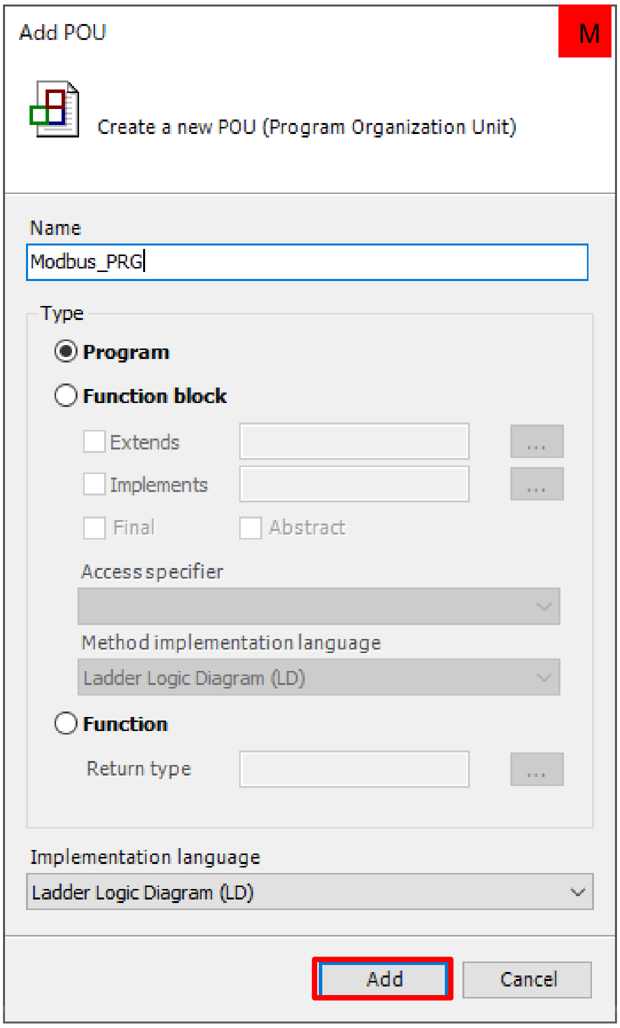
Step 2

In the **Add POU** dialog box, specify settings as below and click **Add**.

Name: **Modbus_PRG**

Type: **Program**

Implementation language: Ladder Logic Diagram (LD)



Add POU

Create a new POU (Program Organization Unit)

Name: Modbus_PRG

Type

● **Program**

○ **Function block**

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

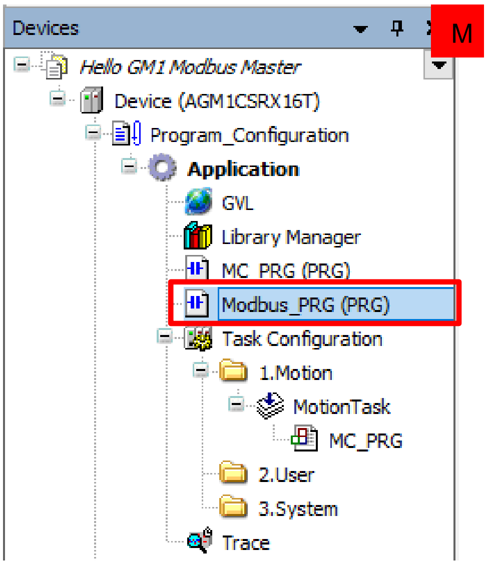
Method implementation language: Ladder Logic Diagram (LD)

○ **Function**

Return type

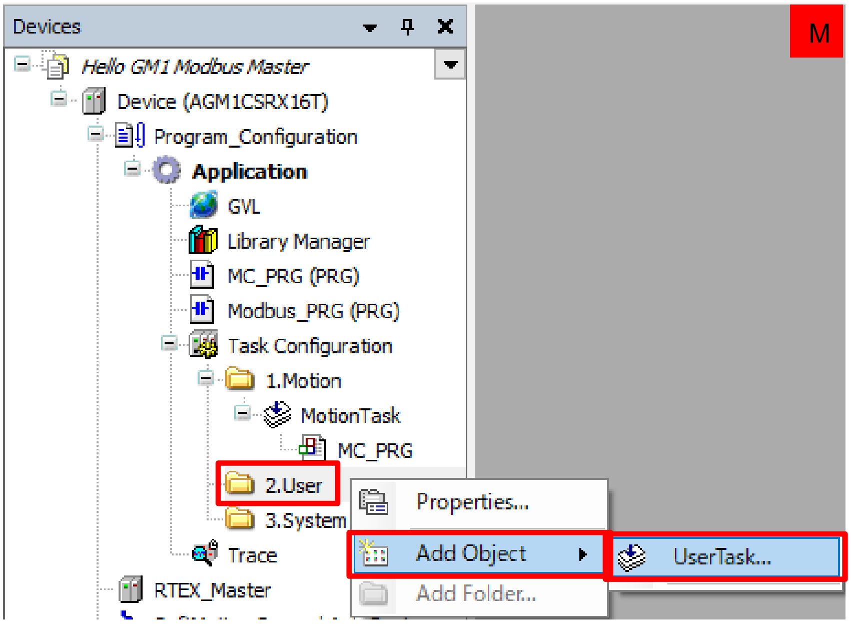
Implementation language: Ladder Logic Diagram (LD)

Modbus_PRG (PRG) will be added to **Application**.



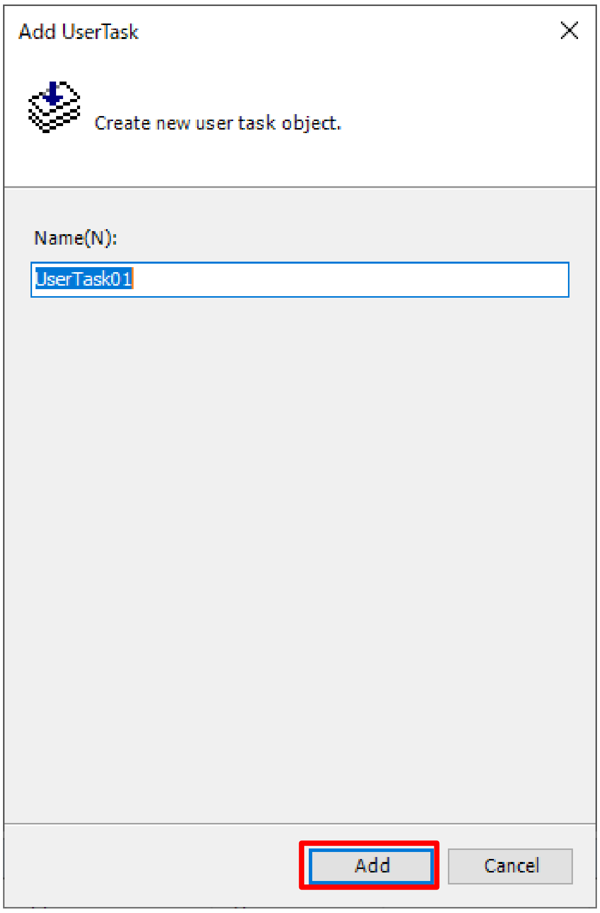
Step 3

Right-click **2.User** and select **Add Object** and then **UserTask**.



Step 4

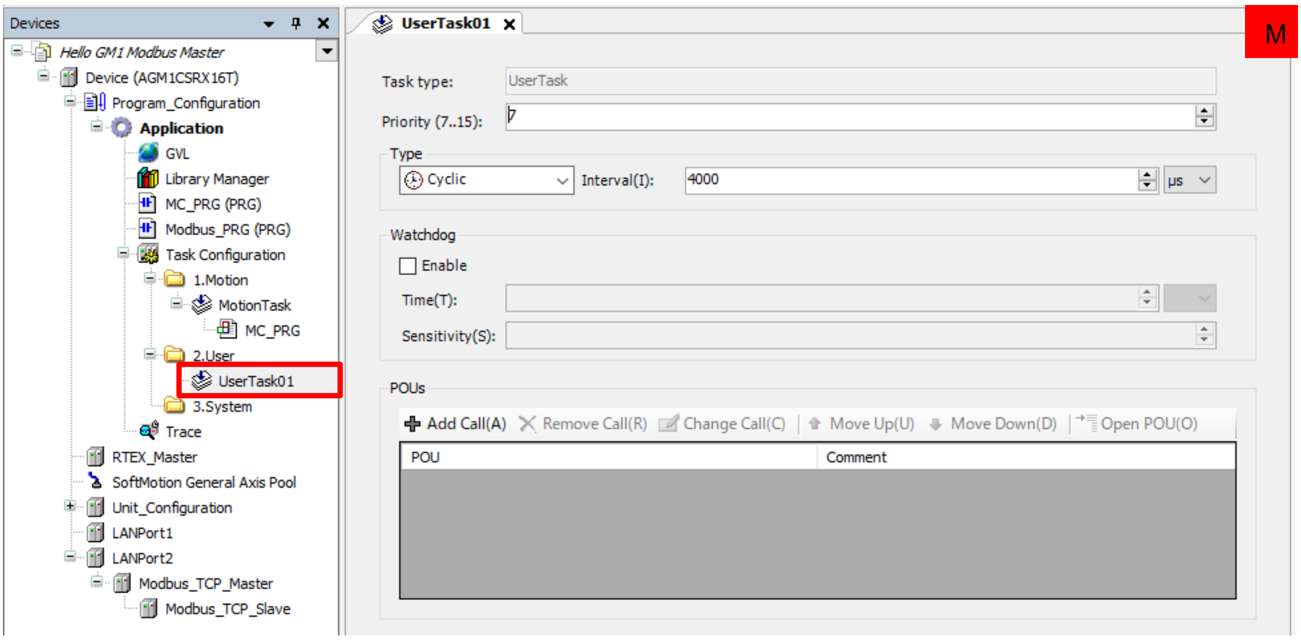
The **Add UserTask** dialog box will be displayed. Leave **UserTask01** (default) unchanged in the **Name(N)** field and click **Add**.


Add UserTask

Create new user task object.

Name(N): UserTask01

UserTask01 will be added.



Task type: UserTask

Priority (7..15): 7

Type

Cyclic Interval(I): 4000 µs

Watchdog

□ Enable

Time(T):

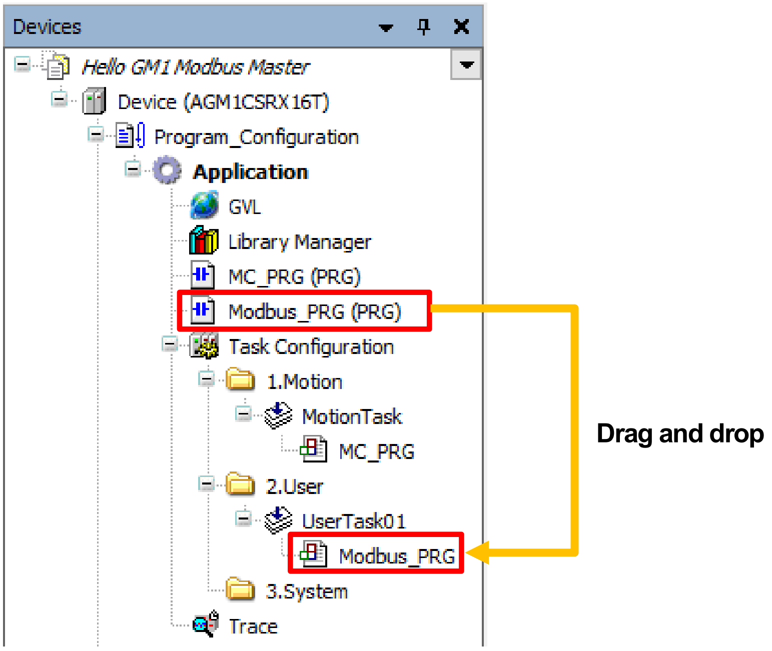
Sensitivity(S):

POUs

POU	Comment

Step 5

Drag and drop **Modbus_PRG (PRG)** into the **UserTask01** object, which has been added, to add it to the task.



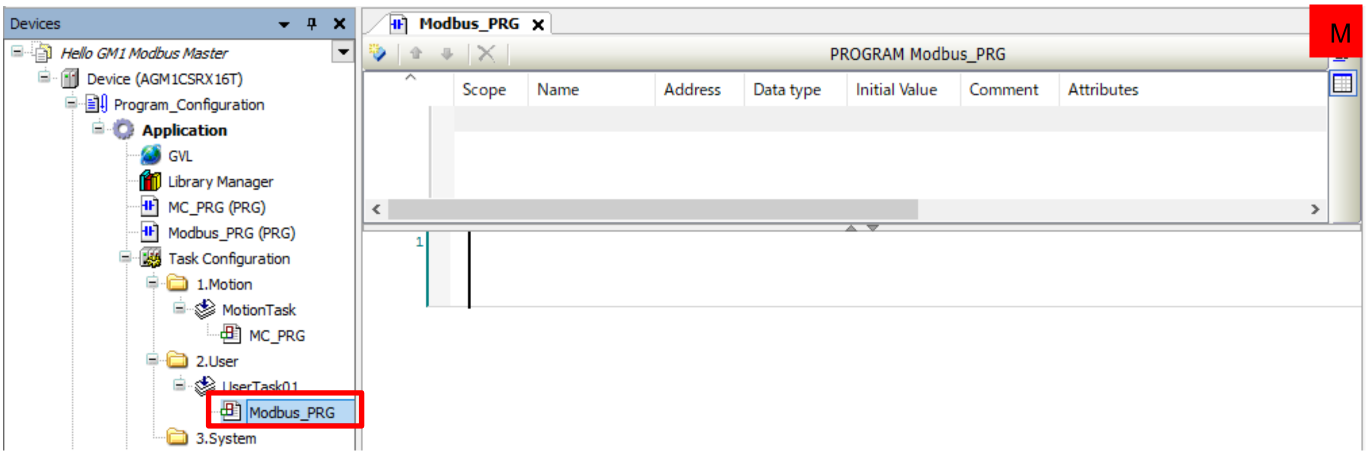
Drag and drop

Column (6): 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.

Step 6

Double-click **Modbus_PRG** to open the program pane.

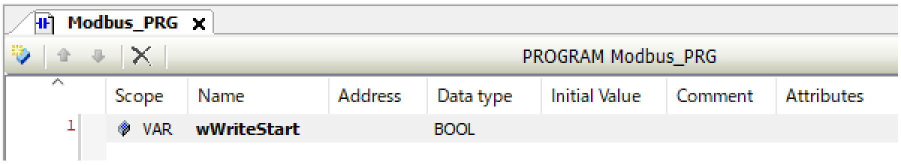


Step 7

Add a local variable. With the **Name** column selected, register the following variable.

Name: xWriteStart

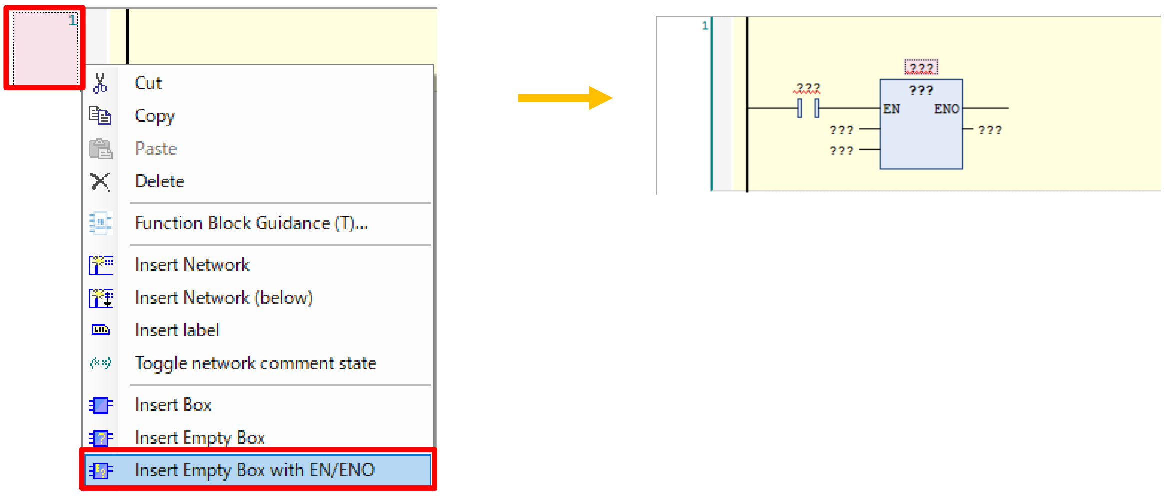
Data type: BOOL



	Scope	Name	Address	Data type	Initial Value	Comment	Attributes
1	VAR	**wWriteStart**		BOOL			

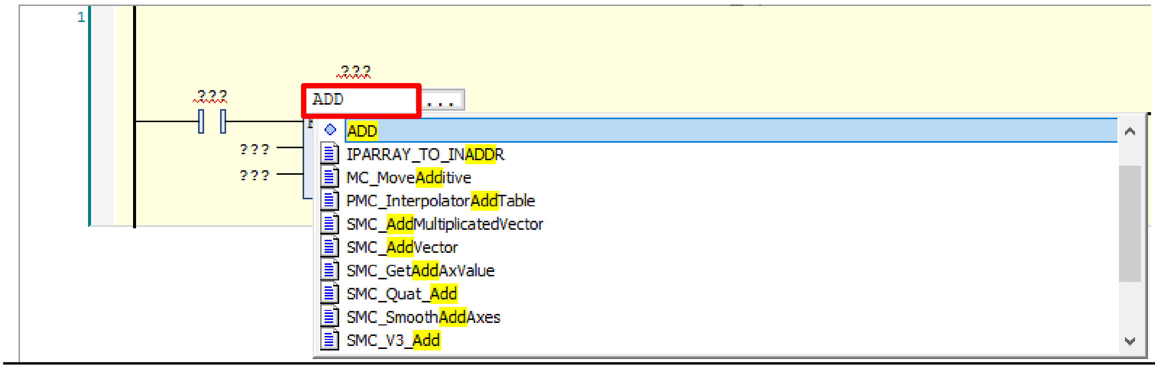
Step 8

Right-click the leftmost section of the network (the red section in the figure below) and select **Insert Empty Box with EN/ENO**.



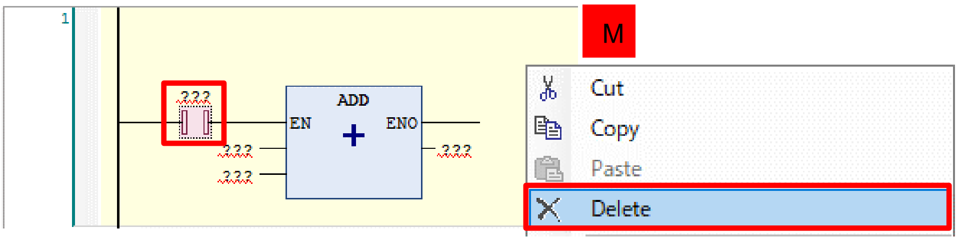
Step 9

Click **???** inside the box, enter ADD, and then press the Enter key.

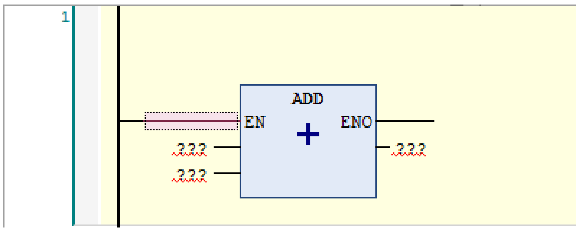


Step 10

Right-click the NO contact for the ADD instruction, which has been entered, and then select **Delete**.

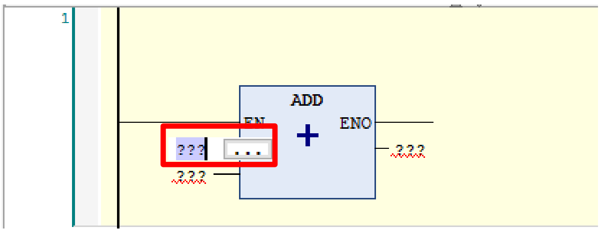


The contact on the input side will be deleted, making the ADD instruction executable at all times.



Step 11

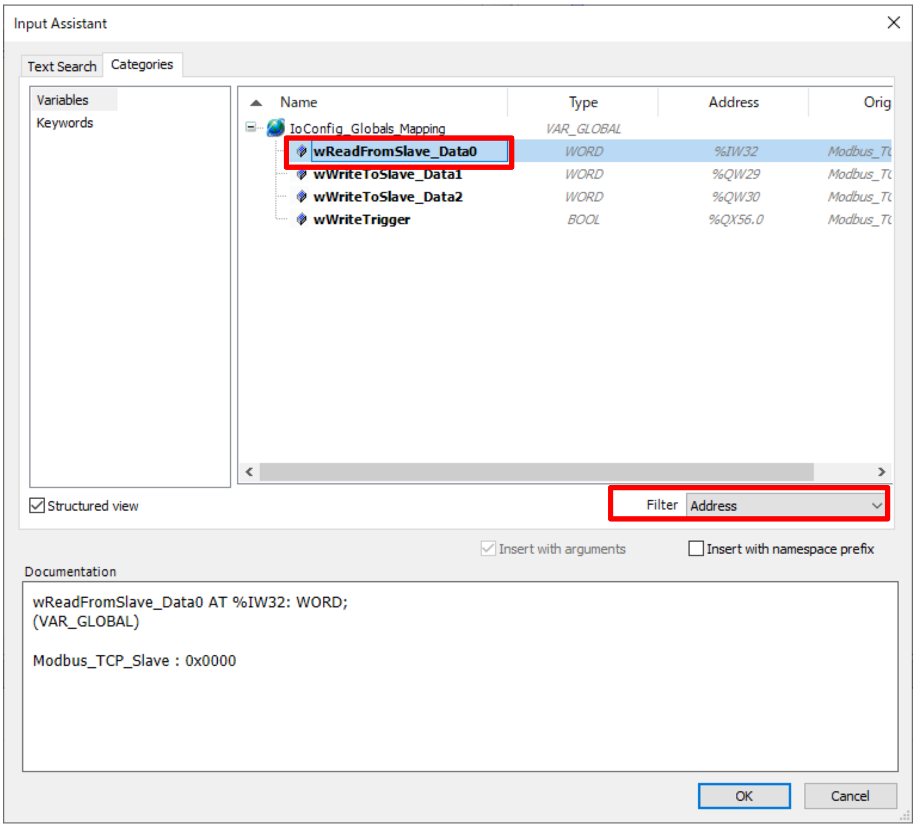
Click **???** representing an input on the upper side and then click [...] to open the **Input Assistant** dialog box.



Step 12

Select **IoConfig_Globals_Mapping** and then **wReadFromSlave_Data0** and click **OK**.

Select **Address** in the **Filter** drop-down list. Only the variable registered in **IoConfig_Globals_Mapping** above will be displayed.



Name	Type	Address	Orig
IoConfig_Globals_Mapping	VAR_GLOBAL		
wReadFromSlave_Data0	WORD	%IW32	Modbus_T(
wWriteToSlave_Data1	WORD	%QW29	Modbus_T(
wWriteToSlave_Data2	WORD	%QW30	Modbus_T(
wWriteTrigger	BOOL	%QX56.0	Modbus_T(

☑ Structured view

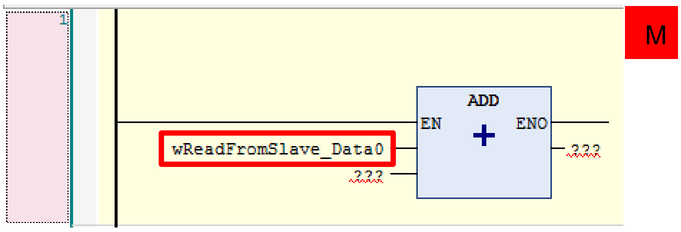
Filter: Address

☑ Insert with arguments ☐ Insert with namespace prefix

Documentation

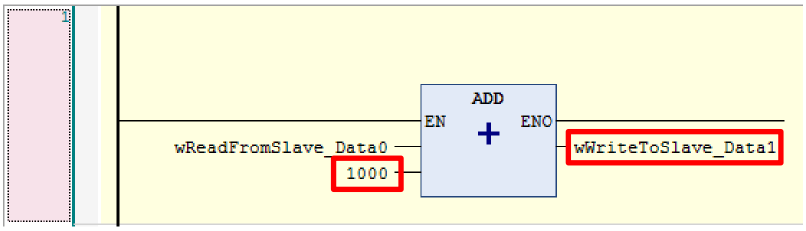
```
wReadFromSlave_Data0 AT %IW32: WORD;
(VAR_GLOBAL)

Modbus_TCP_Slave : 0x0000
```

wReadFromSlave_Data0 will be inserted.



Step 13

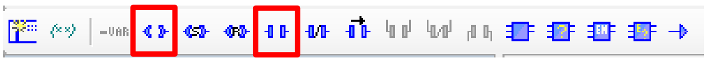
Similarly, use the **Input Assistant** dialog box to insert "wWriteToSlave_Data1" into the "**???**" position representing an output. Enter 1000 as a fixed value into the **???** position representing an input on the lower side.

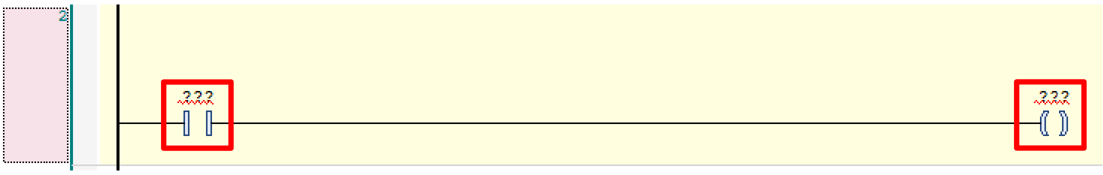


A simple calculation will be performed by offsetting 1000 against the value read from the slave GM1 controller.

Step 14

Insert a new network and then insert a contact and coil from the toolbar.



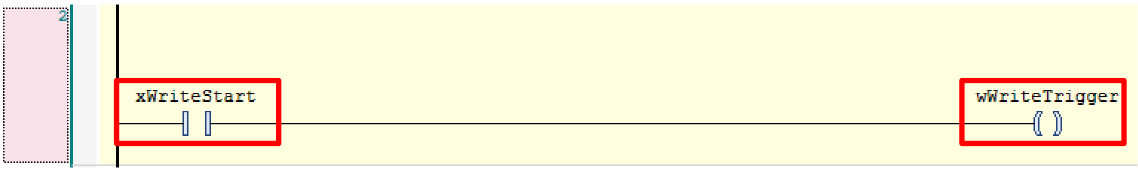


Step 15

Use the **Input Assistant** dialog box to select the following respective variables for the contact and coil.

Contact: wWriteStart

Coil: wWriteTrigger

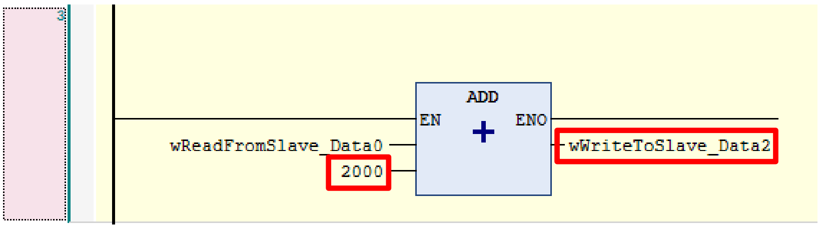


3.4 Creating Write Program (Trigger: Application)

First, create a program that adds 2,000 to the value of "wReadFromSlave_Data0" read from the slave GM1 controller.

Step 1

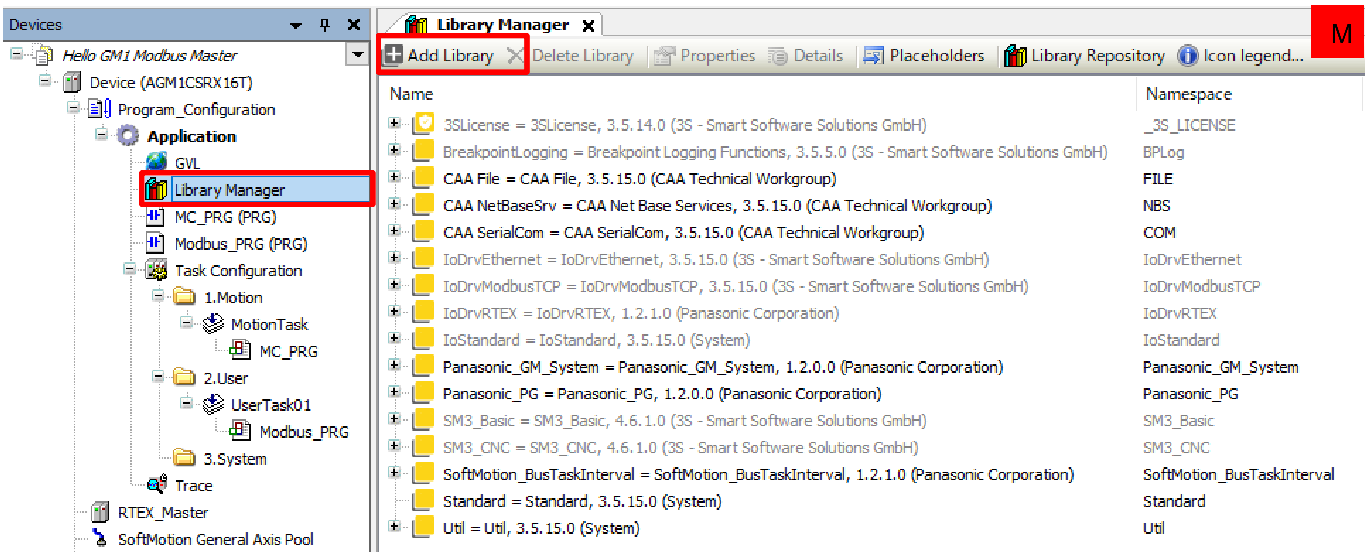
Insert a new network and then insert **ADD** as shown below.



Next, use the ModbusChannel instruction when **Trigger** is set to **Application**.

Step 1

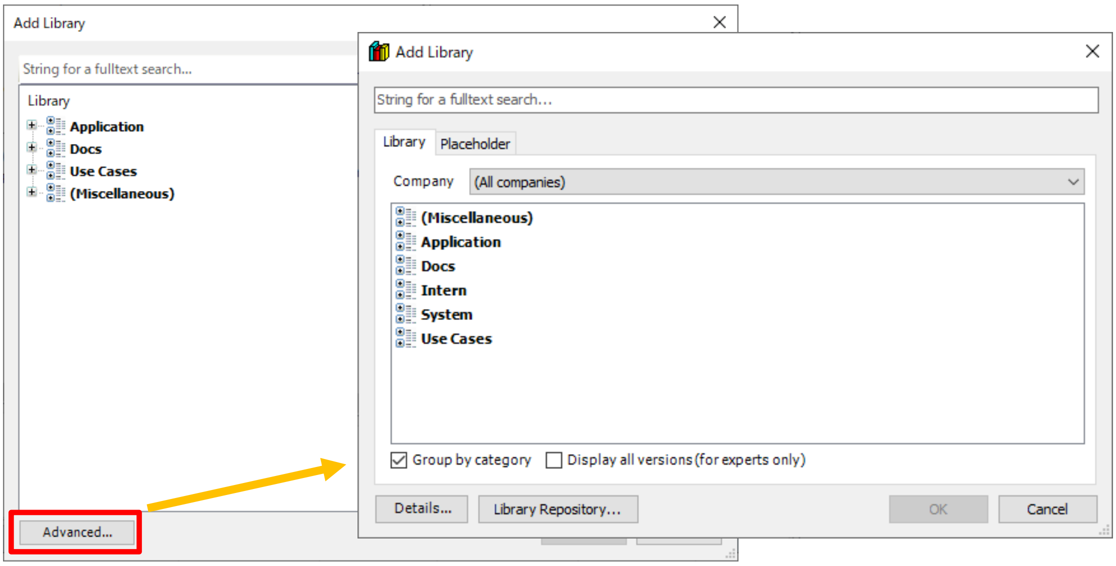
In the navigation pane, double-click **Library Manager**.
In the setting pane that is displayed, click **Add Library**.



Name	Namespace
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog
CAA File = CAA File, 3.5.15.0 (CAA Technical Workgroup)	FILE
CAA NetBaseSrv = CAA Net Base Services, 3.5.15.0 (CAA Technical Workgroup)	NBS
CAA SerialCom = CAA SerialCom, 3.5.15.0 (CAA Technical Workgroup)	COM
IoDrvEthernet = IoDrvEthernet, 3.5.15.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet
IoDrvModbusTCP = IoDrvModbusTCP, 3.5.15.0 (3S - Smart Software Solutions GmbH)	IoDrvModbusTCP
IoDrvRTEX = IoDrvRTEX, 1.2.1.0 (Panasonic Corporation)	IoDrvRTEX
IoStandard = IoStandard, 3.5.15.0 (System)	IoStandard
Panasonic_GM_System = Panasonic_GM_System, 1.2.0.0 (Panasonic Corporation)	Panasonic_GM_System
Panasonic_PG = Panasonic_PG, 1.2.0.0 (Panasonic Corporation)	Panasonic_PG
SM3_Basic = SM3_Basic, 4.6.1.0 (3S - Smart Software Solutions GmbH)	SM3_Basic
SM3_CNC = SM3_CNC, 4.6.1.0 (3S - Smart Software Solutions GmbH)	SM3_CNC
SoftMotion_BusTaskInterval = SoftMotion_BusTaskInterval, 1.2.1.0 (Panasonic Corporation)	SoftMotion_BusTaskInterval
Standard = Standard, 3.5.15.0 (System)	Standard
Util = Util, 3.5.15.0 (System)	Util

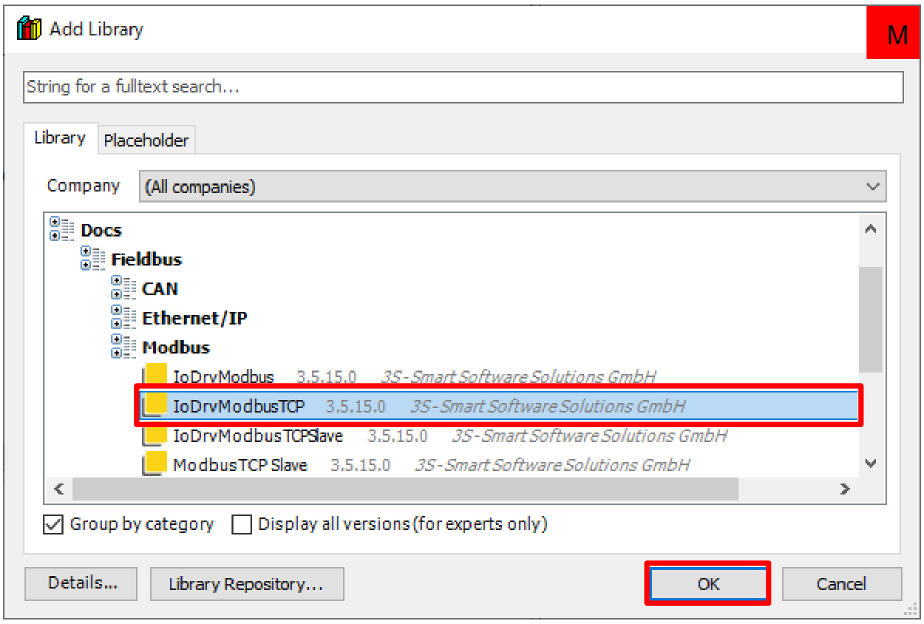
Step 2

In the **Add Library** dialog box that is displayed, click **Advanced** to switch the display of the dialog box.



Step 3

In the dialog box, select **Docs**, **Fieldbus**, **Modbus**, and then **IoDrvModbusTCP** and click **OK**.



Add Library

String for a fulltext search...

Library Placeholder

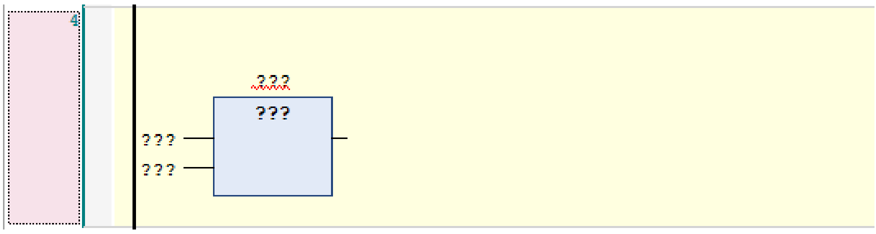
Company: (All companies)

- Docs
  - Fieldbus
    - CAN
    - Ethernet/IP
    - Modbus
      - IoDrvModbus 3.5.15.0 3S-Smart Software Solutions GmbH
      - IoDrvModbusTCP 3.5.15.0 3S-Smart Software Solutions GmbH
      - IoDrvModbusTCPSlave 3.5.15.0 3S-Smart Software Solutions GmbH
      - ModbusTCP Slave 3.5.15.0 3S-Smart Software Solutions GmbH

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

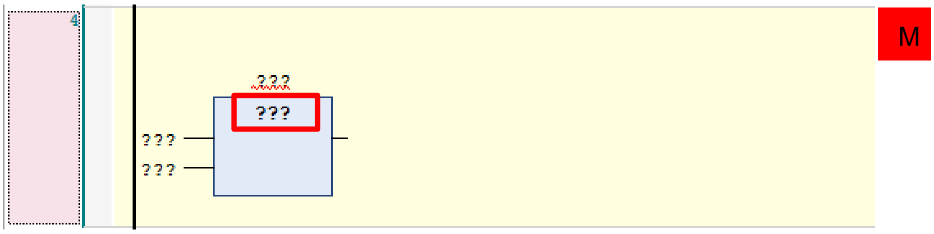
Step 4

Insert a new network, and right-click and then select **Insert Empty Box**.



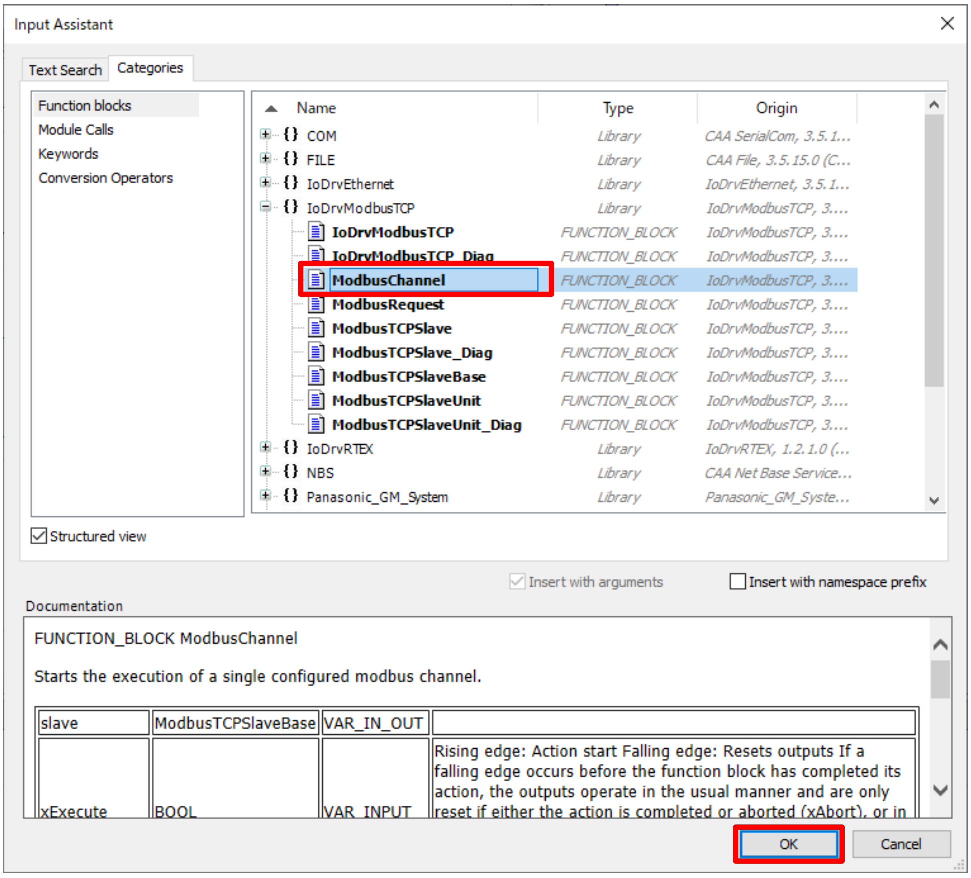
Step 5

Display the **Input Assistant** dialog box from the **???** position in the box.



Step 6

Select **IoDrvModbusTCP** and then **ModbusChannel** and click **OK**.


Input Assistant

Text Search | Categories

- Function blocks
- Module Calls
- Keywords
- Conversion Operators

Name	Type	Origin
{} COM	Library	CAA SerialCom, 3.5.1...
{} FILE	Library	CAA File, 3.5.15.0 (C...
{} IoDrvEthernet	Library	IoDrvEthernet, 3.5.1...
{} IoDrvModbusTCP	Library	IoDrvModbusTCP, 3....
IoDrvModbusTCP	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
IoDrvModbusTCP_Diag	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusChannel	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusRequest	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusTCPSlave	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusTCPSlave_Diag	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusTCPSlaveBase	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusTCPSlaveUnit	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
ModbusTCPSlaveUnit_Diag	FUNCTION_BLOCK	IoDrvModbusTCP, 3....
{} IoDrvRTEX	Library	IoDrvRTEX, 1.2.1.0 (...
{} NBS	Library	CAA Net Base Service...
{} Panasonic_GM_System	Library	Panasonic_GM_Syste...

☑ Structured view

☑ Insert with arguments   □ Insert with namespace prefix

Documentation

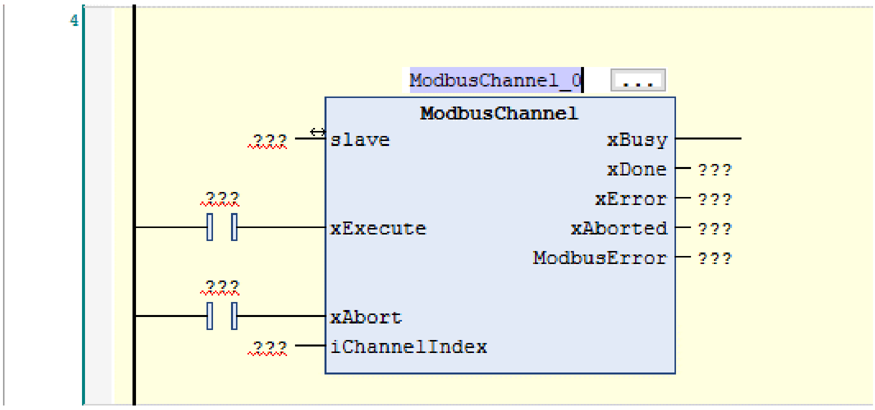
FUNCTION_BLOCK ModbusChannel

Starts the execution of a single configured modbus channel.

slave	ModbusTCPSlaveBase	VAR_IN_OUT	
xExecute	BOOL	VAR_INPUT	Rising edge: Action start Falling edge: Resets outputs If a falling edge occurs before the function block has completed its action, the outputs operate in the usual manner and are only reset if either the action is completed or aborted (xAbort), or in

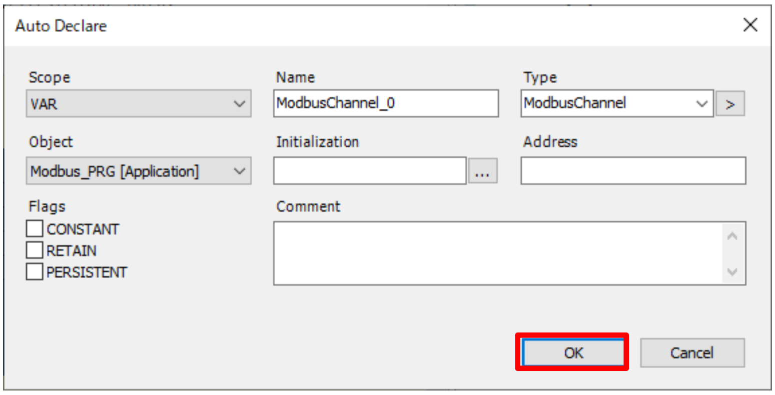
OK | Cancel

Function block "ModbusChannel" will be inserted in Network 4.



Step 7

In this situation, press the Enter key on the PC. The **Auto Declare** dialog box will be displayed. Check the content of the dialog box and click **OK**.



Auto Declare

Scope: VAR

Name: ModbusChannel_0

Type: ModbusChannel

Object: Modbus_PRG [Application]

Initialization:

Address:

Flags

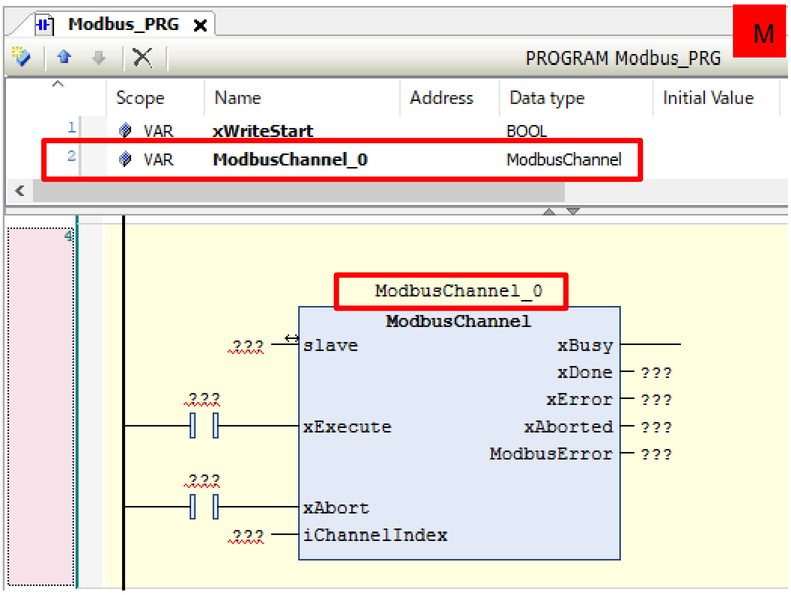
□ CONSTANT

□ RETAIN

□ PERSISTENT

Comment:

ModbusChannnel_0 will be automatically inserted into the variable editor pane.



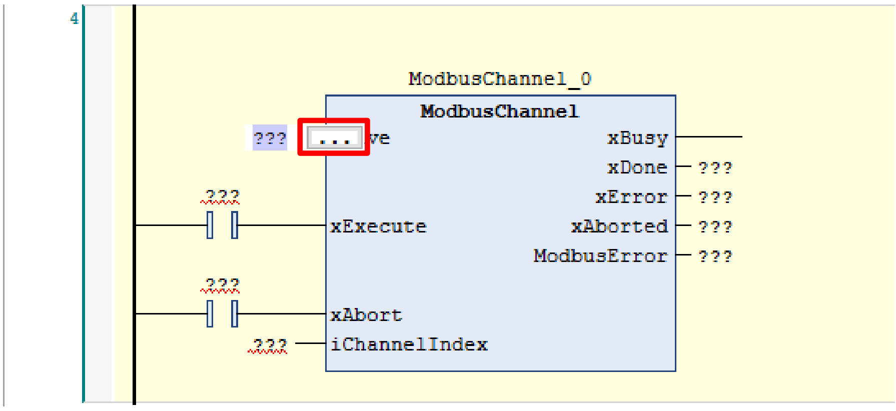
Modbus_PRG

PROGRAM Modbus_PRG

	Scope	Name	Address	Data type	Initial Value
1	VAR	xWriteStart		BOOL	
2	VAR	ModbusChannel_0		ModbusChannel	

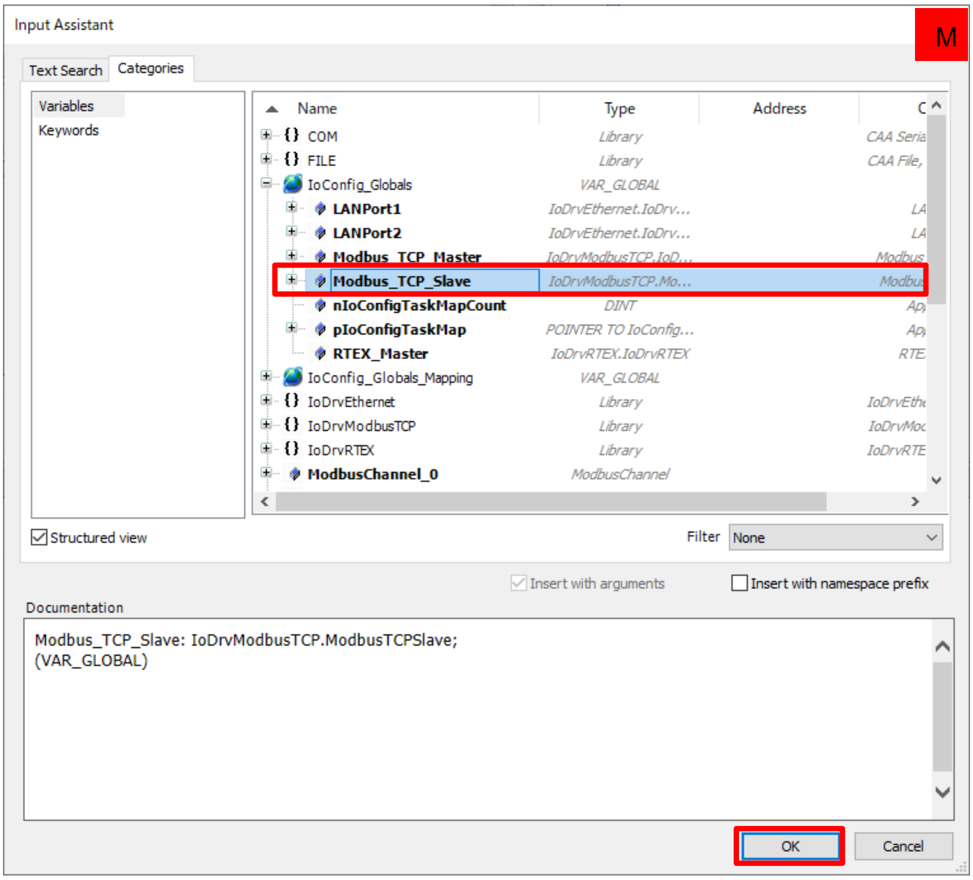
Step 8

Configure I/O settings for the ModbusChannel instruction.
Click **???** representing input "Slave" and then click ... .



Step 9

The **Input Assistant** dialog box will be displayed. Select **IoConfig_Globals** and then **Modbus_TCP_Slave** and click **OK**.



Name	Type	Address	C
{} COM	Library		CAA Seria
{} FILE	Library		CAA File,
IoConfig_Globals	VAR_GLOBAL		
LANPort1	IoDrvEthernet.IoDrv...		LA
LANPort2	IoDrvEthernet.IoDrv...		LA
Modbus_TCP_Master	IoDrvModbusTCP.IoD...		Modbus
Modbus_TCP_Slave	IoDrvModbusTCP.Mo...		Modbu
nIoConfigTaskMapCount	DINT		Ap
pIoConfigTaskMap	POINTER TO IoConfig...		Ap
RTEX_Master	IoDrvRTEX.IoDrvRTEX		RTE
IoConfig_Globals_Mapping	VAR_GLOBAL		
{} IoDrvEthernet	Library		IoDrvEthe
{} IoDrvModbusTCP	Library		IoDrvMoc
{} IoDrvRTEX	Library		IoDrvRTE
ModbusChannel_0	ModbusChannel		

☑ Structured view

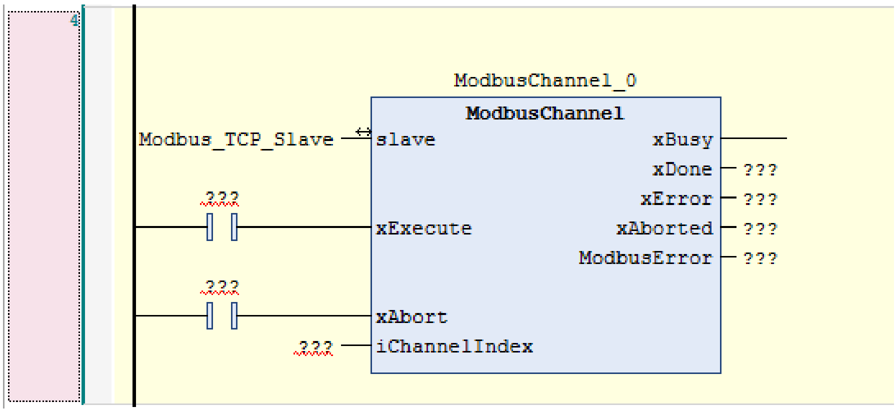
Filter: None

☑ Insert with arguments ☐ Insert with namespace prefix

Documentation

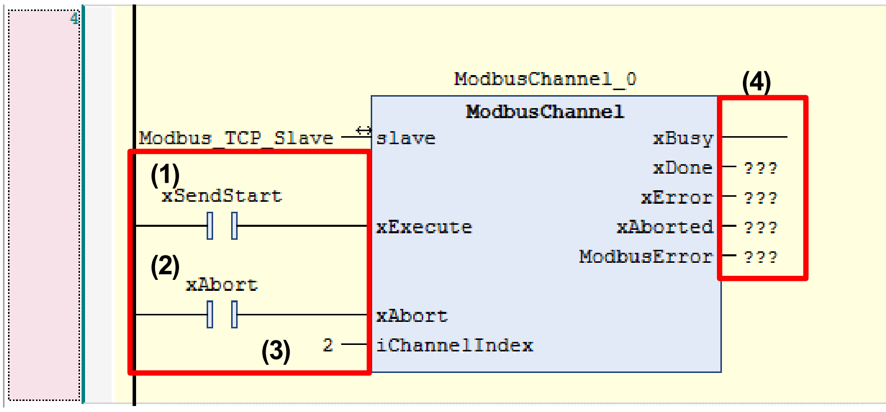
Modbus_TCP_Slave: IoDrvModbusTCP.ModbusTCPSlave;
(VAR_GLOBAL)

Modbus_TCP_Slave will be inserted into input "slave".



Step 10

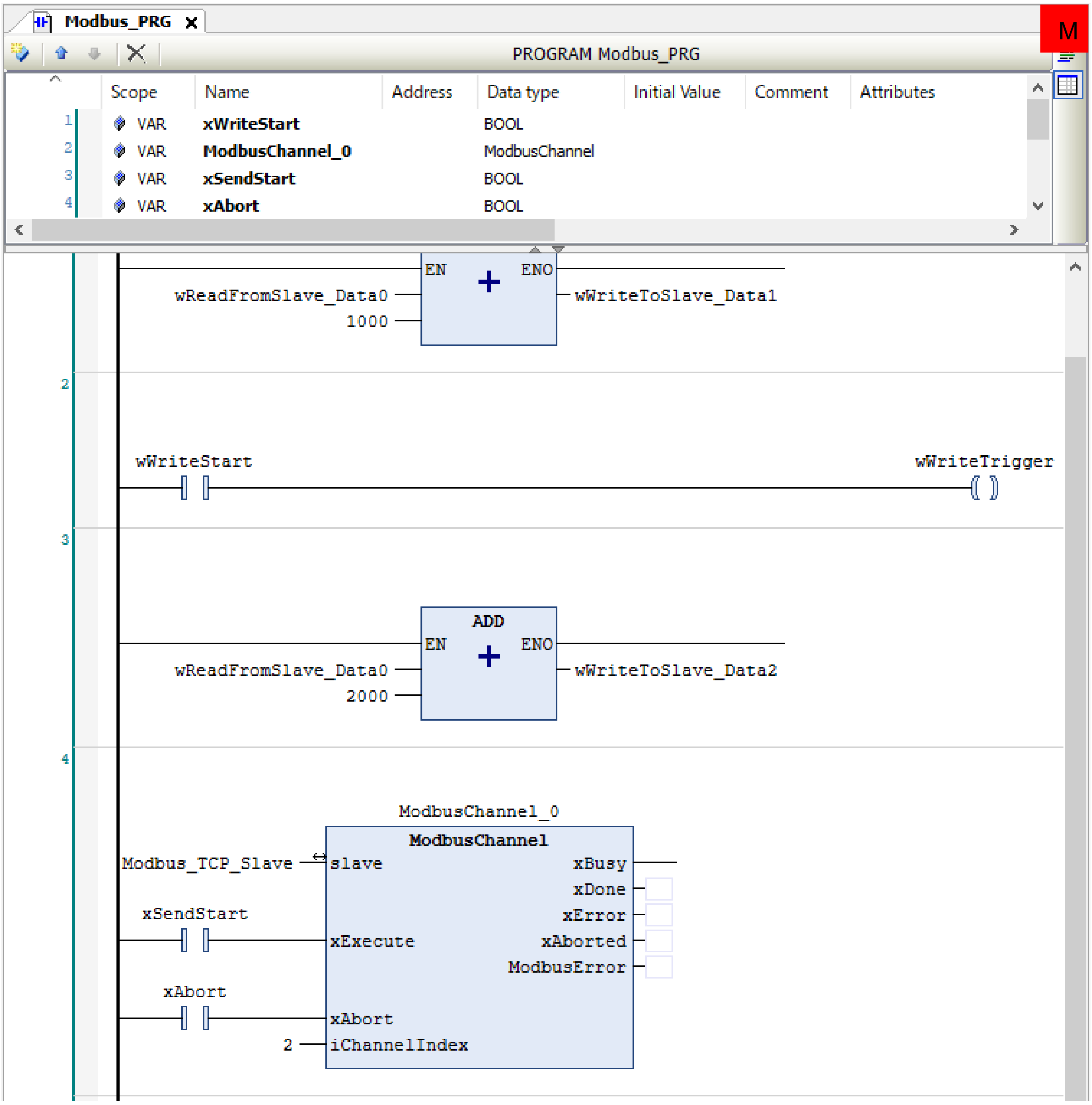
In addition, complete the function block as show below.



	Type	Parameter name	Set value	Description
(1)	Input	xExecute	xSendStart	Starts sending commands at the rising edge
(2)		xAbort	xAbort	TRUE: Stops execution and cancels all outputs
(3)		iChannelIndex	2	Channel number where commands to be sent are set
(4)	Output	xBusy	Delete ???	TRUE: Processing of the FB is incomplete
		xDone	Delete ???	TRUE: Processing is completed
		xError	Delete ???	TRUE: Error has occurred within FB
		xAborted	Delete ???	TRUE: Execution is stopped by the user’s xAbort input
		ModbusError	Delete ???	An error code is output.

Step 11

This completes Modbus master communication setup and programming.
Temporarily save the project data in overwrite mode.
Execute build in GM Programmer and make sure that no error occurs.
Perform a download on the GM1 controller and switch the operation mode to RUN.



	Scope	Name	Address	Data type	Initial Value	Comment	Attributes
1	VAR	**xWriteStart**		BOOL			
2	VAR	**ModbusChannel_0**		ModbusChannel			
3	VAR	**xSendStart**		BOOL			
4	VAR	**xAbort**		BOOL			

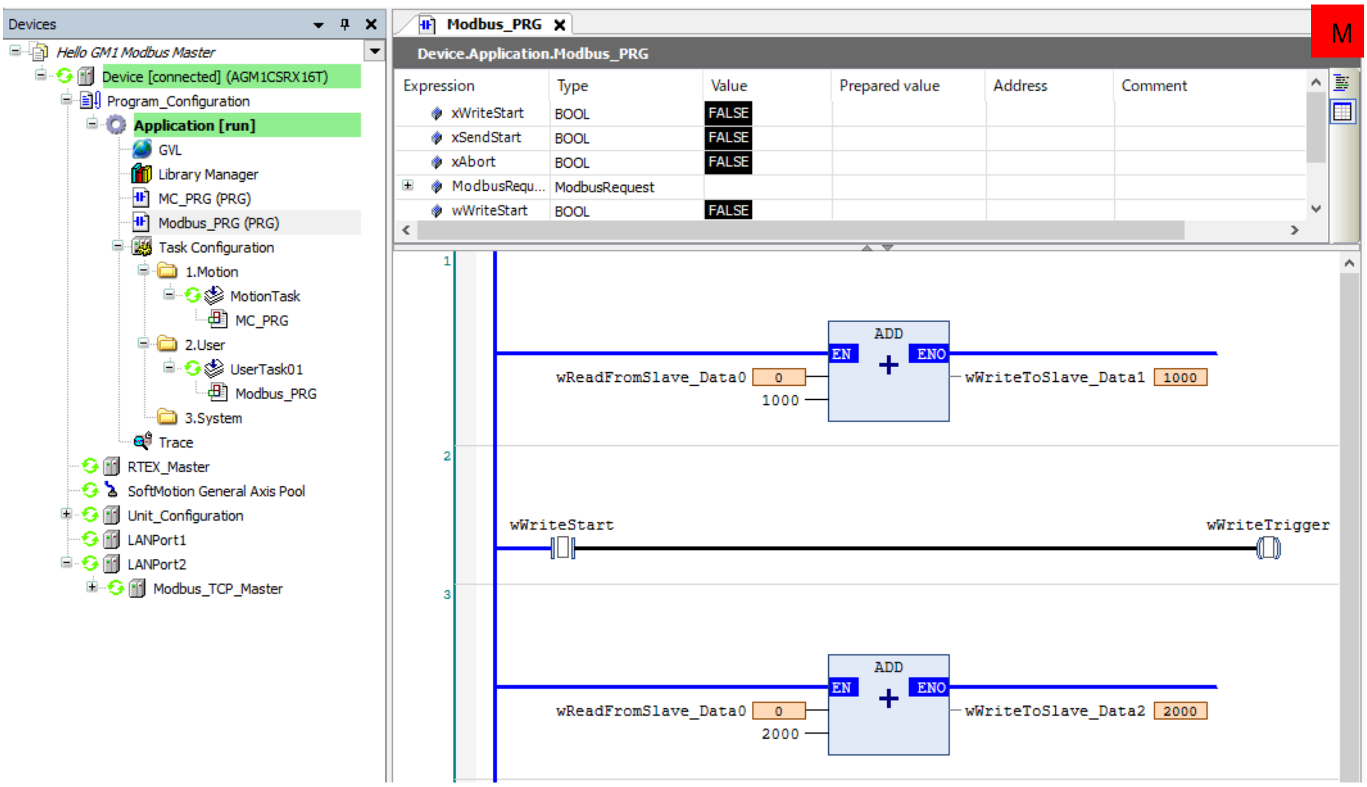
4 Communication Operation Check

Step 1

Check that the master GM1 controller and slave GM1 controller are in RUN state and GM Programmer on each GM1 controller is in a login state.

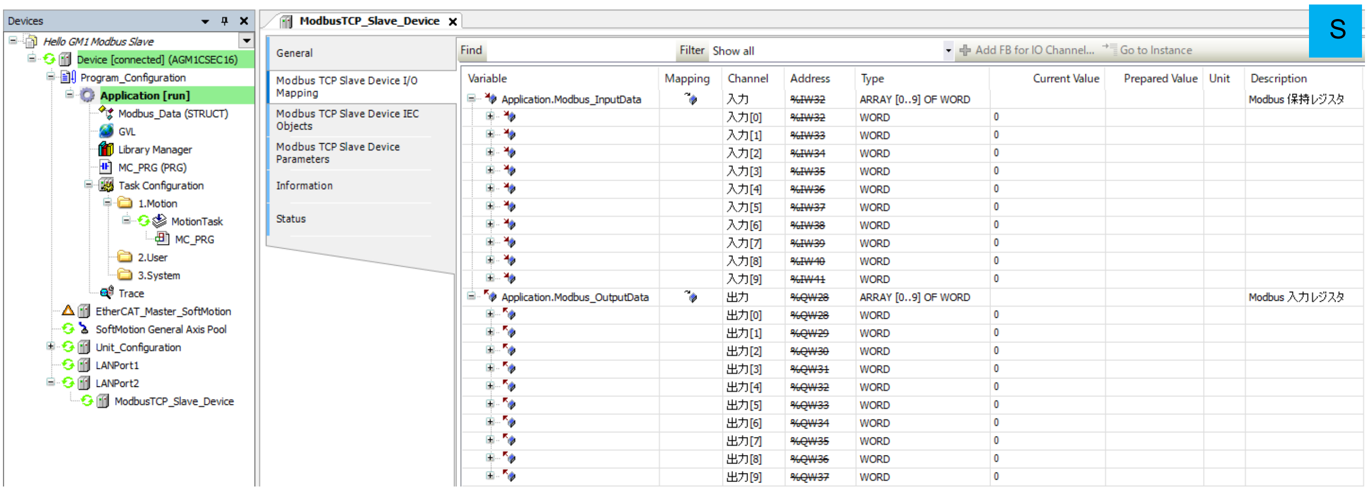
In GM Programmer on the master device, open **Modbus_PRG** in POU. In GM Programmer on the slave device, open **Modbus TCP Slave Device I/O Mapping** in the **ModbusTCP_Slave_Device** pane.

GM Programmer on the master device



Expression	Type	Value	Prepared value	Address	Comment
xWriteStart	BOOL	FALSE			
xSendStart	BOOL	FALSE			
xAbort	BOOL	FALSE			
ModbusRequ...	ModbusRequest				
wWriteStart	BOOL	FALSE			

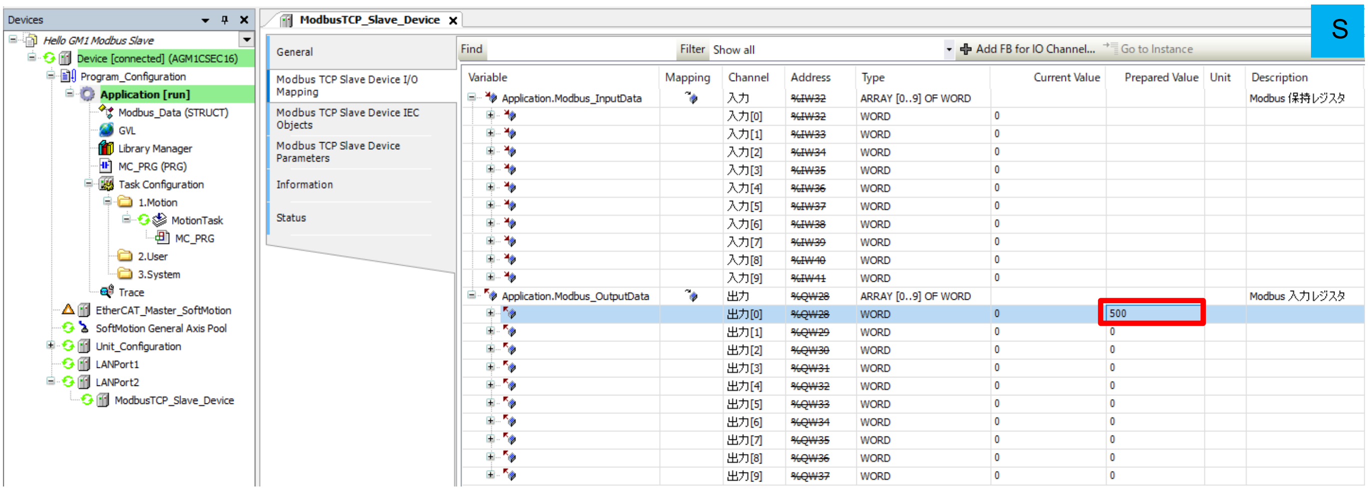
GM Programmer on the slave device



Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
Application.Modbus_InputData		入力	~~%IW32~~	ARRAY [0..9] OF WORD				Modbus 保持レジスタ
		入力[0]	~~%IW32~~	WORD	0			
		入力[1]	~~%IW33~~	WORD	0			
		入力[2]	~~%IW34~~	WORD	0			
		入力[3]	~~%IW35~~	WORD	0			
		入力[4]	~~%IW36~~	WORD	0			
		入力[5]	~~%IW37~~	WORD	0			
		入力[6]	~~%IW38~~	WORD	0			
		入力[7]	~~%IW39~~	WORD	0			
		入力[8]	~~%IW40~~	WORD	0			
		入力[9]	~~%IW41~~	WORD	0			
Application.Modbus_OutputData		出力	~~%QW28~~	ARRAY [0..9] OF WORD				Modbus 入力レジスタ
		出力[0]	~~%QW28~~	WORD	0			
		出力[1]	~~%QW29~~	WORD	0			
		出力[2]	~~%QW30~~	WORD	0			
		出力[3]	~~%QW31~~	WORD	0			
		出力[4]	~~%QW32~~	WORD	0			
		出力[5]	~~%QW33~~	WORD	0			
		出力[6]	~~%QW34~~	WORD	0			
		出力[7]	~~%QW35~~	WORD	0			
		出力[8]	~~%QW36~~	WORD	0			
		出力[9]	~~%QW37~~	WORD	0			

Step 2

In GM Programmer on the slave device, enter "500" in the **Prepared Value** column corresponding to **Output [0]** in the **Modbus TCP Slave Device I/O Mapping** tab and then press Ctr+F7.

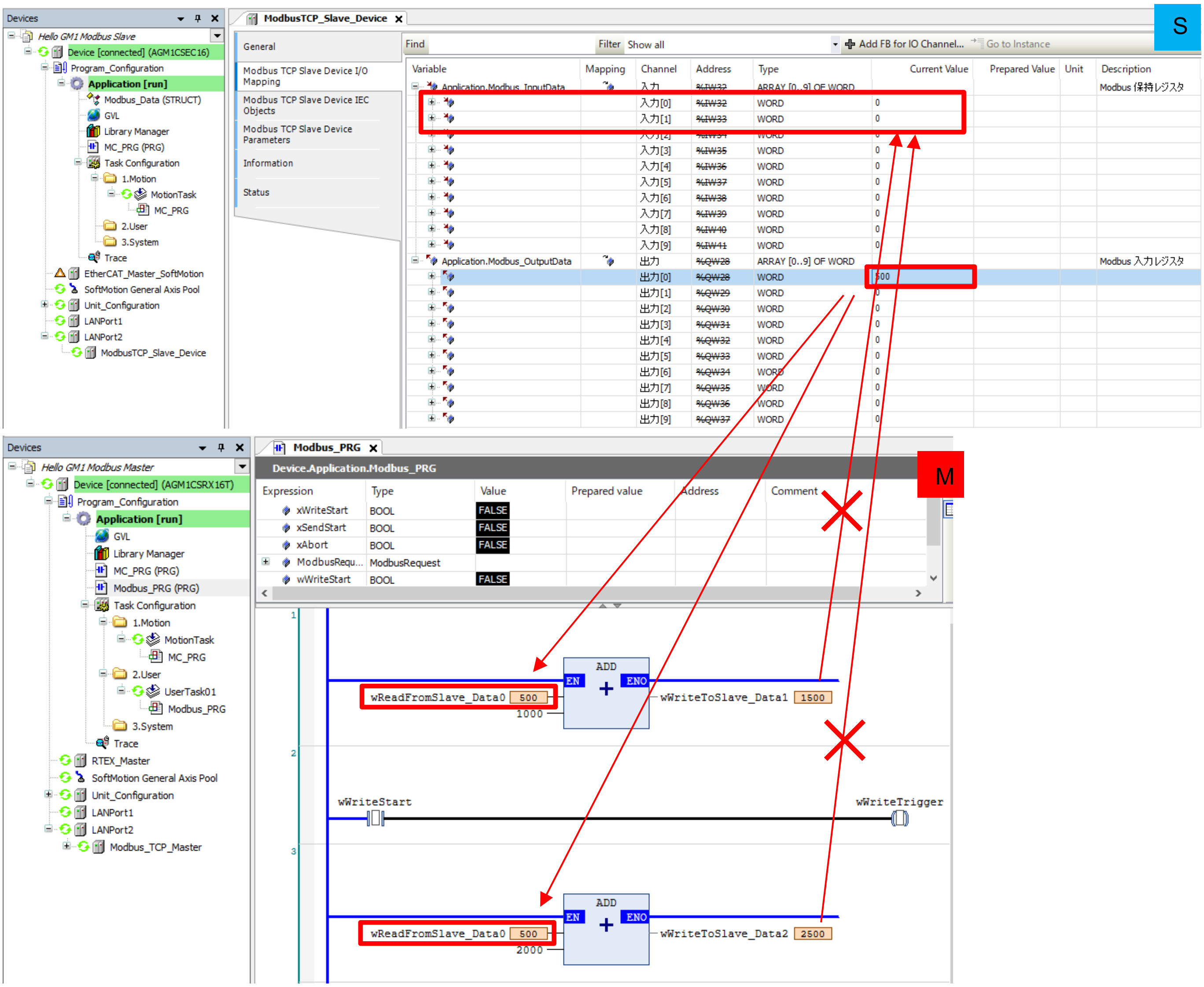


Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
Application.Modbus_InputData		入力	~~%IW32~~	ARRAY [0..9] OF WORD				Modbus 保持レジスタ
		入力[0]	~~%IW32~~	WORD	0			
		入力[1]	~~%IW33~~	WORD	0			
		入力[2]	~~%IW34~~	WORD	0			
		入力[3]	~~%IW35~~	WORD	0			
		入力[4]	~~%IW36~~	WORD	0			
		入力[5]	~~%IW37~~	WORD	0			
		入力[6]	~~%IW38~~	WORD	0			
		入力[7]	~~%IW39~~	WORD	0			
		入力[8]	~~%IW40~~	WORD	0			
		入力[9]	~~%IW41~~	WORD	0			
Application.Modbus_OutputData		出力	~~%QW28~~	ARRAY [0..9] OF WORD				Modbus 入力レジスタ
		出力[0]	~~%QW28~~	WORD	0	500		
		出力[1]	~~%QW29~~	WORD	0	0		
		出力[2]	~~%QW30~~	WORD	0	0		
		出力[3]	~~%QW31~~	WORD	0	0		
		出力[4]	~~%QW32~~	WORD	0	0		
		出力[5]	~~%QW33~~	WORD	0	0		
		出力[6]	~~%QW34~~	WORD	0	0		
		出力[7]	~~%QW35~~	WORD	0	0		
		出力[8]	~~%QW36~~	WORD	0	0		
		出力[9]	~~%QW37~~	WORD	0	0		

Step 3

As soon as the value is written to the **Current Value** column, the value of Slave_Data0 on the slave GM1 controller is written to wReadFromSlave_Data0.

However, the respective values of wWriteToSlave_Data1 and wWriteToSlave_Data2 are not written to Slave_Data1 and Slave_Data2.



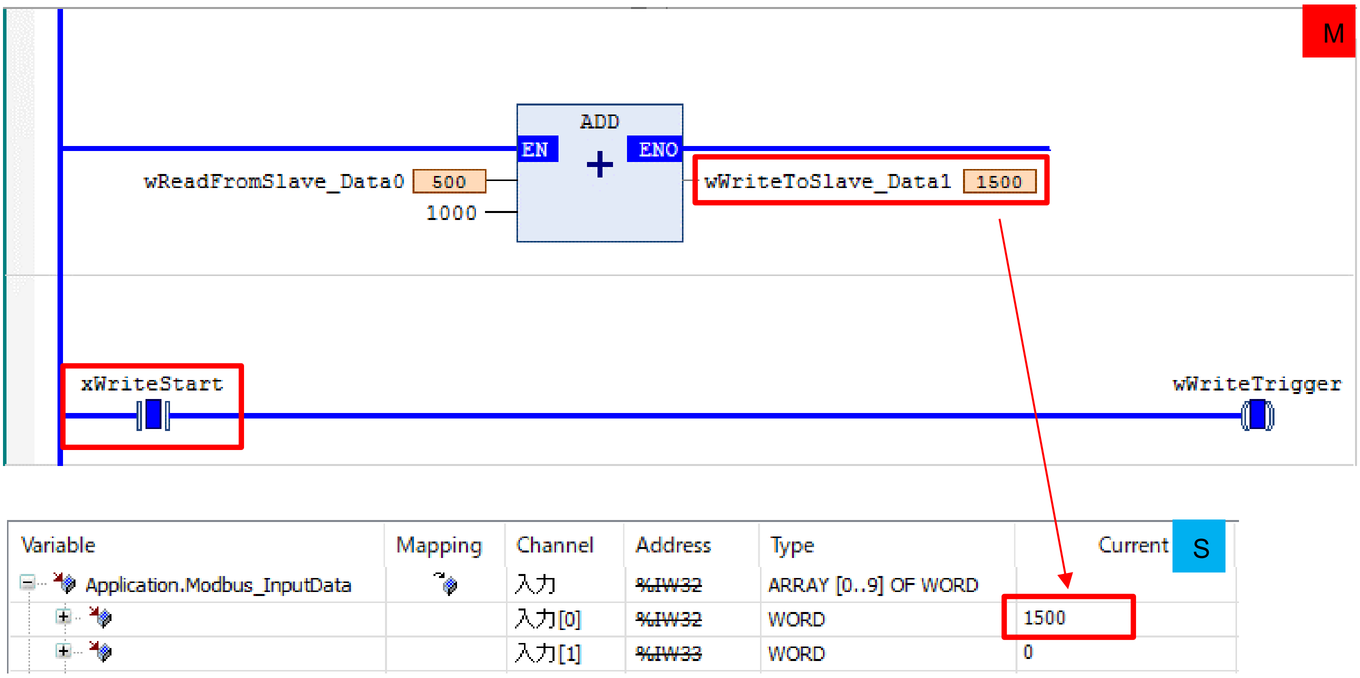
Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
Application.Modbus_InputData		入力	~~%IW32~~	ARRAY [0..9] OF WORD				Modbus 保持レジスタ
		入力[0]	~~%IW32~~	WORD	0			
		入力[1]	~~%IW33~~	WORD	0			
		入力[3]	~~%IW35~~	WORD	0			
		入力[4]	~~%IW36~~	WORD	0			
		入力[5]	~~%IW37~~	WORD	0			
		入力[6]	~~%IW38~~	WORD	0			
		入力[7]	~~%IW39~~	WORD	0			
		入力[8]	~~%IW40~~	WORD	0			
		入力[9]	~~%IW41~~	WORD	0			
Application.Modbus_OutputData		出力	~~%QW28~~	ARRAY [0..9] OF WORD				Modbus 入力レジスタ
		出力[0]	~~%QW28~~	WORD	500			
		出力[1]	~~%QW29~~	WORD	0			
		出力[2]	~~%QW30~~	WORD	0			
		出力[3]	~~%QW31~~	WORD	0			
		出力[4]	~~%QW32~~	WORD	0			
		出力[5]	~~%QW33~~	WORD	0			
		出力[6]	~~%QW34~~	WORD	0			
		出力[7]	~~%QW35~~	WORD	0			
		出力[8]	~~%QW36~~	WORD	0			
		出力[9]	~~%QW37~~	WORD	0			

Expression	Type	Value	Prepared value	Address	Comment
xWriteStart	BOOL	FALSE			
xSendStart	BOOL	FALSE			
xAbort	BOOL	FALSE			
ModbusRequ...	ModbusRequest				
wWriteStart	BOOL	FALSE			

Reading of Slave_Data0 is set to "Cyclic" and automatic reading is performed every 100 ms (specified in "Cycle time"), but writing is set to "Rising edge" or "Application". Therefore, writing is not performed.

Step 4

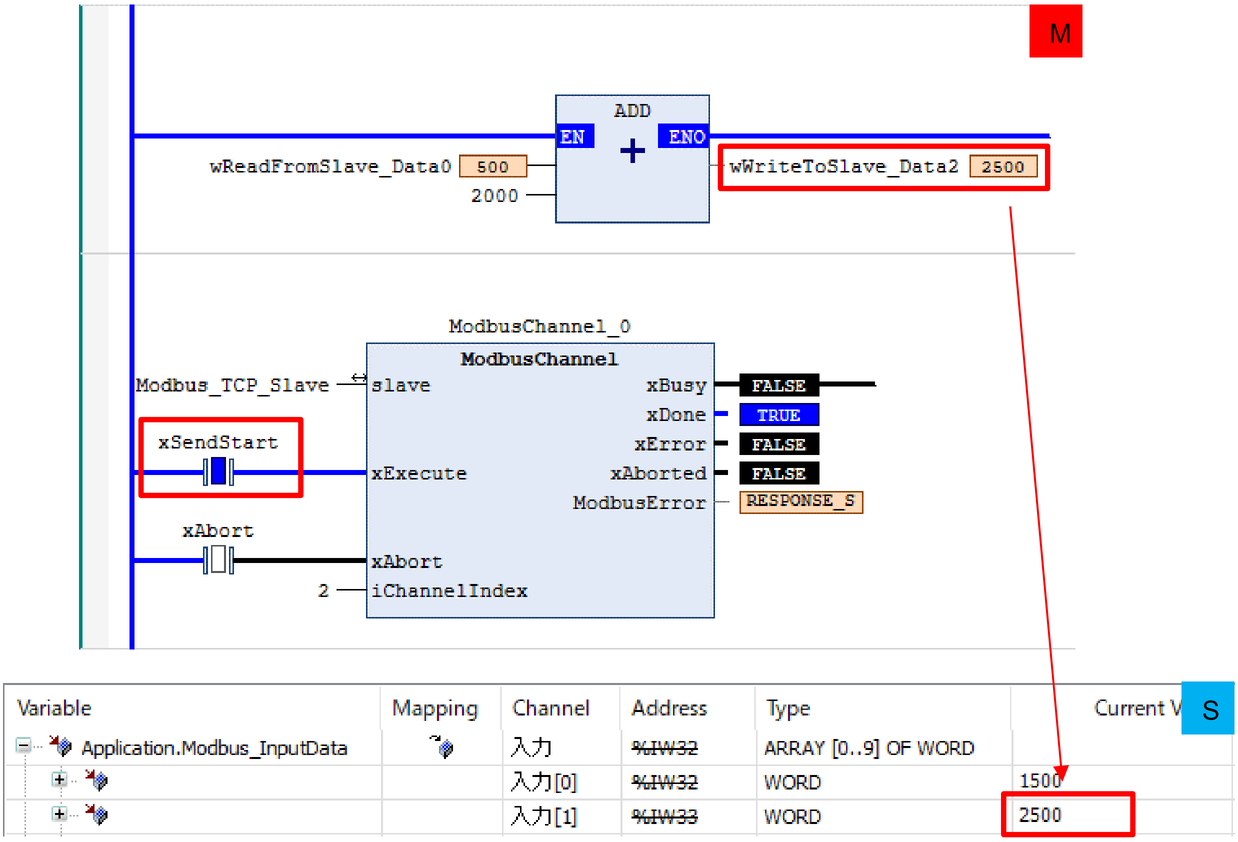
By setting "xWriteStart" to TRUE, "xWriteTrigger" is set to TRUE ("Rising edge"), and the conditions of writing to Slave_Data1 are satisfied and the value of "wWriteToSlave_Data1" is written.



Variable	Mapping	Channel	Address	Type	Current
Application.Modbus_InputData		入力	~~%IW32~~	ARRAY [0..9] OF WORD	
		入力[0]	~~%IW32~~	WORD	1500
		入力[1]	~~%IW33~~	WORD	0

Step 5

By setting "xSendStart" to TRUE, "ModbusChannel_0" is executed and the value of "wWriteToSlave_Data2" is written to Slave_Data2.



Variable	Mapping	Channel	Address	Type	Current V
Application.Modbus_InputData		入力	~~%IW32~~	ARRAY [0..9] OF WORD	
		入力[0]	~~%IW32~~	WORD	1500
		入力[1]	~~%IW33~~	WORD	2500

INFO

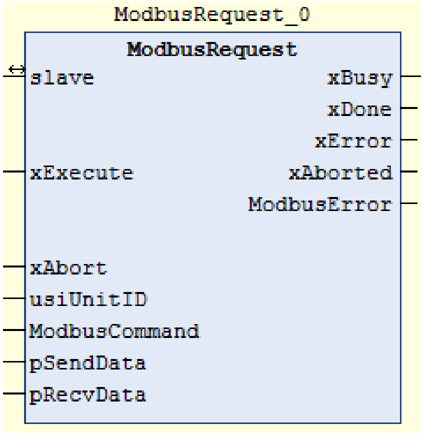
The "ModbusChannnel" instruction is equipped with "xBusy" and "xDone" for output.
While the instruction is being executed, "xBusy" is set to TRUE.
When the instruction finishes executing, “xBusy" is set to FALSE and "xDone" is set to TRUE.

The "ModbusChannnel" instruction is also executed at the rising edge of "xExecute". Therefore, when performing write operation again, set "xSendStart" to FALSE and then back to TRUE.

Column (7): ModbusRequest function block

This is a function block that processes the Modbus command specified by I/O without using the Modbus_TCP_Slave device.

Supported commands	
Command 1	Reads multi-point coil state
Command 2	Reads multi-point input state
Command 3	Reads from multi-point holding registers
Command 4	Reads from multi-point input registers
Command 5	Writes to single-point coil
Command 6	Writes to single-point holding register
Command 15	Writes to multi-point coils
Command 16	Writes to multi-point holding registers
Command 23	Reads from or writes to multi-point holding registers



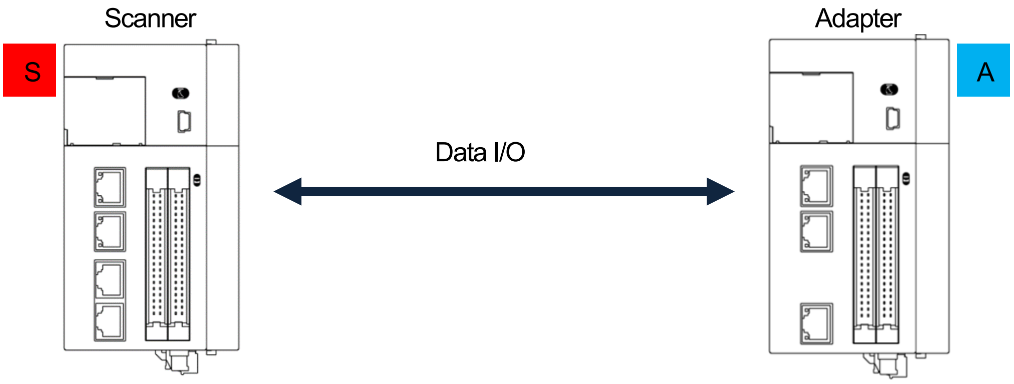
Type	Parameter name	Type	Description
Input	Slave	ModbusTCPSlave	Handle of the Modbus_TCP_Slave device
	xExecute	BOOL	Starts sending commands at the rising edge
	xAbort	BOOL	TRUE: Stops execution and resets all outputs
	usiUnitID	USINT	Slave address: 1 to 247
	ModbusCommand	ModbusCommand	Structure that stores parameters of the commands to be issued
	pSendData	POINTER TO BYTE	Pointer to the sent data buffer
	pRecvData	POINTER TO BYTE	Pointer to the received data buffer
Output	xBusy	BOOL	TRUE: Processing of the FB is incomplete
	xDone	BOOL	TRUE: Processing is completed
	xError	BOOL	TRUE: Error has occurred within FB
	xAborted	BOOL	TRUE: Execution is stopped by the user’s xAbort input
	ModbusError	BYTE	An error code is output.

Ethernet Communication: EtherNet/IP

1 Basic Setup

1.1 Behavior Overview

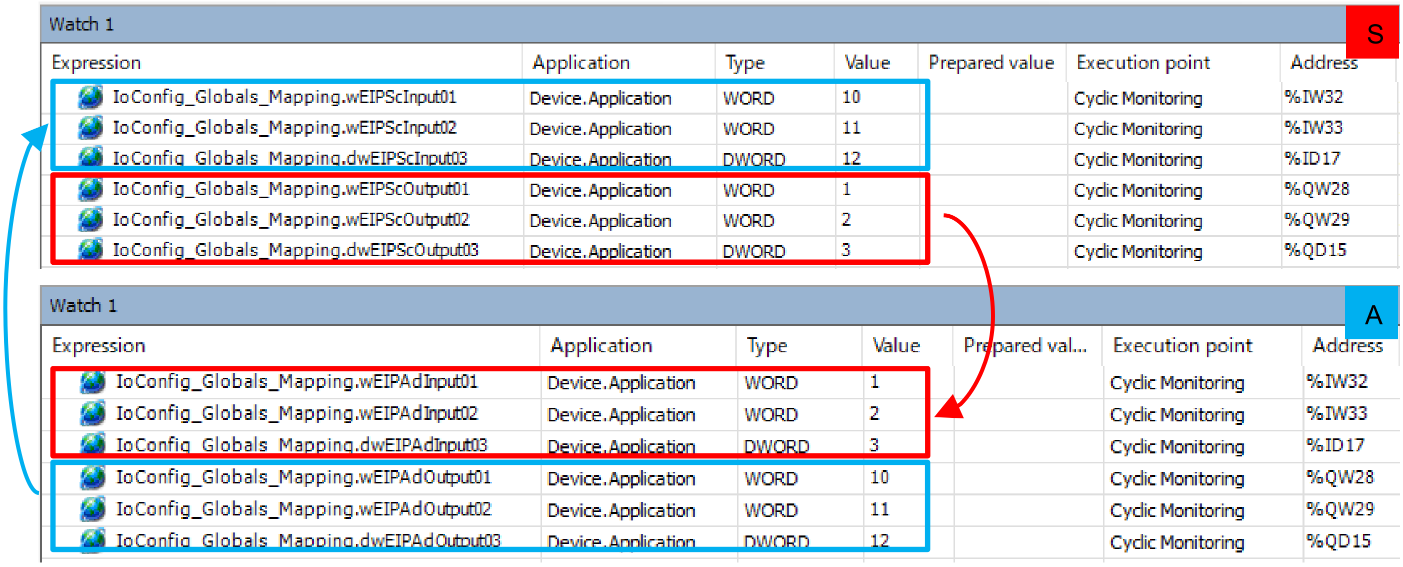
EtherNet/IP
Two GM1 controllers are used as a scanner and adapter.
LAN Port 2 on each device is used to perform EtherNet/IP communication.



* Because two copies of GM Programmer are installed on the scanner and adapter, to identify each copy easily, S or A is shown at the top right corner of the GM Programmer window on the scanner or adapter device, as defined below.

S : Scanner
A : Adapter

Transmission of scanner output to adapter input
Transmission of adapter output to scanner input



Watch 1 S

Expression	Application	Type	Value	Prepared value	Execution point	Address
IoConfig_Globals_Mapping.wEIPScInput01	Device.Application	WORD	10		Cyclic Monitoring	%IW32
IoConfig_Globals_Mapping.wEIPScInput02	Device.Application	WORD	11		Cyclic Monitoring	%IW33
IoConfig_Globals_Mapping.dwEIPScInput03	Device.Application	DWORD	12		Cyclic Monitoring	%ID17
IoConfig_Globals_Mapping.wEIPScOutput01	Device.Application	WORD	1		Cyclic Monitoring	%QW28
IoConfig_Globals_Mapping.wEIPScOutput02	Device.Application	WORD	2		Cyclic Monitoring	%QW29
IoConfig_Globals_Mapping.dwEIPScOutput03	Device.Application	DWORD	3		Cyclic Monitoring	%QD15

Watch 1 A

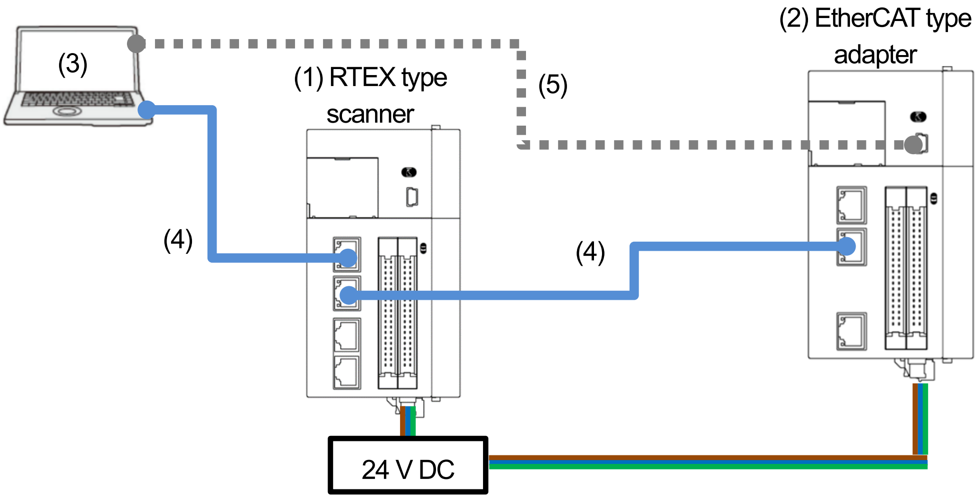
Expression	Application	Type	Value	Prepared val...	Execution point	Address
IoConfig_Globals_Mapping.wEIPAdInput01	Device.Application	WORD	1		Cyclic Monitoring	%IW32
IoConfig_Globals_Mapping.wEIPAdInput02	Device.Application	WORD	2		Cyclic Monitoring	%IW33
IoConfig_Globals_Mapping.dwEIPAdInput03	Device.Application	DWORD	3		Cyclic Monitoring	%ID17
IoConfig_Globals_Mapping.wEIPAdOutput01	Device.Application	WORD	10		Cyclic Monitoring	%QW28
IoConfig_Globals_Mapping.wEIPAdOutput02	Device.Application	WORD	11		Cyclic Monitoring	%QW29
IoConfig_Globals_Mapping.dwEIPAdOutput03	Device.Application	DWORD	12		Cyclic Monitoring	%QD15

1.2 Preparing and Wiring the Required Devices

Prepare the following devices.

No.	Name	
(1)	GM1 controller (RTEX type) x 1: Scanner	(In this textbook, one RTEX type GM1 controller and one EtherCAT type GM1 controller are used.)
(2)	GM1 controller (EtherCAT type) x 1: Adapter	
(3)	PC (with GM Programmer installed)	
(4)	LAN cable: x 2	
(5)	USB cable (Mini-B)	

Wire each device as shown below.

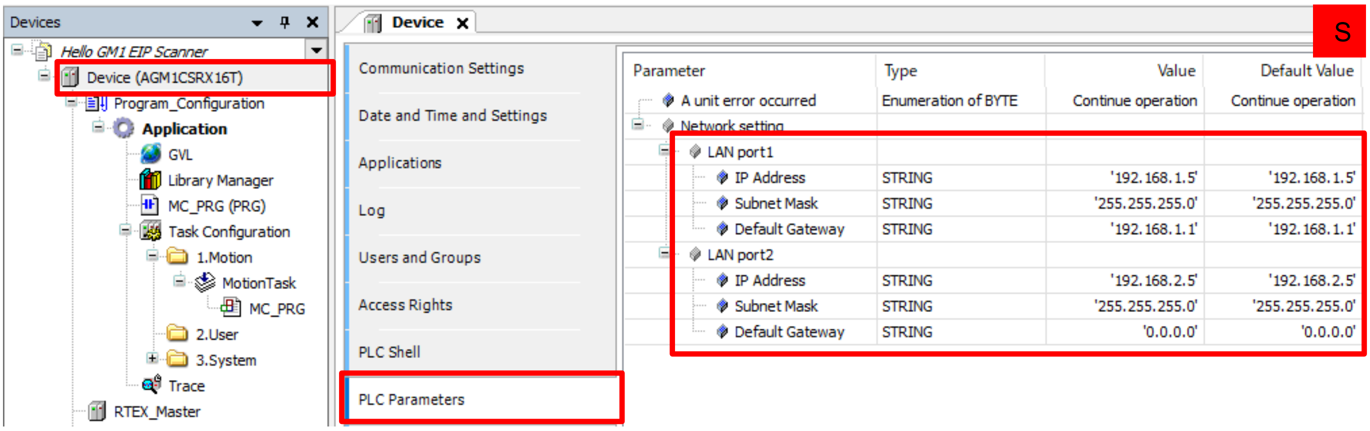


* Because EtherNet/IP requires an EDS file, download it from the following Panasonic web page.
https://www3.panasonic.biz/ac/j/motor/motion-controller/mc/gm1/index.jsp#Software

1.3 RTEX Type: Scanner IP Address Setting to Network Scanning

Step 1

Open GM Programmer and double-click **Device**.
Select **PLC Parameters** and check the respective IP addresses of LAN port 1 and LAN port 2.



Parameter	Type	Value	Default Value
A unit error occurred	Enumeration of BYTE	Continue operation	Continue operation
Network setting			
LAN port1			
IP Address	STRING	'192.168.1.5'	'192.168.1.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'
LAN port2			
IP Address	STRING	'192.168.2.5'	'192.168.2.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'

LAN port 1 (default value)

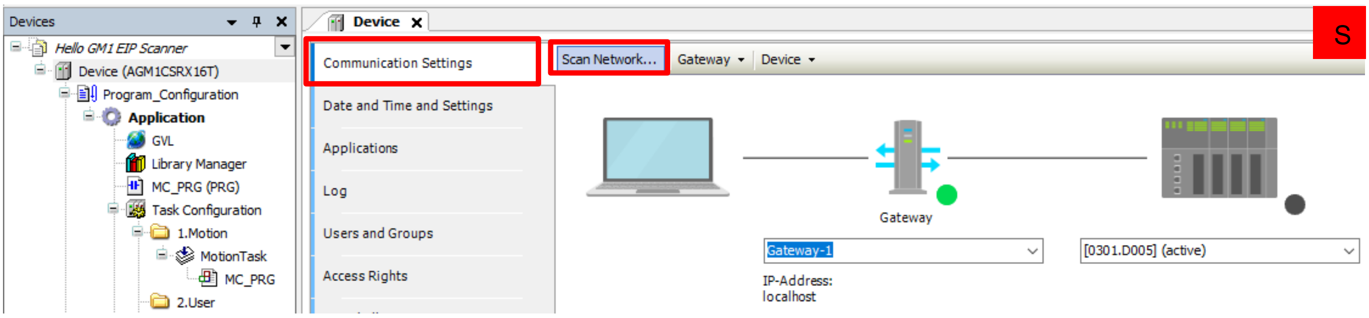
IP address	192.168.1.5
Subnet mask	255.255.255.0
Default gateway	192.168.1.1

LAN port 2 (default value)

IP address	192.168.2.5
Subnet mask	255.255.255.0
Default gateway	0.0.0.0

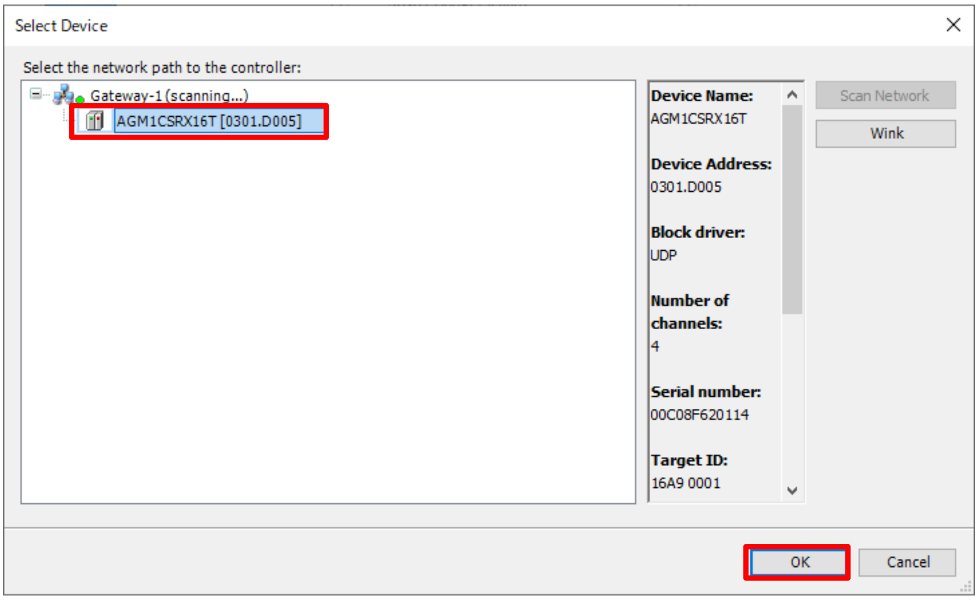
Step 2

Select **Communication Settings** and click **Scan Network**.



Step 3

Select a device to be connected and click **OK**.



Device Name:
AGM1CSRX16T

Device Address:
0301.D005

Block driver:
UDP

Number of channels:
4

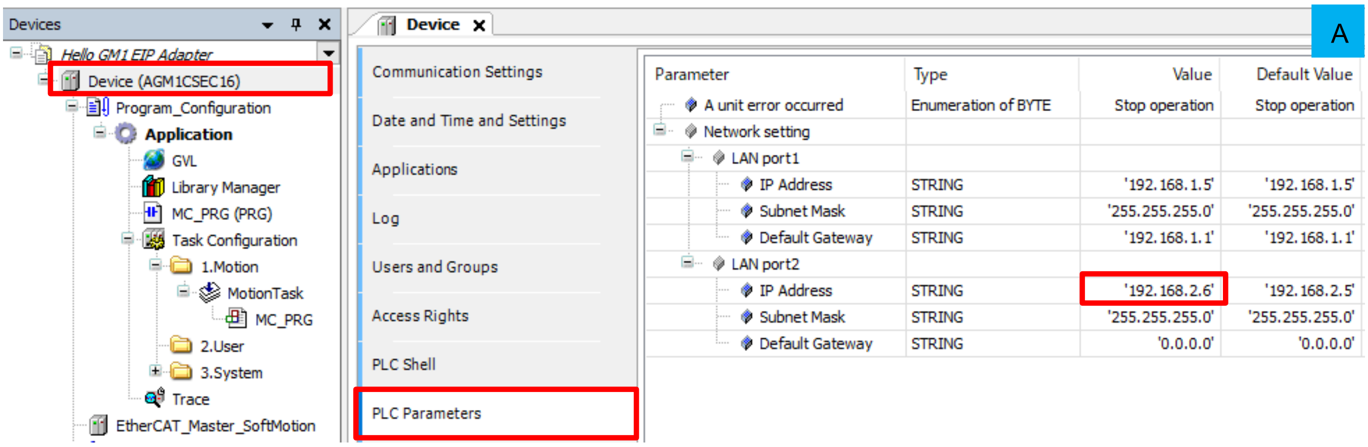
Serial number:
00C08F620114

Target ID:
16A9 0001

1.4 EtherCAT Type: Adapter IP Address Setting to USB Addition

Step 1

Open GM Programmer and double-click **Device**.
Select **PLC Parameters** and change the IP address of LAN port 2 to 192.168.2.6.



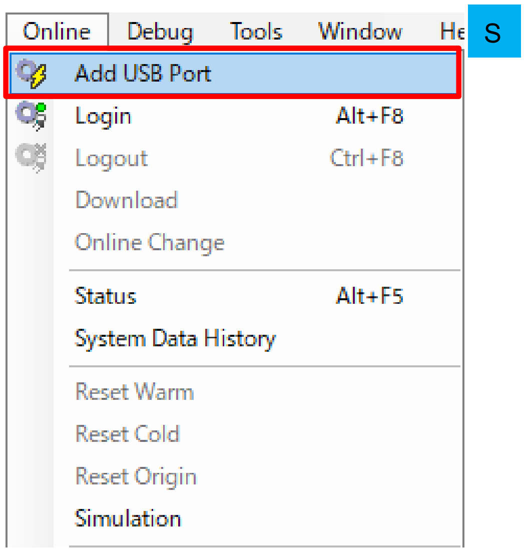
Parameter	Type	Value	Default Value
A unit error occurred	Enumeration of BYTE	Stop operation	Stop operation
Network setting			
LAN port1			
IP Address	STRING	'192.168.1.5'	'192.168.1.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'
LAN port2			
IP Address	STRING	'192.168.2.6'	'192.168.2.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'

LAN port 2

IP address	192.168.2.6
Subnet mask	255.255.255.0
Default gateway	0.0.0.0

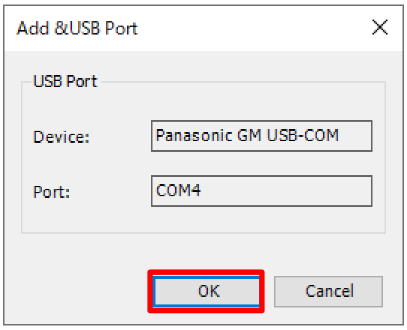
Step 2

Select **Add USB Port** from the **Online** menu.



Step 3

The **Add USB Port** dialog box will be displayed. Check the device and the port to be used and click **OK**.



Add &USB Port

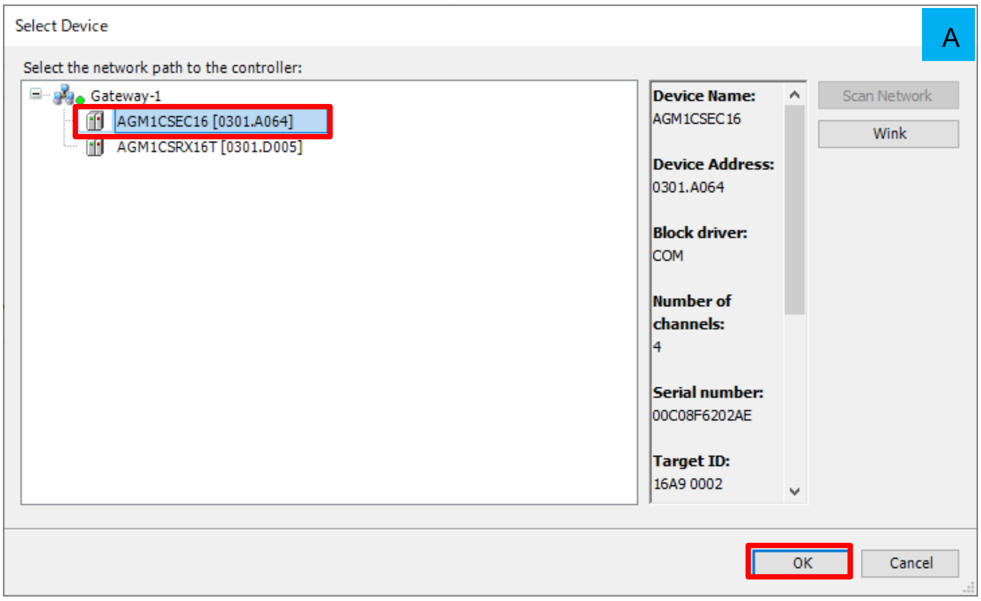
USB Port

Device: Panasonic GM USB-COM

Port: COM4

Step 4

The **Select Device** dialog box will be displayed.
Select a device to be connected and click **OK**.



Select Device

Select the network path to the controller:

- Gateway-1
  - AGM1CSEC16 [0301.A064]
  - AGM1CSRX16T [0301.D005]

Device Name:
AGM1CSEC16

Device Address:
0301.A064

Block driver:
COM

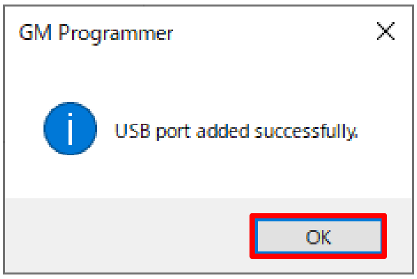
Number of channels:
4

Serial number:
00C08F6202AE

Target ID:
16A9 0002

Step 5

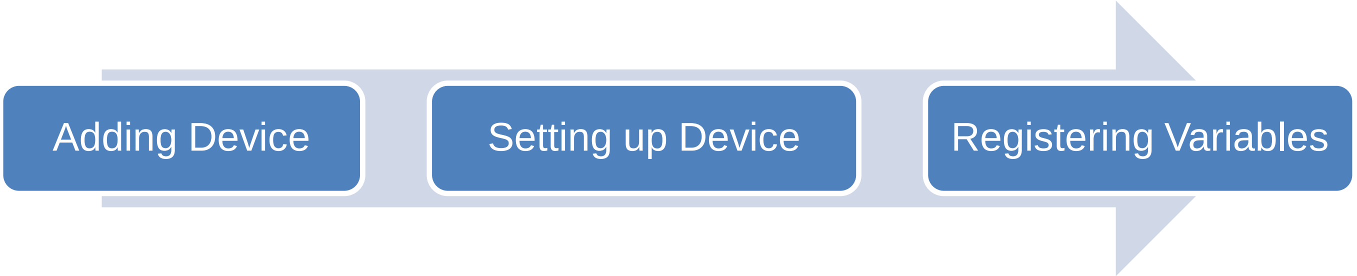
When a connection is established, the USB port is added as a communication interface between the PC and GM1 controller.


GM Programmer

USB port added successfully.

2 Scanner-side Setup

Start setup on the scanner side in the following order.



Adding Device

Setting up Device

Registering Variables

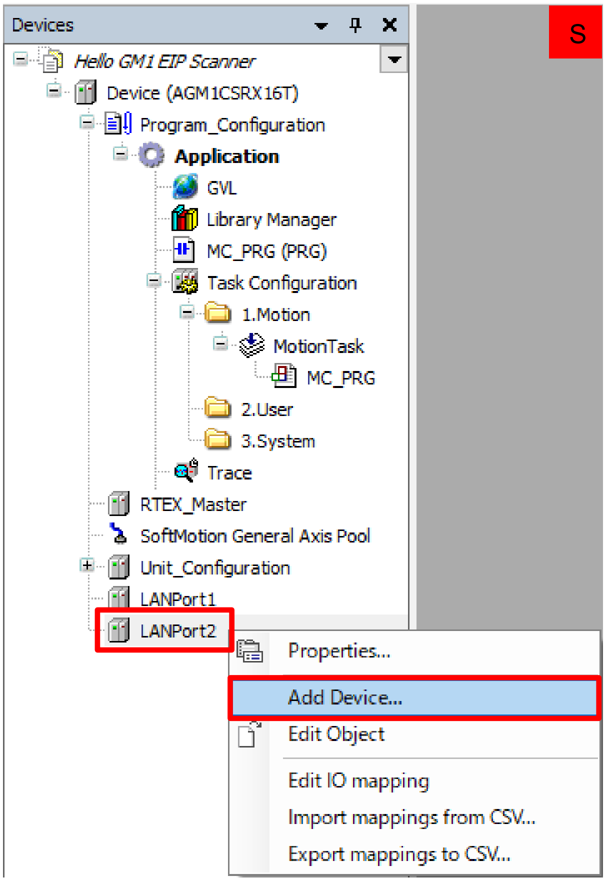
2.1 Adding Device

Register an adapter to be connected.

Step 1

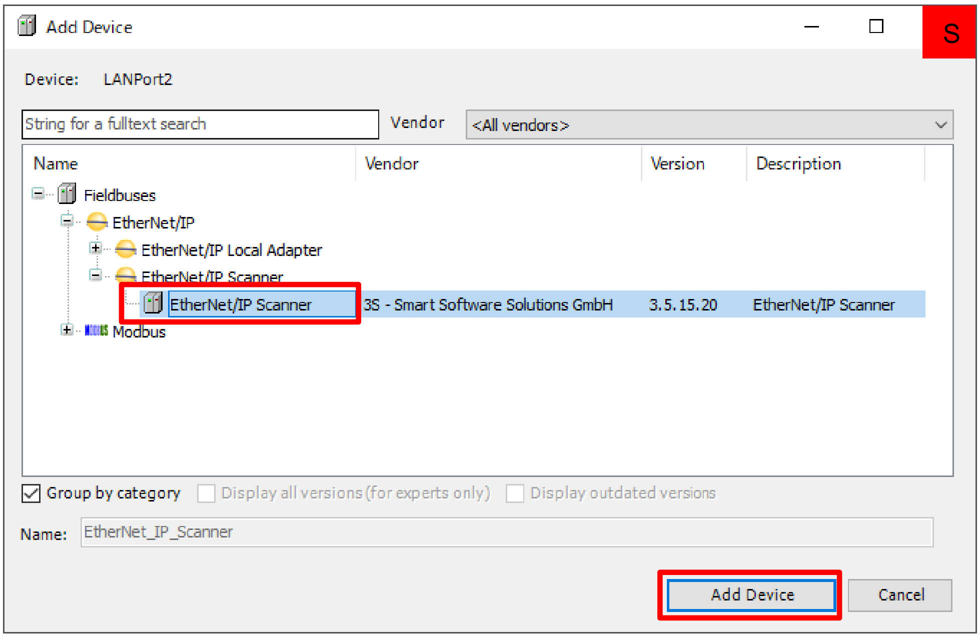
Add an EtherNet/IP scanner device to LAN Port2.

In the navigation pane, right-click **LANPort2** and then select **Add Device**.



Step 2

Select **EtherNet/IP**, **EtherNet/IP Scanner**, and then **EtherNet/IP Scanner** and click **Add Device**.


Add Device

Device: LANPort2

String for a fulltext search

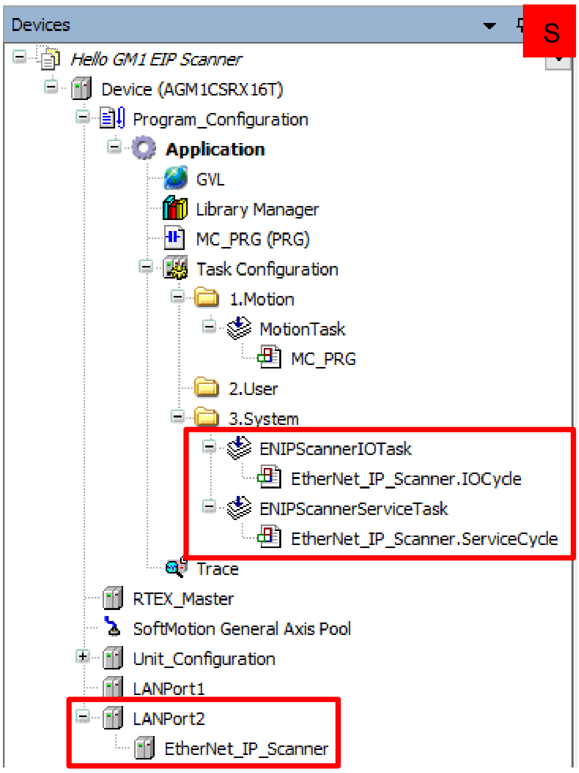
Vendor <All vendors>

Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Local Adapter			
EtherNet/IP Scanner			
EtherNet/IP Scanner	3S - Smart Software Solutions GmbH	3.5.15.20	EtherNet/IP Scanner
Modbus			

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

Name: EtherNet_IP_Scanner

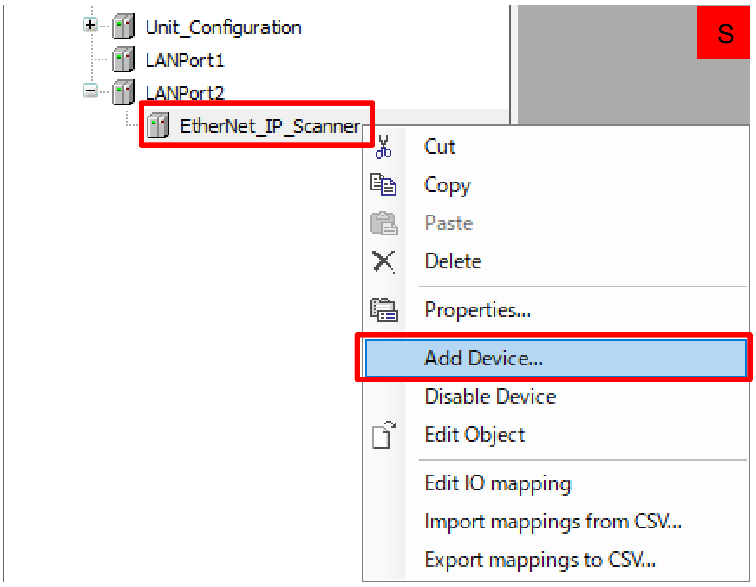
ENIPScannerIOTask and **ENIPScannerServiceTask** will be added to **3.System** in **Task Configuration**.
EtherNet_IP_Scanner will be added to **LANPort2**.


Devices

```
Hello GM1 EIP Scanner
  Device (AGM1CSRX16T)
    Program_Configuration
      Application
        GVL
        Library Manager
        MC_PRG (PRG)
        Task Configuration
          1.Motion
            MotionTask
              MC_PRG
          2.User
          3.System
            ENIPScannerIOTask
              EtherNet_IP_Scanner.IOCycle
            ENIPScannerServiceTask
              EtherNet_IP_Scanner.ServiceCycle
        Trace
    RTEX_Master
    SoftMotion General Axis Pool
    Unit_Configuration
    LANPort1
    LANPort2
      EtherNet_IP_Scanner
```

Step 3

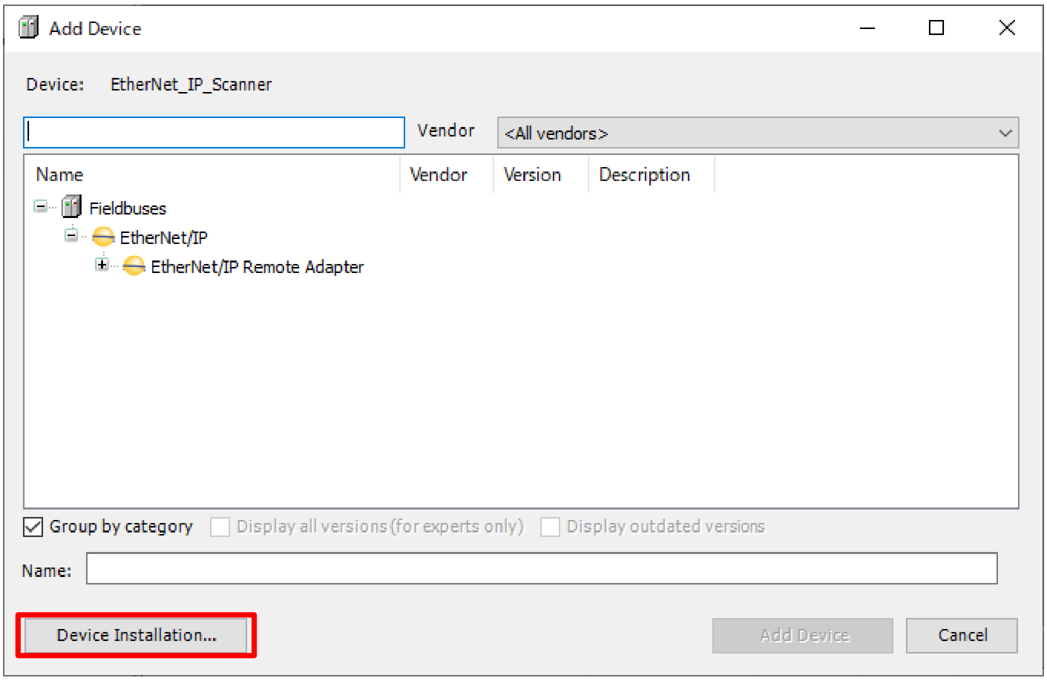
Right-click **EtherNet_IP_Scanner** and select **Add Device**.



Step 4

In the dialog box that is displayed, click **Device Installation**.

Select an EDS file that was downloaded beforehand.


Add Device

Device: EtherNet_IP_Scanner

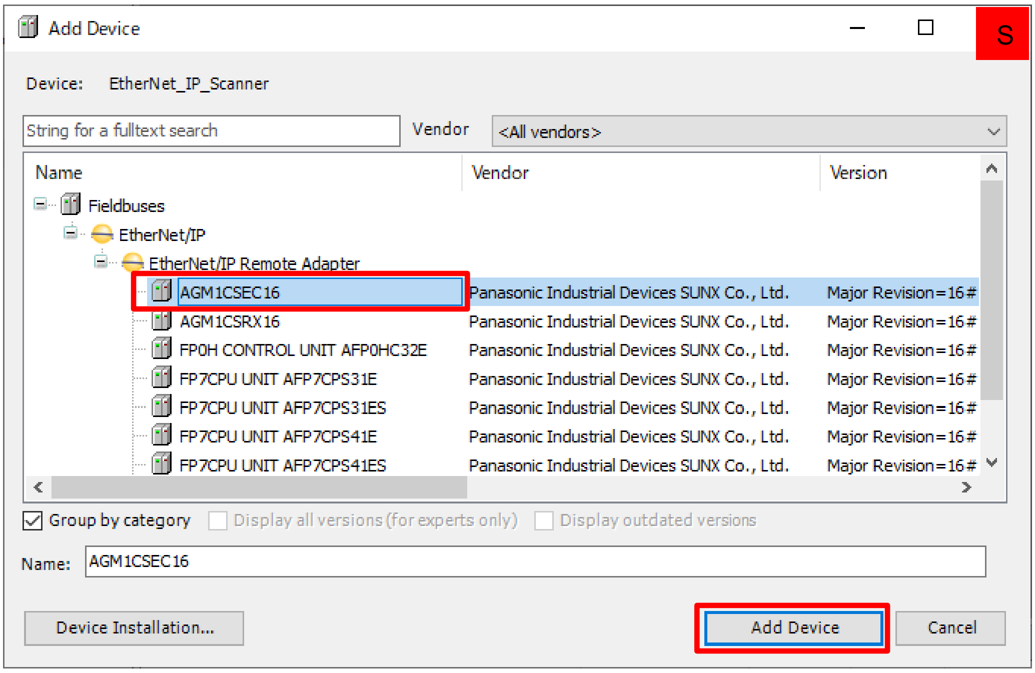
Vendor <All vendors>

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

Name:

Step 5

AGM1CSEC16 and **AGM1CSRX16** will be added. Select **AGM1CSEC16** and then click **Add Device**.



Add Device

Device: EtherNet_IP_Scanner

String for a fulltext search | Vendor: <All vendors>

Name	Vendor	Version
Fieldbuses		
EtherNet/IP		
EtherNet/IP Remote Adapter		
AGM1CSEC16	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#
AGM1CSRX16	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#
FP0H CONTROL UNIT AFP0HC32E	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#
FP7CPU UNIT AFP7CPS31E	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#
FP7CPU UNIT AFP7CPS31ES	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#
FP7CPU UNIT AFP7CPS41E	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#
FP7CPU UNIT AFP7CPS41ES	Panasonic Industrial Devices SUNX Co., Ltd.	Major Revision=16#

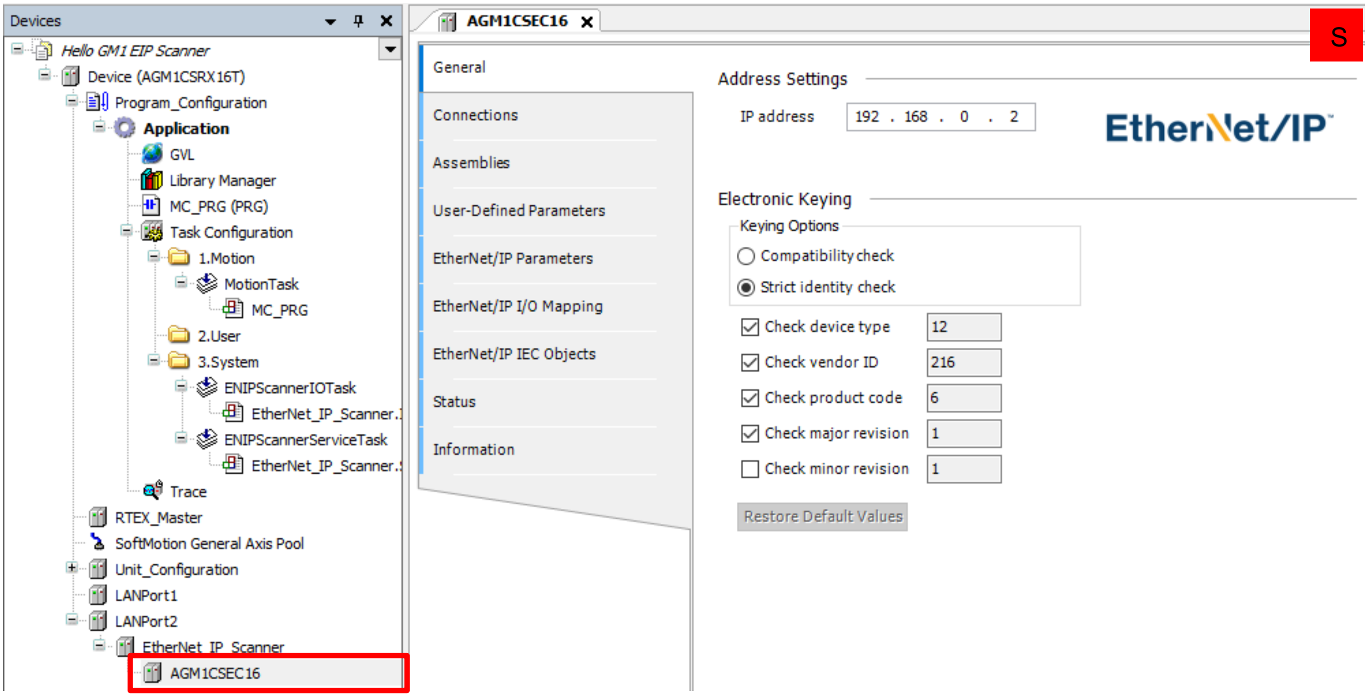
☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: AGM1CSEC16

Device Installation... | Add Device | Cancel

Step 6

AGM1CSEC16 will be added to **EtherNet IP Scanner**.
Double-click **AGM1CSEC16** to open the **AGM1CSEC16** pane.



Devices

- Hello GM1 EIP Scanner
  - Device (AGM1CSRX16T)
    - Program_Configuration
      - **Application**
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Task Configuration
          - 1.Motion
            - MotionTask
              - MC_PRG
          - 2.User
          - 3.System
            - ENIPScannerIOTask
              - EtherNet_IP_Scanner.
            - ENIPScannerServiceTask
              - EtherNet_IP_Scanner.
        - Trace
    - RTEX_Master
    - SoftMotion General Axis Pool
    - Unit_Configuration
    - LANPort1
    - LANPort2
      - EtherNet_IP_Scanner
        - AGM1CSEC16

AGM1CSEC16

General | Connections | Assemblies | User-Defined Parameters | EtherNet/IP Parameters | EtherNet/IP I/O Mapping | EtherNet/IP IEC Objects | Status | Information

Address Settings

IP address: 192 . 168 . 0 . 2

EtherNet/IP

Electronic Keying

Keying Options
○ Compatibility check
● Strict identity check

☑ Check device type: 12
☑ Check vendor ID: 216
☑ Check product code: 6
☑ Check major revision: 1
☐ Check minor revision: 1

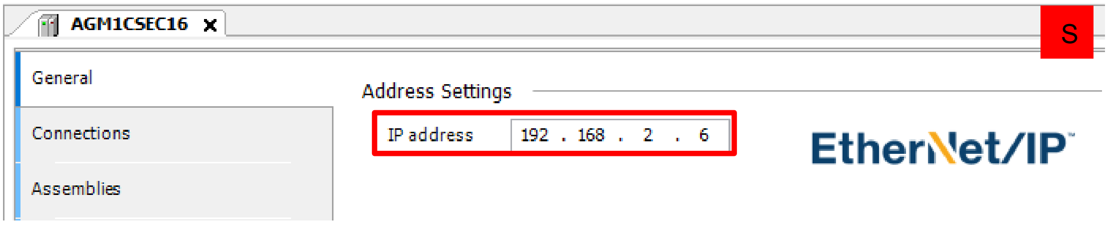
Restore Default Values

2.2 Setting up Device

Step 1

In the **General** tab, set an IP address for the scanner device.

In this textbook, "192.168.2.6" is used as the IP address of the scanner GM1 controller. Therefore, set an IP address as shown below.

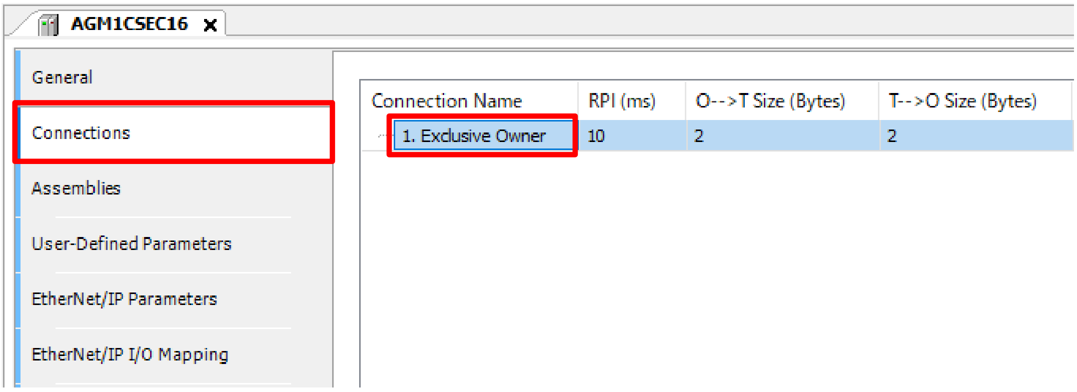


Address Settings

IP address 192 . 168 . 2 . 6

Step 2

Select the **Connections** tab and double-click **1.Exclusive Owner**.

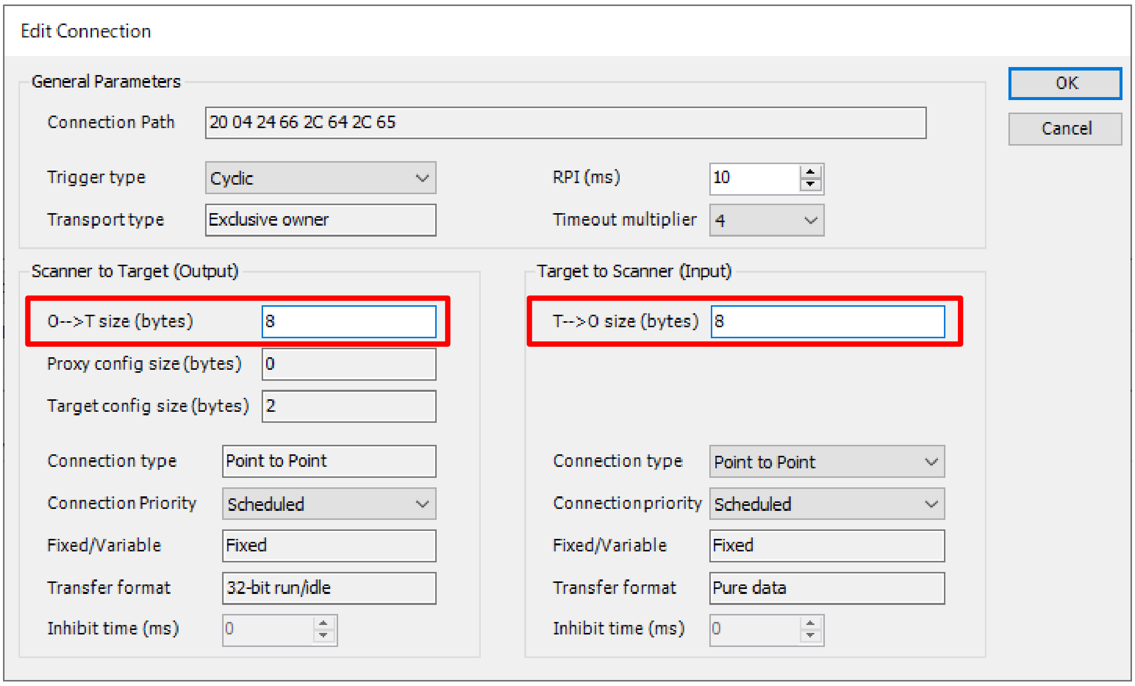


Connection Name	RPI (ms)	O-->T Size (Bytes)	T-->O Size (Bytes)
1. Exclusive Owner	10	2	2

Step 3

The **Edit Connection** dialog box will be displayed. Set values for communication data, as below.

Scanner to Target (Output)	Set value
O —>T size (bytes)	8
Target to Scanner (Input)	Set value
T —> O size (bytes)	8



General Parameters

Connection Path: 20 04 24 66 2C 64 2C 65

Trigger type: Cyclic

RPI (ms): 10

Transport type: Exclusive owner

Timeout multiplier: 4

Scanner to Target (Output)

O-->T size (bytes): 8

Proxy config size (bytes): 0

Target config size (bytes): 2

Connection type: Point to Point

Connection Priority: Scheduled

Fixed/Variable: Fixed

Transfer format: 32-bit run/idle

Inhibit time (ms): 0

Target to Scanner (Input)

T-->O size (bytes): 8

Connection type: Point to Point

Connection priority: Scheduled

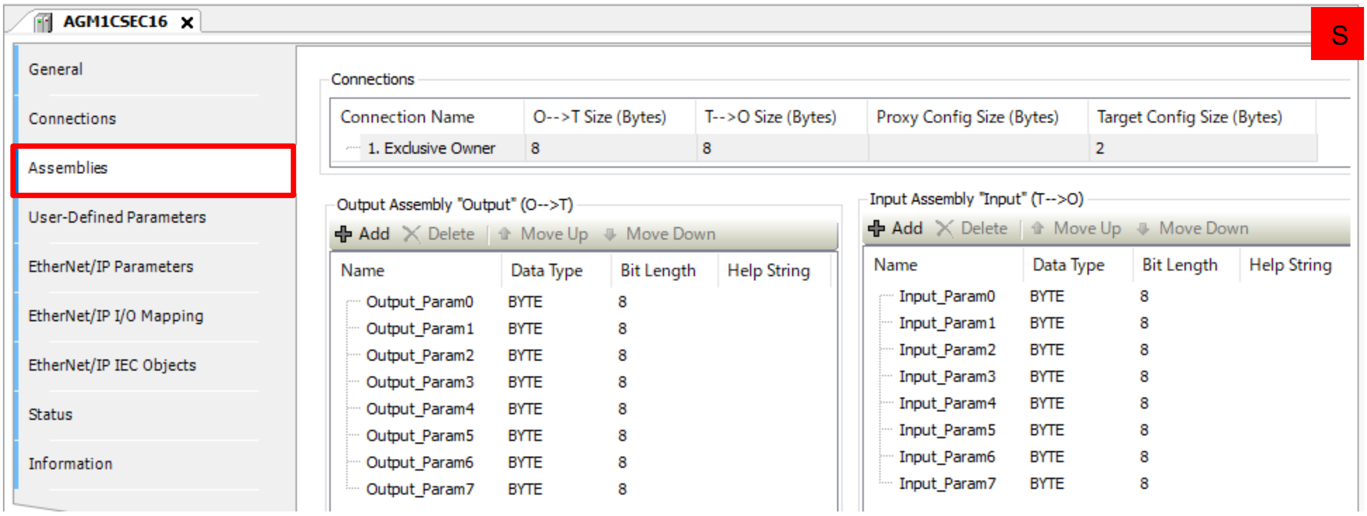
Fixed/Variable: Fixed

Transfer format: Pure data

Inhibit time (ms): 0

Step 4

Select **Assemblies**.



Connections

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)
1. Exclusive Owner	8	8		2

Output Assembly "Output" (O-->T)

Name	Data Type	Bit Length	Help String
Output_Param0	BYTE	8	
Output_Param1	BYTE	8	
Output_Param2	BYTE	8	
Output_Param3	BYTE	8	
Output_Param4	BYTE	8	
Output_Param5	BYTE	8	
Output_Param6	BYTE	8	
Output_Param7	BYTE	8	

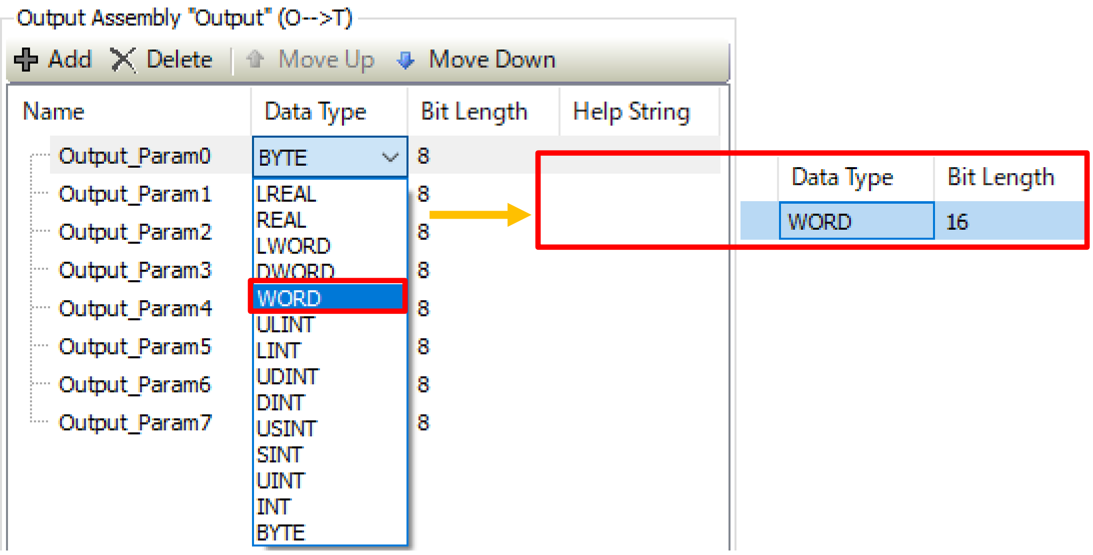
Input Assembly "Input" (T-->O)

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	

In **Output Assembly** and **Input Assembly**, assignment of previously set 8-byte data is set.
By default, each data item is set as 1-byte data.

Step 5

Select the **Data Type** column corresponding to **Output_Param0** and then select **WORD**. The set data type will be changed.



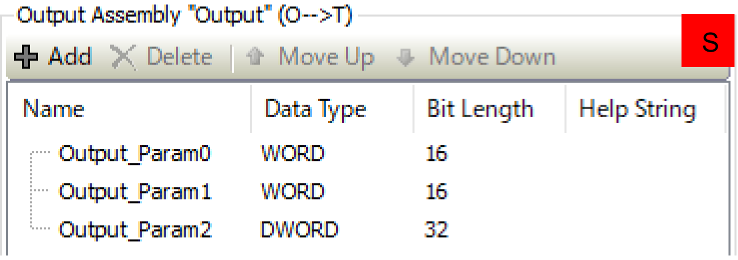
Output Assembly "Output" (O-->T)

Name	Data Type	Bit Length	Help String
Output_Param0	BYTE	8	
Output_Param1		8	
Output_Param2		8	
Output_Param3		8	
Output_Param4		8	
Output_Param5		8	
Output_Param6		8	
Output_Param7		8	

Data Type	Bit Length
WORD	16

Step 6

Change the respective data types of **Output_Param1** and **Output_Param2** to **WORD** and **DWORD** in the same way. Delete **Output_Param3** and all subsequent items as they are unnecessary.

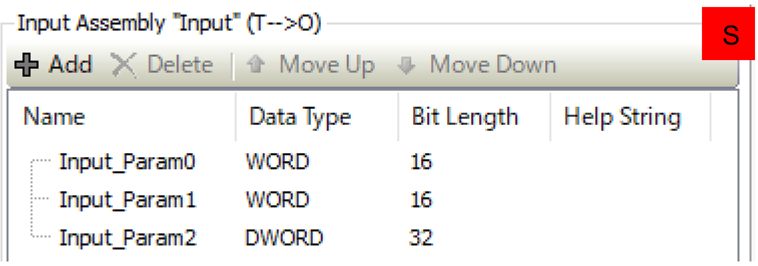


Output Assembly "Output" (O-->T)

Name	Data Type	Bit Length	Help String
Output_Param0	WORD	16	
Output_Param1	WORD	16	
Output_Param2	DWORD	32	

Step 7

Configure settings for **Input Assembly** in the same way as **Output Assembly**.

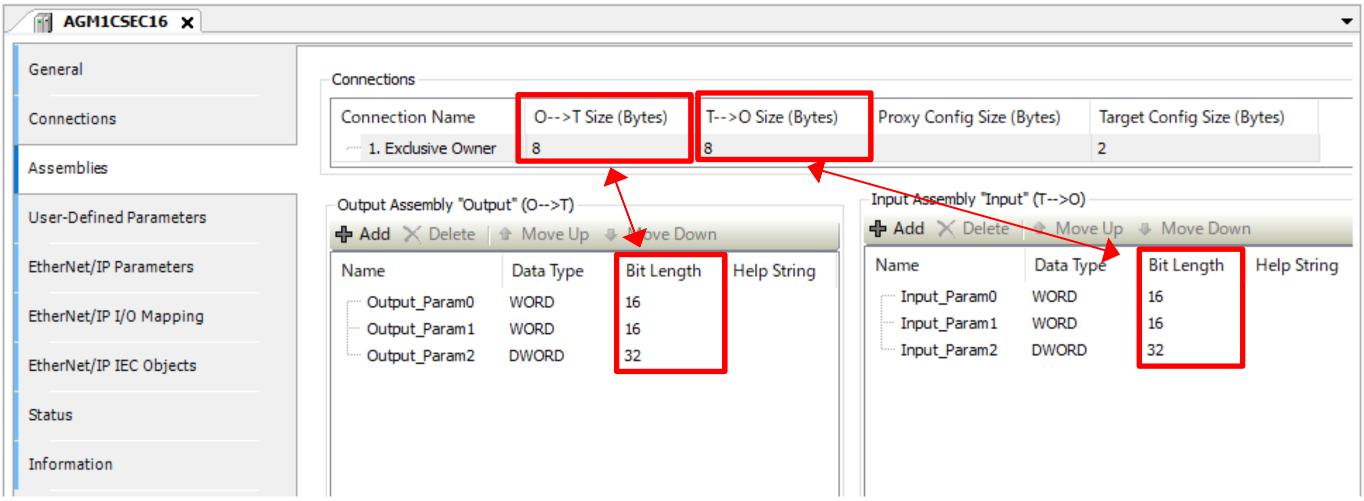


Input Assembly "Input" (T-->O)

Name	Data Type	Bit Length	Help String
Input_Param0	WORD	16	
Input_Param1	WORD	16	
Input_Param2	DWORD	32	

Column

Unless unnecessary data items are deleted, the data size will not become "eight bytes" that was registered beforehand. Because the data size is also set to eight bytes on the scanner GM1 controller, the preset data size will differ between the scanner and adapter, resulting in a communication setting error.



Connections

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)
1. Exclusive Owner	8	8		2

Output Assembly "Output" (O-->T)

Name	Data Type	Bit Length	Help String
Output_Param0	WORD	16	
Output_Param1	WORD	16	
Output_Param2	DWORD	32	

Input Assembly "Input" (T-->O)

Name	Data Type	Bit Length	Help String
Input_Param0	WORD	16	
Input_Param1	WORD	16	
Input_Param2	DWORD	32	

2.3 Registering Variables

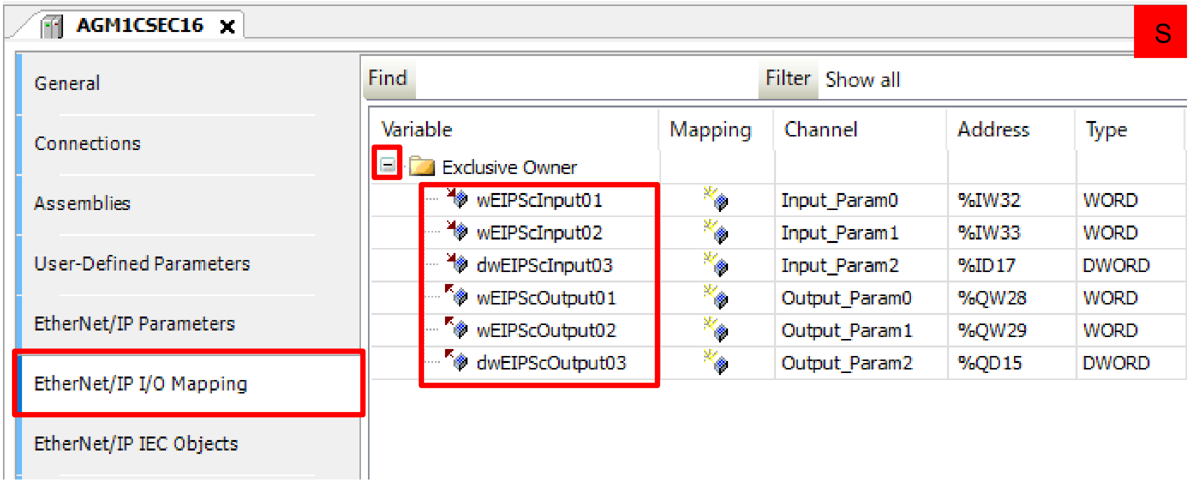
Register variables for each data item that has been set.

Step 1

Select the **EtherNet/IP I/O Mapping** tab.

Expand **Exclusive Owner** and register variables.

Variable	Channel
wEIPScInput01	Input_Param0
wEIPScInput02	Input_Param1
dwEIPScInput03	Input_Param2
wEIPScOutput01	Output_Param0
wEIPScOutput02	Output_Param1
dwEIPScOutput03	Output_Param2

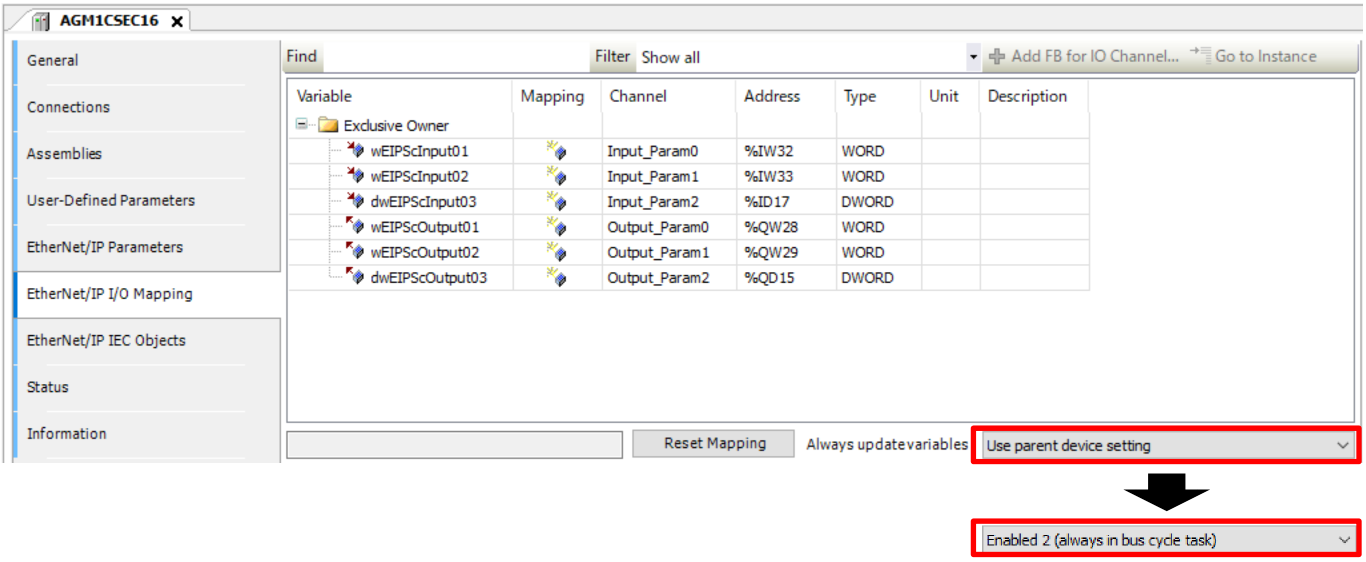


Variable	Mapping	Channel	Address	Type
Exclusive Owner				
wEIPScInput01		Input_Param0	%IW32	WORD
wEIPScInput02		Input_Param1	%IW33	WORD
dwEIPScInput03		Input_Param2	%ID17	DWORD
wEIPScOutput01		Output_Param0	%QW28	WORD
wEIPScOutput02		Output_Param1	%QW29	WORD
dwEIPScOutput03		Output_Param2	%QD15	DWORD

Step 2

Select **Use parent device setting** from the drop-down list on the right side of **Always update variables** and then select **Enable 2 (always in bus cycle task)** from the drop-down list below the above one.

* For detailed settings, refer to the relevant manuals.



Variable	Mapping	Channel	Address	Type	Unit	Description
Exclusive Owner						
wEIPScInput01		Input_Param0	%IW32	WORD		
wEIPScInput02		Input_Param1	%IW33	WORD		
dwEIPScInput03		Input_Param2	%ID17	DWORD		
wEIPScOutput01		Output_Param0	%QW28	WORD		
wEIPScOutput02		Output_Param1	%QW29	WORD		
dwEIPScOutput03		Output_Param2	%QD15	DWORD		

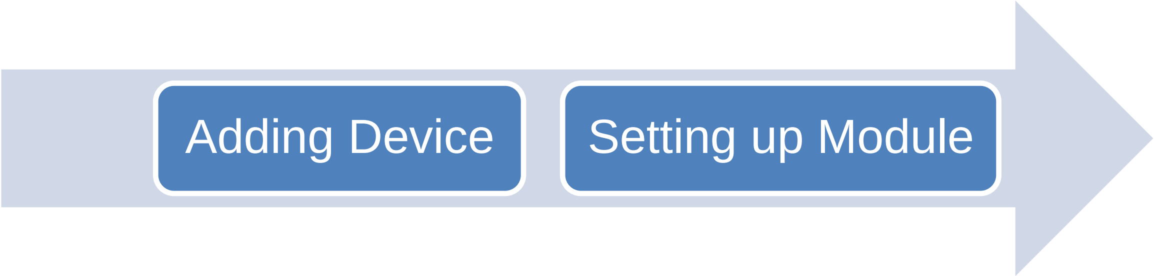
Always update variables: Use parent device setting

Enabled 2 (always in bus cycle task)

This completes setup for the scanner GM1 controller.

3 Adapter-side Setup

Start setup on the adapter side in the following order.



Adding Device

Setting up Module

Column (8): EtherNet/IP adapter function modules

When the GM1 controller is used as an adapter, modules are set for each I/O data item.
Module are selected from the following 10 types.

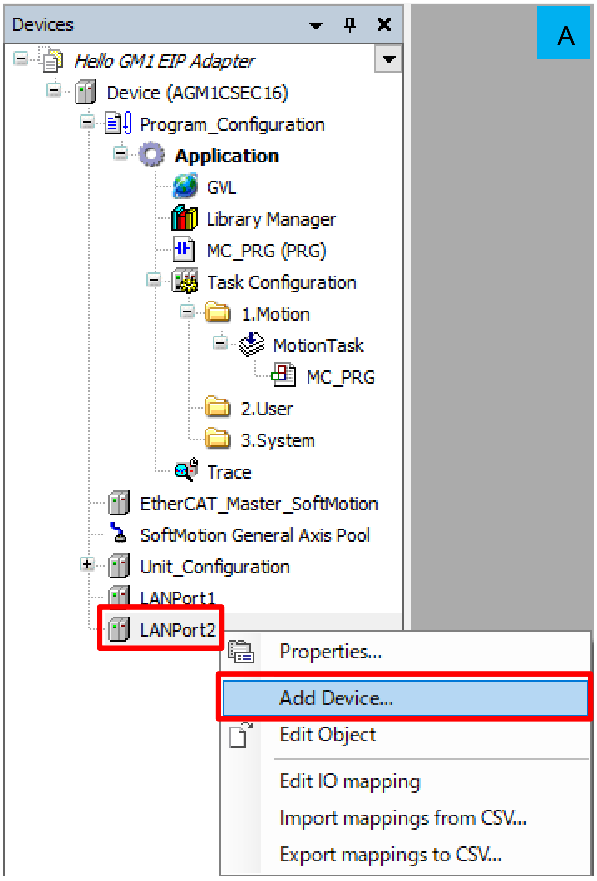
Module type	Size	Direction
Byte Input	1 byte	O → T
Byte Output	1 byte	T → O
Word Input	1 word (2 bytes)	O → T
Word Output	1 word (2 bytes)	T → O
DWord Input	1 dword (4 bytes)	O → T
DWord Output	1 dword (4 bytes)	T → O
Real Input 1	Single-precision real number (4 bytes)	O → T
Real Output 1	Single-precision real number (4 bytes)	T → O
Big Input	509 bytes	O → T
Big Output	505 bytes	T → O

When multiple data items are handled, multiple modules must be set.
Example) When two data items whose direction is O → T and whose size is 1 word are used: Two "Word Input" modules must be registered.

3.1 Adding Device

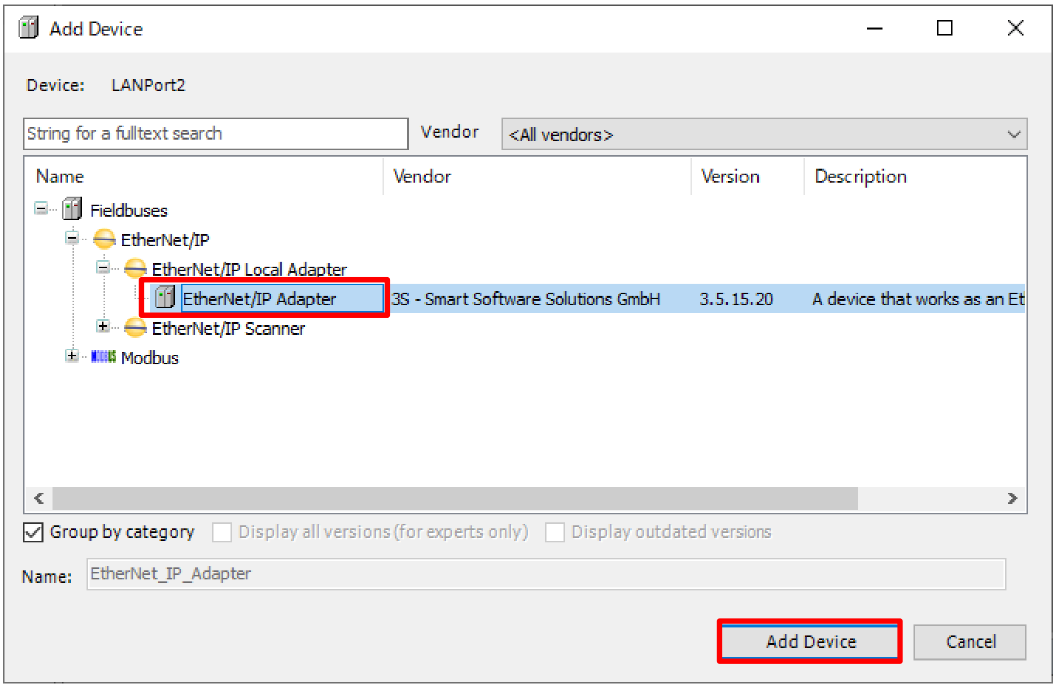
Step 1

Add an EtherNet/IP adapter device to LAN Port2.
In the navigation pane, right-click **LANPort2** and then select **Add Device**.



Step 2

Select **EtherNet/IP Local Adapter** and then **EtherNet/IP Adapter** and click **Add Device**.

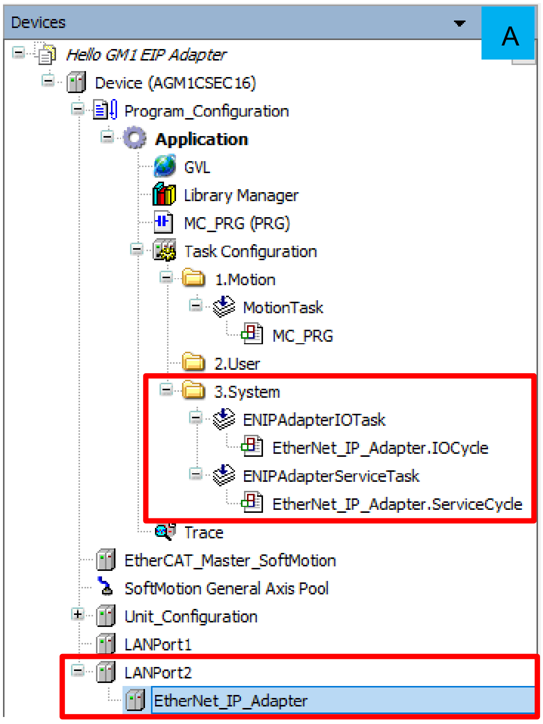

Add Device
String for a fulltext search
Vendor
<All vendors>
Fieldbuses
EtherNet/IP
EtherNet/IP Local Adapter
EtherNet/IP Scanner
Modbus
☑ Group by category
□ Display all versions (for experts only)
□ Display outdated versions
Add Device
Cancel

Device: LANPort2

Name	Vendor	Version	Description
EtherNet/IP Adapter	3S - Smart Software Solutions GmbH	3.5.15.20	A device that works as an Et

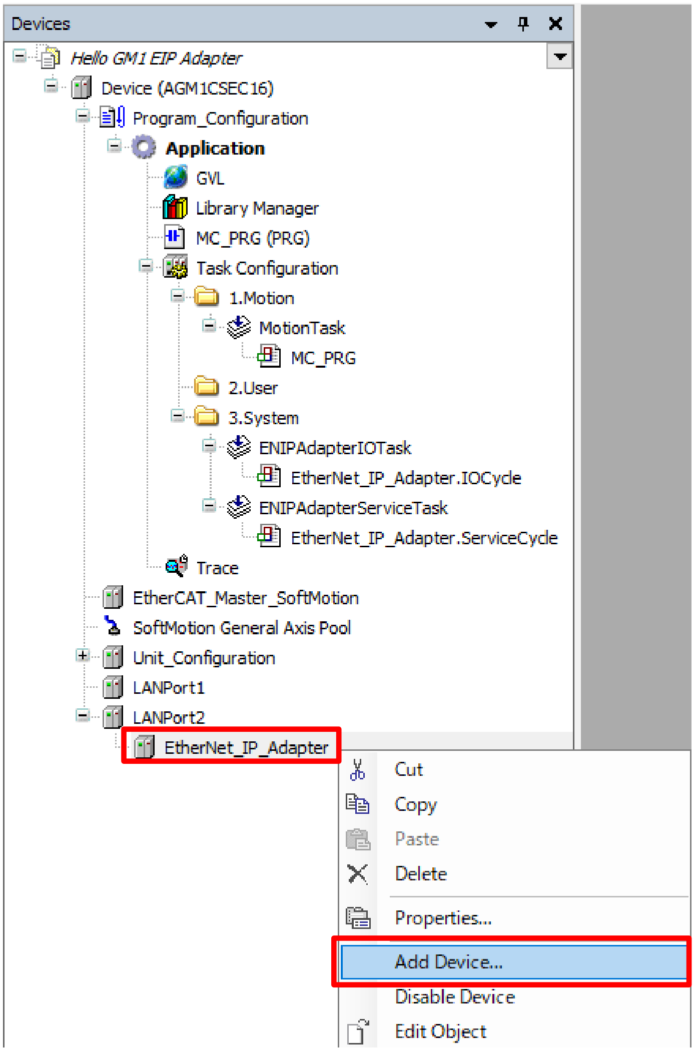
Name: EtherNet_IP_Adapter

ENIPAdapterIOTask and **ENIPAdapterServiceTask** will be added to **3.System** in **Task Configuration**. **EtherNet_IP_Adapter** will be added to **LANPort2**.



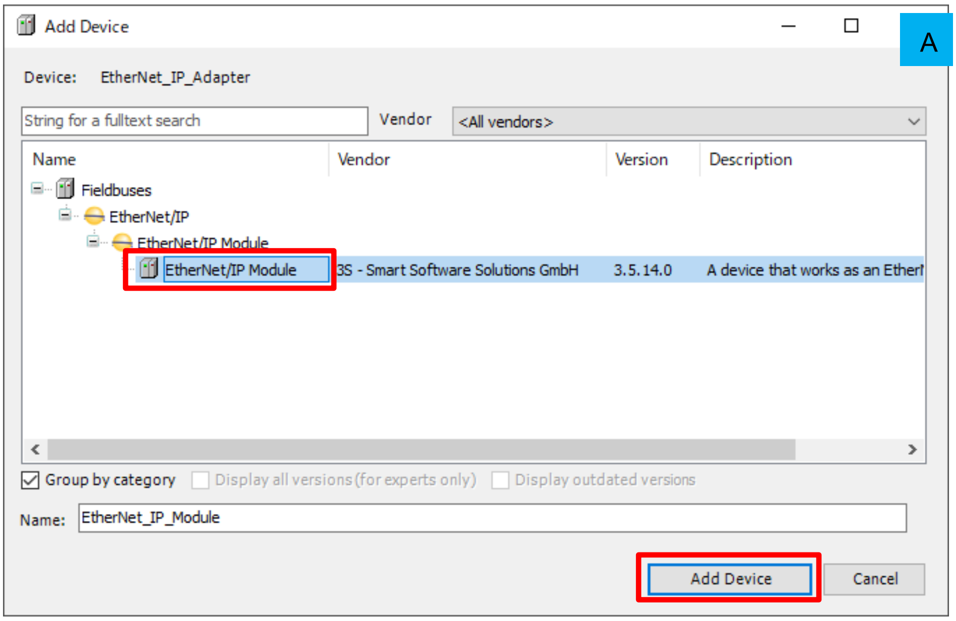
Step 3

Right-click **EtherNet_IP_Adapter**, which has been added, and select **Add Device**.



Step 4

Select **EtherNet/ IP Module** and then **EtherNet/ IP Module**, leave **EtherNet_IP_Module** (default) unchanged in the **Name** field, and click **Add Device**.



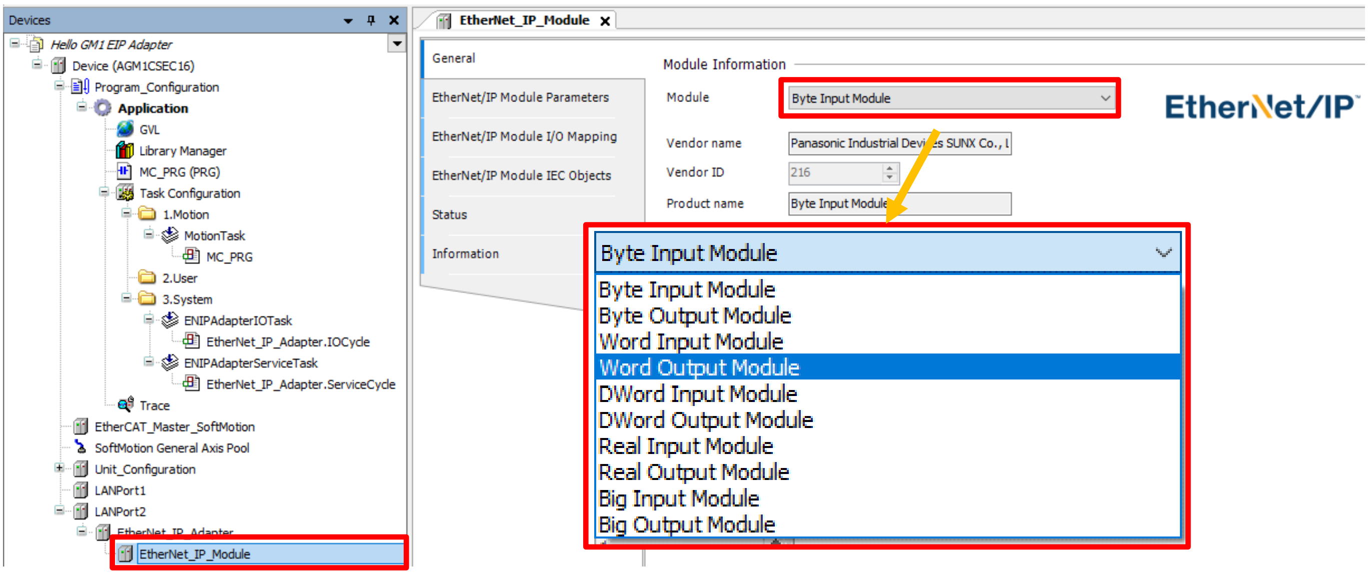
Add Device
A
Device: EtherNet_IP_Adapter
String for a fulltext search
Vendor
<All vendors>
Name
Vendor
Version
Description
Fieldbuses
EtherNet/IP
EtherNet/IP Module
EtherNet/IP Module
3S - Smart Software Solutions GmbH
3.5.14.0
A device that works as an Ethe
☑ Group by category
□ Display all versions (for experts only)
□ Display outdated versions
Name: EtherNet_IP_Module
Add Device
Cancel

3.2 Setting up Module

Step 1

Double-click **EtherNet_IP_Module** and select the **General** tab.

The **Module** drop-down list is set to **Byte Input Module** by default. Click in the frame and then select **Word Output Module** from the drop-down list.

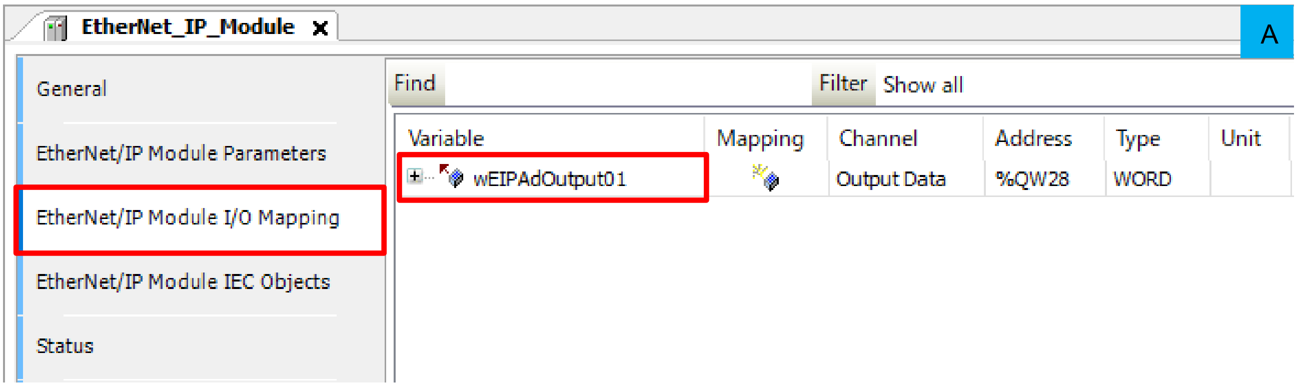


Step 2

Set a variable for this one word worth of data.

* Setting a variable is required to register for monitoring in the **Watch** window described later in this document.

Select the **EtherNet/IP Module I/O Mapping** tab and set variable name "wEIPAdOutput01" as shown below.


EtherNet_IP_Module
A
General
EtherNet/IP Module Parameters
EtherNet/IP Module I/O Mapping
EtherNet/IP Module IEC Objects
Status
Find
Filter
Show all

Variable	Mapping	Channel	Address	Type	Unit
wEIPAdOutput01		Output Data	%QW28	WORD	

Step 3

Add "EtehrNet/IP Module", select a module, and set a variable in the same way as above.

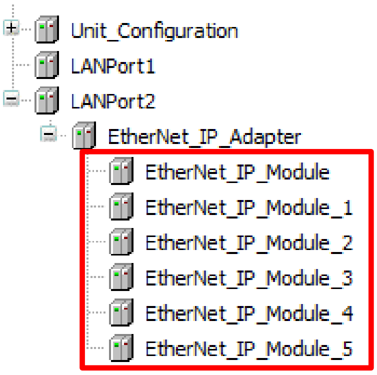
･Data settings for output (from adapter GM1 controller to scanner GM1 controller)

EtehrNet/IP Module name	Module	Registered variable name	Data content
EtherNet_IP_Module	Word Output Mobule	wEipAdOutput01	1-word data (Already registered)
EtherNet_IP_Module_1	Word Output Mobule	wEipAdOutput02	1-word data
EtherNet_IP_Module_2	DWord Output Mobule	dwEipAdOutput03	2-word data

･Data settings for input (to adapter GM1 controller from scanner GM1 controller)

EtehrNet/IP Module name	Module	Registered variable name	Data content
EtherNet_IP_Module_3	Word Input Mobule	wEipAdInput01	1-word data
EtherNet_IP_Module_4	Word Input Mobule	wEipAdInput02	1-word data
EtherNet_IP_Module_5	DWord Input Mobule	dwEipAdInput03	2-word data

Check that each item has been set under **EtherNet_IP_Adapter** in **LANPort2**.

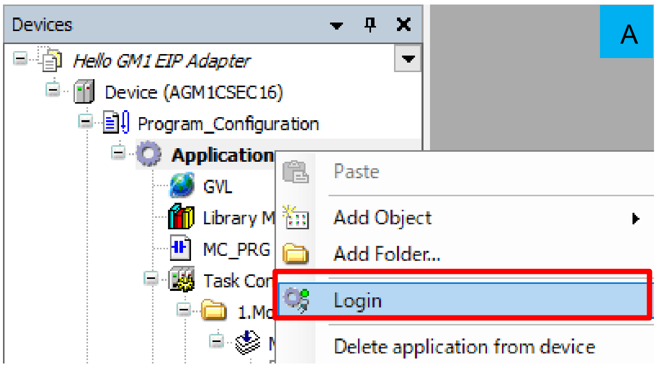


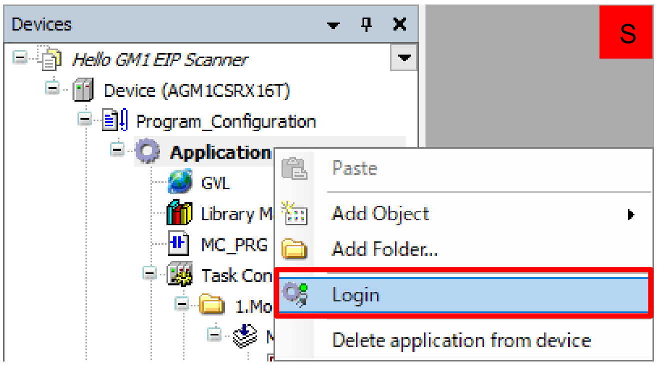
This completes EtherNet/IP setup for the adapter GM1 controller.

4 Communication Operation Check

Step 1

In GM Programmer on each of the scanner and adapter devices, right-click **Application** and then select **Login**.

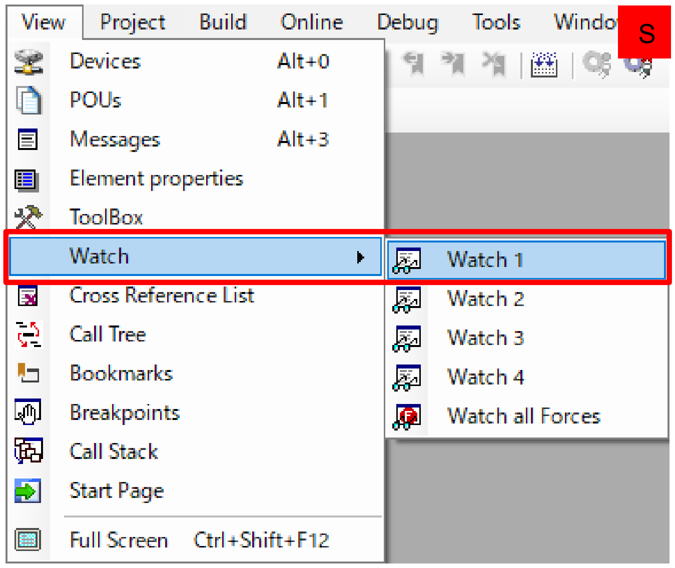




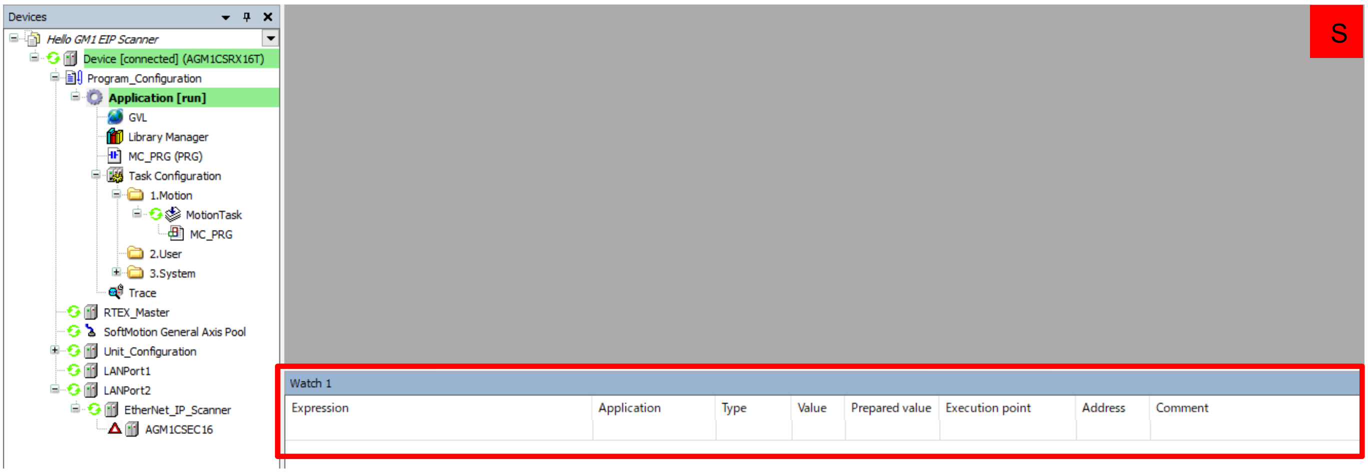
Step 2

First, open GM Programmer on the scanner device.

From the **View** menu, select **Watch** and then **Watch 1**.



"Watch 1" will be displayed on the bottom of the window.

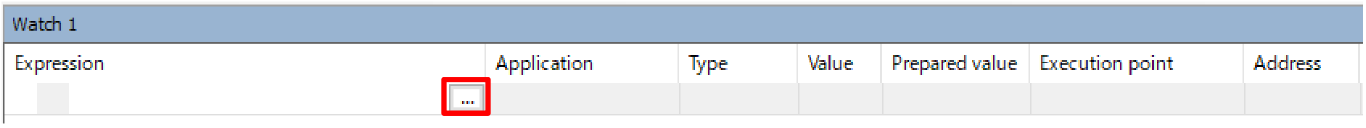


"POU:EIP02_PRG" will open in GM Programmer and "User Monitor" will open in FPWIN Pro7. Check that the device is in a state in which the communication status can be monitored.

Data transmission from the adapter device (GM1 controller) to the scanner device (FP0H)

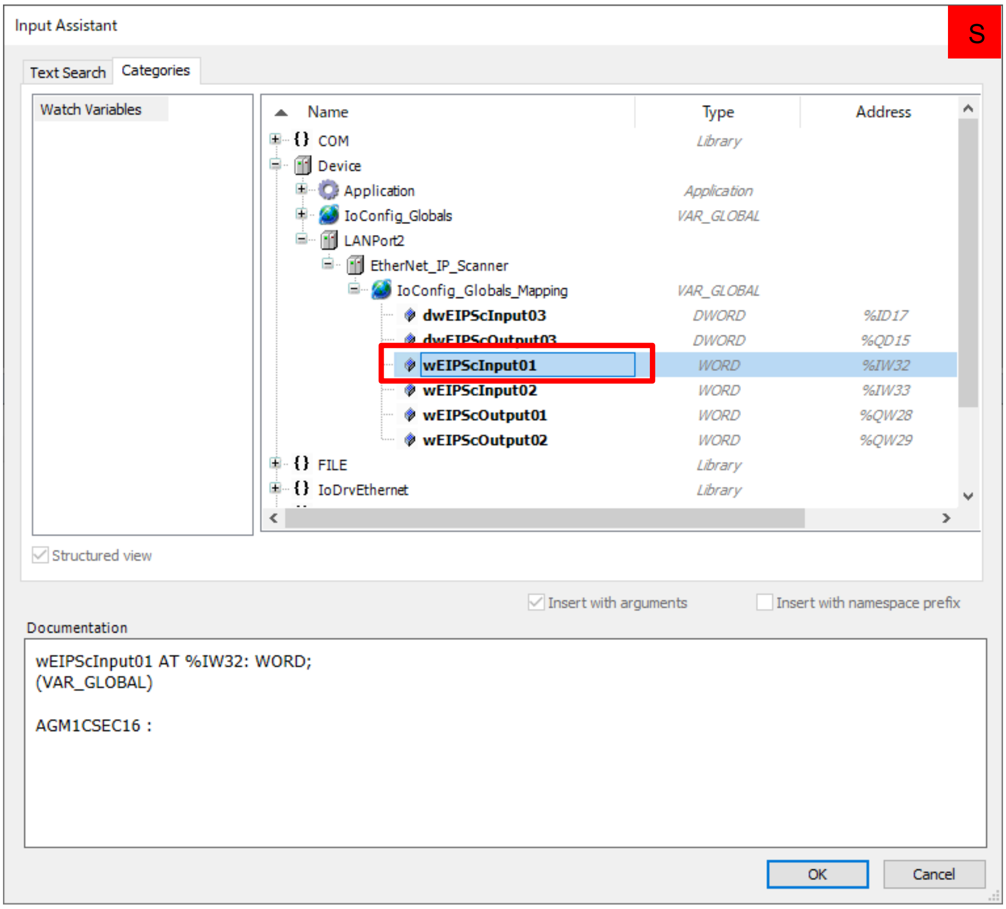
Step 3

In the **Watch 1** pane, click in the **Expression** column and then click [...].



Step 4

Click **Device**, **LANPort2**, **EtherNet_IP_Scanner**, and then **IoConfig_Globals_Mapping** and click ⊞ to expand the **IoConfig_Globals_Mapping** object. Select **wEIPScInput01** and click **OK**.

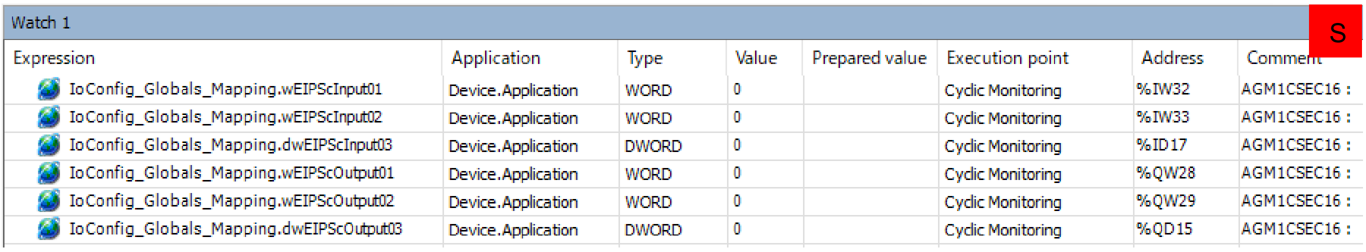


Name	Type	Address
⊞ {} COM	Library	
⊟ Device		
⊞ Application	Application	
⊞ IoConfig_Globals	VAR_GLOBAL	
⊟ LANPort2		
⊟ EtherNet_IP_Scanner		
⊟ IoConfig_Globals_Mapping	VAR_GLOBAL	
dwEIPScInput03	DWORD	%ID17
dwEIPScOutput03	DWORD	%QD15
wEIPScInput01	WORD	%IW32
wEIPScInput02	WORD	%IW33
wEIPScOutput01	WORD	%QW28
wEIPScOutput02	WORD	%QW29
⊞ {} FILE	Library	
⊞ {} IoDrvEthernet	Library	

Documentation

```
wEIPScInput01 AT %IW32: WORD;
(VAR_GLOBAL)

AGM1CSEC16 :
```

Register **wEIPScInput02**, **dwEIPScInput03**, **wEIPScOutput01**, **wEIPScOutput02**, and **dwEIPScOutput03** in the **Watch 1** window, in the same way as above.



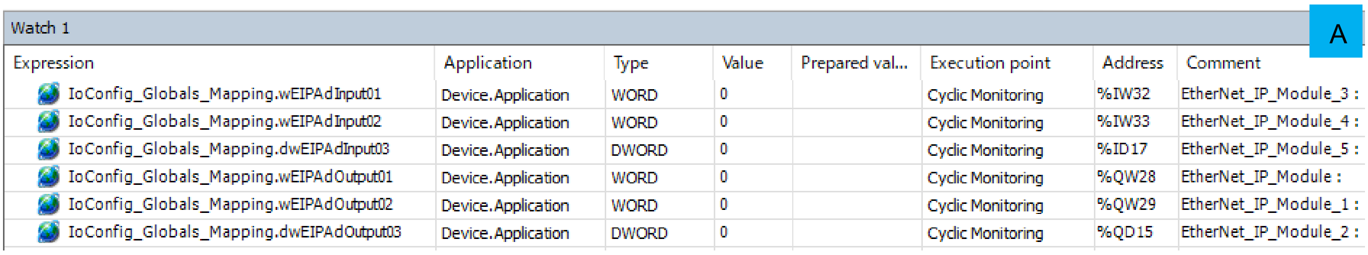
Expression	Application	Type	Value	Prepared value	Execution point	Address	Comment
IoConfig_Globals_Mapping.wEIPScInput01	Device.Application	WORD	0		Cyclic Monitoring	%IW32	AGM1CSEC16 :
IoConfig_Globals_Mapping.wEIPScInput02	Device.Application	WORD	0		Cyclic Monitoring	%IW33	AGM1CSEC16 :
IoConfig_Globals_Mapping.dwEIPScInput03	Device.Application	DWORD	0		Cyclic Monitoring	%ID17	AGM1CSEC16 :
IoConfig_Globals_Mapping.wEIPScOutput01	Device.Application	WORD	0		Cyclic Monitoring	%QW28	AGM1CSEC16 :
IoConfig_Globals_Mapping.wEIPScOutput02	Device.Application	WORD	0		Cyclic Monitoring	%QW29	AGM1CSEC16 :
IoConfig_Globals_Mapping.dwEIPScOutput03	Device.Application	DWORD	0		Cyclic Monitoring	%QD15	AGM1CSEC16 :

Step 5

Next, open GM Programmer on the adapter device.

Display the **Watch 1** window in the same way as for GM Programmer on the scanner device and register the following variables.

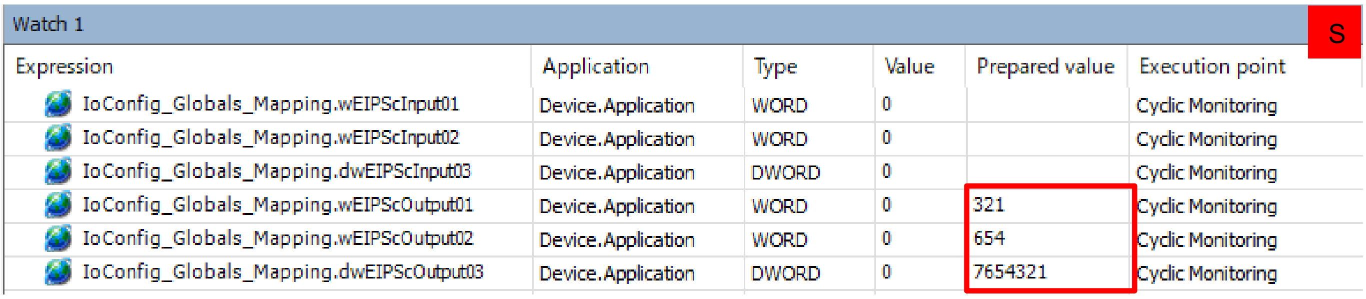
wEIPAdInput01, wEIPAdInput02, dwEIPAdInput03, wEIPAdOutput01, wEIPAdOutput02, and dwEIPAdOutput03



Expression	Application	Type	Value	Prepared val...	Execution point	Address	Comment
IoConfig_Globals_Mapping.wEIPAdInput01	Device.Application	WORD	0		Cyclic Monitoring	%IW32	EtherNet_IP_Module_3 :
IoConfig_Globals_Mapping.wEIPAdInput02	Device.Application	WORD	0		Cyclic Monitoring	%IW33	EtherNet_IP_Module_4 :
IoConfig_Globals_Mapping.dwEIPAdInput03	Device.Application	DWORD	0		Cyclic Monitoring	%ID17	EtherNet_IP_Module_5 :
IoConfig_Globals_Mapping.wEIPAdOutput01	Device.Application	WORD	0		Cyclic Monitoring	%QW28	EtherNet_IP_Module :
IoConfig_Globals_Mapping.wEIPAdOutput02	Device.Application	WORD	0		Cyclic Monitoring	%QW29	EtherNet_IP_Module_1 :
IoConfig_Globals_Mapping.dwEIPAdOutput03	Device.Application	DWORD	0		Cyclic Monitoring	%QD15	EtherNet_IP_Module_2 :

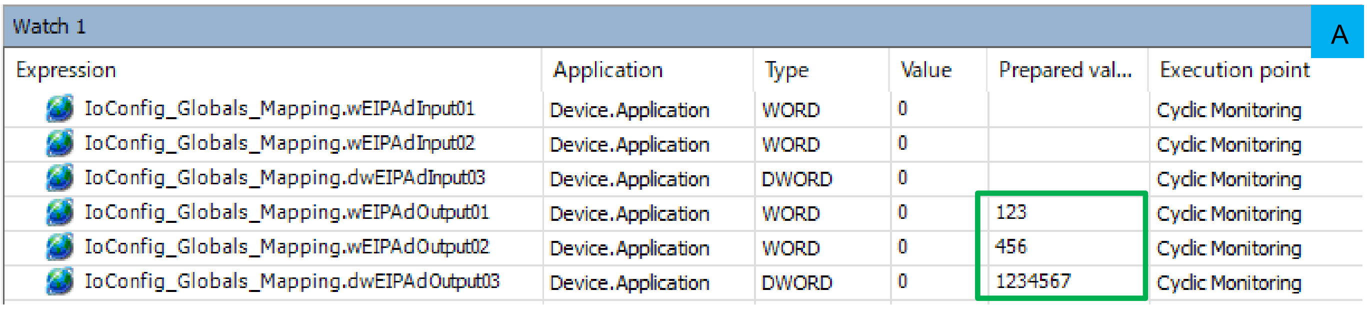
Step 6

Enter any desired values in the **Prepared value** columns corresponding to **wEIPScOutput01**, **wEIPScOutput02**, and **dwEIPScOutput03** in the **Watch 1** window on the scanner device and corresponding to **wEIPAdOutput01**, **wEIPAdOutput02**, and **dwEIPAdOutput03** in the **Watch 1** window on the adapter device.



Watch 1

Expression	Application	Type	Value	Prepared value	Execution point
IoConfig_Globals_Mapping.wEIPScInput01	Device.Application	WORD	0		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPScInput02	Device.Application	WORD	0		Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPScInput03	Device.Application	DWORD	0		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPScOutput01	Device.Application	WORD	0	321	Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPScOutput02	Device.Application	WORD	0	654	Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPScOutput03	Device.Application	DWORD	0	7654321	Cyclic Monitoring

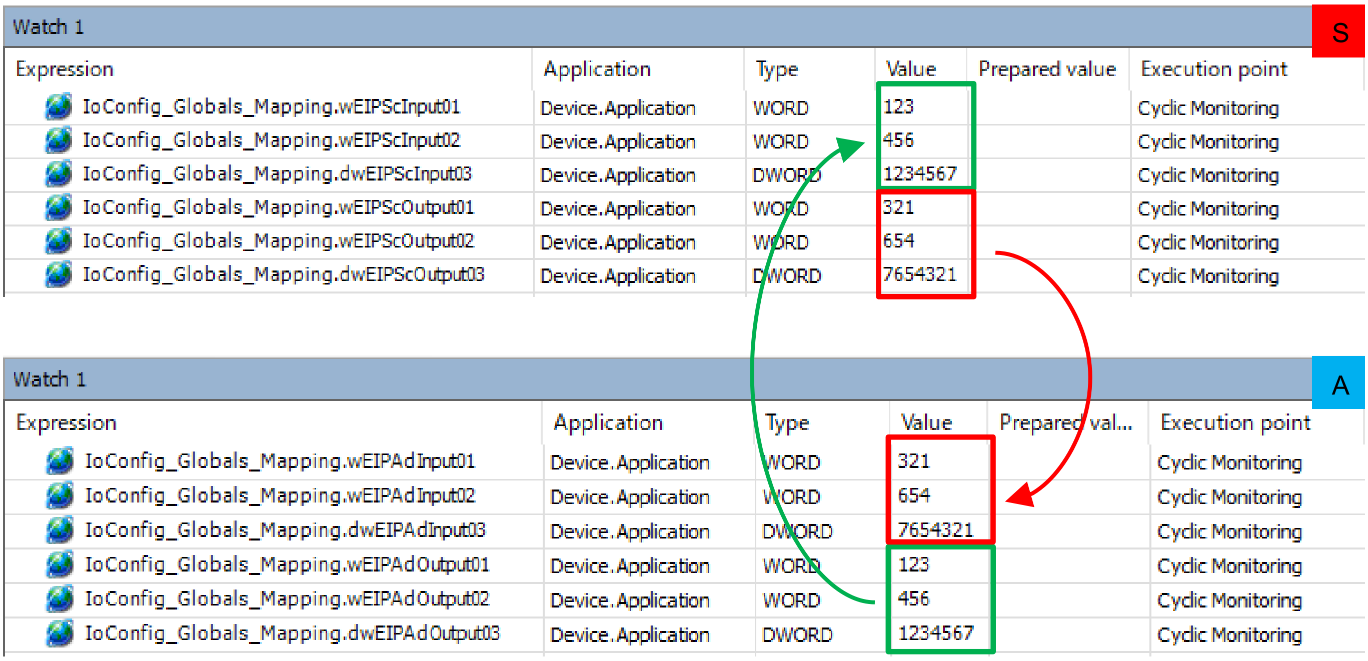


Watch 1

Expression	Application	Type	Value	Prepared val...	Execution point
IoConfig_Globals_Mapping.wEIPAdInput01	Device.Application	WORD	0		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPAdInput02	Device.Application	WORD	0		Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPAdInput03	Device.Application	DWORD	0		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPAdOutput01	Device.Application	WORD	0	123	Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPAdOutput02	Device.Application	WORD	0	456	Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPAdOutput03	Device.Application	DWORD	0	1234567	Cyclic Monitoring

Step 7

Press Ctr+F7 to write the values.



Watch 1

Expression	Application	Type	Value	Prepared value	Execution point
IoConfig_Globals_Mapping.wEIPScInput01	Device.Application	WORD	123		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPScInput02	Device.Application	WORD	456		Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPScInput03	Device.Application	DWORD	1234567		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPScOutput01	Device.Application	WORD	321		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPScOutput02	Device.Application	WORD	654		Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPScOutput03	Device.Application	DWORD	7654321		Cyclic Monitoring

Watch 1

Expression	Application	Type	Value	Prepared val...	Execution point
IoConfig_Globals_Mapping.wEIPAdInput01	Device.Application	WORD	321		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPAdInput02	Device.Application	WORD	654		Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPAdInput03	Device.Application	DWORD	7654321		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPAdOutput01	Device.Application	WORD	123		Cyclic Monitoring
IoConfig_Globals_Mapping.wEIPAdOutput02	Device.Application	WORD	456		Cyclic Monitoring
IoConfig_Globals_Mapping.dwEIPAdOutput03	Device.Application	DWORD	1234567		Cyclic Monitoring

This completes EtherNet/IP communication operation check for the GM1 controllers.

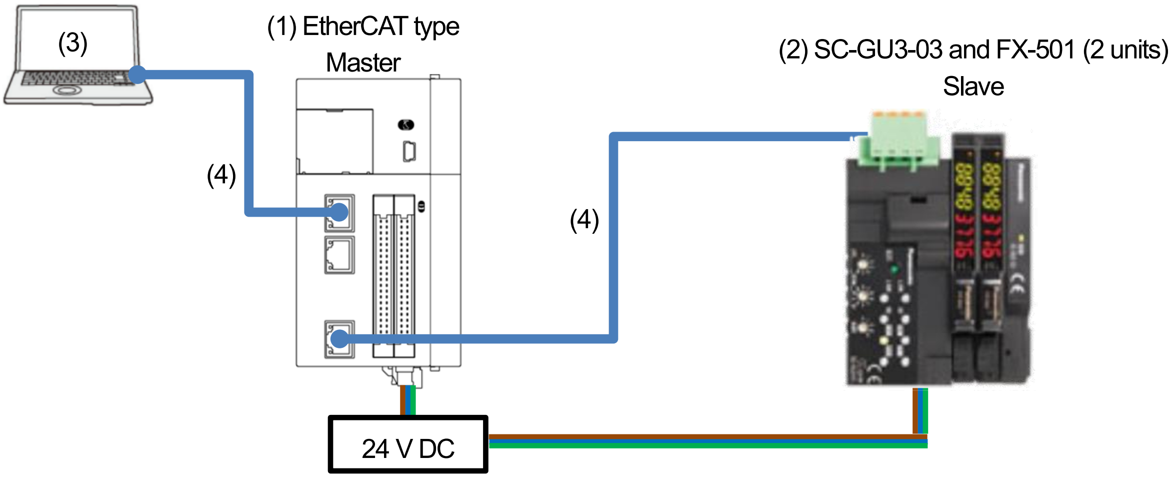
Ethernet Communication: EtherCAT Slave

1 Basic Setup

1.1 Preparing and Wiring the Required Devices

Prepare the following devices.

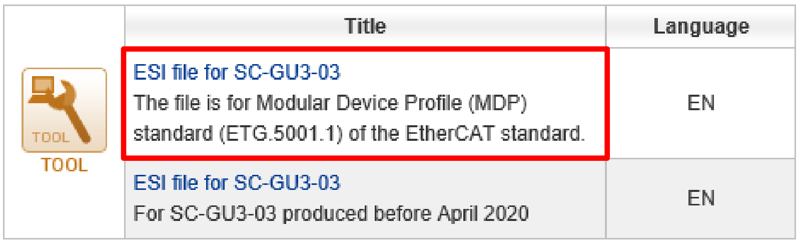
No.	Name
(1)	GM1 controller (EtherCAT type) x 1: Master
(2)	SC-GU3-03 and FX-501 (fiber sensor): 2 units
(3)	PC (with GM Programmer installed)
(4)	LAN cable: x 2



1.2 Installing ESI File

Step 1

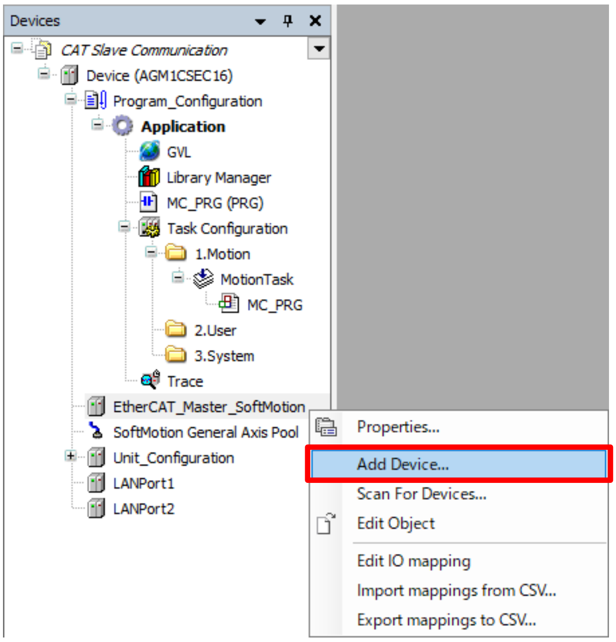
Download an ESI file from the following website.
https://www3.panasonic.biz/ac/j/dl/software/index.jsp?series_cd=1454



	Title	Language
TOOL	ESI file for SC-GU3-03 The file is for Modular Device Profile (MDP) standard (ETG.5001.1) of the EtherCAT standard.	EN
	ESI file for SC-GU3-03 For SC-GU3-03 produced before April 2020	EN

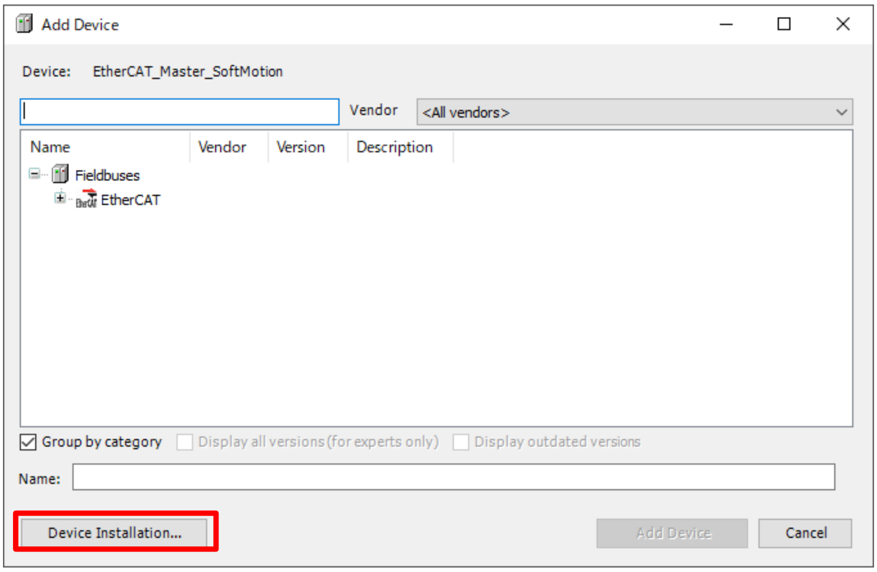
Step 2

Right-click **EtherCAT_Master_SoftMotion** and select **Add Device**.



Step 3

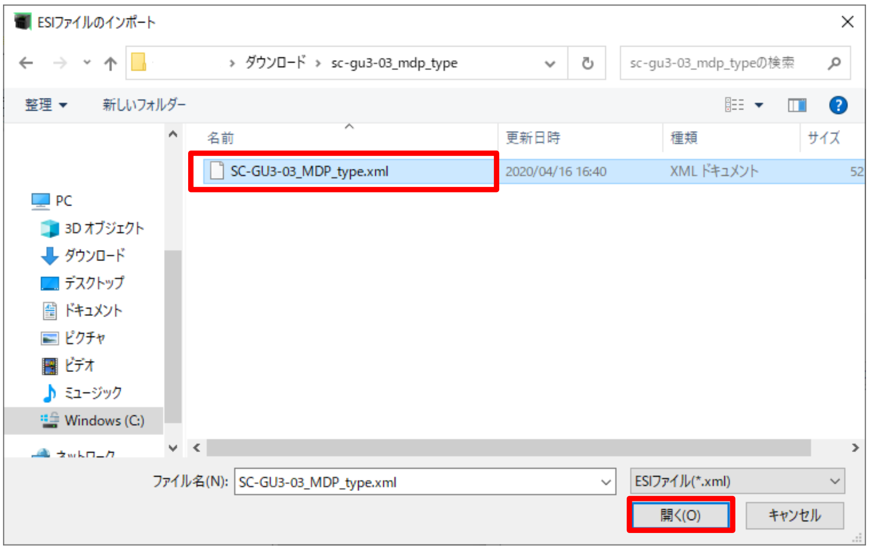
The **Add Device** dialog box will be displayed. Click **Device Installation**.



Device: EtherCAT_Master_SoftMotion

Step 4

Select an ESI file that was downloaded previously and click **Open**.



名前	更新日時	種類	サイズ
SC-GU3-03_MDP_type.xml	2020/04/16 16:40	XML ドキュメント	52

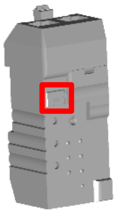
ファイル名(N): SC-GU3-03_MDP_type.xml

This completes installation of the SC-GU3-03 communication unit.

1.3 Adding Device (SC-GU3-03)

Step 1

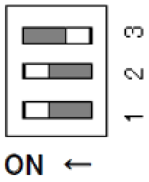
Open the cover indicated by the red frame and set the DIP switch as below.



1	ON
2	ON
3	OFF

INFO

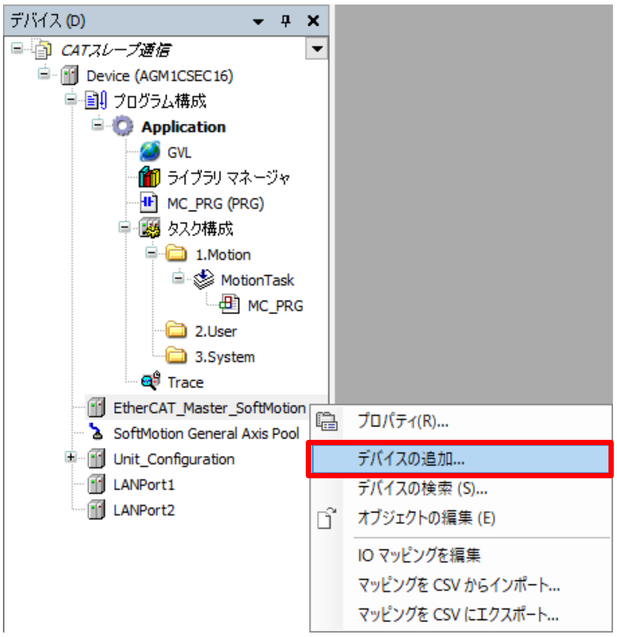
I/O communication is divided into three operation modes. The operation mode can be switched using the DIP switch. The factory default setting is full mode.



		bit321	T×PDO	R×PDO
mode1	I/O mode	001	2 byte	0 byte
mode2	Check mode	010	4 byte	0 byte
mode3	Full mode	011	44 byte	10 byte

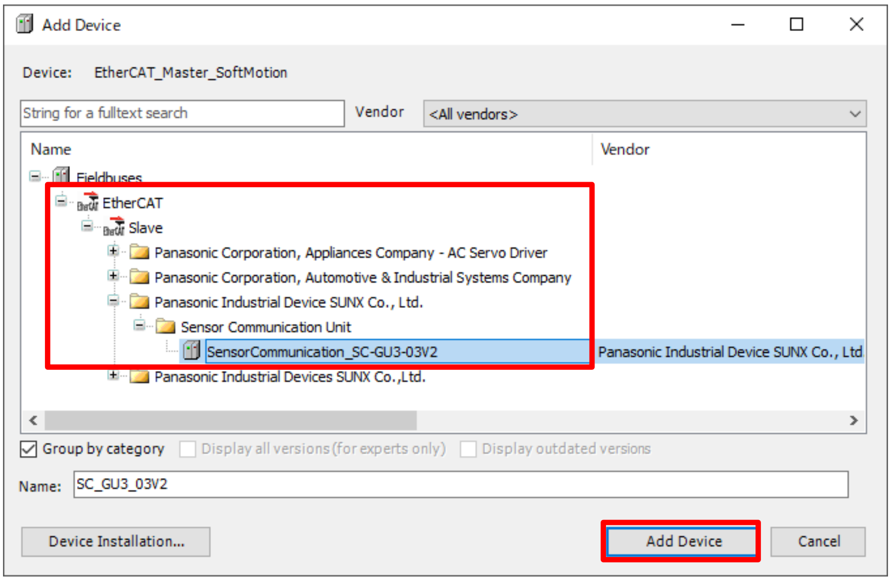
Step 2

Right-click **EtherCAT_Master_SoftMotion** and select **Add Device**.



Step 3

Select **EtherCAT**, **Slave**, **Panasonic Industrial Device SUNX Co.,Ltd.**, and then **SensorCommunication_SC_GU3_03V2**. Click **Add Device**.


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 Corporation, Automotive & Industrial Systems Company
Panasonic Industrial Device SUNX Co., Ltd.
Sensor Communication Unit
SensorCommunication_SC-GU3-03V2
Panasonic Industrial Device SUNX Co., Ltd
Panasonic Industrial Devices SUNX Co.,Ltd.

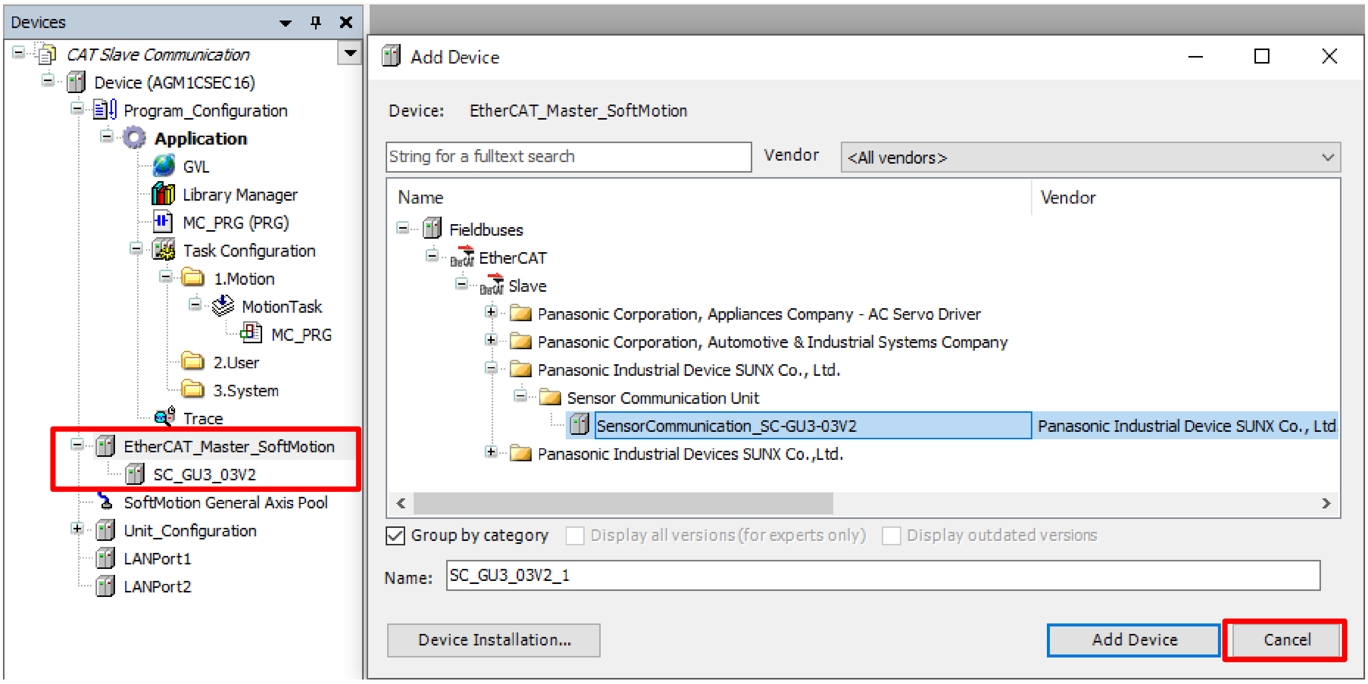
☑ Group by category □ Display all versions (for experts only) □ Display outdated versions

Name: SC_GU3_03V2

Device Installation...
Add Device
Cancel

Step 4

Check that "SensorCommunication_SC_GU3_03V2" has been added and then click **Cancel**.


Devices

CAT Slave Communication
- Device (AGM1CSEC16)
  - Program_Configuration
    - Application
      - GVL
      - Library Manager
      - MC_PRG (PRG)
      - Task Configuration
        - 1.Motion
          - MotionTask
            - MC_PRG
        - 2.User
        - 3.System
      - Trace
  - EtherCAT_Master_SoftMotion
    - SC_GU3_03V2
  - SoftMotion General Axis Pool
  - Unit_Configuration
  - LANPort1
  - LANPort2

Add Device

Device: EtherCAT_Master_SoftMotion

String for a fulltext search | Vendor: <All vendors>

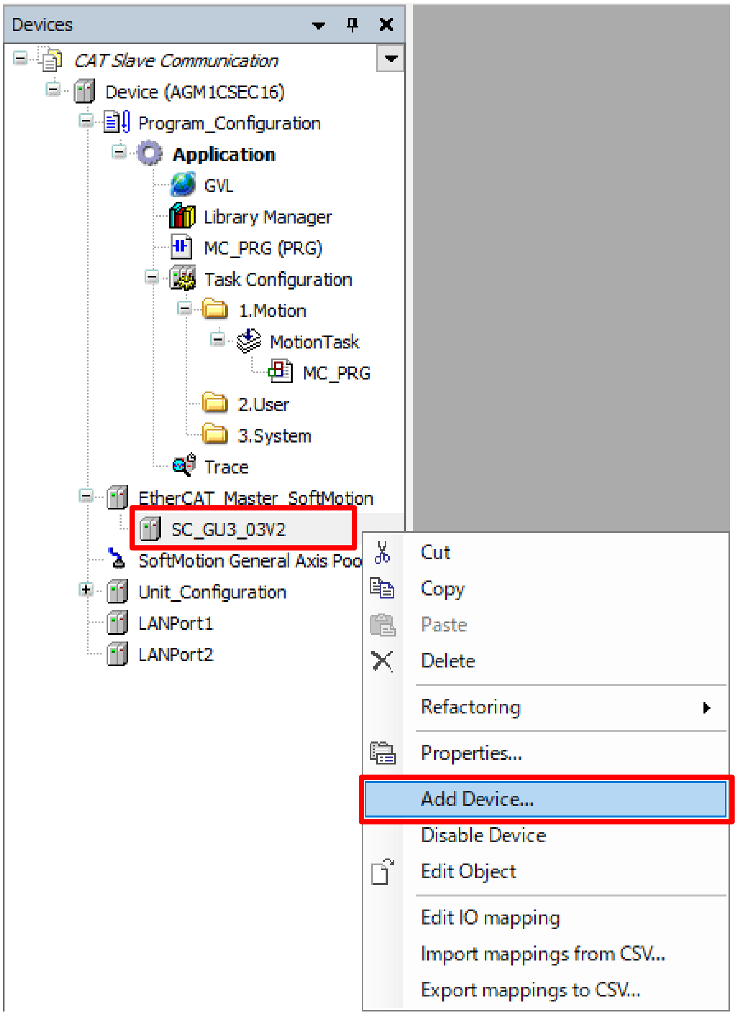
Name	Vendor
Fieldbuses	
EtherCAT	
Slave	
Panasonic Corporation, Appliances Company - AC Servo Driver	
Panasonic Corporation, Automotive & Industrial Systems Company	
Panasonic Industrial Device SUNX Co., Ltd.	
Sensor Communication Unit	
SensorCommunication_SC-GU3-03V2	Panasonic Industrial Device SUNX Co., Ltd
Panasonic Industrial Devices SUNX Co.,Ltd.	

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

Name: SC_GU3_03V2_1

Step 5

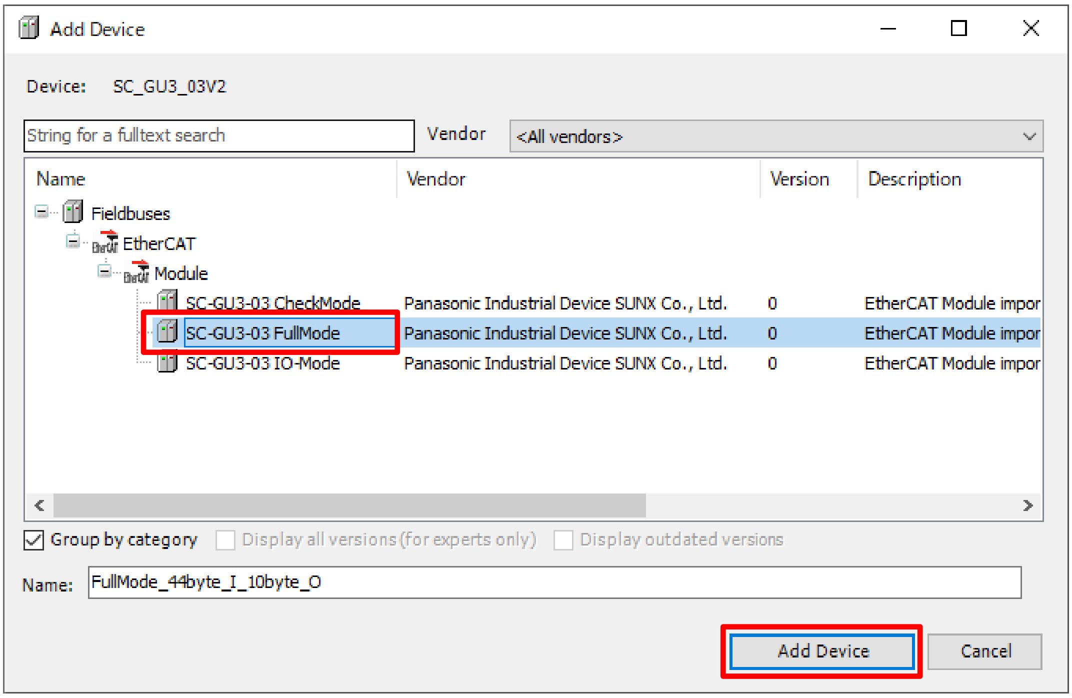
Right-click **SC-GU3-03V2** and then select **Add Device**.


Devices

CAT Slave Communication
- Device (AGM1CSEC16)
  - Program_Configuration
    - Application
      - GVL
      - Library Manager
      - MC_PRG (PRG)
      - Task Configuration
        - 1.Motion
          - MotionTask
            - MC_PRG
        - 2.User
        - 3.System
      - Trace
  - EtherCAT_Master_SoftMotion
    - SC_GU3_03V2
  - SoftMotion General Axis Pool
  - Unit_Configuration
  - LANPort1
  - LANPort2

Step 6

Because the DIP switch was set to full mode in Step 1, select **SC-GU3-03 FullMode** and click **Add Device**.



Add Device

Device: SC_GU3_03V2

String for a fulltext search

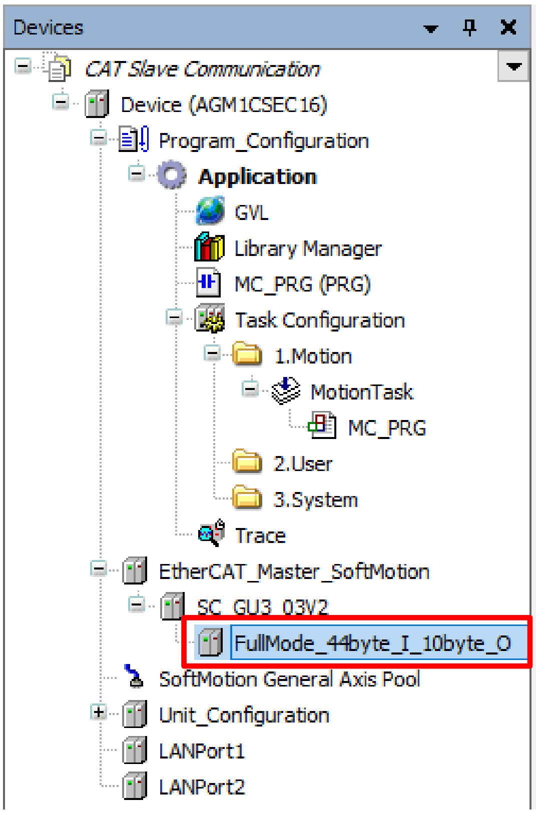
Vendor <All vendors>

Name	Vendor	Version	Description
Fieldbuses			
EtherCAT			
Module			
SC-GU3-03 CheckMode	Panasonic Industrial Device SUNX Co., Ltd.	0	EtherCAT Module impor
SC-GU3-03 FullMode	Panasonic Industrial Device SUNX Co., Ltd.	0	EtherCAT Module impor
SC-GU3-03 IO-Mode	Panasonic Industrial Device SUNX Co., Ltd.	0	EtherCAT Module impor

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

Name: FullMode_44byte_I_10byte_O

FullMode_44byte_I_10byte_O will be added.



Column (9): Station ID setting

This section explains how to use the rotary switch on the front panel of the SC-GU3-03 communication unit. Write "0" to 2001h/00h via SDO communication.

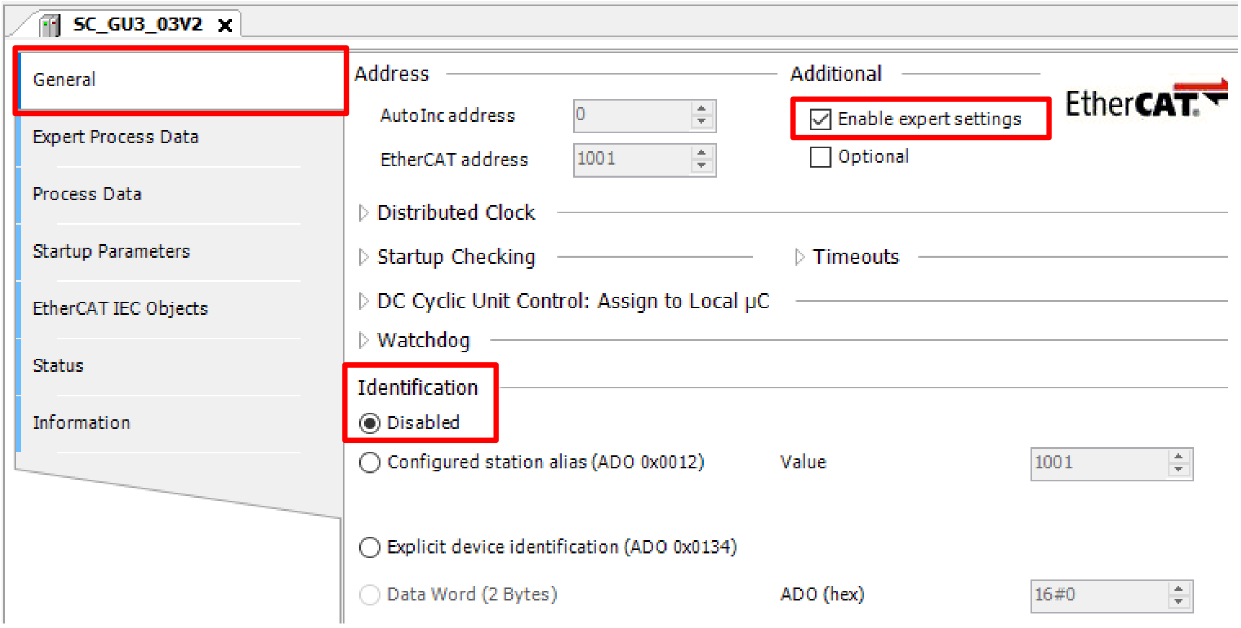
	Index	SubIndex	Name	Flag	Function
Manufacturer Specific Area	2001h	00h	R×PDO Station Alias setup (Hi)	RW	The station alias setting method is selected. (Factory default setting: 0001h) 0000h: "R×PDO Station Alias setup (Hi)" setting of rotary switch 0001h: SII EEPROM setting Refer to "1.6 Setting ESC Registers 0012h and 0013h (Configured Station Alias)".

Step 1

Connect the GM1 controller and SC-GU3-03 communication unit on a one-to-one basis.

Step 2

Double-click **SC-GU3-03** and open the **General** tab.
Select the **Enable expert settings** check box and select the **Disabled** option in **Identification**.


SC_GU3_03V2

General
Expert Process Data
Process Data
Startup Parameters
EtherCAT IEC Objects
Status
Information

Address
AutoInc address: 0
EtherCAT address: 1001

Additional
☑ Enable expert settings
□ Optional

Identification
● Disabled
○ Configured station alias (ADO 0x0012) Value 1001
○ Explicit device identification (ADO 0x0134)
○ Data Word (2 Bytes) ADO (hex) 16#0

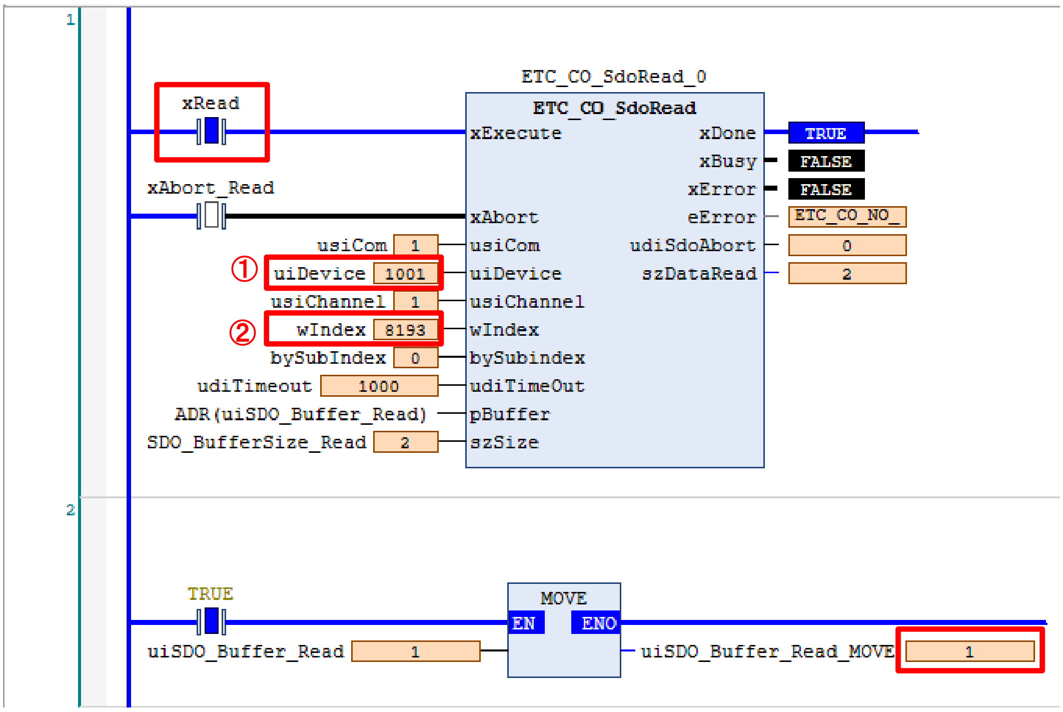
Step 3

Use Sdo_Read to read the current value.

Create a program as shown in the figure below and rewrite **xRead** to TRUE.

If **xDone** is set to TRUE, read processing is complete.

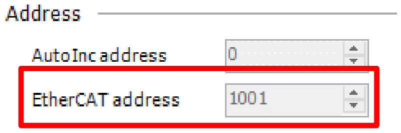
Check that **uiSDO_Buffer_Read_MOVE** is set to 1.



```
1
                                  ETC_CO_SdoRead_0
                                  ETC_CO_SdoRead
 xRead                            xExecute        xDone     TRUE
                                                  xBusy     FALSE
 xAbort_Read                                      xError    FALSE
                                  xAbort          eError    ETC_CO_NO_
            usiCom       1        usiCom     udiSdoAbort    0
 ①        uiDevice    1001        uiDevice    szDataRead    2
          usiChannel     1        usiChannel
 ②          wIndex    8193        wIndex
         bySubIndex      0        bySubindex
       udiTimeout     1000        udiTimeOut
   ADR(uiSDO_Buffer_Read)         pBuffer
 SDO_BufferSize_Read     2        szSize

2
 TRUE                       MOVE
                            EN      ENO
 uiSDO_Buffer_Read    1                    uiSDO_Buffer_Read_MOVE    1
```

① Double-click **SC-GU3-03V2**, open the **General** tab, and enter a value in **EtherCAT address** in **Address**.

EtherCAT addresses are assigned serial numbers such as 1001, 1002, and 1003 in the order of connection.



Address

AutoInc address 0

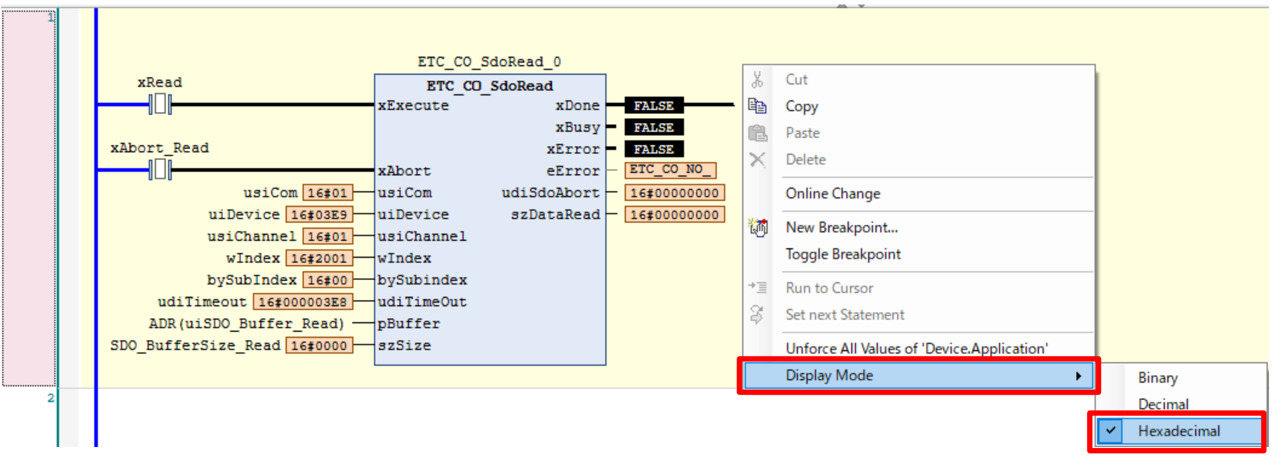
EtherCAT address 1001

② Enter the index number of R×PDO Station Alias setup (Hi).

This index is "2001h", so it is "8193" in hexadecimal notation.

INFO

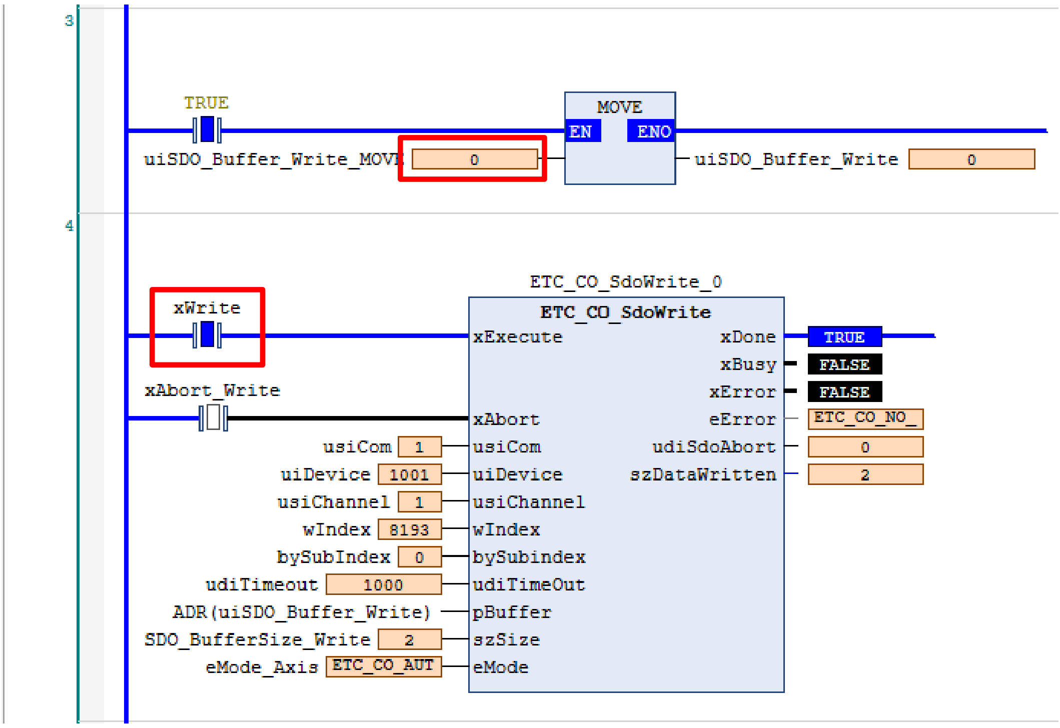
If you right-click on the screen and then select **Display Mode**, you can switch the display mode between Binary, Decimal, and Hexadecimal.



```
                                  ETC_CO_SdoRead_0
                                  ETC_CO_SdoRead
 xRead                            xExecute        xDone     FALSE
                                                  xBusy     FALSE
 xAbort_Read                                      xError    FALSE
                                  xAbort          eError    ETC_CO_NO_
            usiCom   16#01        usiCom     udiSdoAbort    16#00000000
          uiDevice   16#03E9      uiDevice    szDataRead    16#00000000
        usiChannel   16#01        usiChannel
            wIndex   16#2001      wIndex
        bySubIndex   16#00        bySubindex
        udiTimeout   16#000003E8  udiTimeOut
   ADR(uiSDO_Buffer_Read)         pBuffer
 SDO_BufferSize_Read 16#0000      szSize
```

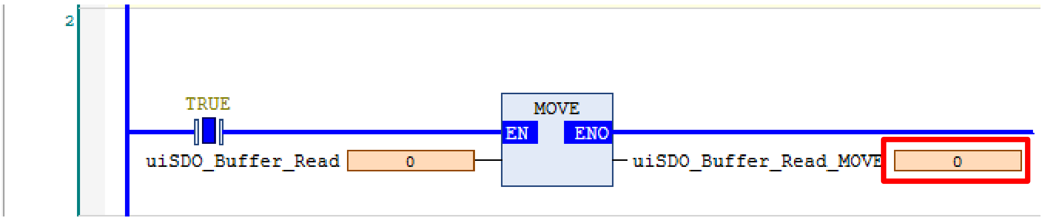
Step 4

Use Sdo_Write to write data.
Create a program as shown in the figure below and enter data to be written in **uiSDO_Buffer_Write_MOVE**.
In this example, enter "0" and rewrite **xWrite** to TRUE.
If **xDone** is set to TRUE, write processing is complete.



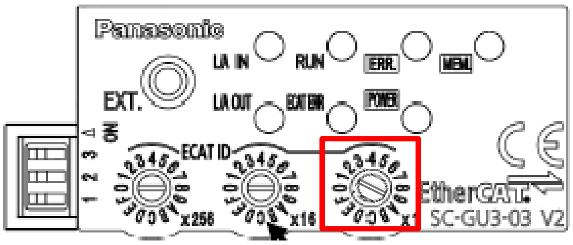
Step 5

Perform Step 3 again and check that **uiSDO_Buffer_Read_MOVE** is set to 0.



Step 6

Change the address setting of the rotary switch on the SC-GU3-03 communication unit to "1" and turn the power OFF and then ON.

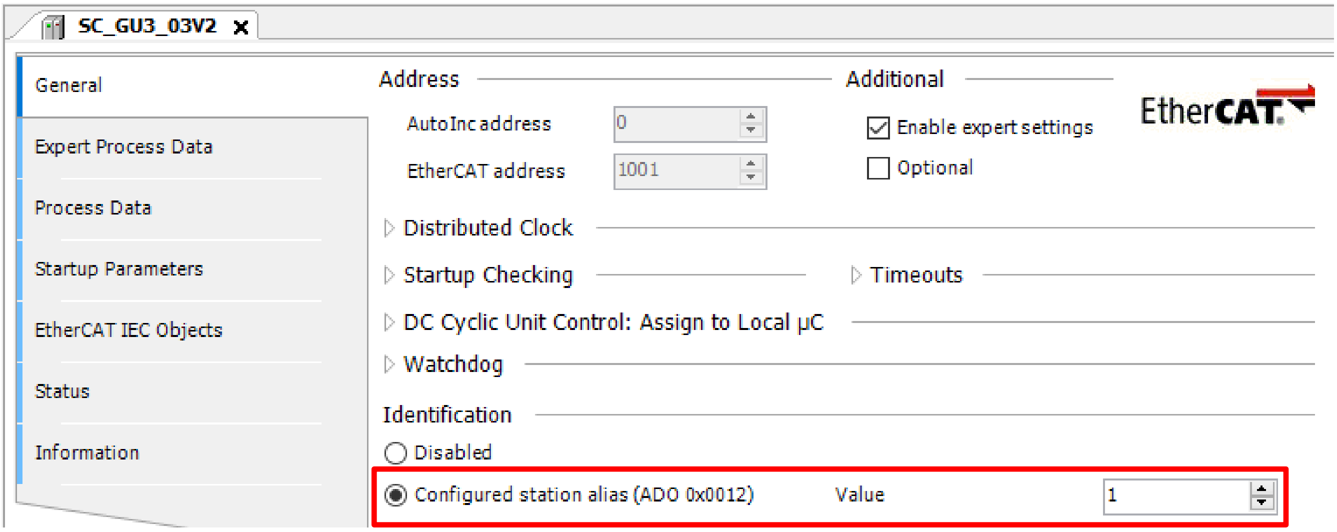


Step 7

Double-click **SC-GU3-03** and open the **General** tab.

In **Identification**, select the **Configured station alias (ADO 0x0012)** option and then set the value to 1.

Download to the GM1 controller.



SC_GU3_03V2

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

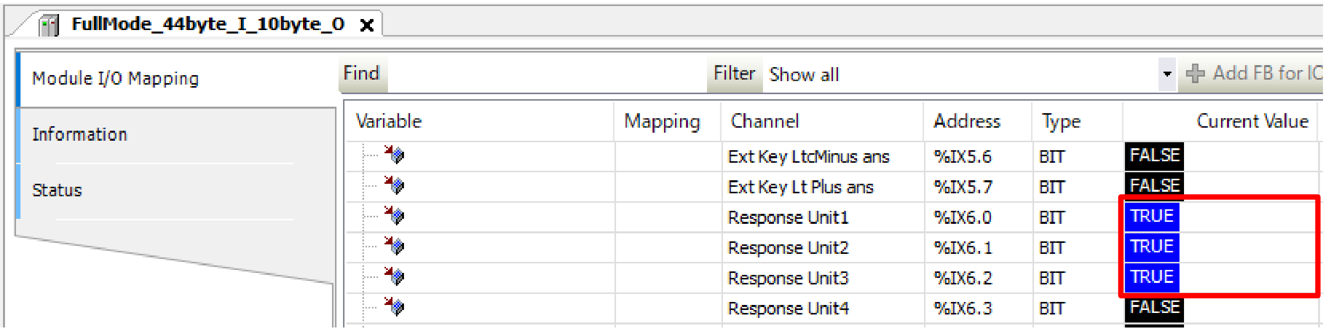
○ Disabled

● Configured station alias (ADO 0x0012) Value 1

Step 8

If the connection status and output can be confirmed as shown in the figure below, the procedure is complete.

(In this example, three fiber sensor amplifiers are connected, Unit1 to Unit3 are set to TRUE.)



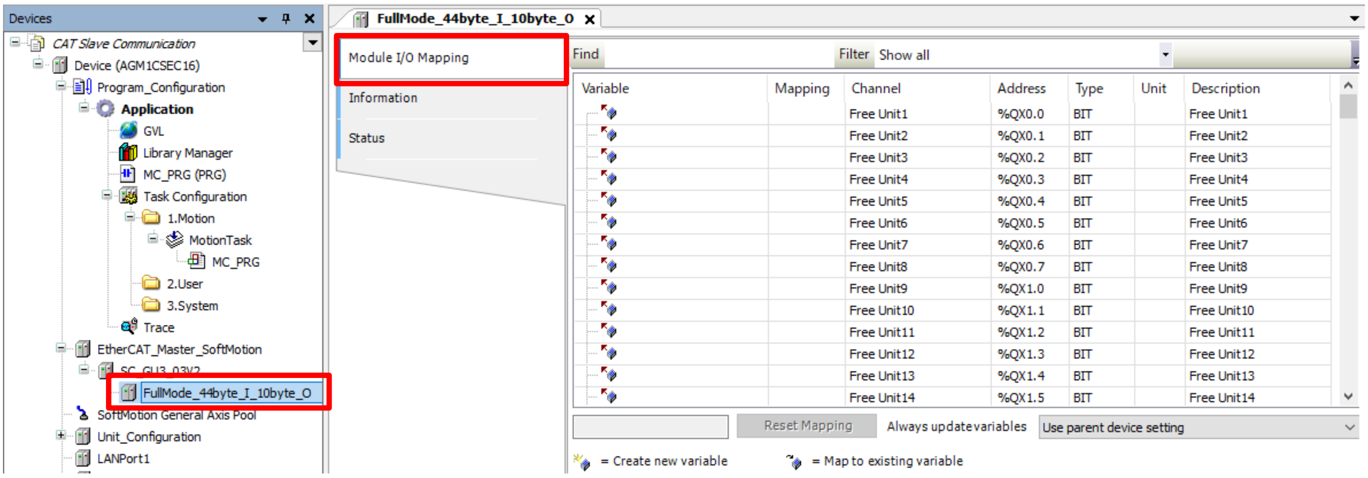
FullMode_44byte_I_10byte_O

Variable	Mapping	Channel	Address	Type	Current Value
		Ext Key LtcMinus ans	%IX5.6	BIT	FALSE
		Ext Key Lt Plus ans	%IX5.7	BIT	FALSE
		Response Unit1	%IX6.0	BIT	TRUE
		Response Unit2	%IX6.1	BIT	TRUE
		Response Unit3	%IX6.2	BIT	TRUE
		Response Unit4	%IX6.3	BIT	FALSE

2 Setting up GM1 Controller

Step 1

Double-click **FullMode_44byte_I_10byte_O**, which has been added.
Open the **Module I/O Mapping** tab.



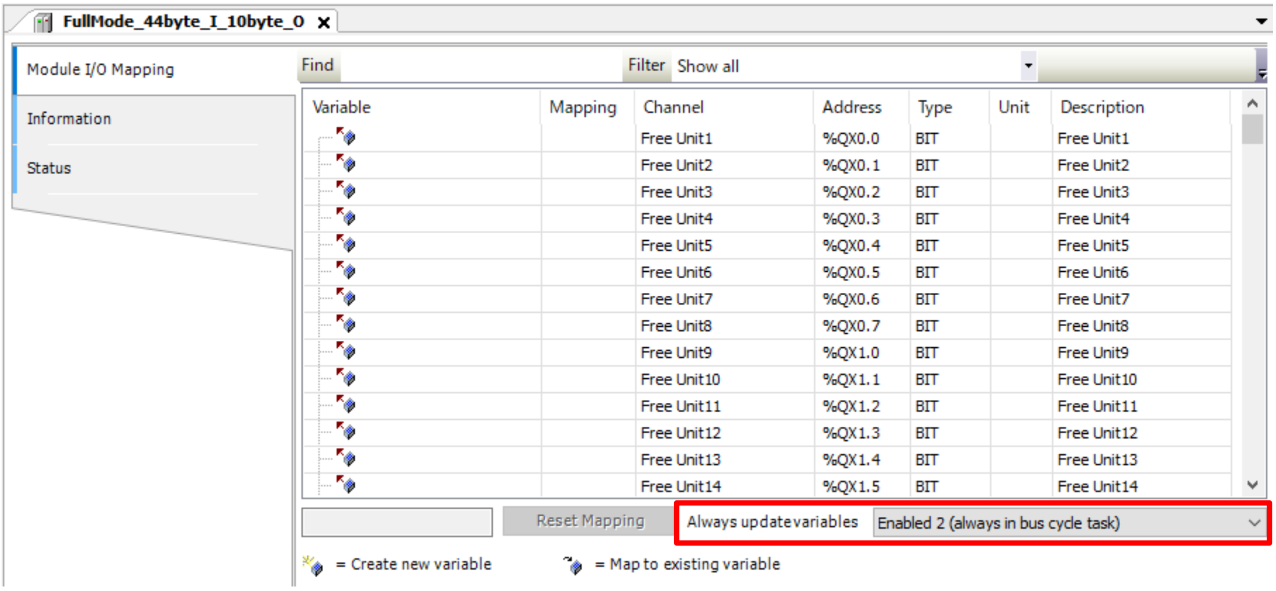
Filter: Show all

Variable	Mapping	Channel	Address	Type	Unit	Description
		Free Unit1	%QX0.0	BIT		Free Unit1
		Free Unit2	%QX0.1	BIT		Free Unit2
		Free Unit3	%QX0.2	BIT		Free Unit3
		Free Unit4	%QX0.3	BIT		Free Unit4
		Free Unit5	%QX0.4	BIT		Free Unit5
		Free Unit6	%QX0.5	BIT		Free Unit6
		Free Unit7	%QX0.6	BIT		Free Unit7
		Free Unit8	%QX0.7	BIT		Free Unit8
		Free Unit9	%QX1.0	BIT		Free Unit9
		Free Unit10	%QX1.1	BIT		Free Unit10
		Free Unit11	%QX1.2	BIT		Free Unit11
		Free Unit12	%QX1.3	BIT		Free Unit12
		Free Unit13	%QX1.4	BIT		Free Unit13
		Free Unit14	%QX1.5	BIT		Free Unit14

Always update variables: Use parent device setting

Step 2

In the **Always update variables** drop-down list, select **Enable 2 (always in bus cycle task)**.



Filter: Show all

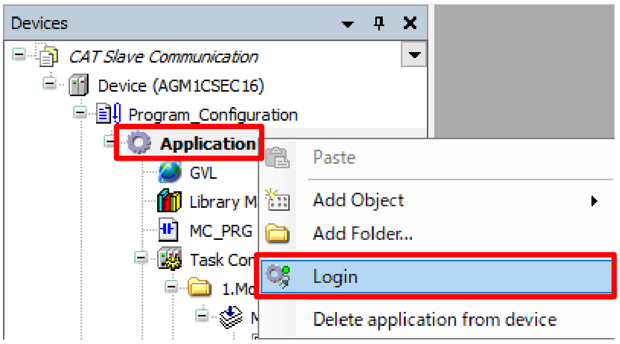
Variable	Mapping	Channel	Address	Type	Unit	Description
		Free Unit1	%QX0.0	BIT		Free Unit1
		Free Unit2	%QX0.1	BIT		Free Unit2
		Free Unit3	%QX0.2	BIT		Free Unit3
		Free Unit4	%QX0.3	BIT		Free Unit4
		Free Unit5	%QX0.4	BIT		Free Unit5
		Free Unit6	%QX0.5	BIT		Free Unit6
		Free Unit7	%QX0.6	BIT		Free Unit7
		Free Unit8	%QX0.7	BIT		Free Unit8
		Free Unit9	%QX1.0	BIT		Free Unit9
		Free Unit10	%QX1.1	BIT		Free Unit10
		Free Unit11	%QX1.2	BIT		Free Unit11
		Free Unit12	%QX1.3	BIT		Free Unit12
		Free Unit13	%QX1.4	BIT		Free Unit13
		Free Unit14	%QX1.5	BIT		Free Unit14

Always update variables: Enabled 2 (always in bus cycle task)

3 Checking Connection

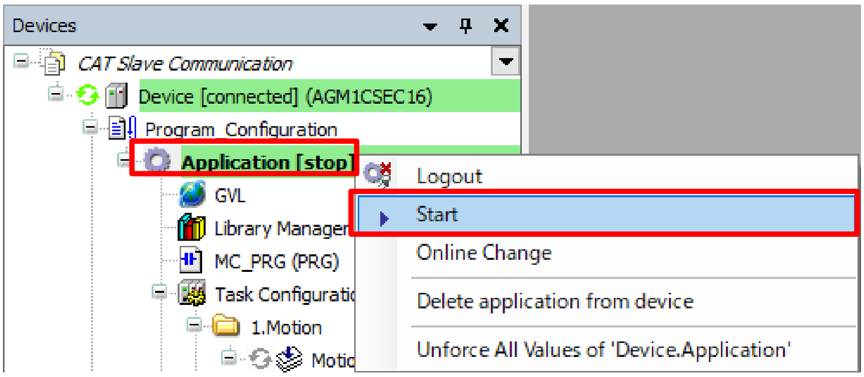
Step 1

Right-click **Application** and then select **Login** to write to the GM1 controller.



Step 2

Right-click **Application [stop]** and then select **Start** to switch the state from STOP to RUN.

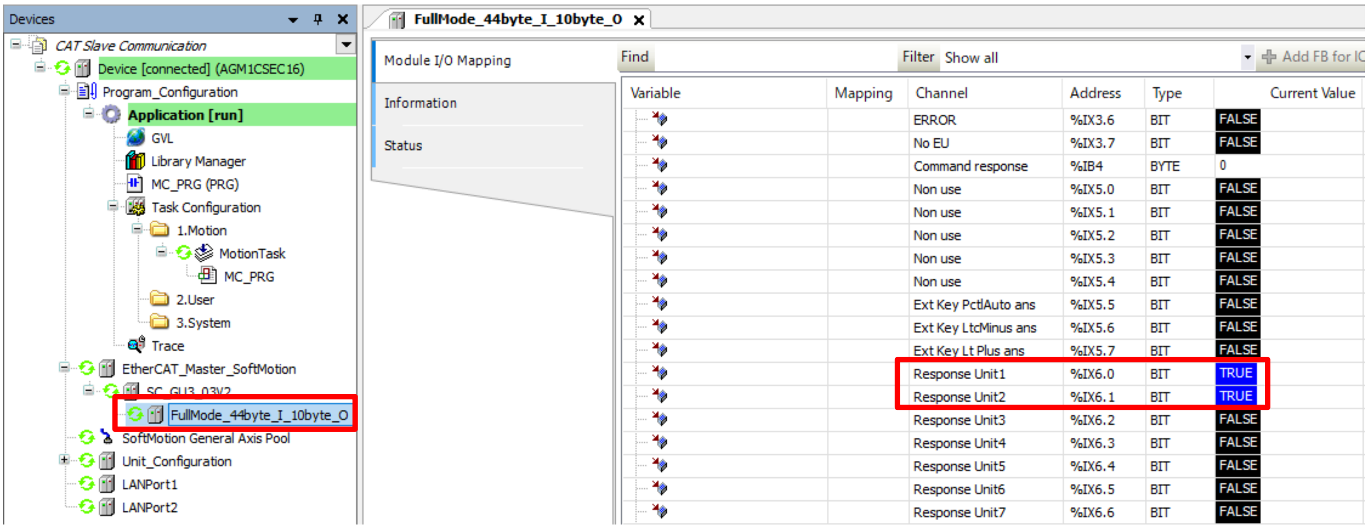


Step 3

Double-click **FullMode_44byte_I_10byte_O** and open the **Module I/O Mapping** tab.

In this example, two FX-501 units are connected, so **Response Unit1** and **Response Unit2** are set to TRUE.

This indicates that they have been connected.



Variable	Mapping	Channel	Address	Type	Current Value
		ERROR	%IX3.6	BIT	FALSE
		No EU	%IX3.7	BIT	FALSE
		Command response	%IB4	BYTE	0
		Non use	%IX5.0	BIT	FALSE
		Non use	%IX5.1	BIT	FALSE
		Non use	%IX5.2	BIT	FALSE
		Non use	%IX5.3	BIT	FALSE
		Non use	%IX5.4	BIT	FALSE
		Ext Key PctlAuto ans	%IX5.5	BIT	FALSE
		Ext Key LtcMinus ans	%IX5.6	BIT	FALSE
		Ext Key Lt Plus ans	%IX5.7	BIT	FALSE
		Response Unit1	%IX6.0	BIT	TRUE
		Response Unit2	%IX6.1	BIT	TRUE
		Response Unit3	%IX6.2	BIT	FALSE
		Response Unit4	%IX6.3	BIT	FALSE
		Response Unit5	%IX6.4	BIT	FALSE
		Response Unit6	%IX6.5	BIT	FALSE
		Response Unit7	%IX6.6	BIT	FALSE

Memo

Revision History

Date of issue	Manual code	Revision details
April 2022	AIM0010_01	First edition

Panasonic Industry Co., Ltd.
7-1-1 Morofuku, Daito City, Osaka, 574-0044, Japan
Industrial. Panasonic. com/ac/e

April 2022