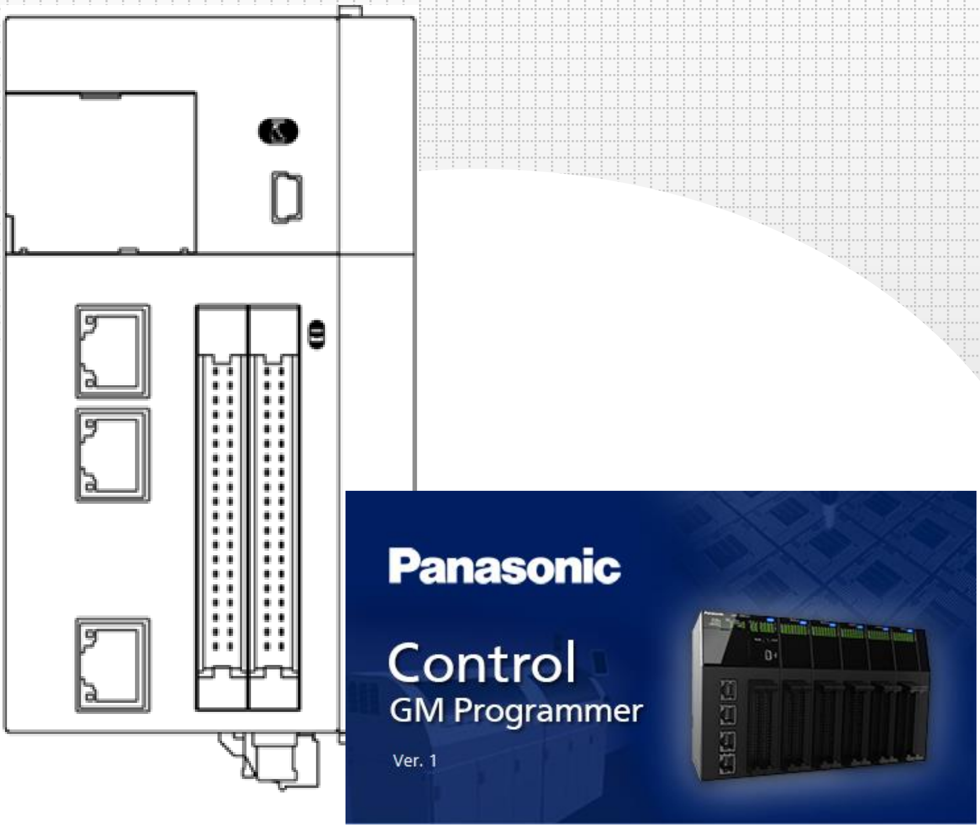
Panasonic®

Hello! GM1 Controller/Display Unit
Communication Edition



AIM0011_01

memo

Copyright and Trademarks

- The copyright of this manual is owned by Panasonic Industry Co., Ltd.
- Unauthorized reproduction of this manual is strictly prohibited.
- Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
- Ethernet is a registered trademark of Fuji Xerox Co., Ltd. and Xerox Corporation in the United States.
- EtherCAT is a registered trademark of and patented technology licensed by Beckhoff Automation GmbH in Germany.
- SDHC and SD logos are trademarks of SD-3C, LLC.
- Other company and product names are trademarks or registered trademarks of their respective companies.

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 Display Unit Communication

Installation Overview

- GM1 controller-compatible function codes

0 Preparation

Installing tool software

- GM Programmer
- xAscender Studio

1 Basic Setup

1.1 Behavior Overview
1.2 Preparing and Wiring the Required Devices

2 Slave Setup

2.1 Adding Slave Settings
2.2 Declaring Data Unit Type (DUT)
2.3 Declaring Global Variables
2.4 Setting Read/Write Variables and Login

3 Programming

3.1 Creating Modbus Program

4 WH Display Unit Setup

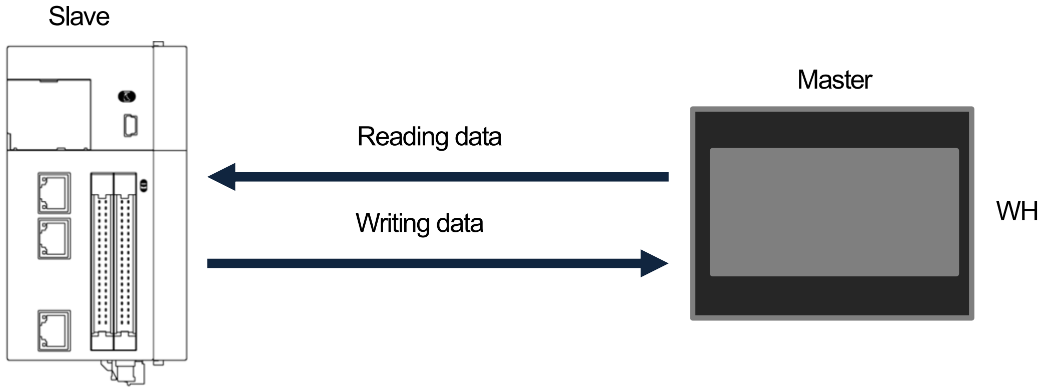
4.1 Setting IP Address
4.2 Downloading Screen Data

5 Communication Operation Check

5.1 Logging in to GM1 Controller
5.2 Operation Check

Installation Overview

Modbus TCP
The GM1 controller is used as s slave and the WH display unit is used as a master.
LAN Port 2 on each device is used to perform Modbus TCP master communication.

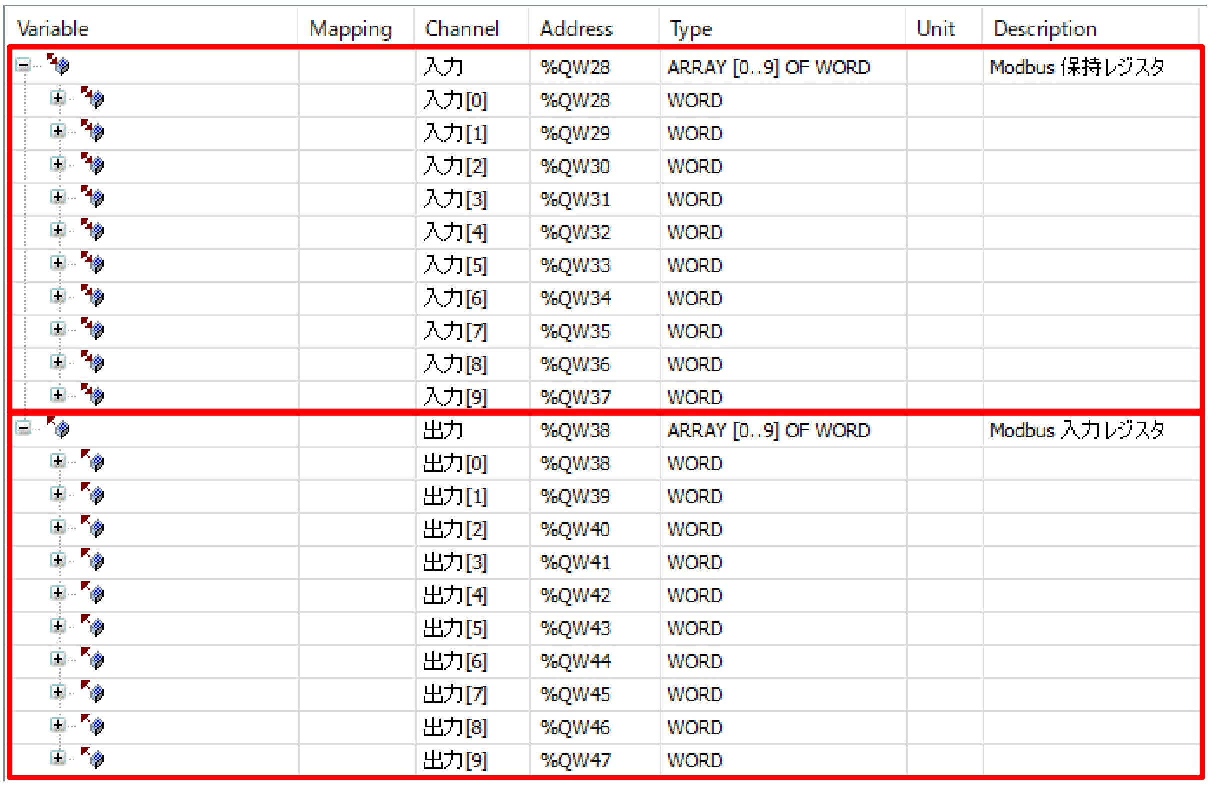


- ● GM1 controller-compatible function codes

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

･The display unit (master) writes and reads data to/from the GM1 controller (slave) via LAN Port 2.
Panasonic WH Series Built-In Web Server Programmable Display is used as the display unit.

･Input and output registers are allocated separately to the GM1 controller used as a Modbus TCP slave.



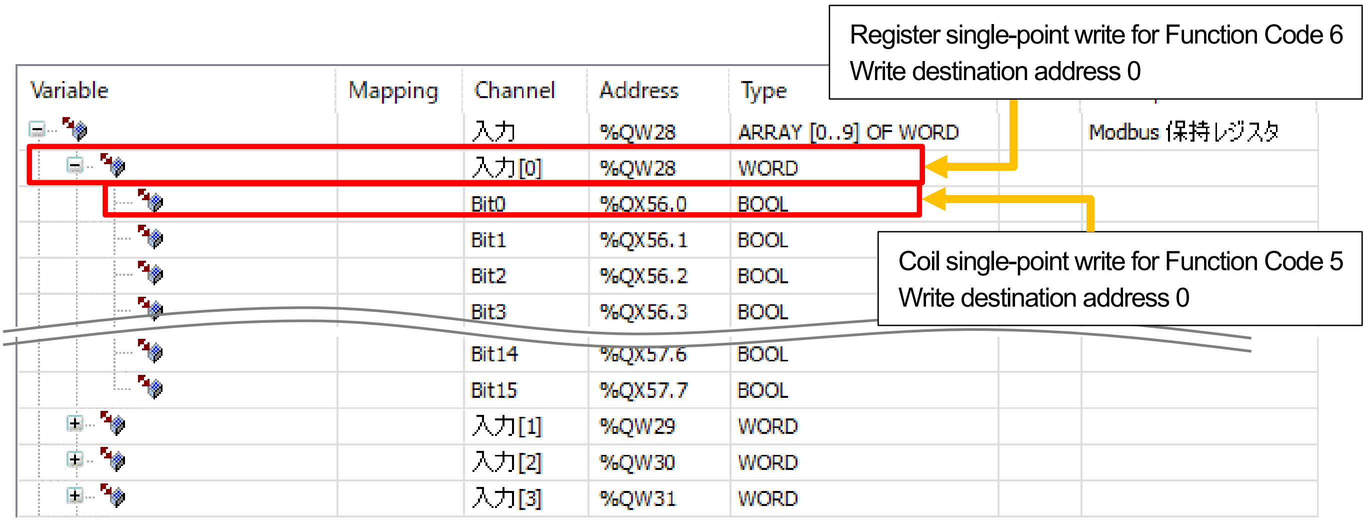
Variable	Mapping	Channel	Address	Type	Unit	Description
		入力	%QW28	ARRAY [0..9] OF WORD		Modbus 保持レジスタ
		入力[0]	%QW28	WORD		
		入力[1]	%QW29	WORD		
		入力[2]	%QW30	WORD		
		入力[3]	%QW31	WORD		
		入力[4]	%QW32	WORD		
		入力[5]	%QW33	WORD		
		入力[6]	%QW34	WORD		
		入力[7]	%QW35	WORD		
		入力[8]	%QW36	WORD		
		入力[9]	%QW37	WORD		
		出力	%QW38	ARRAY [0..9] OF WORD		Modbus 入力レジスタ
		出力[0]	%QW38	WORD		
		出力[1]	%QW39	WORD		
		出力[2]	%QW40	WORD		
		出力[3]	%QW41	WORD		
		出力[4]	%QW42	WORD		
		出力[5]	%QW43	WORD		
		出力[6]	%QW44	WORD		
		出力[7]	%QW45	WORD		
		出力[8]	%QW46	WORD		
		出力[9]	%QW47	WORD		

Write/read target areas used as **Modbus holding registers**: Function Code 1 / 3 / 5 / 6 / 15 / 16 / 23
Read target areas used as **Modbus input registers**: Function Code 2 / 4

Column (1): I/O mapping

Because the write and read areas for coils and registers are shared for I/O mapping, care must be taken when specifying write and read destinations from the master device.

In the following example, data is written from the master device. For the coil single-point write area for Function Code 5 and the register single-point write area for Function Code 6, when "0" is specified as the write destination address, data is written to the following addresses on the I/O map.



Register single-point write for Function Code 6
Write destination address 0

Coil single-point write for Function Code 5
Write destination address 0

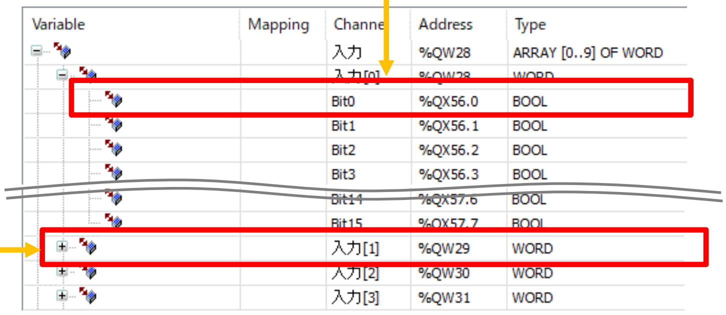
Variable	Mapping	Channel	Address	Type		
		入力	%QW28	ARRAY [0..9] OF WORD		Modbus 保持レジスタ
		入力[0]	%QW28	WORD		
		Bit0	%QX56.0	BOOL		
		Bit1	%QX56.1	BOOL		
		Bit2	%QX56.2	BOOL		
		Bit3	%QX56.3	BOOL		
		Bit14	%QX57.6	BOOL		
		Bit15	%QX57.7	BOOL		
		入力[1]	%QW29	WORD		
		入力[2]	%QW30	WORD		
		入力[3]	%QW31	WORD		

If the write destination addresses are specified as above, the register write area and coil write area will be duplicately specified as below.
This is because Bit0 is handled as bit 0 of Input [0].

Input	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When using a unique coil and register, specify I/O mapping as below.
･Coil write: Input [0] – Bit0
･Register write: Input [1]

Input	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input	1															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

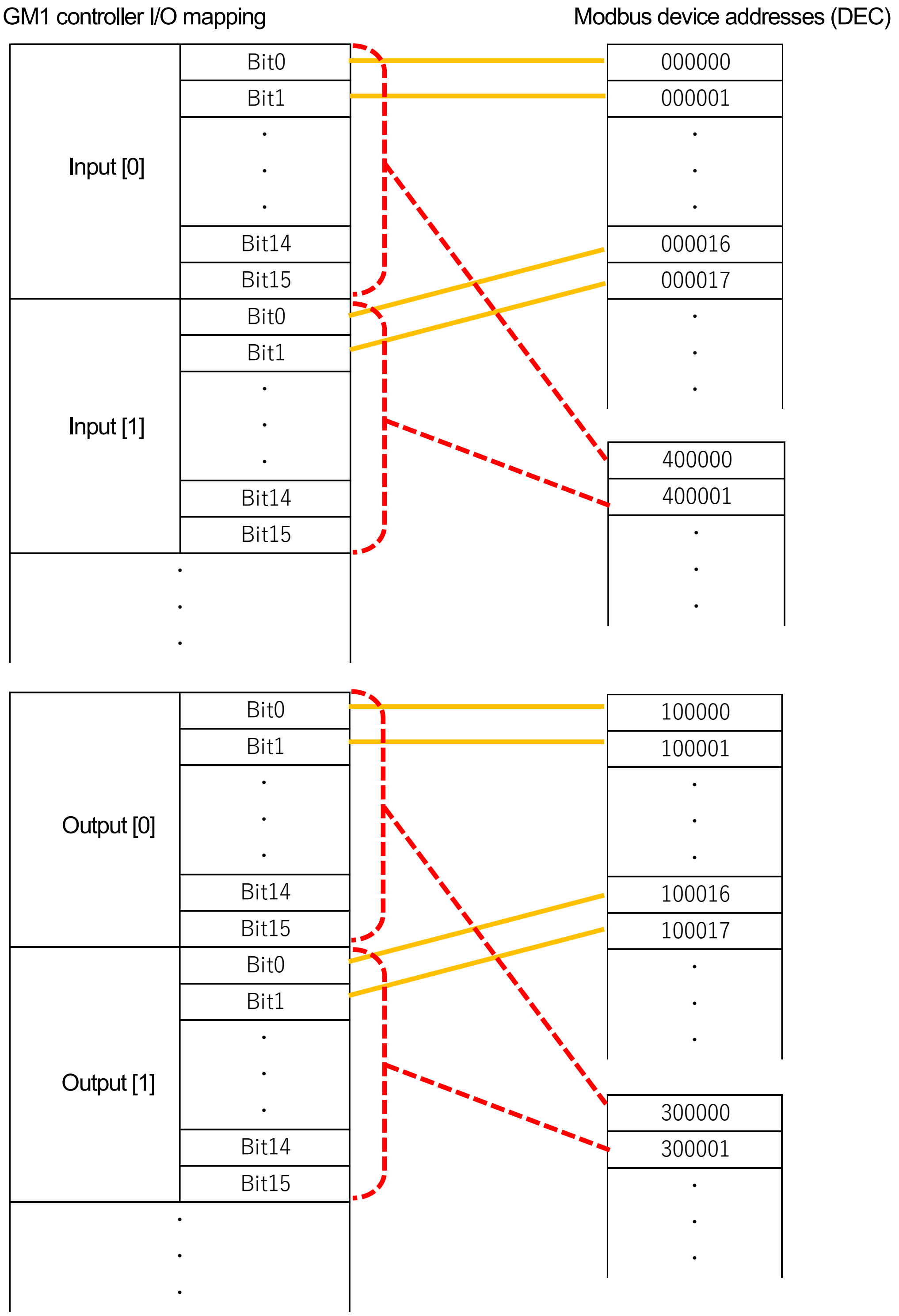


Variable	Mapping	Channe	Address	Type
		入力	%QW28	ARRAY [0..9] OF WORD
		入力[0]	%QW28	WORD
		Bit0	%QX56.0	BOOL
		Bit1	%QX56.1	BOOL
		Bit2	%QX56.2	BOOL
		Bit3	%QX56.3	BOOL
		Bit14	%QX57.6	BOOL
		Bit15	%QX57.7	BOOL
		入力[1]	%QW29	WORD
		入力[2]	%QW30	WORD
		入力[3]	%QW31	WORD

Column (2): Modbus device addresses

When accessing the GM1 controller by using a WH series display unit as the Modbus master, use Modbus device addresses. Even if I/O mapping is specified using Modbus device addresses, duplicate read/write areas for coils and registers must be considered.

Overview of Modbus device address allocation



* Modbus device addresses "0*****" and "1*****" store bit information and "3*****" and "4*****" store word information.

0 Preparation

Installing tool software

Install GM Programmer and xAscender Studio from the following websites.
GM Programmer : https://industrial.panasonic.com/ac/j/motor/motion-controller/mc/gm1/index.jsp
xAscender Studio : https://www3.panasonic.biz/ac/j/dl/software/index.jsp?series_cd=3854

INFO

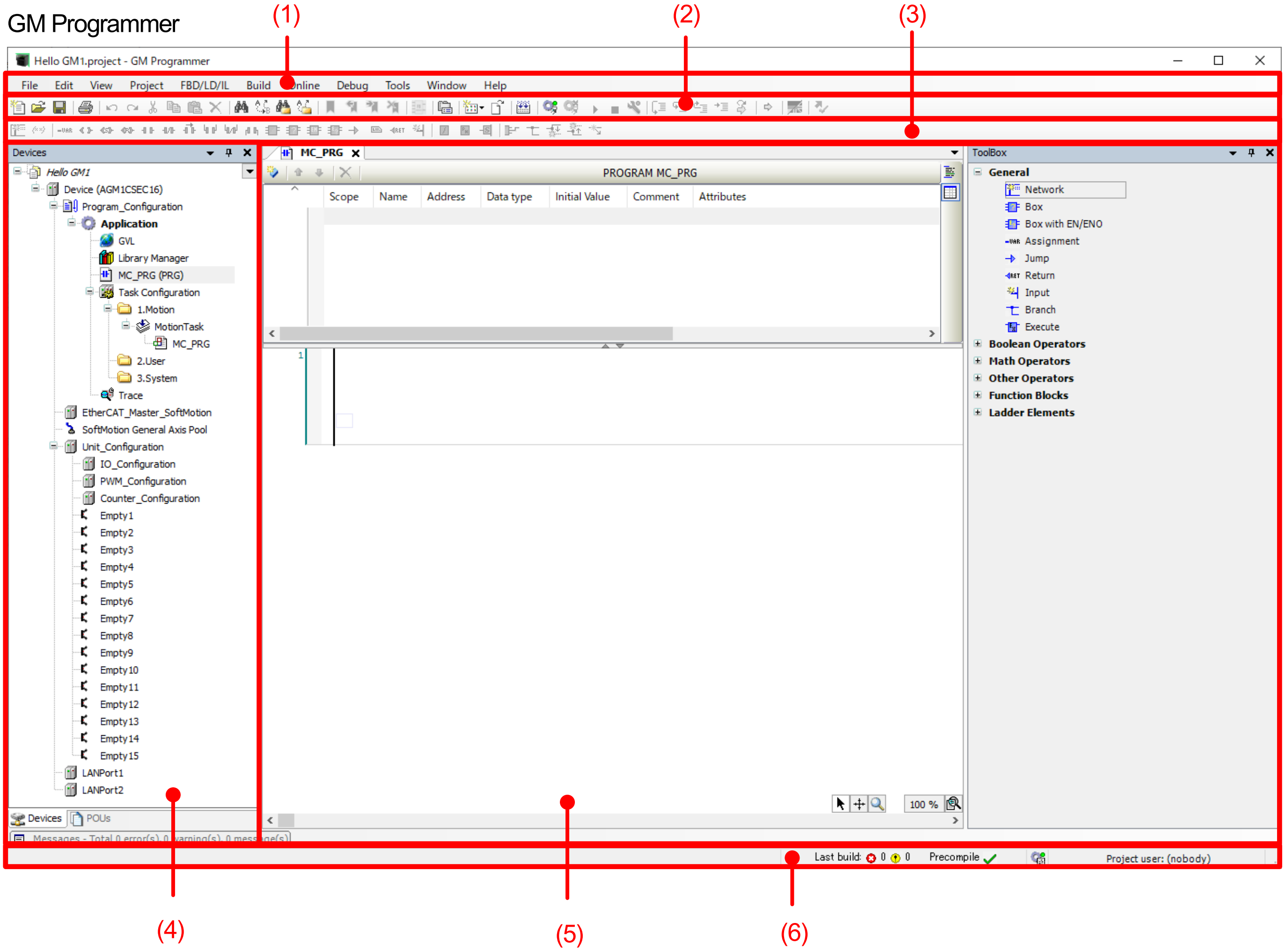
When GM Programmer is installed, PANATERM Lite for GM, Gateway (CODESYS Gateway), and CodeMeter applications are installed at the same time.

- 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: 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.

Column (3): Components of each tool software product

GM Programmer



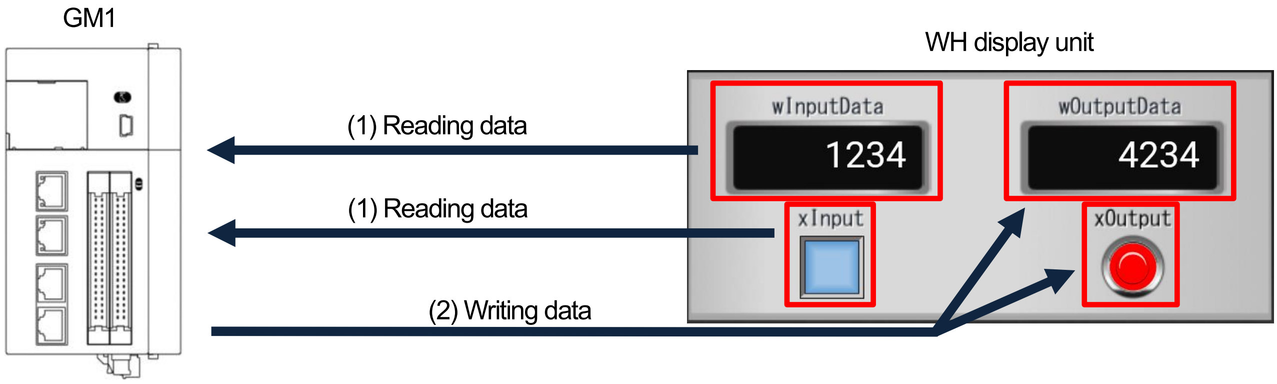
No.	Name	Description
(1)	Title bar	The title bar displays the project file name, minimize button, maximize button, and close button.
(2)	Menu bar	The menu bar displays the menu commands for each purpose in list format.
(3)	Toolbar	The toolbar displays each command as an icon.
(4)	Navigation pane	The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

1 Basic Setup

1.1 Behavior Overview

(1) The GM1 controller reads the value entered in the **wInputData** field in the WH display unit.

(2) When the **xInput** button in the WH display unit is pressed, a value that is the value entered in the **wInputData** field plus 3000 is written to the **wOutputData** field. Output is written to **xOutput** at the same time.

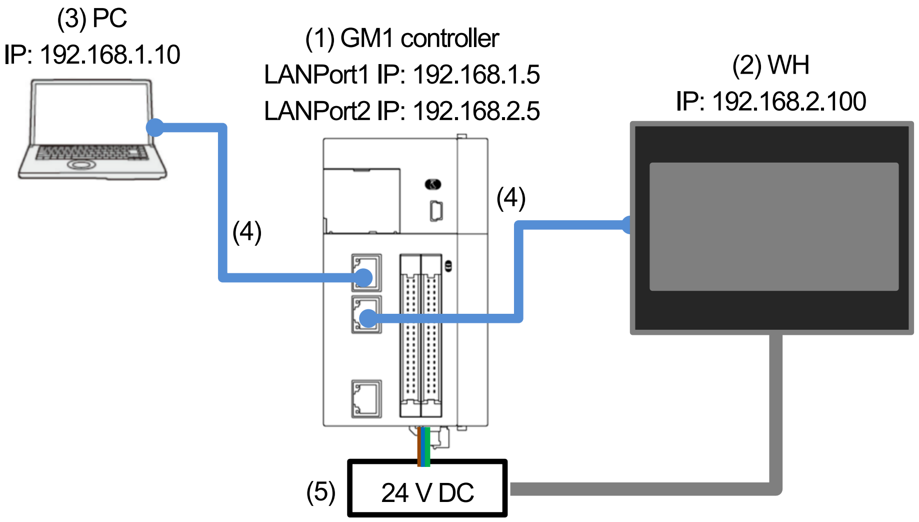


1.2 Preparing and Wiring the Required Devices

Prepare the following devices.

No.	Name
(1)	GM1 controller
(2)	Display unit WH series (Resistive touch screen: 4.3/7/10 inch type)
(3)	PC (with GM Programmer and xAscender Studio installed)
(4)	LAN cable: 2 pcs.
(5)	24 V DC power supply

Wire each device as shown below.



The PC communicates with the GM1 controller via LAN ports.
The IP address (default) of LAN port 1 is set as below.
An IP address on the same network that is different from the IP address of LAN port 1 must be set for the PC.

Interface: LAN port 1

IP address:	192.168.1.5
Subnet mask:	255.255.255.0
Default gateway:	192.168.1.1

Interface: PC

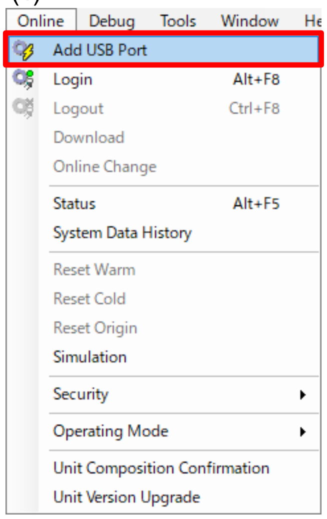
IP address:	192.168.1.10
Subnet mask:	255.255.255.0
Default gateway:	192.168.1.1

Column (4): Communication using USB ports

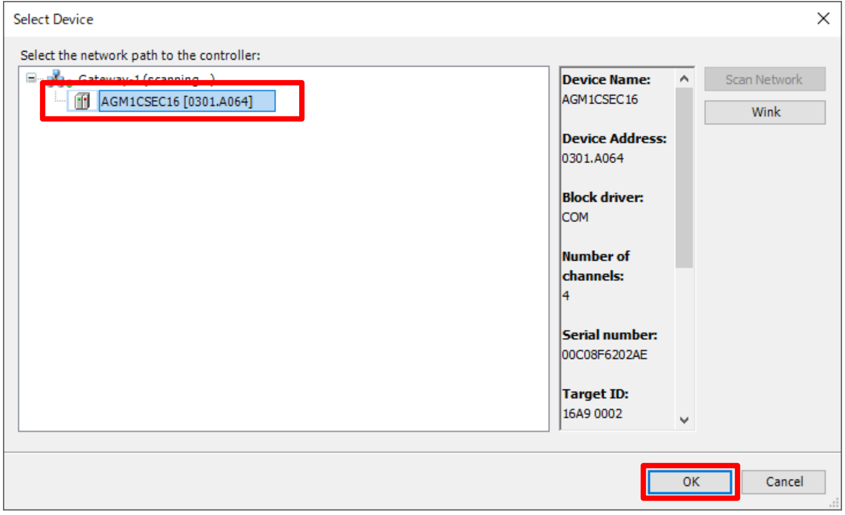
USB ports can be set as a communication interface between the GM1 controller and tool software such as GM Programmer or PANATERA Lite for GM, as below.

(1) Select **Add USB Port** from the **Online** menu.
(2) The **Add USB Port** dialog box will be displayed.
(3) Click **OK** to display a dialog box that restarts the gateway.
(4) Click **OK** to display the **Select Device** dialog box.
(5) Select a GM1 controller that you want to connect and click **OK**.
(6) When a connection is established, USB ports are added as a communication interface between the PC and GM1 controller.

(1)



(5)



Device Name:
AGM1CSEC16

Device Address:
0301.A064

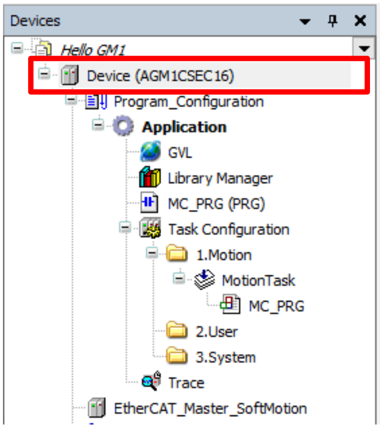
Block driver:
COM

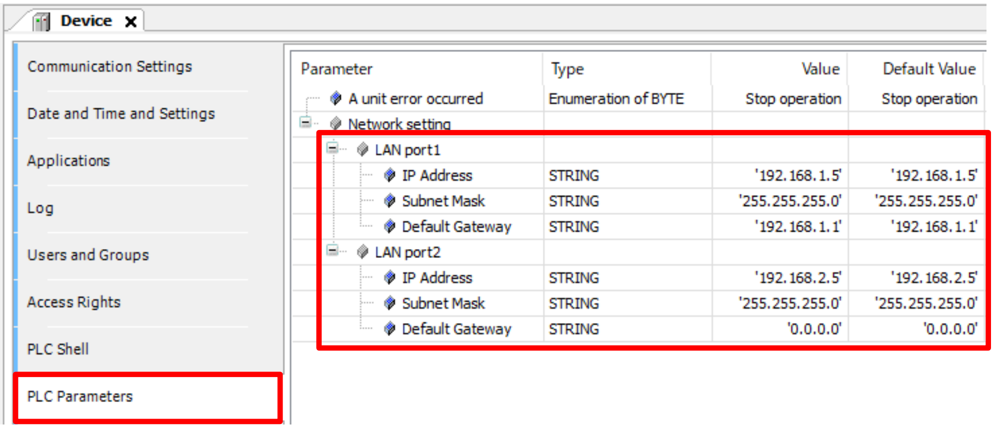
Number of channels:
4

Serial number:
00C08F6202AE

Target ID:
16A9 0002

* The IP address of the GM1 controller can be checked and set in the main pane displayed by selecting **PLC Parameters** in the **Device** tab.

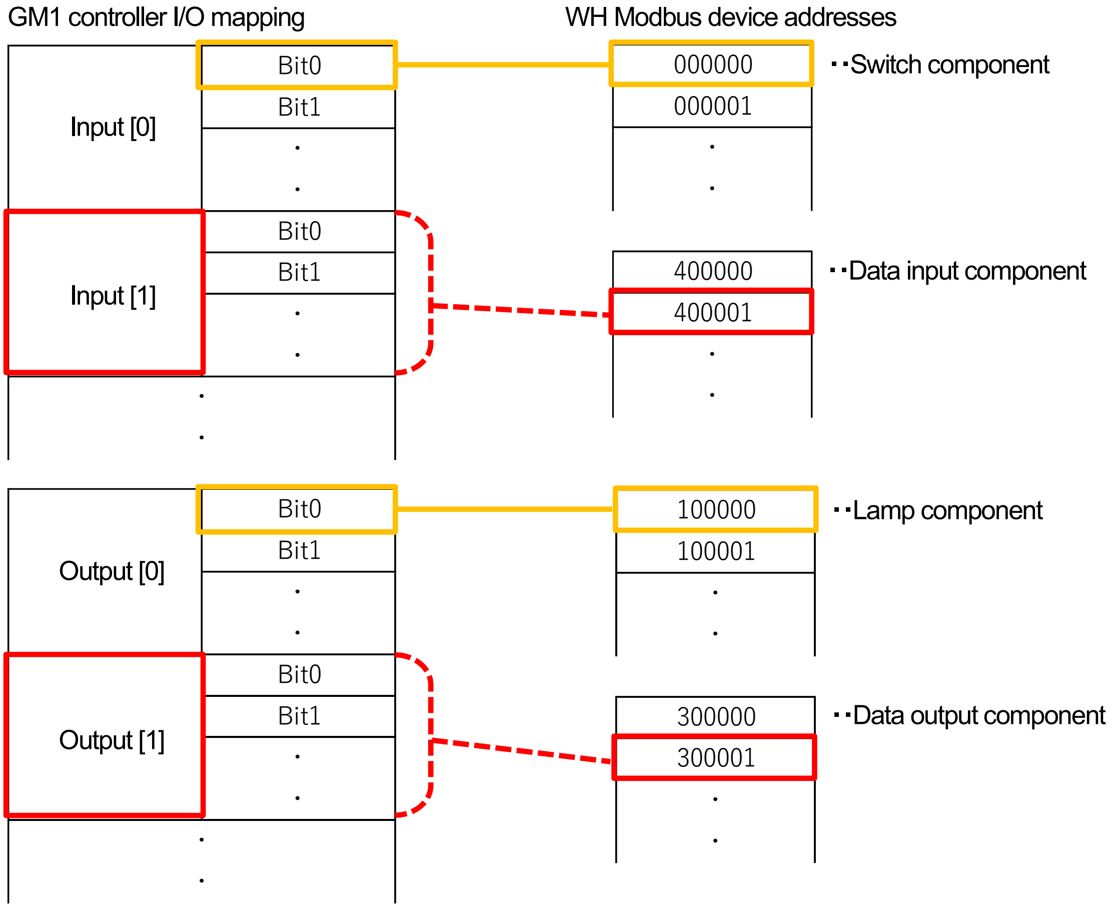




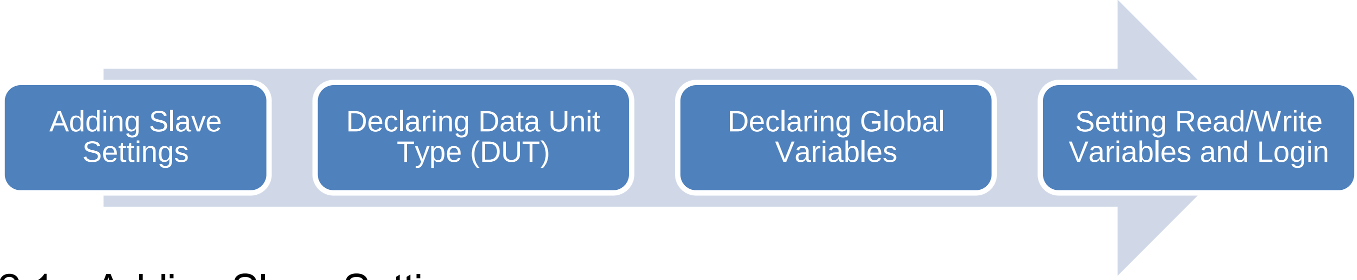
Parameter	Type	Value	Default Value
A unit error occurred	Enumeration of BYTE	Stop operation	Stop operation
Network setting			
LAN port1			
IP Address	STRING	'192.168.1.5'	'192.168.1.5'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'192.168.1.1'	'192.168.1.1'
LAN port2			
IP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

INFO

Overview of Modbus device address allocation



2 Slave Setup

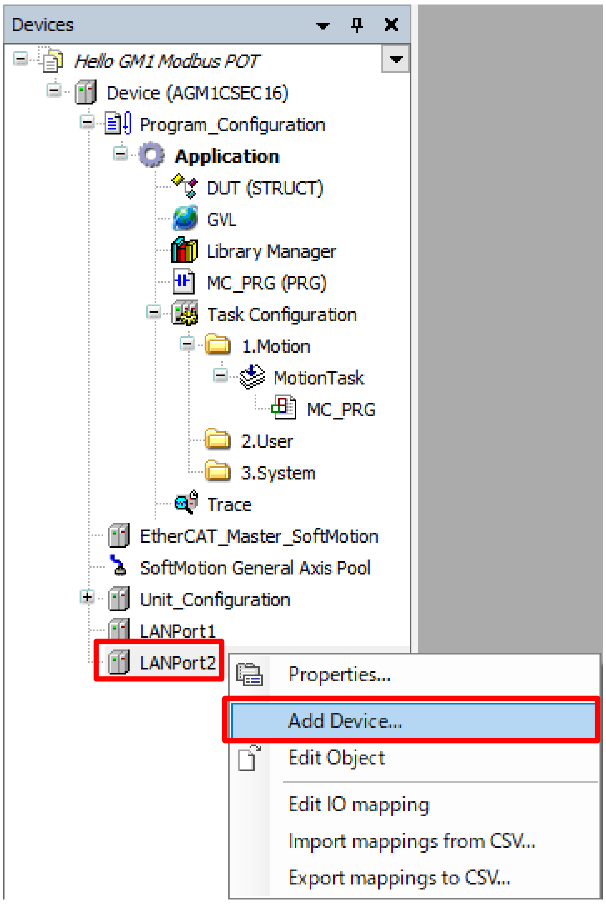

Adding Slave Settings
Declaring Data Unit Type (DUT)
Declaring Global Variables
Setting Read/Write Variables and Login

2.1 Adding Slave Settings

Step 1

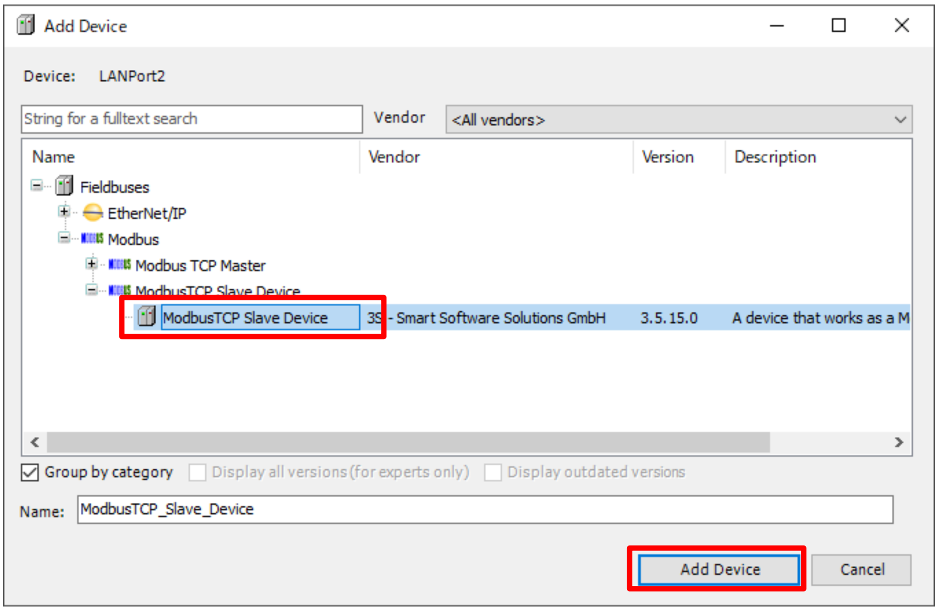
Open GM Programmer and right-click **LANPort2**.

Select **Add Device** to add a device.



Step 2

Select **Modbus**, **ModbusTCP Slave Device**, and then **ModbusTCP Slave Device** and click **Add Device**.



Device: LANPort2

String for a fulltext search

Vendor <All vendors>

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)

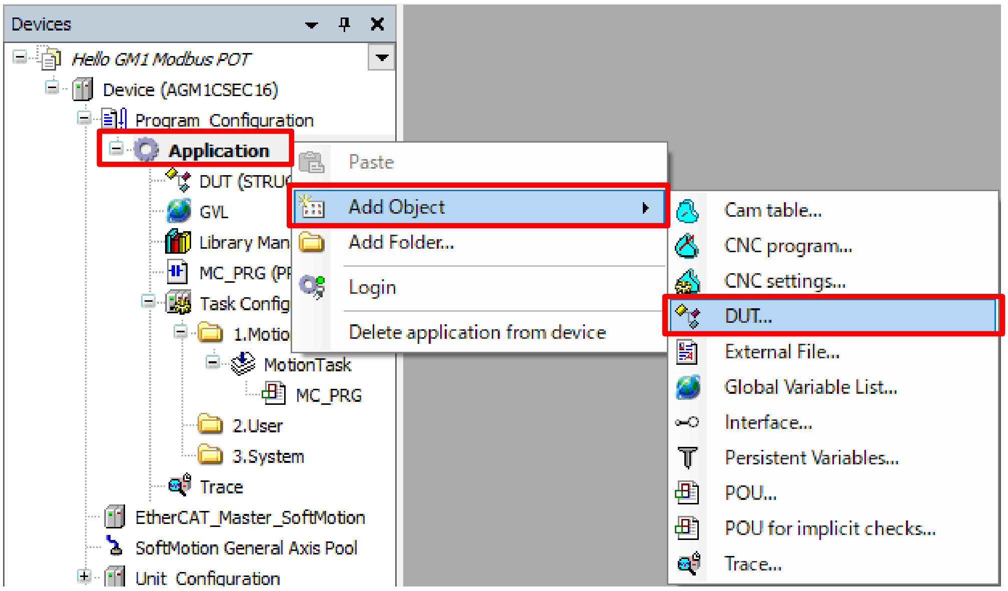
INFO

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

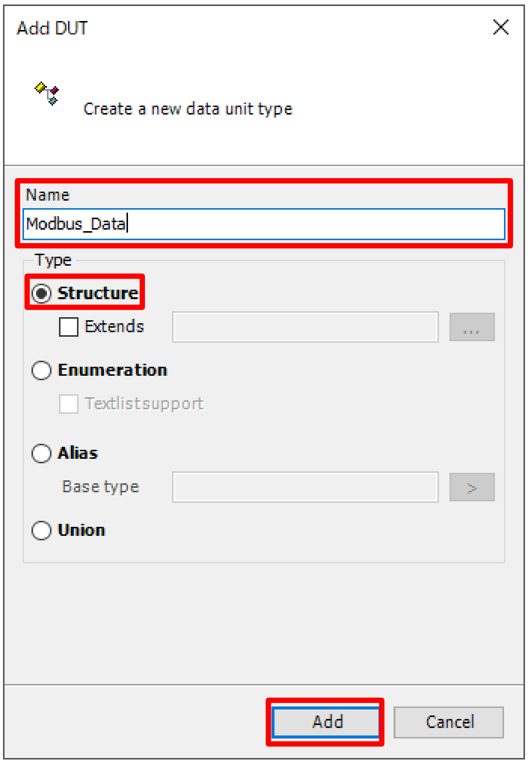
Right-click **Application** and select **Add Object** and then **DUT**.



Step 2

The **Add DUT** dialog box will be displayed. Assign any desired name.

Type: Select the **Structure** check box and click **Add**.


Add DUT

Create a new data unit type

Name: Modbus_Data

Type

● Structure

□ Extends

○ Enumeration

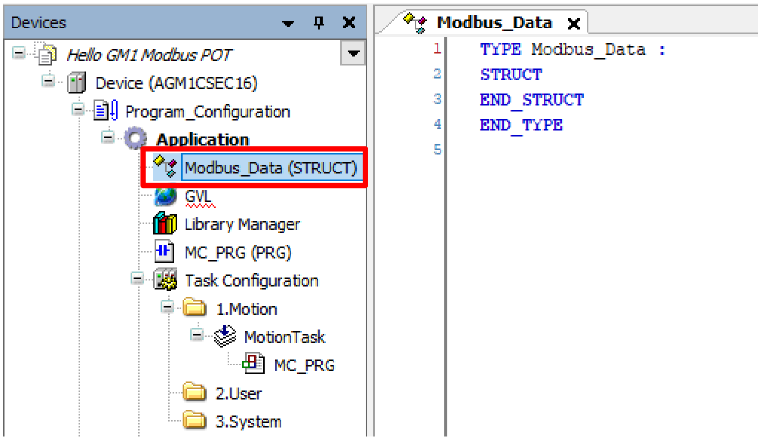
□ Textlistsupport

○ Alias

Base type

○ Union

Modbus_Data (STRUCT) will be added.

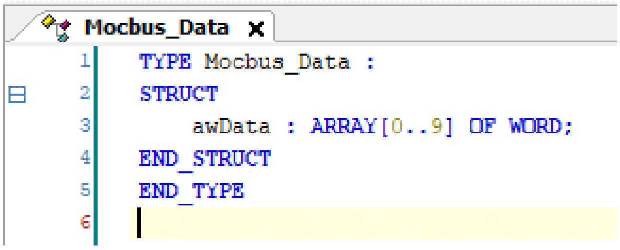


```
TYPE Modbus_Data :
STRUCT
END_STRUCT
END_TYPE
```

Step 3

Configure a DUT as shown in the figure below.

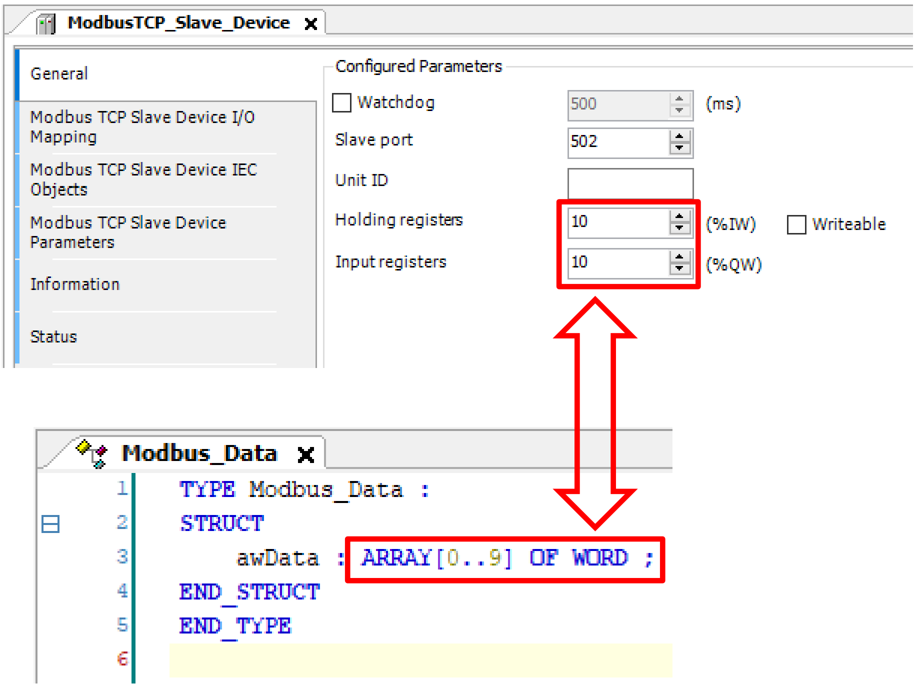
Variable name	Data type
awData	ARRAY[0..9]OF WORD



```
TYPE Mocbus_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 registers 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.

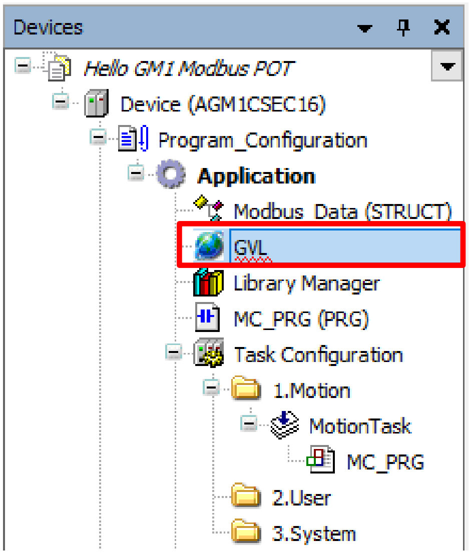
2.3 Declaring Global Variables

INFO

Global variables can be used throughout the entire project.

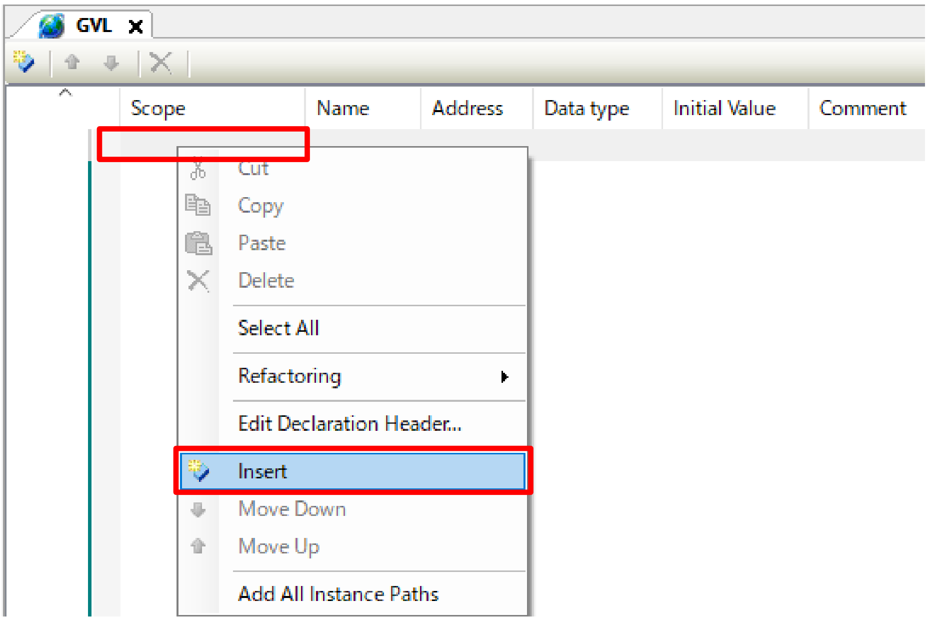
Step 1

Double-click **GVL**.



Step 2

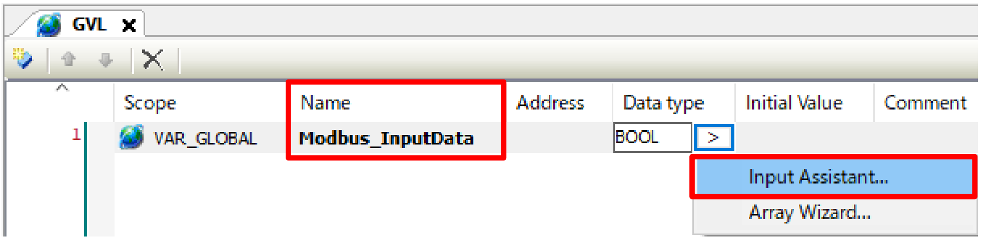
Right-click in the gray area and select **Insert**.



Step 3

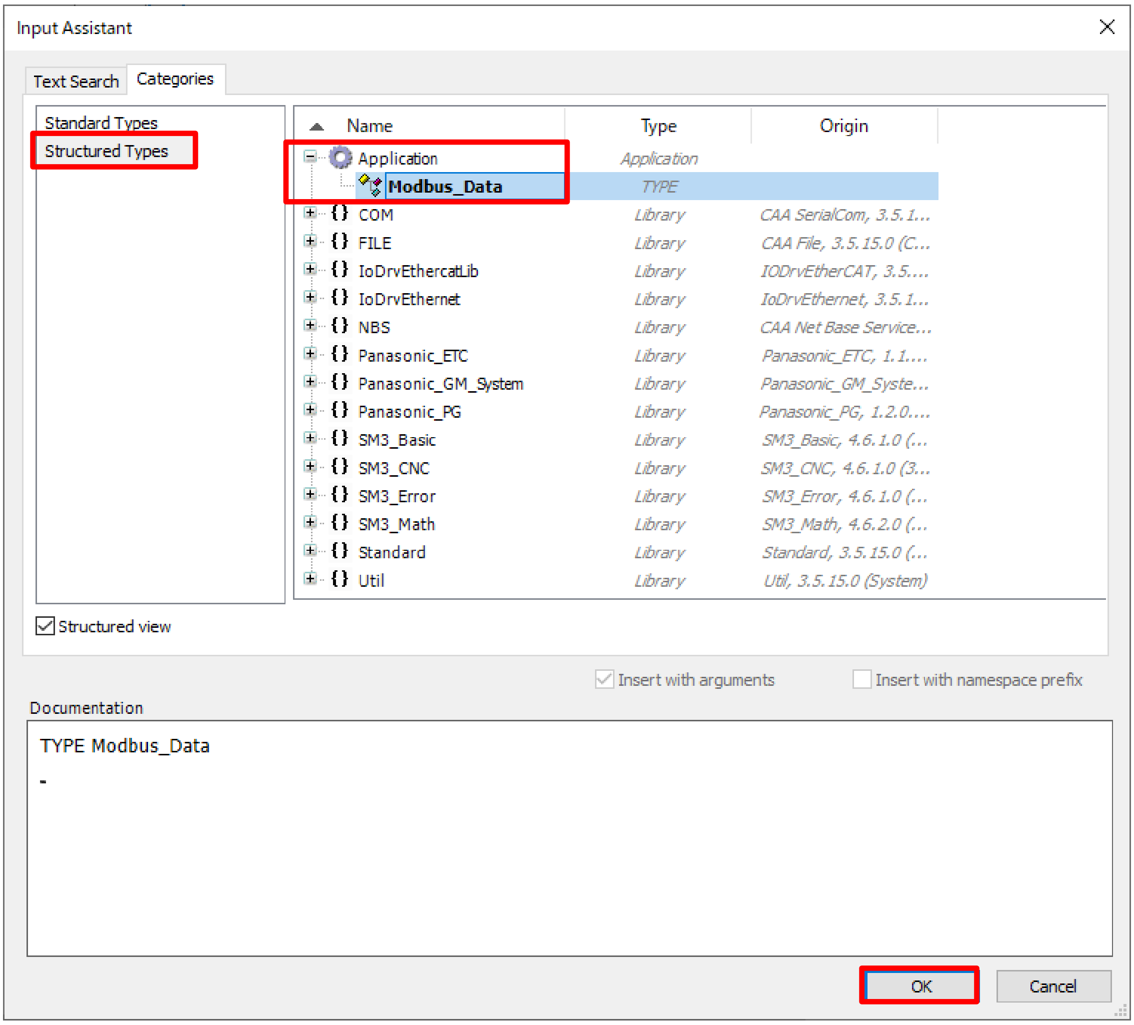
Enter "Modbus_InputData" in the **Name** column.

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



Step 4

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



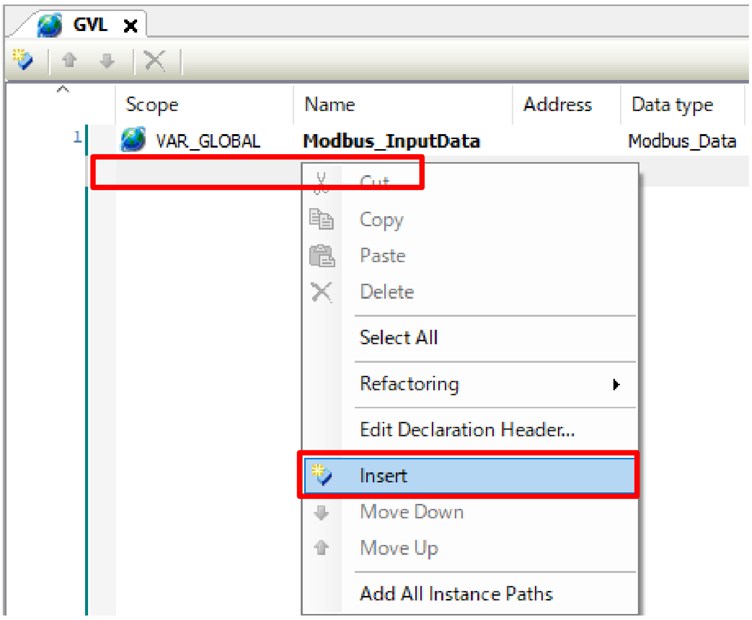
Name	Type	Origin
Application	Application	
Modbus_Data	TYPE	
COM	Library	CAA SerialCom, 3.5.1...
FILE	Library	CAA File, 3.5.15.0 (C...
IoDrvEthercatLib	Library	IODrvEtherCAT, 3.5....
IoDrvEthernet	Library	IoDrvEthernet, 3.5.1...
NBS	Library	CAA Net Base Service...
Panasonic_ETC	Library	Panasonic_ETC, 1.1....
Panasonic_GM_System	Library	Panasonic_GM_Syste...
Panasonic_PG	Library	Panasonic_PG, 1.2.0....
SM3_Basic	Library	SM3_Basic, 4.6.1.0 (...
SM3_CNC	Library	SM3_CNC, 4.6.1.0 (3...
SM3_Error	Library	SM3_Error, 4.6.1.0 (...
SM3_Math	Library	SM3_Math, 4.6.2.0 (...
Standard	Library	Standard, 3.5.15.0 (...
Util	Library	Util, 3.5.15.0 (System)

TYPE Modbus_Data

-

Step 5

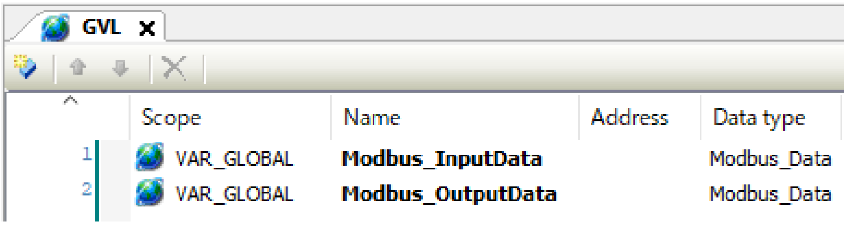
Right-click in the area below **Modbus_InputData**, which has been inserted, and select **Insert**.



	Scope	Name	Address	Data type
1	VAR_GLOBAL	Modbus_InputData		Modbus_Data

Step 6

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



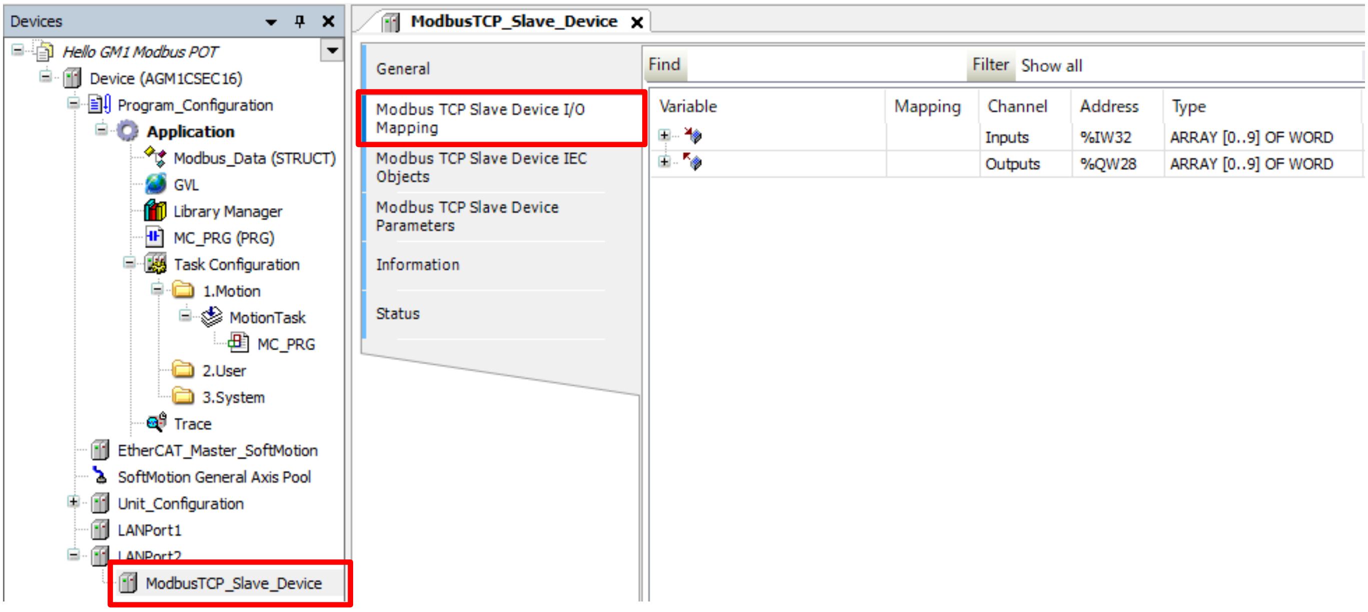
	Scope	Name	Address	Data type
1	VAR_GLOBAL	Modbus_InputData		Modbus_Data
2	VAR_GLOBAL	Modbus_OutputData		Modbus_Data

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 open **Modbus TCP Slave Device I/O Mapping**.

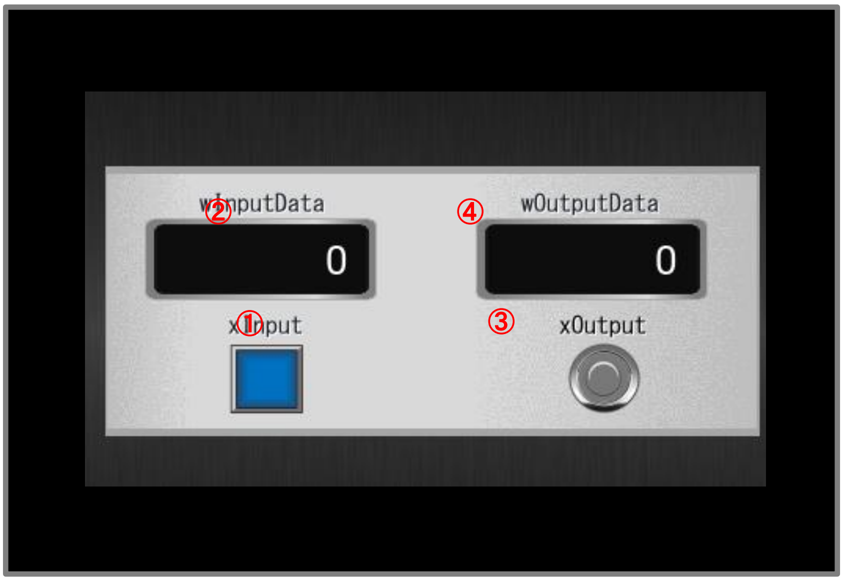


Variable	Mapping	Channel	Address	Type
		Inputs	%IW32	ARRAY [0..9] OF WORD
		Outputs	%QW28	ARRAY [0..9] OF WORD

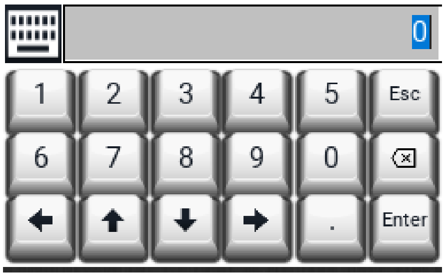
INFO

In this textbook, the GM1 controller uses the WH display unit as a master in Modbus slave communication.
The WH display unit uses the four components (① to ④) in the figure below.
Different functions are assigned to each component to send different commands.

WH display unit



*

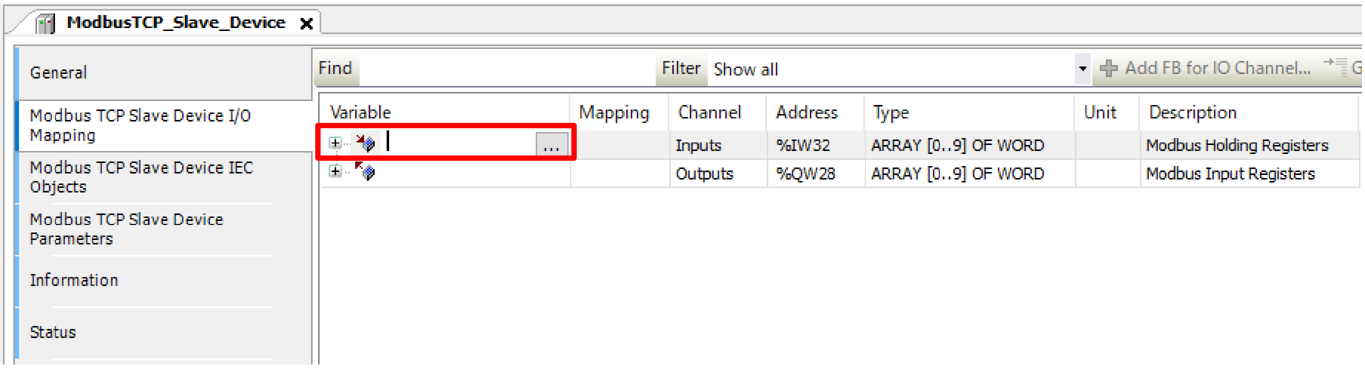


	Modbus device address	Function Code	Write destination address
① Switch	000000	15 (Write Multiple Coils)	0
② Data input	400001	16 (Write Multiple Registers)	1
③ Lamp	100000	2 (Read Discrete Inputs)	0
④ Data input	300001	4 (Read Input Registers)	1

* When a numerical value is entered from the keyboard in "② Data input", the value is displayed by tapping numeric keys on the keyboard.

Step 2

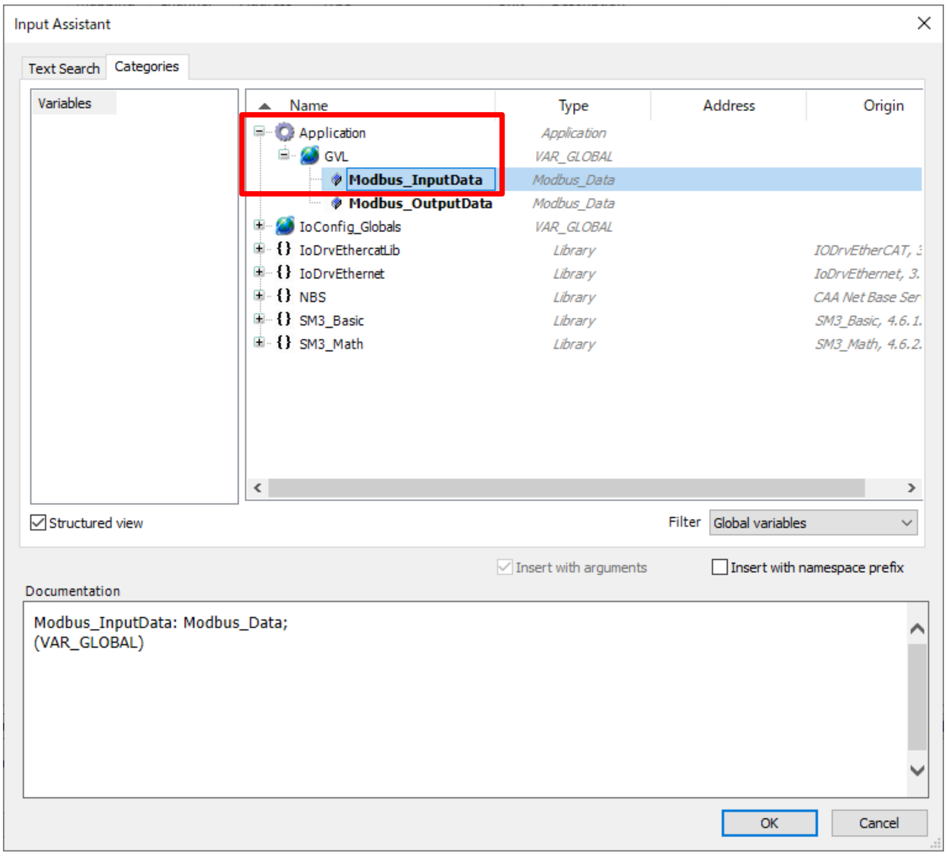
Assign a variable to the address written from the WH display unit.
Assign a variable to "① Switch".
Double-click the inside of the red frame in the **Variable** column and then click ... .


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...
...

Variable	Mapping	Channel	Address	Type	Unit	Description
		Inputs	%IW32	ARRAY [0..9] OF WORD		Modbus Holding Registers
		Outputs	%QW28	ARRAY [0..9] OF WORD		Modbus Input Registers

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	Origin
Application	Application		
GVL	VAR_GLOBAL		
Modbus_InputData	Modbus_Data		
Modbus_OutputData	Modbus_Data		
IoConfig_Globals	VAR_GLOBAL		
IoDrvEthercatLib	Library		IODrvEtherCAT, 3
IoDrvEthernet	Library		IoDrvEthernet, 3.
NBS	Library		CAA Net Base Ser
SM3_Basic	Library		SM3_Basic, 4.6.1.
SM3_Math	Library		SM3_Math, 4.6.2.

☑ Structured view

Filter: Global variables

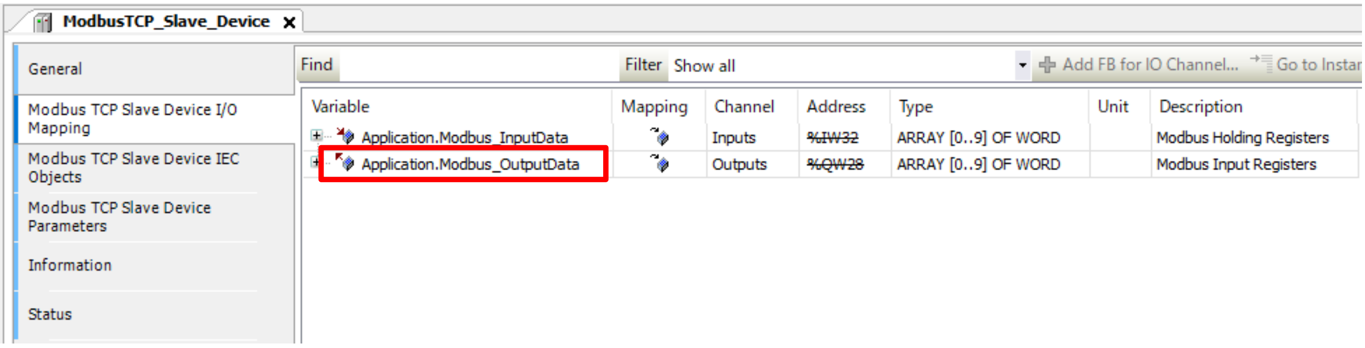
☑ Insert with arguments ☐ Insert with namespace prefix

Documentation

```
Modbus_InputData: Modbus_Data;
(VAR_GLOBAL)
```

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
Add FB for IO Channel...
Go to Instar

Filter: Show all

Variable	Mapping	Channel	Address	Type	Unit	Description
Application.Modbus_InputData		Inputs	~~%IW32~~	ARRAY [0..9] OF WORD		Modbus Holding Registers
Application.Modbus_OutputData		Outputs	~~%QW28~~	ARRAY [0..9] OF WORD		Modbus Input Registers

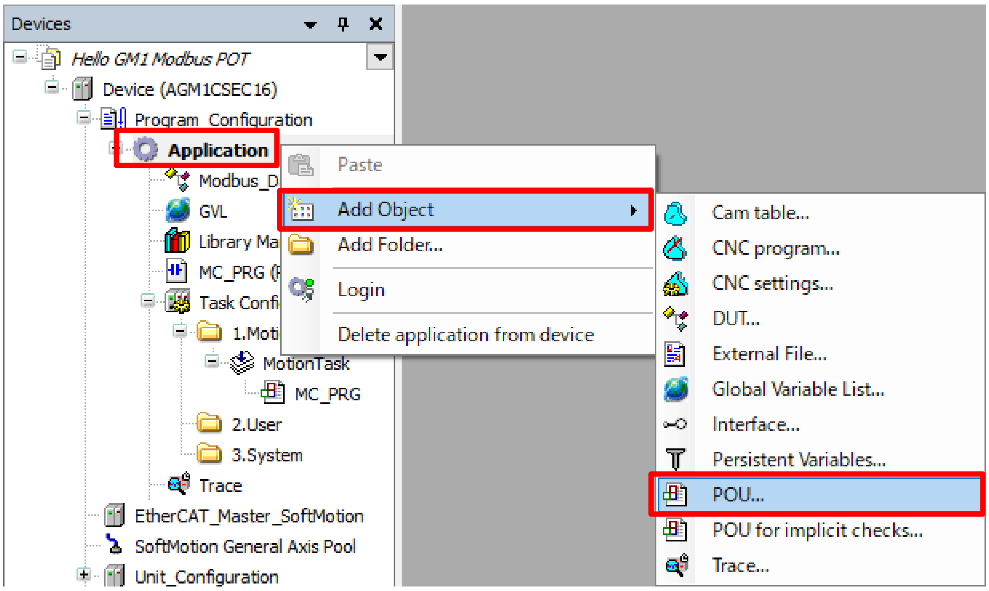
3 Programming

3.1 Creating Modbus Program

Add a program that outputs the results of calculation processing to the read destination of the WH display unit in response to the numerical value written from the WH display unit.

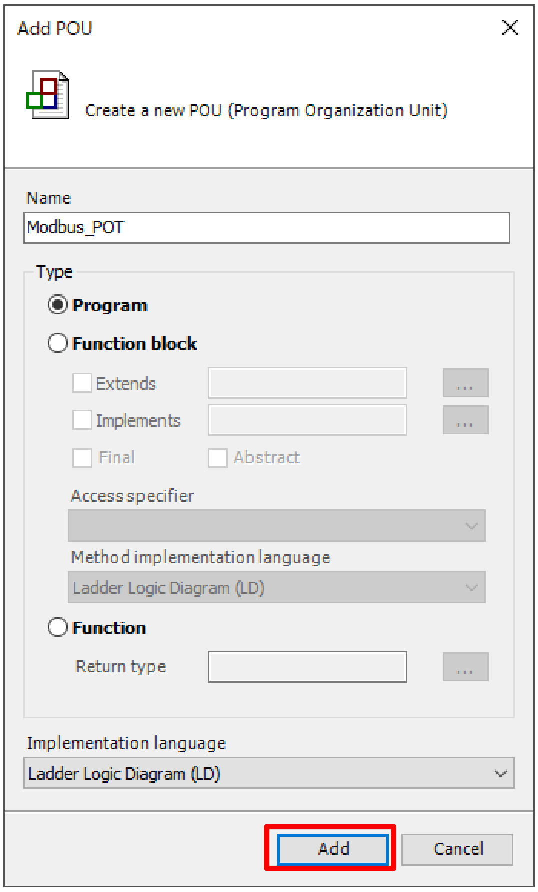
Step 1

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



Step 2

Enter any desired name and click **Add**.



Add POU

Create a new POU (Program Organization Unit)

Name: Modbus_POT

Type

● Program

○ Function block

□ Extends

□ Implements

□ Final □ Abstract

Access specifier

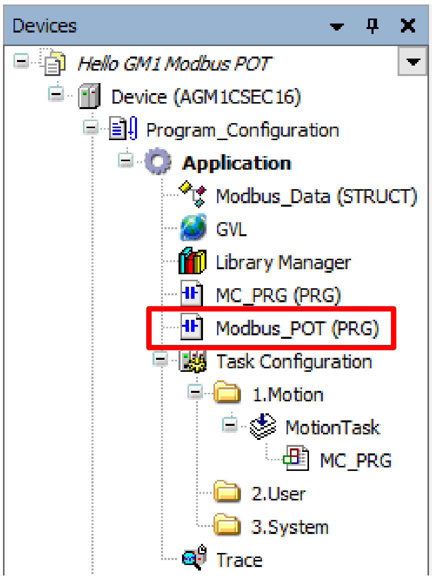
Method implementation language: Ladder Logic Diagram (LD)

○ Function

Return type

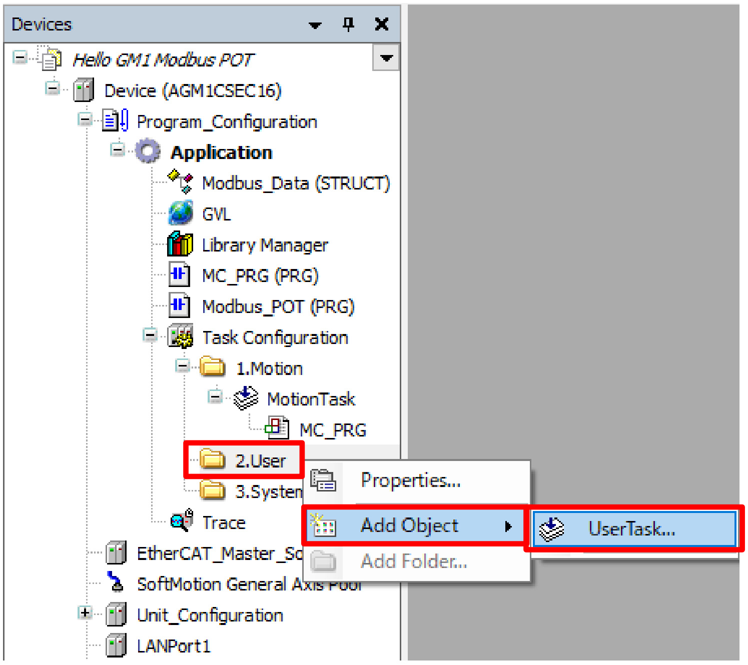
Implementation language: Ladder Logic Diagram (LD)

Modbus_POT (PRG) will be added.



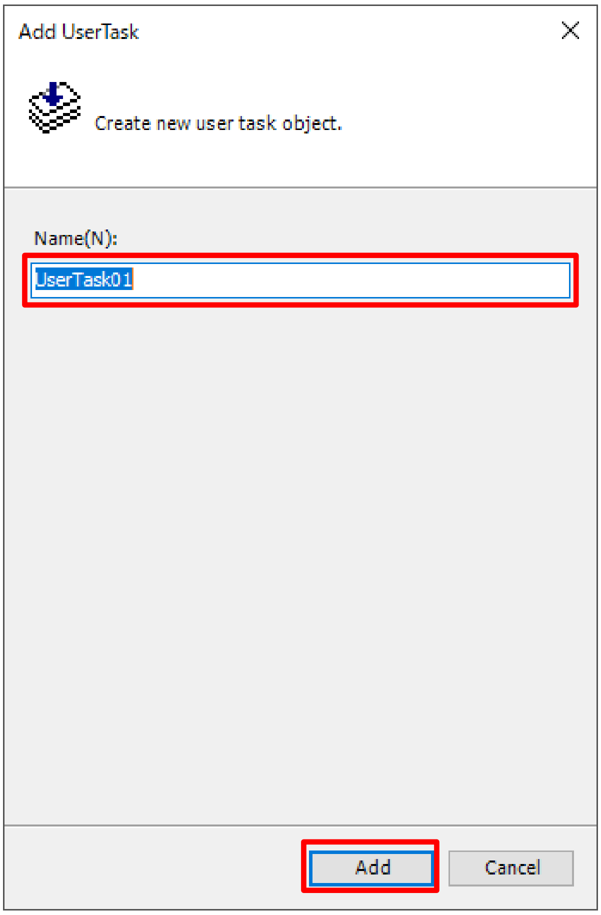
Step 3

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



Step 4

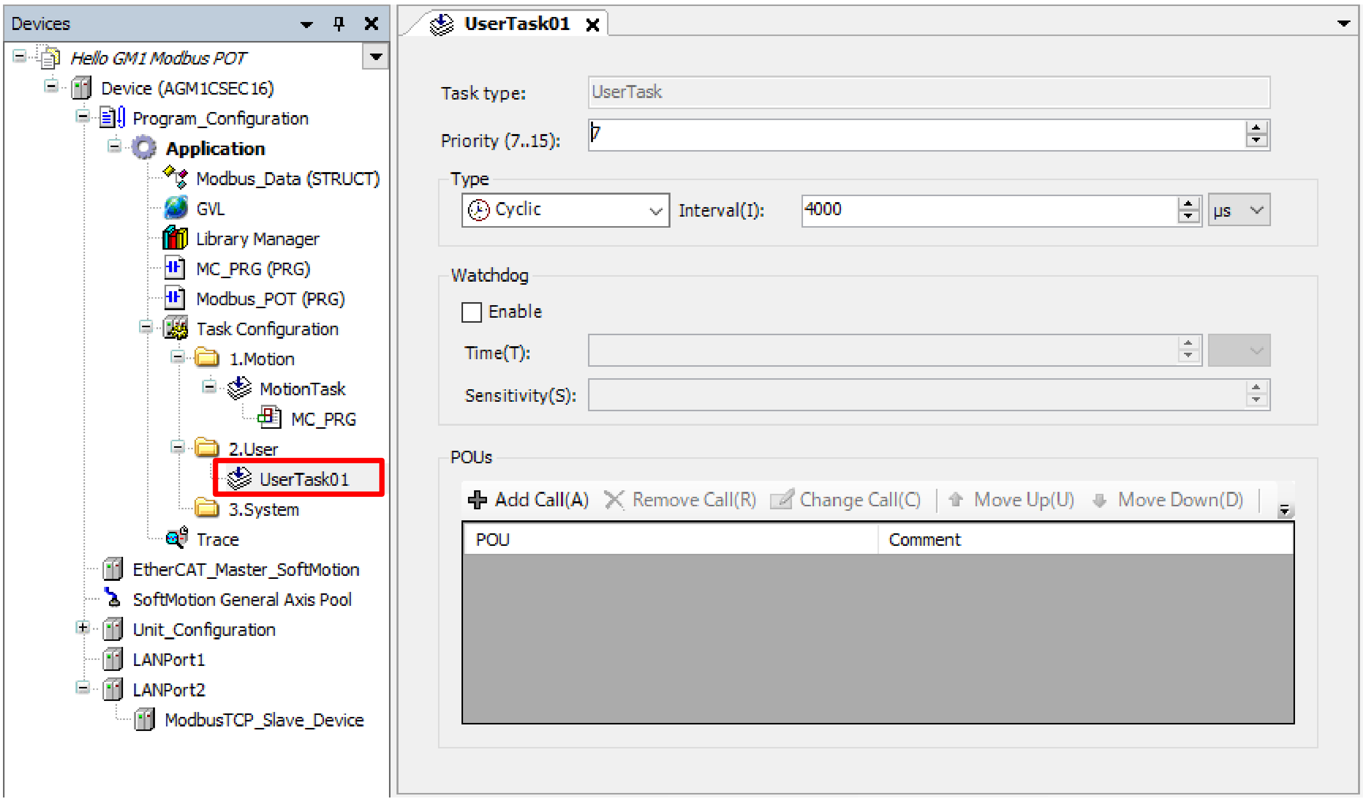
The **Add UserTask** dialog box will be displayed. Enter any desired name and click **Add**.



Name(N): UserTask01

UserTask01 will be added.
For **Priority**, **Type**, **Interval**, and **Watchdog**, leave the default values unchanged.

Priority	This field displays the priority of the task. The smaller the value, the higher the priority. For UserTask, the priority can be set between 7 and 15 inclusive.
Type	For UserTask, one of the following two types can be selected. **Cyclic**: Processes the task at intervals. Specify a task interval in the **Interval** field. **Event**: Starts task processing as soon as a rising edge of the global variable specified in the **Event** field is detected
Watchdog	If the **Enable** check box is selected, when the program execution time exceeds the preset time, the task will enter an error state and comes to a halt.


- *Hello GM1 Modbus POT*
  - Device (AGM1CSEC16)
    - Program_Configuration
      - **Application**
        - Modbus_Data (STRUCT)
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Modbus_POT (PRG)
        - Task Configuration
          - 1.Motion
            - MotionTask
              - MC_PRG
          - 2.User
            - UserTask01
          - 3.System
        - Trace
    - EtherCAT_Master_SoftMotion
    - SoftMotion General Axis Pool
    - Unit_Configuration
    - LANPort1
    - LANPort2
      - ModbusTCP_Slave_Device

UserTask01

Task type: UserTask

Priority (7..15): 7

Type

Cyclic Interval(I): 4000 µs

Watchdog

□ Enable

Time(T):

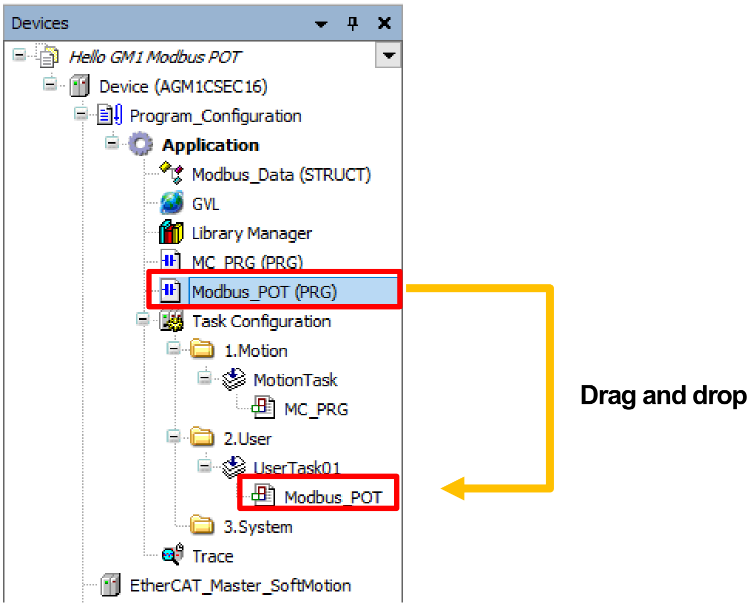
Sensitivity(S):

POUs

POU	Comment

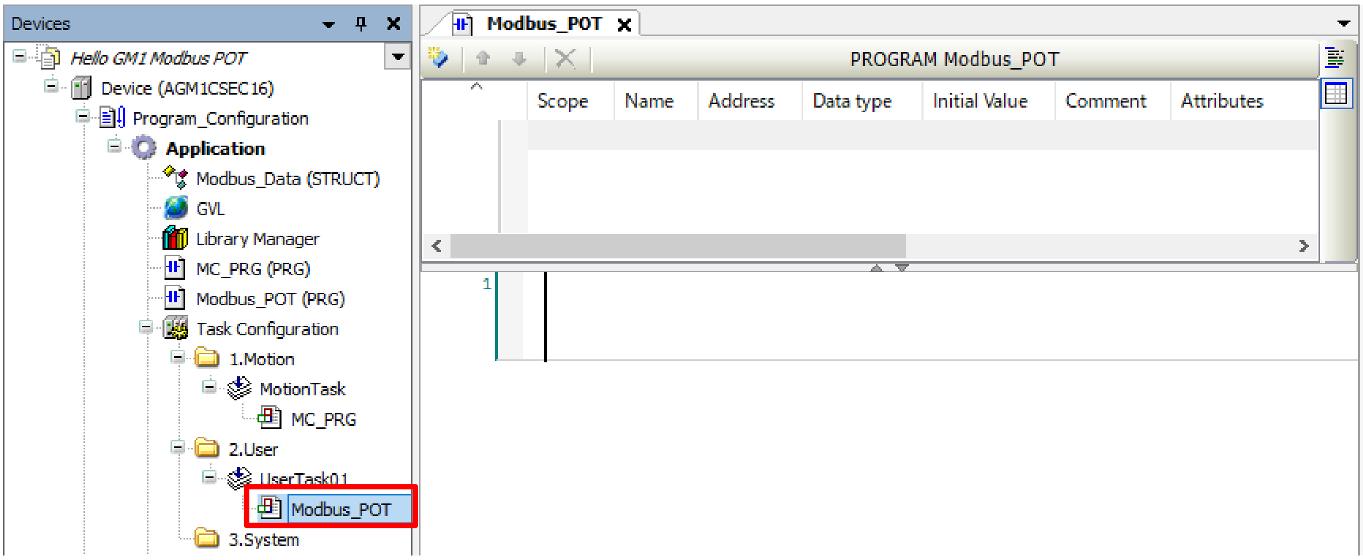
Step 5

Drag and drop the **Modbus_POT (PRG)** object added in Step 2 into **UserTask01** to add it to the task.


- *Hello GM1 Modbus POT*
  - Device (AGM1CSEC16)
    - Program_Configuration
      - **Application**
        - Modbus_Data (STRUCT)
        - GVL
        - Library Manager
        - MC_PRG (PRG)
        - Modbus_POT (PRG)
        - Task Configuration
          - 1.Motion
            - MotionTask
              - MC_PRG
          - 2.User
            - UserTask01
              - Modbus_POT
          - 3.System
        - Trace
    - EtherCAT_Master_SoftMotion

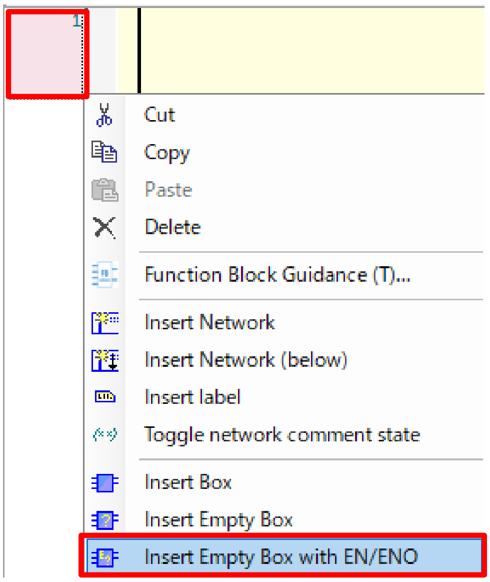
Step 6

Double-click the **Modbus_POT** object added to the task to open the program editing pane.



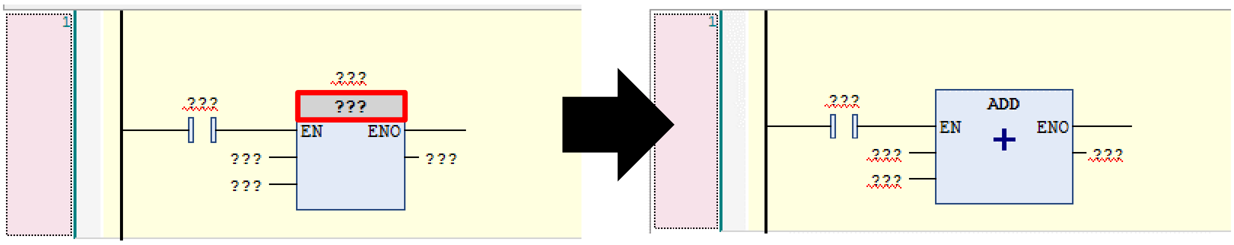
Step 7

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



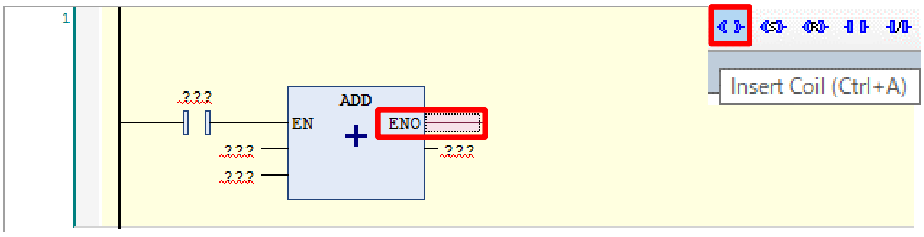
Step 8

Click **???** inside the red frame in the following figure, enter ADD, and then press the Enter key.

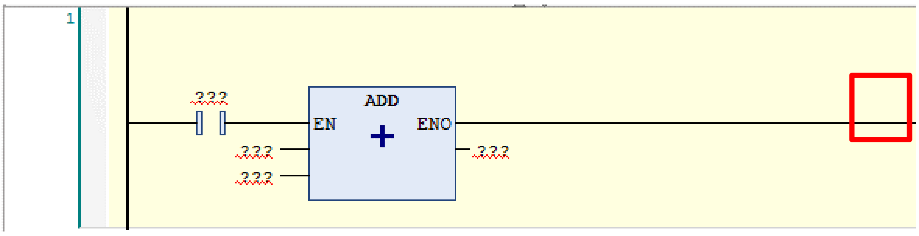


Step 9

With **ENO** selected in the **ADD** instruction, click the **Insert Coil** icon on the toolbar.



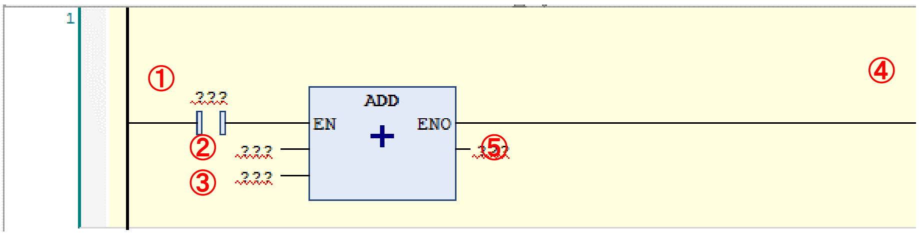
A coil will be inserted.



Step 10

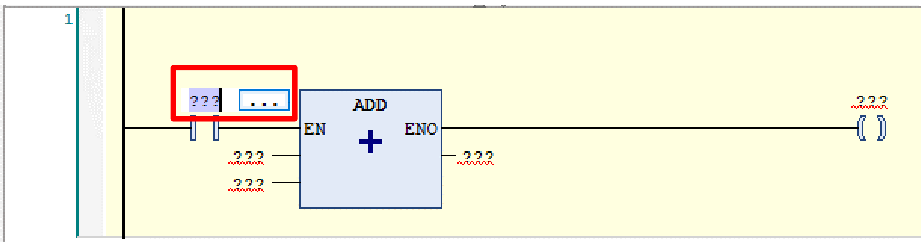
Insert a variable and fixed value into the **???** part.

Enter a variable from the **Input Assistant** dialog box.



①	Modbus_InputData.awData[0].0
②	Modbus_InputData.awData[1]
③	3000
④	Modbus_OutputData.awData[0].0
⑤	Modbus_OutputData.awData[1]

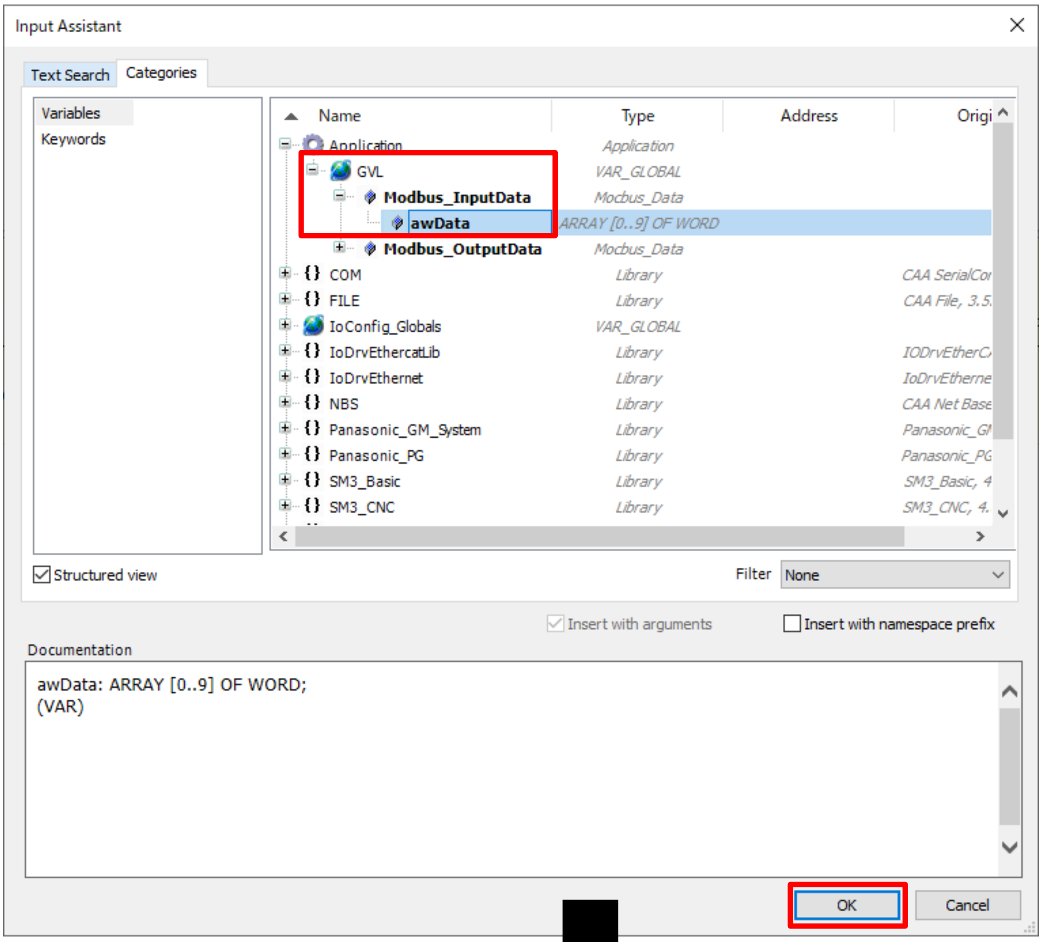
Click **???** connected to **EN** and then click ... which is displayed.



Step 11

The **Input Assistant** dialog box will be displayed.

Select **Application**, **GVL**, **Modbus_InputData**, and then **awData** and click **OK**.



Name	Type	Address	Origi
Application	Application		
GVL	VAR_GLOBAL		
Modbus_InputData	Modbus_Data		
awData	ARRAY [0..9] OF WORD		
Modbus_OutputData	Modbus_Data		
COM	Library		CAA SerialCo
FILE	Library		CAA File, 3.5.
IoConfig_Globals	VAR_GLOBAL		
IoDrvEthercatLib	Library		IODrvEtherC
IoDrvEthernet	Library		IoDrvEtherne
NBS	Library		CAA Net Base
Panasonic_GM_System	Library		Panasonic_G
Panasonic_PG	Library		Panasonic_PG
SM3_Basic	Library		SM3_Basic, 4
SM3_CNC	Library		SM3_CNC, 4.

☑ Structured view

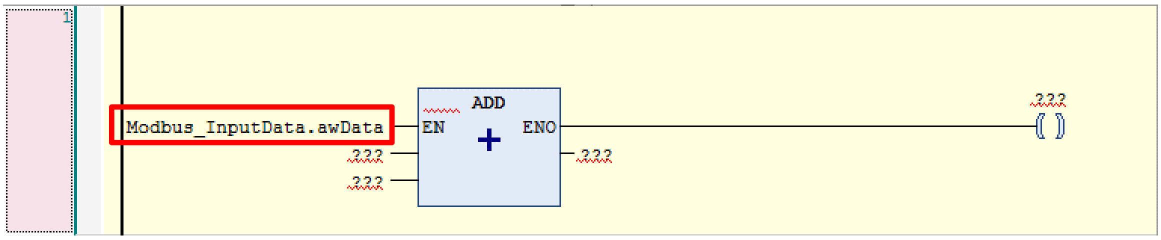
Filter: None

☑ Insert with arguments ☐ Insert with namespace prefix

Documentation

```
awData: ARRAY [0..9] OF WORD;
(VAR)
```

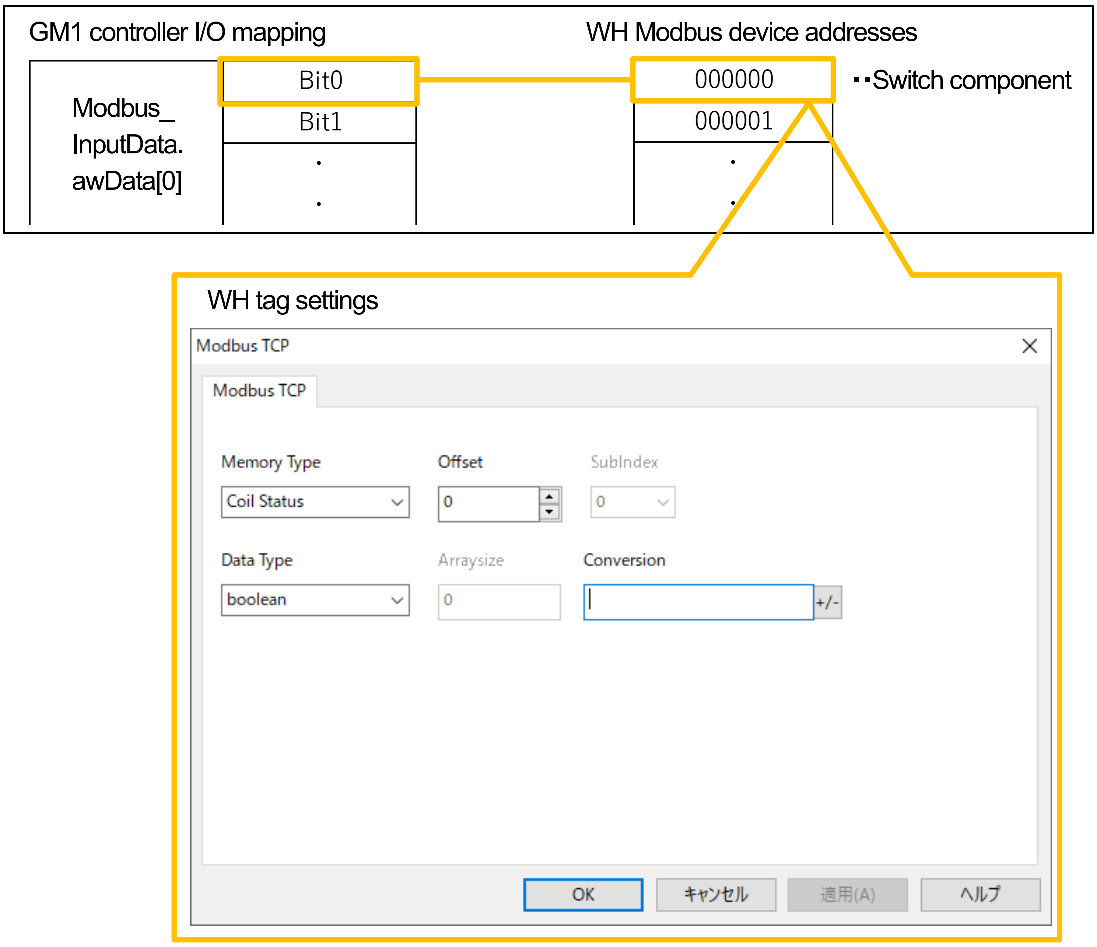
Modbus_InputData.awData will be inserted.



Step 12

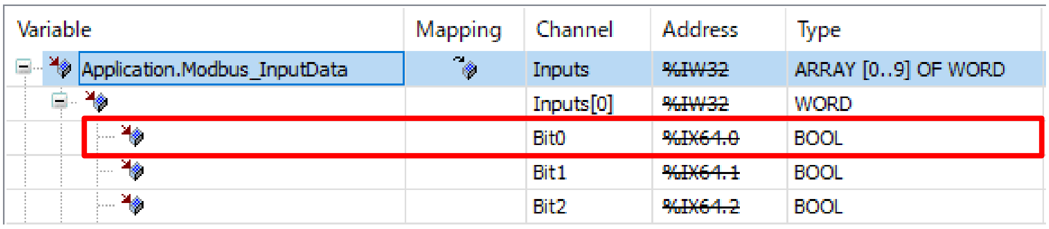
Configure settings for the switch component of the WH display unit.

	Modbus device address	Function Code	Write destination address
① Switch	000000	15 (Write Multiple Coils)	0

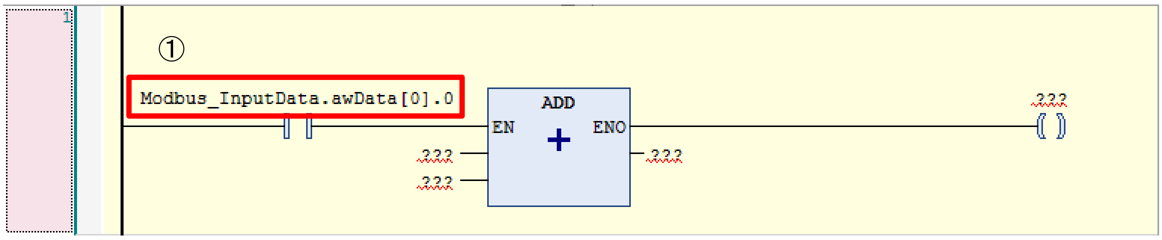


Memory Type: Coil Status
Offset: 0
SubIndex: 0
Data Type: boolean
Arraysize: 0
Conversion:

Bit0 must be specified for Inputs[0]. To do so, click **Modbus_InputData.awData** and enter "Modbus_InputData.awData[0].0".



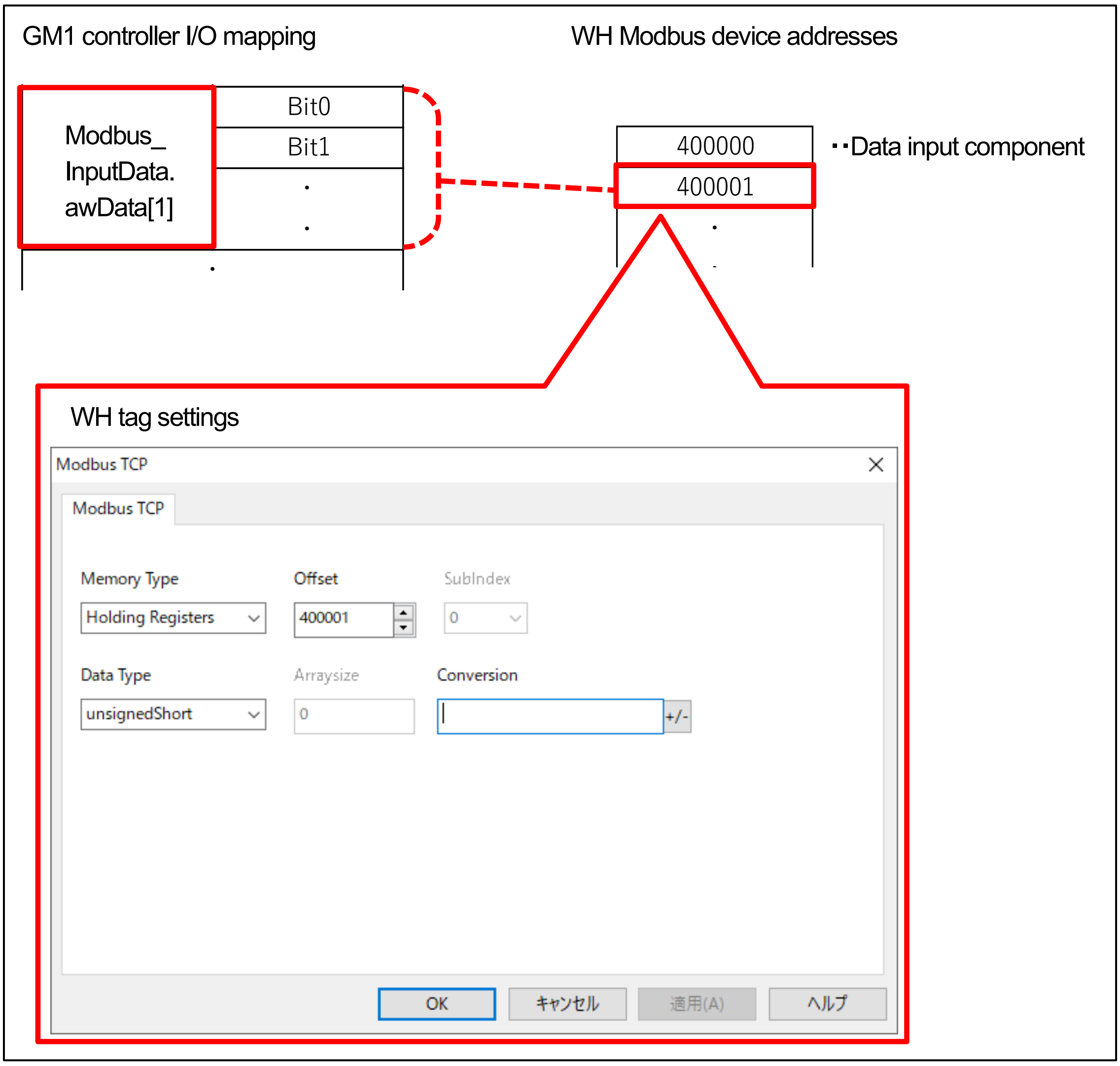
Variable	Mapping	Channel	Address	Type
Application.Modbus_InputData		Inputs	~~%IW32~~	ARRAY [0..9] OF WORD
		Inputs[0]	~~%IW32~~	WORD
		Bit0	~~%IX64.0~~	BOOL
		Bit1	~~%IX64.1~~	BOOL
		Bit2	~~%IX64.2~~	BOOL



Step 13

Configure settings for the data components of the WH display unit.

	Modbus device address	Function Code	Write destination address
② Data input	400001	16 (Write Multiple Registers)	1
③ Lamp	100000	2 (Read Discrete Inputs)	0
④ Data input	300001	4 (Read Input Registers)	1



Memory Type: Holding Registers

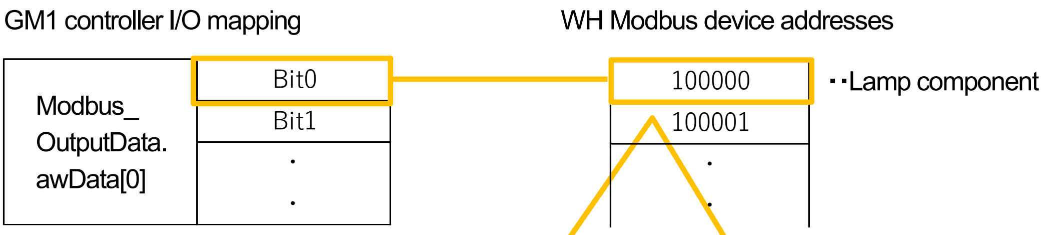
Offset: 400001

SubIndex: 0

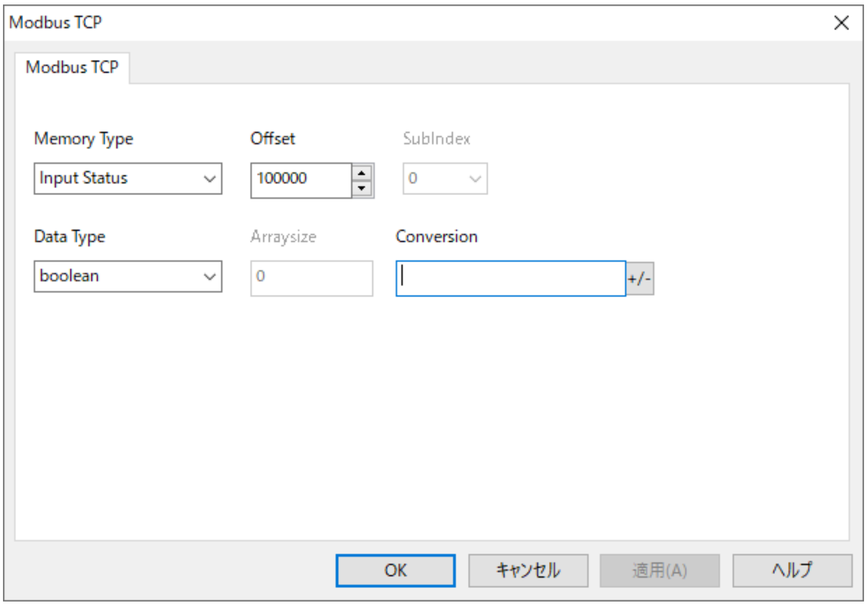
Data Type: unsignedShort

Arraysize: 0

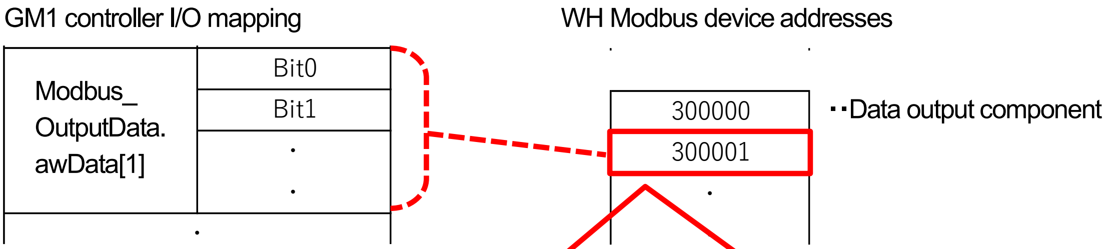
Conversion: +/-



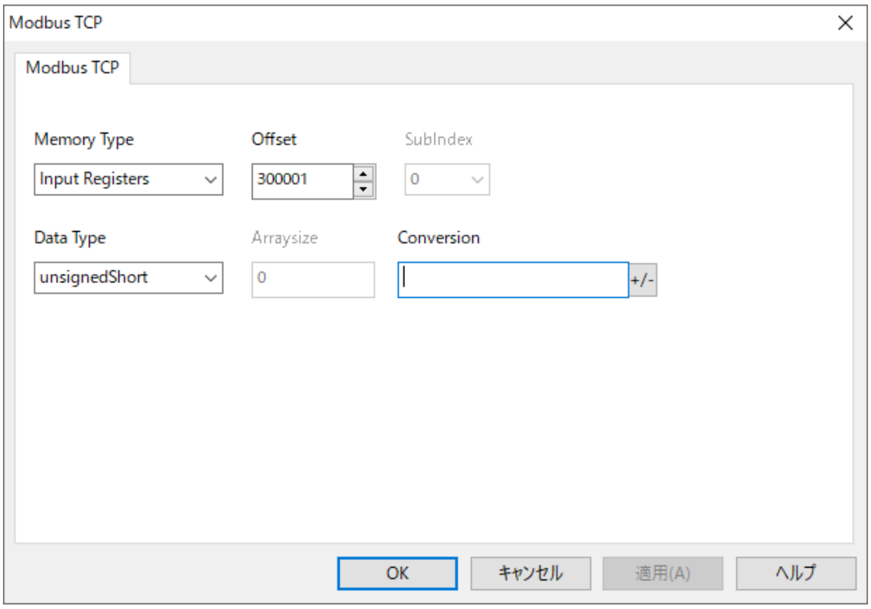
WH tag settings



Memory Type: Input Status
Offset: 100000
SubIndex: 0
Data Type: boolean
Arraysize: 0
Conversion:

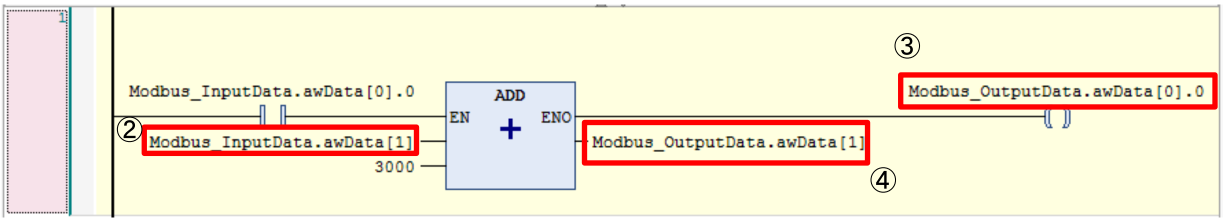

GM1 controller I/O mapping
WH Modbus device addresses
Modbus_
OutputData.
awData[1]
Bit0
Bit1
300000
300001
・・Data output component

WH tag settings



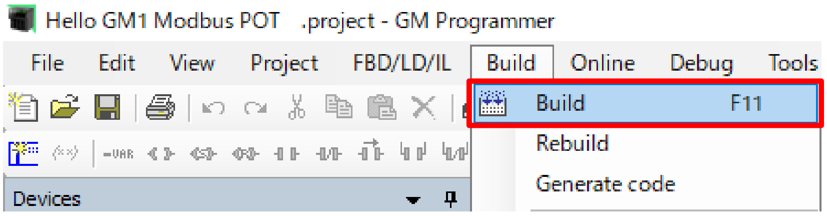
Memory Type: Input Registers
Offset: 300001
SubIndex: 0
Data Type: unsignedShort
Arraysize: 0
Conversion:

Enter data as shown below, in the same way as Step 12.

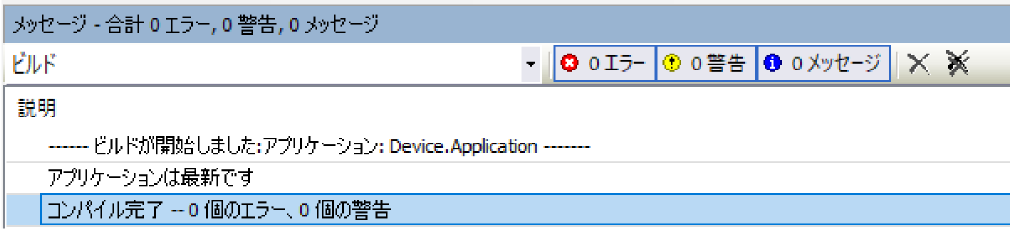

③
Modbus_InputData.awData[0].0
ADD
EN
ENO
Modbus_OutputData.awData[0].0
②
Modbus_InputData.awData[1]
3000
Modbus_OutputData.awData[1]
④

Step 14

Select **Build** from the **Build** menu to execute build.



If the message shows no error and no warning, this procedure is complete.



------ ビルドが開始しました:アプリケーション: Device.Application -------

アプリケーションは最新です

コンパイル完了 -- 0 個のエラー、0 個の警告

4 WH Display Unit Setup

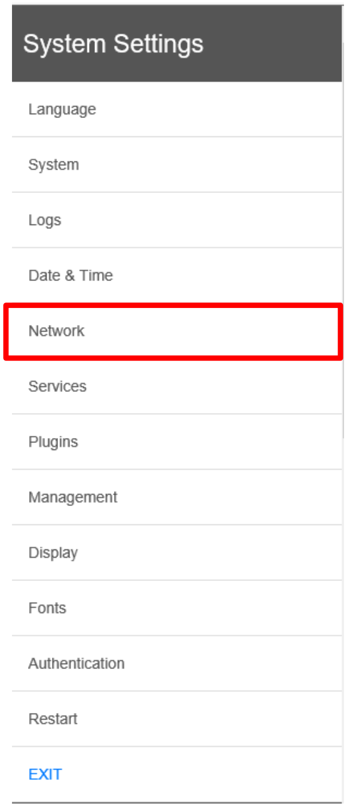
Set up the WH display unit.

4.1 Setting IP Address

Step 1

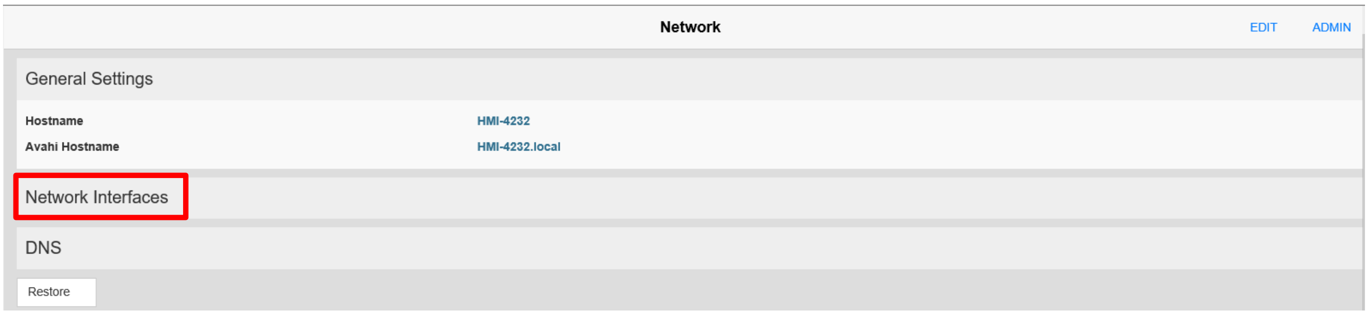
Press and hold down the screen of the WH display unit for two seconds to display the menu and then select **Show system setting**.

Each menu item will be displayed on the **System Settings** pane.
Select Network.



Step 2

Tap **Network Interfaces**.



Network EDIT ADMIN

General Settings

Hostname	HMI-4232
Avahi Hostname	HMI-4232.local

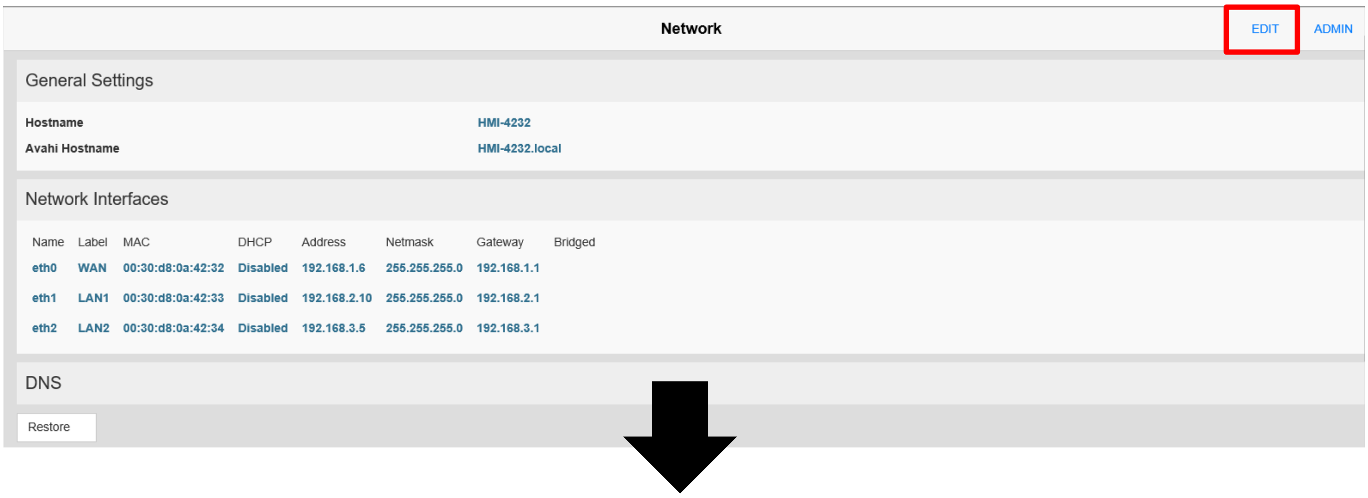
Network Interfaces

DNS

Restore

Step 3

Tapping **EDIT** on the top right of the screen invokes the edit mode. Set **eth0** as below.
Address (IP address): 192.168.2.100
Netmask (subnet mask): 255.255.255.0
Gateway (gateway): 192.168.2.1
After entering all values, tap **SAVE**.



Network EDIT ADMIN

General Settings

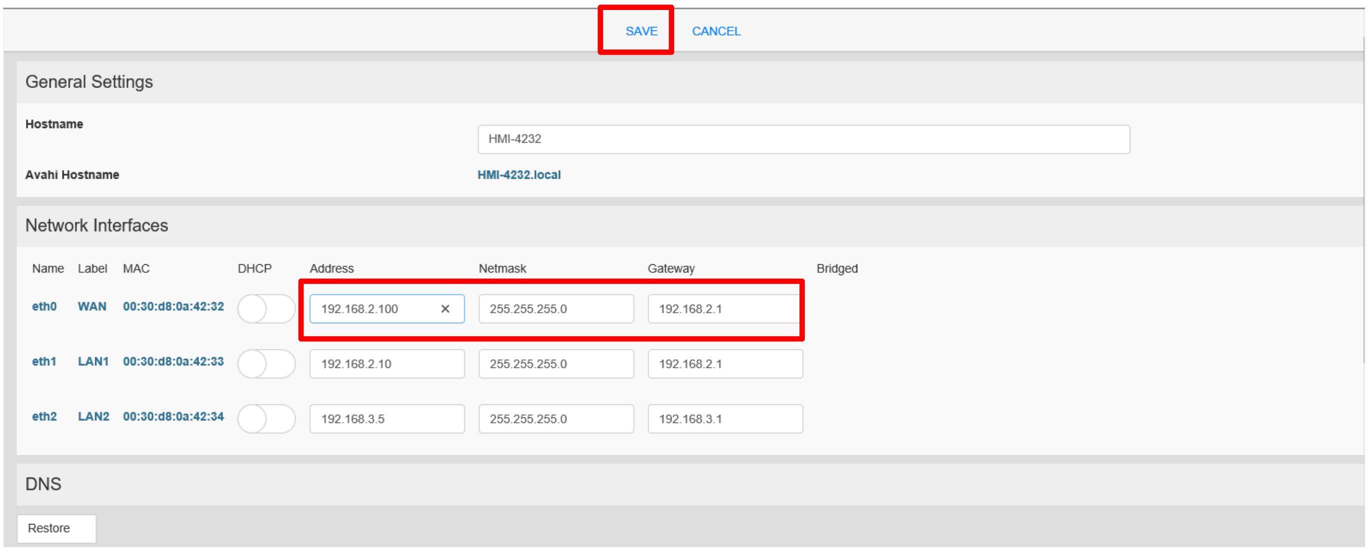
Hostname	HMI-4232
Avahi Hostname	HMI-4232.local

Network Interfaces

Name	Label	MAC	DHCP	Address	Netmask	Gateway	Bridged
eth0	WAN	00:30:d8:0a:42:32	Disabled	192.168.1.6	255.255.255.0	192.168.1.1	
eth1	LAN1	00:30:d8:0a:42:33	Disabled	192.168.2.10	255.255.255.0	192.168.2.1	
eth2	LAN2	00:30:d8:0a:42:34	Disabled	192.168.3.5	255.255.255.0	192.168.3.1	

DNS

Restore



SAVE CANCEL

General Settings

Hostname	HMI-4232
Avahi Hostname	HMI-4232.local

Network Interfaces

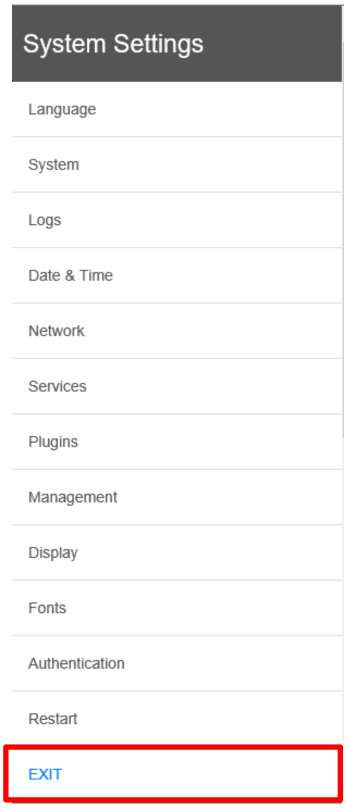
Name	Label	MAC	DHCP	Address	Netmask	Gateway	Bridged
eth0	WAN	00:30:d8:0a:42:32	○	192.168.2.100 ×	255.255.255.0	192.168.2.1	
eth1	LAN1	00:30:d8:0a:42:33	○	192.168.2.10	255.255.255.0	192.168.2.1	
eth2	LAN2	00:30:d8:0a:42:34	○	192.168.3.5	255.255.255.0	192.168.3.1	

DNS

Restore

Step 4

Tap **EXIT** to close the **System Settings** pane.



System Settings

Language

System

Logs

Date & Time

Network

Services

Plugins

Management

Display

Fonts

Authentication

Restart

EXIT

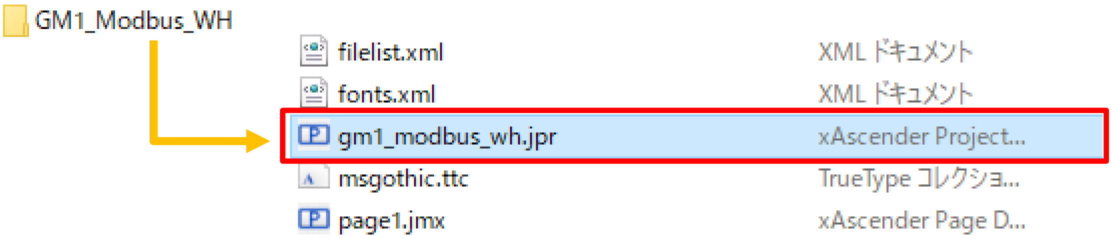
4.2 Downloading Screen Data

Step 1

Because the IP address of the WH display unit has been set to "192.168.2.100", set the IP address of the PC (which you use) to the same network as the WH display unit.

Step 2

Open **gm1_modbus_wh.jpr** in the **GM1_Modbus_WH** folder.

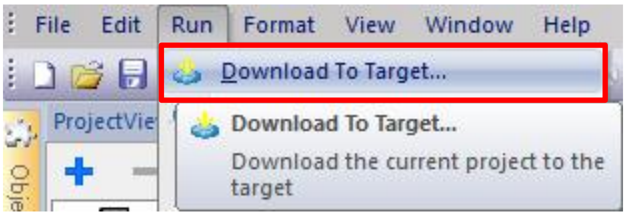


GM1_Modbus_WH

filelist.xml	XML ドキュメント
fonts.xml	XML ドキュメント
gm1_modbus_wh.jpr	xAscender Project...
msgothic.ttc	TrueType コレクショ...
page1.jmx	xAscender Page D...

Step 3

The project of the WH display unit will be displayed. Select **Download To Target** from the **Run** menu.

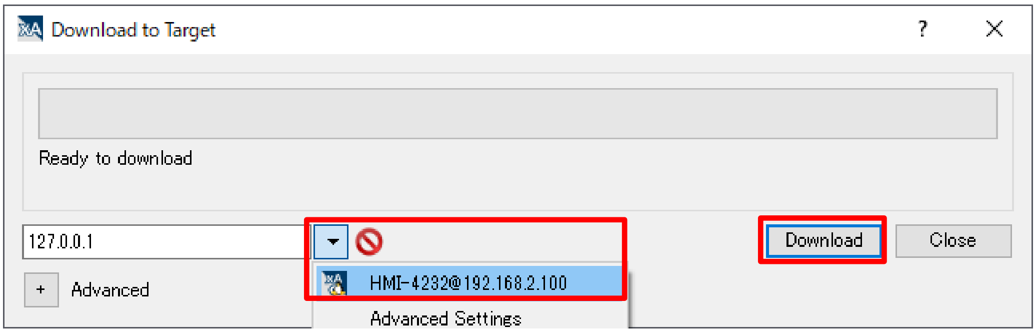


Download To Target...

Download the current project to the target

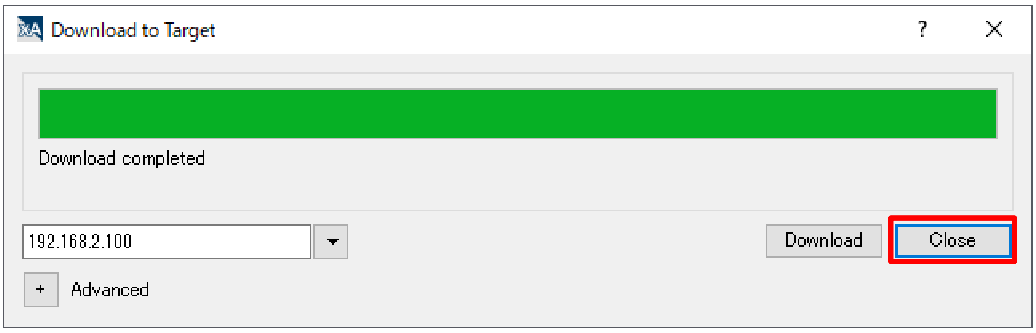
Step 4

The **Download To Target** dialog box will be displayed.
Select the IP address of the WH display unit and click **Download**.

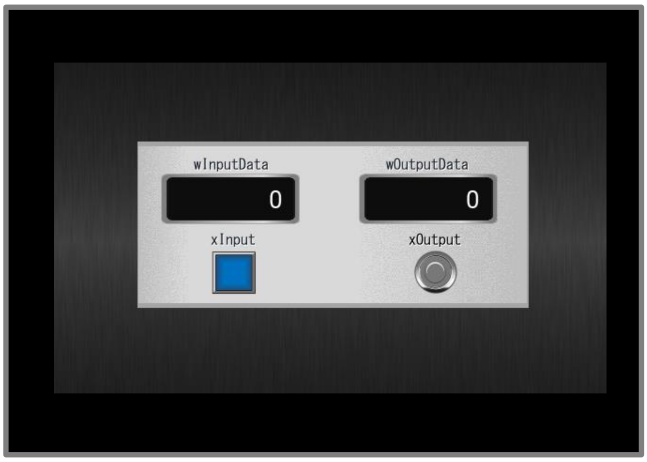


Step 5

After download is completed, click **Close**.



If download is completed successfully, the following screen will be displayed in the WH display unit.



* The layout of the screen may differ, depending on the screen size of the WH display unit that is used.

Step 6

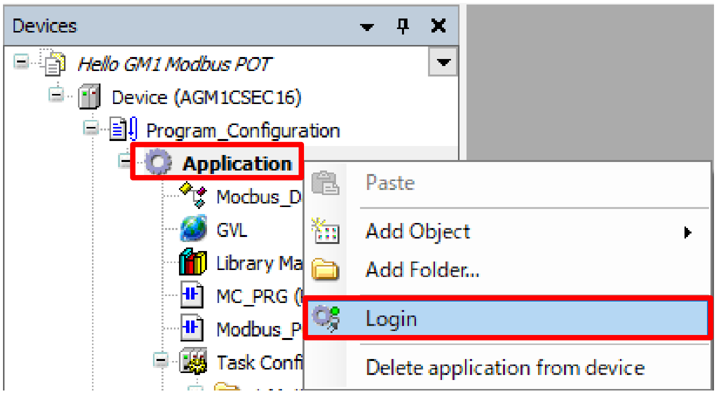
Disconnect the LAN cable connecting the PC and WH display unit and then connect it to each LAN port 2 on the WH display unit and GM1 controller.
Return the IP address of the PC to the same network as LAN port 1 (192.168.1.5) on the GM1 controller.

5 Communication Operation Check

5.1 Logging in to GM1 Controller

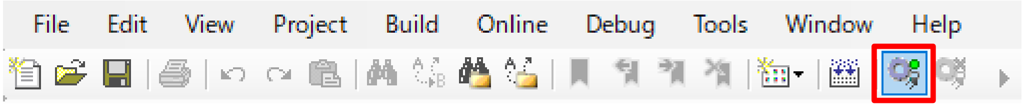
Step 1

Right-click the **Application** object and select **Login** to execute a download.



INFO

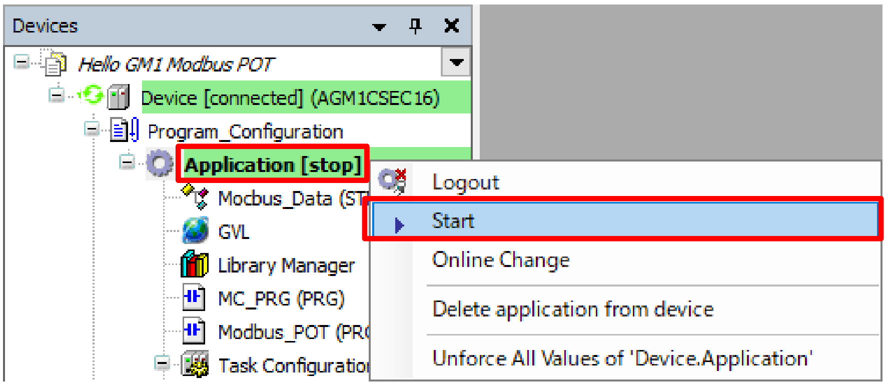
You can also log in from the toolbar.

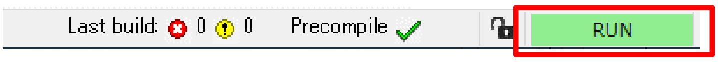


Step 2

Right-click the **Application [stop]** object and select **Start**.

When the application enters a running state, **RUN** appears in the status area on the bottom of the GM Programmer window.





INFO

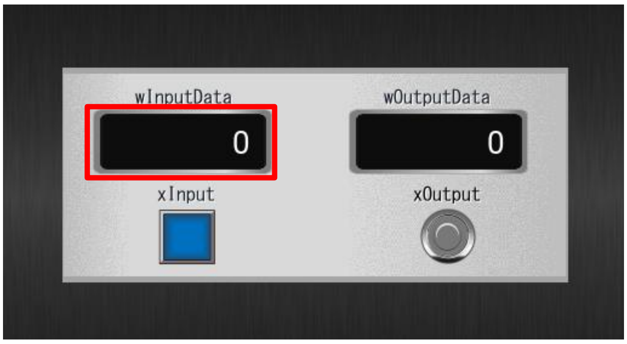
At the time of login (connection), there are two states: stopped and run.

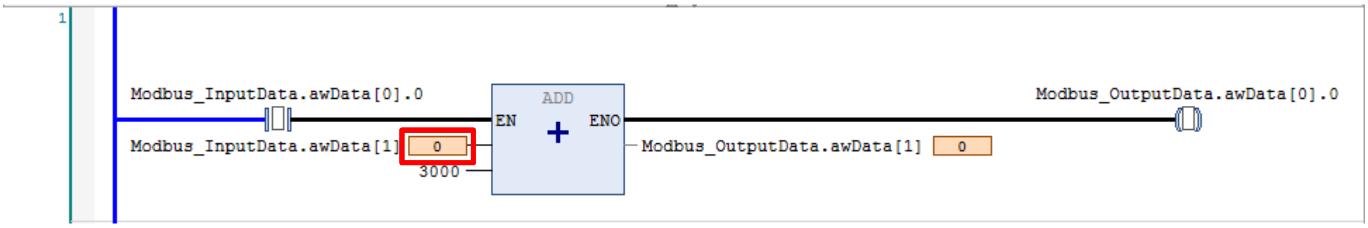
"Stopped" indicates that the program is not running, and "Run" indicates that the program is running.

5.2 Operation Check

While checking the screen of the WH display unit, monitor the state of the program coded in Network 1 in Modbus_PRG in POU.

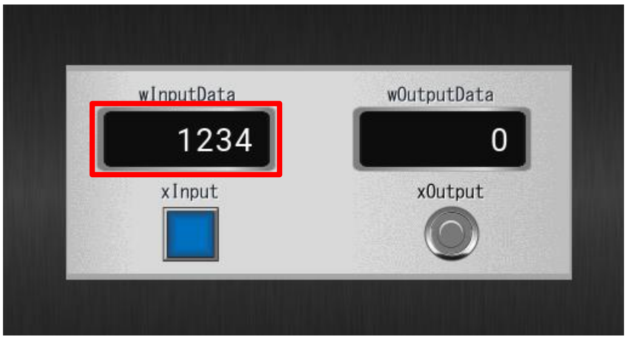
When no operation is performed from the WH display unit (initial state), the value of the "Data input" component is "0". Therefore, the value of "Modbus_InputData.awData[1]" in GM Programmer is also "0".



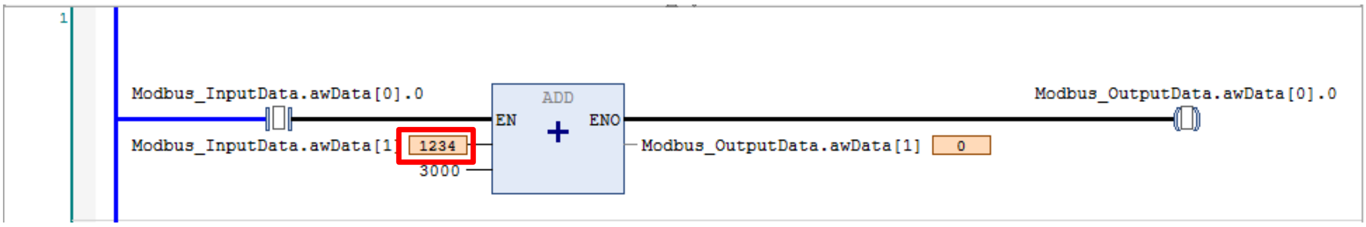

1
Modbus_InputData.awData[0].0
Modbus_InputData.awData[1]
0
3000
ADD
EN
ENO
+
Modbus_OutputData.awData[0].0
Modbus_OutputData.awData[1]
0

Step 1

Tap **wInputData** on the screen of the WH display unit to display a keypad and then enter "1234".

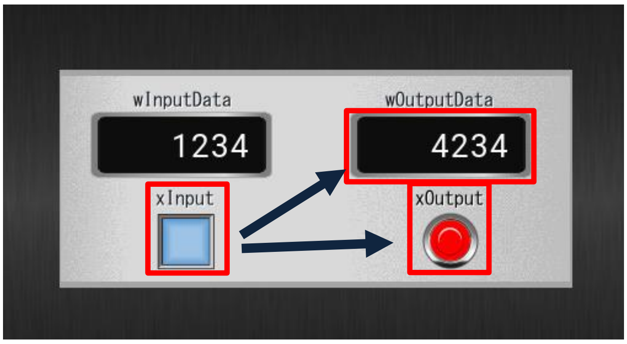


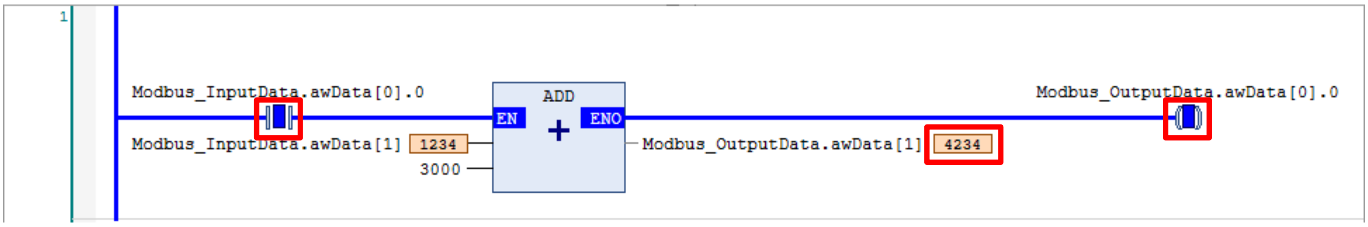
The value of **Modbus_InputData.awData[1]** in GM Programmer will be updated.


1
Modbus_InputData.awData[0].0
Modbus_InputData.awData[1]
1234
3000
ADD
EN
ENO
+
Modbus_OutputData.awData[0].0
Modbus_OutputData.awData[1]
0

Step 2

Tap **xInput** on the screen of the WH display unit. **xOutput** will light up and the value of **wOutputData** will be updated.
The ADD (addition) circuit in GM Programmer will be executed, **Modbus_OutputData.awData[0].0** will be set to TRUE, and the value of **Modbus_OutputData.awData[1]** will be updated.




1
Modbus_InputData.awData[0].0
Modbus_InputData.awData[1]
1234
3000
ADD
EN
ENO
+
Modbus_OutputData.awData[1]
4234
Modbus_OutputData.awData[0].0

This completes the Modbus slave operation check procedure for the WH display unit and GM1 controller.

Memo

Revision History

Date of issue	Manual code	Revision details
April 2022	AIM0011_01	First edition

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

April 2022