

Mitsubishi AnyWireASLINK Master Module

QJ51AW12AL(MELSEC-Q)

LJ51AW12AL(MELSEC-L)

GOT Function Sample

AnyWireASLINK Network Monitor Function Manual

Mitsubishi Electric Corporation

## Using the Samples

---

The sample screen data and files such as the instruction manual can be used upon agreement to the following matters.

- (1) This data is available for use by customers currently using or considering use of Mitsubishi products.
- (2) The intellectual property rights of the files provided by Mitsubishi (hereinafter referred to as the "Files") belong to Mitsubishi.
- (3) Alteration, reproduction, transfer or sales of the Files is prohibited.  
This does not apply when the content, in part or full, is used for Mitsubishi products incorporated in a device or system created by the customer. Furthermore, this does not apply to the transfer, reproduction, reference or change of layout in the specifications, designs or instruction manuals of built-in products prepared by the customer using Mitsubishi products.
- (4) Mitsubishi will not be held liable for any damages resulting from the use of the Files or the data extracted from the Files. The customer is responsible for all use.
- (5) If any usage conditions are appended to the Files, those conditions must be observed.
- (6) The Files may be deleted or the contents changed without prior notice.
- (7) When using the Files, please always read the corresponding manuals and related manuals indicated therein. Please pay special attention to safety, and correctly handle the product.

## CONTENTS

---

REVISIONS .....	5
1. OUTLINE.....	6
2. SYSTEM CONFIGURATION .....	6
3. GOT.....	7
3.1 System Applications That Are Automatically Selected .....	7
3.2 Controller Setting of Screen Design Software .....	7
3.3 Ethernet Setting of Screen Design Software .....	7
3.4 Overlap Window Setting of Screen Design Software .....	7
4. AnyWireASLINK MASTER MODULE .....	7
4.1 Start I/O Number of Buffer Memory .....	7
5. SCREEN SPECIFICATIONS .....	8
5.1 Display Language .....	8
5.2 Screen List/Transition .....	8
5.3 Explanation of Screens.....	14
5.3.1 AnyWireASLINK Master Module SEL (B-30000) .....	14
5.3.2 AnyWireASLINK Menu (B-30001) .....	15
5.3.3 AnyWireASLINK System Map (B-30002) .....	16
5.3.4 Input Signal Monitor (B-30003) .....	17
5.3.5 Output Signal Monitor (B-30004) .....	18
5.3.6 Sensor List (B-30005) .....	19
5.3.7 Detail (Cable [I/O Non-isolated]) (B-30006) .....	20
5.3.8 Detail (Cable [I/O Isolated]) (B-30007) .....	21
5.3.9 Detail (I/O Waterproof Connector) (B-30008) .....	22
5.3.10 Detail (Trans, I/O Waterproof) (B-30009) .....	23
5.3.11 Detail (Photoelectric [CAM, CAS]) (B-30010) .....	24
5.3.12 Detail (Proximity Amp) (B-30011) .....	25
5.3.13 Detail (Fiber Amp [CAM, CAS]) (B-30012) .....	26
5.3.14 Detail (Cylinder Switch) (B-30013) .....	27
5.3.15 Detail (Photoelectric Sensor) (B-30014) .....	28
5.3.16 Detail (Photoelectric [Rcvr.]) (B-30015) .....	29
5.3.17 Detail (Photoelectric [LT Src.]) (B-30016) .....	30
5.3.18 Detail (Proximity Switch) (B-30017) .....	31
5.3.19 Detail (ASLINKTERMINAL [Driver]) (B-30018) .....	32
5.3.20 Detail(Fiber Amp[LT Src. Only]) (B-30019) .....	33
5.3.21 Detail(+Pressure Sensor 1 Pt) (B-30020) .....	34
5.3.22 Detail(+Pressure Sensor 2 Pts) (B-30021) .....	35
5.3.23 Detail(+Pressure Sensor 16 Pts) (B-30023) .....	36
5.3.24 Detail(-Pressure Sensor 1 Pt) (B-30024) .....	37
5.3.25 Detail(-Pressure Sensor 2 Pts) (B-30025) .....	38
5.3.26 Detail(-Pressure Sensor 16 Pts) (B-30027) .....	39
5.3.27 Detail(Compound Pressure 1 Pt) (B-30028) .....	40
5.3.28 Detail(Compound Pressure 2 Pt) (B-30029) .....	41
5.3.29 Detail(Compound Pressure 16 Pt) (B-30031) .....	42
5.3.30 Detail(Photo Interrupter) (B-30032) .....	43
5.3.31 Detail(Mapping [Comb Master]) (B-30033) .....	44
5.3.32 Detail(Mapping [Comb Slave]) (B-30034) .....	45
5.3.33 Detail(Mapping Sensor[Rcvr.]) (B-30035) .....	46
5.3.34 Detail(Mapping Sensor[LT Src.]) (B-30036) .....	47
5.3.35 Detail(Incorporated[e-con]) (B-30037) .....	48
5.3.36 Detail(Incorporated[JST]) (B-30038) .....	49
5.3.37 Detail(Incorporated[MOLEX]) (B-30039) .....	50

5.3.38	Detail(Manifold Driver) (B-30040) .....	51
5.3.39	Detail(Relay Output[Driver]) (B-30041) .....	52
5.3.40	Detail(Relay Output) (B-30042) .....	53
5.3.41	Detail(Handy Remote[SwitchSide]) (B-30051) .....	54
5.3.42	Detail(Handy Remote[LED Side]) (B-30052) .....	55
5.3.43	Detail(+Pressure 2 Pts Alarm) (B-30053) .....	56
5.3.44	Detail(+Pressure 3 Pts Alarm) (B-30054) .....	57
5.3.45	Detail(-Pressure 2 Pts Alarm) (B-30057) .....	58
5.3.46	Detail(-Pressure 3 Pts Alarm) (B-30058) .....	59
5.3.47	Detail(Compound 2 Pts Alarm) (B-30061) .....	60
5.3.48	Detail(Compound 3 Pts Alarm) (B-30062) .....	61
5.3.49	iQSS Menu (B-30100) .....	62
5.3.50	iQSS Backup (B-30101) .....	63
5.3.51	iQSS Backup Setting (B-30102) .....	64
5.3.52	iQSS Backup Progress (B-30103) .....	65
5.3.53	iQSS Restoration (B-30104) .....	66
5.3.54	iQSS Restoration Setting (B-30105) .....	68
5.3.55	iQSS Restoration Progress (B-30106) .....	69
5.3.56	Alarm Reset (W-30001) .....	70
5.3.57	Language Setting (W-30002) .....	71
5.3.58	Clock Setting (W-30003) .....	72
5.3.59	Parameter Access Error (W-30004) .....	73
5.3.60	Accessing Parameters (W-30005) .....	74
5.3.61	Zero Correction ExecutionMessage (W-30006) .....	75
5.3.62	Alarm Bit Switching Message (W-30007) .....	76
5.3.63	Data Deletion Check Dialog (W-30100) .....	77
5.3.64	Notification Dialog (W-30101) .....	78
5.3.65	Execution Check Dialog (W-30102) .....	79
5.4	Slave Module Detail Information Screen Correspondence Table .....	80
5.5	Device List .....	82
5.6	Comment List .....	89
5.7	Device Data Transfer List .....	94
5.8	Recipe List .....	114
5.9	Script List .....	116
6.	OTHERS .....	211
6.1	User-Defined Name Registration .....	211
6.2	Changing System Configuration .....	211
6.3	Changing CPU .....	212
6.4	Changing Start I/O Number .....	213
6.5	iQSS Backup Folder Configuration .....	220
6.6	To Change the number of Master Module .....	221

## REVISIONS

### Sample Screen Manual

Date	Control No.*	Description
2013/9	BCN-P5999-0119	First edition
2015/1	BCN-P5999-0119-2	The format of whole manual is modified. "1. Outline" modified. "2. SYSTEM CONFIGURATION" modified. "4. AnyWireASLINK MASTER MODULE" modified. "5.2.1 Screen list/transition (common)" modified. "5.2.2 Screen list/transition (individual)" modified. "5.3 SCREEN SPECIFICATIONS" modified. "5.4 Slave Module Detail Information Screen Correspondence Table" modified. "5.5 Device List" modified. "5.6 Comment List" modified. "5.7 Device Data Transfer List" modified. "5.9 Script List" modified. "6. TEMPLATES" deleted "6.2 Changing System Configuration" modified. "6.4 Changing Start I/O Number" modified. "6.4.1 How to apply changing Start I/O Number with Device Batch Edit" added "6.4.2 To change devices set in the screen" modified. "6.4.3 To change devices set in [Common]" modified. "6.4.4 To change devices use in [Script Text]" modified. "6.4.5 To change project script" added "6.4.6 To change switch" added "6.4.7 Changing [Buffer Memory Module No. Switching]" added "6.6 To Change the number of Master Module" added

\* The Control No. is noted at the lower right of each page.

### Project Data

Date	Project data	GT Designer3*	Description
2013/9	AnyWireASLINK_V_Ver1_E.GTX	1.100E	First edition
2015/1	AnyWireASLINK_V_Ver2_E.GTX	1.123D	Master Module switching is supported. More supported Slave Modules. "TEMPLATES" is deleted.

\* The version number of screen design software used to create the project data is listed. Please use the screen design software with the listed version or later.

## 1. OUTLINE

This manual explains the sample screens of GOT2000 connected to a MELSEC-Q Series or MELSEC-L Series CPU module via Ethernet, as many as 4 AnyWireASLINK master modules can be connected to 1 CPU. The sample screens can be used to monitor, change, or perform backup/restoration (MELSEC-L Series only) of the status and parameters of the slave modules that are connected to the master module. If there are less than 4 master modules, it is necessary to modify some of the data. For how to modify the data, please refer to "6.6 To Change the number of Master Module".

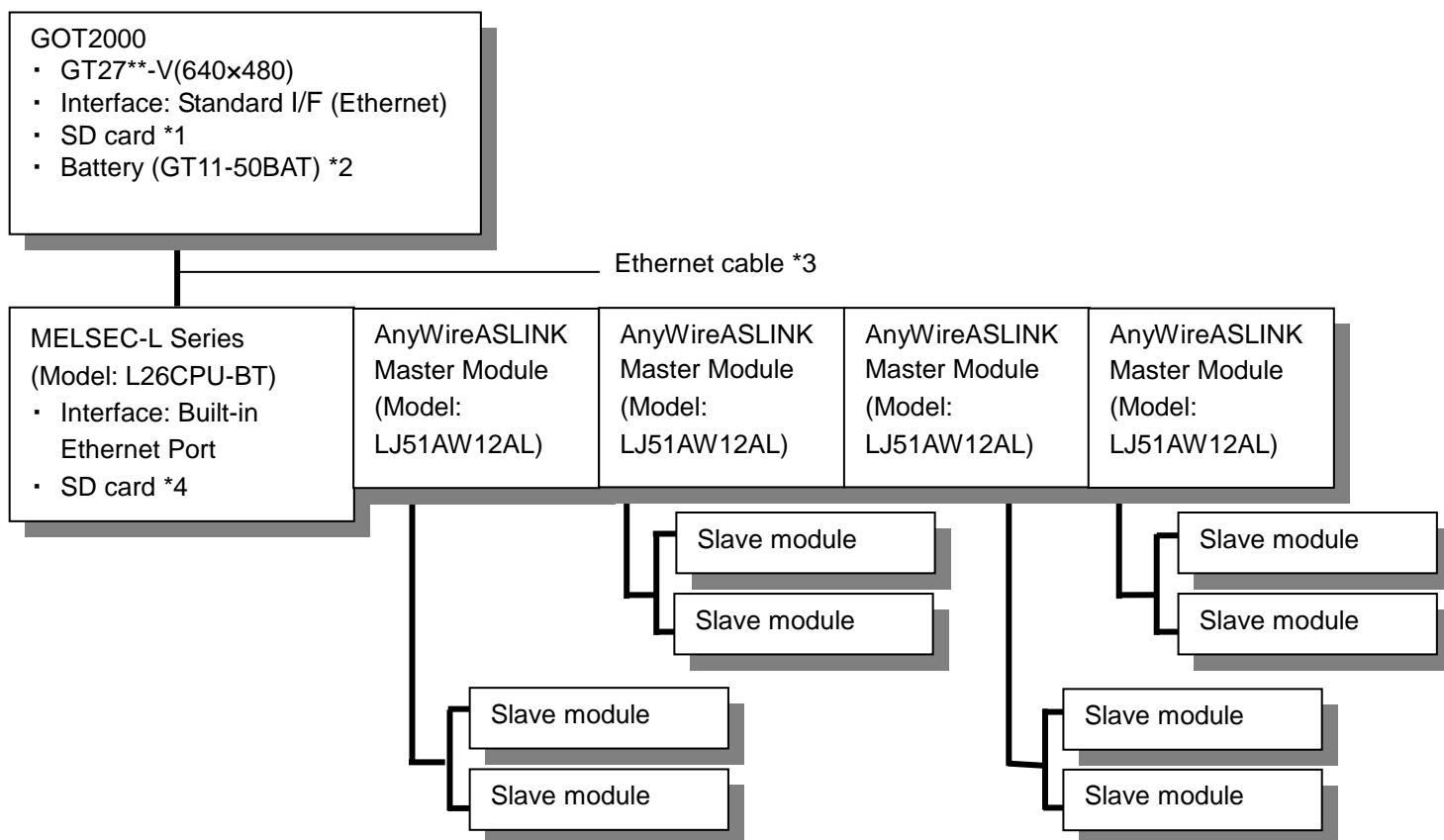
The sample screens were created to connect to a MELSEC-L Series CPU. To connect to a MELSEC-Q Series CPU, it is necessary to modify some of the data. For how to modify the data, please refer to "6.3 Changing CPU". Note that the MELSEC-Q Series CPUs do not support the backup/restoration of slave module parameters.

### <Precautions>

The backup/restoration of slave module parameters is executed by the iQSS backup/restoration (PLC↔sensor) function. In the iQSS backup/restoration (PLC↔sensor) function, the backup/restoration function of the PLC side, which is compatible with the iQSS Sensor Solution, is executed from the GOT. The specifications of the function are different from those of the backup/restoration function of the GOT side.

The backup/restoration function that is compatible with the iQ Sensor Solution is the function to backup parameters of the iQ Sensor Solution compatible sensor to a SD card in the PLC CPU or to restore the data from the SD card.

## 2. SYSTEM CONFIGURATION



\*1: The SD card is used for the recipe function.

\*2: The battery is used for the backup of the clock data. (The battery is provided with the GOT as standard.)

\*3: For more details about the cable, please refer to the "GOT2000 Series Connection Manual (Mitsubishi Products)".

\*4: The SD card is used for the iQSS backup/restoration (PLC↔sensor) function.

### 3. GOT

#### 3.1 System Applications That Are Automatically Selected

Type	System application name	
Standard Function	Standard System Application	
	Standard Font	Japanese
Communication Driver	Ethernet Connection	Ethernet(MELSEC),Q17nNC,CRnD-700, Gateway
Extended Function	Standard Font	
	Chinese (Simplified)	
	Outline Font	Alphanumeric/Kana
		Japanese (Kanji)
		Chinese (Simplified)
	Device Data Transfer	

#### 3.2 Controller Setting of Screen Design Software

##### Detail Setting

Item	Set value	Remarks
GOT NET No.	1	
GOT Station	2	
GOT Ethernet Setting	Refer to table below	
GOT Communication Port No.	5001	
Retry (Times)	3	
Startup Time (Sec)	3	
Timeout Time (Sec)	3	
Delay Time (ms)	0	

##### GOT Ethernet Setting

Item	Set value	Remarks
Reflect GOT Ethernet setting in the GOT	Checked	
GOT IP Address	192.168.3.18	
Subnet Mask	255.255.255.0	
Default Gateway	0.0.0.0	
Peripheral S/W Communication Port No.	5015	
Transparent Port No.	5014	

#### 3.3 Ethernet Setting of Screen Design Software

	Host	Net No.	Station	Unit Type	IP Address	Port No.	Communication
1	*	1	1	LCPU	192.168.3.39	5006	UDP

#### 3.4 Overlap Window Setting of Screen Design Software

[Close the window when switching base screens] of [Detail Setting] for overlap window in the [Screen Switching/Window] setting is enabled to close the window when switching base screens.

### 4. AnyWireASLINK MASTER MODULE

#### 4.1 Start I/O Number of Buffer Memory

The buffer memory's start I/O number is set to 0, 2, 4, 6. For more details about changing the buffer memory's start I/O number, please refer to "6.4 Changing Start I/O Number".

## 5. SCREEN SPECIFICATIONS

### 5.1 Display Language

The language of the text displayed on the screen can be switched between Japanese, English and Chinese (Simplified). The text strings in each language are registered in the columns No. 1 to No. 3 in the comment groups No. 495 to No. 500 as shown below. When the column No. is set in the language switching device, the language corresponding to the column No. will appear.

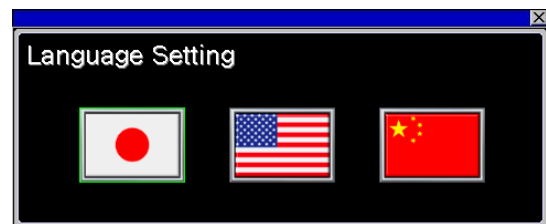
Column No.	Language
1	English
2	Japanese
3	Chinese (Simplified)

### 5.2 Screen List/Transition

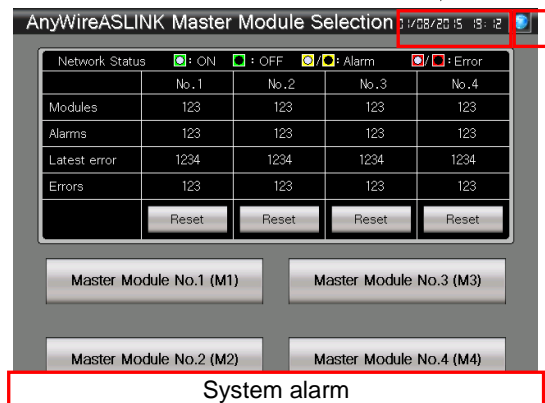
#### 5.2.1 Screen list/transition (common)



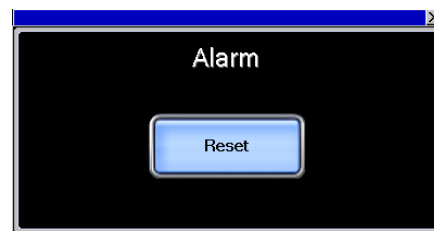
Window screen W-30003:  
Clock Setting



Window screen W-30002:  
Language Setting



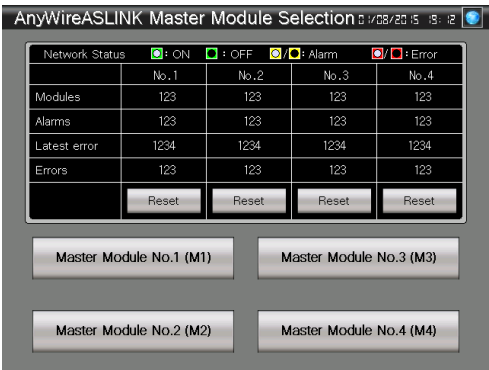
System alarm  
Base screen  
(B-30000 Menu and other base screens)



Window screen W-30001: Alarm Reset



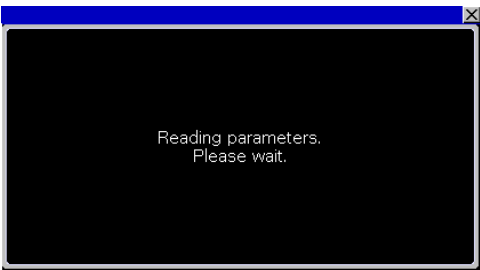
5.2.2 Screen list/transition (individual)



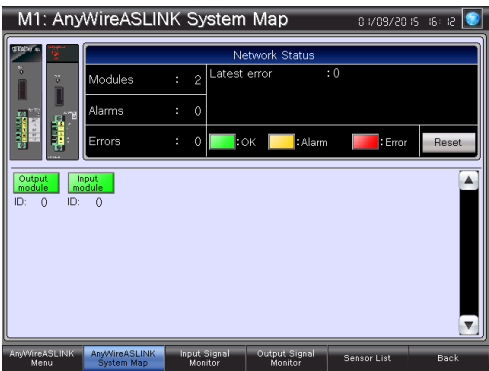
Base screen B-30000: AnyWireASLINK Master Module SEL



Base screen B-30001: AnyWireASLINK Menu

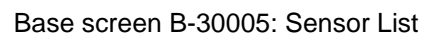
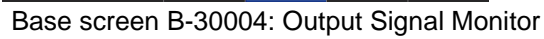
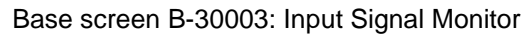


Window screen W-30005: Accessing Parameters



Base screen B-30002: AnyWireASLINK System Map

To next page



To

From  
previous  
page

Base screen B-30006 to 30062: Detail  
(The screen varies depending on the slave module type)

Window screen W-30004:  
Parameter Access Error

Window screen W-30006:  
Zero Correction ExecutionMessage  
(To be displayed in B-30020 to 30021,  
B-30023 to 30025, B-30027 to 30029,  
B-30031, B-30053 to 30054,  
B-30057 to 30058, B-30061 to 30062)

Window screen W-30007:  
Alarm Bit Switching Message  
(To be displayed in B-30020 to 30021,  
B-30023 to 30025, B-30027 to 30029,  
B-30031, B-30053 to 30054,  
B-30057 to 30058, B-30061 to 30062)

To next  
page

From previous page

Base screen B-30100: iQSS Menu

Base screen B-30102: iQSS Backup Setting

No.	Setting name	Target model	Execution unit	ID/station number	Station sub-ID	I/O
123	ABCDEFGHIJKLMN	A	Station sub-ID	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12
123	ABCDEFGHIJKLMN	A	All IDs	123	123	12

Base screen B-30101: iQSS Backup

Window screen W-30101: Notification Dialog (B-30101 to 30106 Common)

Window screen W-30102: Execution Check Dialog (B-30101, B-30103, B-30104, B-30106 Common)

Base screen B-30103: iQSS Backup Progress

To next page

From  
previous  
page

**iQSS Restoration (PLC -> Sensor)** 01/08/2015 13:12

No.	Date/time	Setting name	Target	Folder number	I/O	Total - success - fail
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12
123	12 / 12 / 12	12 : 12	ABCDEFGH I JKLMNOP	A	12	12 - 12 - 12

Buttons: Delete history, Delete all histories, Check setting, Execute

Navigation: iQSS Menu, iQSS Backup, iQSS Restoration, Back

Base screen B-30104: iQSS Restoration

This will delete the selected iQSS backup history data.  
Do you want to proceed?  
If you delete the history, please also check the target model and the folder number and delete the corresponding backup folder in the SD card that is inserted in the programmable controller.

OK Cancel

Window screen W-30100:  
Data Deletion Check Dialog (B-30101,  
B-30104 Common)

**iQSS Restoration: Progress** 01/08/2015 13:12

Restoration target	
Setting name	ABCDEFGH I JKLMNOP
Execution unit	
Target	123
I/O	Output 12

Restoration processing	
Start date/time	12 / 12 / 12 12 : 12
Progress	123%
Total number of target devices	12
Restoration succeeded	12
Restoration failed	12

Buttons: iQSS Menu, Suspend restoration

Base screen B-30106:  
iQSS Restoration Progress

**iQSS Restoration Setting** 01/08/2015 13:12

No. 123

Setting name: ABCDEFGH I JKLMNOP

Target model: [Green Box]

Execution unit: Station number [All stations] ID [ ]  
Module type: [Green Box] Output

Folder number: 12

I/O: 12

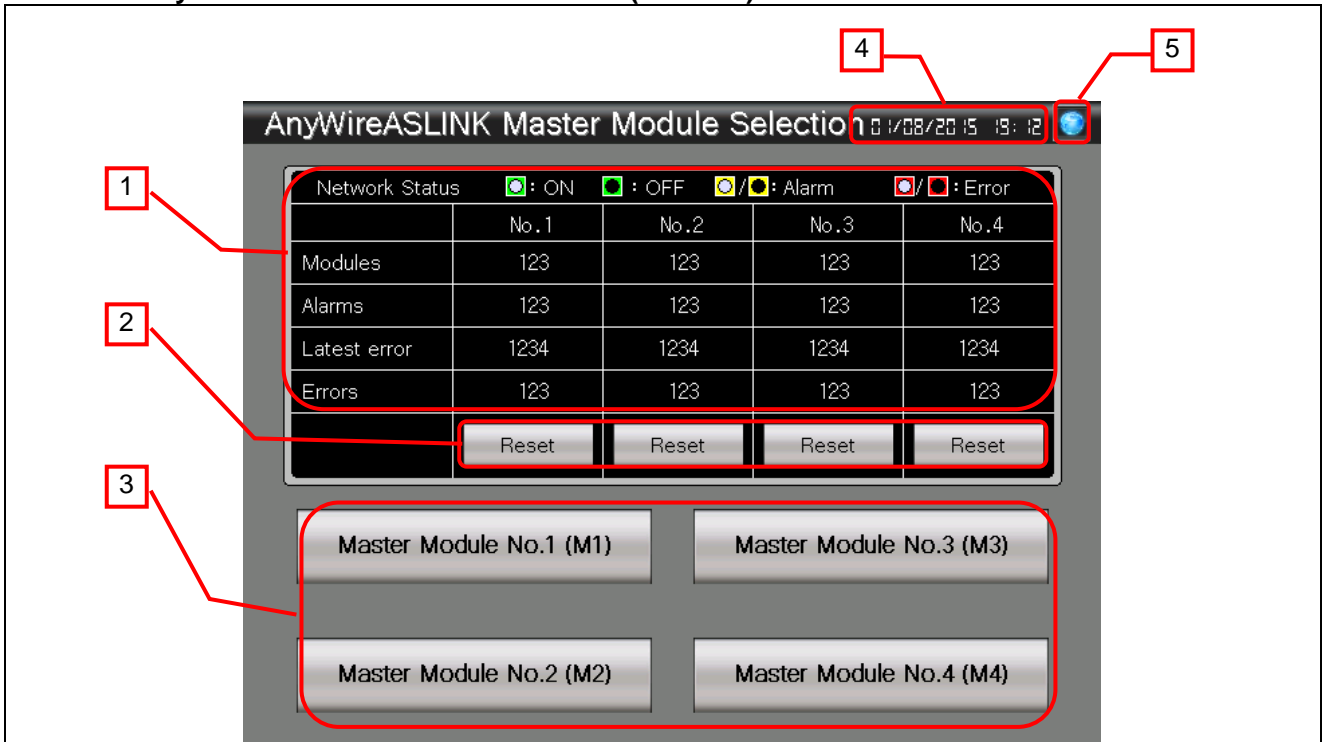
Error time action setting: [Continue] [Stop]

Back

Base screen B-30105:  
iQSS Restoration Setting

## 5.3 Explanation of Screens

### 5.3.1 AnyWireASLINK Master Module SEL (B-30000)



#### Outline

This screen can be used to select the AnyWireASLINK master module to be monitored.

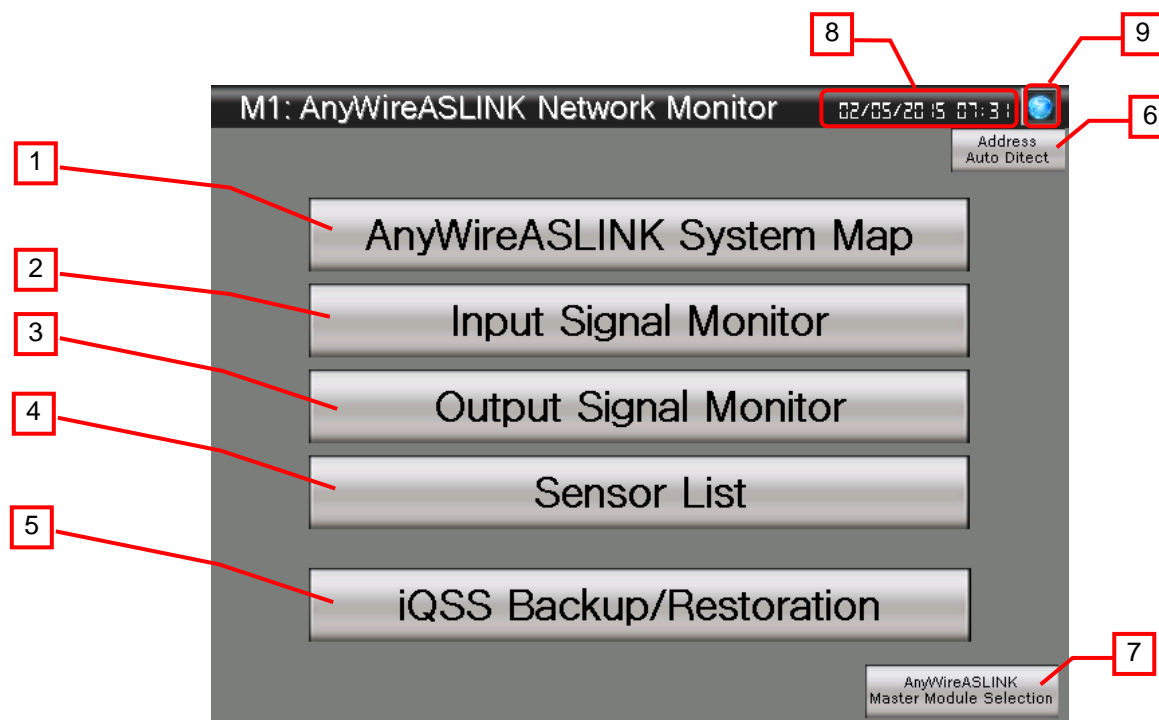
#### Description

1. Displays the number of modules, the number of alarm IDs, the latest error code and the number of error IDs of each AnyWireASLINK master module.
2. Resets alarms and errors. When reset is performed, alarms, errors, and the latest error become 0.
3. Switches to the [AnyWireASLINK Menu] screen.
4. Displays the current date and time. Touch the area to open the [Clock Setting] window.
5. Opens the [Language Setting] window.

#### Remarks

- When switching to the [AnyWireASLINK Menu] screen, set the master module to be connected.
- When GOT is started, the iQSS backup target device and the I/O No. are set with the project script. For more details about scripts, please refer to "5.9 Script List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.2 AnyWireASLINK Menu (B-30001)



#### Outline

This is the AnyWireASLINK Network Monitor menu screen.

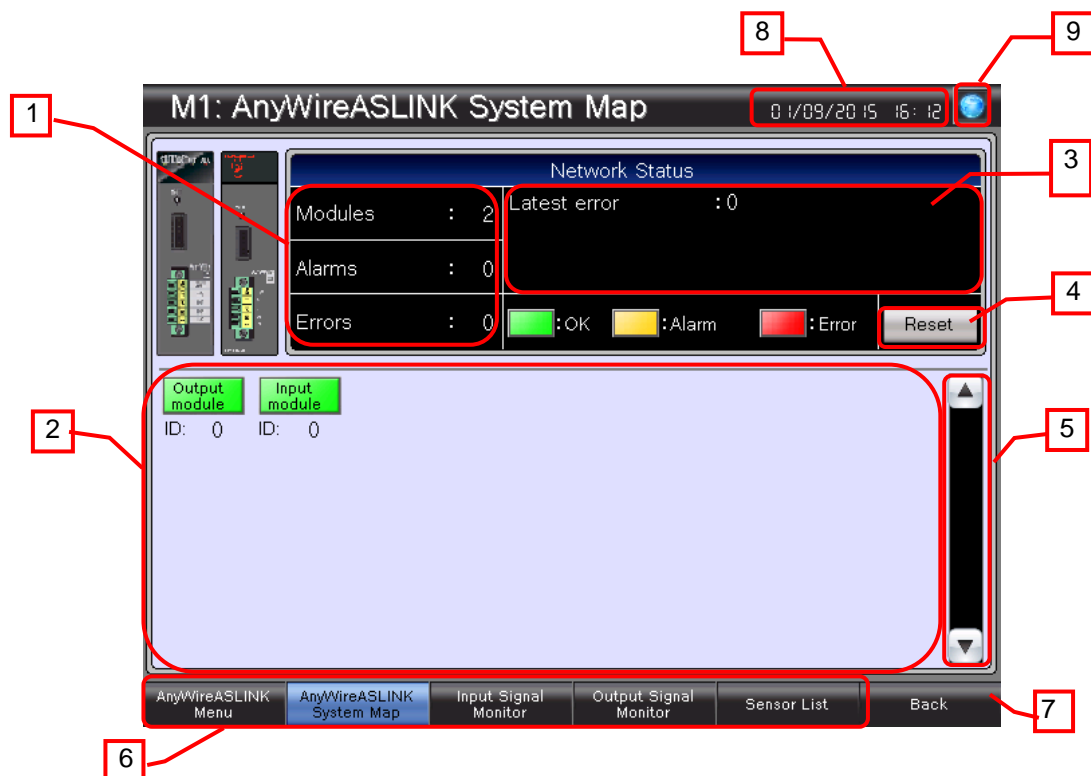
#### Description

1. Switches to the [AnyWireASLINK System Map] screen.
2. Switches to the [Input Signal Monitor] screen.
3. Switches to the [Output Signal Monitor] screen.
4. Switches to the [Sensor List] screen.
5. Switches to the [iQSS Backup/Restoration] menu screen of the iQSS backup/restoration (PLC↔Sensor) function.
6. Reads parameters of all slave modules that are recognized by the master module. Touch the switch for 1 second.
7. Switches to the [AnyWireASLINK Master Module SEL] screen.
8. Displays the current date and time. Touch the area to open the [Clock Setting] window.
9. Opens the [Language Setting] window.

#### Remarks

- The device data transfer function is used to read parameters of slave modules. For more details about the device data transfer function, please refer to "5.7 Device Data Transfer List".
- Screens cannot be switched while reading parameters.
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.3 AnyWireASLINK System Map (B-30002)



#### Outline

This is the AnyWireASLINK System Map screen. This screen displays the connection status of slave modules, and touching a slave module allows switching to the detail screen.

#### Description

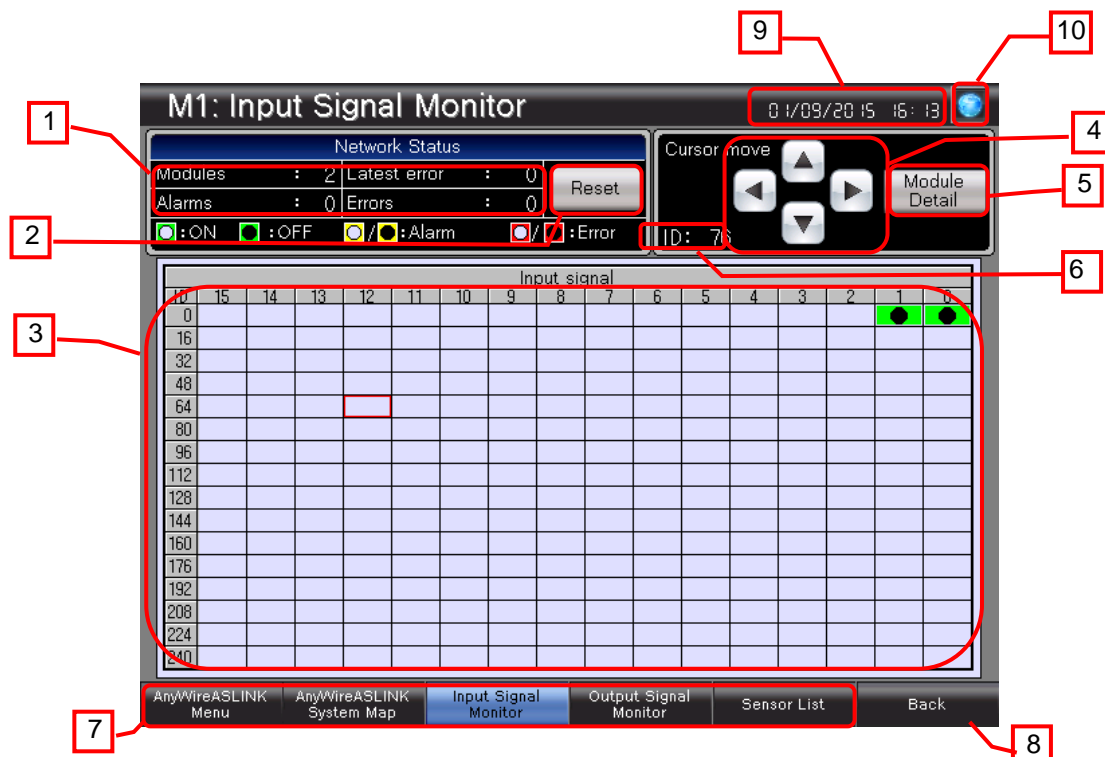
1. Displays the number of connected modules, alarms, and errors.
2. Displays the connection status of slave modules. Displays the module ID, type, and status (normal, alarm occurrence, error occurrence). By touching the switch on which the module type is displayed, the screen switches to the detail screen of the correspondent slave module.
3. Displays the occurring alarm, or the error code and the descriptions that correspond to the error code. If multiple system alarms or errors occur, the latest error code will appear.
4. Resets alarms and errors. When reset is performed, alarms, errors, and the latest error become 0.
5. Scrolls the slave module display up and down. The screen display changes per 32 modules.
6. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
7. Switches to the previously opened screen.
8. Displays the current date and time. Touch the area to open the [Clock Setting] window.
9. Opens the [Language Setting] window.

#### Remarks

- Scripts are used to scroll the slave module display. For more details about scripts, please refer to "5.9 Script List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.4 Input Signal Monitor (B-30003)



#### Outline

This is the Input Signal Monitor screen. This screen displays the status of signals (ON/OFF, alarms occurring, errors occurring) of the connected input slave module. The number of signals differs depending on the type of the slave module.

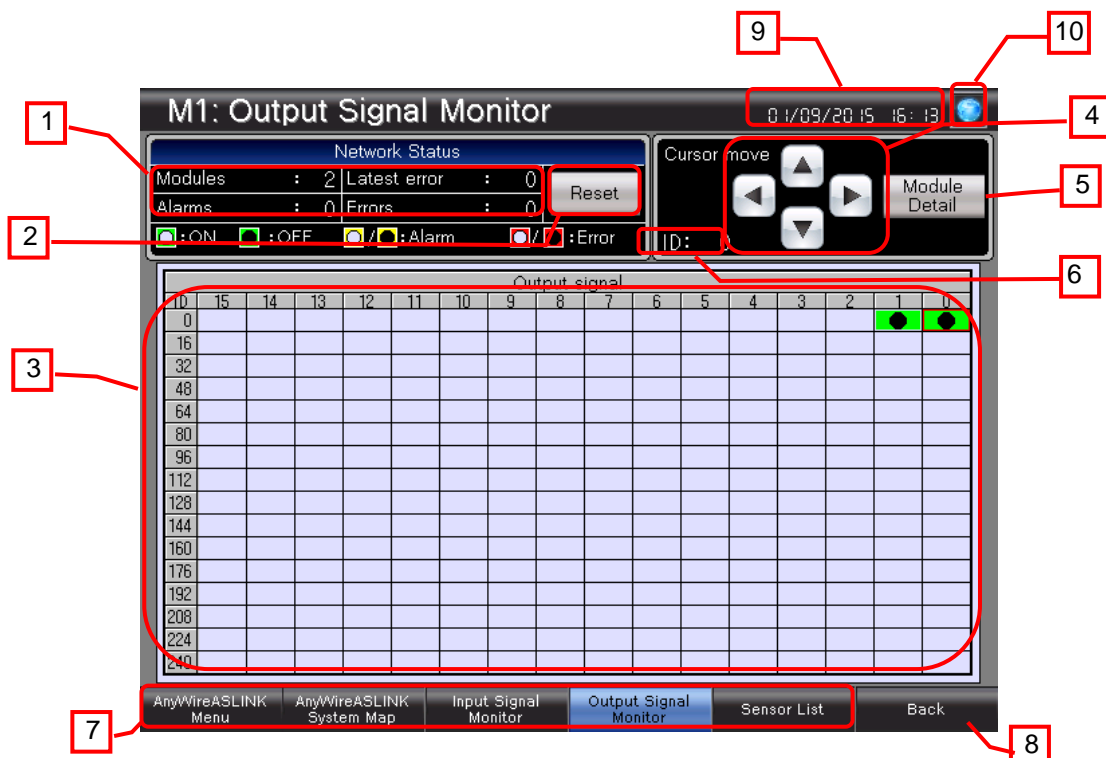
#### Description

1. Displays the number of connected modules, alarms, and errors, and the latest error information.
2. Resets alarms and errors. When reset is performed, alarms, errors, and the latest error become 0.
3. Displays the status of the input slave module. Touching the cell moves the cursor to the touched position.
4. Moves the cursor up and down and left and right.
5. Switches to the detail screen of the slave module where the cursor is being displayed. The screen will not switch to the detail screen if the cursor exists in the position where the ID is not recognized.
6. Displays the ID of the position where the cursor is being displayed.
7. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
8. Switches to the previously opened screen.
9. Displays the current date and time. Touch the area to open the [Clock Setting] window.
10. Opens the [Language Setting] window.

#### Remarks

- Scripts are used to move the cursor. For more details about scripts, please refer to "5.9 Script List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.5 Output Signal Monitor (B-30004)



#### Outline

This is the Output Signal Monitor screen. This screen displays the status of signals (ON/OFF, alarms occurring, errors occurring) of the connected output slave module. The number of signals differs depending on the type of the slave module.

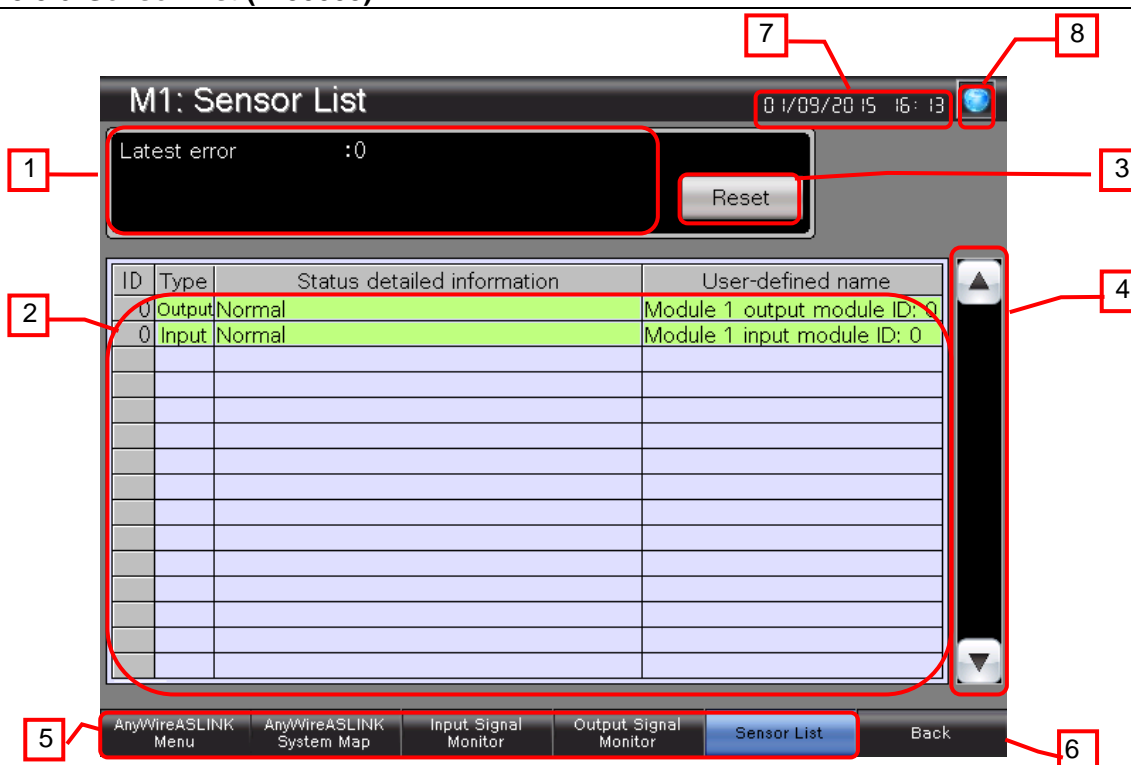
#### Description

1. Displays the number of connected modules, alarms, and errors, and the latest error information.
2. Resets alarms and errors. When reset is performed, alarms, errors, and the latest error become 0.
3. Displays the status of the output slave module. Touching the cell moves the cursor to the touched position.
4. Moves the cursor up and down and left and right.
5. Switches to the detail screen of the slave module where the cursor is being displayed. The screen will not switch to the detail screen if the cursor exists in the position where the ID is not recognized.
6. Displays the ID of the position where the cursor is being displayed.
7. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
8. Switches to the previously opened screen.
9. Displays the current date and time. Touch the area to open the [Clock Setting] window.
10. Opens the [Language Setting] window.

#### Remarks

- Scripts are used to move the cursor. For more details about scripts, please refer to "5.9 Script List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.6 Sensor List (B-30005)



## Outline

This screen displays the information of slave modules. By touching the line on which the slave module information is displayed, the screen switches to the detail screen of the target slave module.

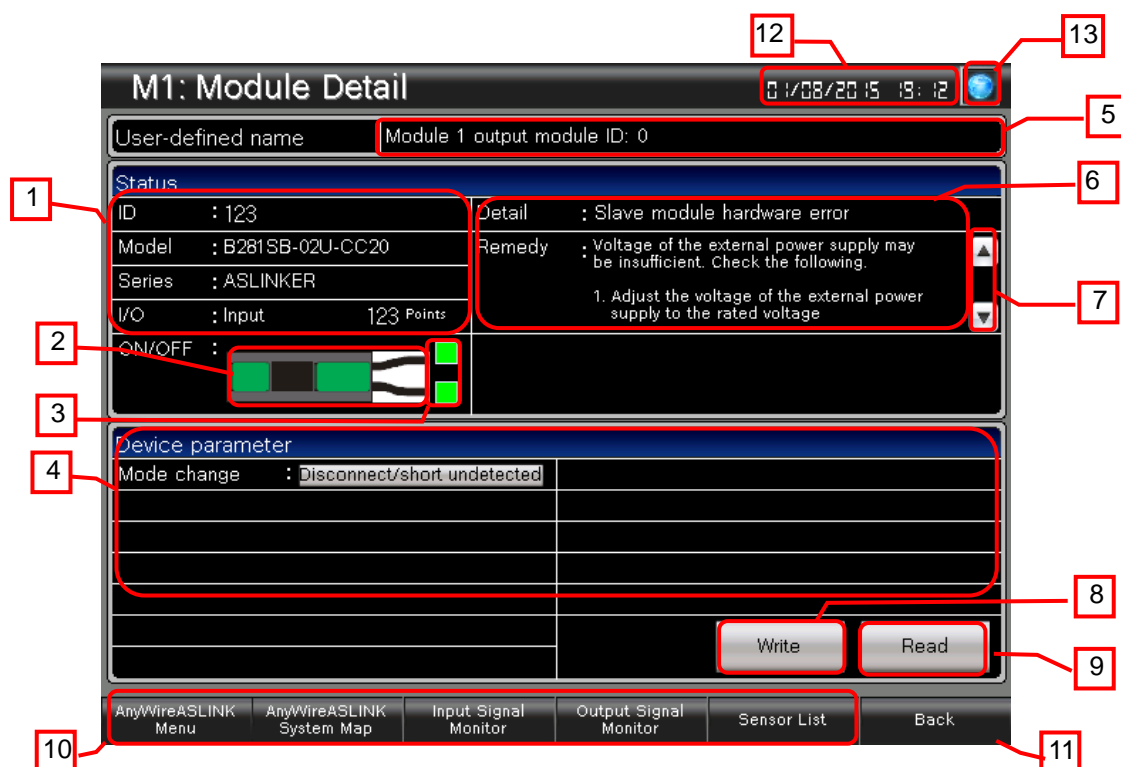
### Description

1. Displays the occurring alarm, or the error code and the descriptions that correspond to the error code. If multiple system alarms or errors occur, the latest error code will appear.
2. Displays the information of slave modules. The ID, type, status detailed information, and user-defined name of the recognized slave module are displayed. By touching the line on which the information is displayed, the screen switches to the detail screen of the slave module.
3. Resets alarms and errors. When reset is performed, the latest error becomes 0.
4. Scrolls the slave module information up and down. The screen display changes per 15 modules.
5. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
6. Switches to the previously opened screen.
7. Displays the current date and time. Touch the area to open the [Clock Setting] window.
8. Opens the [Language Setting] window.

## Remarks

- Scripts are used to scroll the slave module information. For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.7 Detail (Cable I/O Non-isolated)) (B-30006)



#### Outline

This screen displays and sets the detail information of the cable connection type (I/O non-isolated) slave module.

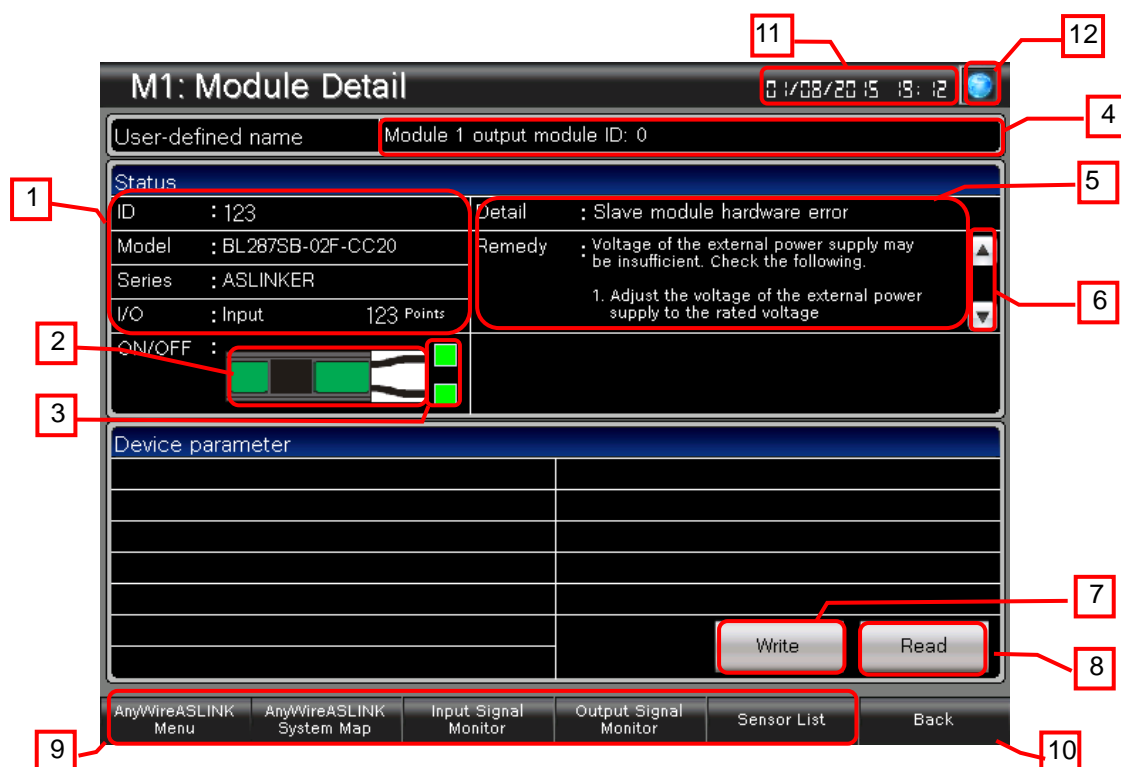
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
9. Reads parameters of the slave module.
10. Switches to each screen.
11. Switches to the previously opened screen.
12. Displays the current date and time. Touch the area to open the [Clock Setting] window.
13. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the slave module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.8 Detail (Cable [I/O Isolated]) (B-30007)



#### Outline

This screen displays the detail information of the cable connection type (I/O isolated) slave module.

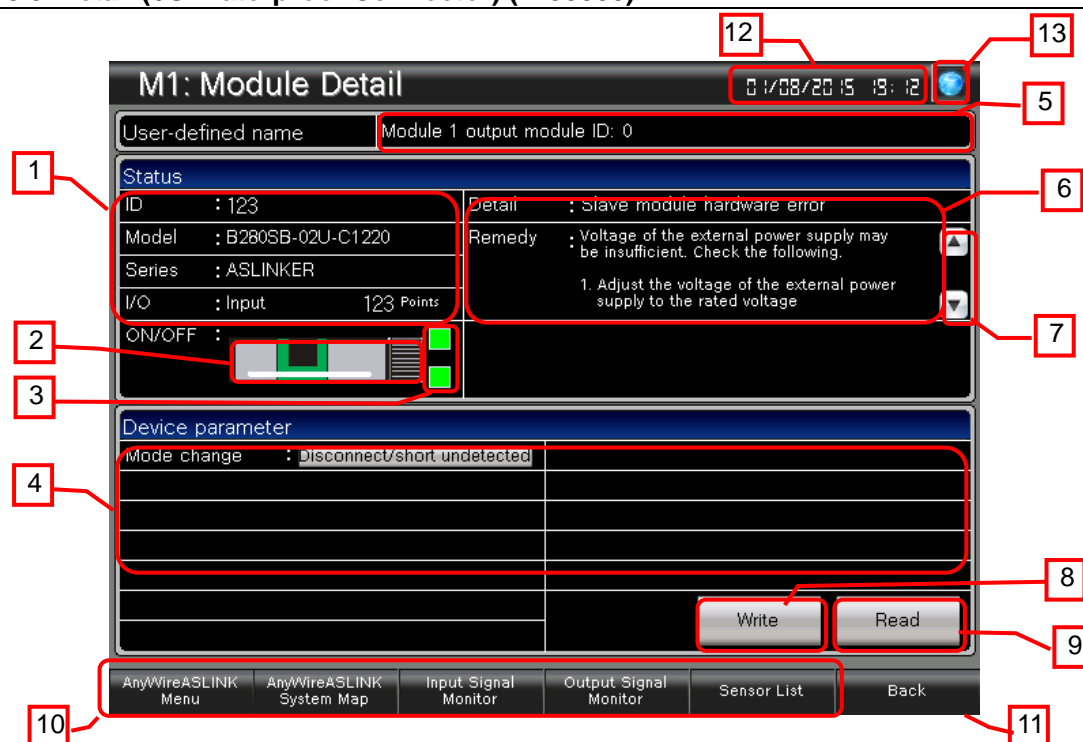
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.9 Detail (I/O Waterproof Connector) (B-30008)



#### Outline

This screen displays and sets the detail information of the I/O waterproof connector connection type slave module.

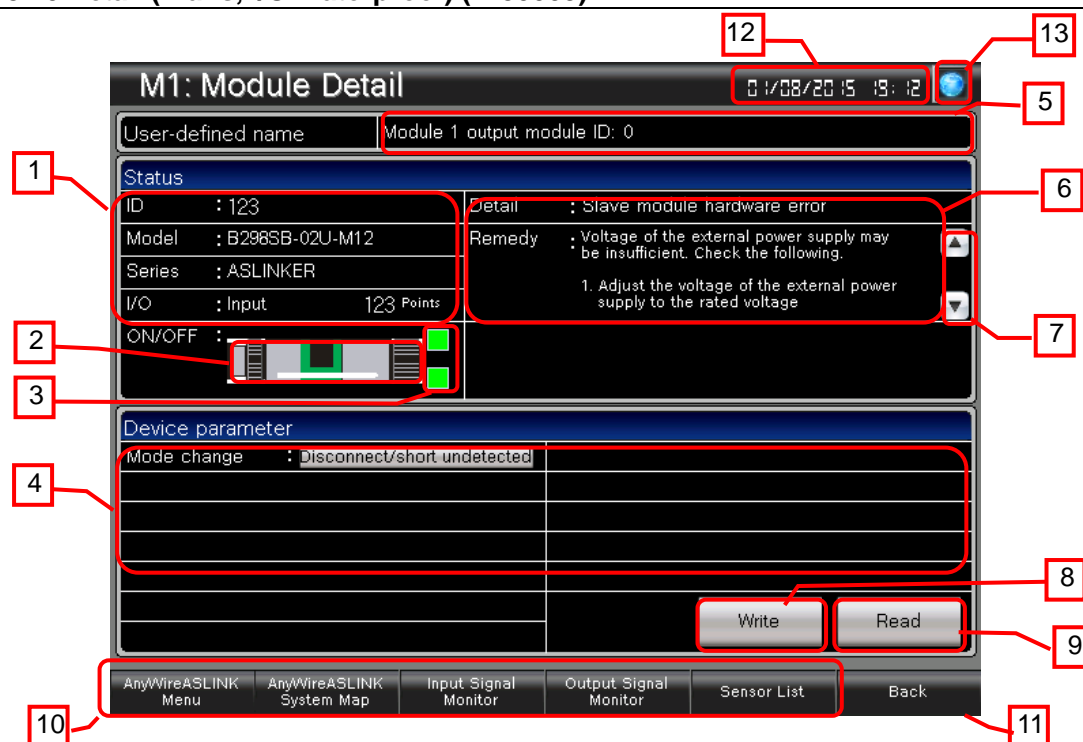
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
9. Reads parameters of the slave module.
10. Switches to each screen.
11. Switches to the previously opened screen.
12. Displays the current date and time. Touch the area to open the [Clock Setting] window.
13. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the slave module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.10 Detail (Trans, I/O Waterproof) (B-30009)



#### Outline

This screen displays the detail information of the transfer, I/O waterproof connector connection type slave module.

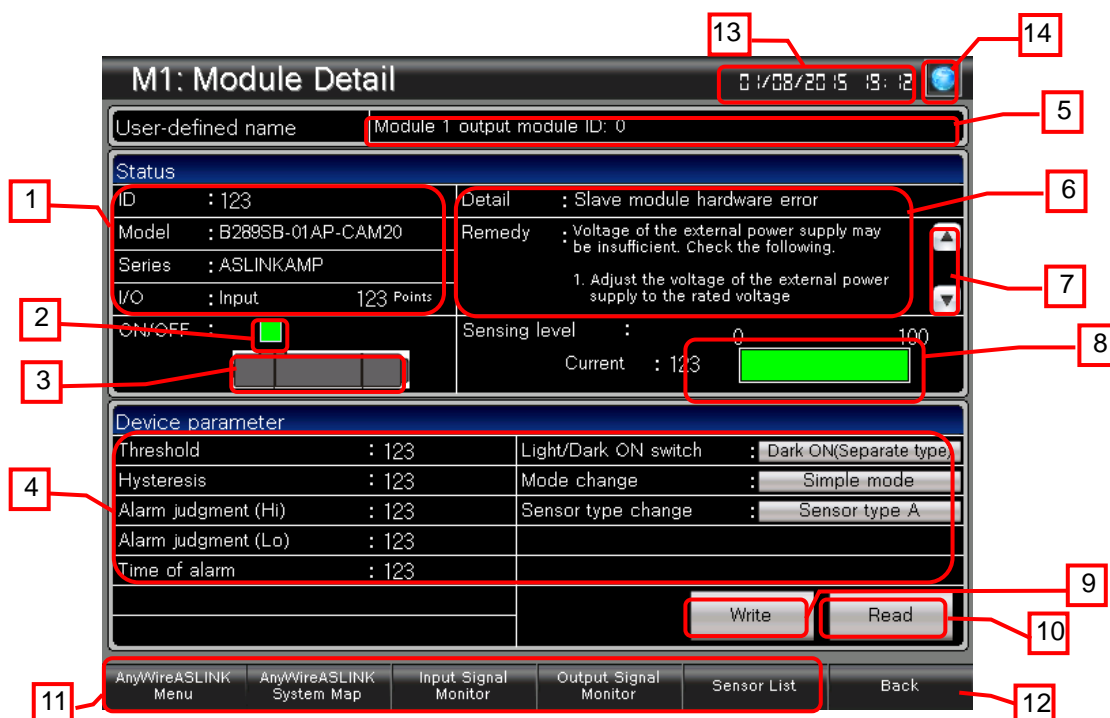
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
9. Reads parameters of the slave module.
10. Switches to each screen.
11. Switches to the previously opened screen.
12. Displays the current date and time. Touch the area to open the [Clock Setting] window.
13. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.11 Detail (Photoelectric [CAM, CAS]) (B-30010)



#### Outline

This screen displays and sets the detail information about the slave module of the photoelectric amplifier module.

#### Description

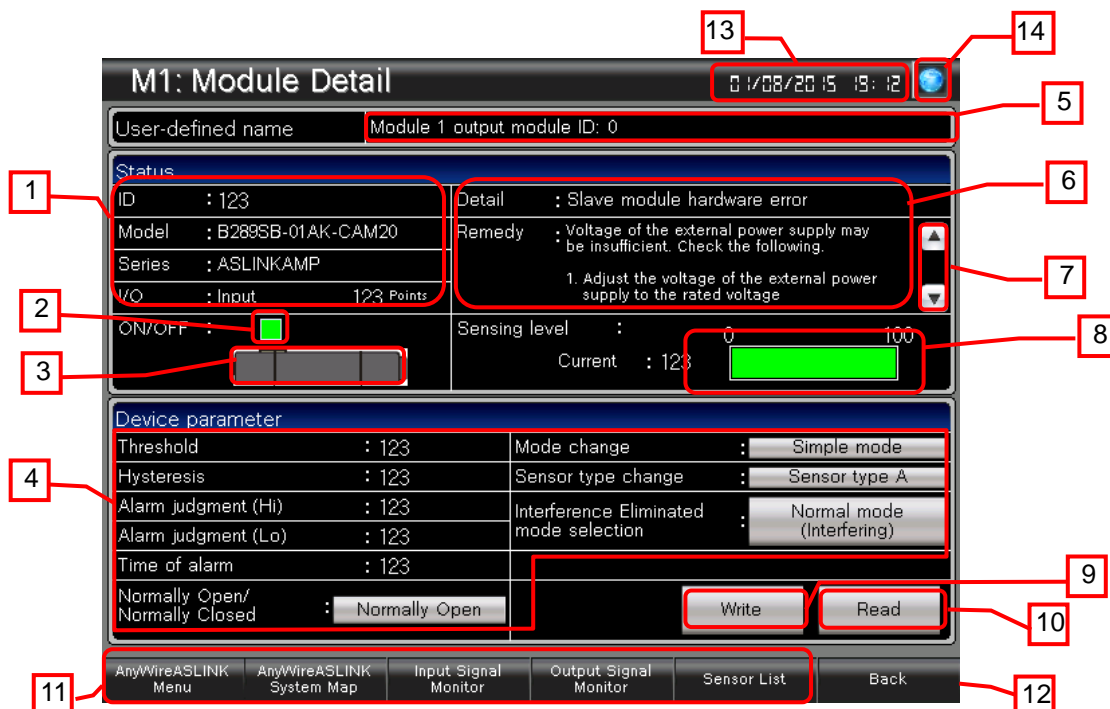
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the I/O status.
3. Displays the image of the slave module.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the slave module may change. Be sure to confirm safety before execution.
- Object scripts are set for the numerical displays of the level for "Sensing level", "Threshold", "Alarm judgment (Hi)", and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.12 Detail (Proximity Amp) (B-30011)



#### Outline

This screen displays and sets the detail information about the slave module of the proximity amplifier module.

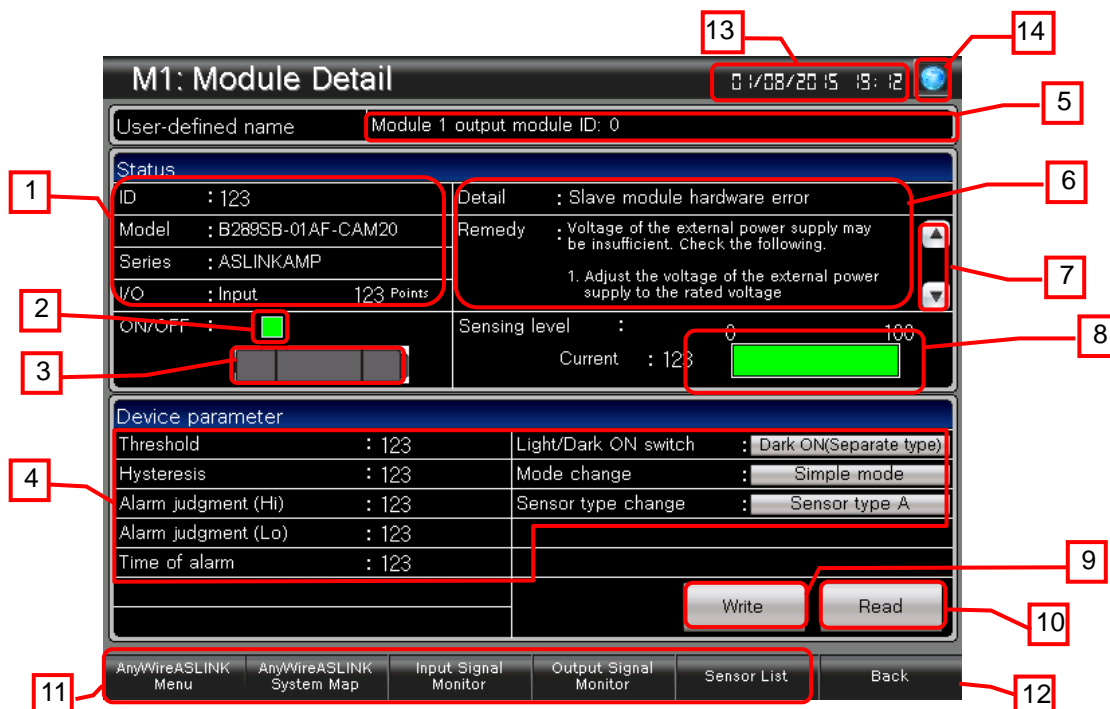
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the I/O status.
3. Displays the image of the slave module.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.
- 15.

#### Remarks

- If parameters are changed while the system is running, the action of the slave module may change. Be sure to confirm safety before execution.
- Object scripts are set for the numerical displays of the level for "Sensing level", "Threshold", "Alarm judgment (Hi)", and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.13 Detail (Fiber Amp [CAM, CAS]) (B-30012)



#### Outline

This screen displays and sets the detail information about the slave module of the fiber amplifier module.

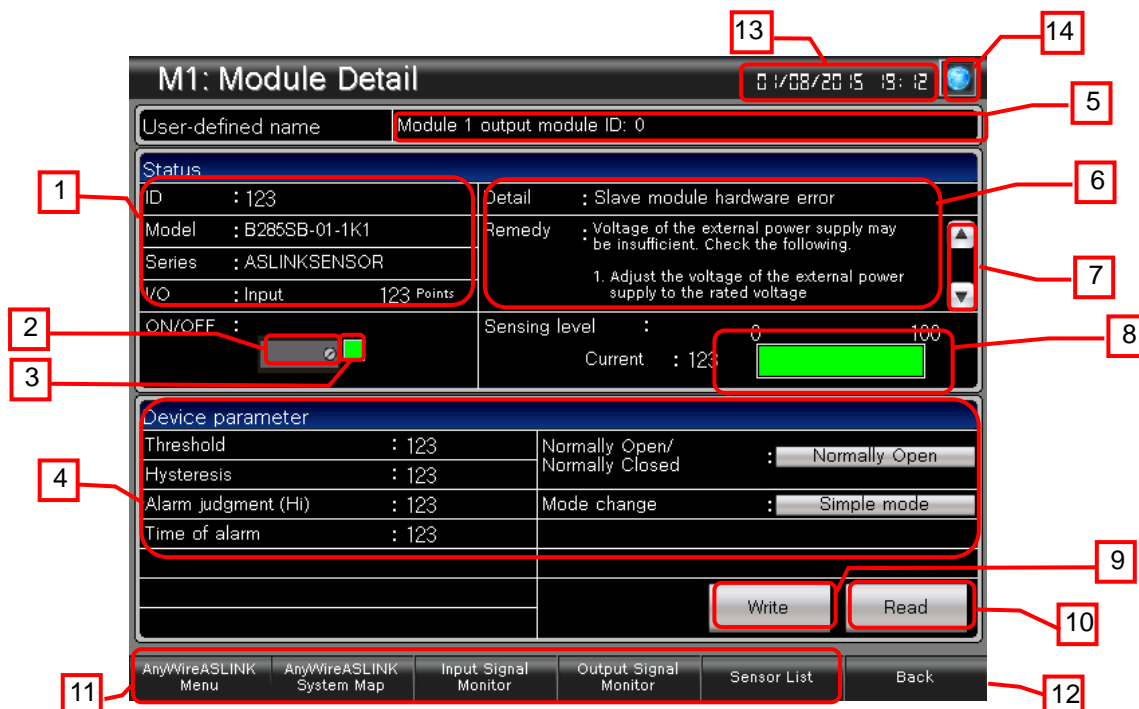
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the I/O status.
3. Displays the image of the slave module.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.
- 15.

#### Remarks

- If parameters are changed while the system is running, the action of the slave module may change. Be sure to confirm safety before execution.
- Object scripts are set for the numerical displays of the level for "Sensing level", "Threshold", "Alarm judgment (Hi)", and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.14 Detail (Cylinder Switch) (B-30013)



#### Outline

This screen displays and sets the detail information about the slave module of the cylinder switch.

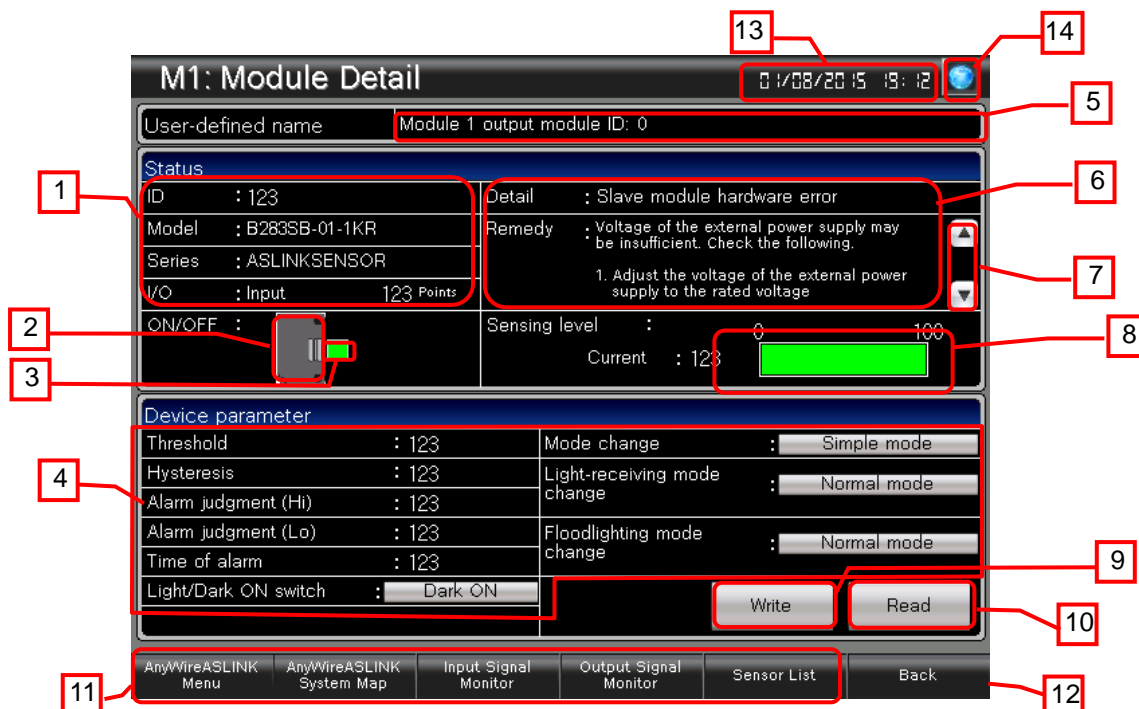
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- Object scripts are set for the level of "Sensing level", and also for the numerical displays of "Threshold" and "Hysteresis". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.15 Detail (Photoelectric Sensor) (B-30014)



#### Outline

This screen displays and sets the detail information about the slave module of the photoelectric sensor.

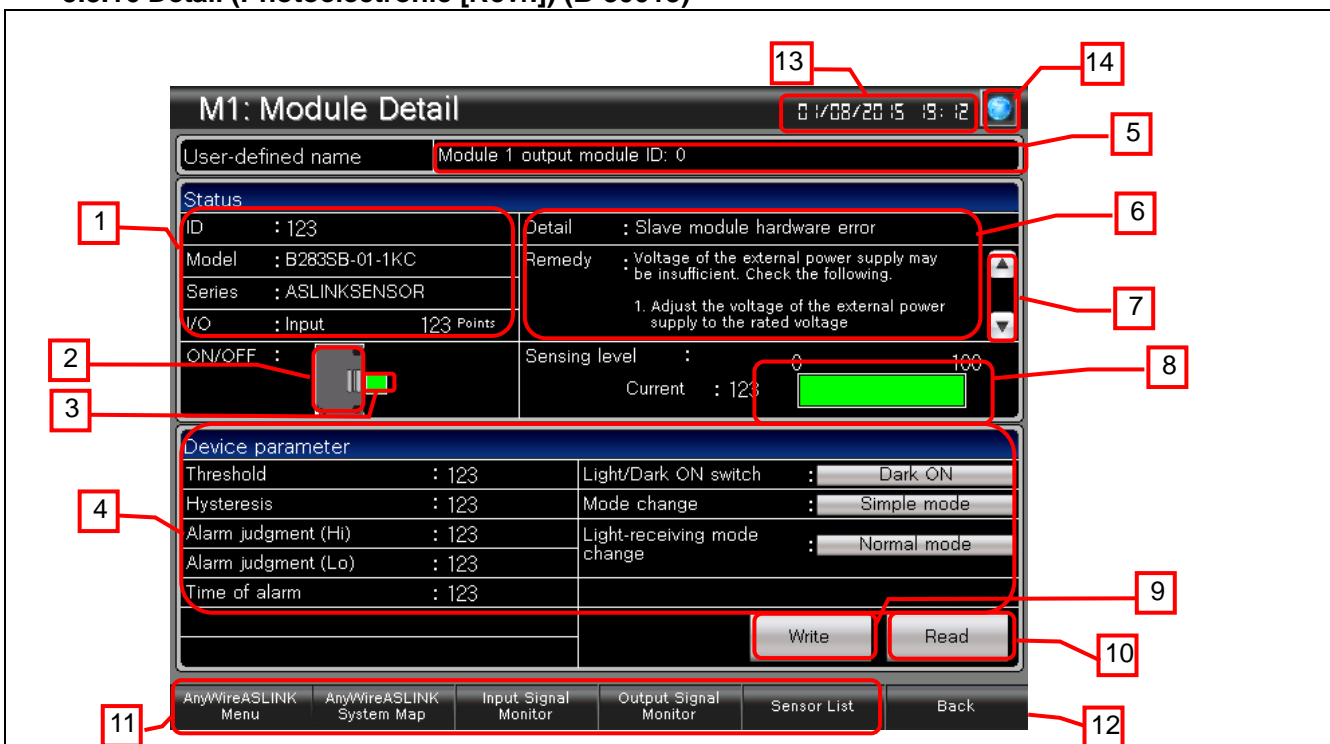
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.
- 15.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- Object scripts are set for the numerical displays of the level for "Sensing level", "Threshold", "Alarm judgment (Hi)", and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.16 Detail (Photoelectric [Rcvr.]) (B-30015)



#### Outline

This screen displays and sets the detail information about the slave module of the photoelectric sensor (receiver).

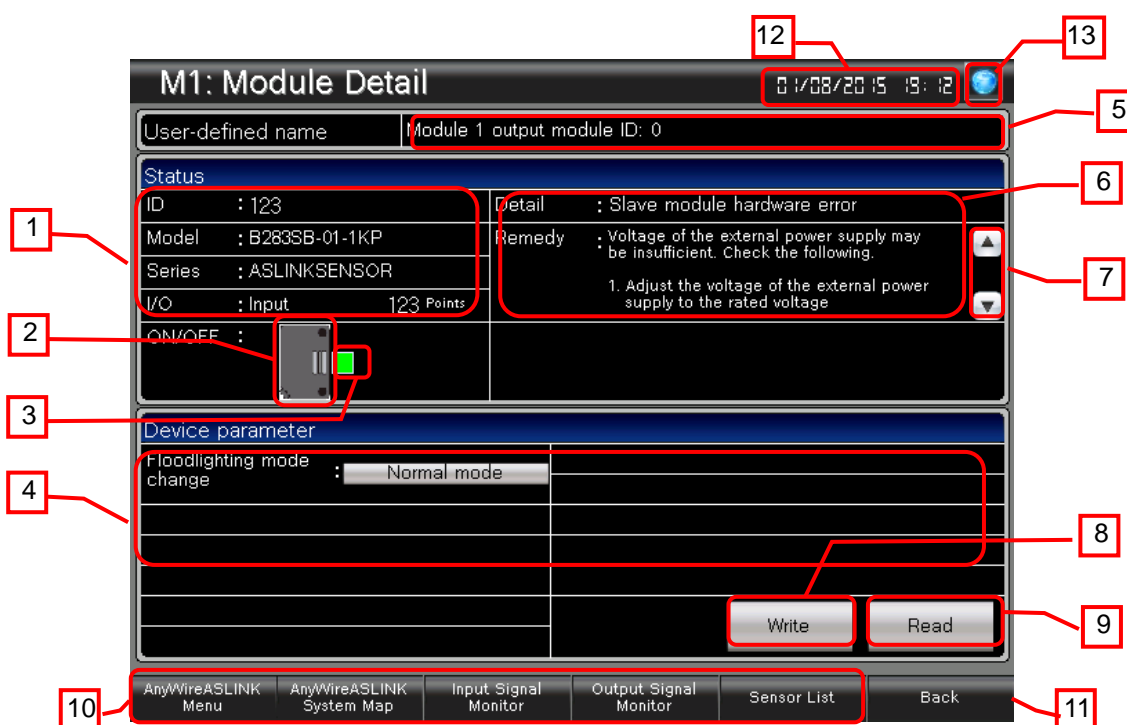
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.
- 15.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- Object scripts are set for the numerical displays of the level for "Sensing level", "Threshold", "Alarm judgment (Hi)", and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.17 Detail (Photoelectric [LT Src.]) (B-30016)



#### Outline

This screen displays and sets the detail information about the slave module of the photoelectric sensor (light source).

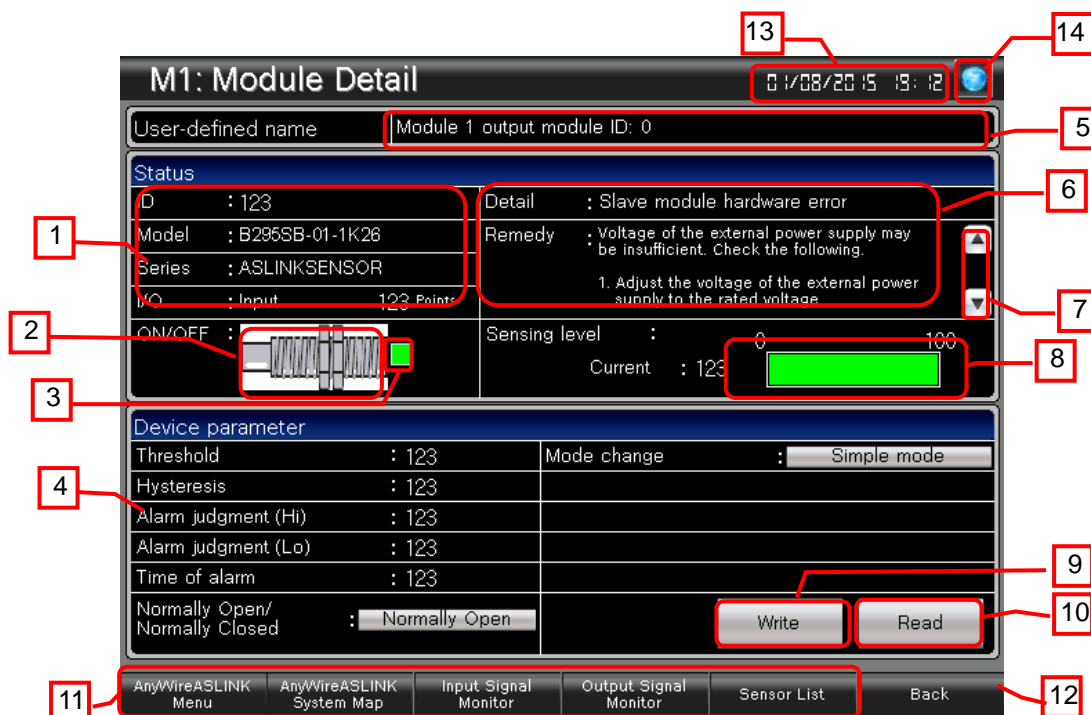
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
9. Reads parameters of the slave module.
10. Switches to each screen.
11. Switches to the previously opened screen.
12. Displays the current date and time. Touch the area to open the [Clock Setting] window.
13. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.18 Detail (Proximity Switch) (B-30017)



#### Outline

This screen displays and sets the detail information about the slave module of the proximity switch.

#### Description

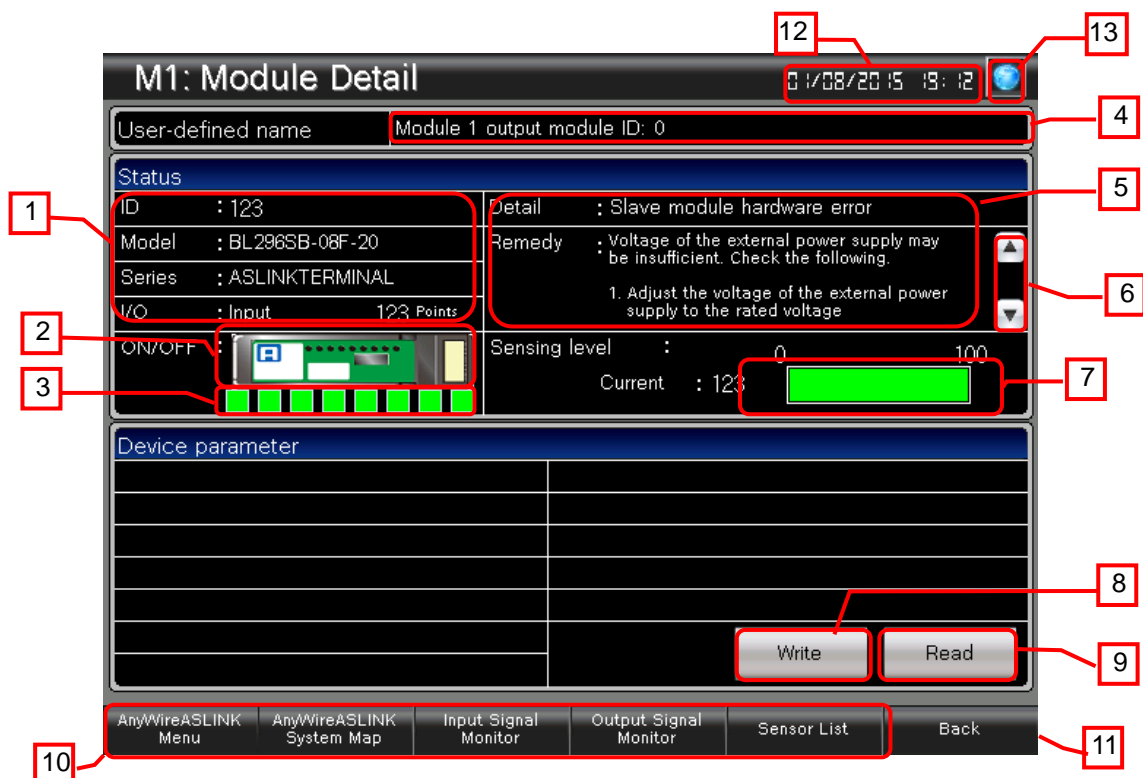
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- Object scripts are set for the level of "Sensing level", and also for the numerical displays of "Threshold", "Hysteresis", "Alarm judgment (Hi)" and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.19 Detail (ASLINKTERMINAL [Driver]) (B-30018)



#### Outline

This screen displays the detail information about the slave module of the ASLINKTERMINAL (driver module).

#### Description

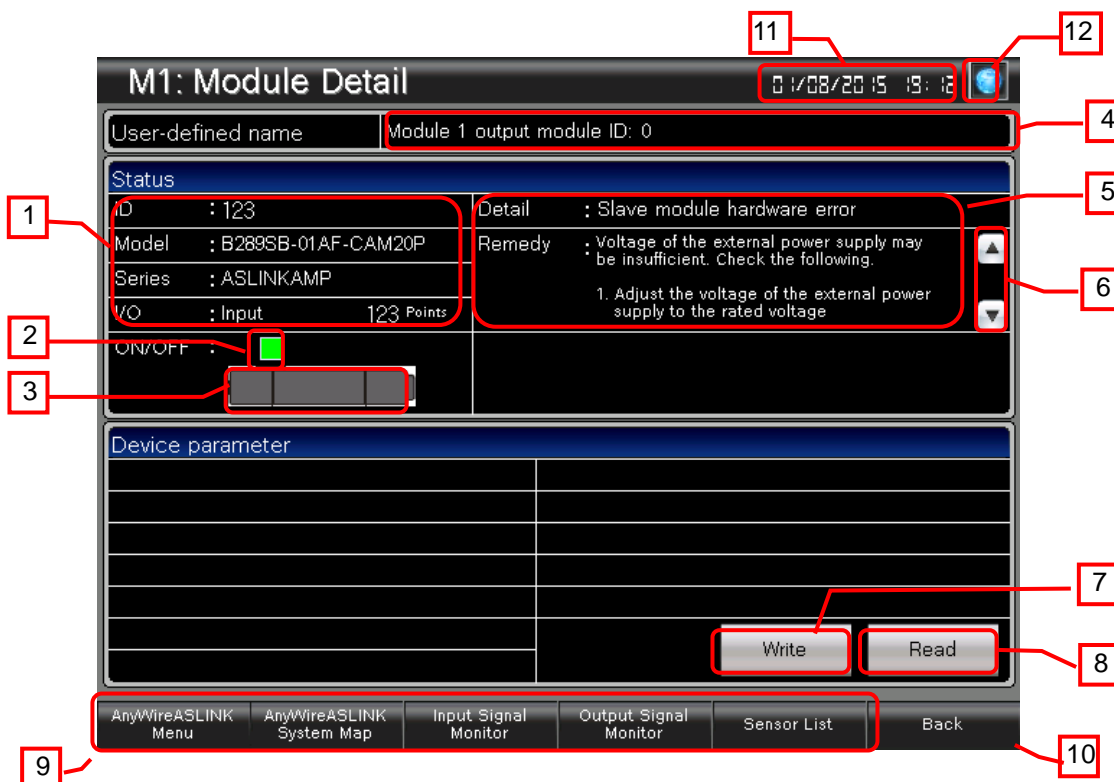
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. Displays the current value of the sensing level with a numerical display and a level.
8. This switch does not work because there are no parameters to write.
9. This switch does not work because there are no parameters to read.
10. Switches to each screen.
11. Switches to the previously opened screen.
12. Displays the current date and time. Touch the area to open the [Clock Setting] window.
13. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- Object script is set for the level of "Sensing level". For more details about scripts, please refer to "5.9 Script List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.20 Detail(Fiber Amp[LT Src. Only]) (B-30019)



#### Outline

This screen displays and sets the detail information about the slave module of the fiber amplifier module (light source only).

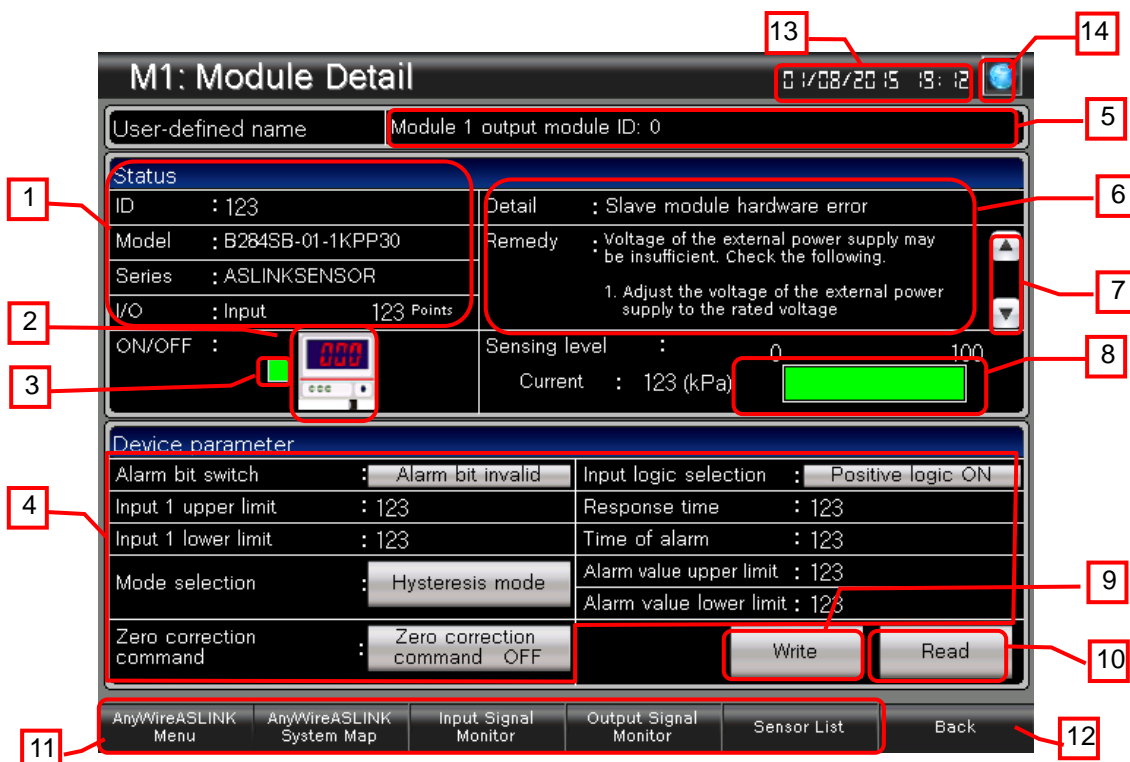
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the I/O status.
3. Displays the image of the slave module.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.21 Detail(+Pressure Sensor 1 Pt) (B-30020)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (positive pressure, 1 point).

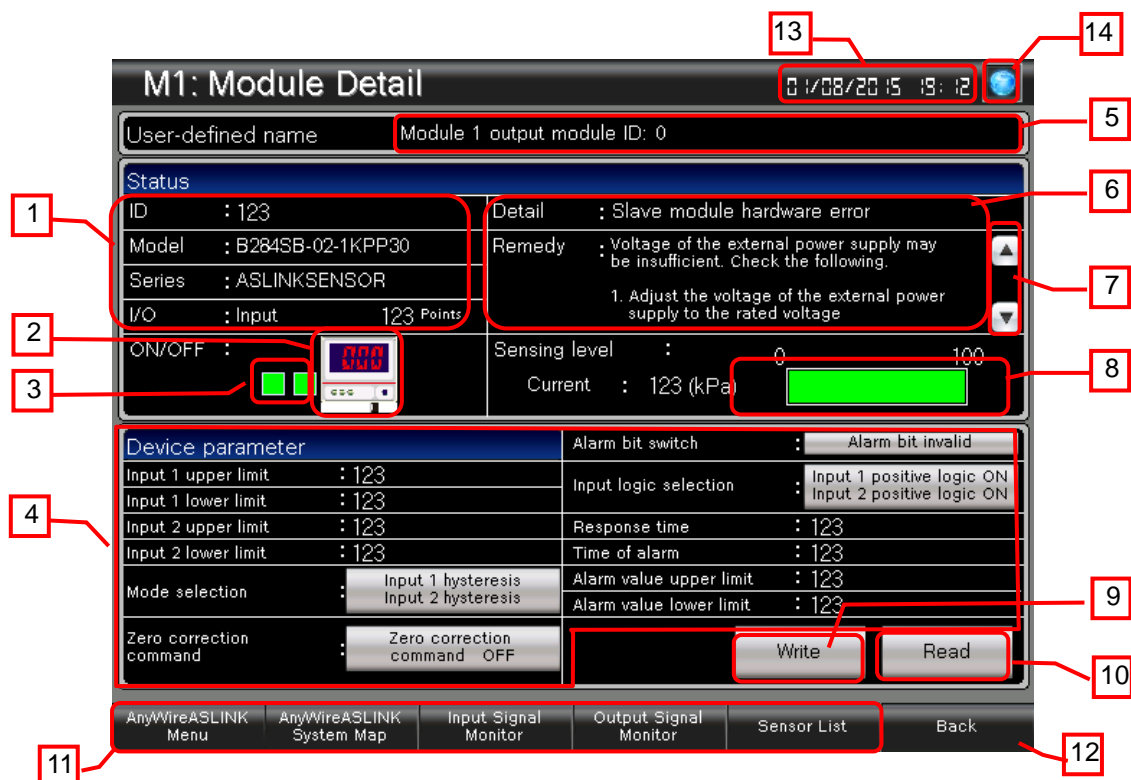
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.22 Detail(+Pressure Sensor 2 Pts) (B-30021)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (positive pressure, 2 points).

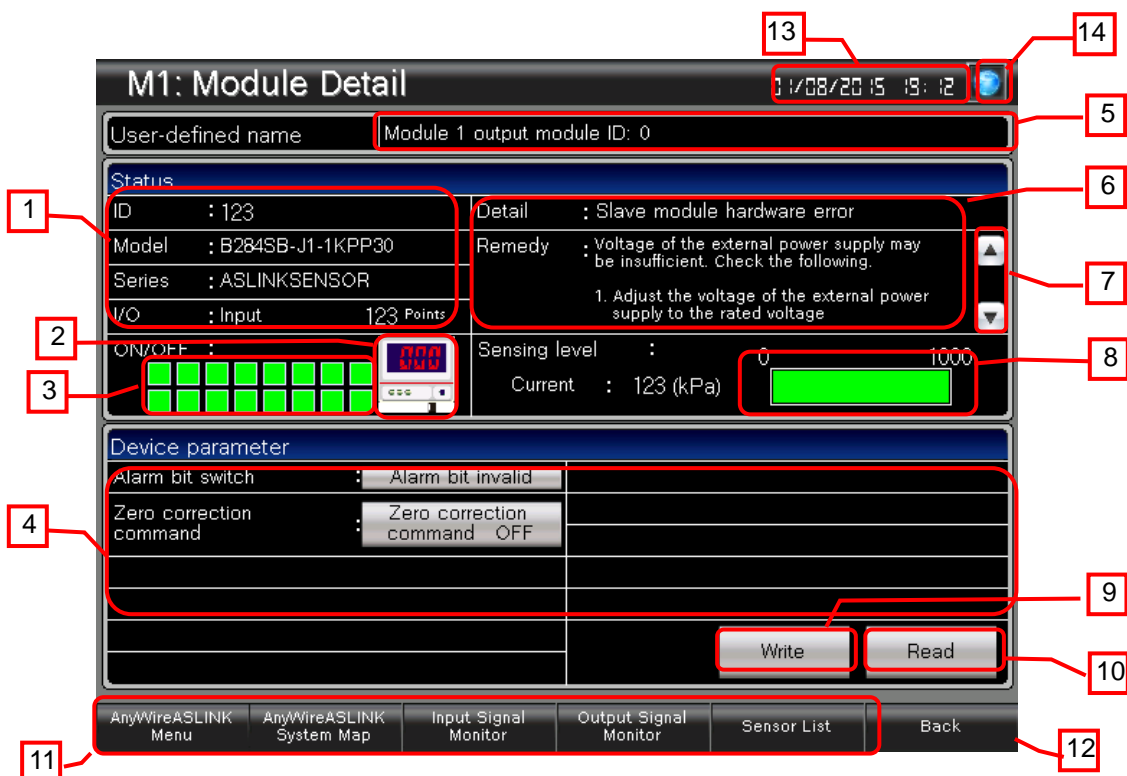
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.23 Detail(+Pressure Sensor 16 Pts) (B-30023)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (positive pressure, 16 points).

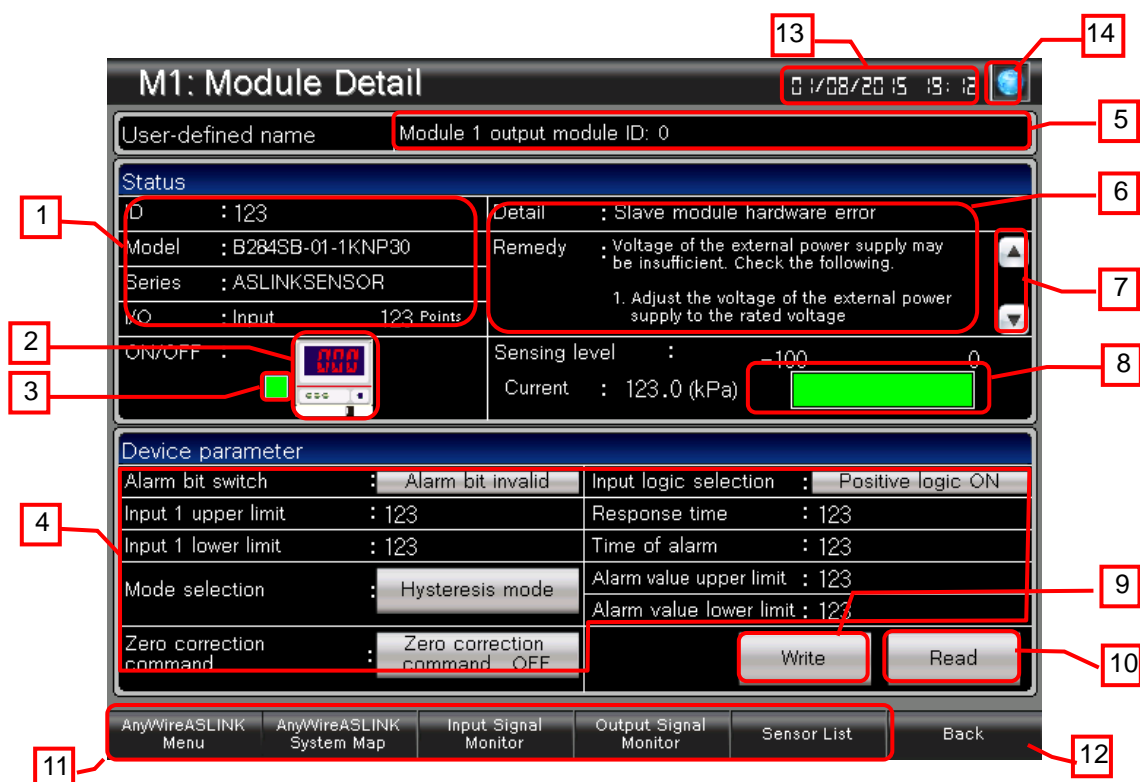
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.24 Detail(-Pressure Sensor 1 Pt) (B-30024)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (negative pressure, 1 point).

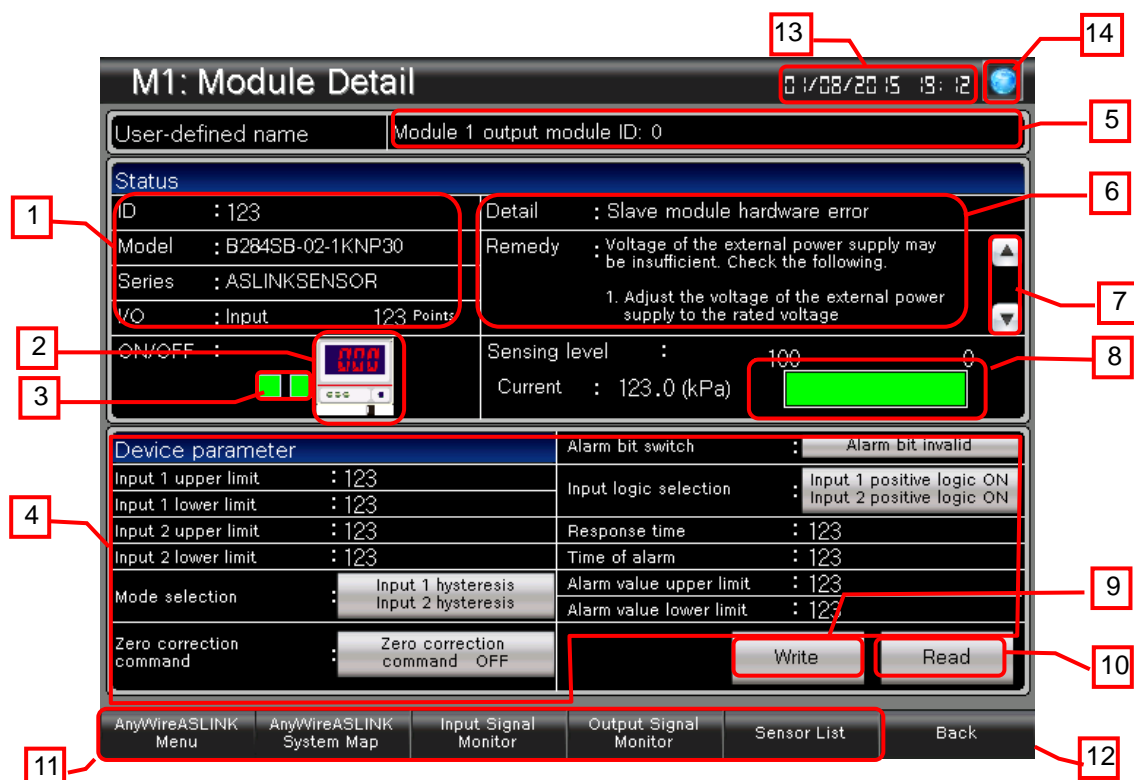
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.25 Detail(-Pressure Sensor 2 Pts) (B-30025)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (negative pressure, 2 points).

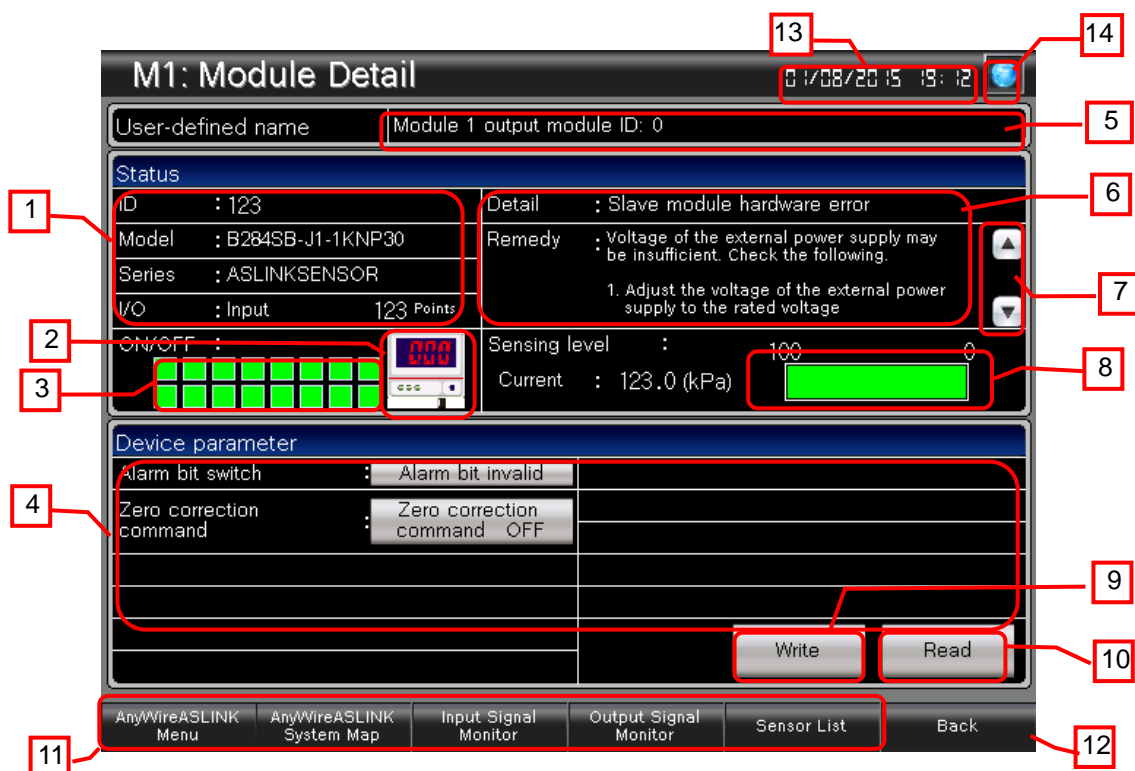
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.26 Detail(-Pressure Sensor 16 Pts) (B-30027)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (negative pressure, 16 points).

#### Description

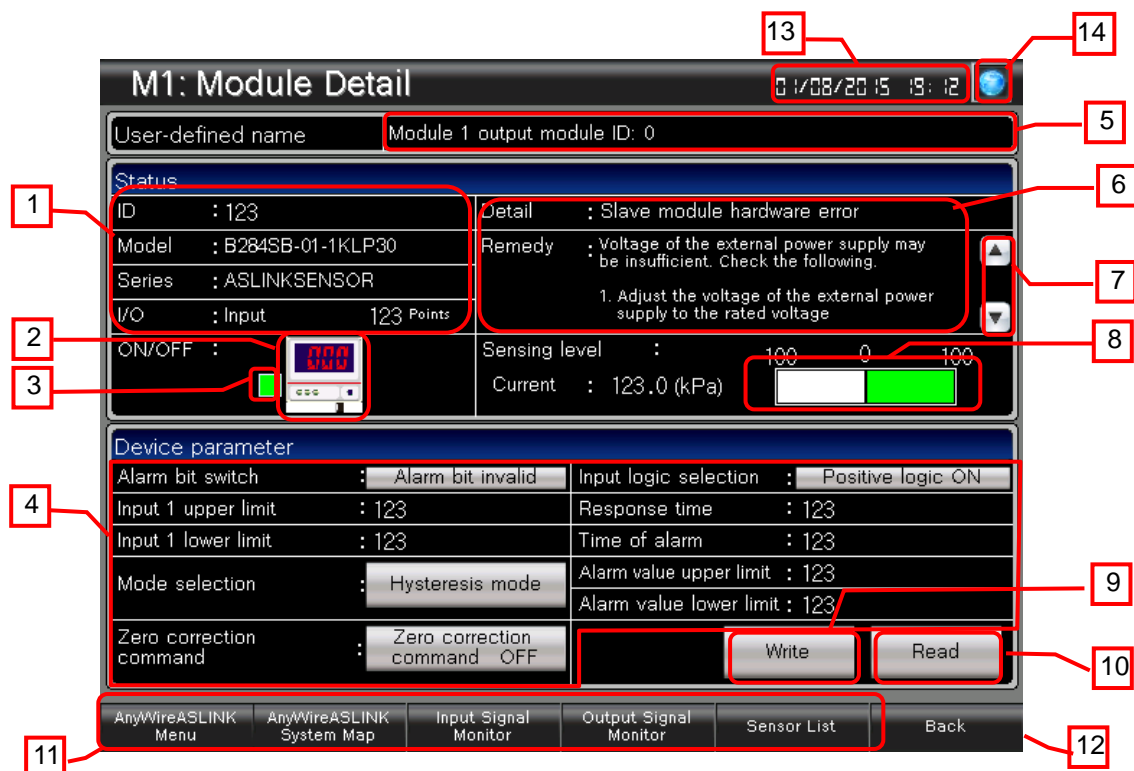
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.27 Detail(Compound Pressure 1 Pt) (B-30028)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (compound pressure, 1 point).

#### Description

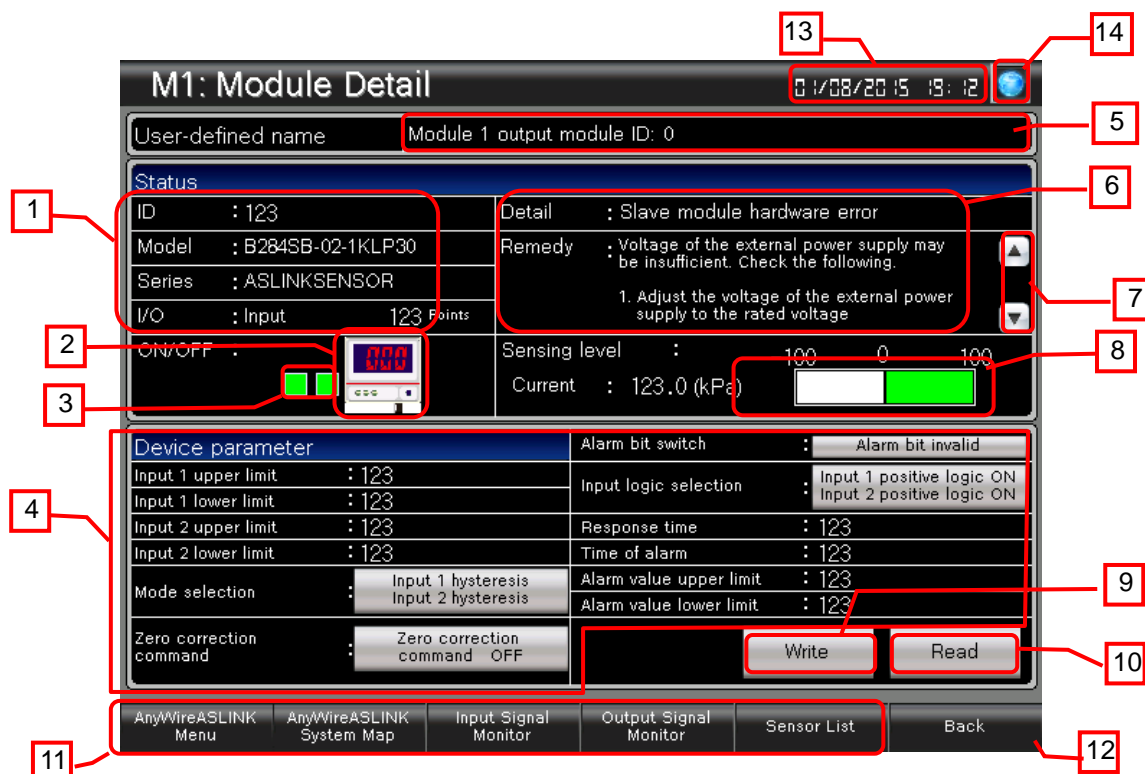
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.28 Detail(Compound Pressure 2 Pt) (B-30029)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (compound pressure, 2 points).

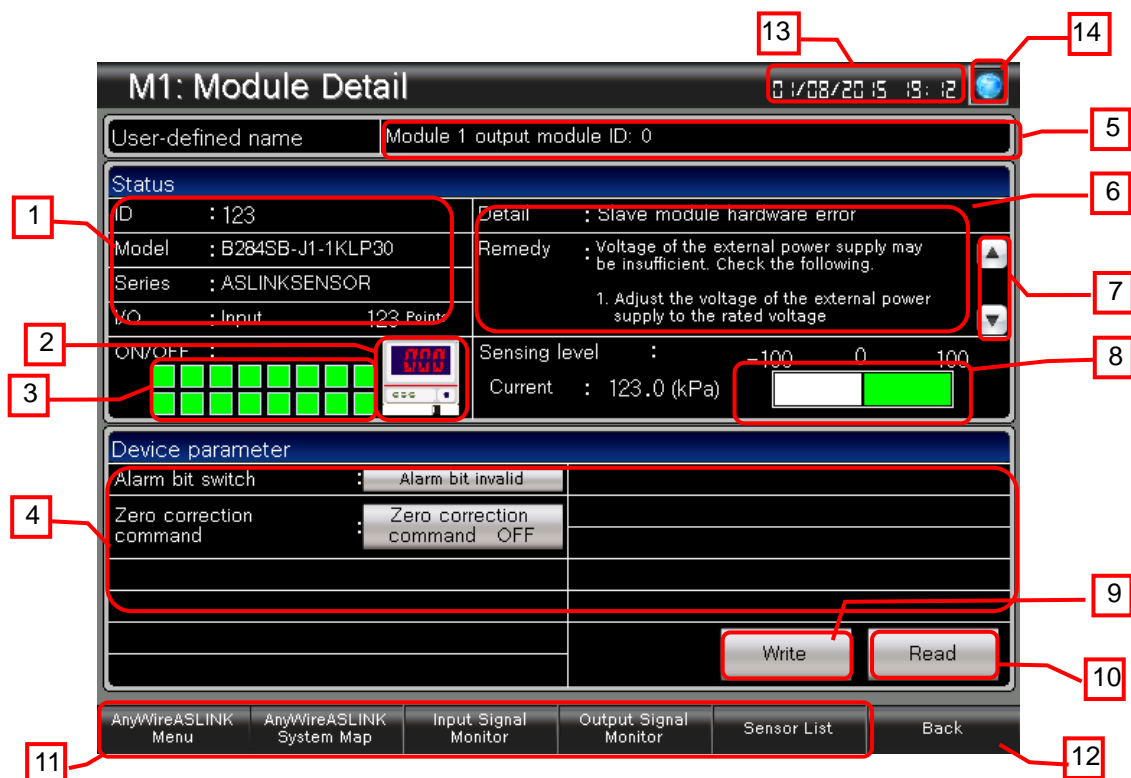
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.29 Detail(Compound Pressure 16 Pt) (B-30031)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (compound pressure, 16 points).

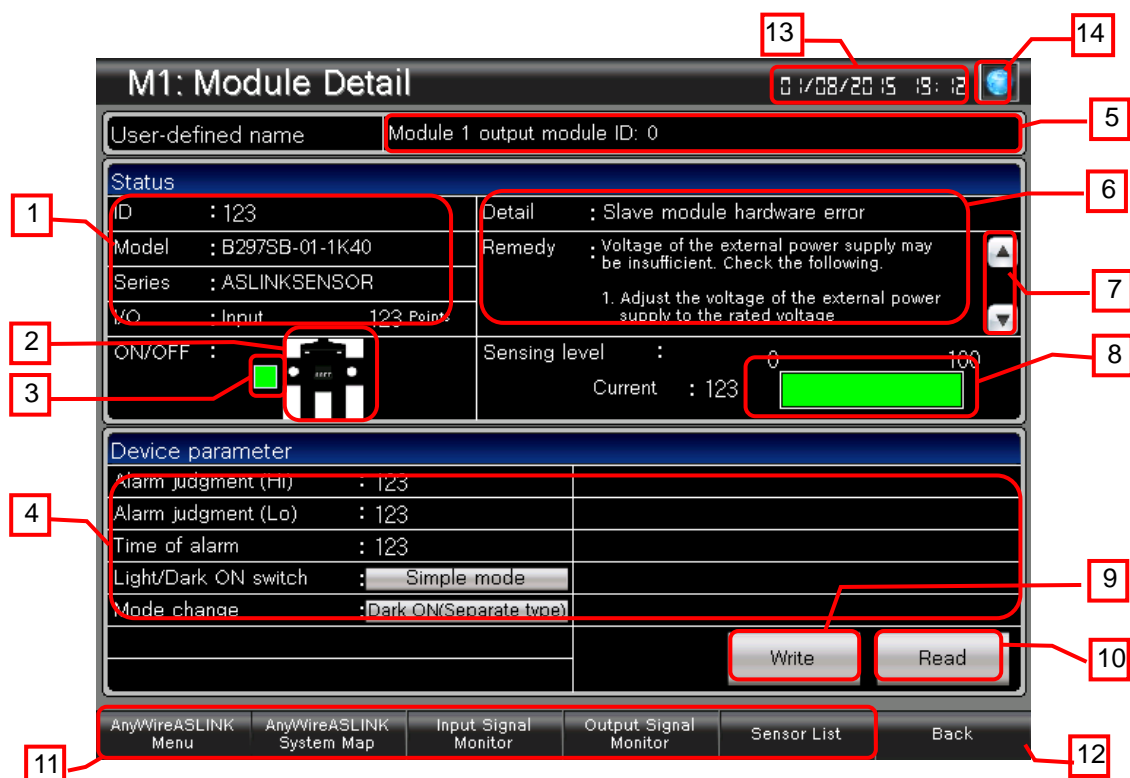
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.30 Detail(Photo Interrupter) (B-30032)



#### Outline

This screen displays and sets the detail information about the slave module of the photo interrupter.

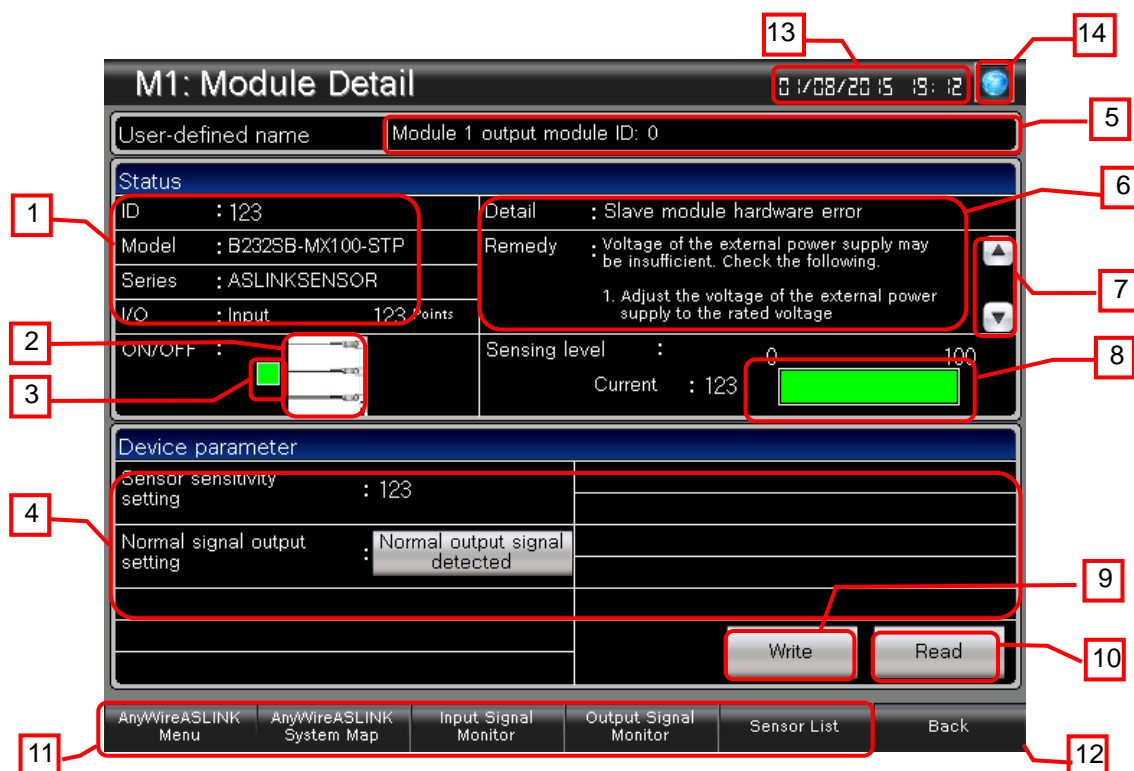
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- Object scripts are set for the level of "Sensing level", and also for the numerical displays of "Alarm judgment (Hi)" and "Alarm judgment (Lo)". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.31 Detail(Mapping [Comb Master]) (B-30033)



#### Outline

This screen displays and sets the detail information about the slave module of the mapping sensor (comb type master).

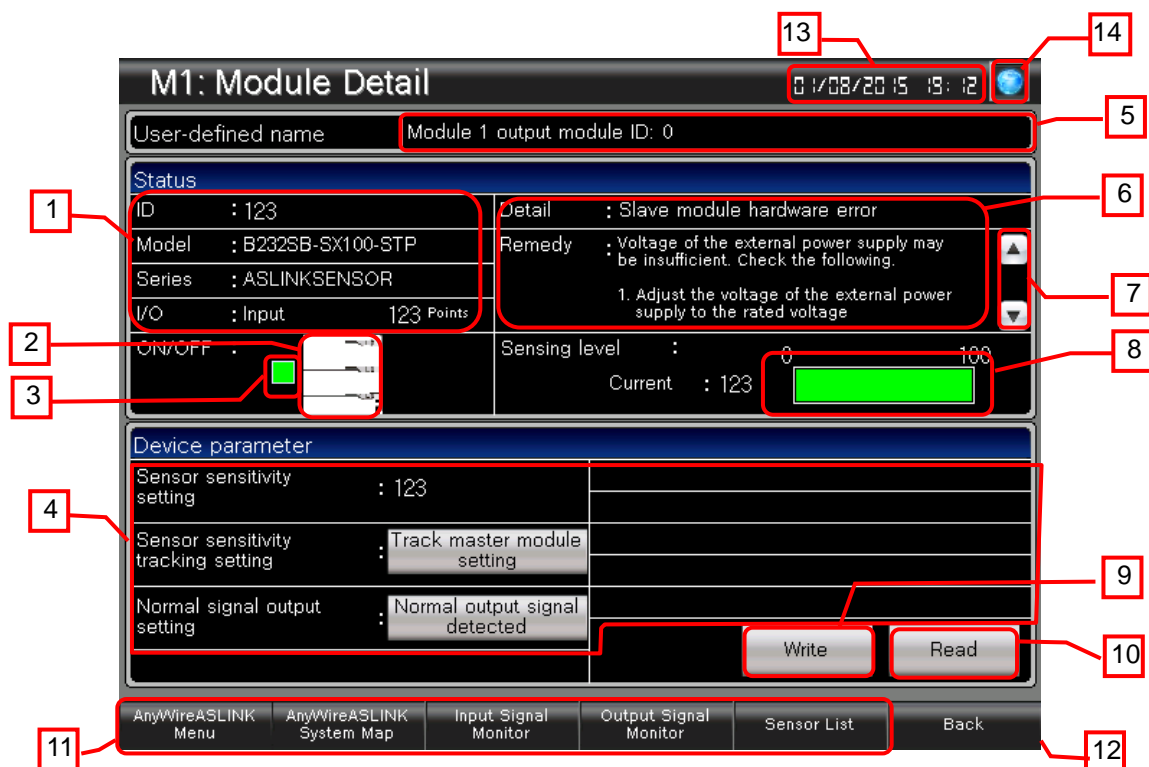
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.32 Detail(Mapping [Comb Slave]) (B-30034)



#### Outline

This screen displays and sets the detail information about the slave module of the mapping sensor (comb type slave).

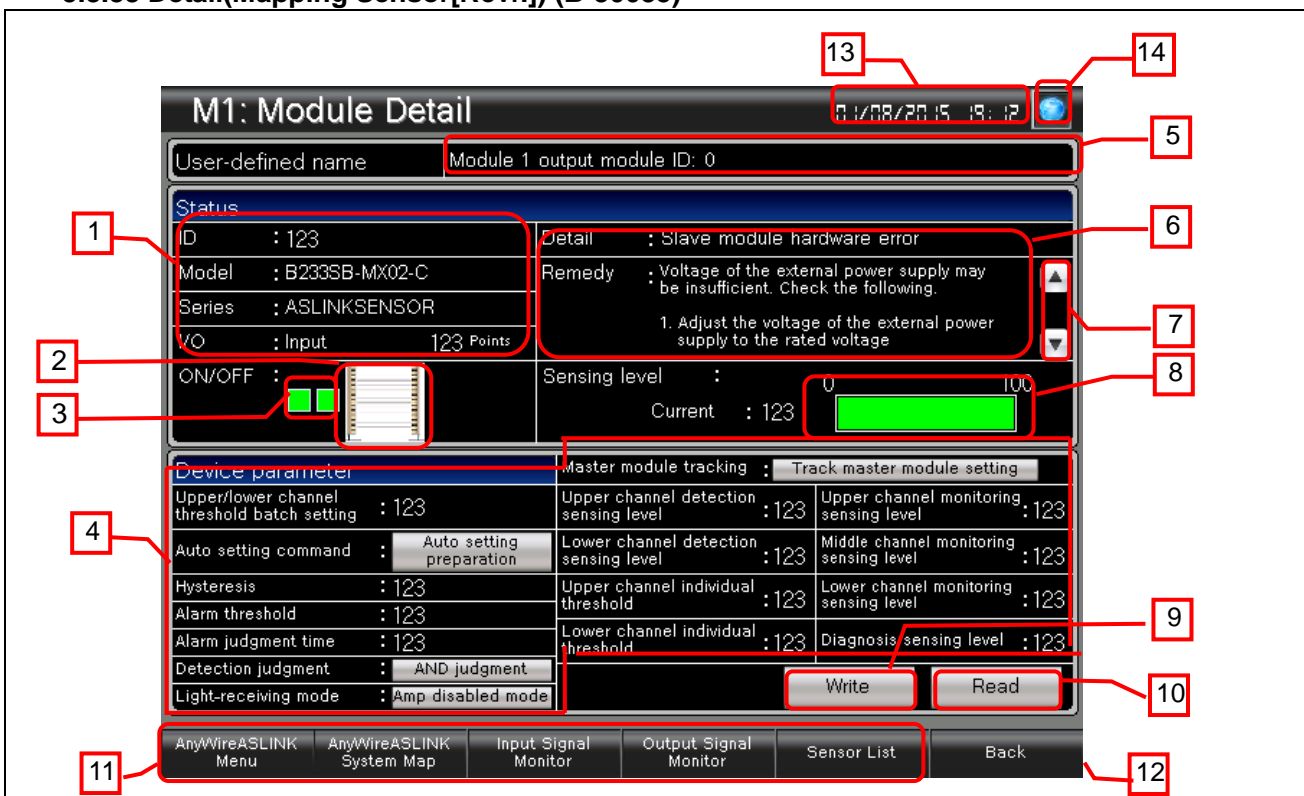
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.33 Detail(Mapping Sensor[Rcvr.]) (B-30035)



#### Outline

This screen displays and sets the detail information about the slave module of the mapping sensor (receiver).

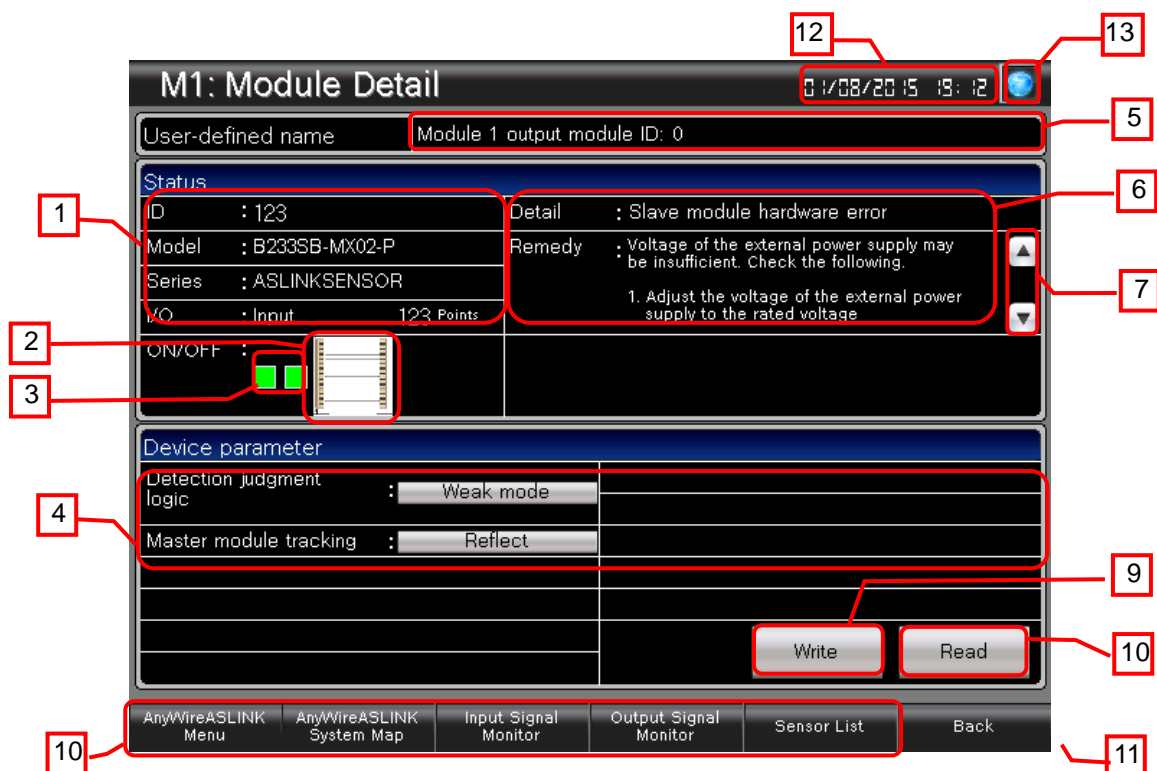
#### Description

- Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
- Displays the image of the slave module.
- Displays the I/O status.
- Displays and changes parameters of the slave module.
- Displays the user-defined name.
- Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
- Scrolls the remedy display.
- Displays the current value of the sensing level with a numerical display and a level.
- Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
- Reads parameters of the slave module.
- Switches to each screen.
- Switches to the previously opened screen.
- Displays the current date and time. Touch the area to open the [Clock Setting] window.
- Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- Object scripts are set for the numerical displays of the level for "Sensing level", and "Alarm Threshold". For more details about scripts, please refer to "5.9 Script List".
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.34 Detail(Mapping Sensor[LT Src.]) (B-30036)



#### Outline

This screen displays and sets the detail information about the slave module of the mapping sensor (light source).

#### Description

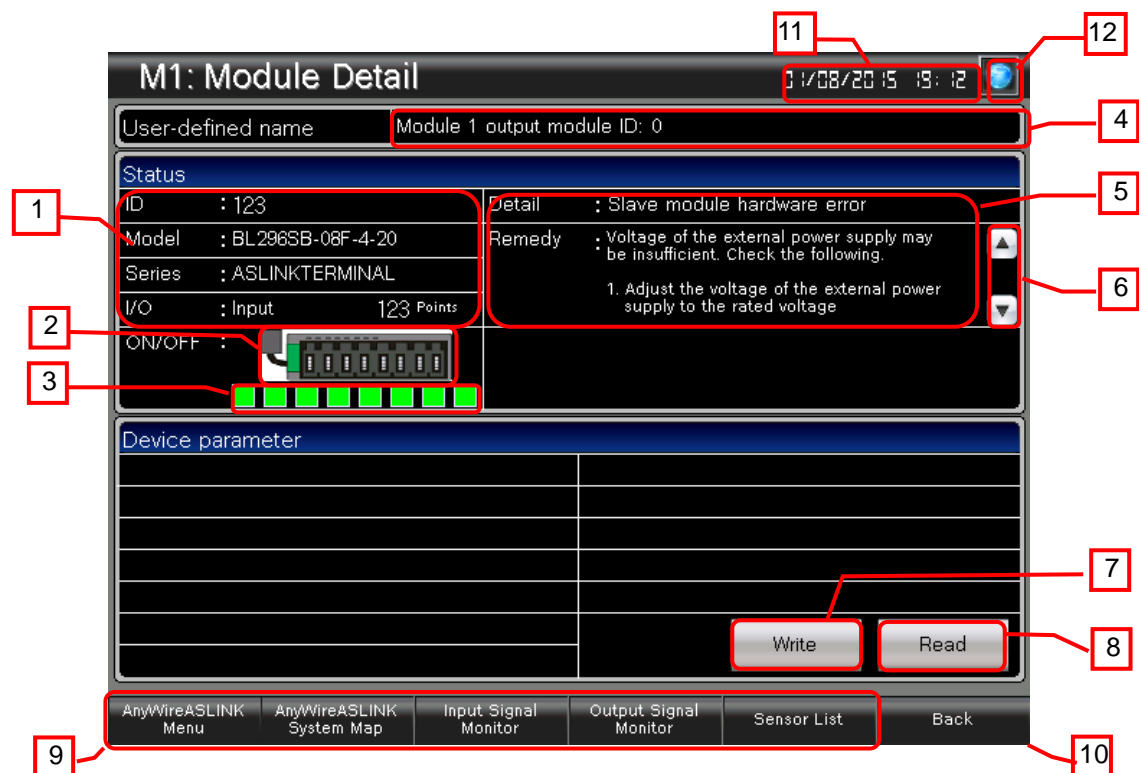
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
9. Reads parameters of the slave module.
10. Switches to each screen.
11. Switches to the previously opened screen.
12. Displays the current date and time. Touch the area to open the [Clock Setting] window.
13. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.35 Detail(Incorporated[e-con]) (B-30037)



#### Outline

This screen displays and sets the detail information about the slave module of the incorporated type [e-con connector].

#### Description

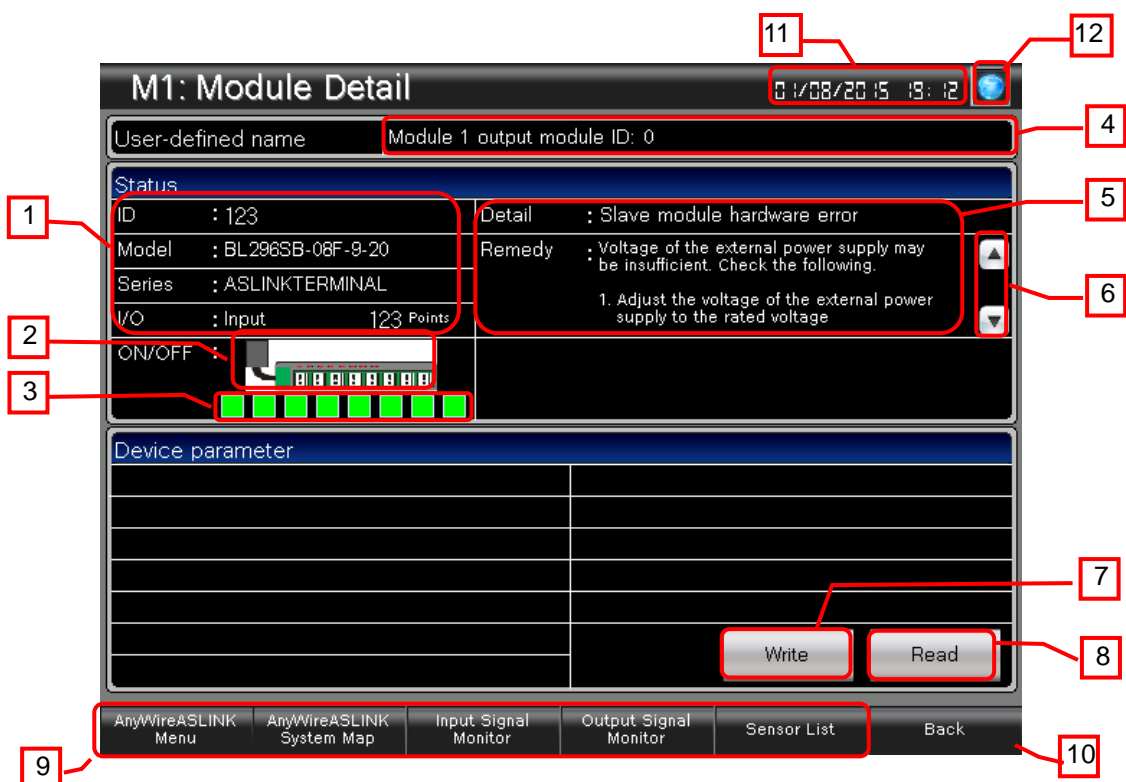
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.36 Detail(Incorporated[JST]) (B-30038)



#### Outline

This screen displays and sets the detail information about the slave module of the incorporated type (JST connector).

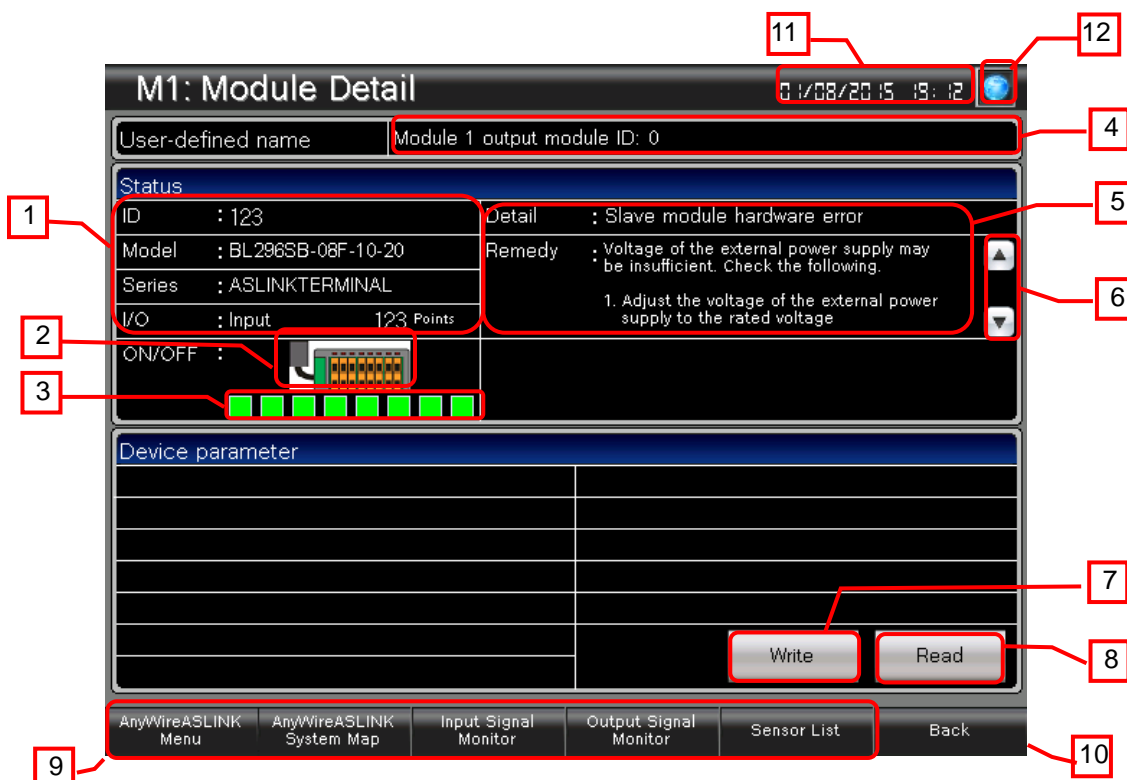
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.37 Detail(Incorporated[MOLEX]) (B-30039)



#### Outline

This screen displays and sets the detail information about the slave module of the incorporated type (MOLEX connector).

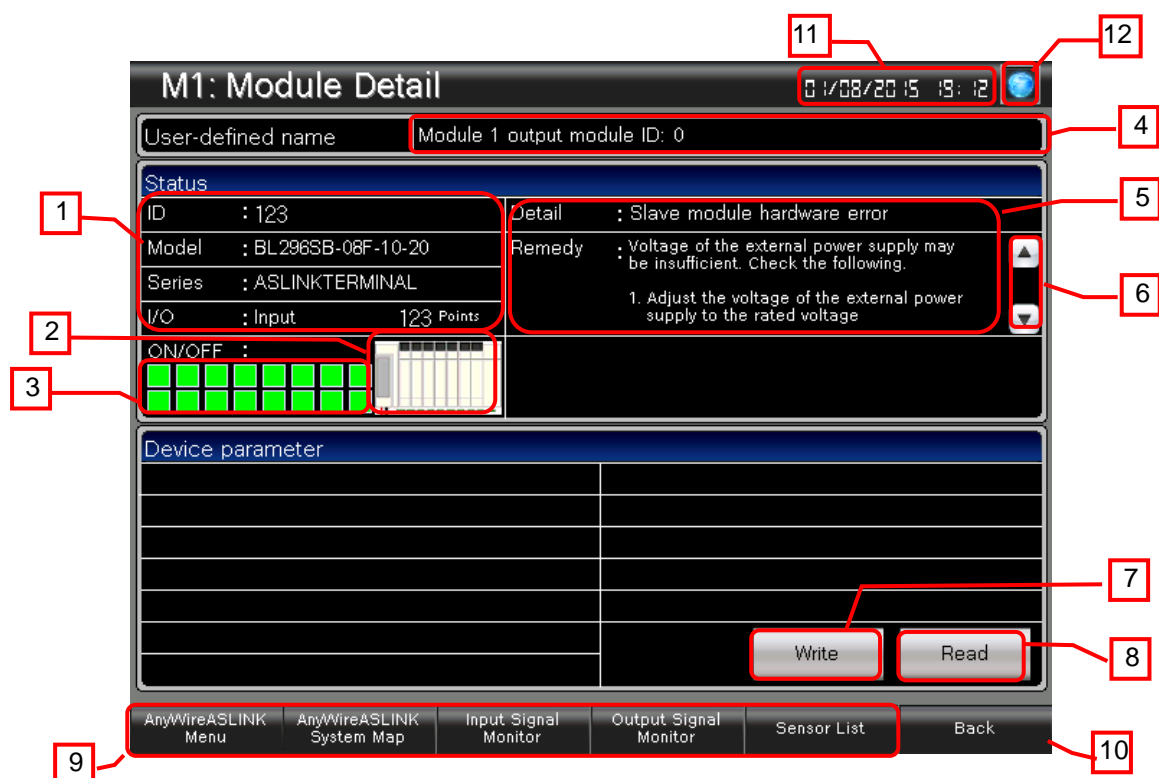
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.38 Detail(Manifold Driver) (B-30040)



#### Outline

This screen displays and sets the detail information about the slave module of the manifold driver.

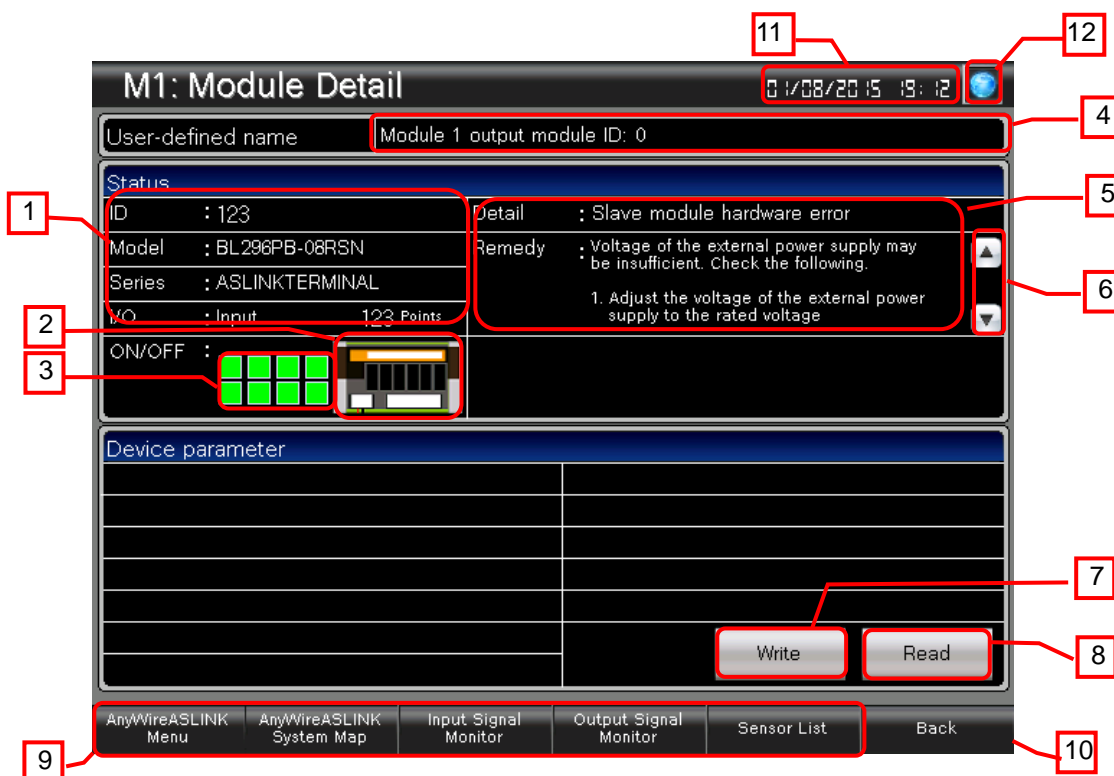
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.39 Detail(Relay Output[Driver]) (B-30041)



#### Outline

This screen displays and sets the detail information about the slave module of the incorporated type relay output (driver).

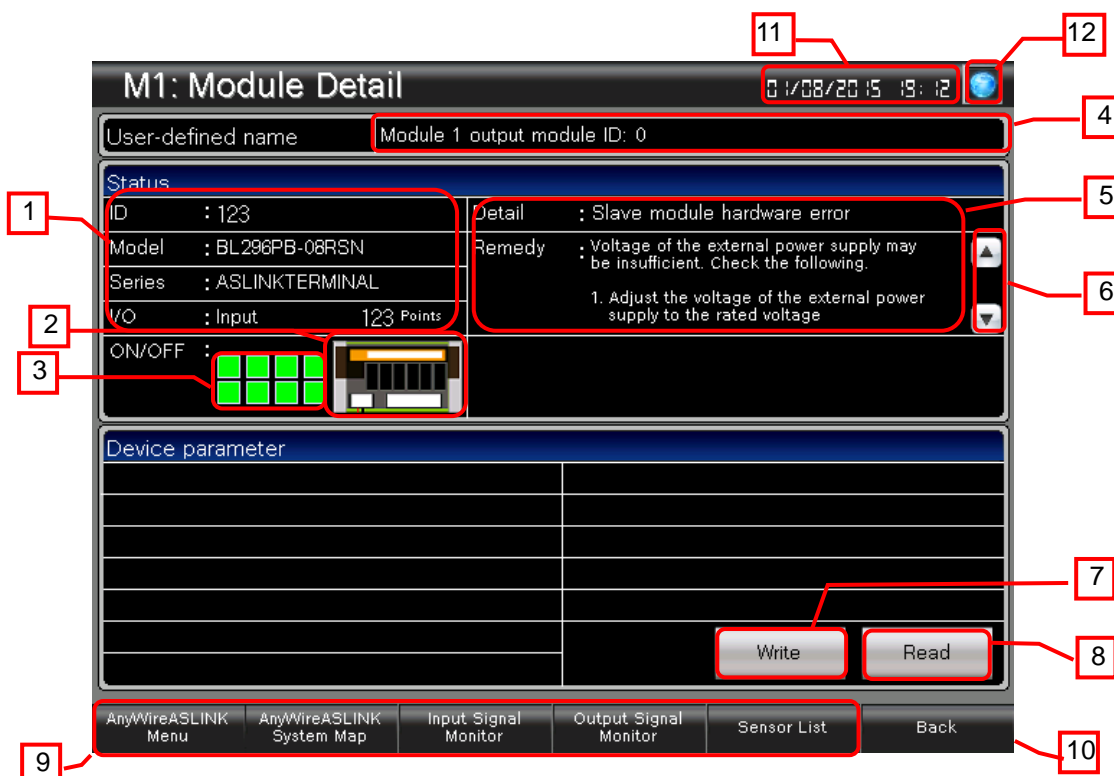
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.40 Detail(Relay Output) (B-30042)



#### Outline

This screen displays and sets the detail information about the slave module of incorporated type relay output.

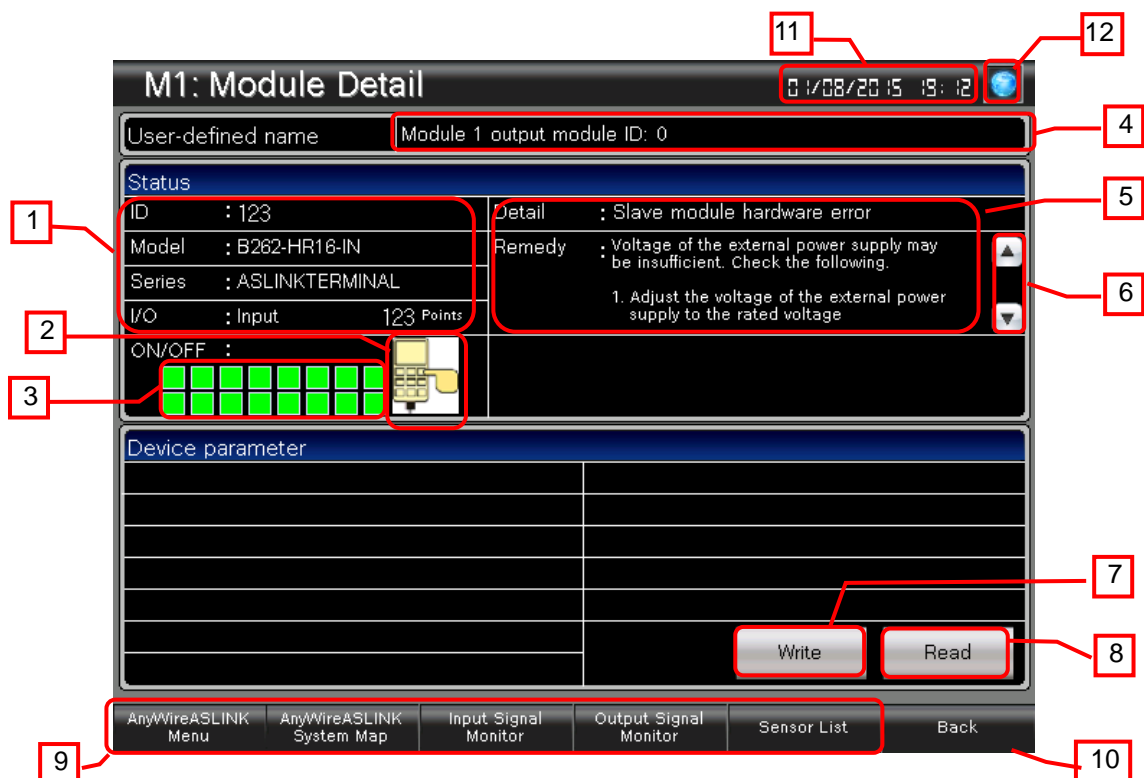
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.41 Detail(Handy Remote[SwitchSide]) (B-30051)



#### Outline

This screen displays and sets the detail information about the slave module of the handy remote (switch side).

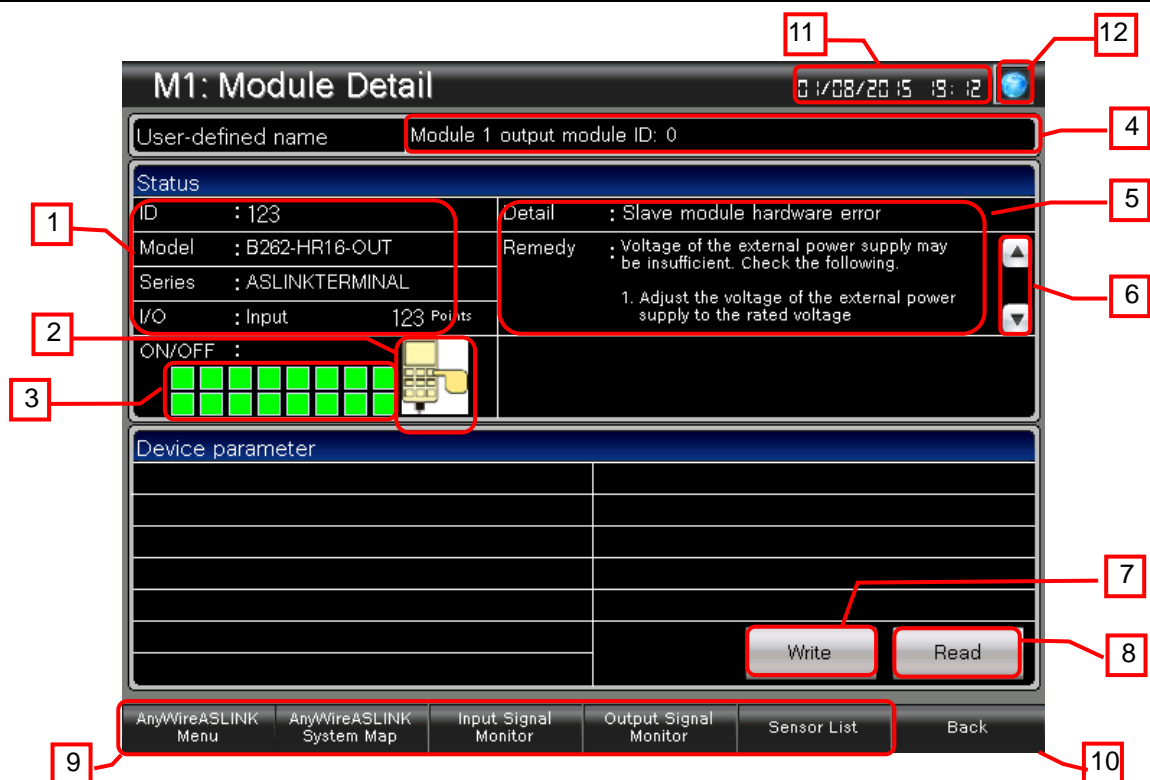
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.42 Detail(Handy Remote[LED Side]) (B-30052)



#### Outline

This screen displays and sets the detail information about the slave module of handy remote (LED side).

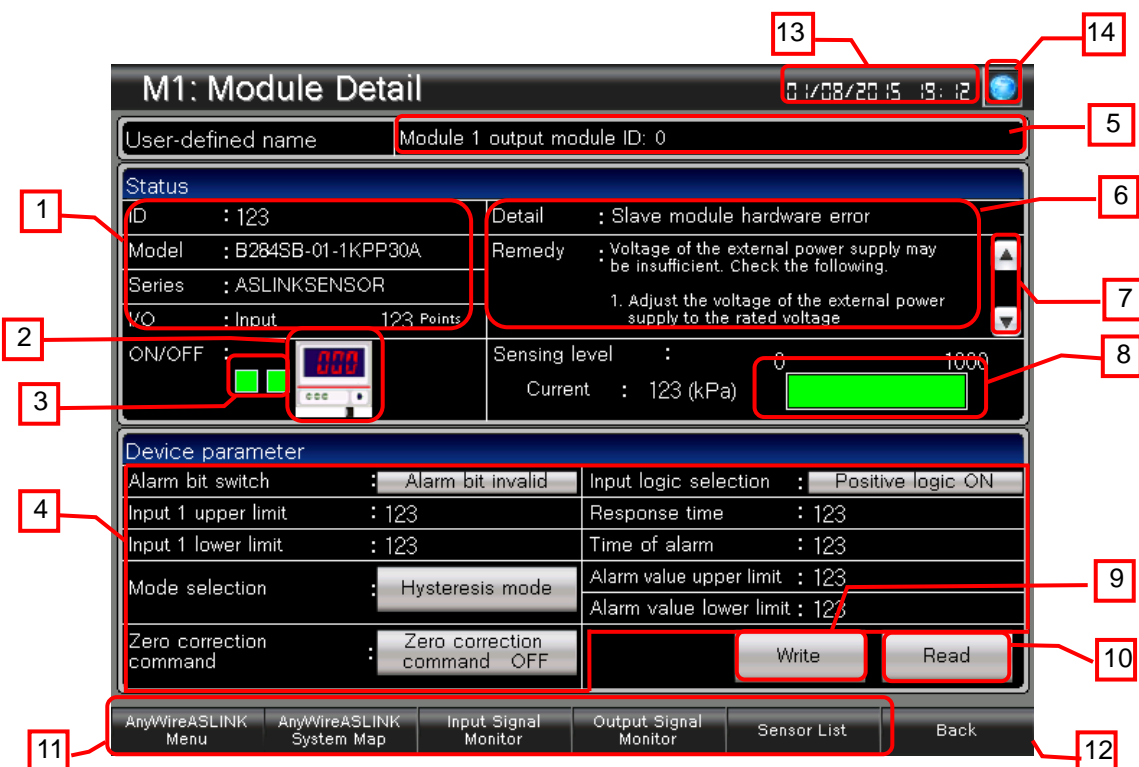
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays the user-defined name.
5. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
6. Scrolls the remedy display.
7. This switch does not work because there are no parameters to write.
8. This switch does not work because there are no parameters to read.
9. Switches to each screen.
10. Switches to the previously opened screen.
11. Displays the current date and time. Touch the area to open the [Clock Setting] window.
12. Opens the [Language Setting] window.

#### Remarks

- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.43 Detail(+Pressure 2 Pts Alarm) (B-30053)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (positive pressure, 2 points, alarm detected).

#### Description

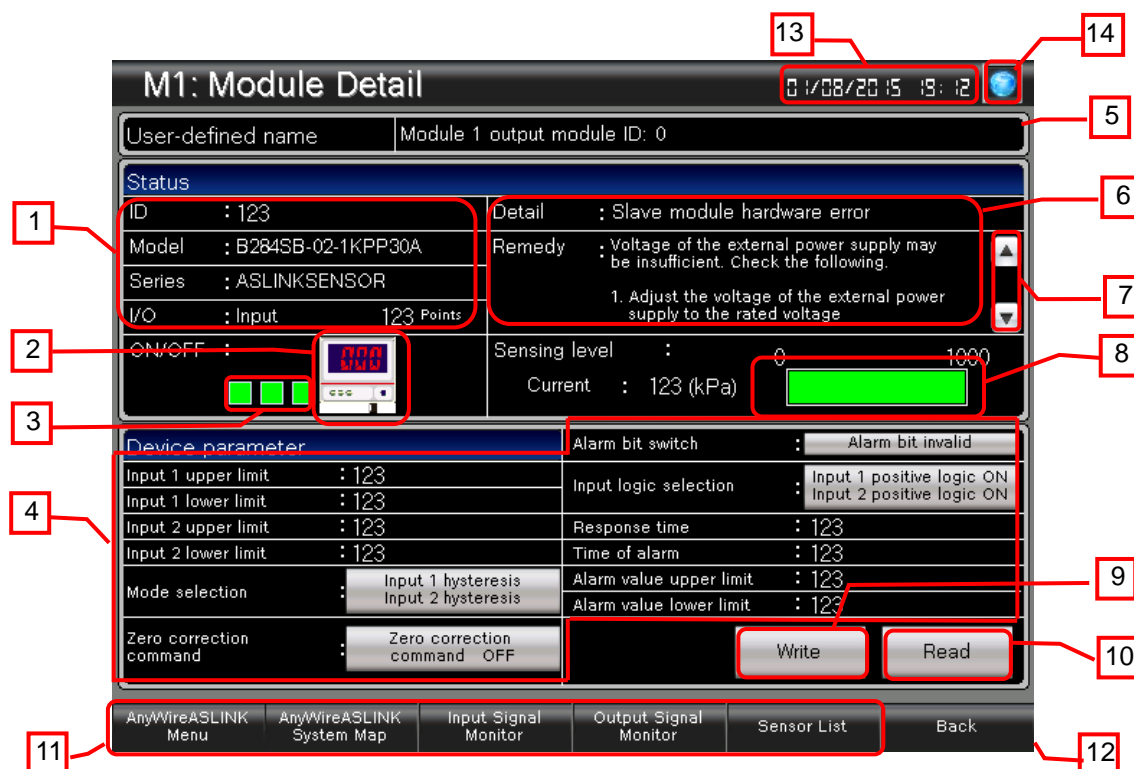
1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.44 Detail(+Pressure 3 Pts Alarm) (B-30054)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (positive pressure, 3 points, alarm detected).

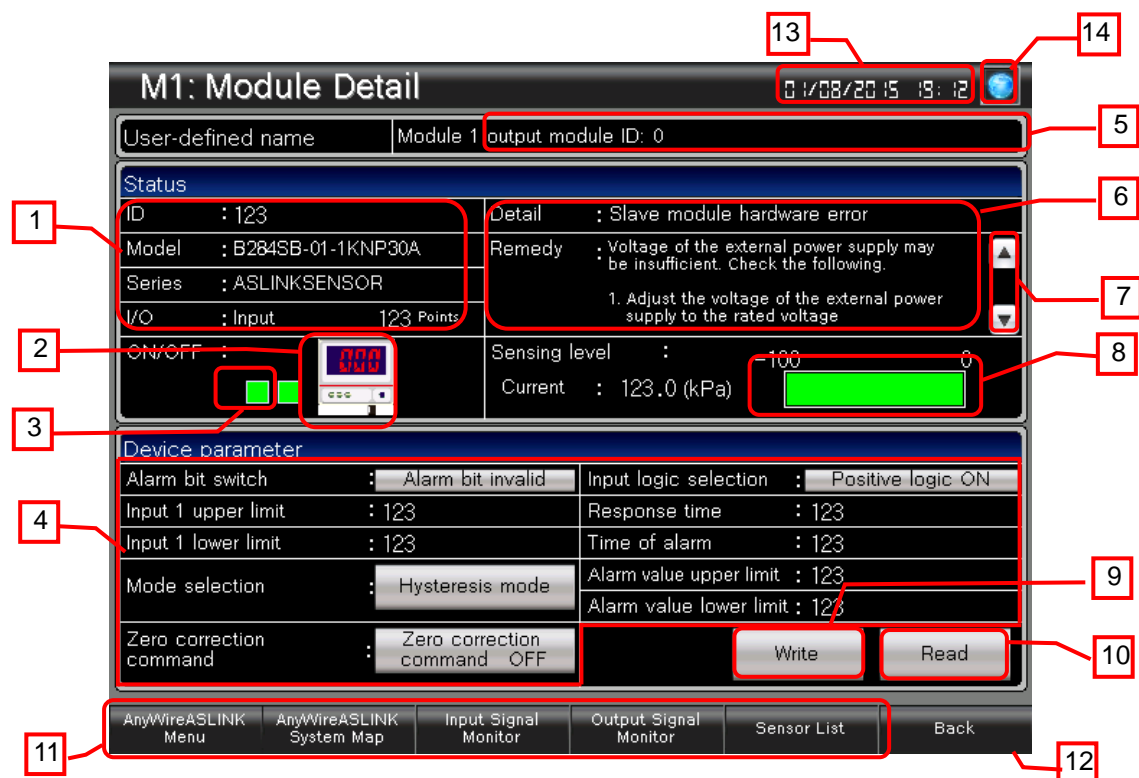
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.45 Detail(-Pressure 2 Pts Alarm) (B-30057)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (negative pressure, 2 points, alarm detected).

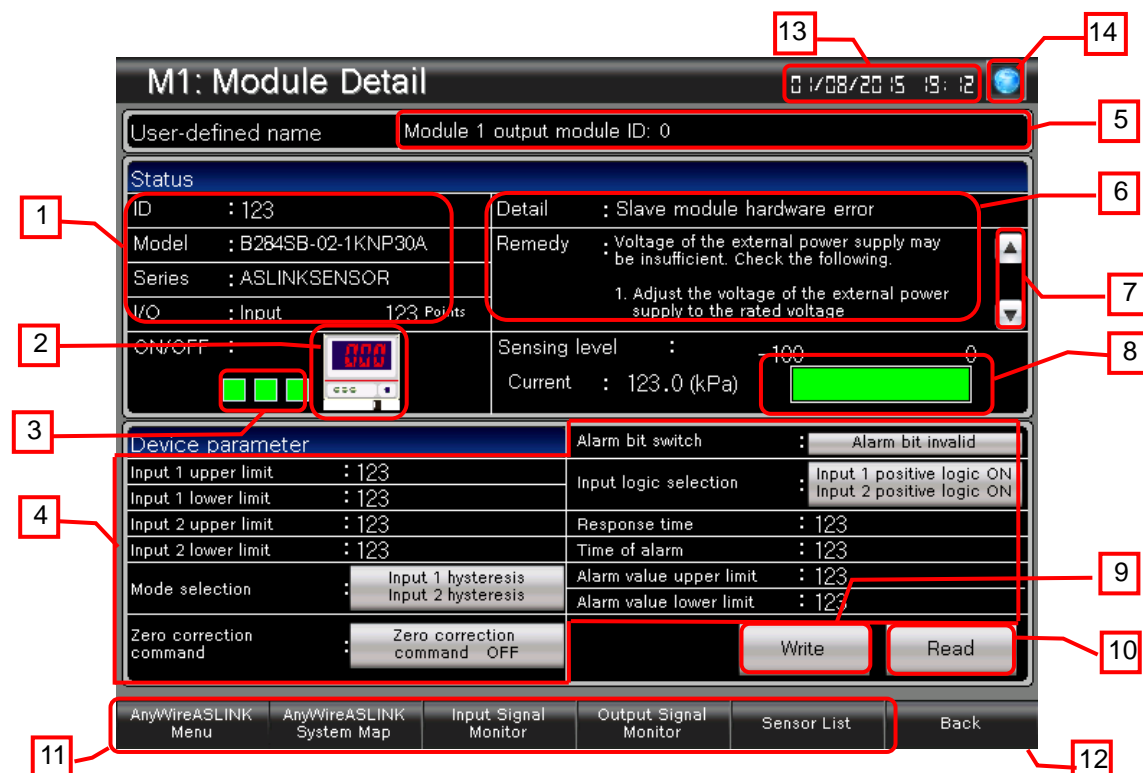
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.46 Detail(-Pressure 3 Pts Alarm) (B-30058)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (negative pressure, 3 points, alarm detected).

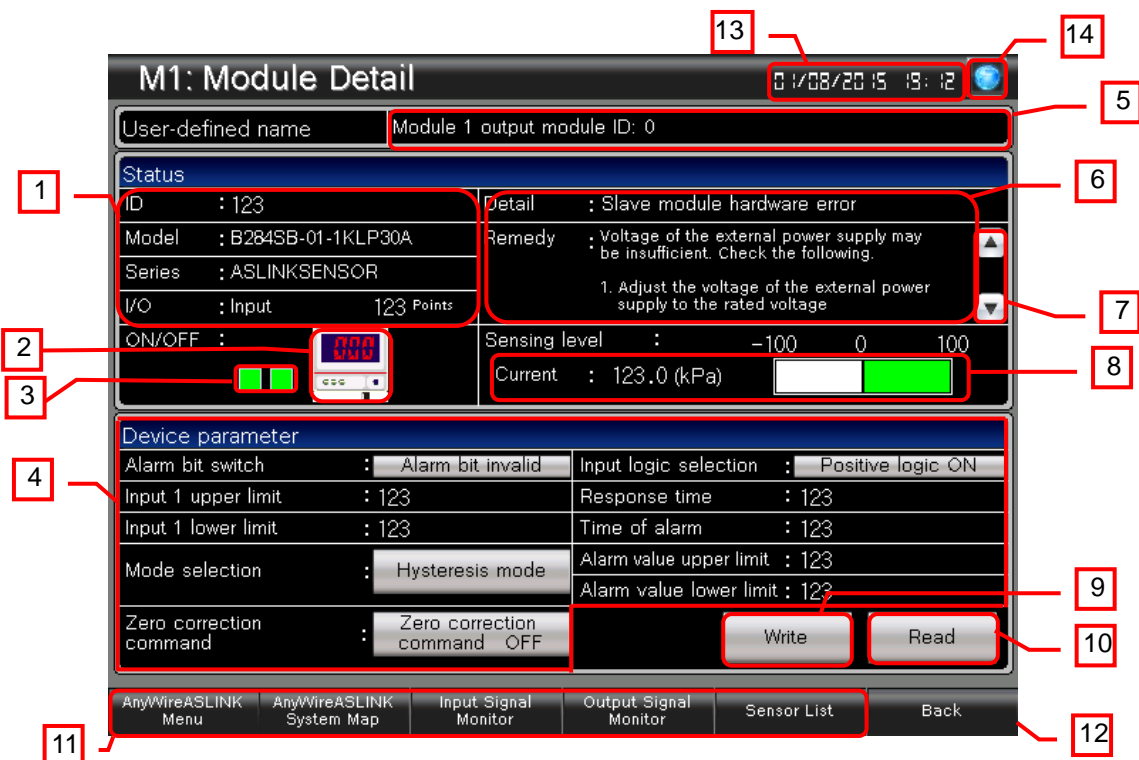
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.47 Detail(Compound 2 Pts Alarm) (B-30061)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (compound pressure, 2 points, alarm detected).

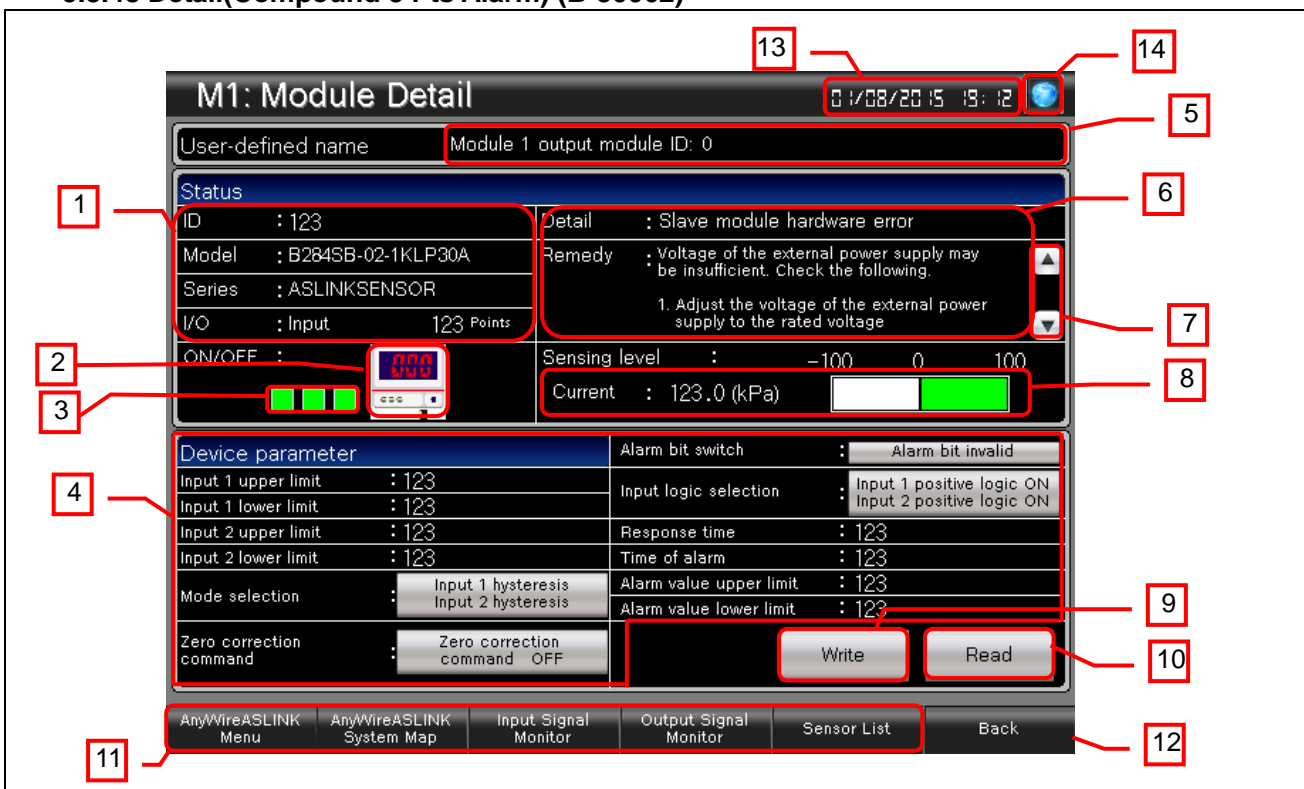
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.48 Detail(Compound 3 Pts Alarm) (B-30062)



#### Outline

This screen displays and sets the detail information about the slave module of the pressure sensor (compound pressure, 3 points, alarm detected).

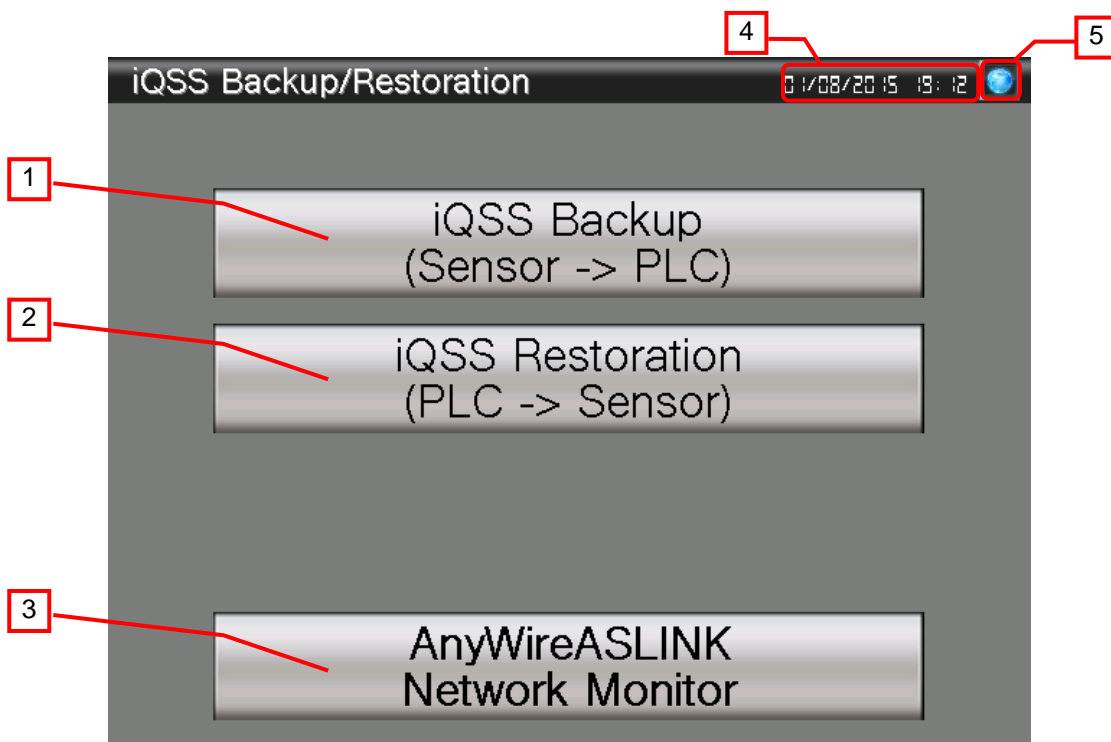
#### Description

1. Displays ID, model, series, I/O type, and the number of I/O points of the slave module.
2. Displays the image of the slave module.
3. Displays the I/O status.
4. Displays and changes parameters of the slave module.
5. Displays the user-defined name.
6. Displays the occurring alarm, or the error code and remedy. If multiple alarms or errors occur, the latest error contents will appear.
7. Scrolls the remedy display.
8. Displays the current value of the sensing level with a numerical display and a level.
9. Writes the parameters changed in 4 to the slave module. After writing, the reading parameters processing automatically starts.
10. Reads parameters of the slave module.
11. Switches to each screen.
12. Switches to the previously opened screen.
13. Displays the current date and time. Touch the area to open the [Clock Setting] window.
14. Opens the [Language Setting] window.

#### Remarks

- If parameters are changed while the system is running, the action of the module may change. Be sure to confirm safety before execution.
- The user-defined name displays the information (installation location of the slave module, etc.) that the users want to display arbitrarily. To display the information, register contents to the GOT comments. For more details, please refer to "6.1 User-Defined Name Registration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.49 iQSS Menu (B-30100)



#### Outline

This is the [iQSS Backup/Restoration] menu screen of the iQSS backup/restoration (PLC↔Sensor) function.

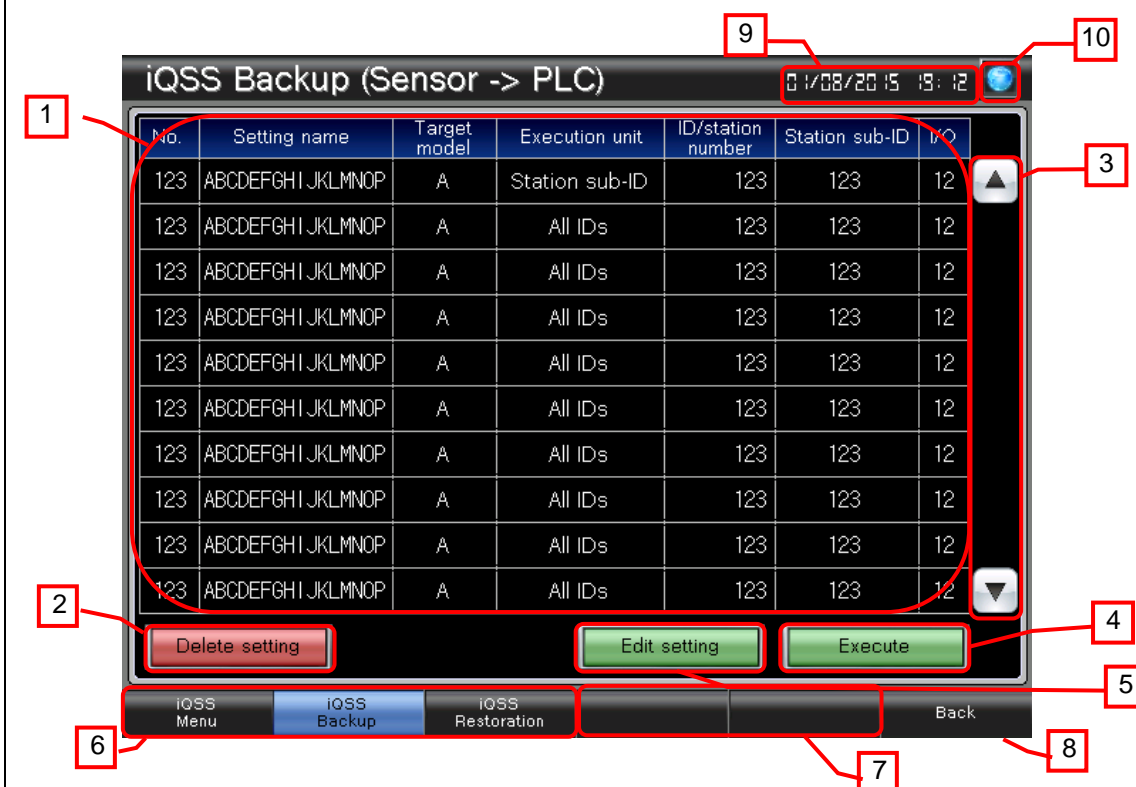
#### Description

1. Switches to the [iQSS Backup] screen.
2. Switches to the [iQSS Restoration] screen.
3. Switches to the [AnyWireASLINK Network Monitor] menu screen (the menu screen of the AnyWireASLINK network monitor function).
4. Displays the current date and time. Touch the area to open the [Clock Setting] window.
5. Opens the [Language Setting] window.

#### Remarks

- The iQSS backup setting and the iQSS backup history are read from the recipe file. For more details about the recipe function, please refer to "5.8 Recipe List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.50 iQSS Backup (B-30101)



#### Outline

This screen is used to execute the iQSS backup according to the parameters of the iQSS backup setting in the selected state. This screen is also used to edit and delete the iQSS backup setting.

#### Description

1. Displays the iQSS backup setting in a list format. By touching an iQSS backup setting, a cursor appears and the touched line becomes the selected state.
2. Deletes the iQSS backup setting in the selected state.
3. Scrolls the page of the list up and down. Switches the list 10 cases each time. The No. is up to 100.
4. Switches to the [iQSS Backup Progress] screen and executes the iQSS backup according to the parameters of the iQSS backup setting in the selected state simultaneously. When the switch is touched, if the iQSS backup setting in the selected state has not been registered, an error dialog appears.
5. Switches to the [iQSS Backup Setting] screen.
6. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
7. Shows unused switches for base screen switching.
8. Switches to the previously opened screen.
9. Displays the current date and time. Touch the area to open the [Clock Setting] window.
10. Opens the [Language Setting] window.

#### Remarks

- The iQSS backup history can be saved up to 100 cases. When executing the 101th case of the backup, a message appears to inform that the oldest data will be overwritten.
- The delete operation deletes the iQSS backup setting that is saved in a recipe file. For more details about the recipe function, please refer to "5.8 Recipe List".
- Object scripts are set for word lamps of the "Execution unit". For more details about scripts, please refer to "5.9 Script List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.51 iQSS Backup Setting (B-30102)

The screenshot shows the 'iQSS Backup Setting' screen. At the top, there is a status bar with a date/time '01/08/2015 18:12' and a globe icon. Below this, the screen is divided into several sections. A red box labeled '1' points to 'No. 123'. Another red box labeled '2' points to the 'Setting name' field containing 'ABCDEFGH IJKLMN O P'. A red box labeled '3' points to the 'Target model' field containing 'AnyWireASLINK'. A red box labeled '4' points to the 'Execution unit' section, which includes 'Station number' (set to 'All IDs') and 'Module type' (set to 'Output'). A red box labeled '5' points to the 'Folder number' field set to 'Automatic'. A red box labeled '6' points to the 'I/O' field set to '12'. A red box labeled '7' points to the 'Error time action setting' section, which has 'Continue' and 'Stop' options. A red box labeled '8' points to the 'Register' button. A red box labeled '9' points to the 'Back' button. A red box labeled '10' points to the date/time display, and a red box labeled '11' points to the globe icon.

#### Outline

This screen registers the iQSS backup setting.

#### Description

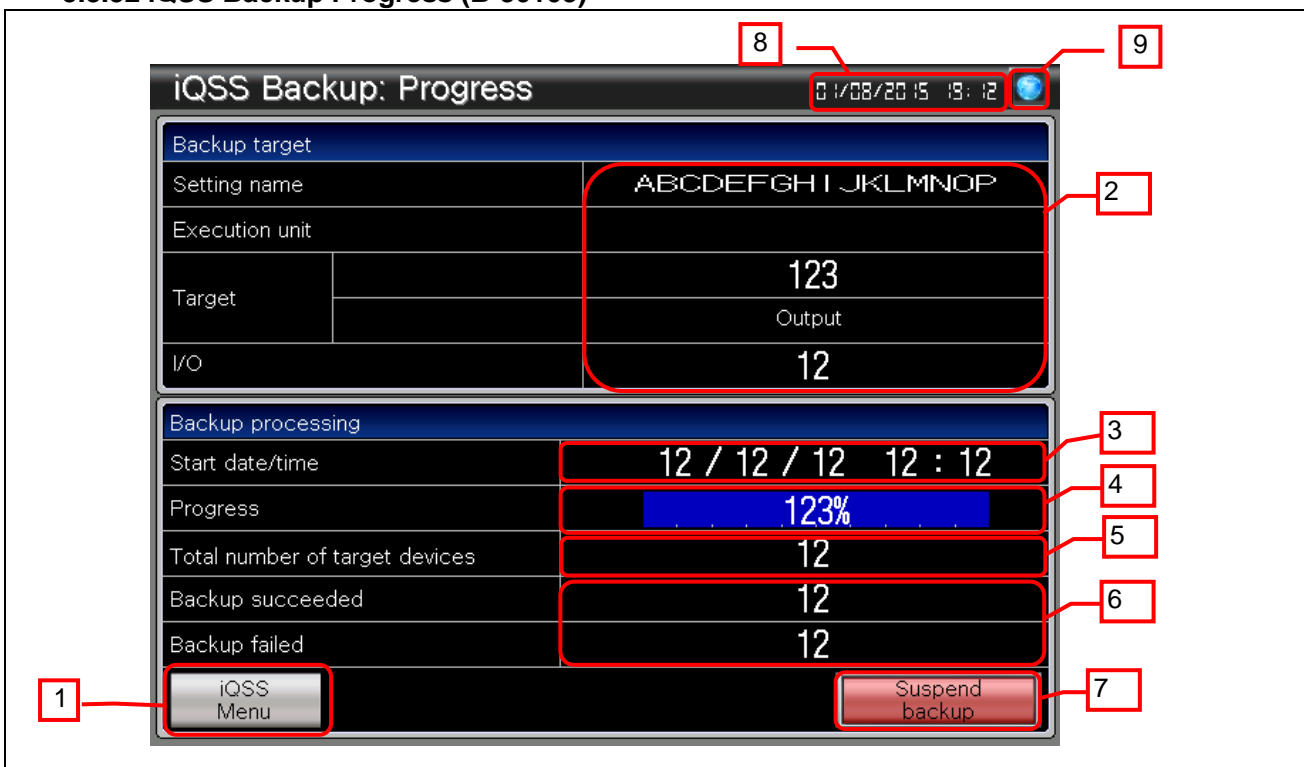
1. Displays the registration number.
2. Enters the name of the iQSS backup setting.
3. Displays the type of the network module connected to the iQSS backup target device.
4. Specifies the iQSS backup target.
  - All IDs: All iQSS compatible sensors
  - ID: iQSS compatible sensor of the specified ID
  - Output, Input/combined: Module type of the specified iQSS compatible sensor
5. The setting is made to automatically assign a folder number, which is the name of the destination folder to store the iQSS backup data. The number from 0 to 99 is automatically assigned to the folder number.
6. Enters the number that is calculated by dividing the I/O No. of the network module connected to the iQSS backup target device by 16.
7. Specifies the action at the occurrence of an error when executing the iQSS backup.
  - Continue: Continues the processing even if the backup of some devices fails while executing the backup of multiple iQSS compatible sensors.
  - Stop: Stops the processing if the backup of some devices fails while executing the backup of multiple iQSS compatible sensors.
8. Switches to the previously opened screen.
9. Saves the setting contents. The completion dialog is displayed when saving is completed. An error dialog is displayed if the setting contents are insufficient or inappropriate.
10. Displays the current date and time. Touch the area to open the [Clock Setting] window.
11. Opens the [Language Setting] window.

#### Remarks

- Screen scripts and the recipe function are used to register the iQSS backup setting. For more details about scripts, please refer to "5.9 Script List", and for the recipe function, please refer to "5.8 Recipe List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".



### 5.3.52 iQSS Backup Progress (B-30103)



#### Outline

This screen displays the progress and results of the iQSS backup.

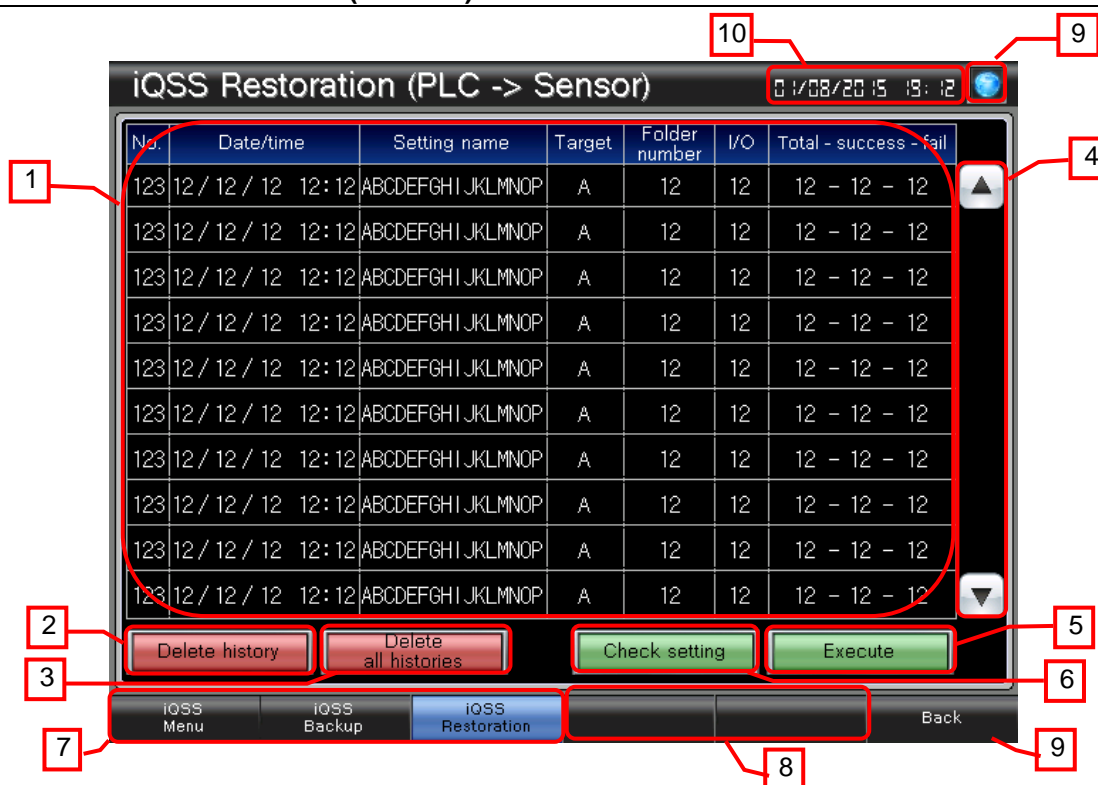
#### Description

1. Switches to the [iQSS Menu] screen. The switch is hidden while executing the iQSS backup and is displayed after the iQSS backup is completed.
2. Displays the iQSS backup setting.
3. Displays the date and time that the iQSS backup was started.
4. Displays the progress of the backup for a single iQSS backup target device with "%" and a bar graph. When the backup for a single iQSS backup target device is completed, "100%" is displayed and the color of the bar graph is changed from black to blue. If there are multiple target devices of the iQSS backup, the progress is displayed for each device.
5. Displays the total number of the iQSS backup target devices.
6. Displays the number of devices that the iQSS backup succeeded or failed.
7. Cancels the iQSS backup. The switch is displayed while executing the iQSS backup and is hidden after the iQSS backup is completed.
8. Displays the current date and time. Touch the area to open the [Clock Setting] window.
9. Opens the [Language Setting] window.

#### Remarks

- The iQSS backup will not be canceled until the [OK] switch is touched in the confirmation window.
- If the iQSS backup fails, there is a case that the total number of target devices, backup succeeded, and backup failed all become zero.
- Screen scripts and the recipe function are used to save the iQSS backup history. For more details about scripts, please refer to "5.9 Script List", and for the recipe function, please refer to "5.8 Recipe List".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.53 iQSS Restoration (B-30104)



#### Outline

This screen is used to execute the iQSS restoration according to the parameters of the iQSS restoration setting that corresponds with the iQSS backup history. This screen is also used to execute confirmation of the iQSS restoration setting that corresponds with the iQSS backup history and deletion of the iQSS backup history.

#### Description

1. Displays the iQSS backup history in a list format. By touching a iQSS backup history, a cursor appears and the touched line becomes the selected state. The iQSS backup history is displayed most recent first.
2. Deletes the iQSS backup history in the selected state. When one case of the backup history is deleted, the cases older than the deleted history will be moved up line by line.
3. Deletes all iQSS backup history.
4. Scrolls the page of the list up and down. Switches the list 10 cases each time. The No. is up to 200.
5. Switches to the [iQSS Restoration Progress] screen and executes the iQSS restoration according to the parameters of the restoration setting that corresponds with the iQSS backup history in the selected state simultaneously. In addition, if the iQSS restoration is executed while selecting the position where no iQSS backup history is displayed, an error dialog appears.
6. Switches to the [iQSS Restoration Setting] screen.
7. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
8. Shows unused switches for base screen switching.
9. Switches to the previously opened screen.
10. Displays the current date and time. Touch the area to open the [Clock Setting] window.
11. Opens the [Language Setting] window.

#### Remarks

- When executing the iQSS backup, the iQSS backup history is created only when the iQSS backup data is created in the SD card that is in the PLC.
- The restoration cannot be executed using the iQSS restoration setting that corresponds with the iQSS backup history which ended abnormally or was suspended.
- The delete operation deletes the iQSS backup history that is saved in a SD card in the GOT. Please note that the actual iQSS backup data is in the SD card in the PLC CPU and it cannot be deleted from the GOT according to the specification. Please also note that if the iQSS backup data is deleted from the SD card, the restoration cannot be executed from the corresponding iQSS backup history.
- For the folder configuration of the SD card in the PLC CPU, please refer to "6.5 iQSS Backup Folder Configuration".
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.54 iQSS Restoration Setting (B-30105)

The screenshot shows the 'iQSS Restoration Setting' screen. It features a title bar at the top with a date and time display (01/08/2015 19:12) and a globe icon. Below the title bar, there is a 'No. 123' field. The main area contains several rows of settings: 'Setting name' (ABCDEFGHIJKLMNOP), 'Target model' (a green box), 'Execution unit' (All stations, ID), 'Module type' (Output), 'Folder number' (12), 'I/O' (12), and 'Error time action setting' (Continue, Stop). A 'Back' button is located at the bottom left. Numbered callouts (1-10) point to specific elements: 1 points to 'No. 123', 2 points to the 'Back' button, 3 points to 'Setting name', 4 points to 'Target model', 5 points to the 'Execution unit' and 'Module type' buttons, 6 points to 'Folder number', 7 points to 'I/O', 8 points to the 'Error time action setting' buttons, 9 points to the date and time display, and 10 points to the globe icon.

#### Outline

This screen is used to confirm the iQSS restoration setting. The contents of the iQSS restoration setting are the same as those when the iQSS backup was executed excluding the folder number.

#### Description

1. Displays the iQSS backup history No.
2. Switches to the previously opened screen.
3. Displays the name of the iQSS backup setting.
4. Displays the target model when the iQSS backup was executed.
5. Displays the restoration target (the iQSS backup target when the iQSS backup was executed).
6. Displays the number of the folder that the iQSS backup data was stored when executing the iQSS backup.
7. Displays the I/O No. of the network module to which the iQSS backup target device was connected when executing the iQSS backup.
8. Displays the action at the occurrence of an error when executing the iQSS restoration. The setting is the same as those when the iQSS backup was executed.
9. Displays the current date and time. Touch the area to open the [Clock Setting] window.
10. Opens the [Language Setting] window.

#### Remarks

- The display is the same as that of the [iQSS Backup Setting] screen.
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.55 iQSS Restoration Progress (B-30106)

The screenshot shows the 'iQSS Restoration: Progress' screen. It features a title bar with a cancel button (7) and a clock/language button (8). The main content is divided into two sections: 'Restoration target' and 'Restoration processing'. The 'Restoration target' section contains fields for 'Setting name' (ABCDEF GHI JKLMNOP), 'Execution unit', 'Target' (123), 'Output', and 'I/O' (12). The 'Restoration processing' section shows 'Start date/time' (12 / 12 / 12 12 : 12), 'Progress' (123% with a blue bar), 'Total number of target devices' (12), 'Restoration succeeded' (12), and 'Restoration failed' (12). At the bottom, there is an 'iQSS Menu' button (1) and a 'Suspend restoration' button (6).

Restoration target	
Setting name	ABCDEF GHI JKLMNOP
Execution unit	
Target	123
	Output
I/O	12

Restoration processing	
Start date/time	12 / 12 / 12 12 : 12
Progress	123%
Total number of target devices	12
Restoration succeeded	12
Restoration failed	12

Buttons: iQSS Menu (1), Suspend restoration (6)

#### Outline

This screen displays the progress of the iQSS restoration.

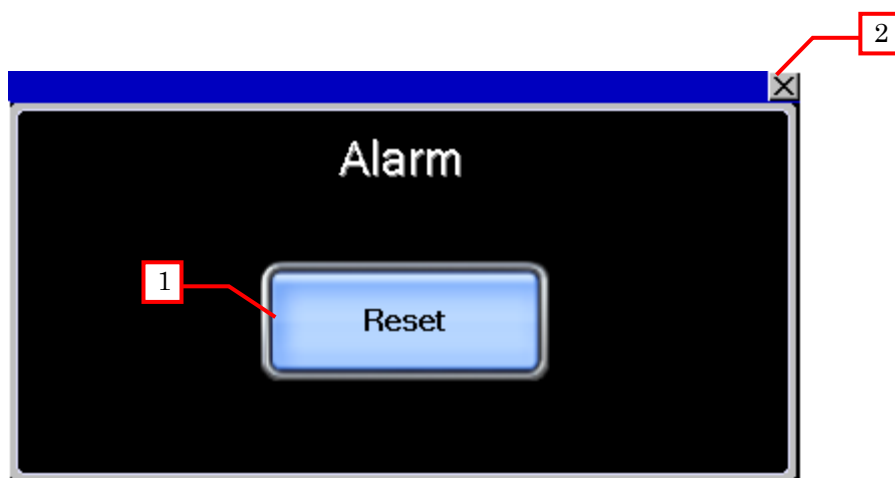
#### Description

1. Switches to the [iQSS Menu] screen. The switch is hidden while executing the iQSS restoration and is displayed after the iQSS restoration is completed.
2. Displays the iQSS restoration setting. The contents of the iQSS restoration setting are the same as those when the iQSS backup was executed.
3. Displays the date and time that the iQSS restoration was started.
4. Displays the progress of the restoration for a single iQSS restoration target device with "%" and a bar graph. When the restoration for a single iQSS restoration target device is completed, "100%" is displayed and the color of the bar graph is changed from black to blue. If there are multiple target devices of the iQSS restoration, the progress is displayed for each device.
5. Displays the total number of the iQSS restoration target devices.
6. Displays the number of devices that the iQSS restoration succeeded or failed.
7. Cancels the iQSS restoration. The switch is displayed while executing the iQSS restoration and is hidden after the iQSS restoration is completed.
8. Displays the current date and time. Touch the area to open the [Clock Setting] window.
9. Opens the [Language Setting] window.

#### Remarks

- The iQSS restoration will not be canceled until the [OK] switch is touched in the confirmation window.
- If an error occurs in the iQSS restoration, there is a case that the total number of target devices, restoration succeeded, and restoration failed all become zero.
- The number of alarms and errors are monitored every second with the project script. If the difference arises in the number of cases, the latest parameters will be read with the device data transfer function. For more details about scripts, please refer to "5.9 Script List", and for the device data transfer function, please refer to "5.7 Device Data Transfer List".

### 5.3.56 Alarm Reset (W-30001)



#### Outline

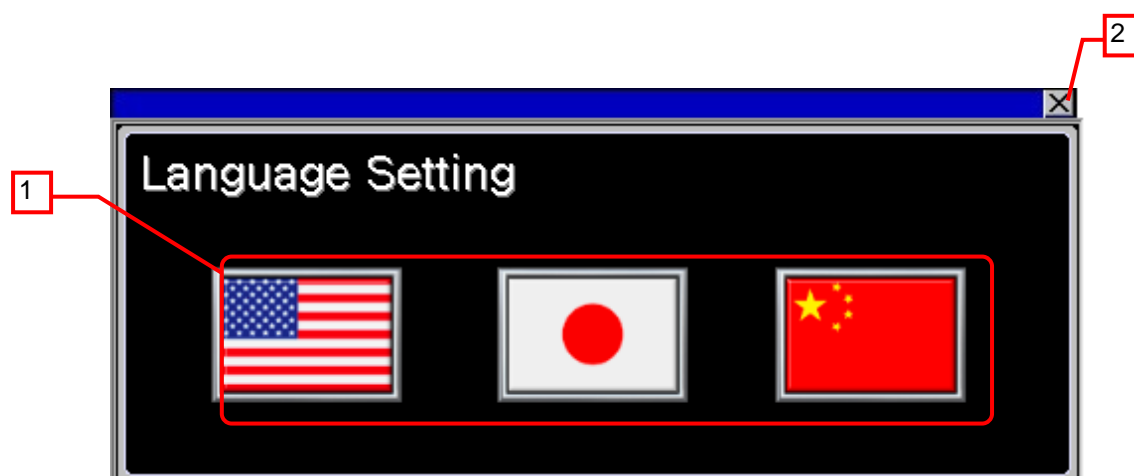
This window screen allows resetting the system alarm.

#### Description

1. Resets the system alarm, and closes the window screen after 1 second.
2. Closes the window screen.

#### Remarks

### 5.3.57 Language Setting (W-30002)



#### Outline

This window screen allows selecting the GOT language.

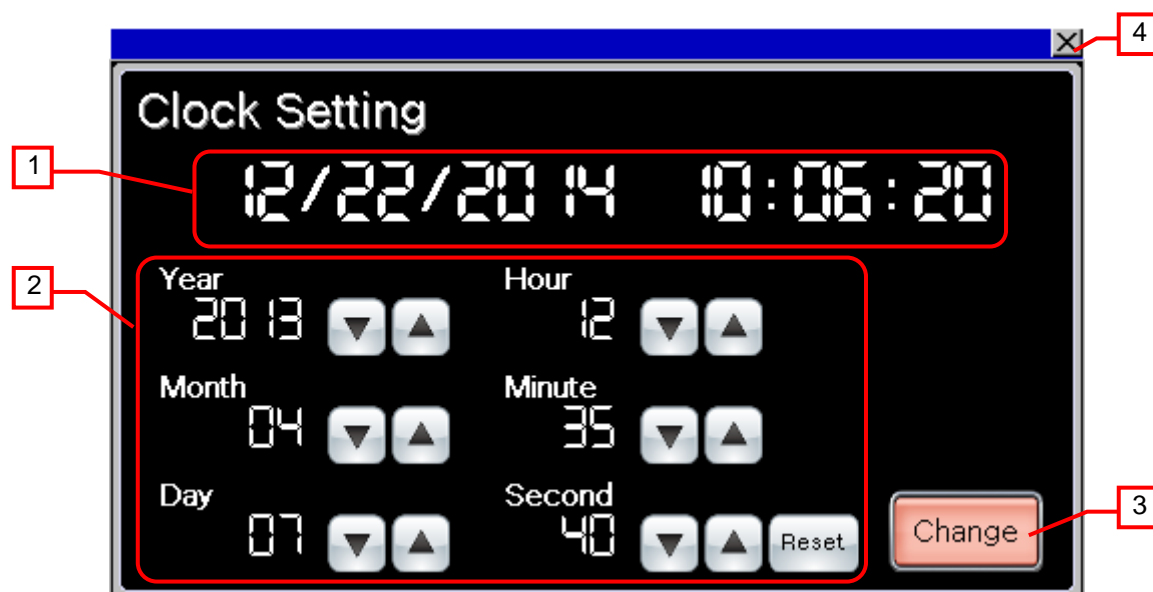
#### Description

1. Switches the language and closes the window screen.
2. Closes the window screen.

#### Remarks

- The system language is also switched according to the display language.



### 5.3.58 Clock Setting (W-30003)



#### Outline

This window screen allows changing the GOT clock data.

#### Description

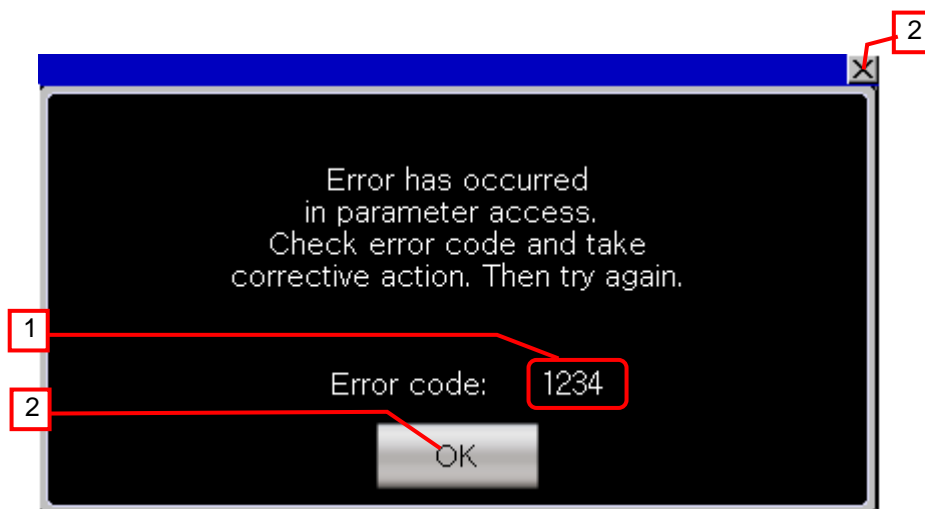
1. Displays the current date and time.
2. Use   switches to change the date and time. Hold down the switches to increment or decrement the value continuously. The [Reset] switch resets the seconds.
3. Applies the set date and time to the GOT clock data, and closes the window screen after 1 second.
4. Closes the window screen.

#### Remarks

- The date and time at window opening are initially set as the clock data to be newly set.
- Object scripts are set for the numerical display of the year, month, date, hour, minute and second in the clock data to be newly set. For more details about scripts, please refer to "5.9 Script List".



### 5.3.59 Parameter Access Error (W-30004)



#### Outline

This window screen is displayed if an error occurred when accessing parameters.

#### Description

1. Displays the occurring alarm, or the error code. If multiple alarms or errors occur, the latest error code will appear.
2. Closes the window screen.

#### Remarks

### 5.3.60 Accessing Parameters (W-30005)



#### Outline

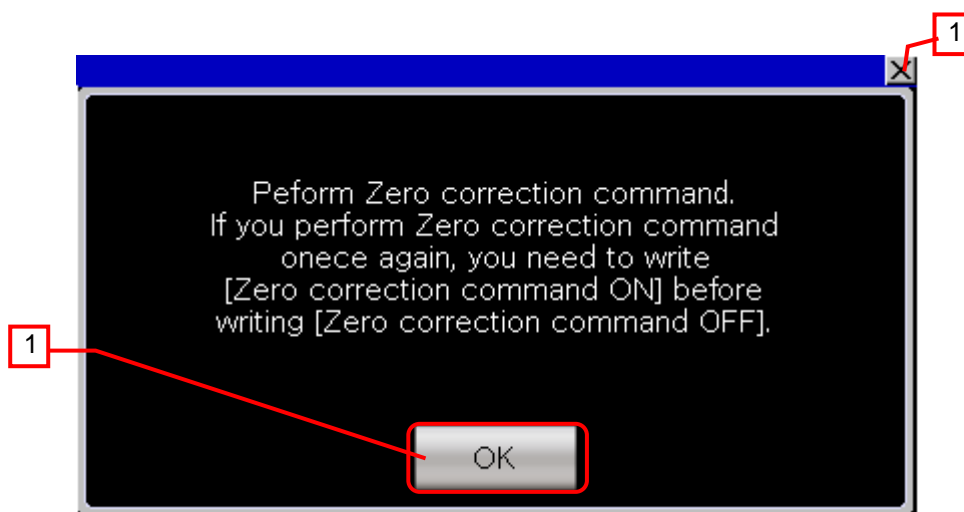
This window screen is displayed when accessing parameters.

#### Description

1. Closes the window screen.

#### Remarks

### 5.3.61 Zero Correction ExecutionMessage (W-30006)



#### Outline

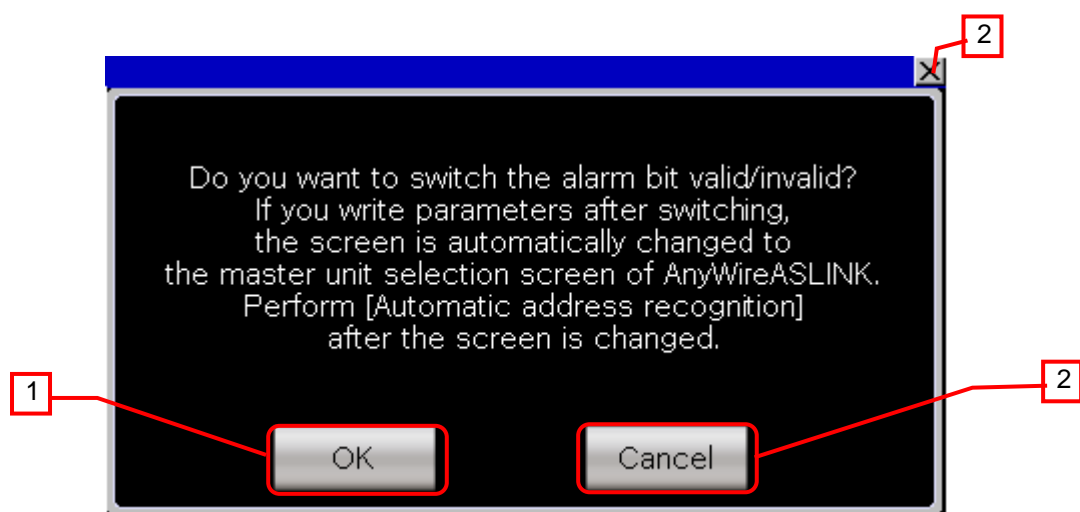
This window screen is displayed when the zero correction command changed from OFF to ON.

#### Description

1. Closes the window screen.

#### Remarks

### 5.3.62 Alarm Bit Switching Message (W-30007)



#### Outline

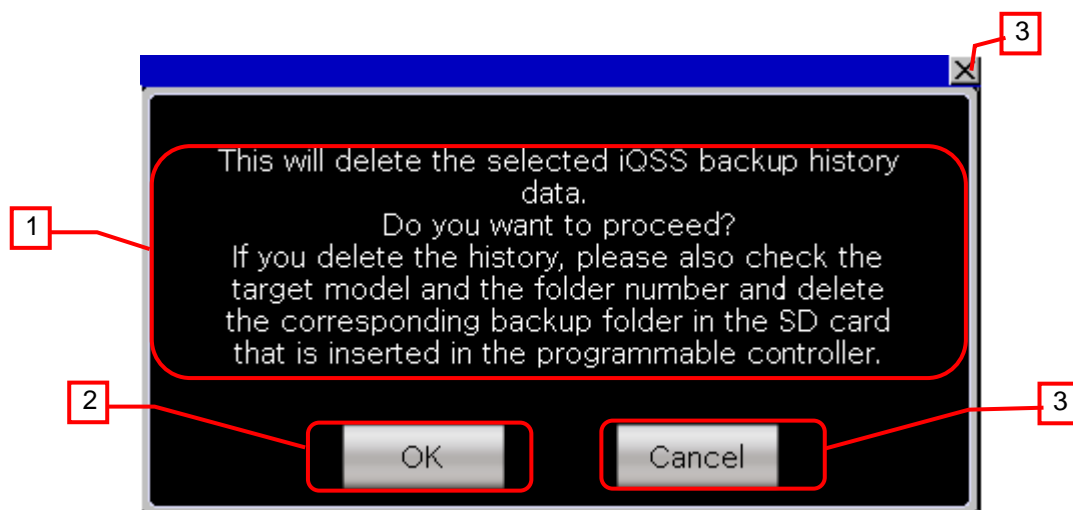
This window screen can be used to switch the alarm bit valid/invalid.

#### Description

1. Switches the alarm bit valid/invalid and closes the window screen.
2. Closes the window screen.

#### Remarks

### 5.3.63 Data Deletion Check Dialog (W-30100)



#### Outline

This screen is used to confirm deletion of the iQSS backup setting or the iQSS backup history.

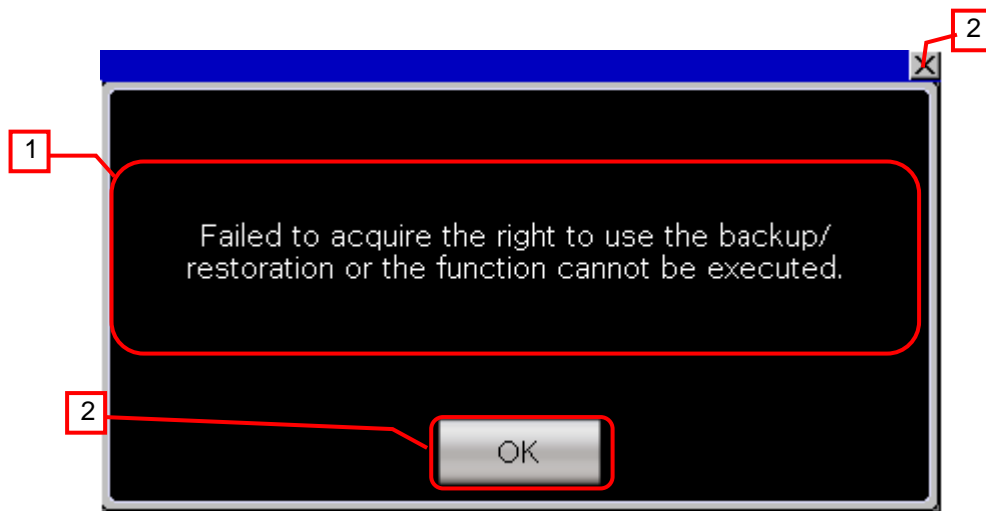
#### Description

1. Displays messages.
2. Deletes the target data and closes the window screen.
3. Closes the window screen.

#### Remarks

- The message to be displayed differs depending on from which switch the window screen was opened.

### 5.3.64 Notification Dialog (W-30101)



#### Outline

This window screen notifies the processing results.

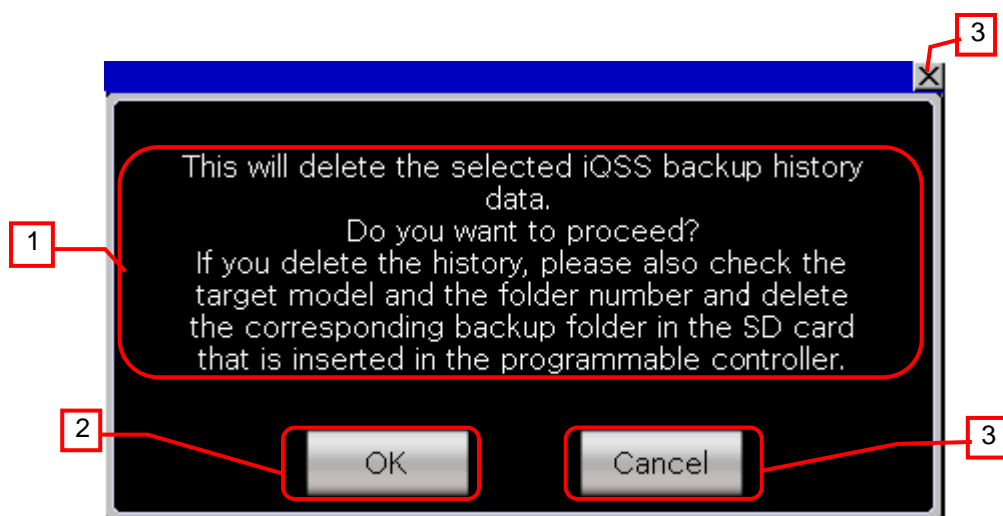
#### Description

1. Displays messages.
2. Closes the window screen.

#### Remarks

- The contents of the message differ depending on the script that called the window screen. For more details about scripts, please refer to "5.9 Script List".

### 5.3.65 Execution Check Dialog (W-30102)



#### Outline

This window screen appears to confirm whether to execute the action.

#### Description

1. Displays messages.
2. Executes various actions and closes the window screen.
3. Closes the window screen.

#### Remarks

- The contents of the message and actions differ depending on the base screen to be displayed.

iQSS Backup Screen:	iQSS backup execution
iQSS Backup Progress Screen:	iQSS backup cancel
iQSS Restoration Screen:	iQSS restoration execution
iQSS Restoration Progress Screen:	iQSS restoration cancel

## 5.4 Slave Module Detail Information Screen Correspondence Table

Slave module model No.	Model	Large classification	Base screen No.
100	B281SB-02U-CC20	ASLINKER	B-30006
101	B281SB-02US-CC20	ASLINKER	B-30006
102	B280SB-02U-C1220	ASLINKER	B-30008
103	B280SB-02US-C1220	ASLINKER	B-30008
104	BL287SB-02F-CC20	ASLINKER	B-30007
105	BL287SB-02FS-CC20	ASLINKER	B-30007
106	B298SB-02U-M12	ASLINKER	B-30009
107	B298SB-02US-M12	ASLINKER	B-30009
400	B281PB-02U-CC20	ASLINKER	B-30006
401	B281PB-02US-CC20	ASLINKER	B-30006
402	B280PB-02U-C1220	ASLINKER	B-30008
403	B280PB-02US-C1220	ASLINKER	B-30008
404	BL287PB-02F-CC20	ASLINKER	B-30007
405	BL287PB-02FS-CC20	ASLINKER	B-30007
406	B298PB-02U-M12	ASLINKER	B-30009
407	B298PB-02US-M12	ASLINKER	B-30009
700	B281XB-02U-CC20	ASLINKER	B-30006
701	B281XB-02US-CC20	ASLINKER	B-30006
702	B280XB-02U-C1220	ASLINKER	B-30008
703	B280XB-02US-C1220	ASLINKER	B-30008
704	BL287XB-02F-CC20	ASLINKER	B-30007
705	BL287XB-02FS-CC20	ASLINKER	B-30007
706	B298XB-02U-M12	ASLINKER	B-30009
707	B298XB-02US-M12	ASLINKER	B-30009
1000	B289SB-01AP-CAM20	ASLINKAMP	B-30010
1050	B289SB-01AP-CAS	ASLINKAMP	B-30010
1200	B289SB-01AK-CAM20	ASLINKAMP	B-30011
1250	B289SB-01AK-CAS	ASLINKAMP	B-30011
1400	B289SB-01AF-CAM20	ASLINKAMP	B-30012
1401	B289SB-01AF-CAM20P	ASLINKAMP	B-30019
1450	B289SB-01AF-CAS	ASLINKAMP	B-30012
1451	B289SB-01AF-CASP	ASLINKAMP	B-30019
1500	B289SB-01V1-CPM20	ASLINKAMP	B-30043
1503	B289SB-J1V1-CPM20	ASLINKAMP	B-30046
1553	B289SB-J1V1-CPS	ASLINKAMP	B-30050
2000	B285SB-01-1K1	ASLINKSENSOR	B-30013
2300	B283SB-01-1KR	ASLINKSENSOR	B-30014
2301	B283SB-01-1KS	ASLINKSENSOR	B-30014
2302	B283SB-01-1KC	ASLINKSENSOR	B-30015
2303	B283SB-01-1KP	ASLINKSENSOR	B-30016
2600	B295SB-01-1K26	ASLINKSENSOR	B-30017
2601	B295SB-01-1K25	ASLINKSENSOR	B-30017
2602	B295SB-01-1K27	ASLINKSENSOR	B-30017
2603	B295SB-01-1K24	ASLINKSENSOR	B-30017
2900	B284SB-01-12	ASLINKSENSOR	B-30020
2903	B284SB-02-PP30	ASLINKSENSOR	B-30021
2909	B284SB-J1-PP30	ASLINKSENSOR	B-30023
2930	B284SB-01-NP30	ASLINKSENSOR	B-30024
2933	B284SB-02-NP30	ASLINKSENSOR	B-30025
2939	B284SB-J1-NP30	ASLINKSENSOR	B-30027
2960	B284SB-01-LP30	ASLINKSENSOR	B-30028
2963	B284SB-02-LP30	ASLINKSENSOR	B-30029
2969	B284SB-J1-LP30	ASLINKSENSOR	B-30031
3100	B297SB-01-1K40	ASLINKSENSOR	B-30032
3300	B232SB-MX100-STP	ASLINKSENSOR	B-30033
3301	B232SB-SX100-STP	ASLINKSENSOR	B-30034
3302	B233SB-02-M-C	ASLINKSENSOR	B-30035



Slave module model No.	Model	Large classification	Base screen No.
3303	B233SB-02-S-C	ASLINKSENSOR	B-30035
3304	B233SB-02-M-P	ASLINKSENSOR	B-30036
3305	B233SB-02-S-P	ASLINKSENSOR	B-30036
10000	BL296SB-08F-20	ASLINKTERMINAL	B-30018
10001	BL296SB-08FS-20	ASLINKTERMINAL	B-30018
10002	BL296SB-08F-4-20	ASLINKTERMINAL	B-30037
10003	BL296SB-08FS-4-20	ASLINKTERMINAL	B-30037
10004	BL296SB-08F-9-20	ASLINKTERMINAL	B-30038
10005	BL296SB-08FS-9-20	ASLINKTERMINAL	B-30038
10006	BL296SB-08F-10-20	ASLINKTERMINAL	B-30039
10007	BL296SB-08FS-10-20	ASLINKTERMINAL	B-30039
10010	B262-HR16-IN	ASLINKTERMINAL	B-30051
10300	BL296PB-08F-20	ASLINKTERMINAL	B-30018
10301	BL296PB-08FS-20	ASLINKTERMINAL	B-30018
10302	BL296PB-08F-4-20	ASLINKTERMINAL	B-30037
10303	BL296PB-08FS-4-20	ASLINKTERMINAL	B-30037
10304	BL296PB-08F-9-20	ASLINKTERMINAL	B-30038
10305	BL296PB-08FS-9-20	ASLINKTERMINAL	B-30038
10306	BL296PB-08F-10-20	ASLINKTERMINAL	B-30039
10307	BL296PB-08FS-10-20	ASLINKTERMINAL	B-30039
10308	BL264PB-16F-T5	ASLINKTERMINAL	B-30040
10309	BL264PB-16FS-T5	ASLINKTERMINAL	B-30040
10310	B262-HR16-OUT	ASLINKTERMINAL	B-30052
10320	BL296PB-08RSN	ASLINKTERMINAL	B-30041
10321	BL296PB-08RS	ASLINKTERMINAL	B-30042
10322	BL296PB-08RSS	ASLINKTERMINAL	B-30042
10323	BL296PB-08RSS1	ASLINKTERMINAL	B-30042
10324	BL296PB-08RSS2	ASLINKTERMINAL	B-30042
10325	BL296PB-08RSS3	ASLINKTERMINAL	B-30042
10326	BL296PB-08RSS4	ASLINKTERMINAL	B-30042
10600	BL296XB-08F-20	ASLINKTERMINAL	B-30018
10601	BL296XB-08FS-20	ASLINKTERMINAL	B-30018
10602	BL296XB-08F-4-20	ASLINKTERMINAL	B-30037
10603	BL296XB-08FS-4-20	ASLINKTERMINAL	B-30037
10604	BL296XB-08F-9-20	ASLINKTERMINAL	B-30038
10605	BL296XB-08FS-9-20	ASLINKTERMINAL	B-30038
10606	BL296XB-08F-10-20	ASLINKTERMINAL	B-30039
10607	BL296XB-08FS-10-20	ASLINKTERMINAL	B-30039

## 5.5 Device List

Some of the devices specified to the on-screen switches and lamps, etc., are also used for common settings of functions such as scripts. Using [Batch Edit] is recommended to change these devices in a batch. For more details about using [Batch Edit], please refer to the "GT Designer3 (GOT2000) Help".

### 5.5.1 Devices of the controller

Type	Device No.	Application
Bit	X0011, X0031, X0051, X0071	Parameter Access Completion Flag
	X0012, X0032, X0052, X0072	Parameter Access Error
	Y0000, Y0020, Y0040, Y0060	Error Flag Clear Command
	Y0001, Y0021, Y0041, Y0061	Sensor Auto Recognition Command
	Y0010, Y0030, Y0050, Y0070	Parameter Access Request Command from Master to Slave
	Y0011, Y0031, Y0051, Y0071	Parameter Batch Read Command from Master to Slave
	SM1435	iQ Sensor Solution Compatible Backup/Restoration Execution Permission Bit
	SM1436	iQ Sensor Solution Compatible Backup Request Bit
	SM1437	iQ Sensor Solution Compatible Backup Normal Completion Bit
	SM1438	iQ Sensor Solution Compatible Backup Abnormal Completion Bit
	SM1439	iQ Sensor Solution Compatible Restoration Request Bit
	SM1440	iQ Sensor Solution Compatible Restoration Normal Completion Bit
	SM1441	iQ Sensor Solution Compatible Restoration Abnormal Completion Bit
	SM1442	iQ Sensor Solution Compatible Backup/Restoration Cancel Request Bit
Word	U00-G0 to U00-G15	Master Module No.1 Input Information Area (U00-G0 is also Used as Standard Device of Device Data Transfer)
	U00-G1	Master Module No.1 Used as Standard Device of Device Data Transfer
	U00-G20	Master Module No.1 Used as Standard Device of Device Data Transfer
	U00-G40	Master Module No.1 Used as Standard Device of Sensing Level
	U00-G4096 to U00-G4111	Master Module No.1 Output Information Area
	U00-G8192	Master Module No.1 Number of Error IDs Information
	U00-G8193 to U00-G8320	Master Module No.1 Error ID Information Storage Area
	U00-G8960	Master Module No.1 Number of Modules Information
	U00-G9216	Master Module No.1 Number of Connection IDs Information
	U00-G9217 to U00-G9344	Master Module No.1 Connection ID Information Storage Area
	U00-G9984	Master Module No.1 Number of Alarm IDs Information
	U00-G9985 to U00-G10112	Master Module No.1 Alarm ID Information Storage Area
	U00-G10256	Master Module No.1 Latest Error Code Storage Area
	U00-G10320	Master Module No.1 Parameter Access Setting
	U00-G10321	Master Module No.1 Parameter Access Target ID Specification
	U00-G10496 to U00-G10751	Master Module No.1 Parameter Storage Destination Memory Number (Output)
	U00-G11008 to U00-G11263	Master Module No.1 Parameter Storage Destination Memory Number (Input)
	U00-G12288 to U00-G18431	Master Module No.1 Parameter Storage Area
	U02-G1	Master Module No.2 Used as Standard Device of Device Data Transfer
	U02-G20	Master Module No.2 Used as Standard Device of Device Data Transfer
	U02-G40	Master Module No.2 Used as Standard Device of Sensing Level
	U02-G4096 to U02-G4111	Master Module No.2 Output Information Area
	U02-G8192	Master Module No.2 Number of Error IDs Information

Type	Device No.	Application
Word	U02-G8193 to U02-G8320	Master Module No.2 Error ID Information Storage Area
	U02-G8960	Master Module No.2 Number of Modules Information
	U02-G9216	Master Module No.2 Number of Connection IDs Information
	U02-G9217 to U02-G9344	Master Module No.2 Connection ID Information Storage Area
	U02-G9984	Master Module No.2 Number of Alarm IDs Information
	U02-G9985 to U02-G10112	Master Module No.2 Alarm ID Information Storage Area
	U02-G10256	Master Module No.2 Latest Error Code Storage Area
	U02-G10320	Master Module No.2 Parameter Access Setting
	U02-G10321	Master Module No.2 Parameter Access Target ID Specification
	U02-G10496 to U02-G10751	Master Module No.2 Parameter Storage Destination Memory Number (Output)
	U02-G11008 to U02-G11263	Master Module No.2 Parameter Storage Destination Memory Number (Input)
	U02-G12288 to U02-G18431	Master Module No.2 Parameter Storage Area
	U04-G0	Master Module No.3 Used as Standard Device of Device Data Transfer
	U04-G1	Master Module No.3 Used as Standard Device of Device Data Transfer
	U04-G20	Master Module No.3 Used as Standard Device of Device Data Transfer
	U04-G40	Master Module No.3 Used as Standard Device of Sensing Level
	U04-G4096 to U04-G4111	Master Module No.3 Output Information Area
	U04-G8192	Master Module No.3 Number of Error IDs Information
	U04-G8193 to U04-G8320	Master Module No.3 Error ID Information Storage Area
	U04-G8960	Master Module No.3 Number of Modules Information
	U04-G9216	Master Module No.3 Number of Connection IDs Information
	U04-G9217 to U04-G9344	Master Module No.3 Connection ID Information Storage Area
	U04-G9984	Master Module No.3 Number of Alarm IDs Information
	U04-G9985 to U04-G10112	Master Module No.3 Alarm ID Information Storage Area
	U04-G10256	Master Module No.3 Latest Error Code Storage Area
	U04-G10320	Master Module No.3 Parameter Access Setting
	U04-G10321	Master Module No.3 Parameter Access Target ID Specification
	U04-G10496 to U04-G10751	Master Module No.3 Parameter Storage Destination Memory Number (Output)
	U04-G11008 to U04-G11263	Master Module No.3 Parameter Storage Destination Memory Number (Input)
	U04-G12288 to U04-G18431	Master Module No.3 Parameter Storage Area
	U06-G0	Master Module No.4 Used as Standard Device of Device Data Transfer
	U06-G1	Master Module No.4 Used as Standard Device of Device Data Transfer
	U06-G20	Master Module No.4 Used as Standard Device of Device Data Transfer
	U06-G40	Master Module No.4 Used as Standard Device of Sensing Level
	U06-G4096 to U06-G4111	Master Module No.4 Output Information Area
	U06-G8192	Master Module No.4 Number of Error IDs Information
	U06-G8193 to U06-G8320	Master Module No.4 Error ID Information Storage Area
	U06-G8960	Master Module No.4 Number of Modules Information
	U06-G9216	Master Module No.4 Number of Connection IDs Information
	U06-G9217 to U06-G9344	Master Module No.4 Connection ID Information Storage Area
	U06-G9984	Master Module No.4 Number of Alarm IDs Information
	U06-G9985 to U06-G10112	Master Module No.4 Alarm ID Information Storage Area
	U06-G10256	Master Module No.4 Latest Error Code Storage Area

Type	Device No.	Application
Word	U06-G10320	Master Module No.4 Parameter Access Setting
	U06-G10321	Master Module No.4 Parameter Access Target ID Specification
	U06-G10496 to U06-G10751	Master Module No.4 Parameter Storage Destination Memory Number (Output)
	U06-G11008 to U06-G11263	Master Module No.4 Parameter Storage Destination Memory Number (Input)
	U06-G12288 to U06-G18431	Master Module No.4 Parameter Storage Area
	SD1435	iQ Sensor Solution Compatible Backup/Restoration Use Request Device
	SD1436	iQ Sensor Solution Compatible Backup/Restoration Use Right Obtaining Status Device
	SD1437	iQ Sensor Solution Compatible Backup/Restoration Target Device/Execution Unit Setting Device
	SD1438	iQ Sensor Solution Compatible Backup/Restoration Target Folder Number Setting Device
	SD1439	iQ Sensor Solution Compatible Backup/Restoration Target Setting Device (Target Module)
	SD1440	iQ Sensor Solution Compatible Backup/Restoration Target Setting Device (Target Device 1)
	SD1441	iQ Sensor Solution Compatible Backup/Restoration Target Setting Device (Target Device 2)
	SD1444	iQ Sensor Solution Compatible Backup/Restoration Action Setting Device
	SD1446	iQ Sensor Solution Compatible Backup/Restoration Status Device
	SD1447	iQ Sensor Solution Compatible Backup/Restoration Execution Status Device (Total Number of Target Devices)
	SD1448	iQ Sensor Solution Compatible Backup/Restoration Execution Status Device (Number of Normal Completion Devices)
	SD1449	iQ Sensor Solution Compatible Backup/Restoration Execution Status Device (Number of Abnormal Completion Devices)
	SD1450	iQ Sensor Solution Compatible Backup/Restoration Execution Status Device (Progress per Device)
	SD1451	iQ Sensor Solution Compatible Backup Folder Number Device
	SD1452	iQ Sensor Solution Compatible Backup/Restoration Module Error Cause
	SD1453	iQ Sensor Solution Compatible Backup/Restoration Target Device Error Cause

## 5.5.2 GOT internal devices

Type	Device No.	Application
Bit	GB40	Script Trigger (Always ON)
	GB41	Bit Devices Clear (Always OFF)
	GB30000	Script No.30005 Start Trigger
	GB30001	Script No.30003 Start Trigger
	GB30002	Script No.30017 Start Trigger
	GB30003	Script No.30004, Initial Start Control Flag
	GB30004	Script No.30006 Start Trigger
	GB30005	Script No.30007 Start Trigger
	GB30006	Script No.30010 Start Trigger
	GB30007	Script No.30023, Initial Start Control Flag
	GB30008	Script No.30015 Start Trigger
	GB30009	Script No.30017, ID Match Flag
	GB30010	Script No.30013 Start Trigger
	GB30011	Script No.30018 Start Trigger
	GB30012	Script No.30022 Start Trigger
	GB30013	Script No.30036 Start Trigger
	GB30014	Script No.30024 Start Trigger
	GB30015	Script No.30025 Start Trigger
	GB30016	OFF Script Control Flag
	GB30017	Script No.30027, Initial Start Control Flag
	GB30018	Script No.30034 Start Trigger
	GB30019	Script No.30038 Start Trigger
	GB30020	Script No.30017, I/O combined module Flag
	GB30021	Script No.30026, Initial Start Control Flag
	GB30024	Parameter Individual Read Bit
	GB30025	Parameter Individual Write Bit
	GB30026	Script No.30040 Start Trigger
	GB30027	B-30000, Initial Start Control Flag
	GB30028	ON/OFF Status Check Script Start Trigger
	GB30029	Script No.30068 Start Trigger
	GB30030	Script No.30069 Start Trigger
	GB30032	Script No.30086 Start Trigger
	GB30033	Script No.30087 Start Trigger
	GB30034	Pressure Sensor, Alarm Bit Switching Flag
	GB30035	Script Initial Action Check Flag
	GB30044	Parameter Batch Read Flag
	GB30049	Script No.30079 Start Trigger
	GB30050	Script No.30080 Start Trigger
	GB30051	Script No.30033 Start Trigger
	GB30052	Script No.30081 Start Trigger
	GB30053	B-30001 Screen Switching Switch Action Conditions
	GB30600	Output Judgment Flag
	GB30601	Input Judgment Flag
	GB54000	Script No.30024 Start Trigger
	GB54001	Script No.30025 Start Trigger
	GB54002	Script No.30030 Start Trigger
	GB54004	iQSS Backup Permission Flag
	GB54005	Script No.30034 Start Trigger
	GB54006	Script No.30036 Start Trigger
	GB54008	Script No.30033 Initial Start Script Start Control Flag
	GB54009	Script No.30005,30008 Control Flag
	GB54010	Script No.30040 Start Trigger
	GB54011	Script No.30042 Start Trigger
	GB54012	Script No.30036 Delete Range Specification Flag
	GB54013	Script No.30028 Start Trigger
	GB54014	B-30100 Switching Switch Action Conditions
	GB54016	Recipe No.30001 Write Trigger
	GB54017	Recipe No.30001 Read Trigger

Type	Device No.	Application
Bit	GB54018	Recipe No.30002 Write Trigger
	GB54019	Recipe No.30002 Read Trigger
Word	GD10100 to GD10131	System Map Device (Parts Display)
	GD10132 to GD10163	System Map Device (ID(Comment))
	GD10164 to GD10195	System Map Device (ID(Numerical Display))
	GD10196 to GD10227	System Map Device (Module Type)
	GD10228	System Map Sensor Type, Offset Device
	GD10230	System Map, Offset Value Operation Device
	GD10231	I/O Check Device
	GD10232	System Map Switch Scroll Offset
	GD10233	System Map Offset
	GD10300	I/O Monitor Cursor Position Display
	GD10301	Cursor Display Bit Shift Device
	GD10302	Cursor Display Offset Device
	GD10303	I/O Monitor Current Cursor Position
	GD10400	Sensor List Selection Position Offset
	GD10401	Sensor List Scroll Offset
	GD10402 to GD10416	Sensor List, Module Type
	GD10417 to GD10431	Sensor List, Status Detail
	GD10432 to GD10446	Sensor List, User Label Define
	GD10447 to GD10461	Sensor List, Selection Position Display(Lamp)
	GD10462 to GD10476	Sensor List, ID
	GD10477 to GD10491	Sensor List, Module Model
	GD10492	Sensor List, ID, Type Display Conditions (Bit)
	GD10493	Sensor List, Offset Value Operation Device
	GD10494	Sensor List Offset
	GD10500	Module Detail Information, Remedy Display Start Position Device
	GD10501	Module Detail Information, Module Model Number Storage Device
	GD10502	Module Detail Information, Detail Information
	GD10503 to GD10518	Module Detail Information, I/O Monitor Parts Display
	GD10519 to GD10534	Module Detail Information, I/O Offset
	GD10535 to GD10550	Module Detail Information, I/O Mask Device (Object Script)
	GD10551 to GD10566	Module Detail Information, I/O Word Lamp Status Device
	GD10572	Module Detail Information, Pressure Sensor, Alarm Bit Judgment Device
	GD10600	Device Data Transfer, Offset
	GD10601	All Screens Common Offset (Individual Information)
	GD10602	All Screens Common Offset (Batch Information)
	GD11000	Device Data Transfer 1-1, External Control Device
	GD11001	Device Data Transfer 1-1, External Notification Device
	GD11002	Device Data Transfer 1-2, External Control Device
	GD11003	Device Data Transfer 1-2, External Notification Device
	GD11004	Device Data Transfer 1-3, External Control Device
	GD11005	Device Data Transfer 1-3, External Notification Device
	GD11006	Device Data Transfer 1-4, External Control Device
	GD11007	Device Data Transfer 1-4, External Notification Device
	GD11008	Device Data Transfer 2-1, External Control Device
	GD11009	Device Data Transfer 2-1, External Notification Device
	GD11010	Device Data Transfer 2-2, External Control Device
	GD11011	Device Data Transfer 2-2, External Notification Device
	GD11012	Device Data Transfer 2-3, External Control Device
	GD11013	Device Data Transfer 2-3, External Notification Device
	GD11014	Device Data Transfer 2-4, External Control Device
	GD11015	Device Data Transfer 2-4, External Notification Device
	GD11016	Device Data Transfer 3-1, External Control Device
	GD11017	Device Data Transfer 3-1, External Notification Device
	GD11018	Device Data Transfer 3-2, External Control Device
	GD11019	Device Data Transfer 3-2, External Notification Device
	GD11020	Device Data Transfer 3-3, External Control Device
	GD11021	Device Data Transfer 3-3, External Notification Device

Type	Device No.	Application
Word	GD11022	Device Data Transfer 3-4, External Control Device
	GD11023	Device Data Transfer 3-4, External Notification Device
	GD11024	Device Data Transfer 4-1, External Control Device
	GD11025	Device Data Transfer 4-1, External Notification Device
	GD11026	Device Data Transfer 4-2, External Control Device
	GD11027	Device Data Transfer 4-2, External Notification Device
	GD11028	Device Data Transfer 4-3, External Control Device
	GD11029	Device Data Transfer 4-3, External Notification Device
	GD11030	Device Data Transfer 4-4, External Control Device
	GD11031	Device Data Transfer 4-4, External Notification Device
	GD11500 to GD11528	Device Parameter (Individual)
	GD12000 to GD12511	I/O Monitor Parts Display
	GD16000 to GD16511	I/O Monitor, Module Type
	GD20000 to GD20931	Module Individual (Module 1)
	GD21000 to GD21931	Module Individual (Module 2)
	GD22000 to GD22931	Module Individual (Module 3)
	GD23000 to GD23931	Module Individual (Module 4)
	GD24000 to GD30143	Device Parameter (Batch) (Module 1)
	GD31000 to GD37143	Device Parameter (Batch) (Module 2)
	GD38000 to GD44143	Device Parameter (Batch) (Module 3)
	GD45000 to GD51143	Device Parameter (Batch) (Module 4)
	GD54000	B-30101 Cursor Display Device
	GD54001	B-30101 Data Display Offset Device
	GD54002	B-30104 Cursor Display Device
	GD54003	B-30104 Data Display Offset Device
	GD54004	Recipe Action Control Device
	GD54005	B-30101No. Display Device
	GD54006	W-30004 to 30006 Comment Display Device
	GD54007	B-30101 Data Move Offset Device
	GD54008	B-30104 Data Move Offset Device
	GD54009	Recipe External Control Device
	GD54010	Recipe No. Storage Device
	GD54011	Record No. Storage Device
	GD54012	Recipe External Notification Device
	GD54013	Recipe No. Notification Device
	GD54014	Record No. Notification Device
	GD54015 to GD54034	B-30101, B-30102 Backup Setting Work Area
	GD54035 to GD54039	B-30102 Backup Setting Comment Display Device
	GD54040 to GD54042	B-30103, B-30106 Backup/Restore Start Time Storage Device
	GD54043 to GD54062	B-30105 Backup Data Detail Display Area
	GD54070 to GD54074	B-30105 Backup Data Detail Comment Display Device
	GD54075	Script No.30042 Target Device Judgment Device
	GD54076 to GD54085	B-30101 Execution Unit Comment Display Device
	GD54086	B-30101, B-30102 Module Type Selection Switch Display Flag
	GD54087	Number of AnyWireASLINK Backup Cases
	GD54088	Number of CC-Link Backup Cases
	GD54089	Script No.30042 I/O No. at Device Dedicated Screen Setting
	GD54090	Recipe No.30001, Record No.
	GD54091	Recipe No.30002, Record No.
	GD54100 to GD56199	Backup Setting Storage Area
	GD56200 to GD61599	Backup Result Storage Area
	GD62000	Base Screen Switching Device
	GD62001	Overlap Window 1 Screen Switching Device
	GD62004	Overlap Window 2 Screen Switching Device
	GD62007	Overlap Window 3 Screen Switching Device
	GD62021	Language Switching Device
	GD62022	System Language Switching Device
	GD62031, GD62041	System Information
	GD63990	Change Time Device (Year)
	GD63991	Change Time Device (Month)

Type	Device No.	Application
Word	GD63992	Change Time Device (Day)
	GD63993	Change Time Device (Hour)
	GD63994	Change Time Device (Minute)
	GD63995	Change Time Device (Second)
	GD64000	Buffer Memory Module No. Switching Device
	GS386	Project/Screen Script Initial Action Control Device
	GS513 to GSS516	Change Time Device
	GS650 to GSS652	Current Time Device
	GS654	Touch Status External Notification (X Coordinate)
	GS655	Touch Status External Notification (Y Coordinate)
	TMP0 to TMP30, TMP100 to TMP101, TMP110 to TMP111, TMP120 to TMP128, TMP200 to TMP202, TMP950 to TMP996 TMP1000	For Script Operation



## 5.6 Comment List

Comment group No.	Comment No.	Where comments are used
495	No.1 to 2, 4, 8, 32, 100 to 104, 200 to 202, 300 to 305, 400 to 401, 500	B-30006 to 30062
496	Please refer to "5.4 Slave Module Detail Information Screen Correspondence Table". In GOT, the comment No. and the base screen No. to display are set for each module model number of slave modules. For the module model numbers of slave modules after A0000, comments are set in the GOT comment No.10000 or later. (A0000 cannot be set to the comment No. of GOT.)	
497		
498	No.1 to 255, 513 to 767, 1001 to 1255, 1513 to 1767, 2001 to 2255, 2513 to 2767, 3001 to 3255, 3513 to 3767	B-30005 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
499	No.1 to 2, 4, 8, 32	B-30002, B-30005 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	100 to 104, 200 to 202, 300 to 305, 400 to 401, 500	B-30002, B-30005 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062, W-30101
	No.18433 to 18443, 32000 to 32017	W-30101
	No.32767	B-30002, B-30005 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
500	No.20	B-30002
	No.22	B-30005
	No.23 to 24	B-30005 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.25	B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.26	B-30002 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.27	B-30001, B-30002
	No.28	B-30002 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.29	B-30001 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.30	B-30001
	No.32 to 35	B-30000, B-30002 to 30004
	No.36	B-30000, B-30002 to 30005
	No.37	B-30002
	No.38	B-30000, B-30002 to 30004, W-30001
	No.39	B-30000, B-30002 to 30004
	No.40	B-30000, B-30002 to 30005, W-30001, W-30003

Comment group No.	Comment No.	Where comments are used
500	No.41	B-30002 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.42	B-30002 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.43	B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.45 to 46	B-30000, B-30003 to B-30004
	No.47 to 48	B-30003 to 30004
	No.49 to 50	B-30005
	No.51	B-30005 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.52 to 58	B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.59	B-30006, B-30008 to 30009, B-30013
	No.60 to 61	B-30006 to 30052
	No.62 to 63	B-30010 to 30015, B-30017 to 30018, B-30020 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30035, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.68	B-30011, B-30013, B-30017
	No.69	B-30010, B-30012, B-30014 to 30015, B-30032
	No.70 to 71	B-30013 to 30015
	No.75	B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.76 to 77	B-30006, B-30008, B-30009
	No.78 to 79	W-30004
	No.81 to 88	W-30003
	No.91	B-30003
	No.92	B-30004
	No.93	B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.94	B-30001
	No.95	B-30002 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.96	B-30001, B-30004
	No.97	B-30001, B-30003
	No.99	B-30100
	No.117	B-30010 to 30015, B-30017
	No.118	B-30010 to 30015, B-30017, B-30035
	No.119	B-30010 to 30015, B-30017, B-30032
	No.120	B-30010 to 30012, B-30014 to 30015, B-30017, B-30032

Comment group No.	Comment No.	Where comments are used
500	No.121	B-30010, B-30012, B-30014 to 30015, B-30020 to 30021, B-30024 to 30025, B-30028 to 30029, B-30032, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.123	B-30010 to 30012, B-30014 to 30015, B-30017, B-30032
	No.124	B-30014 to 30015
	No.125	B-30014, B-30016
	No.126	B-30010 to 30015, B-30017, B-30032
	No.127	B-30014 to 30016
	No.128	W-30002
	No.129 to 132	B-30010, B-30012, B-30032
	No.133 to 135	B-30010 to 30012
	No.136 to 137	B-30010, B-30012
	No.138 to 140	B-30011, B-30013, B-30017
	No.141 to 143	B-30011
	No.144	B-30014, B-30016
	No.145	B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.146 to 149	B-30000
	No.152	W-30005
	No.153 to 156	B-30001 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062
	No.165 to 166	B-30020 to 30021, B-30024 to 30025, B-30028 to 30029, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.170	B-30020 to 30021, B-30024 to 30025, B-30028 to 30029, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.171 to 172	B-30020, B-30024, B-30028, B-30053, B-30057, B-30061
	No.173	B-30020 to 30021, B-30024 to 30025, B-30028 to 30029, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.174 to 175	B-30020, B-30024, B-30028, B-30053, B-30057, B-30061
	No.176 to 178	B-30020, B-30021, B-30024, B-30025, B-30028, B-30029, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.179 to 180	B-30020 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.181 to 182	B-30021, B-30025, B-30029, B-30054, B-30058, B-30062
	No.189 to 192	B-30033, B-30034
	No.193 to 195	B-30034
	No.203 to 206	B-30035
	No.207 to 214	B-30021, B-30025, B-30029, B-30054, B-30058, B-30062
	No.216	B-30000
	No.217 to 220	B-30035
	No.221	B-30035, B-30036
	No.222	B-30035
	No.223 to 224	B-30035
	No.225	B-30035, B-30036

Comment group No.	Comment No.	Where comments are used
500	No.226 to 235	B-30035
	No.237 to 240	B-30036
	No.241	B-30020 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.242	W-30006
	No.243	B-30020 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.244	B-30020 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.245	B-30020 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031, B-30053 to 30054, B-30057 to 30058, B-30061 to 30062
	No.246	W-30007
	No.247	B-30001
	No.1001	B-30001, B-30100
	No.1002	B-30101
	No.1003	B-30102
	No.1004	B-30103
	No.1005	B-30104
	No.1006	B-30105
	No.1007	B-30106
	No.1008 to 1009	B-30100
	No.1010	B-30101 to 30102, B-30104, B-30105
	No.1011	B-30102 to 30103, B-30105, B-30106
	No.1012	B-30101 to 30106
	No.1013	B-30103, B-30106
	No.1014 to 1017	B-30101
	No.1018	B-30101, B-30104
	No.1019	B-30101, B-30103, B-30104, B-30106
	No.1020 to 1021	B-30101, B-30104
	No.1022	B-30101, B-30102, B-30104, B-30105
	No.1023 to 1024	B-30102, B-30105
	No.1025 to 1026	B-30101, B-30103, B-30106
	No.1027	B-30103, B-30106
	No.1028	B-30102, B-30105
	No.1029	B-30105
	No.1030	B-30102, B-30105
	No.1031	B-30102
	No.1032 to 1033	B-30103, B-30106
	No.1034 to 1035	B-30102, B-30105
	No.1036	B-30102
	No.1037 to 1038	B-30103
	No.1039 to 1044	B-30103, B-30106
	No.1045	B-30101 to 30103, B-30105 to 30106
	No.1046 to 1048	B-30103
	No.1049 to 1051	B-30104
	No.1052 to 1056	B-30106
	No.1057	W-30004, W-30006 to 30007, W-30100 to 30102
	No.1058	W-30007, W-30100, W-30102
	No.1068	W-30006
	No.1069 to 1071	W-30102
	No.1072	W-30100
	No.1073 to 1074	B-30104
	No.1075 to 1077	B-30101, B-30104
	No.1078	B-30104
	No.1079	B-30101, B-30104

Comment group No.	Comment No.	Where comments are used
500	No.1080 to 1081	W-30102
	No.1082	W-30100
	No.1083 to 1084	B-3010 to 30103, B-30105 to 30106
	No.1086	B-30103, B-30106
	No.1087 to 1088	B-30101
	No.1089	W-30100

## 5.7 Device Data Transfer List

### ID: 201 Transfer 1-1

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11000
	Trigger Device	GD11000.b0
	Transfer Inverting Flag Device	GD11000.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11001
	Device Data Transfer Notification Signal	GD11001.b0
	Device Data Transfer Error Notification Signal	GD11001.b15
Device	Block Number	10
Block 1	Device Type	Signed BIN16
	Points	16
	Source Device	U00-G0
	Destination Device	GD20000
	Offset	None
Block 2	Device Type	Signed BIN16
	Points	16
	Source Device	U00-G4096
	Destination Device	GD20016
	Offset	None
Block 3	Device Type	Signed BIN16
	Points	129
	Source Device	U00-G8192
	Destination Device	GD20032
	Offset	None
Block 4	Device Type	Signed BIN16
	Points	1
	Source Device	U00-G8960
	Destination Device	GD20161
	Offset	None

Item		Settings
Block 5	Device Type	Signed BIN16
	Points	129
	Source Device	U00-G9216
	Destination Device	GD20162
	Offset	None
Block 6	Device Type	Signed BIN16
	Points	129
	Source Device	U00-G9984
	Destination Device	GD20291
	Offset	None
Block 7	Device Type	Signed BIN16
	Points	256
	Source Device	U00-G10496
	Destination Device	GD20420
	Offset	None
Block 8	Device Type	Signed BIN16
	Points	256
	Source Device	U00-G11008
	Destination Device	GD20676
	Offset	None
Block 9	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11000.b0
	Offset	None
Block 10	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30016
	Offset	None

**ID: 202 Transfer 1-2**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11002
	Trigger Device	GD11002.b0
	Transfer Inverting Flag Device	GD11002.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11003
	Device Data Transfer Notification Signal	GD11003.b0
	Device Data Transfer Error Notification Signal	GD11003.b15
Device	Block Number	5
Block 1	Device Type	Signed BIN16
	Points	1
	Source Device	U00-G0
	Destination Device	GD11500
	Offset	Source GD10600
Block 2	Device Type	Signed BIN16
	Points	28
	Source Device	U00-G20
	Destination Device	GD11501
	Offset	Source GD10600
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11002.b0
	Offset	None
Block 4	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30021
	Offset	None



**ID: 203 Transfer 1-3**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11004
	Trigger Device	GD11004.b0
	Transfer Inverting Flag Device	GD11004.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11005
	Device Data Transfer Notification Signal	GD11005.b0
	Device Data Transfer Error Notification Signal	GD11005.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	6144
	Source Device	U00-G12288
	Destination Device	GD24000
	Offset	None
Block 2	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11004.b0
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	GB30051
	Offset	None

**ID: 204 Transfer 1-4**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11006
	Trigger Device	GD11006.b0
	Transfer Inverting Flag Device	GD11006.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11007
	Device Data Transfer Notification Signal	GD11007.b0
	Device Data Transfer Error Notification Signal	GD11007.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	19
	Source Device	GD11501
	Destination Device	U00-G1
	Offset	Destination GD10600
Block 2	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	Y0010
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11006.b0
	Offset	None

**ID: 205 Transfer 2-1**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11008
	Trigger Device	GD11008.b0
	Transfer Inverting Flag Device	GD11008.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11009
	Device Data Transfer Notification Signal	GD11009.b0
	Device Data Transfer Error Notification Signal	GD11009.b15
Device	Block Number	10
Block 1	Device Type	Signed BIN16
	Points	16
	Source Device	U02-G0
	Destination Device	GD21000
	Offset	None
Block 2	Device Type	Signed BIN16
	Points	16
	Source Device	U02-G4096
	Destination Device	GD21016
	Offset	None
Block 3	Device Type	Signed BIN16
	Points	129
	Source Device	U02-G8192
	Destination Device	GD21032
	Offset	None
Block 4	Device Type	Signed BIN16
	Points	1
	Source Device	U02-G8960
	Destination Device	GD21161
	Offset	None

Item		Settings
Block 5	Device Type	Signed BIN16
	Points	129
	Source Device	U02-G9216
	Destination Device	GD21162
	Offset	None
Block 6	Device Type	Signed BIN16
	Points	129
	Source Device	U02-G9984
	Destination Device	GD21291
	Offset	None
Block 7	Device Type	Signed BIN16
	Points	256
	Source Device	U02-G10496
	Destination Device	GD21420
	Offset	None
Block 8	Device Type	Signed BIN16
	Points	256
	Source Device	U02-G11008
	Destination Device	GD21676
	Offset	None
Block 9	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11008.b0
	Offset	None
Block 10	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30016
	Offset	None

**ID: 206 Transfer 2-2**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11010
	Trigger Device	GD11010.b0
	Transfer Inverting Flag Device	GD11010.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11011
	Device Data Transfer Notification Signal	GD11011.b0
	Device Data Transfer Error Notification Signal	GD11011.b15
Device	Block Number	5
Block 1	Device Type	Signed BIN16
	Points	1
	Source Device	U02-G0
	Destination Device	GD11500
	Offset	Source GD10600
Block 2	Device Type	Signed BIN16
	Points	28
	Source Device	U02-G20
	Destination Device	GD11501
	Offset	Source GD10600
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11010.b0
	Offset	None
Block 4	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30021
	Offset	None

**ID: 207 Transfer 2-3**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11012
	Trigger Device	GD11012.b0
	Transfer Inverting Flag Device	GD11012.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11013
	Device Data Transfer Notification Signal	GD11013.b0
	Device Data Transfer Error Notification Signal	GD11013.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	6144
	Source Device	U02-G12288
	Destination Device	GD31000
	Offset	None
Block 2	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11012.b0
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	GB30051
	Offset	None

**ID: 208 Transfer 2-4**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11014
	Trigger Device	GD11014.b0
	Transfer Inverting Flag Device	GD11014.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11015
	Device Data Transfer Notification Signal	GD11015.b0
	Device Data Transfer Error Notification Signal	GD11015.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	19
	Source Device	GD11501
	Destination Device	U02-G1
	Offset	Destination GD10600
Block 2	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	Y0030
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11014.b0
	Offset	None

**ID: 209 Transfer 3-1**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11016
	Trigger Device	GD11016.b0
	Transfer Inverting Flag Device	GD11016.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11017
	Device Data Transfer Notification Signal	GD11017.b0
	Device Data Transfer Error Notification Signal	GD11017.b15
Device	Block Number	10
Block 1	Device Type	Signed BIN16
	Points	16
	Source Device	U04-G0
	Destination Device	GD22000
	Offset	None
Block 2	Device Type	Signed BIN16
	Points	16
	Source Device	U04-G4096
	Destination Device	GD22016
	Offset	None
Block 3	Device Type	Signed BIN16
	Points	129
	Source Device	U04-G8192
	Destination Device	GD22032
	Offset	None
Block 4	Device Type	Signed BIN16
	Points	1
	Source Device	U04-G8960
	Destination Device	GD22161
	Offset	None



Item		Settings
Block 5	Device Type	Signed BIN16
	Points	129
	Source Device	U04-G9216
	Destination Device	GD22162
	Offset	None
Block 6	Device Type	Signed BIN16
	Points	129
	Source Device	U04-G9984
	Destination Device	GD22291
	Offset	None
Block 7	Device Type	Signed BIN16
	Points	256
	Source Device	U04-G10496
	Destination Device	GD22420
	Offset	None
Block 8	Device Type	Signed BIN16
	Points	256
	Source Device	U04-G11008
	Destination Device	GD22676
	Offset	None
Block 9	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11016.b0
	Offset	None
Block 10	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30016
	Offset	None

**ID: 210 Transfer 3-2**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11018
	Trigger Device	GD11018.b0
	Transfer Inverting Flag Device	GD11018.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11019
	Device Data Transfer Notification Signal	GD11019.b0
	Device Data Transfer Error Notification Signal	GD11019.b15
Device	Block Number	5
Block 1	Device Type	Signed BIN16
	Points	1
	Source Device	U04-G0
	Destination Device	GD11500
	Offset	Source GD10600
Block 2	Device Type	Signed BIN16
	Points	28
	Source Device	U04-G20
	Destination Device	GD11501
	Offset	Source GD10600
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11018.b0
	Offset	None
Block 4	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30021
	Offset	None

**ID: 211 Transfer 3-3**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11020
	Trigger Device	GD11020.b0
	Transfer Inverting Flag Device	GD11020.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11021
	Device Data Transfer Notification Signal	GD11021.b0
	Device Data Transfer Error Notification Signal	GD11021.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	6144
	Source Device	U04-G12288
	Destination Device	GD38000
	Offset	None
Block 2	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11020.b0
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	GB30051
	Offset	None

**ID: 212 Transfer 3-4**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11022
	Trigger Device	GD11022.b0
	Transfer Inverting Flag Device	GD11022.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11023
	Device Data Transfer Notification Signal	GD11023.b0
	Device Data Transfer Error Notification Signal	GD11023.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	19
	Source Device	GD11501
	Destination Device	U04-G1
	Offset	Destination GD10600
Block 2	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	Y0050
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11022.b0
	Offset	None

**ID: 213 Transfer 4-1**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11024
	Trigger Device	GD11024.b0
	Transfer Inverting Flag Device	GD11024.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11025
	Device Data Transfer Notification Signal	GD11025.b0
	Device Data Transfer Error Notification Signal	GD11025.b15
Device	Block Number	10
Block 1	Device Type	Signed BIN16
	Points	16
	Source Device	U06-G0
	Destination Device	GD23000
	Offset	None
Block 2	Device Type	Signed BIN16
	Points	16
	Source Device	U06-G4096
	Destination Device	GD23016
	Offset	None
Block 3	Device Type	Signed BIN16
	Points	129
	Source Device	U06-G8192
	Destination Device	GD23032
	Offset	None
Block 4	Device Type	Signed BIN16
	Points	1
	Source Device	U06-G8960
	Destination Device	GD23161
	Offset	None

Item		Settings
Block 5	Device Type	Signed BIN16
	Points	129
	Source Device	U06-G9216
	Destination Device	GD23162
	Offset	None
Block 6	Device Type	Signed BIN16
	Points	129
	Source Device	U06-G9984
	Destination Device	GD23291
	Offset	None
Block 7	Device Type	Signed BIN16
	Points	256
	Source Device	U06-G10496
	Destination Device	GD23420
	Offset	None
Block 8	Device Type	Signed BIN16
	Points	256
	Source Device	U06-G11008
	Destination Device	GD23676
	Offset	None
Block 9	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11024.b0
	Offset	None
Block 10	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30016
	Offset	None

**ID: 214 Transfer 4-2**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11026
	Trigger Device	GD11026.b0
	Transfer Inverting Flag Device	GD11026.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11027
	Device Data Transfer Notification Signal	GD11027.b0
	Device Data Transfer Error Notification Signal	GD11027.b15
Device	Block Number	5
Block 1	Device Type	Signed BIN16
	Points	1
	Source Device	U06-G0
	Destination Device	GD11500
	Offset	Source GD10600
Block 2	Device Type	Signed BIN16
	Points	28
	Source Device	U06-G20
	Destination Device	GD11501
	Offset	Source GD10600
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11026.b0
	Offset	None
Block 4	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30021
	Offset	None

**ID: 215 Transfer 4-3**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11028
	Trigger Device	GD11028.b0
	Transfer Inverting Flag Device	GD11028.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11029
	Device Data Transfer Notification Signal	GD11029.b0
	Device Data Transfer Error Notification Signal	GD11029.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	6144
	Source Device	U06-G12288
	Destination Device	GD45000
	Offset	None
Block 2	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11028.b0
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	GB30051
	Offset	None



**ID: 216 Transfer 4-4**

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD11030
	Trigger Device	GD11030.b0
	Transfer Inverting Flag Device	GD11030.b1
External Notification Information	<input checked="" type="checkbox"/> External Notification Device	GD11031
	Device Data Transfer Notification Signal	GD11031.b0
	Device Data Transfer Error Notification Signal	GD11031.b15
Device	Block Number	3
Block 1	Device Type	Signed BIN16
	Points	19
	Source Device	GD11501
	Destination Device	U06-G1
	Offset	Destination GD10600
Block 2	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	Y0070
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD11030.b0
	Offset	None

## 5.8 Recipe List

### 5.8.1 Common Setting

External Control Information	
External Control Device	GD54009
Recipe No. Storage Device	GD54010
Record No. Storage Device	GD54011
External Notification Information	
External Notification Device	GD54012
Recipe No. Notification Device	GD54013
Record No. Notification Device	GD54014

### 5.8.2 Individual Setting

#### Recipe No.30001 Recipe 1

Item		Settings
Recipe File	-	Use a recipe file
	Drive Name	A: Standard SD Card
	Folder Name	Pacage1
	File Name	ARP30001.G2P
Trigger Device	Write Trigger 1	GB54016
	Read Trigger 1	GB54017
	Record No. Device	GD54090
Block Number		1
Record Number		1
Block 1	Device	GD54100
	Device Type	Signed BIN16
	Points	2100

**Recipe No.30002 Recipe 2**

Item		Settings
Recipe File	-	Use a recipe file
	Drive Name	A: Standard SD Card
	Folder Name	Pacage1
	File Name	ARP30002.G2P
Trigger Device	Write Trigger 1	GB54018
	Read Trigger 1	GB54019
	Record No. Device	GD54091
Block Number		2
Record Number		1
Block 1	Device	GD56200
	Device Type	Signed BIN16
	Points	5400
Block 2	Device	GD54087
	Device Type	Signed BIN16
	Points	2

## 5.9 Script List

Item	Settings
Project script	Specified
Screen script	Specified: B-30001 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062, B-30100 to 30106
Object script	Specified: B-30006 to 30021, B-30023 to 30025, B-30027 to 30029, B-30031 to 30042, B-30051 to 30054, B-30057 to 30058, B-30061 to 30062, B-30101, W-30003
Script symbol	Specified
Object script symbol	Specified

### 5.9.1 Project script

Script No.	30020	Script name	Script30020
Comment	Initial Start Control		
Data type	Signed BIN16	Trigger type	Rise, GB40
//Actions at Screen Startup [w:GS386] = 1; //Inhibit Initial Start of Scripts			
Script No.	30033	Script name	Script30033
Comment	Status Read		
Data type	Signed BIN16	Trigger type	ON, GB30051
//Read Statuses switch([w:GD64000]) { case Master_1: set([b:GD11000.b0]); break;  case Master_2: set([b:GD11008.b0]); break;  case Master_3: set([b:GD11016.b0]); break;  case Master_4: set([b:GD11024.b0]); break;  default: break; } rst([b:GB30051]);			
Script No.	30042	Script name	Script30042
Comment	Process at Screen Start		
Data type	Signed BIN16	Trigger type	Rise, GB40
//Initialize iQSS Backup/Restore Sample Screen [w:GS386] = 1; //Inhibit Initial Start of Scripts [w:GD54005] = 1; //Set 1 to the first number of iQSS backup //Determine Target Device [w:GD54075] = 1; //0: Not Specified, 1: ASLINK, 2: CC-Link //[w:GD54089] = 0; //To use as a dedicated screen, enter I/O No. here  //When initially displaying iQSS Backup Screen or iQSS Restoration Screen, //Display Cursor in No.1 set([b:GD54000.b0]); set([b:GD54002.b0]);  //When starting screen, to read iQSS backup setting and iQSS backup history, //start script.			

set([b:GB54010]);			
//Specify action conditions of screen switching switch on the menu screen.			
set([b:GB54014]);			
Script No.	30029	Script name	Script30029
Comment	Alarm, Error Info Acquisition		
Data type	Signed BIN16	Trigger type	Sampling, 1 Sec
//Observe Error and Alarm			
<pre> switch([w:GD64000]) {   case Master_1: if(([w:GD20032] != [w:U00-G8192])    ([w:GD20291] != [w:U00-G9984]))     {       set([b:GD11004.b0]);     }     break;    case Master_2: if(([w:GD21032] != [w:U02-G8192])    ([w:GD21291] != [w:U02-G9984]))     {       set([b:GD11012.b0]);     }     break;    case Master_3: if(([w:GD22032] != [w:U04-G8192])    ([w:GD22291] != [w:U04-G9984]))     {       set([b:GD11020.b0]);     }     break;    case Master_4: if(([w:GD23032] != [w:U06-G8192])    ([w:GD23291] != [w:U06-G9984]))     {       set([b:GD11028.b0]);     }     break;    default:      break; } </pre>			

## 5.9.2 Screen script

### Base screen 30001

Script No.	30041	Script name	Script30041
Comment	NW Map, Sensor List Disp. Control		
Data type	Signed BIN16	Trigger type	ON, GB40
<pre> //Control not to display the Network Map and Sensor List at the initial startup. if([b:GB30027] == OFF) {   [w:TMP0000] = 0xFFFF;   fmov([w:TMP0000],[w:GD10100],128);   fmov([w:TMP0000],[w:GD10402],48);   set([b:GB30027]); } </pre>			
Script No.	30085	Script name	Script30085
Comment	Sensor Auto Recognition Start		
Data type	Signed BIN16	Trigger type	ON, GB30052
<pre> //Take the sensor auto recognition. switch([w:GD64000]) {   case Master_1: set([b:Y0001]);     break;    case Master_2: set([b:Y0021]); </pre>			

```

        break;

    case Master_3: set([b:Y0041]);
        break;

    case Master_4: set([b:Y0061]);
        break;

    default:      break;
}

```

```
rst([b:GB30052]);
```

Script No.	30072	Script name	Script30072
Comment	Module No.1 Auto Recognition		
Data type	Signed BIN16	Trigger type	ON, Y0001

```

//Master Module No.1
//Perform the sensor auto recognition.
if([b:X0011] == ON)
{
    //When the parameter batch read flag turns on,
    //Reset the auto recognition flag, start device data transfer.
    if([b:GB30044] == ON)
    {
        rst([b:Y0001]);
        set([b:GD11004.b0]);
        [w:GD62007] = 0;
        rst([b:GB30044]);
        rst([b:GB30053]); //Switch Action Conditions
    }
}
else{
    //When the parameter batch read start,
    //turn on the flag.
    if([b:GB30044] == OFF)
    {
        set([b:GB30044]);
    }
}
}

```

Script No.	30074	Script name	Script30074
Comment	Module No.2 Auto Recognition		
Data type	Signed BIN16	Trigger type	ON, Y0021

```

//Master Module No.2
//Perform the sensor auto recognition.
if([b:X0031] == ON)
{
    //When the parameter batch read flag turns on,
    //Reset the auto recognition flag, start device data transfer.
    if([b:GB30044] == ON)
    {
        rst([b:Y0021]);
        set([b:GD11012.b0]);
        [w:GD62007] = 0;
        rst([b:GB30044]);
        rst([b:GB30053]); //Switch Action Conditions
    }
}
else{
    //When the parameter batch read start,
    //turn on the flag.
    if([b:GB30044] == OFF)
    {
        set([b:GB30044]);
    }
}
}

```

Script No.	30077	Script name	Script30077
Comment	Module No.3 Auto Recognition		
Data type	Signed BIN16	Trigger type	ON, Y0041
<pre>//Master Module No.3 //Perform the sensor auto recognition. if([b:X0051] == ON) {     //When the parameter batch read flag turns on,     //Reset the auto recognition flag, start device data transfer.     if([b:GB30044] == ON)     {         rst([b:Y0041]);         set([b:GD11020.b0]);         [w:GD62007] = 0;         rst([b:GB30044]);         rst([b:GB30053]); //Switch Action Conditions     } } else{     //When the parameter batch read start,     //turn on the flag.     if([b:GB30044] == OFF)     {         set([b:GB30044]);     } }</pre>			
Script No.	30080	Script name	Script30080
Comment	Module No.4 Auto Recognition		
Data type	Signed BIN16	Trigger type	ON, Y0061
<pre>//Master Module No.4 //Perform the sensor auto recognition. if([b:X0071] == ON) {     //When the parameter batch read flag turns on,     //Reset the auto recognition flag, start device data transfer.     if([b:GB30044] == ON)     {         rst([b:Y0061]);         set([b:GD11028.b0]);         [w:GD62007] = 0;         rst([b:GB30044]);         rst([b:GB30053]); //Switch Action Conditions     } } else{     //When the parameter batch read start,     //turn on the flag.     if([b:GB30044] == OFF)     {         set([b:GB30044]);     } }</pre>			
Script No.	30035	Script name	Script30035
Comment	Module No.1 Transfer Start		
Data type	Signed BIN16	Trigger type	ON, Y0011
<pre>//Master Module No.1 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0011] == 1) {     set([b:GD11004.b0]);     rst([b:Y0011]);     [w:GD62007] = 0;     rst([b:GB30053]); //Switch Action Conditions }</pre>			
Script No.	30073	Script name	Script30073

Comment	Module No.2 Data Transfer Start		
Data type	Signed BIN16	Trigger type	ON, Y0031
<pre>//Master Module No.2 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0031] == ON) {     set([b:GD11012.b0]);     rst([b:Y0031]);     [w:GD62007] = 0;     rst([b:GB30053]); //Switch Action Conditions }</pre>			
Script No.	30076	Script name	Script30076
Comment	Module No.3 Data Transfer Start		
Data type	Signed BIN16	Trigger type	ON, Y0051
<pre>//Master Module No.3 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0051] == ON) {     set([b:GD11020.b0]);     rst([b:Y0051]);     [w:GD62007] = 0;     rst([b:GB30053]); //Switch Action Conditions }</pre>			
Script No.	30079	Script name	Script30079
Comment	Module No.4 Data Transfer Start		
Data type	Signed BIN16	Trigger type	ON, Y0071
<pre>//Master Module No.4 //Transfer parameters to internal devices using Device Data Transfer. if([b:X0071] == ON) {     set([b:GD11028.b0]);     rst([b:Y0071]);     [w:GD62007] = 0;     rst([b:GB30053]); //Switch Action Conditions }</pre>			

#### Base screen 30002

Script No.	30023	Script name	Script30023
Comment	B-30002,30005 Scr. Start Process		
Data type	Signed BIN16	Trigger type	ON, GB40
<pre>//Read Statuses or Parameters if([b:GB30007] == OFF) {     if(([w:U00-G8192] != 0)    ([w:U00-G9984] != 0))     {         switch([w:GD64000])         {             case Master_1: set([b:GD11004.b0]);                           break;              case Master_2: set([b:GD11012.b0]);                           break;              case Master_3: set([b:GD11020.b0]);                           break;              case Master_4: set([b:GD11028.b0]);                           break;              default:      break;         }     }     }else{         switch([w:GD64000])</pre>			



```

{
    case Master_1: set([b:GD11000.b0]);
        break;

    case Master_2: set([b:GD11008.b0]);
        break;

    case Master_3: set([b:GD11016.b0]);
        break;

    case Master_4: set([b:GD11024.b0]);
        break;

    default:      break;
}
}

```

```
set([b:GB30007]);
```

Script No.	30019	Script name	Script30019
Comment	Script No.30005 Start		
Data type	Signed BIN16	Trigger type	Ordinary

```

//Display Objects
if([b:GB30016] == OFF)
{
    set([b:GB30000]);
    set([b:GB30016]);
}

```

Script No.	30005	Script name	Script30005
Comment	B-30002 Parts Display		
Data type	Signed BIN16	Trigger type	ON, GB30000

```

//Display Objects
[w:GD10233] = 0;
[w:TMP0004] = 0;
[w:TMP0002] = [w:GD10232];
[w:TMP0009] = 0;
fmov([w:TMP0004],[w:TMP0010],7);

//Make Settings to Switch Parts Display According To Language
if([w:GD62021] > 0)
{
    [w:TMP0009] = [w:GD62021] - 1;
}

while([w:GD10233] < 32)
{
    //Check Connection Status
    if(([w:GD20162[w:GD10601]] != 0) && ([w:TMP0002] < [w:GD20162[w:GD10601]]))
    {
        [w:TMP0003] = [w:GD10601] + [w:TMP0002];
        [w:TMP0010] = [w:GD20163[w:TMP0003]]; //Connection ID
        [w:TMP0002] = [w:TMP0002] + 1;

        if(([w:TMP0010] & 0x0200) == 0)
        {
            set([b:GB30600]); //Output Bit
            [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9;
        }else{
            set([b:GB30601]); //Input Bit
            [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9;
        }
    }

    //Device Parameter

```

```

[w:TMP0004] = ([w:TMP0010] & 0x00FF) + [w:GD10601]; //Offset to Obtain Device Parameter
if([b:GB30600] == ON)
{
    //Output
    [w:TMP0005] = [w:GD20420[w:TMP0004]];
}else{
    //Input
    [w:TMP0005] = [w:GD20676[w:TMP0004]];
}
[w:TMP0014] = ([w:TMP0005] - 12288) + [w:GD10602];

//I/O Points Pattern
//[w:TMP0006] = [w:TMP0014] + 43; //Parameter Offset
[w:TMP0007] = [w:GD24043[w:TMP0014]];
[w:TMP0015] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0016] = ([w:TMP0007] & 0x003F) + 1; //Points

//Module Model Number
//[w:TMP0008] = [w:TMP0014] + 44; //Parameter Offset
[w:TMP0017] = [w:GD24044[w:TMP0014]];

//Alarm Information
if([w:GD20291[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0018] = [w:TMP0004] + [w:GD10601];
    while([w:TMP0004] < [w:GD20291[w:GD10601]])
    {
        if([w:GD20292[w:TMP0018]] == [w:TMP0010])
        {
            [w:TMP0012] = 1; //Alarm Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0018] = [w:TMP0018] + 1;
    }
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0018] = [w:TMP0004] + [w:GD10601];
    while([w:TMP0004] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0018]] == [w:TMP0010])
        {
            [w:TMP0013] = 1; //Error Occurrence
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0018] = [w:TMP0018] + 1;
    }
}

//Display
if([w:TMP0013] == 1)
{
    //Error Occurrence
    switch([w:TMP0015])
    {
        case 0: [w:GD10100[w:GD10233]] = 30030 + [w:TMP0009]; //Input

```

```

        break;

        case 1: [w:GD10100[w:GD10233]] = 30021 + [w:TMP0009]; //Output
        break;

        case 2: [w:GD10100[w:GD10233]] = 30039 + [w:TMP0009]; //Combined
        break;
    }

    [w:GD10132[w:GD10233]] = 0;
    [w:GD10164[w:GD10233]] = [w:TMP0010] & 0x00FF; //ID
    [w:GD10196[w:GD10233]] = [w:TMP0017]; //Module Model No.
}else{
    if([w:TMP0012] == 1)
    {
        //Alarm Occurrence
        switch([w:TMP0015])
        {
            case 0: [w:GD10100[w:GD10233]] = 30027 + [w:TMP0009]; //Input
            break;

            case 1: [w:GD10100[w:GD10233]] = 30018 + [w:TMP0009]; //Output
            break;

            case 2: [w:GD10100[w:GD10233]] = 30036 + [w:TMP0009]; //Combined
            break;

        }
        [w:GD10132[w:GD10233]] = 0;
        [w:GD10164[w:GD10233]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10196[w:GD10233]] = [w:TMP0017]; //Module Model No.
    }else{
        //Normal
        switch([w:TMP0015])
        {
            case 0: [w:GD10100[w:GD10233]] = 30024 + [w:TMP0009]; //Input
            break;

            case 1: [w:GD10100[w:GD10233]] = 30015 + [w:TMP0009]; //Output
            break;

            case 2: [w:GD10100[w:GD10233]] = 30033 + [w:TMP0009]; //Combined
            break;

        }
        [w:GD10132[w:GD10233]] = 0;
        [w:GD10164[w:GD10233]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10196[w:GD10233]] = [w:TMP0017]; //Module Model No.
    }
}

[w:GD10233] = [w:GD10233] + 1;
rst([b:GB30600]); //Output Bit
rst([b:GB30601]); //Input Bit

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004],[w:TMP0010],5);
}else{
    //If data does not exist, do not display.
    [w:GD10100[w:GD10233]] = 0xFFFF;
    [w:GD10132[w:GD10233]] = 0xFFFF;
    [w:GD10164[w:GD10233]] = 0xFFFF;

```

<pre> [w:GD10196[w:GD10233]] = 0xFFFF;  [w:GD10233] = [w:GD10233] + 1; rst([b:GB30600]); //Output Bit rst([b:GB30601]); //Input Bit } }  //Trigger Reset rst([b:GB30000]);  //Initial Action Check Flag. set([b:GB30035]); </pre>			
Script No.	30021	Script name	Script30021
Comment	Flags Clear		
Data type	Signed BIN16	Trigger type	When closing a screen
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]); rst([b:GB30018]); rst([b:GB30035]);  [w:TMP0004] = 0; fmov([w:TMP0004],[w:GD12000],512); //Initialize Object Values(B-30003, 4) fmov([w:TMP0004],[w:GD10301],3); //Clear Cursor Position Information(B-30003,4) fmov([w:TMP0004],[w:GD10432],15); //Work Area Clear [w:GD10300] = 0; //Cursor Position Information Clear(B-30003,4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
Script No.	30006	Script name	Script30006
Comment	Scroll Up		
Data type	Signed BIN16	Trigger type	Rise, GB30004
<pre> //Scroll List Up if([w:GD10232] != 0) {     [w:GD10232] = [w:GD10232] - 32;     set([b:GB30000]); } </pre>			
Script No.	30007	Script name	Script30007
Comment	Scroll Down		
Data type	Signed BIN16	Trigger type	Rise, GB30005
<pre> //Scroll List Down if((([w:GD10232] + 32) &lt; [w:U00-G8960]) &amp;&amp; ([w:U00-G8960] &gt; 32)) {     [w:GD10232] = [w:GD10232] + 32;     set([b:GB30000]); } </pre>			
Script No.	30010	Script name	Script30010
Comment	B-30002 Detail Screen Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30006
<pre> //Determine detail screen to display.  [w:TMP0000] = [w:GD10230] &amp; 0x00FF; [w:TMP0002] = [w:TMP0000] + [w:GD10601];  if(([w:GD10231] &gt;= 30015) &amp;&amp; ([w:GD10231] &lt;= 30023)) {     //Output     [w:GD10600] = [w:GD20420[w:TMP0002]]; //Offset </pre>			

```

[w:TMP0001] = [w:TMP0000];
}else{
  //Input
  [w:GD10600] = [w:GD20676[w:TMP0002]]; //Offset
  [w:TMP0001] = [w:TMP0000] + 0x0200;
}

[w:U00-G10320] = 0; //Read Individual Parameter
[w:U00-G10321] = [w:TMP0001]; //Read Target ID

//Select base screen to display.
switch([w:GD10196[w:GD10228]] & 0xF000)
{
  case 0x0000: [w:TMP0003] = 0;
               break;

  case 0x1000: [w:TMP0003] = 1;
               break;

  case 0x2000: [w:TMP0003] = 2;
               break;

  case 0x3000: [w:TMP0003] = 3;
               break;

  case 0xA000: [w:TMP0003] = 10;
               break;

}

if([w:TMP0003] == 0)
{
  switch([w:GD10196[w:GD10228]])
  {
    case 0x0100: [w:GD62000] = 30006;
                 break;

    case 0x0101: [w:GD62000] = 30006;
                 break;

    case 0x0102: [w:GD62000] = 30008;
                 break;

    case 0x0103: [w:GD62000] = 30008;
                 break;

    case 0x0104: [w:GD62000] = 30007;
                 break;

    case 0x0105: [w:GD62000] = 30007;
                 break;

    case 0x0106: [w:GD62000] = 30009;
                 break;

    case 0x0107: [w:GD62000] = 30009;
                 break;

    case 0x0400: [w:GD62000] = 30006;
                 break;

    case 0x0401: [w:GD62000] = 30006;
                 break;
  }
}

```

```

    case 0x0402: [w:GD62000] = 30008;
        break;

    case 0x0403: [w:GD62000] = 30008;
        break;

    case 0x0404: [w:GD62000] = 30007;
        break;

    case 0x0405: [w:GD62000] = 30007;
        break;

    case 0x0406: [w:GD62000] = 30009;
        break;

    case 0x0407: [w:GD62000] = 30009;
        break;

    case 0x0700: [w:GD62000] = 30006;
        break;

    case 0x0701: [w:GD62000] = 30006;
        break;

    case 0x0702: [w:GD62000] = 30008;
        break;

    case 0x0703: [w:GD62000] = 30008;
        break;

    case 0x0704: [w:GD62000] = 30007;
        break;

    case 0x0705: [w:GD62000] = 30007;
        break;

    case 0x0706: [w:GD62000] = 30009;
        break;

    case 0x0707: [w:GD62000] = 30009;
        break;
}
}

if([w:TMP0003] == 1)
{
    switch([w:GD10196[w:GD10228]])
    {
        case 0x1000: [w:GD62000] = 30010;
            break;

        case 0x1050: [w:GD62000] = 30010;
            break;

        case 0x1200: [w:GD62000] = 30011;
            break;

        case 0x1250: [w:GD62000] = 30011;
            break;

        case 0x1400: [w:GD62000] = 30012;
            break;
    }
}

```

```

    case 0x1401: [w:GD62000] = 30019;
        break;

    case 0x1450: [w:GD62000] = 30012;
        break;

    case 0x1451: [w:GD62000] = 30019;
        break;

    case 0x1500: [w:GD62000] = 30043;
        break;

    // case 0x1503: [w:GD62000] = 30046;
    //         break;

    case 0x1553: [w:GD62000] = 30050;
        break;
}
}

if([w:TMP0003] == 2)
{
    switch([w:GD10196[w:GD10228]])
    {

        case 0x2000: [w:GD62000] = 30013;
            break;

        case 0x2300: [w:GD62000] = 30014;
            break;

        case 0x2301: [w:GD62000] = 30014;
            break;

        case 0x2302: [w:GD62000] = 30015;
            break;

        case 0x2303: [w:GD62000] = 30016;
            break;

        case 0x2600: [w:GD62000] = 30017;
            break;

        case 0x2601: [w:GD62000] = 30017;
            break;

        case 0x2602: [w:GD62000] = 30017;
            break;

        case 0x2603: [w:GD62000] = 30017;
            break;

        case 0x2900: [w:GD62000] = 30020;
            break;

        case 0x2903: [w:GD62000] = 30021;
            break;

        case 0x2909: [w:GD62000] = 30023;
            break;

        case 0x290C: [w:GD62000] = 30053;
            break;
    }
}

```

```

        case 0x290F: [w:GD62000] = 30054;
            break;

        case 0x2930: [w:GD62000] = 30024;
            break;

        case 0x2933: [w:GD62000] = 30025;
            break;

        case 0x2939: [w:GD62000] = 30027;
            break;

        case 0x293C: [w:GD62000] = 30057;
            break;

        case 0x293F: [w:GD62000] = 30058;
            break;

        case 0x2960: [w:GD62000] = 30028;
            break;

        case 0x2963: [w:GD62000] = 30029;
            break;

        case 0x2969: [w:GD62000] = 30031;
            break;

        case 0x293C: [w:GD62000] = 30061;
            break;

        case 0x293F: [w:GD62000] = 30062;
            break;
    }
}

if([w:TMP0003] == 3)
{
    switch([w:GD10196[w:GD10228]])
    {
        case 0x3100: [w:GD62000] = 30032;
            break;

        case 0x3300: [w:GD62000] = 30033;
            break;

        case 0x3301: [w:GD62000] = 30034;
            break;

        case 0x3302: [w:GD62000] = 30035;
            break;

        case 0x3303: [w:GD62000] = 30035;
            break;

        case 0x3304: [w:GD62000] = 30036;
            break;

        case 0x3305: [w:GD62000] = 30036;
            break;
    }
}

if([w:TMP0003] == 10)

```



```

{
switch([w:GD10196[w:GD10228]])
{

case 0xA000: [w:GD62000] = 30018;
break;

case 0xA001: [w:GD62000] = 30018;
break;

case 0xA002: [w:GD62000] = 30037;
break;

case 0xA003: [w:GD62000] = 30037;
break;

case 0xA004: [w:GD62000] = 30038;
break;

case 0xA005: [w:GD62000] = 30038;
break;

case 0xA006: [w:GD62000] = 30039;
break;

case 0xA007: [w:GD62000] = 30039;
break;

case 0xA010: [w:GD62000] = 30051;
break;

case 0xA300: [w:GD62000] = 30018;
break;

case 0xA301: [w:GD62000] = 30018;
break;

case 0xA302: [w:GD62000] = 30037;
break;

case 0xA303: [w:GD62000] = 30037;
break;

case 0xA304: [w:GD62000] = 30038;
break;

case 0xA305: [w:GD62000] = 30038;
break;

case 0xA306: [w:GD62000] = 30039;
break;

case 0xA307: [w:GD62000] = 30039;
break;

case 0xA308: [w:GD62000] = 30040;
break;

case 0xA309: [w:GD62000] = 30040;
break;

case 0xA310: [w:GD62000] = 30052;
break;

```

```

case 0xA320: [w:GD62000] = 30041;
    break;

case 0xA321: [w:GD62000] = 30042;
    break;

case 0xA322: [w:GD62000] = 30042;
    break;

case 0xA323: [w:GD62000] = 30042;
    break;

case 0xA324: [w:GD62000] = 30042;
    break;

case 0xA325: [w:GD62000] = 30042;
    break;

case 0xA326: [w:GD62000] = 30042;
    break;

case 0xA600: [w:GD62000] = 30018;
    break;

case 0xA601: [w:GD62000] = 30018;
    break;

case 0xA602: [w:GD62000] = 30037;
    break;

case 0xA603: [w:GD62000] = 30037;
    break;

case 0xA604: [w:GD62000] = 30038;
    break;

case 0xA605: [w:GD62000] = 30038;
    break;

case 0xA606: [w:GD62000] = 30039;
    break;

case 0xA607: [w:GD62000] = 30039;
    break;
}
}

```

rst([b:GB30006]);

Script No.	30083	Script name	Script30083
Comment	Alarm Reset		
Data type	Signed BIN16	Trigger type	Rise, GB30049

//Turns on the alarm reset signal.

```

switch([w:GD64000])
{
    case Master_1: set([b:Y0000]);
        break;

    case Master_2: set([b:Y0020]);
        break;

    case Master_3: set([b:Y0040]);
        break;
}

```

```

case Master_4: set([b:Y0060]);
               break;

```

```

default:      break;

```

```

}
set([b:GB30050]);

```

```

rst([b:GB30049]);

```

Script No.	30084	Script name	Script30084
Comment	Alarm Clear Trigger Reset		
Data type	Signed BIN16	Trigger type	ON Sampling, 1 Sec, GB30050

```

//Turns off the alarm reset signal.

```

```

switch([w:GD64000])
{
  case Master_1: rst([b:Y0000]);
                 break;

  case Master_2: rst([b:Y0020]);
                 break;

  case Master_3: rst([b:Y0040]);
                 break;

  case Master_4: rst([b:Y0060]);
                 break;

  default:      break;
}

```

```

rst([b:GB30050]);

```

#### Base screen 30003

Script No.	30004	Script name	Script30004
Comment	B-30003,30004 Screen Initialization		
Data type	Signed BIN16	Trigger type	ON, GB40

```

f([b:GB30003] == OFF)
{

```

```

  fmov([w:TMP0000],[w:TMP0020],10);
  [w:TMP0110] = 0; //Initialize Temporary Area for Comparison
  [w:TMP0111] = 0; //Initialize Temporary Area for Comparison

```

```

  //Obtain Parameters and Display Parts

```

```

  switch([w:GD64000])
  {
    case Master_1: set([b:GD11000.b0]);
                   break;

    case Master_2: set([b:GD11008.b0]);
                   break;

    case Master_3: set([b:GD11016.b0]);
                   break;

    case Master_4: set([b:GD11024.b0]);
                   break;

    default:      break;
  }

```

<pre> set([b:GB30003]); //Turn on the flag not to work when displaying the screen for the second time or later. } </pre>			
Script No.	30008	Script name	Script30008
Comment	Script No.30003 Start		
Data type	Signed BIN16	Trigger type	Ordinary
<pre> //After reading the status, start Script No.30003. if([b:GB30016] == OFF) {     set([b:GB30001]);     set([b:GB30016]); } </pre>			
Script No.	30003	Script name	Script30003
Comment	B-30003,30004 Parts Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30001
<pre> //Display Parts [w:TMP0015] = 0; [w:TMP0008] = 0; fmov([w:TMP0008],[w:TMP0020],6);  while(([w:TMP0015] &lt; [w:GD20161[w:GD10601]]) &amp;&amp; ([w:GD20161[w:GD10601]] &gt; 0)) {     //Obtain Connection ID     [w:TMP0000] = [w:GD10601] + [w:TMP0015];     [w:TMP0020] = [w:GD20163[w:TMP0000]];     [w:TMP0015] = [w:TMP0015] + 1;      if(([w:TMP0020] &amp; 0x0200) == 0)     {         set([b:GB30600]); //Output Bit     }else{         set([b:GB30601]); //Input Bit     }      //Alarm Information     if([w:GD20291[w:GD10601]] != 0)     {         [w:TMP0004] = 0;         [w:TMP0001] = [w:GD10601] + [w:TMP0004];         while([w:TMP0004] &lt; [w:GD20291[w:GD10601]])         {             if([w:GD20292[w:TMP0001]] == [w:TMP0020])             {                 [w:TMP0022] = 1; //Alarm Occurrence                 break;             }             [w:TMP0004] = [w:TMP0004] + 1;             [w:TMP0001] = [w:TMP0001] + 1;         }     }      //Error Information     if([w:GD20032[w:GD10601]] != 0)     {         [w:TMP0004] = 0;         [w:TMP0001] = [w:GD10601] + [w:TMP0004];         while([w:TMP0004] &lt; [w:GD20032[w:GD10601]])         {             if([w:GD20033[w:TMP0001]] == [w:TMP0020])             {                 [w:TMP0023] = 1; //Error Occurrence                 break;             } </pre>			

```

    }
    [w:TMP0004] = [w:TMP0004] + 1;
    [w:TMP0001] = [w:TMP0001] + 1;
}
}

//Device Parameter
[w:TMP0004] = [w:TMP0020] & 0x00FF; //Offset to Obtain Device Parameter
[w:TMP0002] = [w:GD10601] + [w:TMP0004];
if([b:GB30600] == ON)
{
    //Output
    [w:TMP0005] = [w:GD20420[w:TMP0002]];
    //[w:TMP0012] = [w:TMP0004] + [w:GD10601];
    [w:TMP0012] = [w:TMP0004];
}else{
    //Input
    [w:TMP0005] = [w:GD20676[w:TMP0002]];
    //[w:TMP0012] = ([w:TMP0004] + 0x0100) + [w:GD10601];
    [w:TMP0012] = [w:TMP0004] + 0x0100;
}

[w:TMP0024] = ([w:TMP0005] - 0x3000) + [w:GD10602];

//I/O Points Pattern
//[w:TMP0006] = [w:TMP0024] + 43; //Parameter Offset
[w:TMP0007] = [w:GD24043[w:TMP0024]];
[w:TMP0025] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0026] = ([w:TMP0007] & 0x003F) + 1; //Points

//Module Model Number
//[w:TMP0006] = [w:TMP0024] + 44;
[w:GD16000[w:TMP0012]] = [w:GD24044[w:TMP0024]];

//ON/OFF Information
[w:TMP0008] = [w:TMP0020] & 0x00FF;
if([w:TMP0025] == 0) //Input Case
{
    [w:TMP0004] = 0;

    while([w:TMP0004] < [w:TMP0026])
    {
        [w:TMP0009] = [w:TMP0008] / 16;
        [w:TMP0010] = [w:TMP0008] % 16;

        if([w:TMP0023] == 1) //Check If Error Occurred
        {
            [w:GD12256[w:TMP0008]] = 3; //Error Occurrence & Input ON
        }else{
            if([w:TMP0022] == 1) //Check If Alarms Occurred
            {
                [w:GD12256[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
            }else{
                [w:GD12256[w:TMP0008]] = 1; //Normal
            }
        }
    }

    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}
}else{
    if([w:TMP0025] == 1) //Output Case
    {

```

```

[w:TMP0004] = 0;

while([w:TMP0004] < [w:TMP0026])
{
    [w:TMP0009] = [w:TMP0008] / 16;
    [w:TMP0010] = [w:TMP0008] % 16;

    if([w:TMP0023] == 1) //Check If Error Occurred
    {
        [w:GD12000[w:TMP0008]] = 3;    //Error Occurrence & Output ON
    }else{
        if([w:TMP0022] == 1) //Check If Alarms Occurred
        {
            [w:GD12000[w:TMP0008]] = 2;    //Alarm Occurrence & Output ON
        }else{
            [w:GD12000[w:TMP0008]] = 1;    //Normal
        }
    }

    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}
}else{
    if([w:TMP0025] == 2) //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:TMP0026] / 2; //If combined, points are reduced by half.
        while([w:TMP0004] < [w:TMP0011])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            //Output
            if([w:TMP0023] == 1) //Check If Error Occurred
            {
                [w:GD12000[w:TMP0008]] = 3;    //Error Occurrence & Output ON
            }else{
                if([w:TMP0022] == 1) //Check If Alarms Occurred
                {
                    [w:GD12000[w:TMP0008]] = 2;    //Alarm Occurrence & Output ON
                }else{
                    [w:GD12000[w:TMP0008]] = 1; //Normal
                }
            }
        }

        //Input
        if([w:TMP0023] == 1) //Check If Error Occurred
        {
            [w:GD12256[w:TMP0008]] = 3;    //Error Occurrence & Input ON
        }else{
            if([w:TMP0022] == 1) //Check If Alarms Occurred
            {
                [w:GD12256[w:TMP0008]] = 2;    //Alarm Occurrence & Input ON
            }else{
                [w:GD12256[w:TMP0008]] = 1;    //Normal
            }
        }

        [w:TMP0008] = [w:TMP0008] + 1;
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}
}
}
}

```

```

rst([b:GB30600]);
rst([b:GB30601]);

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004],[w:TMP0020],5);
}

//Display Cursor
if([w:GD62000] == 30004)
{
    //Output
    [w:GD10301] = 0; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303]; //Cursor Position Display
    //[w:TMP0030] = [w:GD10303] + [w:GD10601];

    if([w:GD12000[w:GD10303]] < 4)
    {
        [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
    }
}
else{
    //Input
    [w:GD10301] = 256; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303] - 256; //Cursor Position Display
    //[w:TMP0030] = [w:GD10303] + [w:GD10601];

    if([w:GD12000[w:GD10303]] < 4)
    {
        [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
    }
}

rst([b:GB30001]);

//Initial Action Check Flag.
set([b:GB30035]);

```

Script No.	30021	Script name	Script30021
Comment	Flags Clear		
Data type	Signed BIN16	Trigger type	When closing a screen

```

//Clear Flags
rst([b:GB30006]);
rst([b:GB30007]);
rst([b:GB30003]);
rst([b:GB30018]);
rst([b:GB30035]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD12000],512); //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004],[w:GD10301],3); //Clear Cursor Position Information(B-30003,4)
fmov([w:TMP0004],[w:GD10432],15); //Work Area Clear
[w:GD10300] = 0; //Cursor Position Information Clear(B-30003,4)

//Close Overlap Window
[w:GD62001] = 0;
[w:GD62004] = 0;
[w:GD62007] = 0;

```

Script No.	30017	Script name	Script30017
Comment	B-30003,30004 Detail Screen		
Data type	Unsigned BIN16	Trigger type	ON, GB30002

```

//Display Detail Information Screen of Cursor Position

//Obtain Temporary ID
if([w:GD10303] < 256)
{
    [w:TMP0000] = [w:GD10303]; //Output ID
}else{
    [w:TMP0000] = [w:GD10303] - 256; //Input ID
}

if([w:GD12000[w:GD10303]] > 4) //Judge whether a module exists in the cursor position.
{
    if([w:GD10303] < 256)
    {
        //Output
        [w:TMP0002] = [w:TMP0000];
        while([w:TMP0002] >= 0)
        {
            if([w:GD12000[w:TMP0002]] != 0)
            {
                //Output
                [w:TMP0001] = 0;
                [w:TMP0005] = [w:GD10601];
                while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID Counts
                {
                    if([w:GD20163[w:TMP0005]] == [w:TMP0002]) //Judge ID
                    {
                        set([b:GB30009]); //ID Match Flag
                        break;
                    }else{
                        [w:TMP0001] = [w:TMP0001] + 1;
                        [w:TMP0005] = [w:TMP0005] + 1;
                    }
                }
            }

            //Combined Output Side Case
            if([b:GB30009] == OFF)
            {
                [w:TMP0001] = 0;
                [w:TMP0005] = [w:GD10601];
                while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID
                Counts
                {
                    if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
                    {
                        //Judge whether the matched ID is that of a combined module or not.
                        [w:TMP0006] = [w:GD10601] + [w:TMP0002];
                        [w:TMP0010] = [w:GD20676[w:TMP0006]]; //Parameter Top Address
                        [w:TMP0011] = ([w:TMP0010] - 0x3000) + [w:GD10602];

                        [w:TMP0012] = [w:GD24043[w:TMP0011]];
                        if(((w:TMP0012] & 0x00C0) >> 6) == 2)
                        {
                            set([b:GB30009]); //ID Match Flag
                            set([b:GB30020]); //Combined Output Flag
                            break;
                        }
                        [w:TMP0001] = [w:TMP0001] + 1;
                        [w:TMP0005] = [w:TMP0005] + 1;
                    }else{
                        [w:TMP0001] = [w:TMP0001] + 1;
                        [w:TMP0005] = [w:TMP0005] + 1;
                    }
                }
            }
        }
    }
}

```



```

    }
}

if([b:GB30009] == ON) //If matched, exit loop.
{
    break;
}
}else{
    //If No Connected Devices
    break;
}

[w:TMP0002] = [w:TMP0002] - 1;
}

}else{
    //Input
    [w:TMP0002] = [w:TMP0000];
    while([w:TMP0002] >= 0)
    {
        [w:TMP0001] = 0;
        [w:TMP0005] = [w:GD10601];
        while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID Counts
        {
            if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
            {
                set([b:GB30009]); //ID Match Flag
                break;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }
        }
    }

    if([b:GB30009] == ON) //If matched, exit loop.
    {
        break;
    }

    [w:TMP0002] = [w:TMP0002] - 1;
}
}

//If matched, display screen.
if([b:GB30009] == ON)
{
    if([w:GD10303] < 256)
    {
        [w:TMP0007] = [w:GD10601] + [w:TMP0002];

        if([b:GB30020] == OFF)
        {
            //Output
            [w:GD10600] = [w:GD20420[w:TMP0007]]; //Offset
            [w:TMP0003] = [w:TMP0002];
            [w:TMP0004] = [w:TMP0002] + [w:GD10601];
            [w:TMP0004] = [w:TMP0002];
        }else{
            //If combined, refer to the input side.
            [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
            [w:TMP0003] = [w:TMP0002] + 0x0200;
        }
    }
}

```

```

        //[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
        [w:TMP0004] = [w:TMP0002] + 0x0100;
    }
} else {
    //Input
    [w:TMP0007] = [w:GD10601] + [w:TMP0002];
    [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
    [w:TMP0003] = [w:TMP0002] + 0x0200;
    //[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
    [w:TMP0004] = [w:TMP0002] + 0x0100;
}

```

```

[w:U00-G10320] = 0; //Read Individual Parameter
[w:U00-G10321] = [w:TMP0003]; //Read Target ID

```

```

//Select base screen to display.
switch([w:GD16000[w:TMP0004]] & 0xF000)
{
    case 0x0000: [w:TMP0008] = 0;
                  break;

    case 0x1000: [w:TMP0008] = 1;
                  break;

    case 0x2000: [w:TMP0008] = 2;
                  break;

    case 0x3000: [w:TMP0008] = 3;
                  break;

    case 0xA000: [w:TMP0008] = 10;
                  break;
}

```

```

if([w:TMP0008] == 0)
{
    switch([w:GD16000[w:TMP0004]])
    {
        case 0x0100: [w:GD62000] = 30006;
                      break;

        case 0x0101: [w:GD62000] = 30006;
                      break;

        case 0x0102: [w:GD62000] = 30008;
                      break;

        case 0x0103: [w:GD62000] = 30008;
                      break;

        case 0x0104: [w:GD62000] = 30007;
                      break;

        case 0x0105: [w:GD62000] = 30007;
                      break;

        case 0x0106: [w:GD62000] = 30009;
                      break;

        case 0x0107: [w:GD62000] = 30009;
                      break;

        case 0x0400: [w:GD62000] = 30006;

```

```

        break;

    case 0x0401: [w:GD62000] = 30006;
        break;

    case 0x0402: [w:GD62000] = 30008;
        break;

    case 0x0403: [w:GD62000] = 30008;
        break;

    case 0x0404: [w:GD62000] = 30007;
        break;

    case 0x0405: [w:GD62000] = 30007;
        break;

    case 0x0406: [w:GD62000] = 30009;
        break;

    case 0x0407: [w:GD62000] = 30009;
        break;

    case 0x0700: [w:GD62000] = 30006;
        break;

    case 0x0701: [w:GD62000] = 30006;
        break;

    case 0x0702: [w:GD62000] = 30008;
        break;

    case 0x0703: [w:GD62000] = 30008;
        break;

    case 0x0704: [w:GD62000] = 30007;
        break;

    case 0x0705: [w:GD62000] = 30007;
        break;

    case 0x0706: [w:GD62000] = 30009;
        break;

    case 0x0707: [w:GD62000] = 30009;
        break;
    }
}

if([w:TMP0008] == 1)
{
    switch([w:GD16000[w:TMP0004]])
    {
        case 0x1000: [w:GD62000] = 30010;
            break;

        case 0x1050: [w:GD62000] = 30010;
            break;

        case 0x1200: [w:GD62000] = 30011;
            break;

        case 0x1250: [w:GD62000] = 30011;
            break;
    }
}

```

```

        case 0x1400: [w:GD62000] = 30012;
                    break;

        case 0x1401: [w:GD62000] = 30019;
                    break;

        case 0x1450: [w:GD62000] = 30012;
                    break;

        case 0x1451: [w:GD62000] = 30019;
                    break;

        case 0x1500: [w:GD62000] = 30043;
                    break;

        // case 0x1503: [w:GD62000] = 30046;
        //             break;

        case 0x1553: [w:GD62000] = 30050;
                    break;
    }
}

if([w:TMP0008] == 2)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0x2000: [w:GD62000] = 30013;
                    break;

        case 0x2300: [w:GD62000] = 30014;
                    break;

        case 0x2301: [w:GD62000] = 30014;
                    break;

        case 0x2302: [w:GD62000] = 30015;
                    break;

        case 0x2303: [w:GD62000] = 30016;
                    break;

        case 0x2600: [w:GD62000] = 30017;
                    break;

        case 0x2601: [w:GD62000] = 30017;
                    break;

        case 0x2602: [w:GD62000] = 30017;
                    break;

        case 0x2603: [w:GD62000] = 30017;
                    break;

        case 0x2900: [w:GD62000] = 30020;
                    break;

        case 0x2903: [w:GD62000] = 30021;
                    break;

        case 0x2909: [w:GD62000] = 30023;
                    break;
    }
}

```

```

        case 0x290C: [w:GD62000] = 30053;
                    break;

        case 0x290F: [w:GD62000] = 30054;
                    break;

        case 0x2930: [w:GD62000] = 30024;
                    break;

        case 0x2933: [w:GD62000] = 30025;
                    break;

        case 0x2939: [w:GD62000] = 30027;
                    break;

        case 0x293C: [w:GD62000] = 30057;
                    break;

        case 0x293F: [w:GD62000] = 30058;
                    break;

        case 0x2960: [w:GD62000] = 30028;
                    break;

        case 0x2963: [w:GD62000] = 30029;
                    break;

        case 0x2969: [w:GD62000] = 30031;
                    break;

        case 0x293C: [w:GD62000] = 30061;
                    break;

        case 0x293F: [w:GD62000] = 30062;
                    break;
    }
}

if([w:TMP0008] == 3)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0x3100: [w:GD62000] = 30032;
                    break;

        case 0x3300: [w:GD62000] = 30033;
                    break;

        case 0x3301: [w:GD62000] = 30034;
                    break;

        case 0x3302: [w:GD62000] = 30035;
                    break;

        case 0x3303: [w:GD62000] = 30035;
                    break;

        case 0x3304: [w:GD62000] = 30036;
                    break;

        case 0x3305: [w:GD62000] = 30036;
                    break;
    }
}

```

```

    }
}

if([w:TMP0008] == 10)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0xA000: [w:GD62000] = 30018;
                     break;

        case 0xA001: [w:GD62000] = 30018;
                     break;

        case 0xA002: [w:GD62000] = 30037;
                     break;

        case 0xA003: [w:GD62000] = 30037;
                     break;

        case 0xA004: [w:GD62000] = 30038;
                     break;

        case 0xA005: [w:GD62000] = 30038;
                     break;

        case 0xA006: [w:GD62000] = 30039;
                     break;

        case 0xA007: [w:GD62000] = 30039;
                     break;

        case 0xA010: [w:GD62000] = 30051;
                     break;

        case 0xA300: [w:GD62000] = 30018;
                     break;

        case 0xA301: [w:GD62000] = 30018;
                     break;

        case 0xA302: [w:GD62000] = 30037;
                     break;

        case 0xA303: [w:GD62000] = 30037;
                     break;

        case 0xA304: [w:GD62000] = 30038;
                     break;

        case 0xA305: [w:GD62000] = 30038;
                     break;

        case 0xA306: [w:GD62000] = 30039;
                     break;

        case 0xA307: [w:GD62000] = 30039;
                     break;

        case 0xA308: [w:GD62000] = 30040;
                     break;

        case 0xA309: [w:GD62000] = 30040;
                     break;
    }
}

```

```

        case 0xA310: [w:GD62000] = 30052;
                    break;

        case 0xA320: [w:GD62000] = 30041;
                    break;

        case 0xA321: [w:GD62000] = 30042;
                    break;

        case 0xA322: [w:GD62000] = 30042;
                    break;

        case 0xA323: [w:GD62000] = 30042;
                    break;

        case 0xA324: [w:GD62000] = 30042;
                    break;

        case 0xA325: [w:GD62000] = 30042;
                    break;

        case 0xA326: [w:GD62000] = 30042;
                    break;

        case 0xA600: [w:GD62000] = 30018;
                    break;

        case 0xA601: [w:GD62000] = 30018;
                    break;

        case 0xA602: [w:GD62000] = 30037;
                    break;

        case 0xA603: [w:GD62000] = 30037;
                    break;

        case 0xA604: [w:GD62000] = 30038;
                    break;

        case 0xA605: [w:GD62000] = 30038;
                    break;

        case 0xA606: [w:GD62000] = 30039;
                    break;

        case 0xA607: [w:GD62000] = 30039;
                    break;
    }
}
//Flag Reset
rst([b:GB30009]);
rst([b:GB30020]);
}

```

```
rst([b:GB30002]);
```

Script No.	30012	Script name	Script30012
Comment	Coordinate Calculation(Input) *1		
Data type	Signed BIN16	Trigger type	Ordinary

```
//Determine the cursor position from the touched coordinate.
```

```
if((([w:TMP0120] != [w:GS654]) || ([w:TMP0121] != [w:GS655])) && ([w:GS654] > Input_X) && ([w:GS655] > Input_Y))
```

```

&& ([w:GS654] < Frame_X) && ([w:GS655] < Frame_Y)))
{
[w:TMP0120] = [w:GS654];
[w:TMP0121] = [w:GS655];

if([b:GD11000.b0] == OFF) && ([b:GB30001] == OFF)
{
//Save Current Cursor Position
[w:TMP0122] = [w:GD10303];
//[w:TMP0127] = [w:TMP0122] + [w:GD10601];

//X Coordinate
//Change values to subtract depending on whether input or output.
//Input
[w:TMP0123] = [w:TMP0120] - Input_X;
[w:TMP0124] = [w:TMP0123] / Object_X;
[w:GD10301] = (16 - ([w:TMP0124] + 1)) + 256;

//Y Coordinate
[w:TMP0125] = [w:TMP0121] - Input_Y;
[w:TMP0126] = [w:TMP0125] / Object_Y;
[w:GD10302] = [w:TMP0126] * 16;

//Offset
[w:GD10303] = [w:GD10301] + [w:GD10302];
//[w:TMP0128] = [w:GD10303] + [w:GD10601];

//If the offset position does not change, displayed parts do not change.
if([w:TMP0122] != [w:GD10303])
{
//Cursor Display
if([w:GD12000[w:GD10303]] < 4)
{
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
}

//Cursor Delete
if([w:GD12000[w:TMP0122]] < 4)
{
[w:GD12000[w:TMP0122]] = 0;
}else{
[w:GD12000[w:TMP0122]] = [w:GD12000[w:TMP0122]] - 4;
}
}
}

[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display
}

```

Script No.	30013	Script name	Script30013
Comment	Cursor Display(To Left)(Input)		
Data type	Signed BIN16	Trigger type	Rise, GB30010

```

//Control Parts Display

//Calculate Cursor Position
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];

if([w:GD10301] == 271)
{
[w:GD10301] = 256;
}else{
[w:GD10301] = [w:GD10301] + 1;
}

```



[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset  
//[w:TMP0022] = [w:GD10303] + [w:GD10601];

if([w:GD12000[w:TMP0020]] >= 4)  
{  
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;  
}else{  
    [w:GD12000[w:TMP0020]] = 0;  
}  
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display

Script No.	30018	Script name	Script30018
Comment	Cursor Display(To Right)(Input)		
Data type	Signed BIN16	Trigger type	Rise, GB30011

//Control Parts Display

//Calculate Cursor Position

[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position  
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];

if([w:GD10301] == 256)  
{  
    [w:GD10301] = 271;  
}else{  
    [w:GD10301] = [w:GD10301] -1;  
}

[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset  
//[w:TMP0022] = [w:GD10303] + [w:GD10601];

if([w:GD12000[w:TMP0020]] >= 4)  
{  
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;  
}else{  
    [w:GD12000[w:TMP0020]] = 0;  
}  
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display

Script No.	30022	Script name	Script30022
Comment	Cursor Display(Down)(Input)		
Data type	Signed BIN16	Trigger type	Rise, GB30012

//Control Parts Display

//Calculate Cursor Position

[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position  
//[w:TMP0021] = [w:TMP0020] + [w:GD10601];

if([w:GD10302] == 240)  
{  
    [w:GD10302] = 0;  
}else{  
    [w:GD10302] = [w:GD10302] + 16;  
}

[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset  
//[w:TMP0022] = [w:GD10303] + [w:GD10601];

```

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

```

[w:GD10300] = [w:GD10303] - 256; //Cursor Position Display

Script No.	30036	Script name	Script30036
Comment	Cursor Display(Up)(Input)		
Data type	Signed BIN16	Trigger type	Rise, GB30013

//Control Parts Display

//Calculate Cursor Position

[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position

//[w:TMP0021] = [w:TMP0020] + [w:GD10601];

```

if([w:GD10302] == 0)
{
    [w:GD10302] = 240;
}else{
    [w:GD10302] = [w:GD10302] - 16;
}

```

[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset

//[w:TMP0022] = [w:GD10303] + [w:GD10601];

```

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

```

[w:GD10300] = [w:GD10303] - 256; //Reduce Cursor Position Display Value.

Script No.	30083	Script name	Script30083
Comment	Alarm Reset		
Data type	Signed BIN16	Trigger type	Rise, GB30049

//Turn ON the alarm reset signal.

```

switch([w:GD64000])
{
    case Master_1: set([b:Y0000]);
                    break;

    case Master_2: set([b:Y0020]);
                    break;

    case Master_3: set([b:Y0040]);
                    break;

    case Master_4: set([b:Y0060]);
                    break;

    default:      break;
}
set([b:GB30050]);
rst([b:GB30049]);

```

Script No.	30084	Script name	Script30084
Comment	Alarm Clear Trigger Reset		
Data type	Signed BIN16	Trigger type	ON Sampling, 1 Sec, GB30050
<pre>//Turn OFF the alarm reset signal.  switch([w:GD64000]) {     case Master_1: rst([b:Y0000]);                     break;      case Master_2: rst([b:Y0020]);                     break;      case Master_3: rst([b:Y0040]);                     break;      case Master_4: rst([b:Y0060]);                     break;      default:       break; }  rst([b:GB30050]);</pre>			

\*1: [Script Symbol] is used. For more details about [Script Symbol], please refer to "5.9.4 Script Symbol".

#### Base screen 30004

Script No.	30004	Script name	Script30004
Comment	B-30003,30004 Screen Initialization		
Data type	Signed BIN16	Trigger type	ON, GB40
<pre>if([b:GB30003] == OFF) {      fmov([w:TMP0000],[w:TMP0020],10);     [w:TMP0110] = 0; //Initialize Temporary Area for Comparison     [w:TMP0111] = 0; //Initialize Temporary Area for Comparison      //Obtain Parameters and Display Parts     switch([w:GD64000])     {         case Master_1: set([b:GD11000.b0]);                         break;          case Master_2: set([b:GD11008.b0]);                         break;          case Master_3: set([b:GD11016.b0]);                         break;          case Master_4: set([b:GD11024.b0]);                         break;          default:       break;     }      set([b:GB30003]); //Turn on the flag not to work when displaying the screen for the second time or later. }</pre>			
Script No.	30008	Script name	Script30008
Comment	Script No.30003 Start		
Data type	Signed BIN16	Trigger type	Ordinary
<pre>//After reading the status, start Script No.30003. if([b:GB30016] == OFF) {</pre>			

<pre> set([b:GB30001]); set([b:GB30016]); } </pre>			
Script No.	30003	Script name	Script30003
Comment	B-30003,30004 Parts Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30001
<pre> //Display Parts [w:TMP0015] = 0; [w:TMP0008] = 0; fmov([w:TMP0008],[w:TMP0020],6);  while(([w:TMP0015] &lt; [w:GD20161[w:GD10601]]) &amp;&amp; ([w:GD20161[w:GD10601]] &gt; 0)) {     //Obtain Connection ID     [w:TMP0000] = [w:GD10601] + [w:TMP0015];     [w:TMP0020] = [w:GD20163[w:TMP0000]];     [w:TMP0015] = [w:TMP0015] + 1;      if(([w:TMP0020] &amp; 0x0200) == 0)     {         set([b:GB30600]); //Output Bit     }else{         set([b:GB30601]); //Input Bit     }      //Alarm Information     if([w:GD20291[w:GD10601]] != 0)     {         [w:TMP0004] = 0;         [w:TMP0001] = [w:GD10601] + [w:TMP0004];         while([w:TMP0004] &lt; [w:GD20291[w:GD10601]])         {             if([w:GD20292[w:TMP0001]] == [w:TMP0020])             {                 [w:TMP0022] = 1; //Alarm Occurrence                 break;             }             [w:TMP0004] = [w:TMP0004] + 1;             [w:TMP0001] = [w:TMP0001] + 1;         }     } }  //Error Information if([w:GD20032[w:GD10601]] != 0) {     [w:TMP0004] = 0;     [w:TMP0001] = [w:GD10601] + [w:TMP0004];     while([w:TMP0004] &lt; [w:GD20032[w:GD10601]])     {         if([w:GD20033[w:TMP0001]] == [w:TMP0020])         {             [w:TMP0023] = 1; //Error Occurrence             break;         }         [w:TMP0004] = [w:TMP0004] + 1;         [w:TMP0001] = [w:TMP0001] + 1;     } }  //Device Parameter [w:TMP0004] = [w:TMP0020] &amp; 0x00FF; //Offset to Obtain Device Parameter [w:TMP0002] = [w:GD10601] + [w:TMP0004]; if([b:GB30600] == ON) { </pre>			

```

//Output
[w:TMP0005] = [w:GD20420[w:TMP0002]];
//[w:TMP0012] = [w:TMP0004] + [w:GD10601];
[w:TMP0012] = [w:TMP0004];
}else{
//Input
[w:TMP0005] = [w:GD20676[w:TMP0002]];
//[w:TMP0012] = ([w:TMP0004] + 0x0100) + [w:GD10601];
[w:TMP0012] = [w:TMP0004] + 0x0100;
}

[w:TMP0024] = ([w:TMP0005] - 0x3000) + [w:GD10602];

//I/O Points Pattern
//[w:TMP0006] = [w:TMP0024] + 43; //Parameter Offset
[w:TMP0007] = [w:GD24043[w:TMP0024]];
[w:TMP0025] = ([w:TMP0007] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0026] = ([w:TMP0007] & 0x003F) + 1; //Points

//Module Model Number
//[w:TMP0006] = [w:TMP0024] + 44;
[w:GD16000[w:TMP0012]] = [w:GD24044[w:TMP0024]];

//ON/OFF Information
[w:TMP0008] = [w:TMP0020] & 0x00FF;
if([w:TMP0025] == 0) //Input Case
{
[w:TMP0004] = 0;

while([w:TMP0004] < [w:TMP0026])
{
[w:TMP0009] = [w:TMP0008] / 16;
[w:TMP0010] = [w:TMP0008] % 16;

if([w:TMP0023] == 1) //Check If Error Occurred
{
[w:GD12256[w:TMP0008]] = 3; //Error Occurrence & Input ON
}else{
if([w:TMP0022] == 1) //Check If Alarms Occurred
{
[w:GD12256[w:TMP0008]] = 2; //Alarm Occurrence & Input ON
}else{
[w:GD12256[w:TMP0008]] = 1; //Normal
}
}

[w:TMP0008] = [w:TMP0008] + 1;
[w:TMP0004] = [w:TMP0004] + 1;
}
}else{
if([w:TMP0025] == 1) //Output Case
{
[w:TMP0004] = 0;

while([w:TMP0004] < [w:TMP0026])
{
[w:TMP0009] = [w:TMP0008] / 16;
[w:TMP0010] = [w:TMP0008] % 16;

if([w:TMP0023] == 1) //Check If Error Occurred
{
[w:GD12000[w:TMP0008]] = 3; //Error Occurrence & Output ON
}else{
if([w:TMP0022] == 1) //Check If Alarms Occurred

```

```

        {
            [w:GD12000[w:TMP0008]] = 2;    //Alarm Occurrence & Output ON
        }else{
            [w:GD12000[w:TMP0008]] = 1; //Normal
        }
    }

    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}
}else{
    if([w:TMP0025] == 2) //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:TMP0026] / 2; //If combined, points are reduced by half.
        while([w:TMP0004] < [w:TMP0011])
        {
            [w:TMP0009] = [w:TMP0008] / 16;
            [w:TMP0010] = [w:TMP0008] % 16;

            //Output
            if([w:TMP0023] == 1) //Check If Error Occurred
            {
                [w:GD12000[w:TMP0008]] = 3;    //Error Occurrence & Output ON
            }else{
                if([w:TMP0022] == 1) //Check If Alarms Occurred
                {
                    [w:GD12000[w:TMP0008]] = 2;    //Alarm Occurrence & Output ON
                }else{
                    [w:GD12000[w:TMP0008]] = 1; //Normal
                }
            }

            //Input
            if([w:TMP0023] == 1) //Check If Error Occurred
            {
                [w:GD12256[w:TMP0008]] = 3;    //Error Occurrence & Input ON
            }else{
                if([w:TMP0022] == 1) //Check If Alarms Occurred
                {
                    [w:GD12256[w:TMP0008]] = 2;    //Alarm Occurrence & Input ON
                }else{
                    [w:GD12256[w:TMP0008]] = 1;    //Normal
                }
            }

            [w:TMP0008] = [w:TMP0008] + 1;
            [w:TMP0004] = [w:TMP0004] + 1;
        }
    }
}
}
}
rst([b:GB30600]);
rst([b:GB30601]);

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004],[w:TMP0020],5);
}

//Display Cursor
if([w:GD62000] == 30004)
{

```

```

//Output
[w:GD10301] = 0; //Left Right
[w:GD10302] = 0; //Up Down
[w:GD10303] = [w:GD10301] + [w:GD10302];
[w:GD10300] = [w:GD10303]; //Cursor Position Display
//[w:TMP0030] = [w:GD10303] + [w:GD10601];

if([w:GD12000[w:GD10303]] < 4)
{
    [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
}
}else{
    //Input
    [w:GD10301] = 256; //Left Right
    [w:GD10302] = 0; //Up Down
    [w:GD10303] = [w:GD10301] + [w:GD10302];
    [w:GD10300] = [w:GD10303] - 256; //Cursor Position Display
    //[w:TMP0030] = [w:GD10303] + [w:GD10601];

    if([w:GD12000[w:GD10303]] < 4)
    {
        [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
    }
}

rst([b:GB30001]);

```

//Initial Action Check Flag.  
set([b:GB30035]);

Script No.	30001	Script name	Script30001
Comment	Cursor Display(To Left)		
Data type	Signed BIN16	Trigger type	Rise, GB30010

//Control Parts Display

//Calculate Cursor Position  
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position

```

if([w:GD10301] == 15)
{
    [w:GD10301] = 0;
}else{
    [w:GD10301] = [w:GD10301] + 1;
}

```

[w:TMP0000] = [w:GD10301];

[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset

```

if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;

```

[w:GD10300] = [w:GD10303]; //Cursor Position Display

Script No.	30030	Script name	Script30030
Comment	Cursor Display(To Right)		
Data type	Signed BIN16	Trigger type	Rise, GB30011

//Control Parts Display

//Calculate Cursor Position

```
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
```

```
if([w:GD10301] == 0)
{
    [w:GD10301] = 15;
}else{
    [w:GD10301] = [w:GD10301] -1;
}
```

```
[w:TMP0000] = [w:GD10301];
```

```
[w:GD10303] = [w:TMP0000] + [w:GD10302]; //Offset
```

```
if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
```

Script No.	30031	Script name	Script30031
Comment	Cursor Display(Down)		
Data type	Signed BIN16	Trigger type	Rise, GB30012

```
//Control Parts Display
```

```
//Calculate Cursor Position
```

```
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
```

```
if([w:GD10302] == 240)
{
    [w:GD10302] = 0;
}else{
    [w:GD10302] = [w:GD10302] + 16;
}
```

```
[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset
```

```
if([w:GD12000[w:TMP0020]] >= 4)
{
    [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4;
}else{
    [w:GD12000[w:TMP0020]] = 0;
}
[w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
```

```
[w:GD10300] = [w:GD10303]; //Cursor Position Display
```

Script No.	30032	Script name	Script30032
Comment	Cursor Display(Up)		
Data type	Signed BIN16	Trigger type	Rise, GB30013

```
//Control Parts Display
```

```
//Calculate Cursor Position
```

```
[w:TMP0020] = [w:GD10303]; //Evacuate Previous Cursor Position
```

```
if([w:GD10302] == 0)
{
    [w:GD10302] = 240;
}else{
    [w:GD10302] = [w:GD10302] - 16;
}
```

```
[w:GD10303] = [w:GD10301] + [w:GD10302]; //Offset
```



<pre> if([w:GD12000[w:TMP0020]] &gt;= 4) {     [w:GD12000[w:TMP0020]] = [w:GD12000[w:TMP0020]] - 4; }else{     [w:GD12000[w:TMP0020]] = 0; } [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4; </pre>			
Script No.	30021	Script name	Script30021
Comment	Flags Clear		
Data type	Signed BIN16	Trigger type	When closing a screen
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]); rst([b:GB30018]); rst([b:GB30035]);  [w:TMP0004] = 0; fmov([w:TMP0004],[w:GD12000],512); //Initialize Object Values(B-30003, 4) fmov([w:TMP0004],[w:GD10301],3); //Clear Cursor Position Information(B-30003,4) fmov([w:TMP0004],[w:GD10432],15); //Work Area Clear [w:GD10300] = 0; //Cursor Position Information Clear(B-30003,4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
Script No.	30017	Script name	Script30017
Comment	B-30003,30004 Detail Screen		
Data type	Unsigned BIN16	Trigger type	ON, GB30002
<pre> //Display Detail Information Screen of Cursor Position  //Obtain Temporary ID if([w:GD10303] &lt; 256) {     [w:TMP0000] = [w:GD10303]; //Output ID }else{     [w:TMP0000] = [w:GD10303] - 256; //Input ID }  if([w:GD12000[w:GD10303]] &gt; 4) //Judge whether a module exists in the cursor position. {     if([w:GD10303] &lt; 256)     {         //Output         [w:TMP0002] = [w:TMP0000];         while([w:TMP0002] &gt;= 0)         {             if([w:GD12000[w:TMP0002]] != 0)             {                 //Output                 [w:TMP0001] = 0;                 [w:TMP0005] = [w:GD10601];                 while([w:TMP0001] &lt; [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID Counts                 {                     if([w:GD20163[w:TMP0005]] == [w:TMP0002]) //Judge ID                     {                         set([b:GB30009]); //ID Match Flag                         break;                     }else{                         [w:TMP0001] = [w:TMP0001] + 1;                         [w:TMP0005] = [w:TMP0005] + 1;                     }                 }             }         }     } } </pre>			

```

    }

    //Combined Output Side Case
    if([b:GB30009] == OFF)
    {
        [w:TMP0001] = 0;
        [w:TMP0005] = [w:GD10601];
        while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID
Counts
        {
            if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
            {

                //Judge whether the matched ID is that of a combined module or not.
                [w:TMP0006] = [w:GD10601] + [w:TMP0002];
                [w:TMP0010] = [w:GD20676[w:TMP0006]]; //Parameter Top Address
                [w:TMP0011] = ([w:TMP0010] - 0x3000) + [w:GD10602];

                [w:TMP0012] = [w:GD24043[w:TMP0011]];
                if(((w:TMP0012] & 0x00C0) >> 6) == 2)
                {
                    set([b:GB30009]); //ID Match Flag
                    set([b:GB30020]); //Combined Output Flag
                    break;
                }
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }
        }
    }

    if([b:GB30009] == ON) //If matched, exit loop.
    {
        break;
    }
}else{
    //If No Connected Devices
    break;
}

[w:TMP0002] = [w:TMP0002] - 1;
}

}else{
    //Input
    [w:TMP0002] = [w:TMP0000];
    while([w:TMP0002] >= 0)
    {
        [w:TMP0001] = 0;
        [w:TMP0005] = [w:GD10601];
        while([w:TMP0001] < [w:GD20162[w:GD10601]]) //Repeat the Number of Connection ID Counts
        {
            if([w:GD20163[w:TMP0005]] == ([w:TMP0002] + 0x0200)) //Judge ID
            {
                set([b:GB30009]); //ID Match Flag
                break;
            }else{
                [w:TMP0001] = [w:TMP0001] + 1;
                [w:TMP0005] = [w:TMP0005] + 1;
            }
        }
    }
}

```

```

    }
}

if([b:GB30009] == ON)    //If matched, exit loop.
{
    break;
}

[w:TMP0002] = [w:TMP0002] - 1;
}
}

//If matched, display screen.
if([b:GB30009] == ON)
{
    if([w:GD10303] < 256)
    {
        [w:TMP0007] = [w:GD10601] + [w:TMP0002];

        if([b:GB30020] == OFF)
        {
            //Output
            [w:GD10600] = [w:GD20420[w:TMP0007]]; //Offset
            [w:TMP0003] = [w:TMP0002];
            //[w:TMP0004] = [w:TMP0002] + [w:GD10601];
            [w:TMP0004] = [w:TMP0002];
        }else{
            //If combined, refer to the input side.
            [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
            [w:TMP0003] = [w:TMP0002] + 0x0200;
            //[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
            [w:TMP0004] = [w:TMP0002] + 0x0100;
        }
    }else{
        //Input
        [w:TMP0007] = [w:GD10601] + [w:TMP0002];
        [w:GD10600] = [w:GD20676[w:TMP0007]]; //Offset
        [w:TMP0003] = [w:TMP0002] + 0x0200;
        //[w:TMP0004] = ([w:TMP0002] + 0x0100) + [w:GD10601];
        [w:TMP0004] = [w:TMP0002] + 0x0100;
    }

    [w:U00-G10320] = 0; //Read Individual Parameter
    [w:U00-G10321] = [w:TMP0003]; //Read Target ID

    //Select base screen to display.
    switch([w:GD16000[w:TMP0004]] & 0xF000)
    {
        case 0x0000: [w:TMP0008] = 0;
                     break;

        case 0x1000: [w:TMP0008] = 1;
                     break;

        case 0x2000: [w:TMP0008] = 2;
                     break;

        case 0x3000: [w:TMP0008] = 3;
                     break;

        case 0xA000: [w:TMP0008] = 10;
                     break;
    }
}

```

```

}

if([w:TMP0008] == 0)
{
    switch([w:GD16000[w:TMP0004]])
    {
        case 0x0100: [w:GD62000] = 30006;
                     break;

        case 0x0101: [w:GD62000] = 30006;
                     break;

        case 0x0102: [w:GD62000] = 30008;
                     break;

        case 0x0103: [w:GD62000] = 30008;
                     break;

        case 0x0104: [w:GD62000] = 30007;
                     break;

        case 0x0105: [w:GD62000] = 30007;
                     break;

        case 0x0106: [w:GD62000] = 30009;
                     break;

        case 0x0107: [w:GD62000] = 30009;
                     break;

        case 0x0400: [w:GD62000] = 30006;
                     break;

        case 0x0401: [w:GD62000] = 30006;
                     break;

        case 0x0402: [w:GD62000] = 30008;
                     break;

        case 0x0403: [w:GD62000] = 30008;
                     break;

        case 0x0404: [w:GD62000] = 30007;
                     break;

        case 0x0405: [w:GD62000] = 30007;
                     break;

        case 0x0406: [w:GD62000] = 30009;
                     break;

        case 0x0407: [w:GD62000] = 30009;
                     break;

        case 0x0700: [w:GD62000] = 30006;
                     break;

        case 0x0701: [w:GD62000] = 30006;
                     break;

        case 0x0702: [w:GD62000] = 30008;
                     break;

        case 0x0703: [w:GD62000] = 30008;
    }
}

```

```

        break;

    case 0x0704: [w:GD62000] = 30007;
        break;

    case 0x0705: [w:GD62000] = 30007;
        break;

    case 0x0706: [w:GD62000] = 30009;
        break;

    case 0x0707: [w:GD62000] = 30009;
        break;
    }
}

if([w:TMP0008] == 1)
{
    switch([w:GD16000[w:TMP0004]])
    {
        case 0x1000: [w:GD62000] = 30010;
            break;

        case 0x1050: [w:GD62000] = 30010;
            break;

        case 0x1200: [w:GD62000] = 30011;
            break;

        case 0x1250: [w:GD62000] = 30011;
            break;

        case 0x1400: [w:GD62000] = 30012;
            break;

        case 0x1401: [w:GD62000] = 30019;
            break;

        case 0x1450: [w:GD62000] = 30012;
            break;

        case 0x1451: [w:GD62000] = 30019;
            break;

        case 0x1500: [w:GD62000] = 30043;
            break;

        // case 0x1503: [w:GD62000] = 30046;
        //         break;

        case 0x1553: [w:GD62000] = 30050;
            break;
    }
}

if([w:TMP0008] == 2)
{
    switch([w:GD16000[w:TMP0004]])
    {

        case 0x2000: [w:GD62000] = 30013;
            break;

        case 0x2300: [w:GD62000] = 30014;

```

```

        break;

    case 0x2301: [w:GD62000] = 30014;
        break;

    case 0x2302: [w:GD62000] = 30015;
        break;

    case 0x2303: [w:GD62000] = 30016;
        break;

    case 0x2600: [w:GD62000] = 30017;
        break;

    case 0x2601: [w:GD62000] = 30017;
        break;

    case 0x2602: [w:GD62000] = 30017;
        break;

    case 0x2603: [w:GD62000] = 30017;
        break;

    case 0x2900: [w:GD62000] = 30020;
        break;

    case 0x2903: [w:GD62000] = 30021;
        break;

    case 0x2909: [w:GD62000] = 30023;
        break;

    case 0x290C: [w:GD62000] = 30053;
        break;

    case 0x290F: [w:GD62000] = 30054;
        break;

    case 0x2930: [w:GD62000] = 30024;
        break;

    case 0x2933: [w:GD62000] = 30025;
        break;

    case 0x2939: [w:GD62000] = 30027;
        break;

    case 0x293C: [w:GD62000] = 30057;
        break;

    case 0x293F: [w:GD62000] = 30058;
        break;

    case 0x2960: [w:GD62000] = 30028;
        break;

    case 0x2963: [w:GD62000] = 30029;
        break;

    case 0x2969: [w:GD62000] = 30031;
        break;

    case 0x293C: [w:GD62000] = 30061;
        break;

```

```

        case 0x293F: [w:GD62000] = 30062;
                    break;
    }
}

if([w:TMP0008] == 3)
{
    switch([w:GD16000[w:TMP0004]])
    {
        case 0x3100: [w:GD62000] = 30032;
                    break;

        case 0x3300: [w:GD62000] = 30033;
                    break;

        case 0x3301: [w:GD62000] = 30034;
                    break;

        case 0x3302: [w:GD62000] = 30035;
                    break;

        case 0x3303: [w:GD62000] = 30035;
                    break;

        case 0x3304: [w:GD62000] = 30036;
                    break;

        case 0x3305: [w:GD62000] = 30036;
                    break;
    }
}

if([w:TMP0008] == 10)
{
    switch([w:GD16000[w:TMP0004]])
    {
        case 0xA000: [w:GD62000] = 30018;
                    break;

        case 0xA001: [w:GD62000] = 30018;
                    break;

        case 0xA002: [w:GD62000] = 30037;
                    break;

        case 0xA003: [w:GD62000] = 30037;
                    break;

        case 0xA004: [w:GD62000] = 30038;
                    break;

        case 0xA005: [w:GD62000] = 30038;
                    break;

        case 0xA006: [w:GD62000] = 30039;
                    break;

        case 0xA007: [w:GD62000] = 30039;
                    break;

        case 0xA010: [w:GD62000] = 30051;

```

```

        break;

    case 0xA300: [w:GD62000] = 30018;
        break;

    case 0xA301: [w:GD62000] = 30018;
        break;

    case 0xA302: [w:GD62000] = 30037;
        break;

    case 0xA303: [w:GD62000] = 30037;
        break;

    case 0xA304: [w:GD62000] = 30038;
        break;

    case 0xA305: [w:GD62000] = 30038;
        break;

    case 0xA306: [w:GD62000] = 30039;
        break;

    case 0xA307: [w:GD62000] = 30039;
        break;

    case 0xA308: [w:GD62000] = 30040;
        break;

    case 0xA309: [w:GD62000] = 30040;
        break;

    case 0xA310: [w:GD62000] = 30052;
        break;

    case 0xA320: [w:GD62000] = 30041;
        break;

    case 0xA321: [w:GD62000] = 30042;
        break;

    case 0xA322: [w:GD62000] = 30042;
        break;

    case 0xA323: [w:GD62000] = 30042;
        break;

    case 0xA324: [w:GD62000] = 30042;
        break;

    case 0xA325: [w:GD62000] = 30042;
        break;

    case 0xA326: [w:GD62000] = 30042;
        break;

    case 0xA600: [w:GD62000] = 30018;
        break;

    case 0xA601: [w:GD62000] = 30018;
        break;

    case 0xA602: [w:GD62000] = 30037;
        break;

```



```

        case 0xA603: [w:GD62000] = 30037;
                    break;

        case 0xA604: [w:GD62000] = 30038;
                    break;

        case 0xA605: [w:GD62000] = 30038;
                    break;

        case 0xA606: [w:GD62000] = 30039;
                    break;

        case 0xA607: [w:GD62000] = 30039;
                    break;
    }
}
}
//Flag Reset
rst([b:GB30009]);
rst([b:GB30020]);
}

```

```
rst([b:GB30002]);
```

Script No.	30009	Script name	Script30009
Comment	Coordinate Calculation(Output) *1		
Data type	Signed BIN16	Trigger type	Ordinary

```
//Determine the cursor position from the touched coordinate.
```

```

if((([w:TMP0120] != [w:GS654]) || ([w:TMP0121] != [w:GS655])) && ([w:GS654] > Input_X) && ([w:GS655] > Input_Y))
&& ([w:GS654] < Frame_X) && ([w:GS655] < Frame_Y)))
{
    [w:TMP0120] = [w:GS654];
    [w:TMP0121] = [w:GS655];

    if([b:GB30001] == OFF)
    {
        //Save Current Cursor Position
        [w:TMP0122] = [w:GD10303];

        //X Coordinate
        //Change values to subtract depending on whether input or output.
        //Output
        [w:TMP0123] = [w:TMP0120] - Input_X;
        [w:TMP0124] = [w:TMP0123] / Object_X;
        [w:GD10301] = 16 - ([w:TMP0124] + 1);

        //Y Coordinate
        [w:TMP0125] = [w:TMP0121] - Input_Y;
        [w:TMP0126] = [w:TMP0125] / Object_Y;
        [w:GD10302] = [w:TMP0126] * 16;

        //Offset
        [w:GD10303] = [w:GD10301] + [w:GD10302];

        //If the offset position does not change, displayed parts do not change.
        if([w:TMP0122] != [w:GD10303])
        {
            //Cursor Display
            if([w:GD12000[w:GD10303]] < 4)
            {
                [w:GD12000[w:GD10303]] = [w:GD12000[w:GD10303]] + 4;
            }
        }
    }
}

```

```

    }

    //Cursor Delete
    if([w:GD12000[w:TMP0122]] < 4)
    {
        [w:GD12000[w:TMP0122]] = 0;
    }else{
        [w:GD12000[w:TMP0122]] = [w:GD12000[w:TMP0122]] - 4;
    }
    }
}

[w:GD10300] = [w:GD10303]; //Cursor Position Display
}

```

Script No.	30083	Script name	Script30083
Comment	Alarm Reset		
Data type	Signed BIN16	Trigger type	Rise, GB30049

//Turns on the alarm reset signal.

```

switch([w:GD64000])
{
    case Master_1: set([b:Y0000]);
                    break;

    case Master_2: set([b:Y0020]);
                    break;

    case Master_3: set([b:Y0040]);
                    break;

    case Master_4: set([b:Y0060]);
                    break;

    default:       break;
}
set([b:GB30050]);

rst([b:GB30049]);

```

Script No.	30084	Script name	Script30084
Comment	Alarm Clear Trigger Reset		
Data type	Signed BIN16	Trigger type	ON Sampling, 1 Sec, GB30050

//Turns off the alarm reset signal.

```

switch([w:GD64000])
{
    case Master_1: rst([b:Y0000]);
                    break;

    case Master_2: rst([b:Y0020]);
                    break;

    case Master_3: rst([b:Y0040]);
                    break;

    case Master_4: rst([b:Y0060]);
                    break;

    default:       break;
}

```

```
rst([b:GB30050]);
```

\*1: [Script Symbol] is used. For more details about [Script Symbol], please refer to "5.9.4 Script Symbol".

### Base screen 30005

Script No.	30023	Script name	Script30023
Comment	B-30002,30005 Scr.Start Process		
Data type	Signed BIN16	Trigger type	ON, GB40
<pre>//Read Statuses or Parameters if([b:GB30007] == OFF) {   if(([w:U00-G8192] != 0)    ([w:U00-G9984] != 0))   {     switch([w:GD64000])     {       case Master_1: set([b:GD11004.b0]);                     break;        case Master_2: set([b:GD11012.b0]);                     break;        case Master_3: set([b:GD11020.b0]);                     break;        case Master_4: set([b:GD11028.b0]);                     break;        default:      break;     }   }else{     switch([w:GD64000])     {       case Master_1: set([b:GD11000.b0]);                     break;        case Master_2: set([b:GD11008.b0]);                     break;        case Master_3: set([b:GD11016.b0]);                     break;        case Master_4: set([b:GD11024.b0]);                     break;        default:      break;     }   }    set([b:GB30007]); }</pre>			
Script No.	30016	Script name	Script30016
Comment	Script No.30015 Start		
Data type	Signed BIN16	Trigger type	Ordinary
<pre>//After reading the status, start Script No.30015. if([b:GB30016] == OFF) {   set([b:GB30008]);   set([b:GB30016]); }</pre>			
Script No.	30015	Script name	Script30015
Comment	B-30005 Model and Others		
Data type	Signed BIN16	Trigger type	ON, GB30008

```

//Display Objects
[w:GD10494] = 0;
[w:TMP0004] = 0;
[w:TMP0006] = 1;
[w:TMP0100] = [w:GD10401];
[w:TMP0101] = [w:GD10401] + [w:GD10601];
fmov([w:TMP0004],[w:TMP0010],5);

while([w:GD10494] < 15)
{
    //Check Connection Status
    if(([w:GD20162[w:GD10601]] != 0) && ([w:TMP0100] < [w:GD20162[w:GD10601]]))
    {
        [w:TMP0010] = [w:GD20163[w:TMP0101]]; //Connection ID
        [w:TMP0100] = [w:TMP0100] + 1;
        [w:TMP0101] = [w:TMP0101] + 1;

        if(([w:TMP0010] & 0x0200) == 0)
        {
            set([b:GB30600]); //Output Bit
            [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9;
        }else{
            set([b:GB30601]); //Input Bit
            [w:TMP0011] = ([w:TMP0010] & 0x0200) >> 9;
        }

        //Offset
        [w:TMP0000] = ([w:TMP0010] & 0x00FF) + [w:GD10601];

        if(([w:TMP0010] & 0x0200) == 0)
        {
            //Output
            [w:TMP0001] = [w:GD20420[w:TMP0000]];
        }else{
            //Input
            [w:TMP0001] = [w:GD20676[w:TMP0000]];
        }
        [w:TMP0014] = ([w:TMP0001] - 12288) + [w:GD10602];

        //Module Type
        [w:TMP0002] = [w:TMP0014] + [w:GD10602];
        [w:GD10402[w:GD10494]] = ([w:GD24043[w:TMP0014]] & 0x00C0) >> 6;

        //Module Model No.
        [w:TMP0005] = [w:TMP0014] + [w:GD10601];
        [w:GD10477[w:GD10494]] = [w:GD24044[w:TMP0014]];

        //Alarm Information
        if([w:GD20291[w:GD10601]] != 0)
        {
            [w:TMP0004] = 0;
            [w:TMP0005] = [w:GD10601];
            while([w:TMP0004] < [w:GD20291[w:GD10601]])
            {
                if([w:GD20292[w:TMP0005]] == [w:TMP0010])
                {
                    [w:TMP0012] = 1; //Alarm Occurrence
                    if(([w:U00-G10256] != 304) && (([w:U00-G10256] < 200) || ([w:U00-G10256] > 202)))
                    {
                        [w:GD10417[w:GD10494]] = [w:U00-G10256];
                    }else{
                        [w:TMP0003] = ([w:TMP0001] - 0x3000) + [w:GD10602];
                        [w:GD10417[w:GD10494]] = [w:GD24039[w:TMP0003]];
                    }
                }
            }
        }
    }
}

```

```

        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
    [w:TMP0005] = [w:TMP0005] + 1;
}

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0004] = 0;
    [w:TMP0005] = [w:GD10601];
    while([w:TMP0004] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0005]] == [w:TMP0010])
        {
            [w:TMP0013] = 1; //Error Occurrence
            if(([w:U00-G10256] == 304) || (([w:U00-G10256] >= 200) && ([w:U00-G10256] <= 202)))
            {
                [w:GD10417[w:GD10494]] = [w:U00-G10256];
            }else{
                [w:TMP0003] = ([w:TMP0001] - 0x3000) + [w:GD10602];
                [w:GD10417[w:GD10494]] = [w:GD24039[w:TMP0003]];
            }
            break;
        }
        [w:TMP0004] = [w:TMP0004] + 1;
        [w:TMP0005] = [w:TMP0005] + 1;
    }
}

//Display
if([w:TMP0013] == 1)
{
    [w:GD10447[w:GD10494]] = 3; //Error Occurrence
    [w:GD10132[w:GD10494]] = [w:TMP0011]; //I/O
    [w:GD10462[w:GD10494]] = [w:TMP0010] & 0x00FF; //ID
    [w:GD10432[w:GD10494]] = [w:TMP0010]; //User-defined name
    [w:GD10492] = [w:GD10492] | ([w:TMP0006] << [w:GD10494]); //Display Conditions
}else{
    if([w:TMP0012] == 1)
    {
        [w:GD10447[w:GD10494]] = 2; //Alarm Occurrence
        [w:GD10132[w:GD10494]] = [w:TMP0011]; //I/O
        [w:GD10462[w:GD10494]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10432[w:GD10494]] = [w:TMP0010]; //User-defined name
        [w:GD10492] = [w:GD10492] | ([w:TMP0006] << [w:GD10494]); //Display Conditions
    }else{
        [w:GD10447[w:GD10494]] = 1; //Normal
        [w:GD10132[w:GD10494]] = [w:TMP0011]; //I/O
        [w:GD10462[w:GD10494]] = [w:TMP0010] & 0x00FF; //ID
        [w:GD10432[w:GD10494]] = [w:TMP0010]; //User-defined name
        [w:GD10417[w:GD10494]] = 0; //Status
        [w:GD10492] = [w:GD10492] | ([w:TMP0006] << [w:GD10494]); //Display Conditions
    }
}

[w:GD10494] = [w:GD10494] + 1;
rst([b:GB30600]); //Output Bit
rst([b:GB30601]); //Input Bit

//Work Area Clear
[w:TMP0004] = 0;

```

<pre> fmov([w:TMP0004],[w:TMP0010],5);  }else{   //If data does not exist, do not display.   [w:GD10447[w:GD10494]] = 0xFFFF;   [w:GD10132[w:GD10494]] = 0xFFFF;   [w:GD10462[w:GD10494]] = 0xFFFF;   [w:GD10477[w:GD10494]] = 0xFFFF;   [w:GD10402[w:GD10494]] = 0xFFFF;   [w:GD10432[w:GD10494]] = 0x7FFE - [w:GD10601];   [w:GD10417[w:GD10494]] = 24219;   [w:GD10492] = [w:GD10492] ^ ([w:TMP0006] &lt;&lt; [w:GD10494]); //Display Conditions   [w:GD10494] = [w:GD10494] + 1;   rst([b:GB30600]); //Output Bit   rst([b:GB30601]); //Input Bit } }  //Trigger Reset rst([b:GB30008]);  //Initial Action Check Flag. set([b:GB30035]); </pre>			
Script No.	30021	Script name	Script30021
Comment	Flags Clear		
Data type	Signed BIN16	Trigger type	When closing a screen
<pre> //Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]); rst([b:GB30018]); rst([b:GB30035]);  [w:TMP0004] = 0; fmov([w:TMP0004],[w:GD12000],512); //Initialize Object Values(B-30003, 4) fmov([w:TMP0004],[w:GD10301],3); //Clear Cursor Position Information(B-30003,4) fmov([w:TMP0004],[w:GD10432],15); //Work Area Clear [w:GD10300] = 0; //Cursor Position Information Clear(B-30003,4)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			
Script No.	30024	Script name	Script30024
Comment	Previous Page		
Data type	Signed BIN16	Trigger type	Rise, GB30014
<pre> //Scroll List Up if([w:GD10401] &gt; 0) {   [w:GD10401] = [w:GD10401] - 15;   set([b:GB30008]); } </pre>			
Script No.	30025	Script name	Script30025
Comment	Next Page		
Data type	Signed BIN16	Trigger type	Rise, GB30015
<pre> //Scroll List Down if(((w:GD10401] + 15) &lt; [w:U00-G8960]) &amp;&amp; ([w:U00-G8960] &gt; 15)) {   [w:GD10401] = [w:GD10401] + 15;   set([b:GB30008]); } </pre>			
Script No.	30034	Script name	Script30034
Comment	B-30005 Detail Screen Display		

Data type	Unsigned BIN16	Trigger type	ON, GB30018
<pre> //Determine detail screen to display. [w:TMP0000] = ([w:GD10493] &amp; 0x00FF) + [w:GD10601];  if((([w:GD10493] &amp; 0x0200) &gt;&gt; 8) == 0) {     //Output     [w:GD10600] = [w:GD20420[w:TMP0000]]; //Offset }else{     //Input     [w:GD10600] = [w:GD20676[w:TMP0000]]; //Offset }  [w:U00-G10320] = 0; //Read Individual Parameter [w:U00-G10321] = [w:GD10493]; //Read Target ID  //Select base screen to display. switch([w:GD10477[w:GD10400]] &amp; 0xF000) {     case 0x0000: [w:TMP0003] = 0;                 break;      case 0x1000: [w:TMP0003] = 1;                 break;      case 0x2000: [w:TMP0003] = 2;                 break;      case 0x3000: [w:TMP0003] = 3;                 break;      case 0x9000: [w:TMP0003] = 9;                 break;      case 0xA000: [w:TMP0003] = 10;                 break;  }  if([w:TMP0003] == 0) {     switch([w:GD10477[w:GD10400]])     {         case 0x0100: [w:GD62000] = 30006;                     break;          case 0x0101: [w:GD62000] = 30006;                     break;          case 0x0102: [w:GD62000] = 30008;                     break;          case 0x0103: [w:GD62000] = 30008;                     break;          case 0x0104: [w:GD62000] = 30007;                     break;          case 0x0105: [w:GD62000] = 30007;                     break;          case 0x0106: [w:GD62000] = 30009;                     break;     } </pre>			

```

        case 0x0107: [w:GD62000] = 30009;
            break;

        case 0x0400: [w:GD62000] = 30006;
            break;

        case 0x0401: [w:GD62000] = 30006;
            break;

        case 0x0402: [w:GD62000] = 30008;
            break;

        case 0x0403: [w:GD62000] = 30008;
            break;

        case 0x0404: [w:GD62000] = 30007;
            break;

        case 0x0405: [w:GD62000] = 30007;
            break;

        case 0x0406: [w:GD62000] = 30009;
            break;

        case 0x0407: [w:GD62000] = 30009;
            break;

        case 0x0700: [w:GD62000] = 30006;
            break;

        case 0x0701: [w:GD62000] = 30006;
            break;

        case 0x0702: [w:GD62000] = 30008;
            break;

        case 0x0703: [w:GD62000] = 30008;
            break;

        case 0x0704: [w:GD62000] = 30007;
            break;

        case 0x0705: [w:GD62000] = 30007;
            break;

        case 0x0706: [w:GD62000] = 30009;
            break;

        case 0x0707: [w:GD62000] = 30009;
            break;
    }
}

if([w:TMP0003] == 1)
{
    switch([w:GD10477[w:GD10400]])
    {
        case 0x1000: [w:GD62000] = 30010;
            break;

        case 0x1050: [w:GD62000] = 30010;
            break;

        case 0x1200: [w:GD62000] = 30011;

```



```

        break;

    case 0x1250: [w:GD62000] = 30011;
        break;

    case 0x1400: [w:GD62000] = 30012;
        break;

    case 0x1401: [w:GD62000] = 30019;
        break;

    case 0x1450: [w:GD62000] = 30012;
        break;

    case 0x1451: [w:GD62000] = 30019;
        break;

    case 0x1500: [w:GD62000] = 30043;
        break;

    // case 0x1503: [w:GD62000] = 30046;
    //         break;

    case 0x1553: [w:GD62000] = 30050;
        break;
    }
}

if([w:TMP0003] == 2)
{
    switch([w:GD10477[w:GD10400]])
    {

        case 0x2000: [w:GD62000] = 30013;
            break;

        case 0x2300: [w:GD62000] = 30014;
            break;

        case 0x2301: [w:GD62000] = 30014;
            break;

        case 0x2302: [w:GD62000] = 30015;
            break;

        case 0x2303: [w:GD62000] = 30016;
            break;

        case 0x2600: [w:GD62000] = 30017;
            break;

        case 0x2601: [w:GD62000] = 30017;
            break;

        case 0x2602: [w:GD62000] = 30017;
            break;

        case 0x2603: [w:GD62000] = 30017;
            break;

        case 0x2900: [w:GD62000] = 30020;
            break;

        case 0x2903: [w:GD62000] = 30021;

```

```

        break;

    case 0x2909: [w:GD62000] = 30023;
        break;

    case 0x290C: [w:GD62000] = 30053;
        break;

    case 0x290F: [w:GD62000] = 30054;
        break;

    case 0x2930: [w:GD62000] = 30024;
        break;

    case 0x2933: [w:GD62000] = 30025;
        break;

    case 0x2939: [w:GD62000] = 30027;
        break;

    case 0x293C: [w:GD62000] = 30057;
        break;

    case 0x293F: [w:GD62000] = 30058;
        break;

    case 0x2960: [w:GD62000] = 30028;
        break;

    case 0x2963: [w:GD62000] = 30029;
        break;

    case 0x2969: [w:GD62000] = 30031;
        break;

    case 0x293C: [w:GD62000] = 30061;
        break;

    case 0x293F: [w:GD62000] = 30062;
        break;
    }
}

if([w:TMP0003] == 3)
{
    switch([w:GD10477[w:GD10400]])
    {

        case 0x3100: [w:GD62000] = 30032;
            break;

        case 0x3300: [w:GD62000] = 30033;
            break;

        case 0x3301: [w:GD62000] = 30034;
            break;

        case 0x3302: [w:GD62000] = 30035;
            break;

        case 0x3303: [w:GD62000] = 30035;
            break;

        case 0x3304: [w:GD62000] = 30036;

```

```

        break;

        case 0x3305: [w:GD62000] = 30036;
            break;
    }
}

if([w:TMP0003] == 10)
{
    switch([w:GD10477[w:GD10400]])
    {

        case 0xA000: [w:GD62000] = 30018;
            break;

        case 0xA001: [w:GD62000] = 30018;
            break;

        case 0xA002: [w:GD62000] = 30037;
            break;

        case 0xA003: [w:GD62000] = 30037;
            break;

        case 0xA004: [w:GD62000] = 30038;
            break;

        case 0xA005: [w:GD62000] = 30038;
            break;

        case 0xA006: [w:GD62000] = 30039;
            break;

        case 0xA007: [w:GD62000] = 30039;
            break;

        case 0xA010: [w:GD62000] = 30051;
            break;

        case 0xA300: [w:GD62000] = 30018;
            break;

        case 0xA301: [w:GD62000] = 30018;
            break;

        case 0xA302: [w:GD62000] = 30037;
            break;

        case 0xA303: [w:GD62000] = 30037;
            break;

        case 0xA304: [w:GD62000] = 30038;
            break;

        case 0xA305: [w:GD62000] = 30038;
            break;

        case 0xA306: [w:GD62000] = 30039;
            break;

        case 0xA307: [w:GD62000] = 30039;
            break;

        case 0xA308: [w:GD62000] = 30040;

```

```

        break;

    case 0xA309: [w:GD62000] = 30040;
        break;

    case 0xA310: [w:GD62000] = 30052;
        break;

    case 0xA320: [w:GD62000] = 30041;
        break;

    case 0xA321: [w:GD62000] = 30042;
        break;

    case 0xA322: [w:GD62000] = 30042;
        break;

    case 0xA323: [w:GD62000] = 30042;
        break;

    case 0xA324: [w:GD62000] = 30042;
        break;

    case 0xA325: [w:GD62000] = 30042;
        break;

    case 0xA326: [w:GD62000] = 30042;
        break;

    case 0xA600: [w:GD62000] = 30018;
        break;

    case 0xA601: [w:GD62000] = 30018;
        break;

    case 0xA602: [w:GD62000] = 30037;
        break;

    case 0xA603: [w:GD62000] = 30037;
        break;

    case 0xA604: [w:GD62000] = 30038;
        break;

    case 0xA605: [w:GD62000] = 30038;
        break;

    case 0xA606: [w:GD62000] = 30039;
        break;

    case 0xA607: [w:GD62000] = 30039;
        break;
    }
}

```

rst([b:GB30018]);

Script No.	30083	Script name	Script30083
Comment	Alarm Reset		
Data type	Signed BIN16	Trigger type	Rise, GB30049

//Turns on the alarm reset signal.

```

switch([w:GD64000])
{

```

```

case Master_1: set([b:Y0000]);
               break;

case Master_2: set([b:Y0020]);
               break;

case Master_3: set([b:Y0040]);
               break;

case Master_4: set([b:Y0060]);
               break;

default:      break;

}
set([b:GB30050]);

```

```
rst([b:GB30049]);
```

Script No.	30084	Script name	Script30084
Comment	Alarm Clear Trigger Reset		
Data type	Signed BIN16	Trigger type	ON Sampling, 1 Sec, GB30050

```
//Turns off the alarm reset signal.
```

```

switch([w:GD64000])
{
  case Master_1: rst([b:Y0000]);
                 break;

  case Master_2: rst([b:Y0020]);
                 break;

  case Master_3: rst([b:Y0040]);
                 break;

  case Master_4: rst([b:Y0060]);
                 break;

  default:      break;

}

```

```
rst([b:GB30050]);
```

**Base screen 30006 to 30021, Base screen 30023 to 30025, Base screen 30027 to 30029, Base screen 30031 to 30042, Base screen 30051 to 30054, Base screen 30057 to 30058, Base screen 30061 to 30062**

Script No.	30027	Script name	Script30027
Comment	Parameter Display		
Data type	Signed BIN16	Trigger type	ON, GB40

```

if([b:GB30017] == OFF)
{
  //Error Information
  if([w:GD20032[w:GD10601]] != 0)
  {
    [w:TMP0000] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0000];
    while([w:TMP0000] < [w:GD20032[w:GD10601]])
    {
      if([w:GD20033[w:TMP0001]] == [w:U00-G10321])
      {
        set([b:GB30028]); //Error Occurrence Flag
        break;
      }
    }
  }
}

```

```

[w:TMP0000] = [w:TMP0000] + 1;
[w:TMP0001] = [w:TMP0001] + 1;

}
}

if([b:GB30028] == ON)
{
  //If errors occurred, read the most recently read settings.
  switch([w:GD64000])
  {
    case Master_1: set([b:GD11002.b0]);
                  break;

    case Master_2: set([b:GD11010.b0]);
                  break;

    case Master_3: set([b:GD11018.b0]);
                  break;

    case Master_4: set([b:GD11026.b0]);
                  break;

    default:      break;

  }
}else{
  //If errors are not occurred, write the latest information.
  switch([w:GD64000])
  {
    case Master_1: set([b:Y0010]);
                  break;

    case Master_2: set([b:Y0030]);
                  break;

    case Master_3: set([b:Y0050]);
                  break;

    case Master_4: set([b:Y0070]);
                  break;

    default:      break;

  }
}
[w:GD10500] = 1; //Specify Start Line of Processing Method
set([b:GB30017]);
}

```

Script No.	30014	Script name	Script30014
Comment	Module No.1 Parameter Access OFF		
Data type	Signed BIN16	Trigger type	ON, Y0010

```

//Master Module No.1
//Processing after Reading Parameters Individually
if([w:U00-G10320] == 0)
{
  set([b:GB30024]); //Read Flag
}else{
  set([b:GB30025]); //Write Flag
}

if([b:X0012] == ON)
{
  [w:GD62007] = 30004;
}

```

```

    rst([b:GB30024]);
    rst([b:GB30025]);
    rst([b:Y0010]);
}else{
    if([b:X0011] == ON)
    {
        rst([b:GB30024]);
        rst([b:GB30025]);
        //If the alarm bit of the pressure sensor has been switched,
        //then return to the [AnyWireASLINK Master Module SEL] screen.
        if([b:GB30034] == ON)
        {
            [w:GD62000] = 30000;
        }else{
            set([b:GB30026]);
        }
        rst([b:Y0010]);
    }
}

```

Script No.	30026	Script name	Script30026
Comment	Module Model No. Correct		
Data type	Signed BIN16	Trigger type	Ordinary

```

if([b:GB30021] == OFF)
{
    [w:TMP1000] = 0;

    //Display contents according to the module model number.
    switch([w:GD11525] & 0xF000)
    {

        case 0x0000: [w:TMP1000] = 0;
                     break;

        case 0x1000: [w:TMP1000] = 1;
                     break;

        case 0x2000: [w:TMP1000] = 2;
                     break;

        case 0x3000: [w:TMP1000] = 3;
                     break;

        case 0x9000: [w:TMP1000] = 9;
                     break;

        case 0xA000: [w:TMP1000] = 10;
                     break;

        default:     [w:TMP1000] = 65535;
                     break;

    }

    if([w:TMP1000] == 0)
    {
        switch([w:GD11525])
        {
            case 0x0100: [w:GD10501] = 10000;
                         break;
            case 0x0101: [w:GD10501] = 10001;
                         break;
            case 0x0102: [w:GD10501] = 10002;
                         break;

```

```

        case 0x0103: [w:GD10501] = 10003;
            break;
        case 0x0104: [w:GD10501] = 10004;
            break;
        case 0x0105: [w:GD10501] = 10005;
            break;
        case 0x0106: [w:GD10501] = 10006;
            break;
        case 0x0107: [w:GD10501] = 10007;
            break;
        case 0x0400: [w:GD10501] = 10008;
            break;
        case 0x0401: [w:GD10501] = 10009;
            break;
        case 0x0402: [w:GD10501] = 10010;
            break;
        case 0x0403: [w:GD10501] = 10011;
            break;
        case 0x0404: [w:GD10501] = 10012;
            break;
        case 0x0405: [w:GD10501] = 10013;
            break;
        case 0x0406: [w:GD10501] = 10014;
            break;
        case 0x0407: [w:GD10501] = 10015;
            break;
        case 0x0700: [w:GD10501] = 10016;
            break;
        case 0x0701: [w:GD10501] = 10017;
            break;
        case 0x0702: [w:GD10501] = 10018;
            break;
        case 0x0703: [w:GD10501] = 10019;
            break;
        case 0x0704: [w:GD10501] = 10020;
            break;
        case 0x0705: [w:GD10501] = 10021;
            break;
        case 0x0706: [w:GD10501] = 10022;
            break;
        case 0x0707: [w:GD10501] = 10023;
            break;
        default: break;
    }
}

if([w:TMP1000] == 1)
{
    switch([w:GD11525])
    {
        case 0x1000: [w:GD10501] = 11000;
            break;
        case 0x1050: [w:GD10501] = 11001;
            break;
        case 0x1200: [w:GD10501] = 11002;
            break;
        case 0x1250: [w:GD10501] = 11003;
            break;
        case 0x1400: [w:GD10501] = 11004;
            break;
        case 0x1401: [w:GD10501] = 11005;
            break;
        case 0x1450: [w:GD10501] = 11006;
            break;
    }
}

```



```

        case 0x1451: [w:GD10501] = 11007;
            break;
        case 0x1500: [w:GD10501] = 11008;
            break;
        case 0x1501: [w:GD10501] = 11009;
            break;
        case 0x1502: [w:GD10501] = 11010;
            break;
        case 0x1503: [w:GD10501] = 11011;
            break;
        case 0x1504: [w:GD10501] = 11012;
            break;
        case 0x1550: [w:GD10501] = 11013;
            break;
        case 0x1551: [w:GD10501] = 11014;
            break;
        case 0x1552: [w:GD10501] = 11015;
            break;
        case 0x1553: [w:GD10501] = 11016;
            break;
        case 0x1554: [w:GD10501] = 11017;
            break;
        default:      break;
    }
}

if([w:TMP1000] == 2)
{
    switch([w:GD11525])
    {
        case 0x2000: [w:GD10501] = 12000;
            break;
        case 0x2300: [w:GD10501] = 12001;
            break;
        case 0x2301: [w:GD10501] = 12002;
            break;
        case 0x2302: [w:GD10501] = 12003;
            break;
        case 0x2303: [w:GD10501] = 12004;
            break;
        case 0x2600: [w:GD10501] = 12005;
            break;
        case 0x2601: [w:GD10501] = 12006;
            break;
        case 0x2602: [w:GD10501] = 12007;
            break;
        case 0x2603: [w:GD10501] = 12008;
            break;
        case 0x2900: [w:GD10501] = 12009;
            break;
        case 0x2901: [w:GD10501] = 12010;
            break;
        case 0x2902: [w:GD10501] = 12011;
            break;
        case 0x2903: [w:GD10501] = 12012;
            break;
        case 0x2904: [w:GD10501] = 12013;
            break;
        case 0x2905: [w:GD10501] = 12014;
            break;
        case 0x2906: [w:GD10501] = 12015;
            break;
        case 0x2907: [w:GD10501] = 12016;
            break;
    }
}

```

```
case 0x2908:[w:GD10501] = 12017;
    break;
case 0x2909:[w:GD10501] = 12018;
    break;
case 0x290A:[w:GD10501] = 12019;
    break;
case 0x290B:[w:GD10501] = 12020;
    break;
case 0x290C:[w:GD10501] = 12021;
    break;
case 0x290D:[w:GD10501] = 12022;
    break;
case 0x290E:[w:GD10501] = 12023;
    break;
case 0x290F:[w:GD10501] = 12024;
    break;
case 0x2910:[w:GD10501] = 12025;
    break;
case 0x2911:[w:GD10501] = 12026;
    break;
case 0x2930:[w:GD10501] = 12027;
    break;
case 0x2931:[w:GD10501] = 12028;
    break;
case 0x2932:[w:GD10501] = 12029;
    break;
case 0x2933:[w:GD10501] = 12030;
    break;
case 0x2934:[w:GD10501] = 12031;
    break;
case 0x2935:[w:GD10501] = 12032;
    break;
case 0x2936:[w:GD10501] = 12033;
    break;
case 0x2937:[w:GD10501] = 12034;
    break;
case 0x2938:[w:GD10501] = 12035;
    break;
case 0x2939:[w:GD10501] = 12036;
    break;
case 0x293A:[w:GD10501] = 12037;
    break;
case 0x293B:[w:GD10501] = 12038;
    break;
case 0x293C:[w:GD10501] = 12039;
    break;
case 0x293D:[w:GD10501] = 12040;
    break;
case 0x293E:[w:GD10501] = 12041;
    break;
case 0x293F:[w:GD10501] = 12042;
    break;
case 0x2940:[w:GD10501] = 12043;
    break;
case 0x2941:[w:GD10501] = 12044;
    break;
case 0x2960:[w:GD10501] = 12045;
    break;
case 0x2961:[w:GD10501] = 12046;
    break;
case 0x2962:[w:GD10501] = 12047;
    break;
case 0x2963:[w:GD10501] = 12048;
    break;
```

```

        case 0x2964:[w:GD10501] = 12049;
            break;
        case 0x2965:[w:GD10501] = 12050;
            break;
        case 0x2966:[w:GD10501] = 12051;
            break;
        case 0x2967:[w:GD10501] = 12052;
            break;
        case 0x2968:[w:GD10501] = 12053;
            break;
        case 0x2969:[w:GD10501] = 12054;
            break;
        case 0x296A:[w:GD10501] = 12055;
            break;
        case 0x296B:[w:GD10501] = 12056;
            break;
        case 0x296C:[w:GD10501] = 12057;
            break;
        case 0x296D:[w:GD10501] = 12058;
            break;
        case 0x296E:[w:GD10501] = 12059;
            break;
        case 0x296F:[w:GD10501] = 12060;
            break;
        case 0x2970:[w:GD10501] = 12061;
            break;
        case 0x2971:[w:GD10501] = 12062;
            break;
        default:    break;
    }
}

```

```

if([w:TMP1000] == 3)
{
    switch([w:GD11525])
    {
        case 0x3100:[w:GD10501] = 13000;
            break;
        case 0x3101:[w:GD10501] = 13001;
            break;
        case 0x3300:[w:GD10501] = 13002;
            break;
        case 0x3301:[w:GD10501] = 13003;
            break;
        case 0x3302:[w:GD10501] = 13004;
            break;
        case 0x3303:[w:GD10501] = 13005;
            break;
        case 0x3304:[w:GD10501] = 13006;
            break;
        case 0x3305:[w:GD10501] = 13007;
            break;
        default:    break;
    }
}

```

```

if([w:TMP1000] == 9)
{
    switch([w:GD11525])
    {
        case 0x9000:[w:GD10501] = 14000;
            break;
        case 0x9001:[w:GD10501] = 14001;

```

```

        break;
    case 0x9002:[w:GD10501] = 14002;
        break;
    case 0x9003:[w:GD10501] = 14003;
        break;
    case 0x9004:[w:GD10501] = 14004;
        break;
    case 0x9005:[w:GD10501] = 14005;
        break;
    case 0x9006:[w:GD10501] = 14006;
        break;
    case 0x9007:[w:GD10501] = 14007;
        break;
    case 0x9008:[w:GD10501] = 14008;
        break;
    case 0x9009:[w:GD10501] = 14009;
        break;
    case 0x900A:[w:GD10501] = 14010;
        break;
    case 0x900B:[w:GD10501] = 14011;
        break;
    case 0x900C:[w:GD10501] = 14012;
        break;
    case 0x900D:[w:GD10501] = 14013;
        break;
    case 0x900F:[w:GD10501] = 14014;
        break;
    case 0x9100:[w:GD10501] = 14015;
        break;
    case 0x9101:[w:GD10501] = 14016;
        break;
    default:    break;
}
}

```

```

if([w:TMP1000] == 10)
{
    switch([w:GD11525])
    {
        case 0xA000:[w:GD10501] = 15000;
            break;
        case 0xA001:[w:GD10501] = 15001;
            break;
        case 0xA002:[w:GD10501] = 15002;
            break;
        case 0xA003:[w:GD10501] = 15003;
            break;
        case 0xA004:[w:GD10501] = 15004;
            break;
        case 0xA005:[w:GD10501] = 15005;
            break;
        case 0xA006:[w:GD10501] = 15006;
            break;
        case 0xA007:[w:GD10501] = 15007;
            break;
        case 0xA008:[w:GD10501] = 15008;
            break;
        case 0xA009:[w:GD10501] = 15009;
            break;
        case 0xA010:[w:GD10501] = 15010;
            break;
        case 0xA011:[w:GD10501] = 15011;
            break;
        case 0xA300:[w:GD10501] = 15012;

```

```

        break;
    case 0xA301:[w:GD10501] = 15013;
        break;
    case 0xA302:[w:GD10501] = 15014;
        break;
    case 0xA303:[w:GD10501] = 15015;
        break;
    case 0xA304:[w:GD10501] = 15016;
        break;
    case 0xA305:[w:GD10501] = 15017;
        break;
    case 0xA306:[w:GD10501] = 15018;
        break;
    case 0xA307:[w:GD10501] = 15019;
        break;
    case 0xA308:[w:GD10501] = 15020;
        break;
    case 0xA309:[w:GD10501] = 15021;
        break;
    case 0xA30A: [w:GD10501] = 15022;
        break;
    case 0xA30B: [w:GD10501] = 15023;
        break;
    case 0xA310:[w:GD10501] = 15024;
        break;
    case 0xA320:[w:GD10501] = 15025;
        break;
    case 0xA321:[w:GD10501] = 15026;
        break;
    case 0xA322:[w:GD10501] = 15027;
        break;
    case 0xA323:[w:GD10501] = 15028;
        break;
    case 0xA324:[w:GD10501] = 15029;
        break;
    case 0xA325:[w:GD10501] = 15030;
        break;
    case 0xA326:[w:GD10501] = 15031;
        break;
    case 0xA600:[w:GD10501] = 15032;
        break;
    case 0xA601:[w:GD10501] = 15033;
        break;
    case 0xA602:[w:GD10501] = 15034;
        break;
    case 0xA603:[w:GD10501] = 15035;
        break;
    case 0xA604:[w:GD10501] = 15036;
        break;
    case 0xA605:[w:GD10501] = 15037;
        break;
    case 0xA606:[w:GD10501] = 15038;
        break;
    case 0xA607:[w:GD10501] = 15039;
        break;
    case 0xA610:[w:GD10501] = 15040;
        break;
    default:    break;
}
}

set([b:GB30019]);
set([b:GB30021]);
}

```

Script No.	30037	Script name	Script30037
Comment	Flags Clear(Detail Screen)		
Data type	Signed BIN16	Trigger type	When closing a screen
<pre>//Clear Flags rst([b:GB30006]); rst([b:GB30007]); rst([b:GB30003]);  rst([b:GB30016]); rst([b:GB30017]); rst([b:GB30021]);  rst([b:GB30028]);  rst([b:GB30034]); rst([b:GB30035]);  [w:TMP0004] = 0; fmov([w:TMP0004],[w:GD12000],512);    //Initialize Object Values(B-30002)  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0;</pre>			
Script No.	30038	Script name	Script30038
Comment	Module Detail Info Display		
Data type	Signed BIN16	Trigger type	ON, GB30019
<pre>//Display Detail Information  //Work Area Clear [w:TMP0004] = 0; fmov([w:TMP0004],[w:TMP0020],7);  //Alarm Information if([w:GD20291[w:GD10601]] != 0) {     [w:TMP0004] = 0;     [w:TMP0001] = [w:GD10601] + [w:TMP0004];     while([w:TMP0004] &lt; [w:GD20291[w:GD10601]])     {         if([w:GD20292[w:TMP0001]] == [w:GD11500])         {             [w:TMP0022] = 1; //Alarm Occurrence             if(([w:U00-G10256] != 305) &amp;&amp; ([w:U00-G10256] != 304) &amp;&amp; ([w:U00-G10256] &lt; 200)    ([w:U00-G10256] &gt; 202)))             {                 [w:GD10502] = [w:U00-G10256];             }else{                 [w:GD10502] = [w:GD11520];             }             break;         }         [w:TMP0004] = [w:TMP0004] + 1;         [w:TMP0001] = [w:TMP0001] + 1;     } }  //Error Information if([w:GD20032[w:GD10601]] != 0) {     [w:TMP0004] = 0;</pre>			

```

[w:TMP0001] = [w:GD10601] + [w:TMP0004];
while([w:TMP0004] < [w:GD20032[w:GD10601]])
{
    if([w:GD20033[w:TMP0001]] == [w:GD11500])
    {
        [w:TMP0023] = 1; //Error Occurrence
        if(([w:U00-G10256] != 305) && (([w:U00-G10256] == 304) || (([w:U00-G10256] >= 200) &&
([w:U00-G10256] <= 202))))
        {
            [w:GD10502] = [w:U00-G10256];
        }else{
            [w:GD10502] = [w:GD11520];
        }
        break;
    }
    [w:TMP0004] = [w:TMP0004] + 1;
    [w:TMP0001] = [w:TMP0001] + 1;
}
}

//Device Parameter
//I/O Points Pattern
[w:TMP0025] = ([w:GD11524] & 0x00C0) >> 6; //0: Input, 1: Output, 2: Combined
[w:TMP0026] = ([w:GD11524] & 0x003F) + 1; //Points

//ON/OFF Information
[w:TMP0008] = 0;
[w:TMP0000] = [w:GD11500] & 0x00FF;
if([w:TMP0025] == 0) //Input Case
{
    [w:TMP0004] = 0;

    while([w:TMP0004] < [w:TMP0026])
    {
        [w:TMP0009] = [w:TMP0000] / 16;
        [w:TMP0010] = [w:TMP0000] % 16;

        [w:GD10519[w:TMP0008]] = [w:TMP0009];
        [w:TMP0012] = 0x0001;
        [w:GD10535[w:TMP0008]] = [w:TMP0012] << [w:TMP0010];

        if([w:TMP0023] == 1) //Check If Error Occurred
        {
            [w:GD10503[w:TMP0008]] = 3; //Error Occurrence
        }else{
            if([w:TMP0022] == 1) //Check If Alarms Occurred
            {
                [w:GD10503[w:TMP0008]] = 2; //Alarm Occurrence
            }else{
                [w:GD10503[w:TMP0008]] = 1; //Normal
                [w:GD10502] = 0;
            }
        }

        [w:TMP0000] = [w:TMP0000] + 1;
        [w:TMP0008] = [w:TMP0008] + 1;
        [w:TMP0004] = [w:TMP0004] + 1;
    }
}
}else{
    if([w:TMP0025] == 1) //Output Case
    {
        [w:TMP0004] = 0;
    }
}

```

```

while([w:TMP0004] < [w:TMP0026])
{
    [w:TMP0009] = [w:TMP0000] / 16;
    [w:TMP0010] = [w:TMP0000] % 16;

    [w:GD10519[w:TMP0008]] = [w:TMP0009] + 4096;
    [w:TMP0012] = 0x0001;
    [w:GD10535[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

    if([w:TMP0023] == 1) //Check If Error Occurred
    {
        [w:GD10503[w:TMP0008]] = 3; //Error Occurrence
    }else{
        if([w:TMP0022] == 1) //Check If Alarms Occurred
        {
            [w:GD10503[w:TMP0008]] = 2; //Alarm Occurrence
        }else{
            [w:GD10503[w:TMP0008]] = 1; //Normal
            [w:GD10502] = 0;
        }
    }

    [w:TMP0000] = [w:TMP0000] + 1;
    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}

}else{
    if([w:TMP0025] == 2) //Combined Case
    {
        [w:TMP0004] = 0;
        [w:TMP0011] = [w:TMP0026] / 2; //If combined, points are reduced by half.
        while([w:TMP0004] < [w:TMP0011])
        {
            [w:TMP0009] = [w:TMP0000] / 16;
            [w:TMP0010] = [w:TMP0000] % 16;

            //Output
            [w:GD10519[w:TMP0008]] = [w:TMP0009] + 4096;
            [w:TMP0012] = 0x0001;
            [w:GD10535[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

            if([w:TMP0023] == 1) //Check If Error Occurred
            {
                [w:GD10503[w:TMP0008]] = 3; //Error Occurrence
            }else{
                if([w:TMP0022] == 1) //Check If Error Occurred
                {
                    [w:GD10503[w:TMP0008]] = 2; //Alarm Occurrence
                }else{
                    [w:GD10503[w:TMP0008]] = 1; //Normal
                    [w:GD10502] = 0;
                }
            }
        }

        //Input
        [w:GD10520[w:TMP0008]] = [w:TMP0009];
        [w:TMP0012] = 0x0001;
        [w:GD10536[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

        if([w:TMP0023] == 1) //Check If Error Occurred
        {
            [w:GD10504[w:TMP0008]] = 3; //Error Occurrence

```



```

    }else{
        if([w:TMP0022] == 1) //Check If Alarms Occurred
        {
            [w:GD10504[w:TMP0008]] = 2; //Alarm Occurrence
        }else{
            [w:GD10504[w:TMP0008]] = 1; //Normal
            [w:GD10502] = 0;
        }
    }

    [w:TMP0000] = [w:TMP0000] + 1;
    [w:TMP0008] = [w:TMP0008] + 1;
    [w:TMP0004] = [w:TMP0004] + 1;
}
}
}
}
}
}
}
}
}
}

```

rst([b:GB30019]);

//Initial Action Check Flag.

set([b:GB30035]);

Script No.	30039	Script name	Script30039
Comment	Parameter Re-read		
Data type	Signed BIN16	Trigger type	Ordinary

//Re-read Parameters

if([b:GB30016] == OFF)

```

{
    switch([w:GD64000])
    {
        case Master_1: set([b:GD11002.b0]);
                        break;

        case Master_2: set([b:GD11010.b0]);
                        break;

        case Master_3: set([b:GD11018.b0]);
                        break;

        case Master_4: set([b:GD11026.b0]);
                        break;

        default:      break;
    }
    set([b:GB30016]);
}

```

Script No.	30040	Script name	Script30040
Comment	Parameter Individual Read		
Data type	Signed BIN16	Trigger type	ON, GB30026

//Processing after Reading Parameters

if([b:Y0010] == OFF && [b:Y0030] == OFF && [b:Y0050] == OFF && [b:Y0070] == OFF)

```

{
    if([w:U00-G10320] == 0)
    {
        switch([w:GD64000])
        {
            case Master_1: set([b:GD11002.b0]);
                            break;

            case Master_2: set([b:GD11010.b0]);
                            break;

            case Master_3: set([b:GD11018.b0]);

```

```

        break;

        case Master_4: set([b:GD11026.b0]);
            break;

        default:      break;
    }
}
}else{
    [w:U00-G10320] = 0;
    switch([w:GD64000])
    {
        case Master_1: set([b:Y0010]);
            break;

        case Master_2: set([b:Y0030]);
            break;

        case Master_3: set([b:Y0050]);
            break;

        case Master_4: set([b:Y0070]);
            break;

        default:      break;
    }
}

```

```

rst([b:GB30026]);
}

```

Script No.	30068	Script name	Script30068
Comment	Device Parameter Read		
Data type	Signed BIN16	Trigger type	ON, GB30029

//Read the detail information of slave module.

```

//Error Information
if([w:GD20032[w:GD10601]] != 0)
{
    [w:TMP0000] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0000];

    while([w:TMP0000] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:U00-G10321])
        {
            set([b:GB30028]); //Error Occurrence Flag
            break;
        }
        [w:TMP0000] = [w:TMP0000] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

if([b:GB30028] == ON)
{
    //If errors occurred, display error message.
    [w:GD62007] = 30004;
    rst([b:GB30024]);
}else{
    //If errors are not occurred, write the latest information.
    [w:U00-G10320] = 0;
    [w:U00-G10321] = [w:GD11500];
}

```

```

switch([w:GD64000])
{
    case Master_1: set([b:Y0010]);
                    break;

    case Master_2: set([b:Y0030]);
                    break;

    case Master_3: set([b:Y0050]);
                    break;

    case Master_4: set([b:Y0070]);
                    break;

    default:       break;
}
}

```

```
rst([b:GB30029]);
```

Script No.	30069	Script name	Script30069
Comment	Device Parameter Write		
Data type	Signed BIN16	Trigger type	ON, GB30030

//Write the detail information of slave module.

//Error Information

```
if([w:GD20032[w:GD10601]] != 0)
```

```

{
    [w:TMP0000] = 0;
    [w:TMP0001] = [w:GD10601] + [w:TMP0000];
    while([w:TMP0000] < [w:GD20032[w:GD10601]])
    {
        if([w:GD20033[w:TMP0001]] == [w:U00-G10321])
        {
            set([b:GB30028]); //Error Occurrence Flag
            break;
        }
        [w:TMP0000] = [w:TMP0000] + 1;
        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

```

```
if([b:GB30028] == ON)
```

```

{
    //If errors occurred, display error message.
    [w:GD62007] = 30004;
    rst([b:GB30025]);
}
else{
    //If errors are not occurred, write the latest information.
    [w:U00-G10320] = 1;
    [w:U00-G10321] = [w:GD11500];

    if(([w:GD11525] >= 0x2900) && ([w:GD11525] <= 0x2971))
    {
        //Judge whether the alarm bit of the pressure sensor has been switched.
        if([w:GD10572] == [w:GD11501])
        {
            rst([b:GB30034]);
        }
        else{
            set([b:GB30034]);
        }
    }
}

```

```
switch([w:GD64000])
```

```

{
    case Master_1: set([b:GD11006.b0]);
        break;

    case Master_2: set([b:GD11014.b0]);
        break;

    case Master_3: set([b:GD11022.b0]);
        break;

    case Master_4: set([b:GD11030.b0]);
        break;

    default:      break;
}
}

```

```
rst([b:GB30030]);
```

Script No.	30075	Script name	Script30075
Comment	Module No.2 Parameter Access OFF		
Data type	Signed BIN16	Trigger type	ON, Y0030

```

//Master Module No.2
//Processing after Reading Parameters Individually
if([w:U02-G10320] == 0)
{
    set([b:GB30024]); //Read Flag
}else{
    set([b:GB30025]); //Write Flag
}

if([b:X0032] == ON)
{
    [w:GD62007] = 30004;
    rst([b:GB30024]);
    rst([b:GB30025]);
    rst([b:Y0030]);
}else{
    if([b:X0031] == ON)
    {
        rst([b:GB30024]);
        rst([b:GB30025]);
        set([b:GB30026]);
        rst([b:Y0030]);
    }
}
}

```

Script No.	30078	Script name	Script30078
Comment	Module No.3 Parameter Access OFF		
Data type	Signed BIN16	Trigger type	ON, Y0050

```

//Master Module No.3
//Processing after Reading Parameters Individually
if([w:U04-G10320] == 0)
{
    set([b:GB30024]); //Read Flag
}else{
    set([b:GB30025]); //Write Flag
}

if([b:X0052] == ON)
{
    [w:GD62007] = 30004;
    rst([b:GB30024]);
    rst([b:GB30025]);
    rst([b:Y0050]);
}

```

```

}else{
  if([b:X0051] == ON)
  {
    rst([b:GB30024]);
    rst([b:GB30025]);
    set([b:GB30026]);
    rst([b:Y0050]);
  }
}

```

Script No.	30081	Script name	Script30081
Comment	Module No.4 Parameter Access OFF		
Data type	Signed BIN16	Trigger type	ON, Y0070

```

//Master Module No.4
//Processing after Reading Parameters Individually.
if([w:U06-G10320] == 0)
{
  set([b:GB30024]); //Read Flag
}else{
  set([b:GB30025]); //Write Flag
}

if([b:X0072] == ON)
{
  [w:GD62007] = 30004;
  rst([b:GB30024]);
  rst([b:GB30025]);
  rst([b:Y0070]);
}else{
  if([b:X0071] == ON)
  {
    rst([b:GB30024]);
    rst([b:GB30025]);
    set([b:GB30026]);
    rst([b:Y0070]);
  }
}
}

```

**Base screen 30020 to 30029, Base screen 30031, Base screen 30053 to 30054, Base screen 30057 to 30058, Base screen 30061 to 30062**

Script No.	30086	Script name	Script30086
Comment	Pressure Sensor Init Value Store		
Data type	Signed BIN16	Trigger type	ON, GB30032

// Store the initial value of the alarm bit switching of the pressure sensor.

```
[w:GD10572] = [w:GD11501];
```

```
rst([b:GB30032]);
```

**Base screen 30100**

Script No.	30062	Script name	Script30062
Comment	B-30100 iQSS BK Set List Read		
Data type	Signed BIN16	Trigger type	ON, GB54010

//Start from Project Script  
 //Read the iQSS Backup Setting list and the iQSS Backup History list from Recipe.

//Check if the A drive is accessible.

```

if([b:GS251.b0] == ON)
{
  set([b:GB54016]); //Recipe No.30001 Write Trigger
  [w:GD54090] = 1; //Recipe No.30001 Record No.

  [w:GD54004] = 1; //Next Data Acquisition Flag
}else{

```

rst([b:GB54014]); //Screen Switching Switch Action Conditions OFF [w:GD54006] = 32015; [w:GD62007] = 30101; } rst([b:GB54010]);			
Script No.	30063	Script name	Script30063
Comment	B-30100 Recipe W Trigger OFF		
Data type	Signed BIN16	Trigger type	ON, GD54012.b4
//Turn OFF Write Trigger of Recipe  //If the flag was on, read the next data. if([w:GD54004] == 1) { set([b:GB54011]); }else{ rst([b:GB54014]); //Screen Switching Switch Action Conditions OFF }  if([w:GD54013] == 30001) { rst([b:GB54016]); //Recipe No.30001 Write Trigger }else{ rst([b:GB54018]); //Recipe No.30002 Write Trigger }  rst([b:GD54012.b4]);			
Script No.	30064	Script name	Script30064
Comment	B-30100 iQSS BK Hist.List Read		
Data type	Signed BIN16	Trigger type	ON, GB54011
//Read iQSS Backup History  [w:GD54091] = 1; //Recipe No.30002 Record No. set([b:GB54018]); //Recipe No.30002 Write Trigger  [w:GD54004] = 0; //Flag Clear  rst([b:GB54011]);			

#### Base screen 30101

Script No.	30046	Script name	Script30046
Comment	B-30101 iQSS BK Setting Transfer		
Data type	Signed BIN16	Trigger type	ON, GB54000
//Transfer iQSS parameters from the iQSS Backup Setting list to the editor area. [w:TMP0000] = [w:GD54007] * 21; [w:GD54100[w:TMP0000]] = [w:GD54007] + 1; bmov([w:GD54100[w:TMP0000]], [w:GD54015], 20); [w:GD54086] = [w:GD54120[w:TMP0000]];			
//If target model is specified, assign target model and I/O No. if([w:GD54075] != 0) { [w:GD54016] = [w:GD54075]; //Target Model  //Execute only when target model is AnyWireASLINK. if([w:GD54075] == 1) { set([b:GD54086.b0]); //Flag to Select Output, Input/Combined } }  //Folder Numbers Fixed [w:GD54029] = -2; //Folder Numbers Automatic Acquisition			

[w:GD62000] = 30102; //Base Screen Switching			
rst([b:GB54000]);			
Script No.	30057	Script name	Script30057
Comment	B-30101 Data Delete		
Data type	Signed BIN16	Trigger type	ON, GB54006
//Delete the specified data from the iQSS Backup Setting list.  //Check if the data is selected if([w:GD54000] != 0) { //Offset [w:TMP0000] = [w:GD54007] * 21;  //If no data exists in the selected place, do not delete if([w:GD54101[w:TMP0000]] != 0) { //Clear Data [w:TMP0002] = 0; fmov([w:TMP0002],[w:GD54100[w:TMP0000]],21);  [w:GD54090] = 1; //Recipe No.30001 Record No. set([b:GB54017]); //Recipe No.30001 Read Trigger }else{ [w:GD54006] = 32012; [w:GD62007] = 30101; } }else{ [w:GD54006] = 32012; [w:GD62007] = 30101; }  rst([b:GB54006]);			
Script No.	30051	Script name	Script30051
Comment	iQSS BK/RS Use Right Info Check		
Data type	Signed BIN16	Trigger type	Sampling, 1 Sec
//Periodically check the status of obtaining the right to use. if([w:SD1435] == [w:SD1436]) { set([b:GB54004]); //iQSS Backup Permission Bit ON }else{ rst([b:GB54004]); //iQSS Backup Permission Bit OFF } 			
Script No.	30046	Script name	Script30046
Comment	B-30101 iQSS BK Setting Transfer		
Data type	Signed BIN16	Trigger type	ON, GD54012.b5
//Transfer iQSS parameters from the iQSS Backup Setting list to the editor area. [w:TMP0000] = [w:GD54007] * 21; [w:GD54100[w:TMP0000]] = [w:GD54007] + 1; bmov([w:GD54100[w:TMP0000]],[w:GD54015],20); [w:GD54086] = [w:GD54120[w:TMP0000]];			
//If target model is specified, assign target model and I/O No. if([w:GD54075] != 0) { [w:GD54016] = [w:GD54075]; //Target Model [w:GD54030] = [w:GD54089]; //I/O No.  //Execute only when target model is AnyWireASLINK. if([w:GD54075] == 1) { set([b:GD54086.b0]); //Flag to Select Output, Input/Combined			

}			
}			
//Folder Numbers Fixed			
[w:GD54029] = -2; //Folder Numbers Automatic Acquisition			
[w:GD62000] = 30102; //Base Screen Switching			
rst([b:GB54000]);			
Script No.	30050	Script name	Script30050
Comment	B-30101 iQSS BK Preprocessing 1		
Data type	Signed BIN16	Trigger type	ON, GB54013
//iQSS Backup Preprocessing			
//Offset			
[w:TMP0000] = [w:GD54007] * 21;			
//Check if the A drive is accessible.			
if([b:GS251.b0] == ON)			
{			
//If module type is 0, no settings are entered and backup will not be executed.			
if([w:GD54101[w:TMP0000]] != 0)			
{			
//Check the number of backed up cases.			
//When exceeding 100 cases, display a message before making backup.			
if([w:GD54101[w:TMP0000]] == 1)			
{			
//AnyWireASLINK			
if([w:GD54087] == 100)			
{			
[w:GD54006] = 1080;			
}else{			
[w:GD54006] = 1068;			
}			
}else{			
//CC-Link			
if([w:GD54088] == 100)			
{			
[w:GD54006] = 1081;			
}else{			
[w:GD54006] = 1068;			
}			
}			
[w:GD62007] = 30102;			
}else{			
[w:GD54006] = 32007;			
[w:GD62007] = 30101;			
}			
}else{			
[w:GD54006] = 32017;			
[w:GD62007] = 30101;			
}			
rst([b:GB54013]);			
Script No.	30052	Script name	Script30052
Comment	B-30101 iQSS BK Preprocessing 2		
Data type	Signed BIN16	Trigger type	ON, GB54002



//Transfer Specified Contents to PLC Devices

//Setting Name

bmov([w:GD54102[w:TMP0000]], [w:GD54017], 12);

//SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting

[w:SD1437] = (([w:GD54116[w:TMP0000]] & 0x00FF) << 8) + ([w:GD54101[w:TMP0000]] & 0x00FF);

//Execution Unit Display

[w:GD54016] = [w:GD54101[w:TMP0000]];

//SD1438: iQSS Backup/Restoration Target Folder No.

[w:SD1438] = [w:GD54114[w:TMP0000]];

//SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)

[w:SD1439] = [w:GD54115[w:TMP0000]];

//SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link: Station No.)

[w:SD1440] = [w:GD54117[w:TMP0000]];

//SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link: Sub Station No.)

[w:SD1441] = [w:GD54118[w:TMP0000]];

//SD1444: iQSS Backup/Restoration Action Setting

[w:SD1444] = [w:GD54119[w:TMP0000]];

//If the right to use the iQSS backup is obtained, start backup.

if([b:GB54004] == ON) && ([w:SD1446] == 0x0001)

{

set([b:SM1436]);

[w:GD62000] = 30103;

}else{

[w:GD54006] = 32006;

[w:GD62007] = 30101;

}

rst([b:GB54002]);

Script No.	30059	Script name	Script30059
Comment	Process at Screen Switching		
Data type	Signed BIN16	Trigger type	When closing a screen

//Clear Flags

rst([b:GB54008]);

rst([b:GB54009]);

//Close Overlap Window

[w:GD62001] = 0;

[w:GD62004] = 0;

[w:GD62007] = 0;

### Base screen 30102

Script No.	30047	Script name	Script30047
Comment	B-30102 iQSS Backup Setting Save		
Data type	Signed BIN16	Trigger type	ON, GB54001

//Save iQSS Backup Setting of Device to Recipe

//Check if the A drive is accessible.

if([b:GS251.b0] == ON)

{

//Check if target model is specified.

if([w:GD54016] != 0)

{

```

//Check whether the execution unit is specified.
if([w:GD54031] != 0)
{
    //Check whether the correct value is in Station No. and Station Sub.
    if(([w:GD54031] == 1) || (([w:GD54031] == 2) && ([w:GD54032] >= 0)) || (([w:GD54031] >= 3) &&
    ([w:GD54032] >= 0) && ([w:GD54033] >= 0))))
    {
        //If all checks are OK, save to recipe.
        [w:TMP0000] = ([w:GD54015] - 1) * 21;
        bmov([w:GD54015],[w:GD54100[w:TMP0000]],20);
        [w:GD54120[w:TMP0000]] = [w:GD54086];
        [w:GD54090] = 1; //Recipe No.30001 Record No.
        set([b:GB54017]); //Recipe No.30001 Read Trigger

    }else{
        [w:GD54006] = 32011;
        [w:GD62007] = 30101;
    }
    }else{
        [w:GD54006] = 32011;
        [w:GD62007] = 30101;
    }
    }else{
        [w:GD54006] = 32009;
        [w:GD62007] = 30101;
    }
    }else{
        [w:GD54006] = 32017;
        [w:GD62007] = 30101;
    }
}
rst([b:GB54001]);

```

Script No.	30067	Script name	Script30067
Comment	B-30102 Recipe Trigger OFF		
Data type	Signed BIN16	Trigger type	ON, GD54012.b5

//Turn OFF Read Trigger of Recipe

rst([b:GB54017]); //Recipe No.30001 Read Trigger

[w:GD54006] = 32016;  
[w:GD62007] = 30101;

rst([b:GD54012.b5]);

Script No.	30054	Script name	Script30054
Comment	B-30102 Comment Display Control		
Data type	Signed BIN16	Trigger type	Ordinary

//Switch Comments Between CC-Link and ASLINK

```

switch([w:GD54016])
{
    //ASLINK
    case 1: [w:GD54035] = 1026; //ID or Station
        [w:GD54036] = 1032; //All IDs or All Stations
        [w:GD54037] = 1033; //ID Specification or Station Specification
        [w:GD54038] = 1085; //Not Specified or Station Sub
        [w:GD54039] = 0; //Not Specified or Station Sub Specified
        break;

    //CC-Link
    case 2: [w:GD54035] = 1025; //ID or Station
        [w:GD54036] = 1042; //All IDs or All Stations
        [w:GD54037] = 1043; //ID Specification or Station Specification
        [w:GD54038] = 1027; //Not Specified or Station Sub
        [w:GD54039] = 1044; //Not Specified or Station Sub Specified

```

<pre> break;  //Other default: [w:GD54035] = 0; //ID or Station         [w:GD54036] = 0; //All IDs or All Stations         [w:GD54037] = 0; //ID Specification or Station Specification         [w:GD54038] = 0; //Not Specified or Station Sub         [w:GD54039] = 0; //Not Specified or Station Sub Specified         break; } </pre>			
Script No.	30053	Script name	Script30053
Comment	B-30102 I/O Judgment		
Data type	Signed BIN16	Trigger type	Rise/Fall, GD54086.b2
<pre> //While using AnyWireASLINK, turn OFF/ON the 9th bit depending on the unit type.  if([b:GD54086.b2] == ON) {     [w:GD54032] = [w:GD54032]   0x0200; //Input/Combined }else{     [w:GD54032] = [w:GD54032] &amp; 0x00FF; //Output } </pre>			

### Base screen 30103

Script No.	30055	Script name	Script30055
Comment	B-30103,30106 Scr. Start Process		
Data type	Signed BIN16	Trigger type	ON, GB40
<pre> //Make various settings when starting screen  //Actions at Screen Startup Only if([b:GB54008] == OFF) {     //Backup Execution Date/Time     [w:GD54040] = [w:GS650];     [w:GD54041] = [w:GS651];     [w:GD54042] = [w:GS652];      set([b:GB54008]); } </pre>			
Script No.	30060	Script name	Script30060
Comment	B-30103 Normal End		
Data type	Signed BIN16	Trigger type	Rise, SM1437
<pre> //After iQSS backup is completed, save results to Recipe.  //When ASLINK or CC-Link exceeding 100 cases, delete the oldest backup history. //If the screen was created as a dedicated screen, it does not work. if([w:GD54075] == 0) {     if(([w:GD54087] == 100) &amp;&amp; ([w:SD1437] &amp; 0x00FF) == 1)     {         [w:TMP0001] = 0;         [w:TMP0003] = 0;         //Search for ASLINK history from all history.         while([w:TMP0001] &lt;= 199)         {             [w:TMP0002] = [w:TMP0001] * 27; //Offset             if([w:GD56201[w:TMP0002]] == 1)             {                 [w:TMP0003] = [w:TMP0003] + 1;                 if([w:TMP0003] == 100)                 {                     //If history reached 100th case, exit loop.                     break;                 }             }         }     } } </pre>			

```

    }
    [w:TMP0001] = [w:TMP0001] + 1;
}

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD56200[w:TMP0002]],27);

//Move the data after deleted portion up.
while([w:TMP0001] < 199)
{
    [w:TMP0002] = [w:TMP0001] * 27; //Offset
    [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy Destination Offset

    bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

    [w:TMP0001] = [w:TMP0001] + 1;
}

}else{
    if([w:GD54088] == 100) && ([w:SD1437] & 0x00FF) == 2)
    {
        [w:TMP0001] = 0;
        [w:TMP0003] = 0;
        //Search for CC-Link history from all history.
        while([w:TMP0001] <= 199)
        {
            [w:TMP0002] = [w:TMP0001] * 27; //Offset
            if([w:GD56201[w:TMP0002]] == 2)
            {
                [w:TMP0003] = [w:TMP0003] + 1;
                if([w:TMP0003] == 100)
                {
                    //If history reached 100th case, exit loop.
                    break;
                }
            }
            [w:TMP0001] = [w:TMP0001] + 1;
        }
    }

    [w:TMP0004] = 0;
    fmov([w:TMP0004],[w:GD56200[w:TMP0002]],27);

    //Move the data after deleted portion up.
    while([w:TMP0001] < 199)
    {
        [w:TMP0002] = [w:TMP0001] * 27; //Offset
        [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy Destination Offset

        bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

        [w:TMP0001] = [w:TMP0001] + 1;
    }
}

}

//Offset
if([w:GD54075] == 1) || ([w:GD54075] == 2)
{
    //ASLINK or CC-Link Dedicated Screen
    [w:TMP0001] = 99;
}else{
    //General Screen
    [w:TMP0001] = 199;
}

```

```

}

//Store the iQSS backup result in descending order.
while([w:TMP0001] > 0)
{
    [w:TMP0000] = [w:TMP0001] * 27;
    [w:TMP0002] = ([w:TMP0001] - 1) * 27;

    bmov([w:GD56200[w:TMP0002]], [w:GD56200[w:TMP0000]], 27);
    [w:GD56200[w:TMP0000]] = [w:TMP0001] + 1;

    [w:TMP0001] = [w:TMP0001] - 1;
}

//No.
[w:GD56200] = 1;

//SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting
[w:GD56201] = [w:SD1437] & 0x00FF;
[w:GD56216] = ([w:SD1437] & 0xFF00) >> 8;

//Setting Name
bmov([w:GD54017], [w:GD56202], 12);

//SD1451: iQSS Backup Folder No.
[w:GD56214] = [w:SD1451];

//SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)
[w:GD56215] = [w:SD1439];

//SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link: Station No.)
[w:GD56217] = [w:SD1440];

//SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link: Sub Station No.)
[w:GD56218] = [w:SD1441];

//SD1444: iQSS Backup/Restoration Action Setting
[w:GD56219] = [w:SD1444];

//iQSS Backup Execution Date/Time
[w:GD56220] = [w:GD54040];
[w:GD56221] = [w:GD54041];
[w:GD56222] = [w:GD54042];

//SD1447: iQSS Backup/Restoration Number of Target Devices
[w:GD56223] = [w:SD1447];

//SD1448: iQSS Backup/Restoration Number of Normal Completion Devices
[w:GD56224] = [w:SD1448];

//SD1449: iQSS Backup/Restoration Number of Abnormal Completion Devices
[w:GD56225] = [w:SD1449];

//iQSS Backup Cases
if([w:GD56201] == 1)
{
    //AnyWireASLINK
    if([w:GD54087] >= 100)
    {
        [w:GD54087] = 100;
    }else{
        [w:GD54087] = [w:GD54087] + 1;
    }
}

```

```

}else{
  //CC-LINK
  if([w:GD54088] >= 100)
  {
    [w:GD54088] = 100;
  }else{
    [w:GD54088] = [w:GD54088] + 1;
  }
}

```

```

//Save to Recipe
[w:GD54091] = 1; //Recipe No.30002 Record No.
set([b:GB54019]); //Recipe No.30002 Read Trigger

```

```

//Change messages depending on whether canceled or not.
if([b:SM1442] == ON)
{
  [w:GD54006] = 32002; //Backup Cancel Message
}else{
  [w:GD54006] = 32000; //Backup Successful Message
}
[w:GD62007] = 30101; //Notification Dialog

```

Script No.	30043	Script name	Script30043
Comment	B-30103 Abnormal End		
Data type	Signed BIN16	Trigger type	ON, SM1438

```

//After iQSS backup is completed, save results to Recipe.

```

```

if([b:GB54009] == OFF)
{
  //If no folders were created, do not save history.
  if(([w:SD1447] != 0) && ([w:SD1448] != 0))
  {
    //When ASLINK or CC-Link exceeding 100 cases, delete the oldest backup history.
    //If the screen was created as a dedicated screen, it does not work.
    if([w:GD54075] == 0)
    {
      if(([w:GD54087] == 100) && ([w:SD1437] & 0x00FF) == 1))
      {
        [w:TMP0001] = 0;
        [w:TMP0003] = 0;
        //Search for ASLINK history from all history.
        while([w:TMP0001] <= 199)
        {
          [w:TMP0002] = [w:TMP0001] * 27; //Offset
          if([w:GD56201[w:TMP0002]] == 1)
          {
            [w:TMP0003] = [w:TMP0003] + 1;
            if([w:TMP0003] == 100)
            {
              //If history reached 100th case, exit loop.
              break;
            }
          }
          [w:TMP0001] = [w:TMP0001] + 1;
        }

        [w:TMP0004] = 0;
        fmov([w:TMP0004],[w:GD56200[w:TMP0002]],27);

        //Move the data after deleted portion up.
        while([w:TMP0001] < 199)
        {
          [w:TMP0002] = [w:TMP0001] * 27; //Offset
          [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy Destination Offset

```

```

        bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

        [w:TMP0001] = [w:TMP0001] + 1;
    }

} else {
    if(([w:GD54088] == 100) && ([w:SD1437] & 0x00FF) == 2)
    {
        [w:TMP0001] = 0;
        [w:TMP0003] = 0;
        //Search for CC-Link history from all history.
        while([w:TMP0001] <= 199)
        {
            [w:TMP0002] = [w:TMP0001] * 27; //Offset
            if([w:GD56201[w:TMP0002]] == 2)
            {
                [w:TMP0003] = [w:TMP0003] + 1;
                if([w:TMP0003] == 100)
                {
                    //If history reached 100th case, exit loop.
                    break;
                }
            }
            [w:TMP0001] = [w:TMP0001] + 1;
        }

        [w:TMP0004] = 0;
        fmov([w:TMP0004], [w:GD56200[w:TMP0002]], 27);

        //Move the data after deleted portion up.
        while([w:TMP0001] < 199)
        {
            [w:TMP0002] = [w:TMP0001] * 27; //Offset
            [w:TMP0005] = ([w:TMP0001] + 1) * 27; //Copy Destination Offset

            bmov([w:GD56200[w:TMP0005]], [w:GD56200[w:TMP0002]], 27);

            [w:TMP0001] = [w:TMP0001] + 1;
        }
    }
}

//Offset Calculation
if(([w:GD54075] == 1) || ([w:GD54075] == 2))
{
    //ASLINK or CC-Link Dedicated Screen
    [w:TMP0001] = 99;
} else {
    //General Screen
    [w:TMP0001] = 199;
}

//Store the iQSS backup result in descending order.
while([w:TMP0001] > 0)
{
    [w:TMP0000] = [w:TMP0001] * 27;
    [w:TMP0002] = ([w:TMP0001] - 1) * 27;

    bmov([w:GD56200[w:TMP0002]], [w:GD56200[w:TMP0000]], 27);
    [w:GD56200[w:TMP0000]] = [w:TMP0001] + 1;

    [w:TMP0001] = [w:TMP0001] - 1;
}

```

```

}

//No.
[w:GD56200] = 1;

//SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting
[w:GD56201] = [w:SD1437] & 0x00FF;
[w:GD56216] = ([w:SD1437] & 0xFF00) >> 8;

//Setting Name
bmov([w:GD54017],[w:GD56202],12);

//SD1451: iQSS Backup Folder No.
[w:GD56214] = [w:SD1451];

//SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)
[w:GD56215] = [w:SD1439];

//SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link: Station
No.)
[w:GD56217] = [w:SD1440];

//SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link: Sub
Station No.)
[w:GD56218] = [w:SD1441];

//SD1444: iQSS Backup/Restoration Action Setting
[w:GD56219] = [w:SD1444];

//iQSS Backup Execution Date/Time
[w:GD56220] = [w:GD54040];
[w:GD56221] = [w:GD54041];
[w:GD56222] = [w:GD54042];

//SD1447: iQSS Backup/Restoration Number of Target Devices
[w:GD56223] = [w:SD1447];

//SD1448: iQSS Backup/Restoration Number of Normal Completion Devices
[w:GD56224] = [w:SD1448];

//SD1449: iQSS Backup/Restoration Number of Abnormal Completion Devices
[w:GD56225] = [w:SD1449];

//iQSS Backup Cases
if([w:GD56201] == 1)
{
    //AnyWireASLINK
    if([w:GD54087] >= 100)
    {
        [w:GD54087] = 100;
    }else{
        [w:GD54087] = [w:GD54087] + 1;
    }
}else{
    //CC-Link
    if([w:GD54088] >= 100)
    {
        [w:GD54088] = 100;
    }else{
        [w:GD54088] = [w:GD54088] + 1;
    }
}
}

//Save to Recipe.

```



```

[w:GD54091] = 1; //Recipe No.30002 Record No.
set([b:GB54019]); //Recipe No.30002 Read Trigger
}

if([w:SD1452] != 0)
{
    if([w:SD1452] == 16894)
    {
        [w:GD54006] = 32001; //Backup Failure Message
        [w:GD62007] = 30101; //Notification Dialog
    }else{
        [w:GD54006] = [w:SD1452]; //Backup Failure Message
        [w:GD62007] = 30101; //Notification Dialog
    }
}
}else{
    if([w:SD1453] != 0)
    {
        [w:GD54006] = [w:SD1453]; //Backup Failure Message
        [w:GD62007] = 30101; //Notification Dialog
    }else{
        [w:GD54006] = 32001; //Backup Failure Message
        [w:GD62007] = 30101; //Notification Dialog
    }
}
}

set([b:GB54009]);
}

```

Script No.	30048	Script name	Script30048
Comment	Recipe Trigger OFF		
Data type	Signed BIN16	Trigger type	ON, GD54012.b5

//Turn OFF Read Trigger of Recipe

```

if([w:GD54013] == 30001)
{
    rst([b:GB54017]); //Recipe No.30001 Read Trigger
}else{
    rst([b:GB54019]); //Recipe No.30002 Read Trigger
}

```

rst([b:GD54012.b5]);

Script No.	30037	Script name	Script30037
Comment	Process at Screen Switching		
Data type	Signed BIN16	Trigger type	When closing a screen

//Clear Flags

```

Rst([b:GB54008]);
rst([b:GB54009]);

```

//Close Overlap Window

```

[w:GD62001] = 0;
[w:GD62004] = 0;

```

Script No.	30059	Script name	Script30059
Comment	Process at Screen Switching		
Data type	Signed BIN16	Trigger type	When closing a screen

//Clear Flags

```

rst([b:GB54008]);
rst([b:GB54009]);

```

//Close Overlap Window

[w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0;			
Script No.	30065	Script name	Script30065
Comment	B-30103,30106 iQSS BK/RS Cancel		
Data type	Signed BIN16	Trigger type	ON, GB54002
//Abort iQSS Backup or iQSS Restoration set([b:SM1442]);//iQSS Backup/Restoration Cancel Request  rst([b:GB54002]);			

#### Base screen 30104

Script No.	30056	Script name	Script30056
Comment	B-30104 iQSS Restore Set Check		
Data type	Signed BIN16	Trigger type	ON, GB54005
//Check iQSS backup contents. [w:TMP0000] = ([w:GD54008] * 27) + [w:GD54003]; bmov([w:GD56200[w:TMP0000]], [w:GD54043], 20);  [w:GD62000] = 30105;  rst([b:GB54005]);			
Script No.	30058	Script name	Script30058
Comment	B-30104 Data Delete		
Data type	Signed BIN16	Trigger type	ON, GB54006
//After deleting the specified data from the iQSS Backup History list, move the rest of the list up.  //Check whether to delete all or delete specified range cases. if([b:GB54012] == OFF) { //Check if the data is selected if([w:GD54002] != 0) { //Offset [w:TMP0000] = ([w:GD54008] * 27) + [w:GD54003]; //Copy Source Offset [w:TMP0001] = (([w:GD54008] + 1) * 27) + [w:GD54003];  //If no backup data exists in the selected place, do not process if([w:GD56201[w:TMP0000]] != 0) { //Reduce Number of Backup Cases if(([w:GD56223[w:TMP0000]] != 0) && ([w:GD56223[w:TMP0000]] == [w:GD56224[w:TMP0000]])) { if([w:GD56201[w:TMP0000]] == 1) { //AnyWireASLINK if([w:GD54087] > 0) { [w:GD54087] = [w:GD54087] - 1; }else{ [w:GD54087] = 0; } }else{ //CC-LINK if([w:GD54088] > 0) { [w:GD54088] = [w:GD54088] - 1; }else{ [w:GD54088] = 0; } } } } } }			

```

//Calculate the Number of Processing Counts to Repeat
[w:TMP0003] = [w:TMP0000] / 27;
if([w:GD54075] == 1) || ([w:GD54075] == 2))
{
    //ASLINK or CC-Link Dedicated Screen
    [w:TMP0002] = 99;
}else{
    //General Screen
    [w:TMP0002] = 199;
}

while([w:TMP0003] <= [w:TMP0002])
{
    if([w:TMP0003] != [w:TMP0002])
    {
        //Overwrite Data
        bmov([w:GD56200[w:TMP0001]], [w:GD56200[w:TMP0000]], 27);
        //Adjust the No. to the current No. If No. is 0 or less, fix it to 0.
        if([w:GD56200[w:TMP0000]] > 0)
        {
            [w:GD56200[w:TMP0000]] = [w:GD56200[w:TMP0000]] - 1;
        }else{
            [w:GD56200[w:TMP0000]] = 0;
        }

        //Add Offset
        [w:TMP0000] = [w:TMP0000] + 27;
        [w:TMP0001] = [w:TMP0001] + 27;

        //Add Count
        [w:TMP0003] = [w:TMP0003] + 1;
    }else{
        //Set the last data to 0
        [w:TMP0004] = 0;

        if([w:GD54075] == 1) || ([w:GD54075] == 2))
        {
            //ASLINK or CC-Link Dedicated Screen
            fmov([w:TMP0004], [w:GD58873], 27);
        }else{
            //General Screen
            fmov([w:TMP0004], [w:GD61573], 27);
        }
        [w:TMP0000] = [w:TMP0000] + 27;

        //Add Count
        [w:TMP0003] = [w:TMP0003] + 1;
    }
}

//Save to Recipe
[w:GD54091] = 1; //Recipe No.30002 Record No.
set([b:GB54019]); //Recipe No.30002 Read Trigger

}else{
    [w:GD54006] = 32012;
    [w:GD62007] = 30101;
}
}else{
    [w:GD54006] = 32012;
    [w:GD62007] = 30101;
}
}
}else{

```

```
//All Delete
[w:TMP0000] = 0;
fmov([w:TMP0000],[w:GD56200],5400);
```

```
//Clear Backup Cases
//AnyWireASLINK
[w:GD54087] = 0;
//CC-Link
[w:GD54088] = 0;
```

```
//Save to Recipe
[w:GD54091] = 1; //Recipe No.30002 Record No.
set([b:GB54019]); //Recipe No.30002 Read Trigger
```

```
}
rst([b:GB54006]);
```

Script No.	30048	Script name	Script30048
Comment	Recipe Trigger OFF		
Data type	Signed BIN16	Trigger type	ON, GD54012.b5

```
//Turn OFF Read Trigger of Recipe
```

```
if([w:GD54013] == 30001)
{
  rst([b:GB54017]); //Recipe No.30001 Read Trigger
}else{
  rst([b:GB54019]); //Recipe No.30002 Read Trigger
}
```

```
rst([b:GD54012.b5]);
```

Script No.	30061	Script name	Script30061
Comment	B-30104 Restore Preprocessing		
Data type	Signed BIN16	Trigger type	ON, GB54002

```
//Transfer Specified Contents to PLC Devices
```

```
//Offset
[w:TMP0000] = ([w:GD54008] * 27) + [w:GD54003];

//If target model is 0, no settings are entered and restoration will not be executed.
if([w:GD56201[w:TMP0000]] != 0)
{
  //It's impossible to restore from the iQSS backup which does not end normally.
  if(([w:GD56223[w:TMP0000]] != 0) && ([w:GD56223[w:TMP0000]] == [w:GD56224[w:TMP0000]]))
  {
    //Setting Name
    bmov([w:GD56202[w:TMP0000]],[w:GD54045],24);

    //Target Device Comment Display
    [w:GD54016] = [w:GD56201[w:TMP0000]];

    //SD1437: iQSS Backup/Restoration Target Model/Execution Unit Setting
    [w:SD1437] = (([w:GD56216[w:TMP0000]] & 0x00FF) << 8) + ([w:GD56201[w:TMP0000]] & 0x00FF);

    //SD1438: iQSS Backup/Restoration Target Folder No.
    [w:SD1438] = [w:GD56214[w:TMP0000]];

    //SD1439: iQSS Backup/Restoration Target Setting (Target Module I/O No.)
    [w:SD1439] = [w:GD56215[w:TMP0000]];

    //SD1440: iQSS Backup/Restoration Target Setting (Target Device 1 ASLINK: ID No., CC-Link: Station No.)
    [w:SD1440] = [w:GD56217[w:TMP0000]];

    //SD1441: iQSS Backup/Restoration Target Setting (Target Device 2 ASLINK: Not Used, CC-Link: Sub
```

Station No.)

```
[w:SD1441] = [w:GD56218[w:TMP0000]];
```

```
//SD1444: iQSS Backup/Restoration Action Setting
```

```
[w:SD1444] = [w:GD56219[w:TMP0000]];
```

```
//If the right to use the iQSS backup is obtained, start backup.
```

```
if(([b:GB54004] == ON) && ([w:SD1446] == 0x0001))
```

```
{
```

```
    set([b:SM1439]);
```

```
    [w:GD62000] = 30106;
```

```
}else{
```

```
    [w:GD54006] = 32006;
```

```
    [w:GD62007] = 30101;
```

```
}
```

```
}else{
```

```
    [w:GD54006] = 32013;
```

```
    [w:GD62007] = 30101;
```

```
}
```

```
}else{
```

```
    [w:GD54006] = 32008;
```

```
    [w:GD62007] = 30101;
```

```
}
```

```
rst([b:GB54002]);
```

Script No.	30051	Script name	Script30051
Comment	iQSS BK/RS Use Right Info Check		
Data type	Signed BIN16	Trigger type	Sampling, 1 Sec

```
//Periodically check the status of obtaining the right to use.
```

```
if([w:SD1435] == [w:SD1436])
```

```
{
```

```
    set([b:GB54004]); //iQSS Backup Permission Bit ON
```

```
}else{
```

```
    rst([b:GB54004]); //iQSS Backup Permission Bit OFF
```

```
}
```

Script No.	30059	Script name	Script30059
Comment	Process at Screen Switching		
Data type	Signed BIN16	Trigger type	When closing a screen

```
//Clear Flags
```

```
rst([b:GB54008]);
```

```
rst([b:GB54009]);
```

```
//Close Overlap Window
```

```
[w:GD62001] = 0;
```

```
[w:GD62004] = 0;
```

```
[w:GD62007] = 0;
```

### Base screen 30105

Script No.	30066	Script name	Script30066
Comment	B-30105 Comment Display Control		
Data type	Signed BIN16	Trigger type	Ordinary

```
//Switch Comments Between CC-Link and ASLINK
```

```
switch([w:GD54044])
```

```
{
```

```
    //ASLINK
```

```
    case 1: [w:GD54070] = 1026; //ID or Station
```

```
        [w:GD54071] = 1032; //All IDs or All Stations
```

```
        [w:GD54072] = 1033; //ID Specification or Station Specification
```

```
        [w:GD54073] = 1085; //Not Specified or Station Sub
```

```
        [w:GD54074] = 0; //Not Specified or Station Sub Specified
```

```
        break;
```

```
//CC-Link
case 2: [w:GD54070] = 1025; //ID or Station
        [w:GD54071] = 1042; //All IDs or All Stations
        [w:GD54072] = 1043; //ID Specification or Station Specification
        [w:GD54073] = 1027; //Not Specified or Station Sub
        [w:GD54074] = 1044; //Not Specified or Station Sub Specified
        break;
}
```

#### Base screen 30106

Script No.	30055	Script name	Script30055
Comment	B-30103,30106 Scr.Start Process		
Data type	Signed BIN16	Trigger type	ON, GB40
<pre>//Make various settings when starting screen  //Actions at Screen Startup Only if([b:GB54008] == OFF) {     //Backup Execution Date/Time     [w:GD54040] = [w:GS650];     [w:GD54041] = [w:GS651];     [w:GD54042] = [w:GS652];      set([b:GB54008]); } </pre>			
Script No.	30044	Script name	Script30044
Comment	B-30106 Normal End		
Data type	Signed BIN16	Trigger type	Rise, SM1440
<pre>//IQSS Restoration Normal End Processing if([b:SM1442] == ON) {     [w:GD54006] = 32005; //IQSS Restoration Cancel }else{     [w:GD54006] = 32003; //IQSS Restoration Normal Completion } [w:GD62007]=30101; //Notification Dialog Display </pre>			
Script No.	30045	Script name	Script30045
Comment	B-30106 Abnormal End		
Data type	Signed BIN16	Trigger type	ON, SM1441
<pre>//IQSS Restoration Abnormal End Processing  if([b:GB54009] == OFF) {     if([w:SD1452] != 0)     {         if([w:SD1452] == 16894)         {             [w:GD54006] = 32004; //Restoration Failure Message             [w:GD62007] = 30101; //Notification Dialog         }else{             [w:GD54006] = [w:SD1452]; //Backup Failure Message             [w:GD62007] = 30101; //Notification Dialog         }     } }else{     if([w:SD1453] != 0)     {         [w:GD54006] = [w:SD1453]; //Backup Failure Message         [w:GD62007] = 30101; //Notification Dialog     }else{         [w:GD54006] = 32004; //Restoration Failure Message         [w:GD62007] = 30101; //Notification Dialog     } } </pre>			

<pre>     } }  set([b:GB54009]); } </pre>			
Script No.	30065	Script name	Script30065
Comment	B-30103,30106 iQSS BK/RS Cancel		
Data type	Signed BIN16	Trigger type	ON, GB54002
<pre> //Abort iQSS Backup or iQSS Restoration set([b:SM1442]); //iQSS Backup/Restoration Cancel Request  rst([b:GB54002]); </pre>			
Script No.	30059	Script name	Script30059
Comment	Process at Screen Switching		
Data type	Signed BIN16	Trigger type	When closing a screen
<pre> //Clear Flags  rst([b:GB54008]); rst([b:GB54009]);  //Close Overlap Window [w:GD62001] = 0; [w:GD62004] = 0; [w:GD62007] = 0; </pre>			

### 5.9.3 Object script

**Base screen 30006 to 30021, Base screen 30023 to 30025, Base screen 30027 to 30029, Base screen 30031 to 30042, Base screen 30051 to 30054, Base screen 30057 to 30058, Base screen 30061 to 30062**

Object	Word lamp *1		
Script user ID	1		
Data type	Unsigned BIN16	Trigger type	Ordinary
<pre> //Display ON/OFF Status [w:GD10551] = \$\$ &amp; [w:GD10535]; </pre>			

#### Base screen 30013

Object	Level *1		
Script user ID	1		
Data type	Signed BIN16	Trigger type	Ordinary
<pre> //Display ON/OFF Threshold Value screen_draw(0); redraw_object();  [w:TMP0200] = (object_width / 100) * [w:GD11501]; [w:TMP0201] = (object_width / 100) * [w:GD11503];  d_line([w:TMP0200],0,[w:TMP0200],object_height,0,1,224); //OFF / ON Threshold Value d_line([w:TMP0201],0,[w:TMP0201],object_height,0,1,3); //Alarm judgment (Hi) </pre>			

\*1: [Object script symbol] is used. For more details about [Object script symbol], please refer to "5.9.5 Object script symbol".

#### Base screen 30010 to 30012, Base screen 30014 to 30015, Base screen 30017 to 30018, Base screen 30032

Object	Level *1		
Script user ID	1		
Data type	Signed BIN16	Trigger type	Ordinary
<pre> //Display Threshold Value, Alarm Judgment(Hi), (Lo) screen_draw(0); redraw_object();  [w:TMP0200] = (object_width / 100) * [w:GD11501]; </pre>			

```
[w:TMP0201] = (object_width / 100) * [w:GD11503];
[w:TMP0202] = (object_width / 100) * [w:GD11504];

d_line([w:TMP0200],0,[w:TMP0200],object_height,0,1,131); //Threshold Value
d_line([w:TMP0201],0,[w:TMP0201],object_height,0,1,224); //Alarm Judgment(Hi)
d_line([w:TMP0202],0,[w:TMP0202],object_height,0,1,3); //Alarm Judgment(Lo)
```

\*1: [Object script symbol] is used. For more details about [Object script symbol], please refer to "5.9.5 Object script symbol".

#### Base screen 30010 to 30015, Base screen 30017, Base screen 30032

Object	Numerical input *1		
Script user ID	1		
Data type	Unsigned BIN16	Trigger type	Input Fixation
//When input is determined, redraw the screen. redraw_screen();			

\*1: Set for Threshold, Alarm judgment (Hi), Alarm judgment (Lo), Threshold: OFF->ON, Threshold: ON->OFF.

#### Base screen 30035

Object	Level *1		
Script user ID	1		
Data type	Signed BIN16	Trigger type	Ordinary
//Display Alarm threshold Value screen_draw(0); redraw_object();  [w:TMP0202] = (object_width / 100) * [w:GD11504];  d_line([w:TMP0202],0,[w:TMP0202],object_height,0,1,3); //Threshold Value			

\*1: [Object script symbol] is used. For more details about [Object script symbol], please refer to "5.9.5 Object script symbol".

#### Base screen 30101

Object	Word lamp		
Script user ID	1		
Data type	Signed BIN16	Trigger type	Ordinary
//Script to Display Comments if([w:GD54122[w:GD54001]] != 0) { [w:GD54077] = (([w:GD54122[w:GD54001]] * 10) + \$\$) + 1021; }else{ [w:GD54077] = 0; }			

The above script is set for all word lamps of the "Execution unit". The specified device varies depending on the object.

#### Window screen 30003

Object	Numerical display		
Script user ID	1		
Data type	Unsigned BIN16	Trigger type	Rise, GB40
//Obtain Today's Year & Month from Clock Data [w:TMP950] = [w:GS650] & 0xF000; //Obtain Tenths Digit of "Last 2-Digits of Year" from Clock Data for Setting [w:TMP960] = [w:TMP950] >> 12; //Decimal Alignment [w:TMP968] = [w:TMP960] * 10; //BCD->BIN [w:TMP951] = [w:GS650] & 0x0F00; //Obtain Ones Digit of "Last 2-Digits of Year" from Clock Data for Setting [w:TMP961] = [w:TMP951] >> 8; //BCD->BIN [w:TMP973] = 2000 + [w:TMP968] + [w:TMP961]; //Set Year to TMP973 as BIN [w:GD63990] = [w:TMP973]; //Set Year  [w:TMP952] = [w:GS650] & 0x00F0; //Obtain Tenths Digit of Month from Clock Data for Setting [w:TMP962] = [w:TMP952] >> 4; //Decimal Alignment [w:TMP969] = [w:TMP962] * 10; //BCD->BIN [w:TMP953] = [w:GS650] & 0x000F; //Obtain Ones Digit of Month from Clock Data for Setting			



```
[w:TMP974] = [w:TMP969] + [w:TMP953]; //Set Month to TMP974 as BIN
[w:GD63991] = [w:TMP974]; //Set Month

[w:TMP954] = [w:GS651] & 0xF000; //Obtain Tenths Digit of "Last 2-Digits of Day" from Clock Data for Setting
[w:TMP963] = [w:TMP954] >> 12; //Decimal Alignment
[w:TMP970] = [w:TMP963] * 10; //BCD->BIN
[w:TMP955] = [w:GS651] & 0x0F00; //Obtain Ones Digit of "Last 2-Digits of Day" from Clock Data for Setting
[w:TMP964] = [w:TMP955] >> 8; //BCD->BIN
[w:TMP975] = [w:TMP970] + [w:TMP964]; //Set Day to TMP975 as BIN
[w:GD63992] = [w:TMP975]; //Set Day
```

```
[w:TMP956] = [w:GS651] & 0x00F0; //Obtain Tenths Digit of Hour from Clock Data for Setting
[w:TMP965] = [w:TMP956] >> 4; //Decimal Alignment
[w:TMP971] = [w:TMP965] * 10; //BCD->BIN
[w:TMP957] = [w:GS651] & 0x000F; //Obtain Ones Digit of Hour from Clock Data for Setting
[w:TMP976] = [w:TMP971] + [w:TMP957]; //Set Year to TMP976 as BIN
[w:GD63993] = [w:TMP976]; //Set Hour
```

```
[w:TMP958] = [w:GS652] & 0xF000; //Obtain Tenths Digit of "Last 2-Digits of Minute" from Clock Data for Setting
[w:TMP966] = [w:TMP958] >> 12; //Decimal Alignment
[w:TMP972] = [w:TMP966] * 10; //BCD->BIN
[w:TMP959] = [w:GS652] & 0x0F00; //Obtain Ones Digit of "Last 2-Digits of Minute" from Clock Data for Setting
[w:TMP967] = [w:TMP959] >> 8; //BCD->BIN
[w:TMP977] = [w:TMP972] + [w:TMP967]; //Set Minute to TMP977 as BIN
[w:GD63994] = [w:TMP977]; //Set Minute
```

```
[w:TMP993] = [w:GS652] & 0x00F0; //Obtain Tenths Digit of Second from Clock Data for Setting
[w:TMP995] = [w:TMP993] >> 4; //Decimal Alignment
[w:TMP996] = [w:TMP995] * 10; //BCD->BIN
[w:TMP994] = [w:GS652] & 0x000F; //Obtain Ones Digit of Second from Clock Data for Setting
[w:TMP978] = [w:TMP996] + [w:TMP994]; //Set Second to TMP978 as BIN
[w:GD63995] = [w:TMP978]; //Set Second
```

Object	Numerical display		
Script user ID	2		
Data type	Unsigned BIN16	Trigger type	Ordinary

//BIN -> BCD Conversion

```
[w:TMP979] = [w:GD63990] - 2000; //Last 2-Digits of Year
```

```
[w:TMP980] = (([w:TMP979] / 10) << 4) + ([w:TMP979] % 10); //Year BIN -> BCD
[w:TMP981] = (([w:GD63991] / 10) << 4) + ([w:GD63991] % 10); //Month BIN -> BCD
[w:TMP982] = (([w:GD63992] / 10) << 4) + ([w:GD63992] % 10); //Day BIN -> BCD
[w:TMP983] = (([w:GD63993] / 10) << 4) + ([w:GD63993] % 10); //Hour BIN -> BCD
[w:TMP984] = (([w:GD63994] / 10) << 4) + ([w:GD63994] % 10); //Minute BIN -> BCD
[w:TMP985] = (([w:GD63995] / 10) << 4) + ([w:GD63995] % 10); //Second BIN -> BCD
```

Object	Numerical display		
Script user ID	3		
Data type	Unsigned BIN16	Trigger type	Ordinary

//Year & Month Setting

```
[w:GS513] = ([w:TMP980] << 8) + [w:TMP981]; //Set Year & Month to Change Time Device
```

Object	Numerical display		
Script user ID	4		
Data type	Unsigned BIN16	Trigger type	Ordinary

//Date & Time Setting

```
[w:GS514] = ([w:TMP982] << 8) + [w:TMP983]; //Set Date & Time to Change Time Device
```

Object	Numerical display		
Script user ID	5		
Data type	Unsigned BIN16	Trigger type	Ordinary

//Minute & Second Setting			
[w:GS515] = ([w:TMP984] << 8) + [w:TMP985]; //Set Minute & Second to Change Time Device			
Object	Numerical display		
Script user ID	6		
Data type	Unsigned BIN16	Trigger type	Ordinary
//Day of Week Setting			
[w:TMP986] = [w:GD63990]; //Year (BIN)			
[w:TMP987] = [w:GD63991]; //Month (BIN)			
[w:TMP988] = [w:GD63992]; //Day (BIN)			
if(([w:TMP987] == 1)    ([w:TMP987] == 2)){ // Correction Processing to Calculate January and February as 13th/14th Month			
[w:TMP986] = [w:TMP986] - 1; //Subtract 1 from Year			
[w:TMP987] = [w:TMP987] + 12; //Add 12 to Month			
}			
[w:TMP989] = [w:TMP986]/4; //Create Items Required for Zeller's Congruence			
[w:TMP990] = [w:TMP986]/100; //Create Items Required for Zeller's Congruence			
[w:TMP991] = [w:TMP986]/400; //Create Items Required for Zeller's Congruence			
[w:TMP992] = (13*[w:TMP987]+8)/5; //Create Items Required for Zeller's Congruence			
//Calculate Day of Week Using Zeller's Congruence and Set the Day to Change Time Device			
[w:GS516] = ([w:TMP986]+[w:TMP989]-[w:TMP990]+[w:TMP991]+[w:TMP992]+[w:TMP988])%7;			

#### 5.9.4 Script symbol

Symbol Name	Device and Constant	Remarks
Input_X	44	Upper-left-most X Coordinate of Input Display Area
Input_Y	165	Upper-left-most Y Coordinate of Input Display Area
Frame_X	620	Lower-right-most X Coordinate of I/O Display Area
Frame_Y	437	Lower-right-most Y Coordinate of I/O Display Area
Object_X	36	Word Lamp Width
Object_Y	17	Word Lamp Height
Master_1	0	I/O No. of Master Module 1
Master_2	2	I/O No. of Master Module 2
Master_3	4	I/O No. of Master Module 3
Master_4	6	I/O No. of Master Module 4

#### 5.9.5 Object script symbol

Symbol Name	Device and Constant	Remarks
object_width	124	Level Width
object_height	25	Level Height

## 6. OTHERS

### 6.1 User-Defined Name Registration

Follow the procedure below to register data.

- Output module: Set the "module ID + 1" value to Comment No. (The comment No. does not accept 0 so that the value of adding 1 to the ID is used as the comment No.)

15	Output module ID: 14	出力ユニット ID:14	輸出模块 ID:14							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Output module ID: 15	出力ユニット ID:15	輸出模块 ID:15							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	Output module ID: 16	出力ユニット ID:16	輸出模块 ID:16							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	Output module ID: 17	出力ユニット ID:17	輸出模块 ID:17			No	None	Regular		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Comment No. of output module ID 14 to ID 17

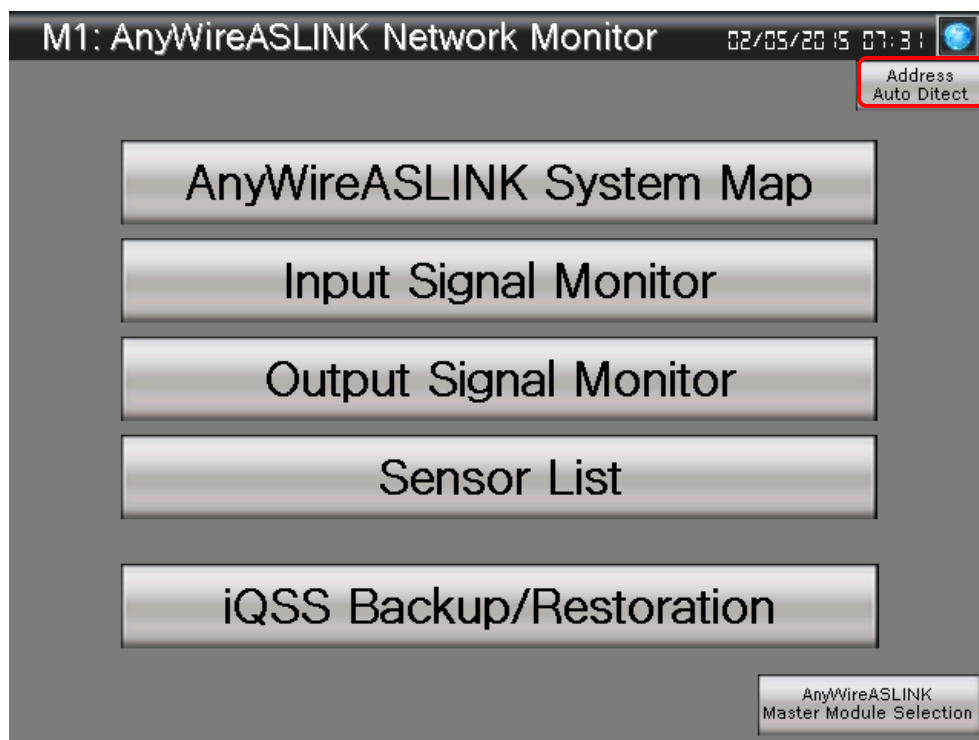
- Input module, I/O combined module: Set the value "module ID + 513" to Comment No. (The IDs of input or I/O combined modules are 0x0200 to 0x02FF in the buffer memory [connection ID information area] so that the value of adding 512(0x0200) and also adding 1 to the ID [in the same way as the output module] is used as the comment No.)

527	Input module ID: 14	入力ユニット ID:14	輸入模块 ID:14							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
528	Input module ID: 15	入力ユニット ID:15	輸入模块 ID:15							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
529	Input module ID: 16	入力ユニット ID:16	輸入模块 ID:16							<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
530	Input module ID: 17	入力ユニット ID:17	輸入模块 ID:17			No	None	Regular		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Comment No. of input module ID 14 to ID 17

### 6.2 Changing System Configuration

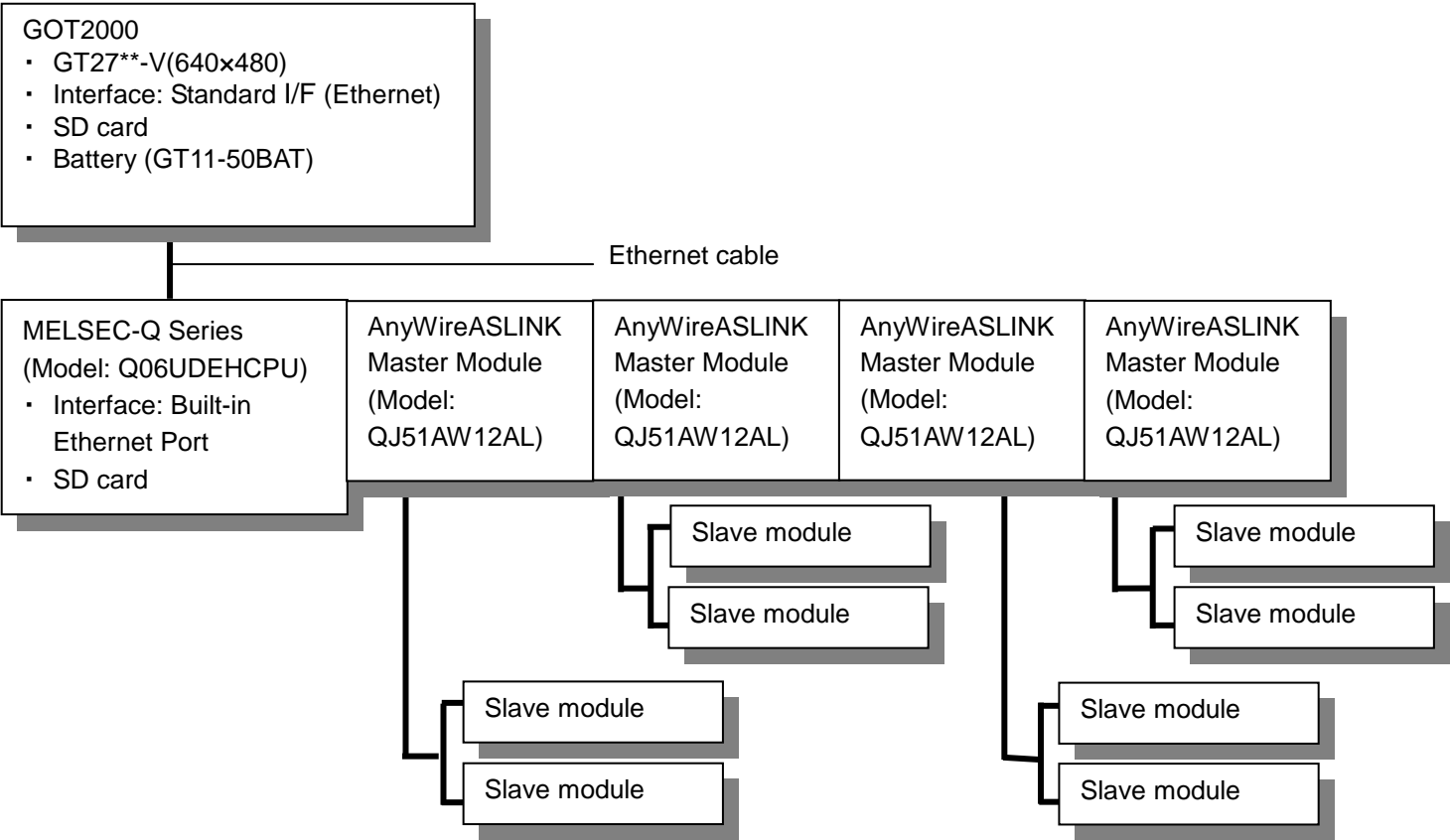
After booting a GOT, if operations such as slave module addition or deletion, or ID change were performed, make sure to return to the base screen B-30001 and touch the [Address Auto Detect] switch for a second or reboot the GOT.



6.3 Changing CPU

To use the MELSEC-Q Series CPU, change the system configuration and the Ethernet setting.

6.3.1 System configuration



6.3.2 Ethernet Setting of Screen Design Software

	Host	Net No.	Station	Unit Type	IP Address	Port No.	Communication
1	*	1	1	QnUD(P)V/QnUDEH	192.168.3.39	5006	UDP

## 6.4 Changing Start I/O Number

Follow the procedure below to change the start I/O number of the module.  
(Example: Changing the start I/O number from 0H to 80H)

### 6.4.1 How to apply changing Start I/O Number with Device Batch Edit

- (1) Start I/O Number of buffer memory

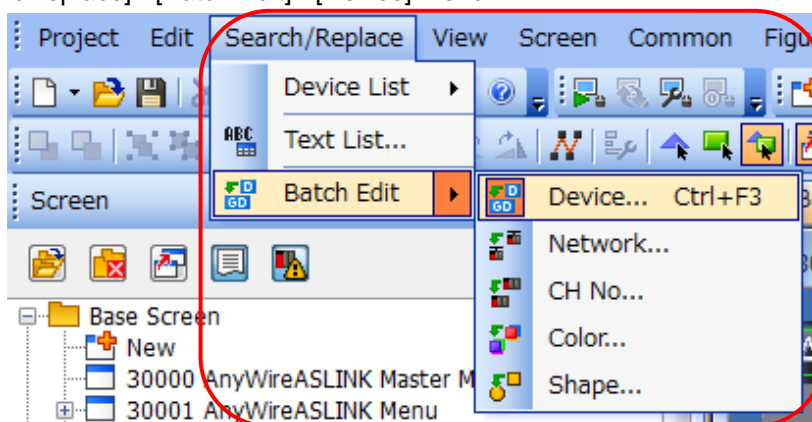
To change [Before]Un-G0 to [After]Um-G0, set [m] to the Start I/O Number for change divided by 16. To change 0H to 80H as Start I/O Number, change U00-G0 to U08-G0. Regarding [Point], specify a Point within minimum to maximum address of displayed settings. To change U00-G0 to U00-G99, set [Point] to 100.

- (2) Start I/O Number of I/O signal

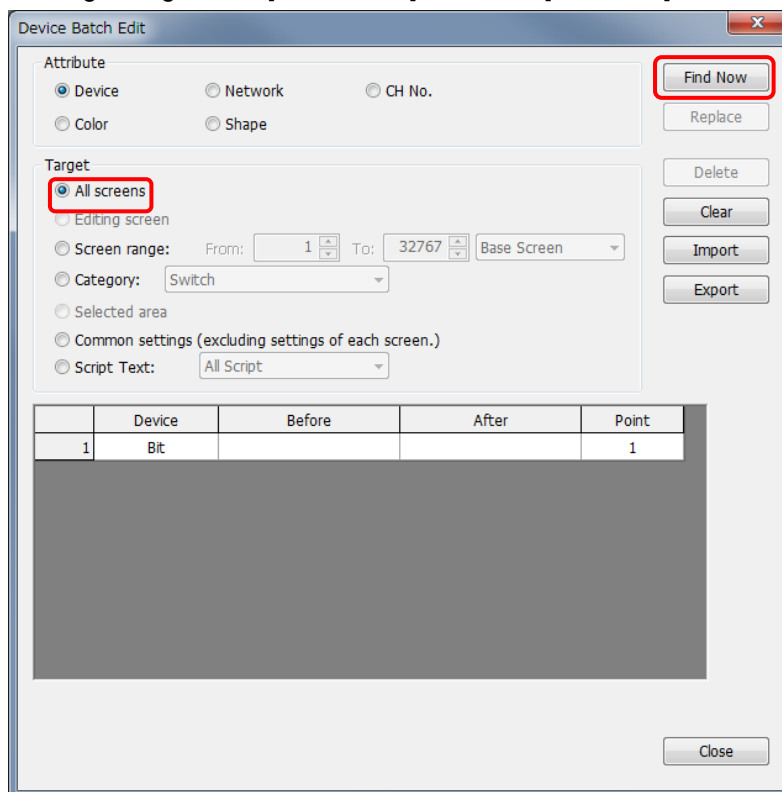
To change [Before]Xn0 to [After]Xm0, set [m] to the Start I/O Number for change. Regarding [Point], specify a Point within minimum to maximum address of displayed settings. To change within X00 to X1F, set [Point] to 32. AnyWireASLINK Master Modules occupy 32 points of I/O signal (Xn0 to X(n+1)F, Yn0 to Y(n+1)F) per slot. Please be careful not to make an unnecessary change when changing.

### 6.4.2 To change devices set in the screen

- (1) Select [Search/Replace] - [Batch Edit] - [Device] menu.



- (2) In the displayed setting dialog, select [All Screens], and click [Find Now].



(3) Set the [After] device and [Point], and execute the batch edit.

- Changing the start I/O number of the buffer memory

Set [Before] to U00-G0, [After] to U08-G0, and [Point] to 10257, and click [Replace]. U00-G0 to U00-G10256 will be changed to U08-G0 to U08-G10256.

	Device	Before	After	Point
1132	Word	SD1449	SD1449	1
1133	Word	SD1450	SD1450	1
1134	Word	U00-G0	U08-G0	10257
1135	Word	U00-G1	U00-G1	1
1136	Word	U00-G2	U00-G2	1
1137	Word	U00-G3	U00-G3	1
1138	Word	U00-G4	U00-G4	1
1139	Word	U00-G5	U00-G5	1
1140	Word	U00-G6	U00-G6	1

- Changing the start I/O number of the I/O signal

To change the output signal (Y device), set [Before] to Y0000, [After] to Y0080, and [Point] to 18, and click [Replace]. Y0000 to Y0011 will be changed to Y0080 to Y0091.

	Device	Before	After	Point
1102	Word	GD63995	GD63995	1
1103	Word	GD64000	GD64000	1
1104	Bit	Y0000	Y0080	18
1105	Bit	Y0001	Y0001	1
1106	Bit	Y0010	Y0010	1
1107	Bit	Y0011	Y0011	1
1108	Bit	Y0020	Y0020	1
1109	Bit	Y0021	Y0021	1
1110	Bit	Y0030	Y0030	1

### 6.4.3 To change devices set in [Common]

To change devices set in [GOT Environmental Setting] or in functions such as the recipe function, select [Common Settings] for the target of the Device Batch Edit. Follow the same procedures as in 6.4.2.

The devices to change by selecting [Common Settings] are U00-G0 to U00-G12288, Y0010.

- Changing the start I/O number of the buffer memory

Set [Before] to U00-G0, [After] to U08-G0, and [Point] to 12289, and click [Replace]. U00-G0 to U00-G12288 will be changed to U08-G0 to U08-G12288.

	Device	Before	After	Point
117	Bit	Y0050	Y0050	1
118	Bit	Y0070	Y0070	1
119	Word	U00-G0	U08-G0	12289
120	Word	U00-G1	U00-G1	1
121	Word	U00-G20	U00-G20	1
122	Word	U00-G4096	U00-G4096	1
123	Word	U00-G8192	U00-G8192	1
124	Word	U00-G8960	U00-G8960	1
125	Word	U00-G9216	U00-G9216	1

- Changing the start I/O number of the I/O signal

Set [Before] to Y0010, [After] to Y0090, and [Point] to 1, and click [Replace]. Y0010 will be changed to Y0090.

	Device	Before	After	Point
113	Word	GD62041	GD62041	1
114	Word	GD64000	GD64000	1
115	Bit	Y0010	Y0090	1
116	Bit	Y0030	Y0030	1
117	Bit	Y0050	Y0050	1
118	Bit	Y0070	Y0070	1
119	Word	U00-G0	U08-G0	12289
120	Word	U00-G1	U00-G1	1
121	Word	U00-G20	U00-G20	1

#### 6.4.4 To change devices use in [Script Text]

To change devices set in [Script Text], select [Script Text] for the target of the Device Batch Edit. Follow the same procedures as in 6.4.2.

The devices to change by selecting [Script Text] are U00-G8192 to U00-G10321, X0011 to X0012, Y0000 to Y0011.

- Changing the start I/O number of the buffer memory

Set [Before] to U00-G8192 [After] to U08-G8192, and [Point] to 2130, and click [Replace]. U00-G8192 to U00-G10321 will be changed to U08-G 8192 to U08-G10321.

Device Batch Edit

Attribute

☒ Device ☐ Network ☐ CH No.

☐ Color ☐ Shape

Find Now

Replace

Target

☐ All screens

☐ Editing screen

☐ Screen range: From: 1 To: 32767 Base Screen

☐ Category: Switch

☐ Selected area

☐ Common settings (excluding settings of each screen.)

☒ Script Text: All Script

Delete

Clear

Import

Export

	Device	Before	After	Point
319	Word	SD1452	SD1452	1
320	Word	SD1453	SD1453	1
321	Word	U00-G8192	U08-G8192	2130
322	Word	U00-G8960	U00-G8960	1
323	Word	U00-G9984	U00-G9984	1
324	Word	U00-G10256	U00-G10256	1
325	Word	U00-G10320	U00-G10320	1
326	Word	U00-G10321	U00-G10321	1
327	Word	U02-G8192	U02-G8192	1

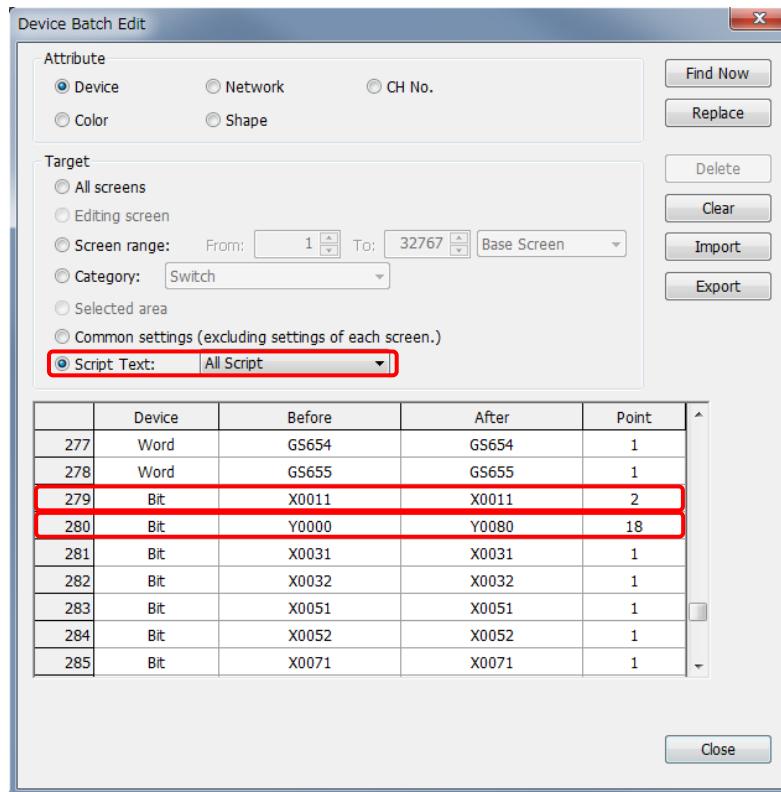
Close



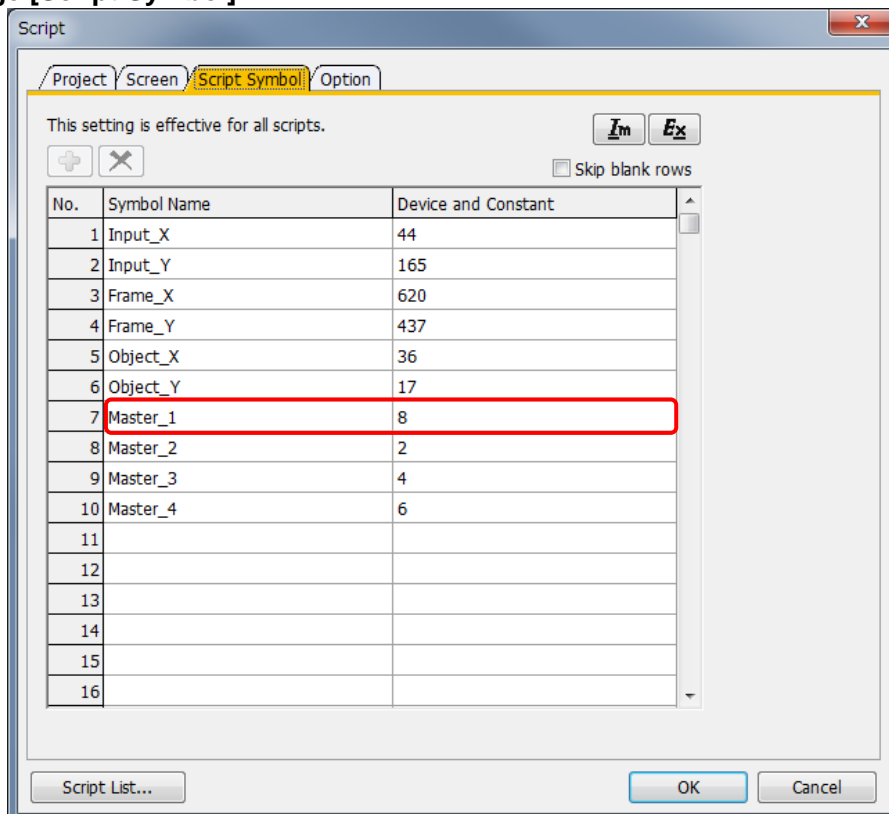
- Changing the start I/O number of the I/O signal

To change the input signal (X device), set [Before] to X0011, [After] to X0091, and [Point] to 2, and click [Replace]. X0011 to X0012 will be changed to X0091 to X0092.

To change the output signal (Y device), set [Before] to Y0000, [After] to Y0080, and [Point] to 18, and click [Replace]. Y0000 to Y0011 will be changed to Y0080 to Y0091.

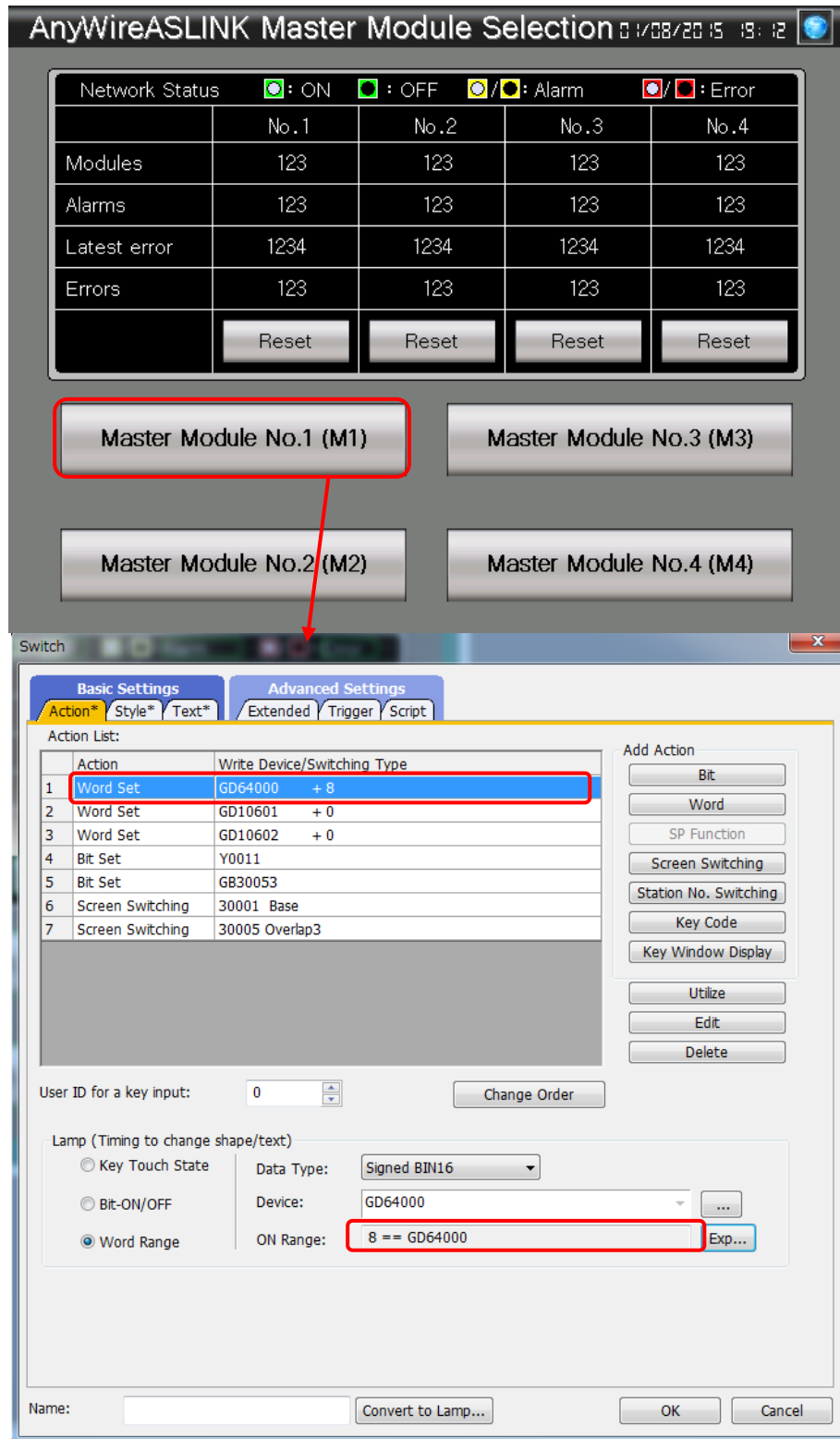


#### 6.4.5 To change [Script Symbol]



Change the [Script Symbol]. Change the [Device and Constant] of [Symbol Name] "Master\_1" from 0 to 8.

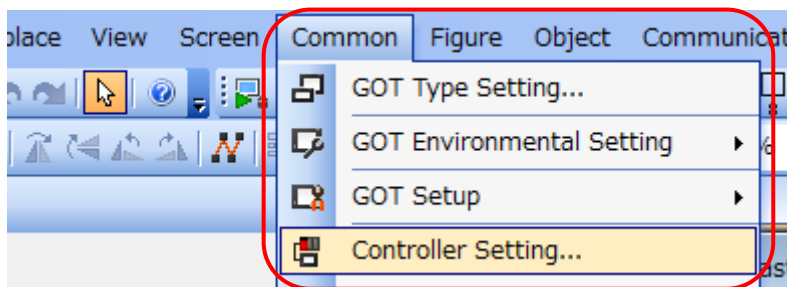
#### 6.4.6 To change switch



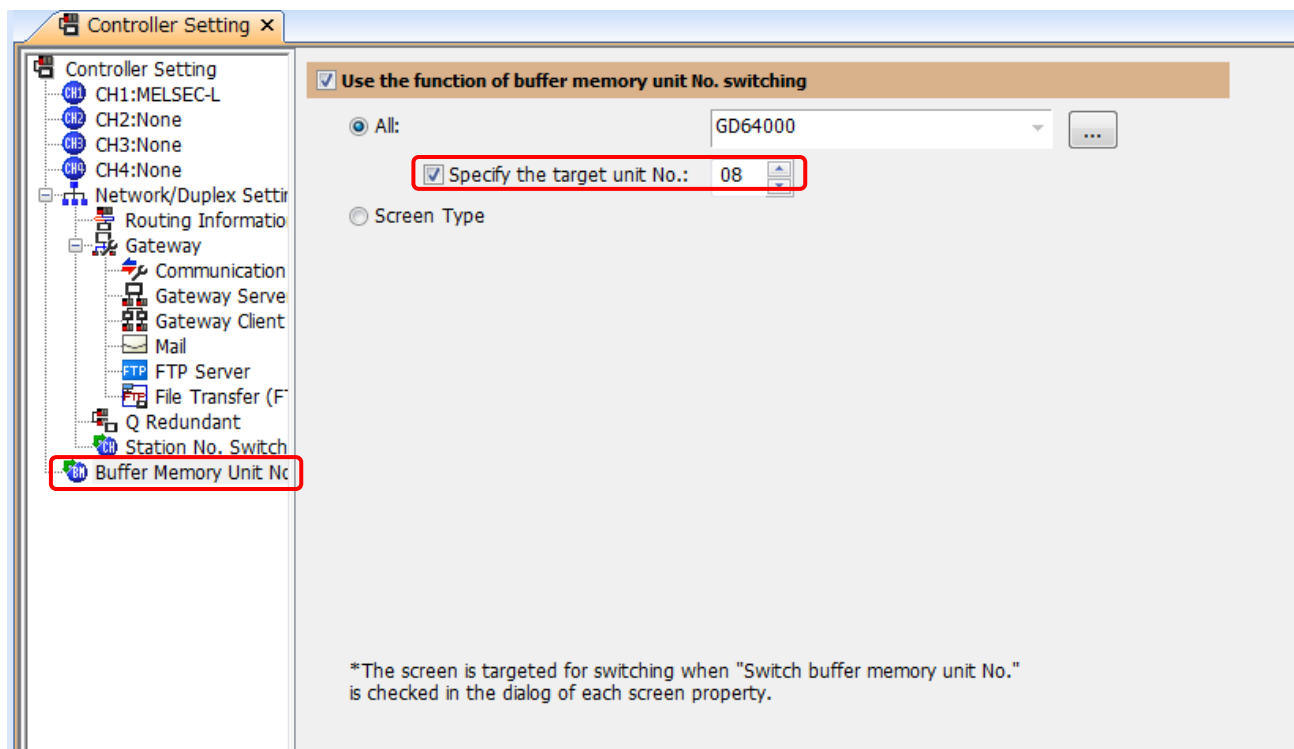
Change the setting of "Master Module No.1 (M1)" on the base screen B-30000. Change the value of GD64000 from 0 to 8. Accordingly, change the [ON Range] of the [Lamp (Timing to change shape/text)] to "8==GD64000".

### 6.4.7 Changing [Buffer Memory Module No. Switching]

- (1) Select [Common]-[Controller Setting] menu.



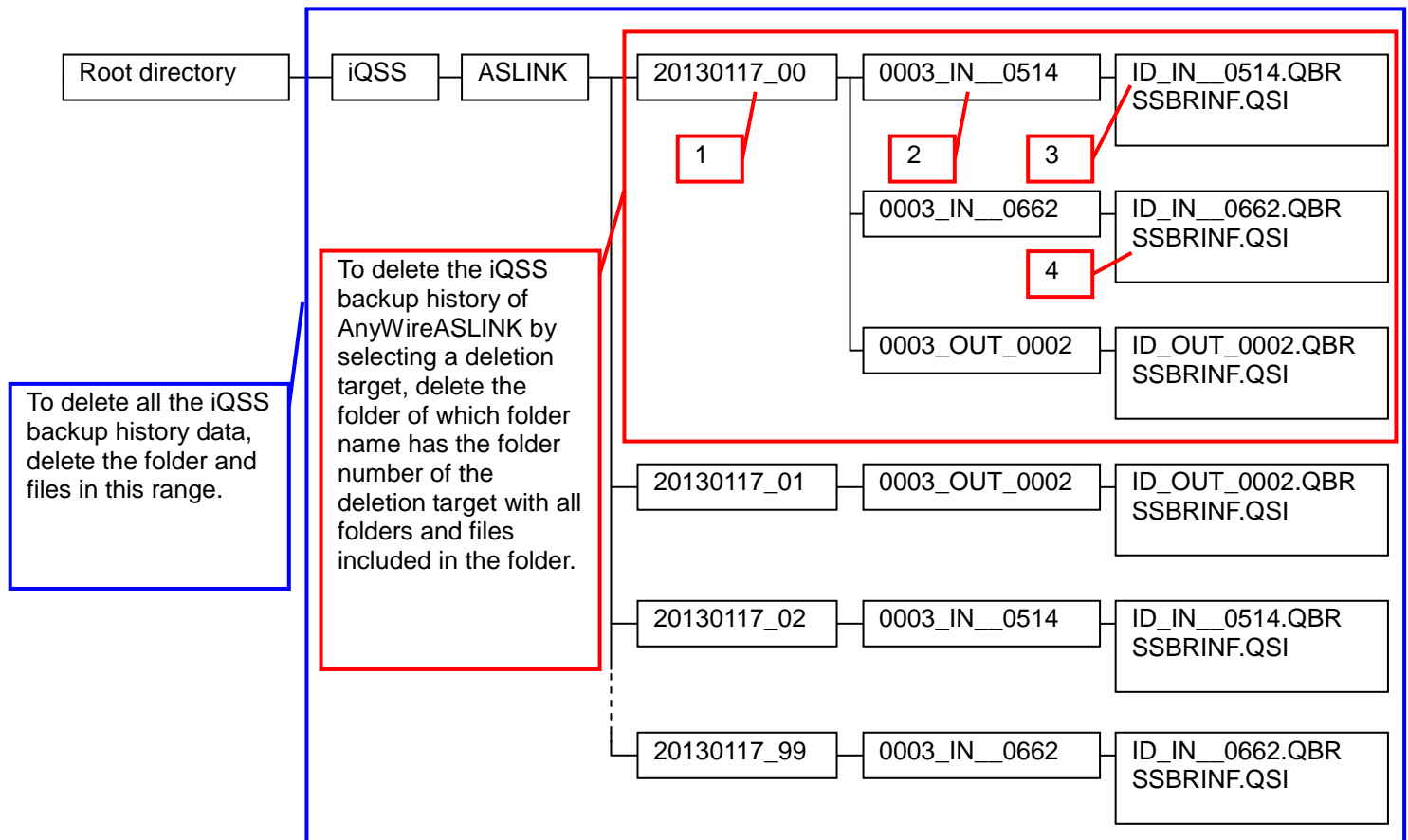
- (2) Select [Buffer Memory Module No. Switching] and change 00 to 08 in [Target Module I/O No. Setting]



## 6.5 iQSS Backup Folder Configuration

### 6.5.1 Folder configuration

The iQSS backup folder configuration of the SD card in the PLC is as follows. For more details about 1 to 4 in the diagram, please refer to "6.5.2 Folder name, file name details".



### 6.5.2 Folder name, file name details

1. 20130117 \_ 00 ... Backup folder name (date, folder number)
  - Number (2 digits [00 to 99] [decimal])
  - Backed up date (YYYYMMDD)
2. 0003 \_ IN \_ 0514 ... Backup folder name (I/O No., ID)
  - ID number (4 digits [decimal]) \*1
  - IN\_: Input/combined slave module
  - OUT: Output slave module
  - I/O No. (4 digits [hexadecimal])
3. ID \_ IN \_ 0514.QBR...Backup file name
  - ID number (4 digits [decimal]) \*1
  - IN\_: Input/combined slave module
  - OUT: Output slave module
4. SSBRINF.QSI ... System file

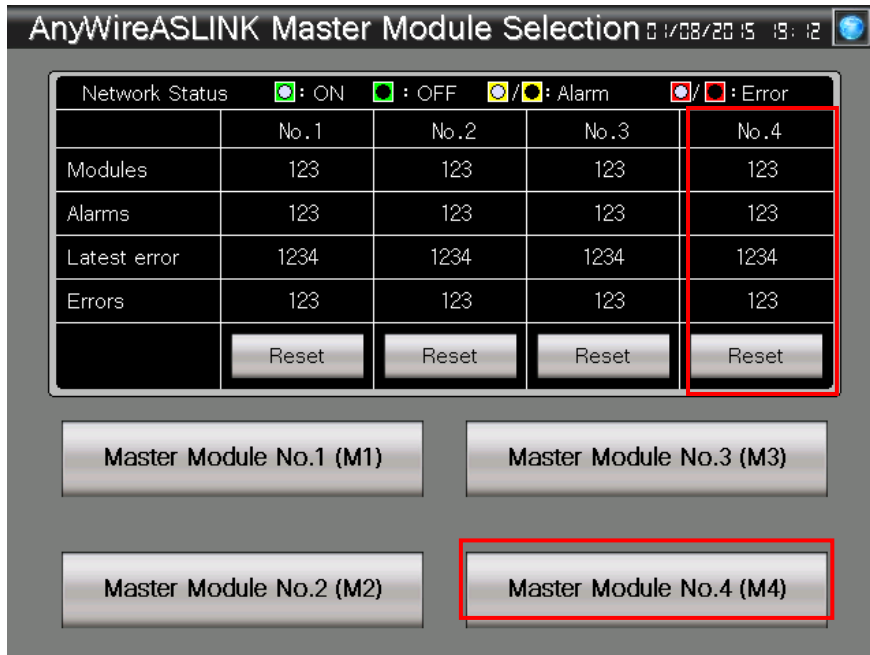
\*1: The ID numbers of the AnyWireASLINK input/combined slave modules are managed by using the number adding 512 to the ID number which is assigned to the module. Therefore, the control number 512 to 766 is added to the end of the backup folder name for input/combined slave modules.

## 6.6 To Change the number of Master Module

To reduce the number of Master Module, follow the procedure below to change the base screen B-30000 and the script.

As an example, the following steps show how to delete the I/O No.4 of Master Module.

### 6.6.1 To change base screen



Change the base screen B-30000. Delete the items of "No.4" in the "Network Status" and the "Master Module No.4 (M4)" switch.

### 6.6.2 To change script

Delete the script No.30079, No.30080 of the base screen B-30001, and also delete the script No.30081 of the base screen B-30006 to B-30051.