AnyWireASLINK Network Monitor Function

Sample Screen 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

Sample Screen Manual

Date	Control No.*	Description
2013/10	BCN-P5999-0119	First edition

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

Project Data

Date	Project Data	GT Designer3*	Description
2013/10	AnyWireASLINK_V_Ver1_E.GTX	1.100E	First edition

* 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. 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 (LJ51AW12AL).

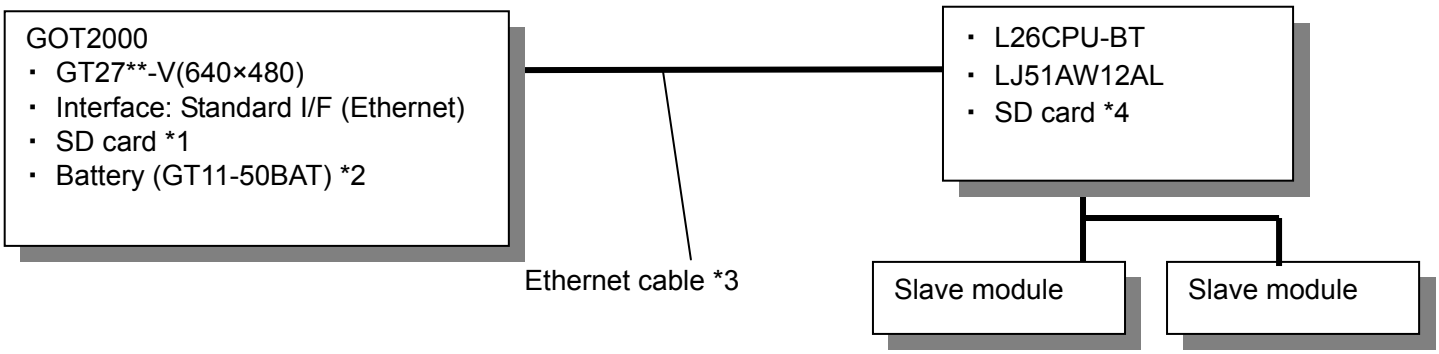
The sample screens were created to connect to a MELSEC-L Series CPU. To connect to a MELSEC-Q Series CPU, it is necessary to modify some of the data. For how to modify the data, please refer to “7.3 Changing CPU”. Note that the MELSEC-Q Series CPUs do not support the backup/restoration of slave module 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	Gothic	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

4. AnyWireASLINK MASTER MODULE

4.1 Start I/O Number of Module

The module's start I/O number is set to 30H. For more details about changing the start I/O number, please refer to "7.4 Changing Start I/O Number".

5. SCREEN SPECIFICATIONS

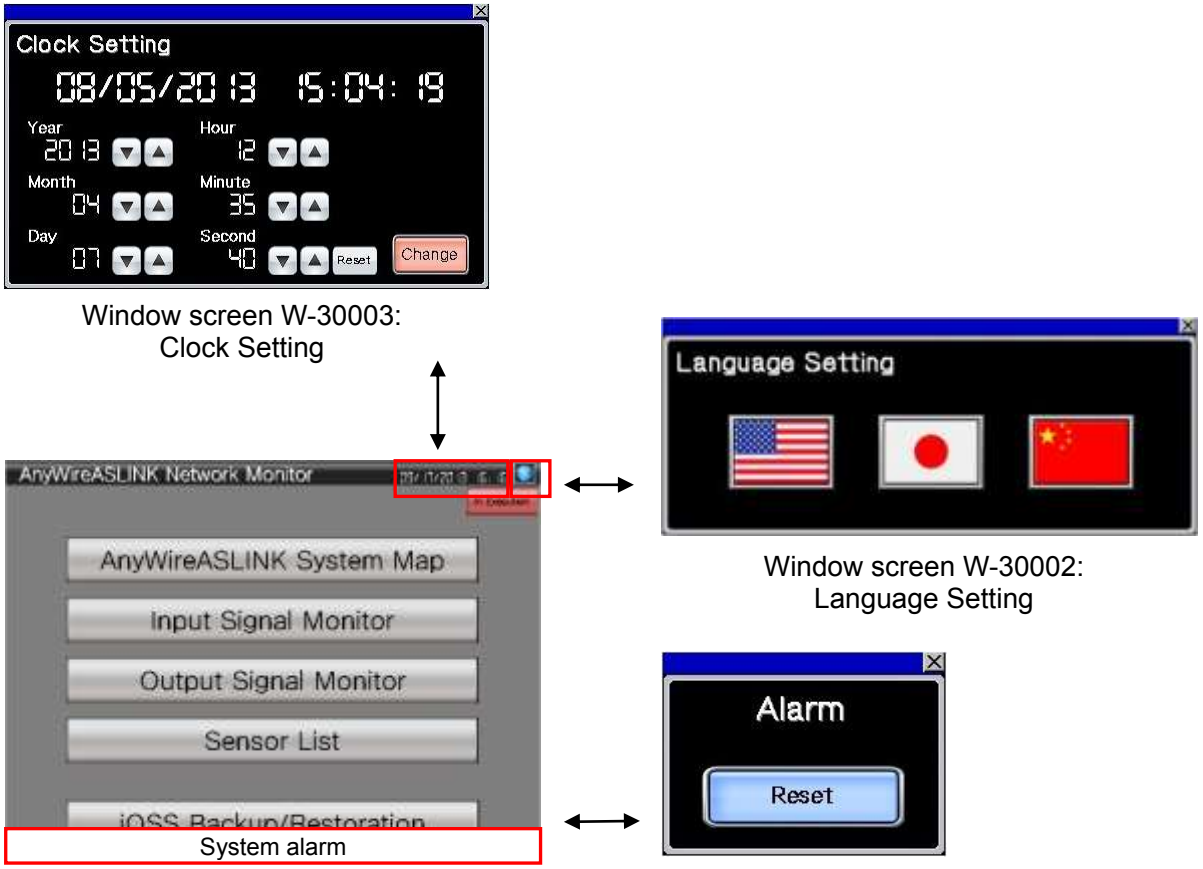
5.1 Display Language

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. 250 and No. 255 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 Transition

5.2.1 Screen transition (common)



Window screen W-30003:
Clock Setting

Window screen W-30002:
Language Setting

Base screen
(B-30001 Menu and other base screens)

Window screen W-30001: Alarm Reset

5.2.2 Screen transition (individual)

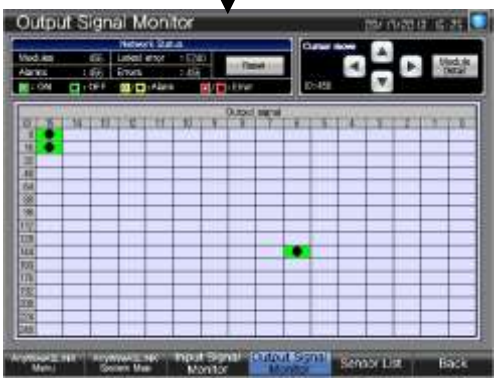
Base screen B-30001: AnyWireASLINK Menu

Base screen B-30002: AnyWireASLINK System Map

Base screen B-30003: Input Signal Monitor

To next page

From previous page



Base screen B-30004: Output Signal Monitor

Base screen B-30005: Sensor List

Base screen B-30006 to B-30019: Detail (The screen varies depending on the slave module type)

Window screen W-30004: Parameter Access Error

To next page

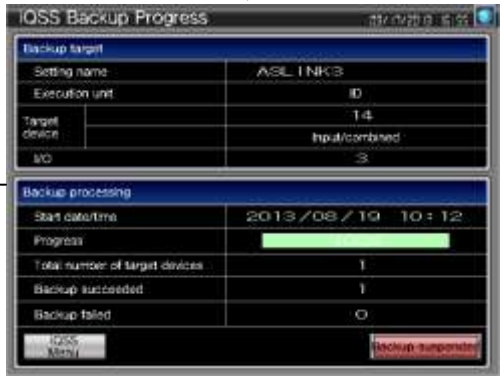
From previous page

Base screen B-30100: iQSS Menu

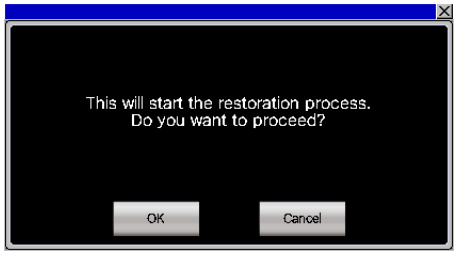
Base screen B-30101: iQSS Backup

Base screen B-30102: iQSS Backup Setting

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



Base screen B-30103: iQSS Backup Progress

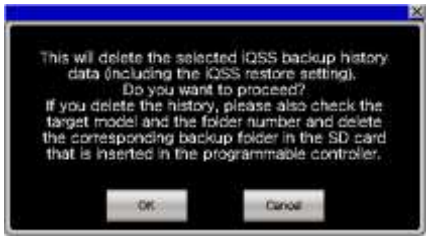


Window screen W-30102: Execution Check Dialog (B-30101, B-30103, B-30104, B-30106 common)

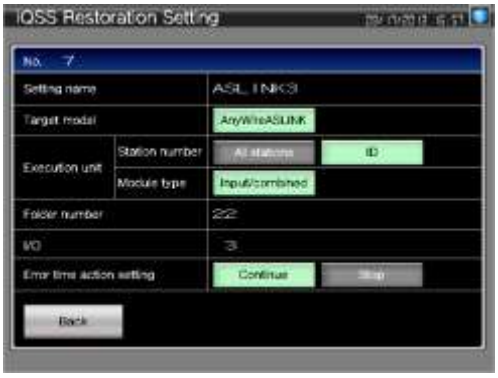
To next page

From previous page

Base screen B-30104: iQSS Restoration



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

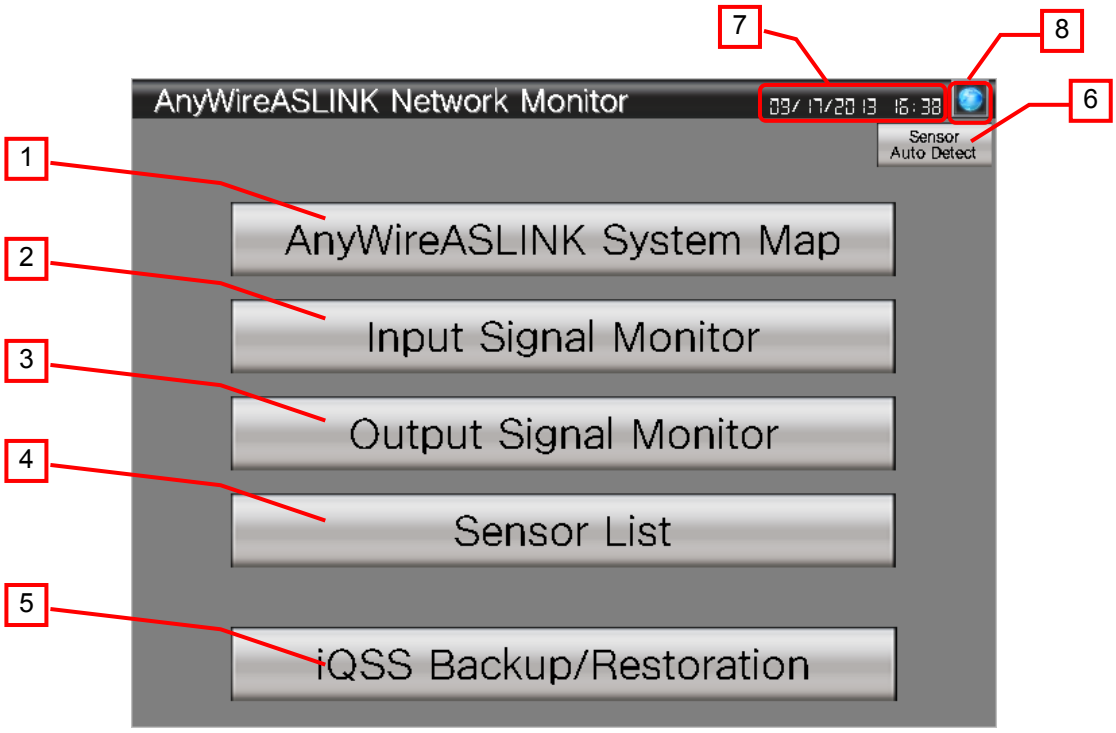


Base screen B-30105:
iQSS Restoration Setting

Base screen B-30106:
iQSS Restoration Progress

5.3 Explanation of Screens

5.3.1 AnyWireASLINK Menu (B-30001)



Outline

This is the AnyWireASLINK Network Monitor menu screen.

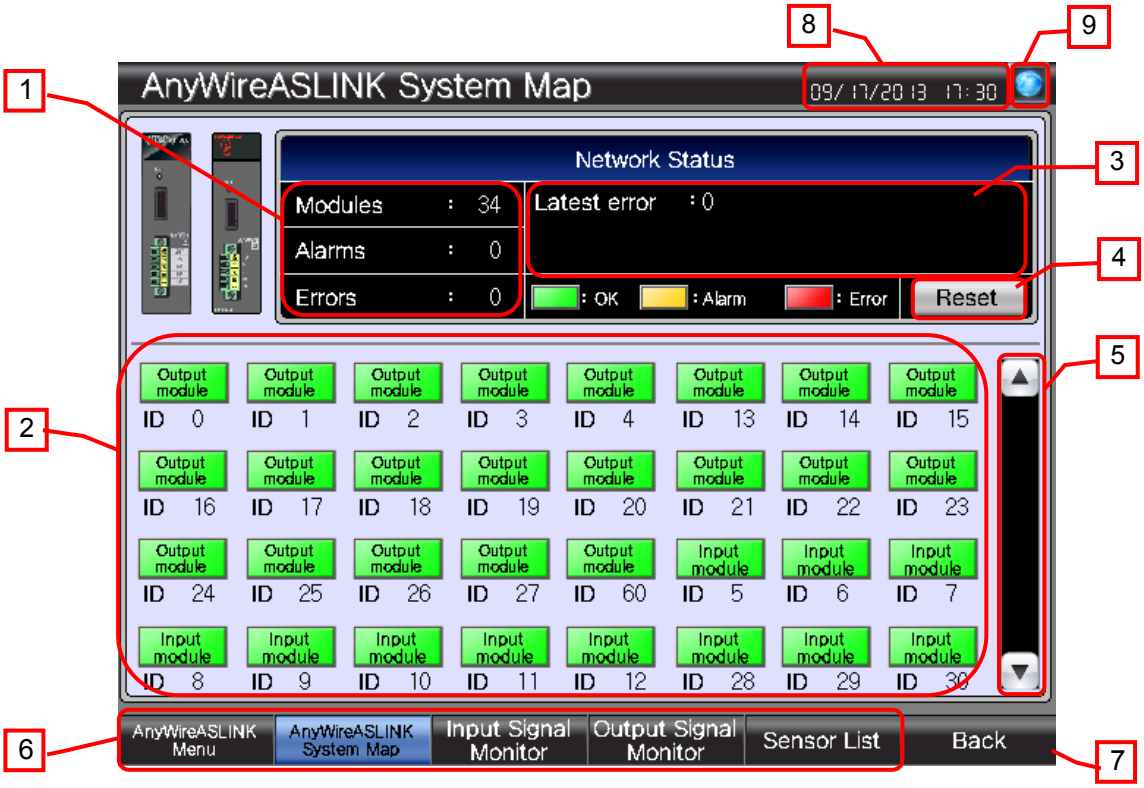
Description

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. Displays the current date and time. Touch the area to open the [Clock Setting] window.
8. 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.
- 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.2 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.

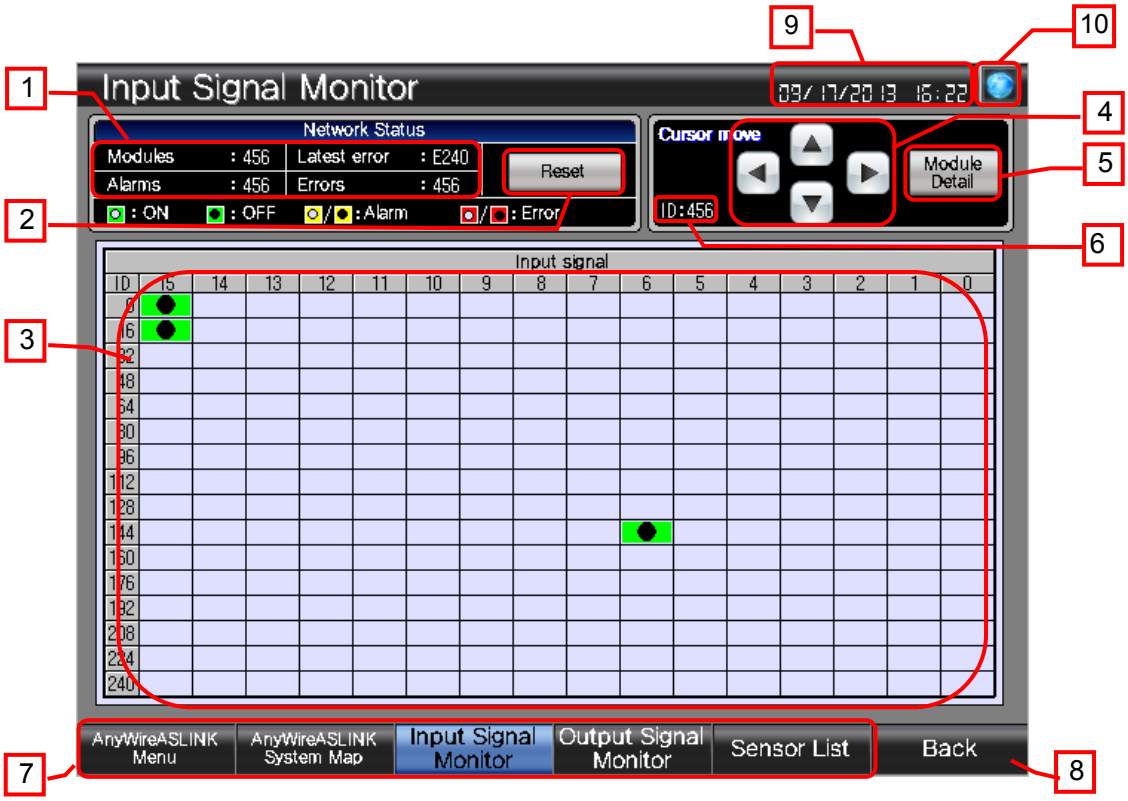
Description

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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.3 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.

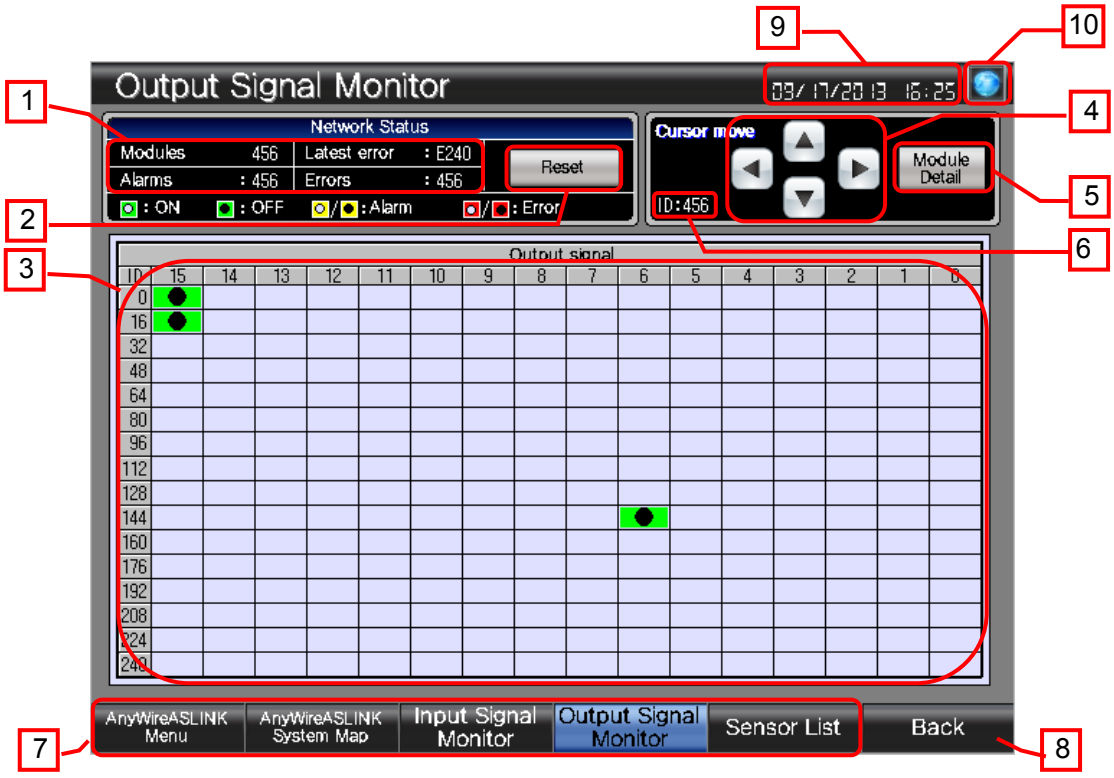
Description

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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.4 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.

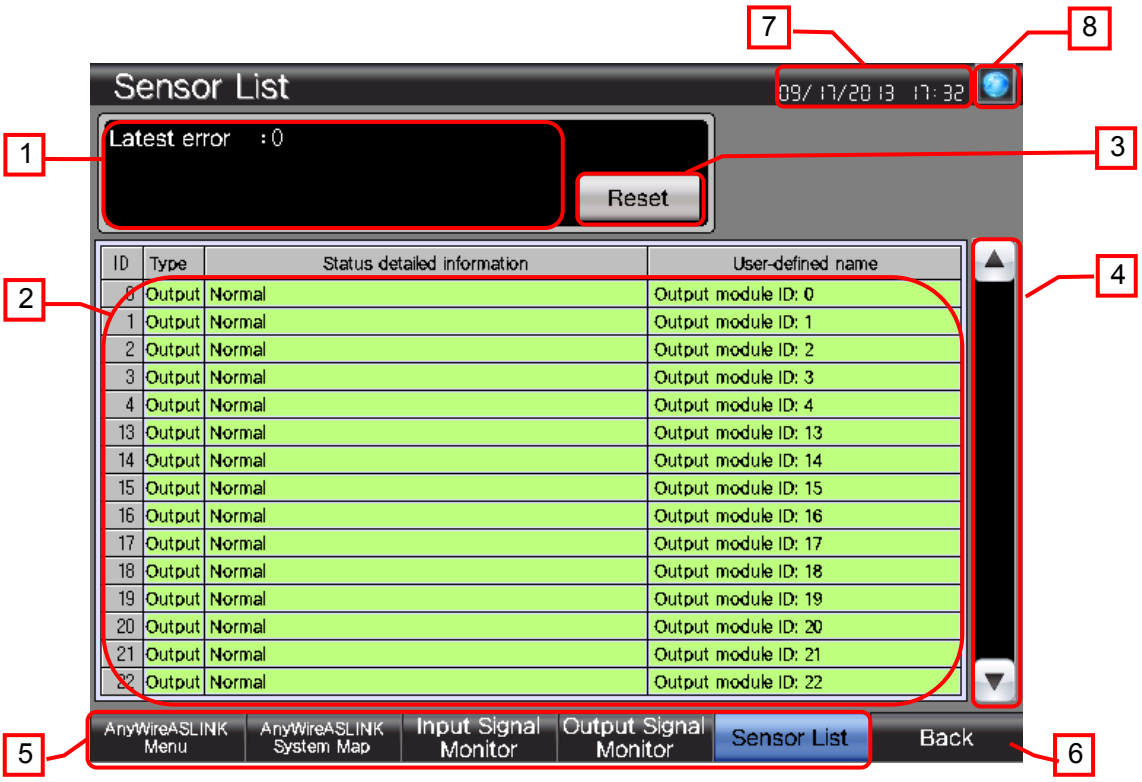
Description

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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.5 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.

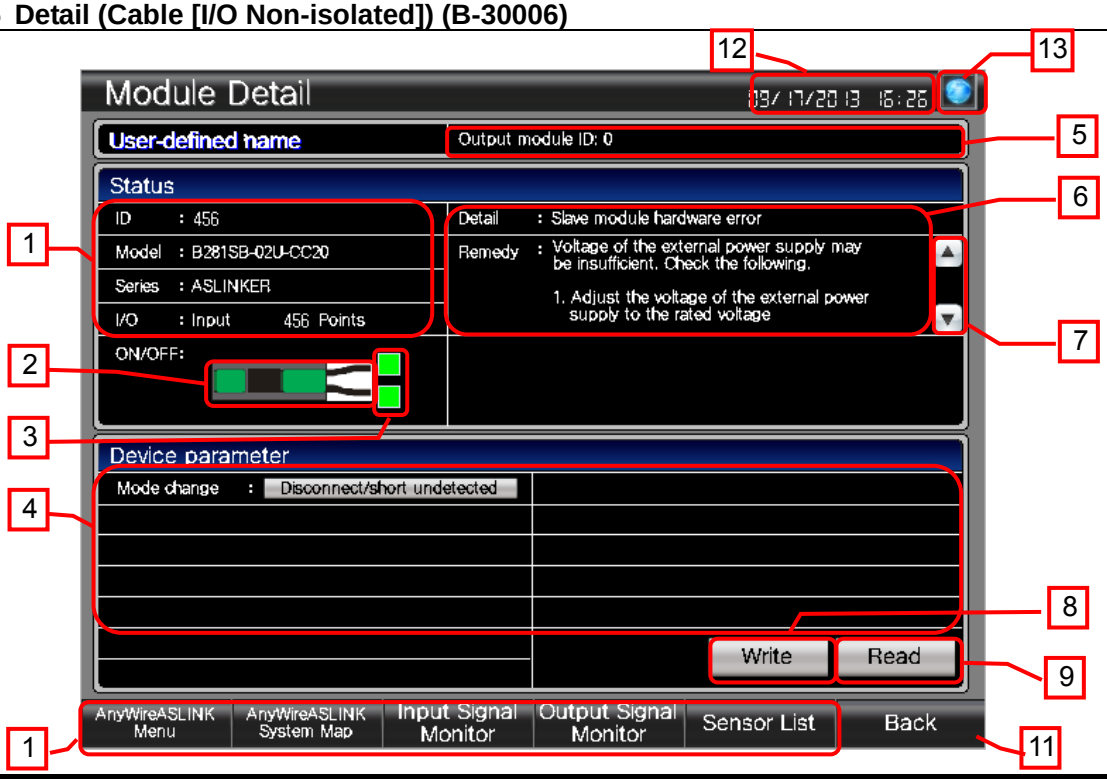
Description

1. Displays the occurring alarm, or the error code and the descriptions that correspond to the error code. If multiple 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 "7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.6 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.

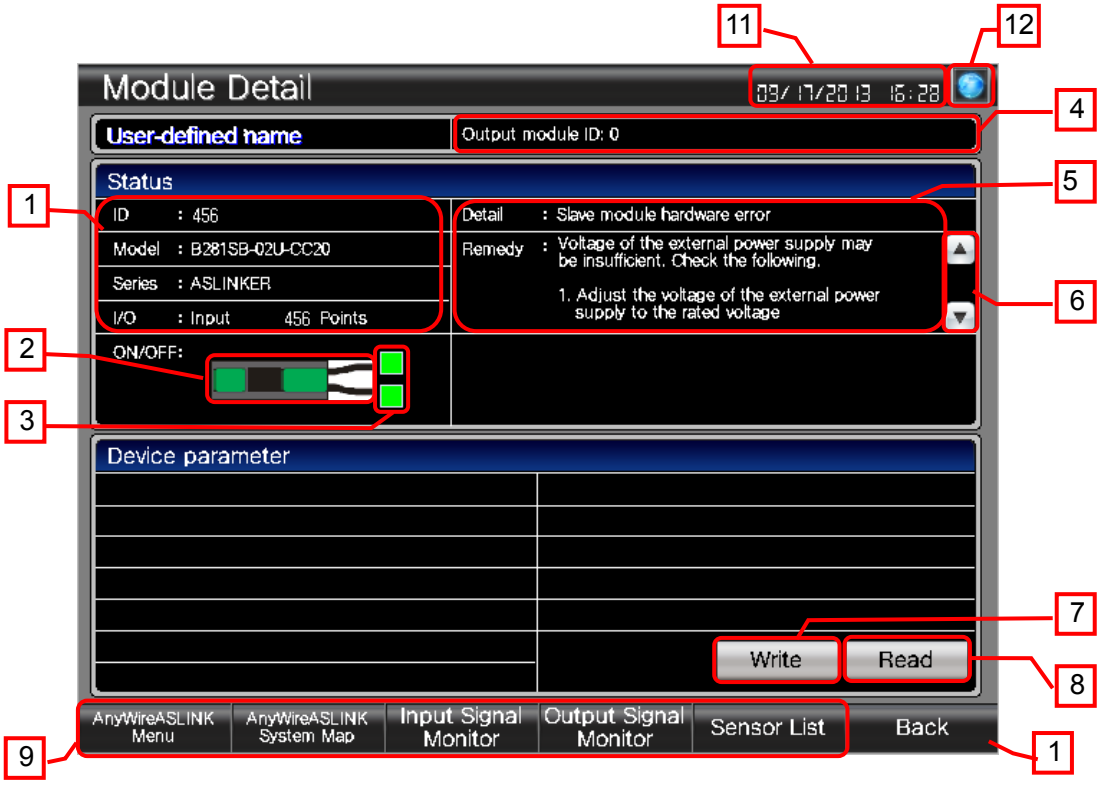
Description

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 "7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

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



Outline

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

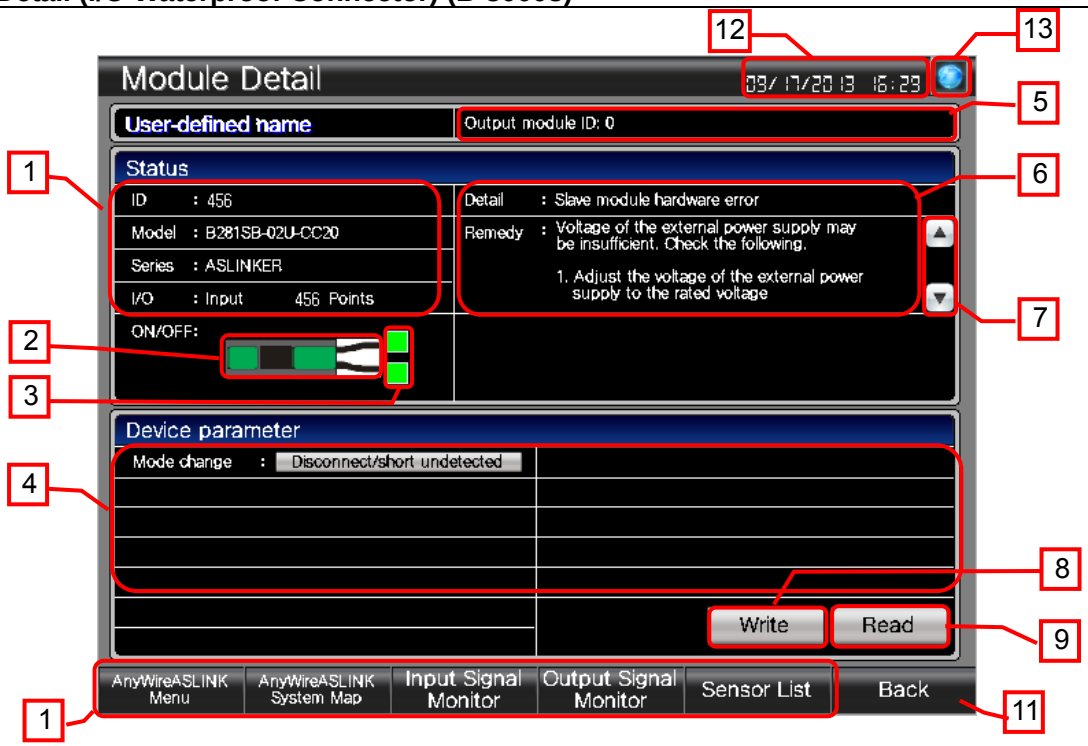
Description

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 "7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.8 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.

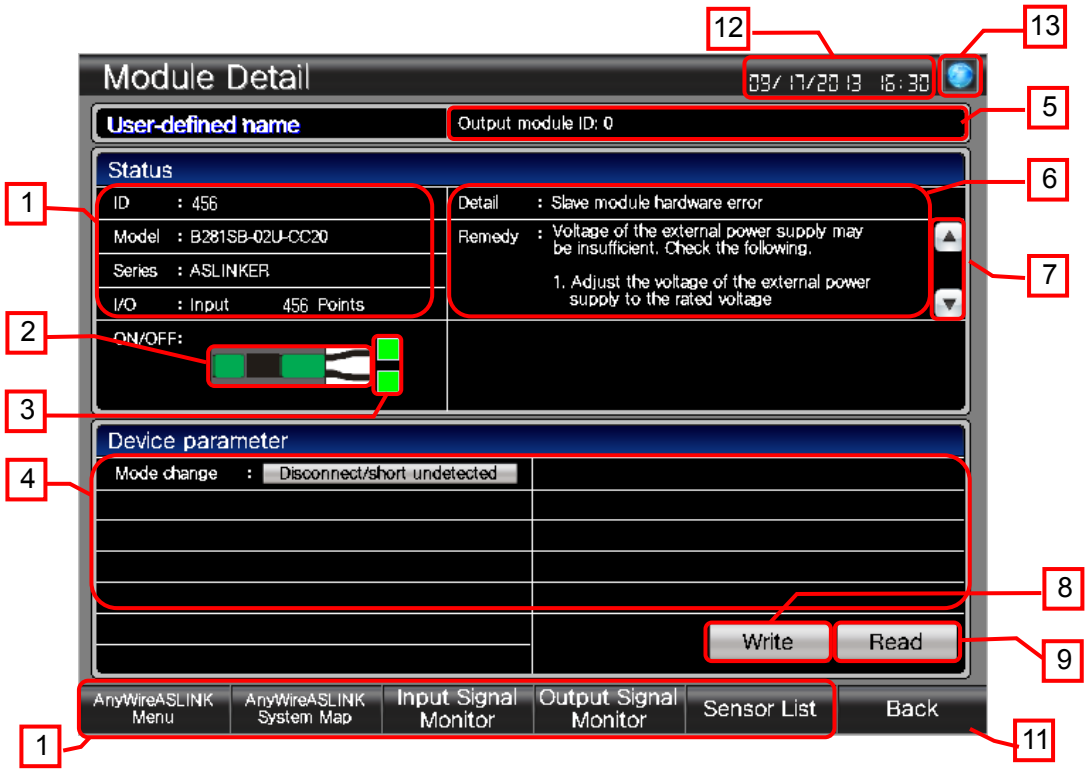
Description

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.9 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.

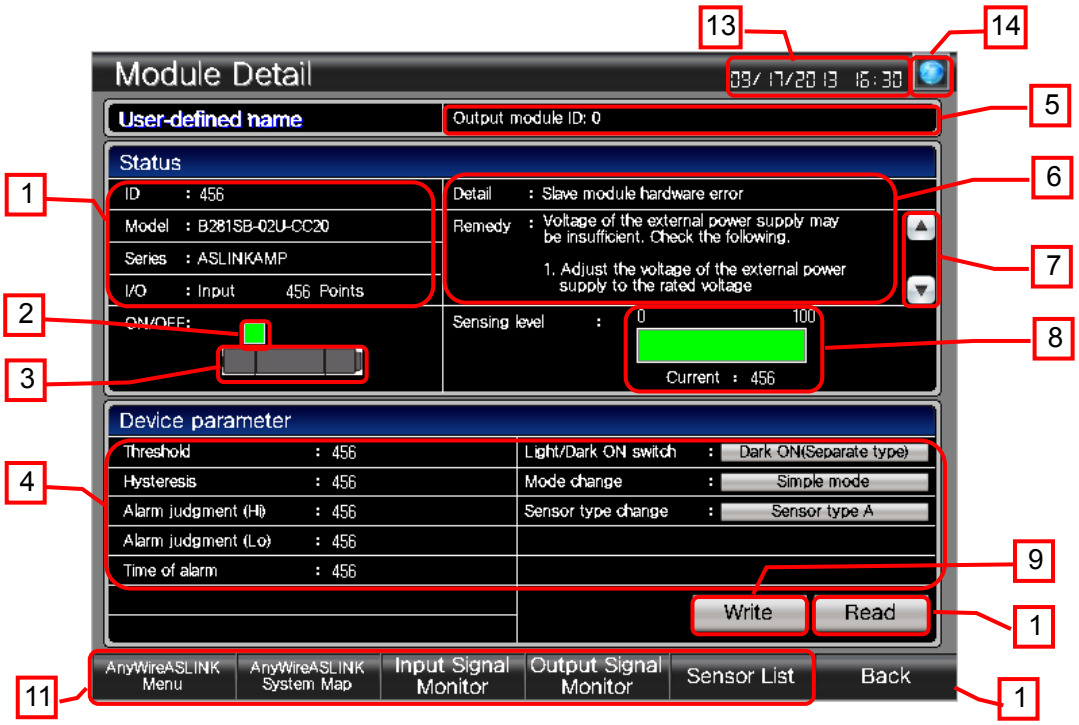
Description

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

- 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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.10 Detail (Photoelectronic [CAM, CAS]) (B-30010)



Outline

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

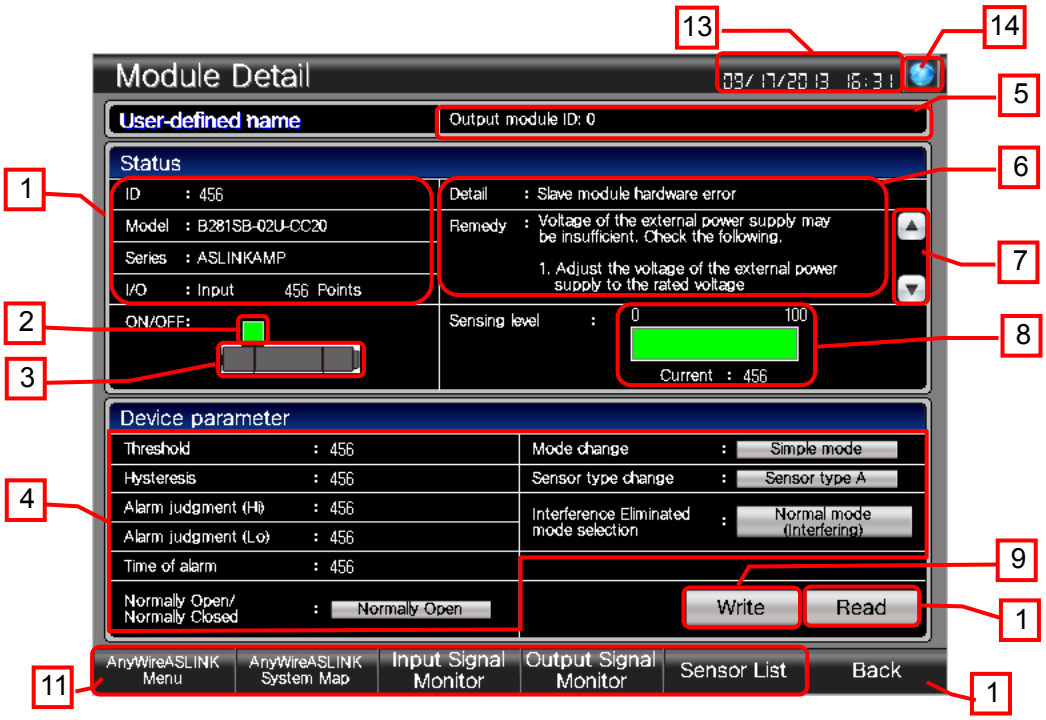
Description

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.11 Detail (Proximity Amp) (B-30011)



Outline

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

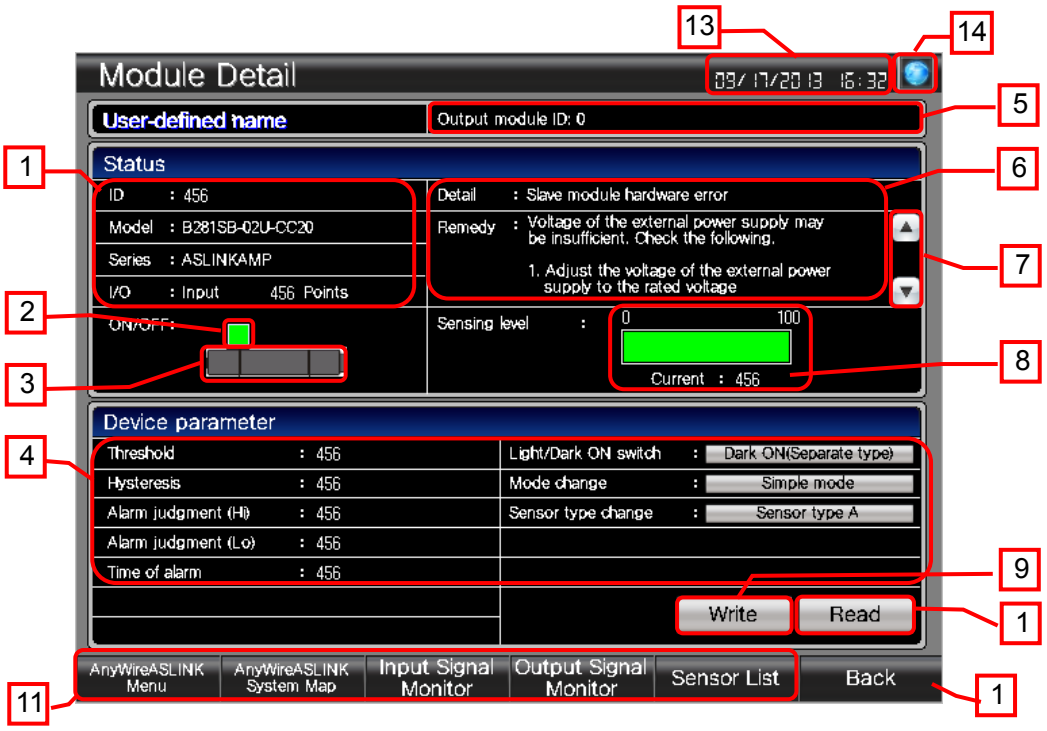
Description

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.

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.12 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.

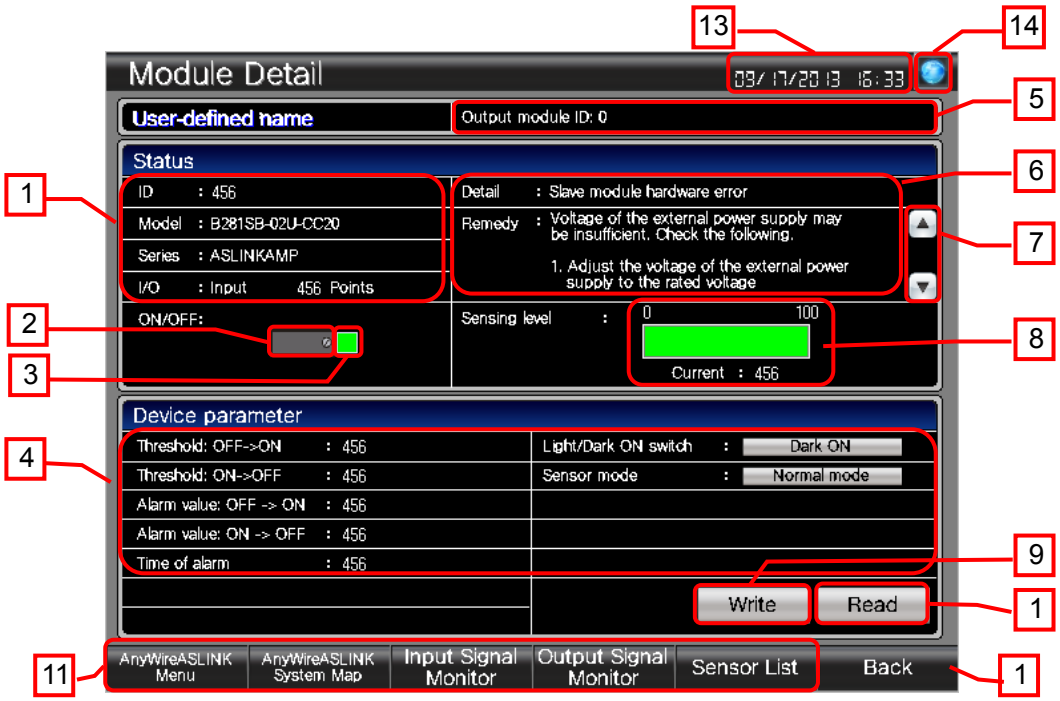
Description

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.

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.13 Detail (Cylinder Switch) (B-30013)



Outline

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

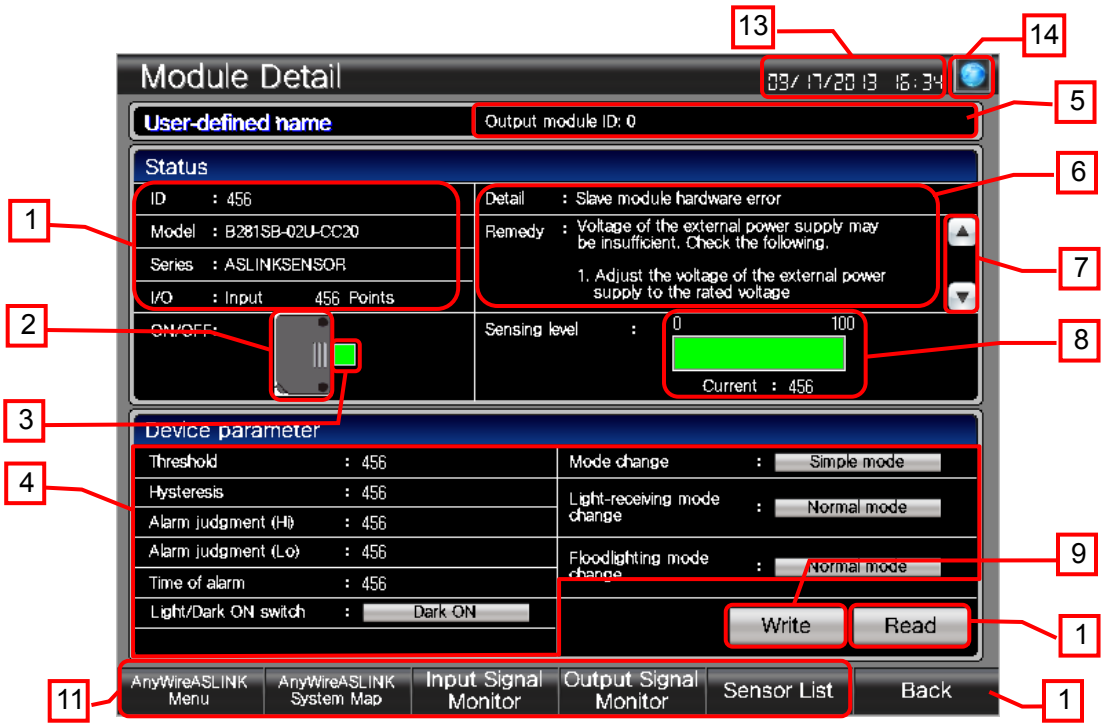
Description

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 numerical displays of the level for “Sensing level”, “Threshold: OFF->ON”, and “Threshold: ON->OFF”. 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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.14 Detail (Photoelectronic Sensor) (B-30014)



Outline

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

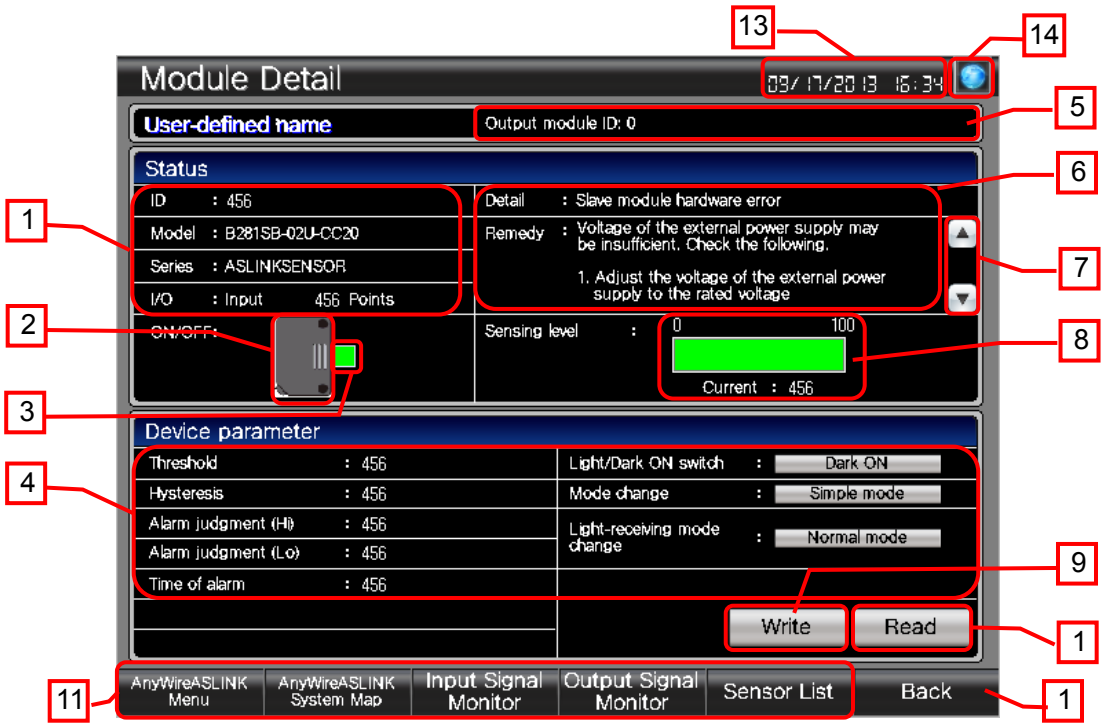
Description

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.

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.15 Detail (Photoelectronic [Rcvr.]) (B-30015)



Outline

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

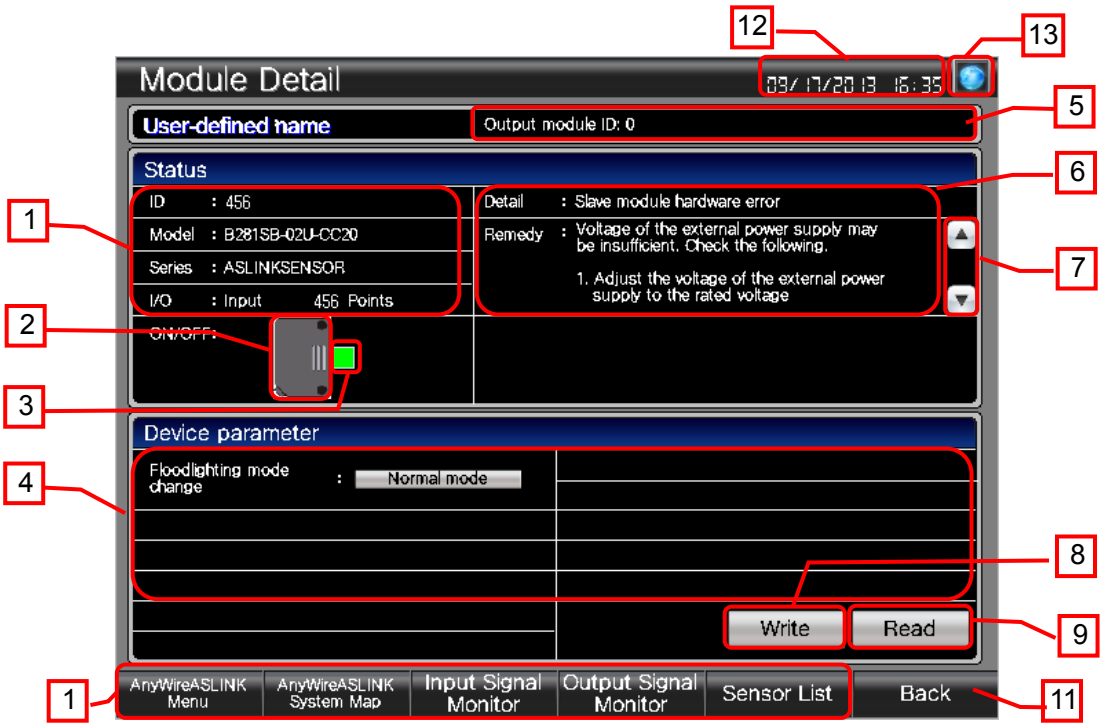
Description

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.

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.16 Detail (Photoelectronic [LT Src.]) (B-30016)



Outline

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

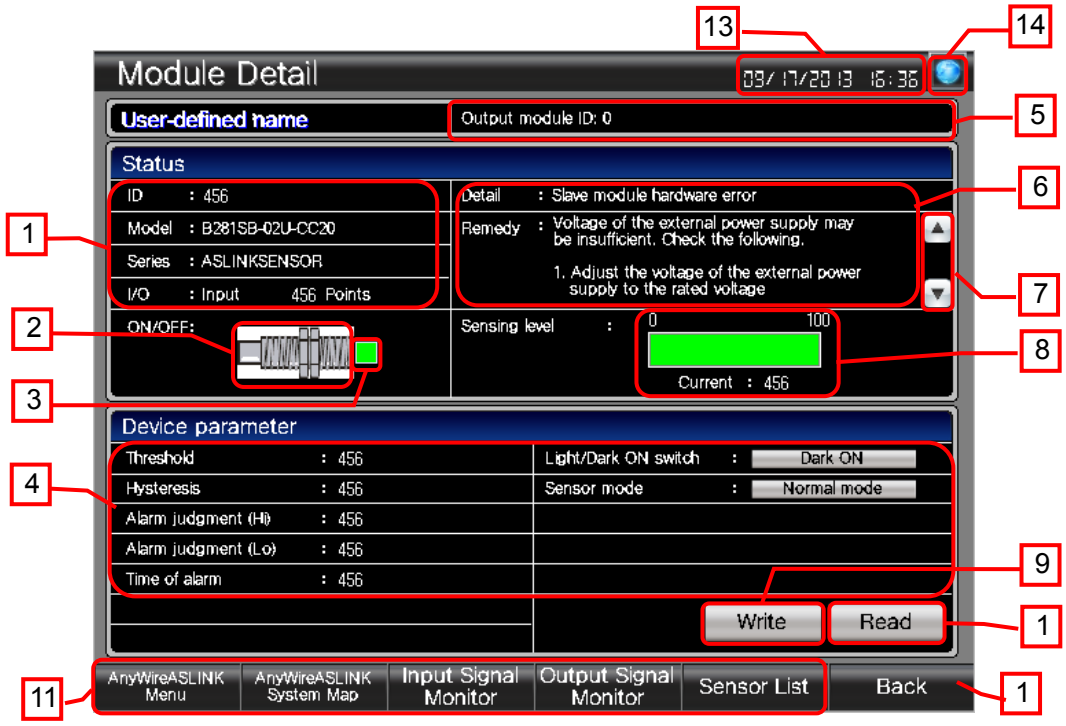
Description

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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.17 Detail (Proximity Switch) (B-30017)



Outline

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

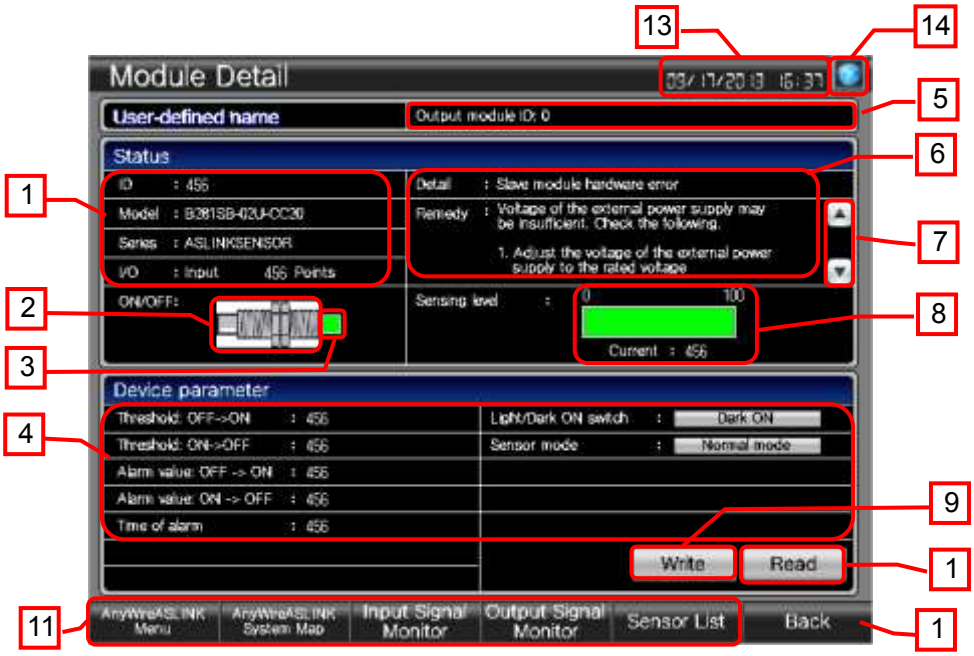
Description

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 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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.18 Detail (Pressure Sensor) (B-30018)



Outline

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

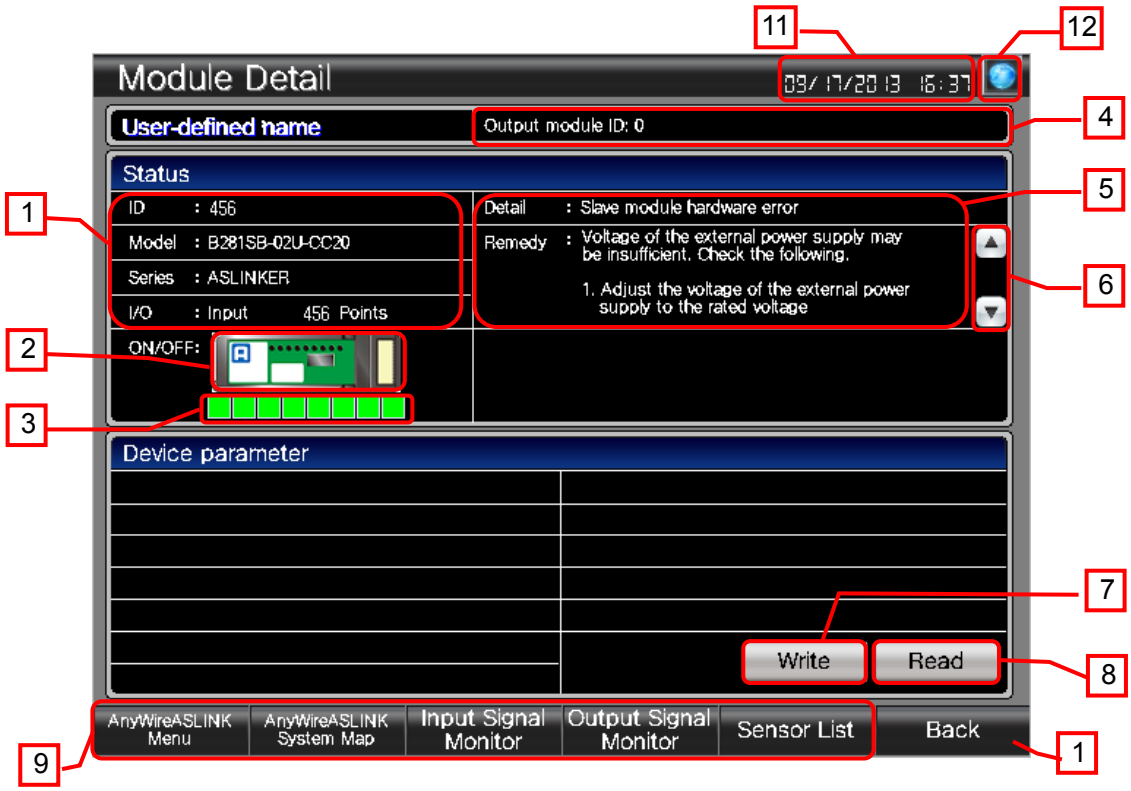
Description

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.

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: OFF->ON”, and “Threshold: ON->OFF”. 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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.19 Detail (ASLINKTERMINAL [Driver]) (B-30019)



Outline

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

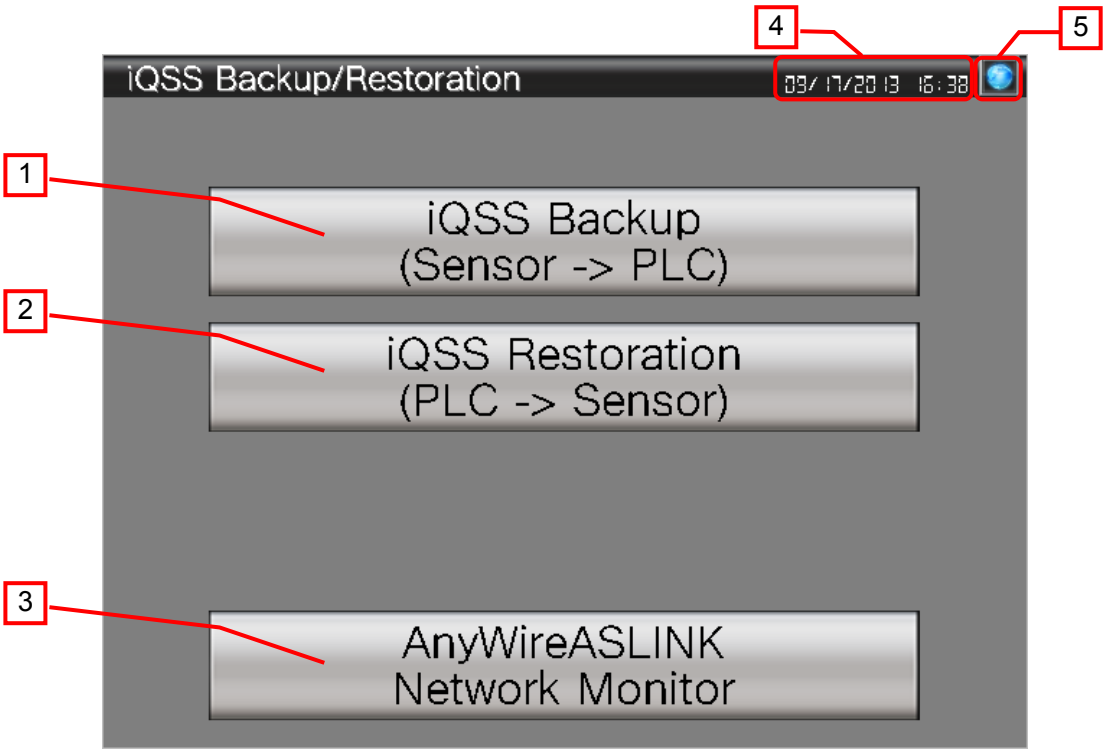
Description

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. 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 “7.1 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.20 iQSS Menu (B-30100)



Outline

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

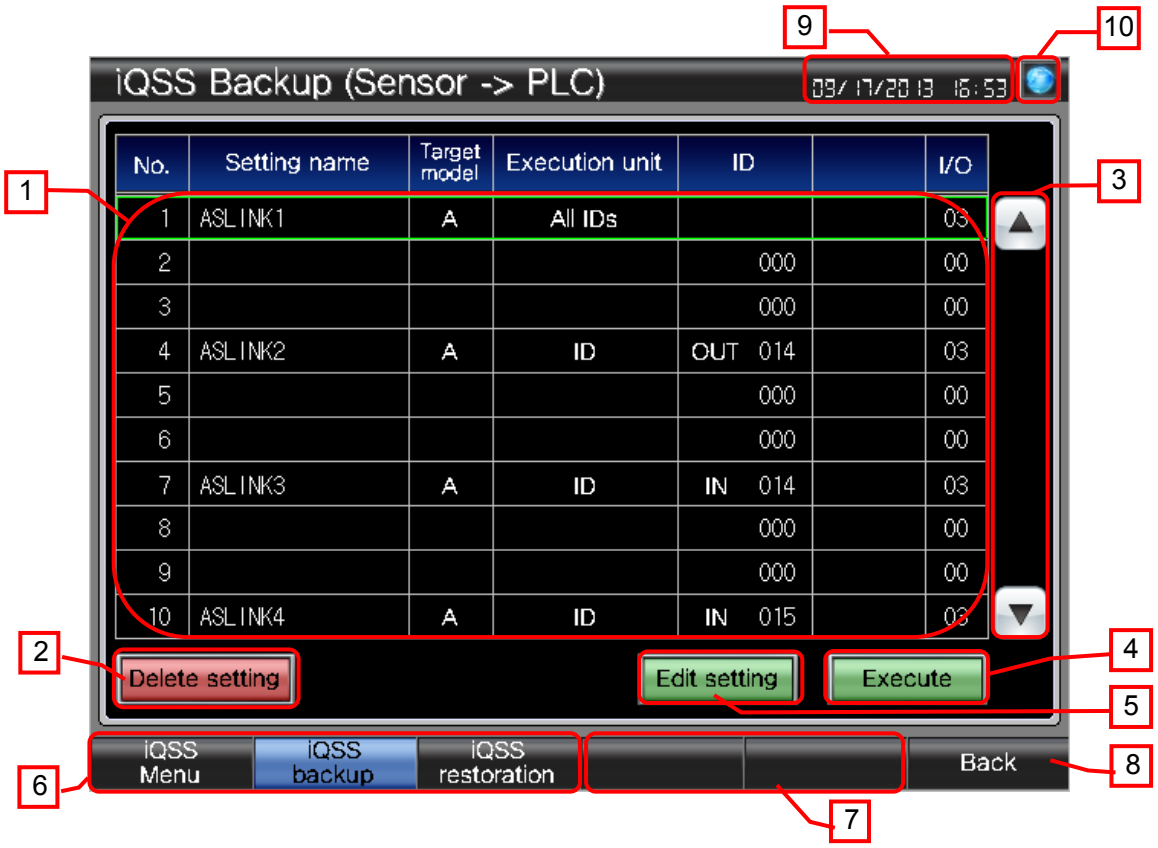
Description

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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.21 iQSSBackup (B-30101)



Outline

This screen is used to exetute 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.

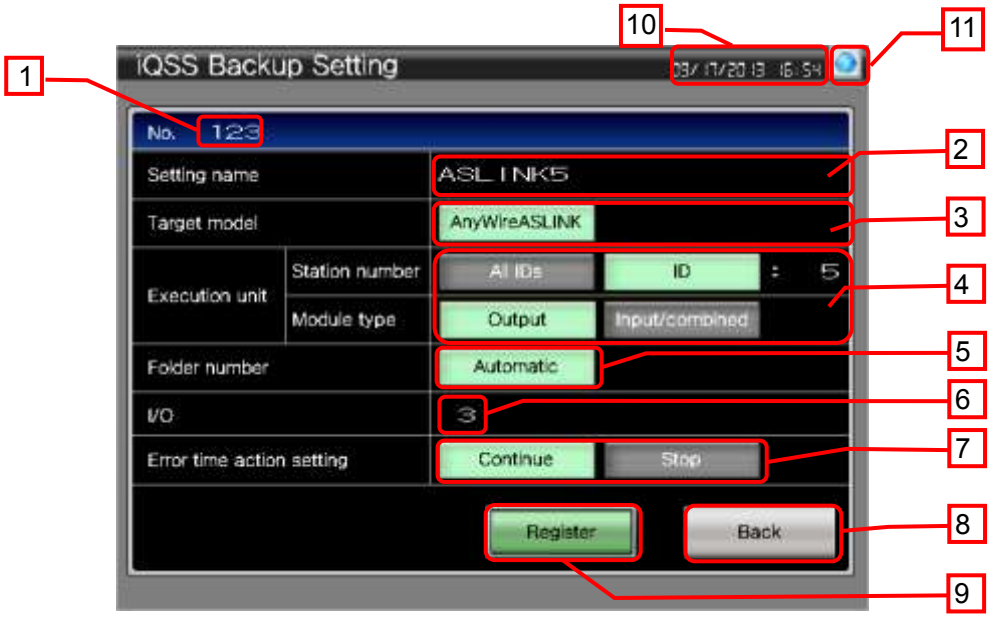
Description

1. Displays the iQSS backup setting 10 cases at a time, up to 100 cases in a list format. By touching a 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.
4. Switches to the [iQSS Backup Progress] screen and exetutes 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 “Execusion 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.22 iQSS Backup Setting (B-30102)



Outline

This screen registers the iQSS backup setting.

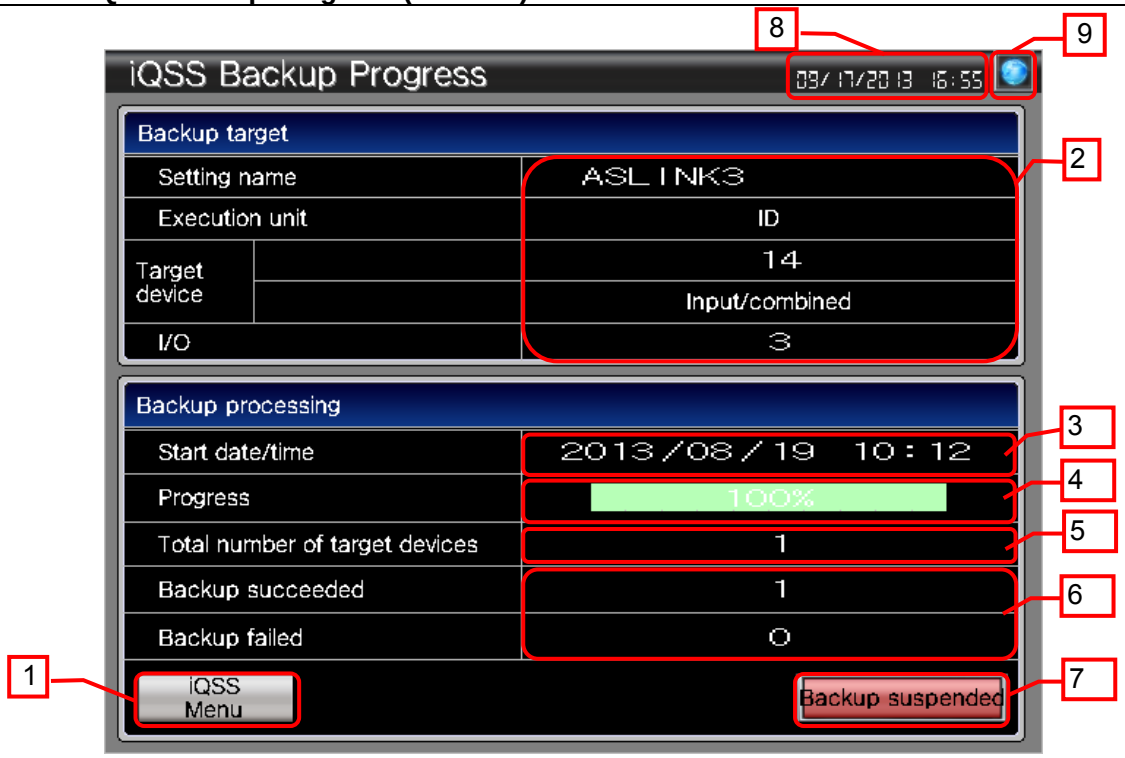
Description

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. Secifies 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.23 iQSS Backup Progress (B-30103)



Outline

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

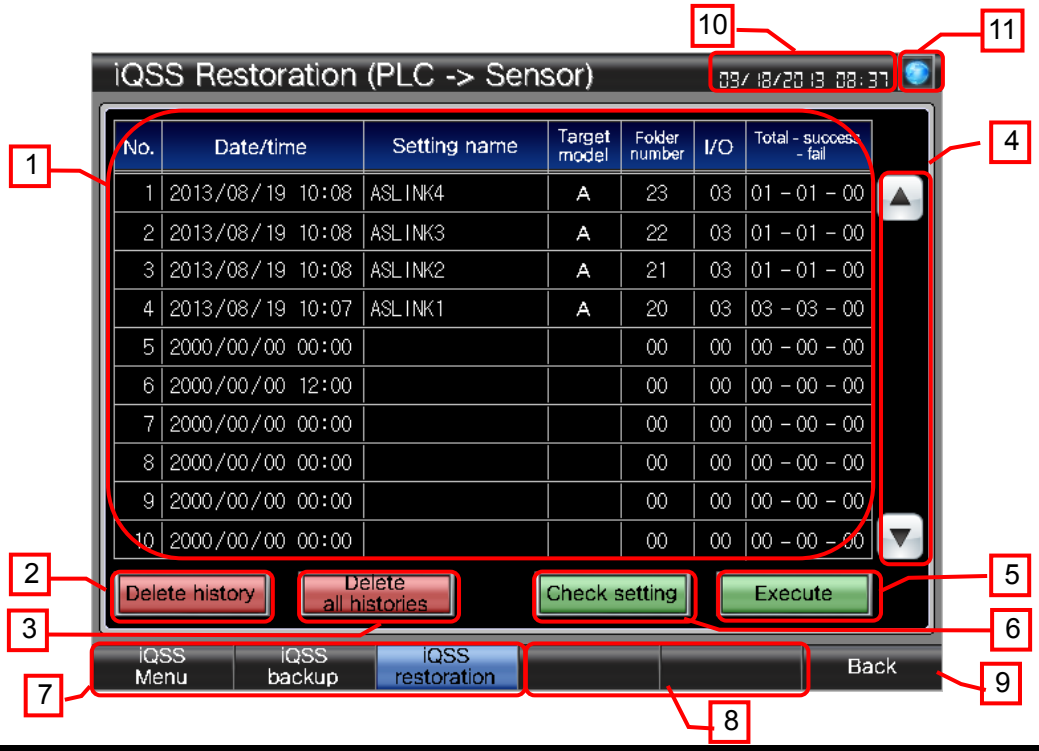
Description

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".
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.24 iQSS Restoration (B-30104)



Outline

This screen is used to exetute 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.

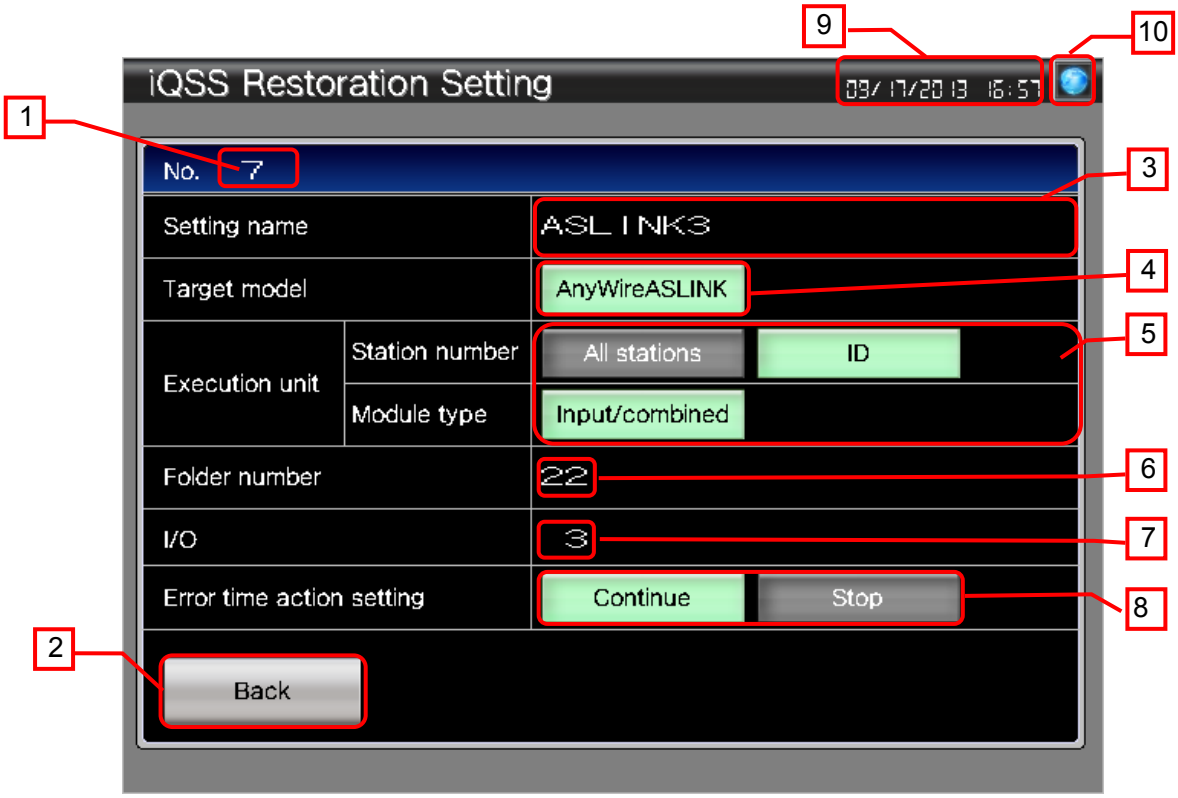
Description

1. Displays the iQSS backup history 10 cases at a time, up to 100 cases 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.
5. Switches to the [iQSS Restoration Progress] screen and exetutes 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 abnormaly 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 “7.5 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.25 iQSS Restoration Setting (B-30105)



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.

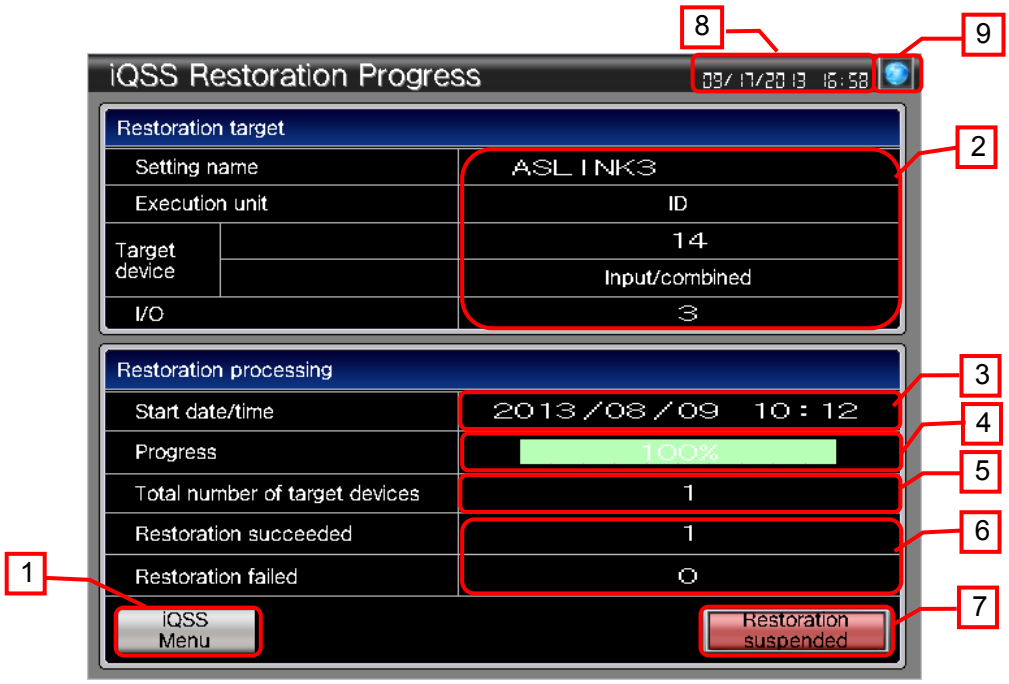
Description

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 settings are 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”.
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.26 iQSS Restoration Progress (B-30106)



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".
- The currently open window closes when the screen is switched.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. Touch the alarm message to open the [Alarm Reset] window.

5.3.27 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

5.3.28 Language Setting (W-30002)

Outline

This window screen allows selecting the GOT language.

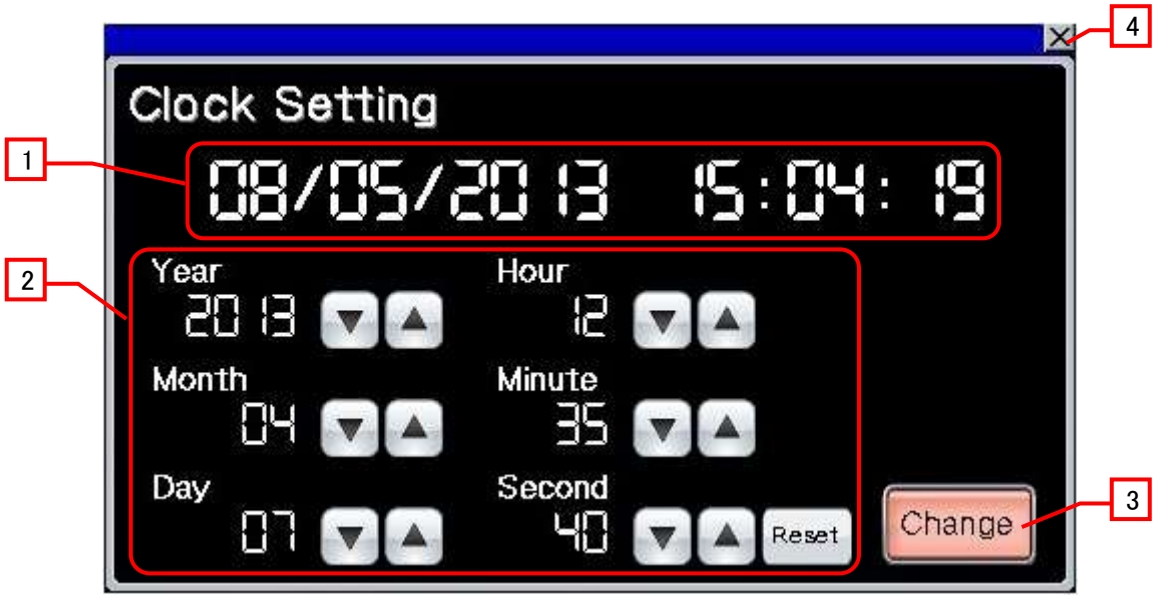
Description

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.

5.3.29 Clock Setting (W-30003)



Outline

This window screen allows changing the GOT clock data.

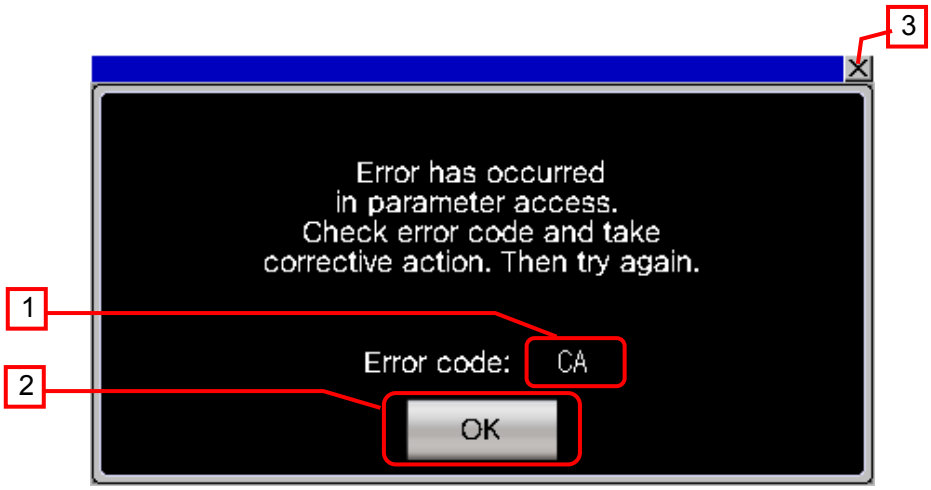
Description

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".

5.3.30 Parameter Access Error (W-30004)



Outline

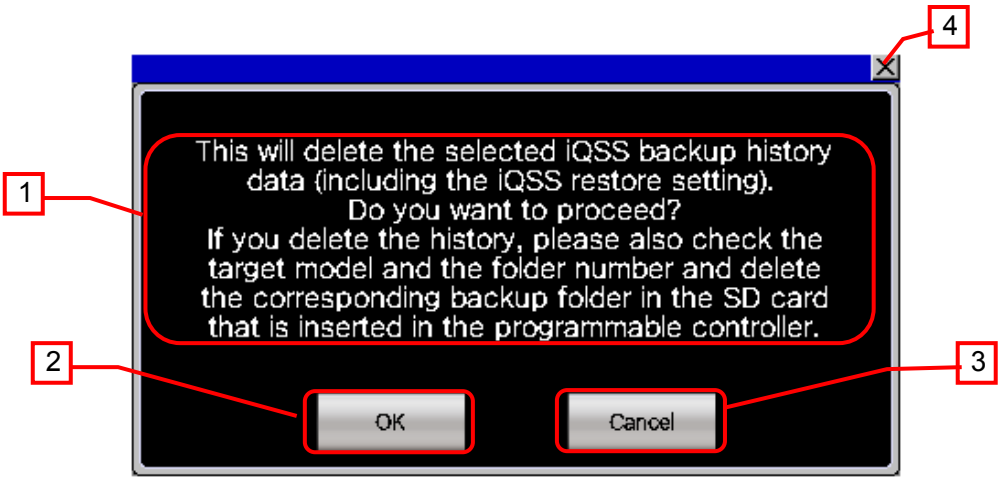
This window screen is displayed if an error occurred when accessing parameters.

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.
3. Closes the window screen.

Remarks

5.3.31 Data Deletion Check Dialog (W-30100)



Outline

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

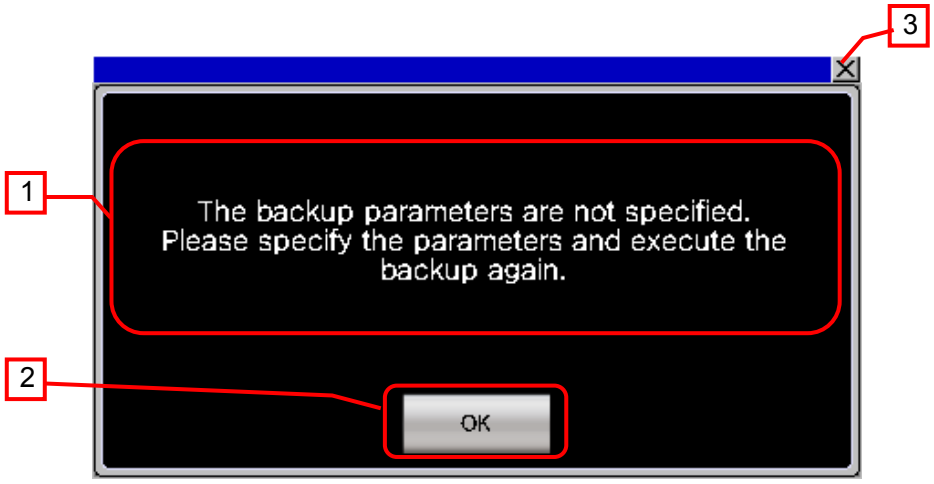
Description

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

Remarks

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

5.3.32 Notification Dialog (W-30101)



Outline

This window screen notifies the processing results.

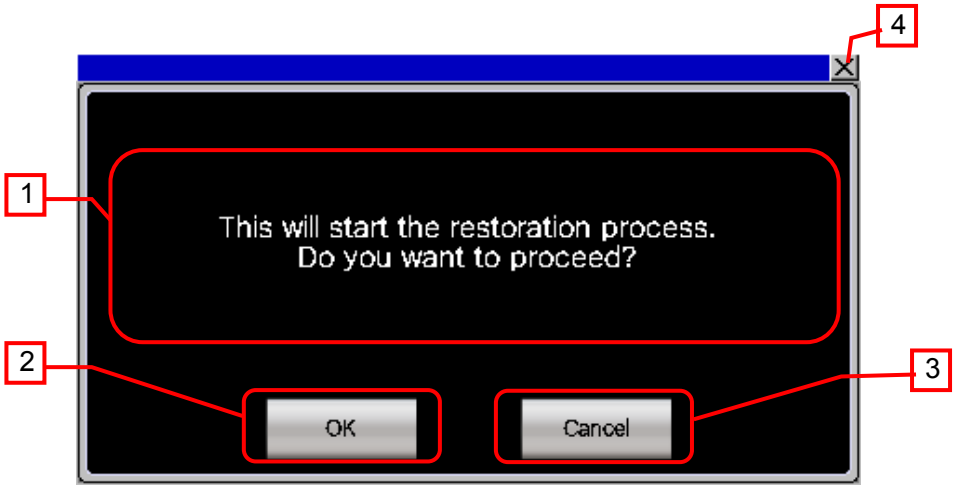
Description

1. Displays messages.
2. Closes the window screen.
3. 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".

5.3.33 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.
4. 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
1450	B289SB-01AF-CAS	ASLINKAMP	B-30012
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
2900	B284SB-01-12	ASLINKSENSOR	B-30018
A000	BL296SB-08F-3	ASLINKTERMINAL	B-30019
A001	BL296SB-08FS-3	ASLINKTERMINAL	B-30019
A300	BL296PB-08F-3	ASLINKTERMINAL	B-30019
A301	BL296PB-08FS-3	ASLINKTERMINAL	B-30019
A600	BL296XB-08F-3	ASLINKTERMINAL	B-30019
A601	BL296XB-08FS-3	ASLINKTERMINAL	B-30019

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	X0041	Parameter Access Completion Flag
	X0042	Parameter Access Error
	Y0030	Error Flag Clear Command
	Y0031	Address Auto Recognition Command
	Y0040	Parameter Access Request Command from Master to Slave
	Y0041	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	U03-G0 to U03-G15	Input Information Area (U03-G0 is also Used as Standard Device of Device Data Transfer)
	U03-G1	Used as Standard Device of Device Data Transfer
	U03-G20	Used as Standard Device of Device Data Transfer
	U03-G40	Used as Standard Device of Sensing Level
	U03-G4096 to U03-G4111	Output Information Area
	U03-G8192	Number of Error IDs Information
	U03-G8193 to U03-G8320	Error ID Information Storage Area
	U03-G8960	Number of Modules Information
	U03-G9216	Number of Connection IDs Information
	U03-G9217 to U03-G9344	Connection ID Information Storage Area
	U03-G9984	Number of Alarm IDs Information
	U03-G9985 to U03-G10112	Alarm ID Information Storage Area
	U03-G10256	Latest Error Code Storage Area
	U03-G10320	Parameter Access Setting
	U03-G10321	Parameter Access Target ID Specification
	U03-G10496 to U03-G10751	Parameter Storage Destination Memory Number (Output)
	U03-G11008 to U03-G11263	Parameter Storage Destination Memory Number (Input)
	U03-G12288 to U03-G18431	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

Type	Device No.	Application
Word	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

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 Script 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 Script Start Control Flag
	GB30008	Script No.30015 Start Trigger
	GB30009	Script No.30017 ID Match Flag
	GB30010	Script No.30001 Start Trigger
	GB30011	Script No.30030 Start Trigger
	GB30012	Script No.30031 Start Trigger
	GB30013	Script No.30032 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 Script Start Control Flag
	GB30018	Script No.30034 Start Trigger
	GB30019	Script No.30038 Start Trigger
	GB30020	Script No.30017 Combined Module Flag
	GB30021	Script No.30026 Control Flag
	GB30024	Parameter Individual Read Lamp Bit
	GB30025	Parameter Individual Write Lamp Bit
	GB30026	Script No.30040 Start Trigger
	GB30027	Script No.30041 Initial Start Script Start Control Flag
	GB30028	Error Occurrence Flag
	GB30029	Script No.30068 Start Trigger
	GB30030	Script No.30069 Start Trigger
	GB30044	Parameter Batch Read Flag
	GB30600	Output Judgment Flag
	GB30601	Input Judgment Flag
	GB54000	Script No.30046 Start Trigger
	GB54001	Script No.30047 Start Trigger
	GB54002	Script No.30065 Start Trigger
	GB54004	iQSS Backup Permission Flag
	GB54005	Script No.30056 Start Trigger
	GB54006	Script No.30058 Start Trigger
	GB54008	Script No.30055 Initial Start Script Start Control Flag
	GB54009	Script No.30043, No.30045 Control Flag
	GB54010	Script No.30062 Start Trigger

Type	Device No.	Application
Bit	GB54011	Script No.30064 Start Trigger
	GB54012	Script No.30058 Delete Range Specification Flag
	GB54013	Script No.30050 Start Trigger
	GB54014	B-30100 Screen Switching Switch Action Conditions
	GB54016	Recipe No.30001 Write Trigger
	GB54017	Recipe No.30001 Read Trigger
	GB54018	Recipe No.30002 Write Trigger
	GB54019	Recipe No.30002 Read Trigger
Word	GD31000 to GD31255	B-30004 Slave Module Status Display Lamp
	GD31256 to GD31511	B-30003 Slave Module Status Display Lamp
	GD31520	B-30003, B-30004 Cursor Display Bit Shift Device
	GD31521	B-30003, B-30004 Cursor Display Offset Device
	GD31522	B-30003, B-30004 Current Cursor Position Device
	GD31523	B-30002 Offset Value Operattion Device
	GD31524	B-30002 I/O Check Device
	GD31525	B-30002 System Map Scroll Offset
	GD31526	B-30002 System Map Parts Display Offset
	GD31528	B-30005 Sensor List Detail Display Offset
	GD31529	B-30005 Sensor List Scroll Offset
	GD31562	Device Data Transfer 1 External Control Device
	GD31563	Device Data Transfer 1 External Notification Device
	GD31564	Device Data Transfer 2 External Control Device
	GD31565	Device Data Transfer 2 External Notification Device
	GD31567	Device Data Transfer 2, Device Data Transfer 4 Offset
	GD31568	Device Data Transfer 3 External Control Device
	GD31569	Device Data Transfer 3 External Notification Device
	GD31570	Device Data Transfer 4 External Control Device
	GD31571	Device Data Transfer 4 External Notification Device
	GD31600 to GD31606	Script Work Area
	GD31607	B-30006 to B-30019 Slave Module Model Number Storage Device
	GD31608	B-30006 to B-30019 Status
	GD31609 to GD31616	B-30006 to B-30019 I/O Monitor Parts Display
	GD31617 to GD31631	B-30005 Module Type
	GD31633 to GD31647	B-30005 Status Detail
	GD31649 to GD31663	B-30005 User Definition Name
	GD31665 to GD31679	B-30005 Status Display Lamp
	GD31681	B-30006 to B-30019 Remedy Display Start Position Device
	GD31682 to GD31989	B-30006 to B-30019 Detail Information I/O Offset
	GD31990 to GD31997	B-30006 to B-30019 Detail Information I/O Mask Device (Object Script)
	GD32000 to GD32007	B-30006 to B-30019 Detail Information I/O Word Lamp Status Device
	GD32008	B-30003, B-30004 Cursor Position Display Device

Type	Device No.	Application
Word	GD33020	Script No.30003 I/O Area Check Device
	GD35000 to GD35127	B-30002 System Map Display Device
	GD35128	B-30002 Sensor Type Offset Device
	GD35130 to GD35642	B-30003, B-30004 Module Type Storage Device
	GD35643	B-30005 Module Information Display Trigger Device
	GD40000 to GD40015	Input Information Area
	GD40016 to GD40031	Output Information Area
	GD40032	Number of Error IDs
	GD40033 to GD40160	Connection ID Information Storage Area
	GD40289	Number of Modules Information
	GD40290	Number of Connection IDs
	GD40291 to GD40418	Connection ID Information Storage Area
	GD40547	Number of Alarms
	GD40548 to GD40675	Alarm ID Information Storage Area
	GD40804 to GD41059	Parameter Storage Destination Memory Number (Output)
	GD41060 to GD41315	Parameter Storage Destination Memory Number (Input)
	GD41316 to GD47459	Device Parameter (Batch)
	GD53604 to GD53632	Device Parameter (Individual)
	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-30101 No. Display Device
	GD54006	W-30100 to W-30101 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

Type	Device No.	Application
Word	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
	GD62300	Change Time Device (Year)
	GD62301	Change Time Device (Month)
	GD62302	Change Time Device (Day)
	GD62303	Change Time Device (Hour)
	GD62304	Change Time Device (Minute)
	GD62305	Change Time Device (Second)
	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 TMP28, TMP100 to TMP105, TMP110 to TMP111, TMP120 to TMP126, TMP200 to TMP206, TMP211 to TMP213, TMP220 to TMP223, TMP1000, TMP1001, TMP1010, TMP1020	For Script Operation

5.6 Comment List

Comment group No.	Comment No.	Where comments are used
250	No.1	B-30006 to B-30019
	No.2	B-30006 to B-30019
	No.4	B-30006 to B-30019
	No.8	B-30006 to B-30019
	No.32	B-30006 to B-30019
	No.100	B-30006 to B-30019
	No.101	B-30006 to B-30019
	No.102	B-30006 to B-30019
	No.103	B-30006 to B-30019
	No.104	B-30006 to B-30019
	No.200	B-30006 to B-30019
	No.201	B-30006 to B-30019
	No.202	B-30006 to B-30019
	No.300	B-30006 to B-30019
	No.301	B-30006 to B-30019
	No.302	B-30006 to B-30019
	No.303	B-30006 to B-30019
	No.304	B-30006 to B-30019
	No.305	B-30006 to B-30019
	No.400	B-30006 to B-30019
	No.401	B-30006 to B-30019
	No.500	B-30006 to B-30019
251	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.)	
252		
253	No.1 to No.255, No. 512 to No.767	B-30005 to B-30019
254	No.1	B-30002, B-30005 to B-30019
	No.2	B-30002, B-30005 to B-30019
	No.4	B-30002, B-30005 to B-30019
	No.8	B-30002, B-30005 to B-30019
	No.32	B-30002, B-30005 to B-30019
	No.100	B-30002, B-30005 to B-30019, W-30101
	No.101	B-30002, B-30005 to B-30019, W-30101
	No.102	B-30002, B-30005 to B-30019, W-30101
	No.103	B-30002, B-30005 to B-30019, W-30101
	No.104	B-30002, B-30005 to B-30019, W-30101
	No.200	B-30002, B-30005 to B-30019, W-30101
	No.201	B-30002, B-30005 to B-30019, W-30101
	No.202	B-30002, B-30005 to B-30019, W-30101
	No.300	B-30002, B-30005 to B-30019, W-30101
	No.301	B-30002, B-30005 to B-30019, W-30101
	No.302	B-30002, B-30005 to B-30019, W-30101
	No.303	B-30002, B-30005 to B-30019, W-30101
	No.304	B-30002, B-30005 to B-30019, W-30101
	No.305	B-30002, B-30005 to B-30019, W-30101
	No.400	B-30002, B-30005 to B-30019, W-30101

Comment group No.	Comment No.	Where comments are used
254	No.401	B-30002, B-30005 to B-30019, W-30101
	No.500	B-30002, B-30005 to B-30019, W-30101
	No.18433	W-30101
	No.18434	W-30101
	No.18435	W-30101
	No.18436	W-30101
	No.18437	W-30101
	No.18438	W-30101
	No.18439	W-30101
	No.18440	W-30101
	No.18441	W-30101
	No.32000	W-30101
	No.32001	W-30101
	No.32002	W-30101
	No.32003	W-30101
	No.32004	W-30101
	No.32005	W-30101
	No.32006	W-30101
	No.32007	W-30101
	No.32008	W-30101
	No.32009	W-30101
	No.32010	W-30101
	No.32011	W-30101
	No.32012	W-30101
	No.32013	W-30101
	No.32014	W-30101
	No.32015	W-30101
	No.32016	W-30101
	No.32017	W-30101
	No.32767	B-30002, B-30005 to B-30019
255	No.20	B-30002
	No.22	B-30005
	No.23	B-30005 to B-30019
	No.24	B-30005 to B-30019
	No.25	B-30006 to B-30019
	No.26	B-30002 to B-30019
	No.27	B-30001, B-30002
	No.28	B-30002 to B-30019
	No.29	B-30001 to B-30019
	No.30	B-30001
	No.31	B-30001
	No.32	B-30002 to B-30004
	No.33	B-30002 to B-30004
	No.34	B-30002 to B-30004
	No.35	B-30002 to B-30004
	No.36	B-30002 to B-30005
	No.37	B-30002
	No.38	B-30002 to B-30004, W-30001
	No.39	B-30002 to B-30004
	No.40	B-30002 to B-30005, W-30001, W-30003

Comment group No.	Comment No.	Where comments are used
255	No.41	B-30002 to B-30019
	No.42	B-30002 to B-30019
	No.43	B-30006 to B-30019
	No.44	-
	No.45	B-30003, B-30004
	No.46	B-30003, B-30004
	No.47	B-30003, B-30004
	No.48	B-30003, B-30004
	No.49	B-30005
	No.50	B-30005
	No.51	B-30005 to B-30019
	No.52	B-30006 to B-30019
	No.53	B-30006 to B-30019
	No.54	B-30006 to B-30019
	No.55	B-30006 to B-30019
	No.56	B-30006 to B-30019
	No.57	B-30006 to B-30019
	No.58	B-30006 to B-30019
	No.59	B-30006, B-30008, B-30009
	No.60	B-30006 to B-30019
	No.61	B-30006 to B-30019
	No.62	B-30010 to B-30015, B-30017, B-30018
	No.63	B-30010 to B-30015, B-30017, B-30018
	No.64	B-30013, B-30018
	No.65	B-30013, B-30018
	No.66	B-30013, B-30018
	No.67	B-30013, B-30018
	No.68	B-30011, B-30013, B-30017, B-30018
	No.69	B-30013, B-30017, B-30018
	No.70	B-30013 to B-30015, B-30017, B-30018
	No.71	B-30013 to B-30015, B-30017, B-30018
	No.72	B-30013, B-30017, B-30018
	No.73	B-30013, B-30017, B-30018
	No.74	B-30013 to B-30015, B-30017, B-30018
	No.75	B-30006 to B-30019
	No.76	B-30006, B-30008, B-30009
	No.77	B-30006, B-30008, B-30009
	No.78	W-30004
	No.79	W-30004
	No.80	W-30004
	No.81	W-30003
	No.82	W-30003
	No.83	W-30003
	No.84	W-30003
	No.85	W-30003
	No.86	W-30003
	No.87	W-30003
	No.88	W-30003
	No.89	-

Comment group No.	Comment No.	Where comments are used
255	No.90	-
	No.91	B-30003
	No.92	B-30004
	No.93	B-30006 to B-30019
	No.94	B-30001
	No.95	B-30002 to B-30019
	No.96	B-30001, B-30004
	No.97	B-30001, B-30003
	No.98	-
	No.99	B-30100
	No.100	-
	No.101	-
	No.102	-
	No.103	-
	No.104	-
	No.105	-
	No.106	-
	No.107	-
	No.108	-
	No.109	-
	No.110	-
	No.111	-
	No.112	-
	No.113	-
	No.114	-
	No.115	-
	No.116	-
	No.117	B-30010 to B-30012, B-30014, B-30015, B-30017
	No.118	B-30010 to B-30012, B-30014, B-30015, B-30017
	No.119	B-30010 to B-30012, B-30014, B-30015, B-30017
	No.120	B-30010 to B-30012, B-30014, B-30015, B-30017
	No.121	B-30010, B-30012, B-30014, B-30015
	No.122	B-30010, B-30012, B-30014, B-30015
	No.123	B-30010 to B-30012, B-30014, B-30015
	No.124	B-30014, B-30015
	No.125	B-30014, B-30016
	No.126	B-30010 to B-30012, B-30014, B-30015
	No.127	B-30010 to B-30012, B-30014 to B-30016
	No.128	W-30002
	No.129	B-30010, B-30012
	No.130	B-30010, B-30012
	No.131	B-30010, B-30012
	No.132	B-30010, B-30012
	No.133	B-30010 to B-30012
	No.134	B-30010 to B-30012
	No.135	B-30010 to B-30012
	No.136	B-30010, B-30012
	No.137	B-30010, B-30012
	No.138	B-30011

Comment group No.	Comment No.	Where comments are used
255	No.139	B-30011
	No.140	B-30011
	No.141	B-30011
	No.142	B-30011
	No.143	B-30011
	No.144	B-30014, B-30016
	No.145	B-30006 to B-30019
	No.201	B-30000, B-30100
	No.202	B-30101
	No.203	B-30102
	No.204	B-30103
	No.205	B-30104
	No.206	B-30105
	No.207	B-30106
	No.208	B-30100
	No.209	B-30100
	No.210	B-30101, B-30102, B-30104, B-30105
	No.211	B-30102, B-30103, B-30105, B-30106
	No.212	B-30101 to B-30106
	No.213	B-30103, B-30106
	No.214	B-30101
	No.215	B-30101
	No.216	B-30101
	No.217	B-30101
	No.218	B-30101, B-30104
	No.219	B-30101, B-30103, B-30104, B-30106
	No.220	B-30101, B-30104
	No.221	B-30101, B-30104
	No.222	B-30101, B-30102, B-30104, B-30105
	No.223	B-30102, B-30105
	No.224	B-30102, B-30105
	No.225	B-30101, B-30103, B-30106
	No.226	B-30101, B-30103, B-30106
	No.227	B-30103, B-30106
	No.228	B-30102, B-30105
	No.229	B-30105
	No.230	B-30102, B-30105
	No.231	B-30102
	No.232	B-30103, B-30105
	No.233	B-30103, B-30105
	No.234	B-30102, B-30105
	No.235	B-30102, B-30105
	No.236	B-30102
	No.237	B-30103
	No.238	B-30103
	No.239	B-30103, B-30106
	No.240	B-30103, B-30106
	No.241	B-30103, B-30106
	No.242	B-30103, B-30105

Comment group No.	Comment No.	Where comments are used
255	No.243	B-30103, B-30105
	No.244	B-30103, B-30105
	No.245	B-30101 to B-30103, B-30105, B-30106
	No.246	B-30103
	No.247	B-30103
	No.248	B-30103
	No.249	B-30104
	No.250	B-30104
	No.251	B-30104
	No.252	B-30106
	No.253	B-30106
	No.254	B-30106
	No.255	B-30106
	No.256	B-30106
	No.257	W-30004, W-30100 to W-30102
	No.258	W-30100, W-30102
	No.259	-
	No.260	-
	No.261	-
	No.262	-
	No.263	-
	No.264	-
	No.265	-
	No.266	-
	No.267	-
	No.268	W-30102
	No.269	W-30102
	No.270	W-30102
	No.271	W-30102
	No.272	W-30100
	No.273	B-30104
	No.274	B-30104
	No.275	B-30101, B-30104
	No.276	B-30101, B-30104
	No.277	B-30101, B-30104
	No.278	B-30104
	No.279	B-30101, B-30104
	No.280	W-30102
	No.281	W-30102
	No.282	W-30100
	No.283	B-30102, B-30103, B-30105, B-30106
	No.284	B-30102, B-30103, B-30105, B-30106
	No.285	-
	No.286	B-30103, B-30106
	No.287	B-30101
	No.288	B-30101
	No.289	W-30100

5.7 Device Data Transfer List

ID: 201 Transfer 1

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD31562
	Trigger Device	GD31562.b0
	Transfer Inverting Flag Device	GD31562.b1
External Notification Information	☑ External Notification Device	GD31563
	Device Data Transfer Notification Signal	GD31563.b0
	Device Data Transfer Error Notification Signal	GD31563.b15
Device	Block Number	10
Block 1	Device Type	Signed BIN16
	Points	16
	Source Device	U03-G0
	Destination Device	GD40000
	Offset	None
Block 2	Device Type	Signed BIN16
	Points	16
	Source Device	U03-G4096
	Destination Device	GD40016
	Offset	None
Block 3	Device Type	Signed BIN16
	Points	129
	Source Device	U03-G8192
	Destination Device	GD40032
	Offset	None
Block 4	Device Type	Signed BIN16
	Points	1
	Source Device	U03-G8960
	Destination Device	GD40289
	Offset	None

Item		Settings
Block 5	Device Type	Signed BIN16
	Points	129
	Source Device	U03-G9216
	Destination Device	GD40290
	Offset	None
Block 6	Device Type	Signed BIN16
	Points	129
	Source Device	U03-G9984
	Destination Device	GD40547
	Offset	None
Block 7	Device Type	Signed BIN16
	Points	256
	Source Device	U03-G10496
	Destination Device	GD40804
	Offset	None
Block 8	Device Type	Signed BIN16
	Points	256
	Source Device	U03-G11008
	Destination Device	GD41060
	Offset	None
Block 9	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD31562.b0
	Offset	None
Block 10	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30016
	Offset	None

ID: 202 Transfer 2

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD31564
	Trigger Device	GD31564.b0
	Transfer Inverting Flag Device	GD31564.b1
External Notification Information	☑ External Notification Device	GD31565
	Device Data Transfer Notification Signal	GD31565.b0
	Device Data Transfer Error Notification Signal	GD31565.b15
Device	Block Number	4
Block 1	Device Type	Signed BIN16
	Points	1
	Source Device	U03-G0
	Destination Device	GD53604
	Offset	Source GD31567
Block 2	Device Type	Signed BIN16
	Points	28
	Source Device	U03-G20
	Destination Device	GD53605
	Offset	Source GD31567
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD31564.b0
	Offset	None
Block 4	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GB30021
	Offset	None

ID: 203 Transfer 3

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD31568
	Trigger Device	GD31568.b0
	Transfer Inverting Flag Device	GD31568.b1
External Notification Information	☑ External Notification Device	GD31569
	Device Data Transfer Notification Signal	GD31569.b0
	Device Data Transfer Error Notification Signal	GD31569.b15
Device	Block Number	2
Block 1	Device Type	Signed BIN16
	Points	6144
	Source Device	U03-G12288
	Destination Device	GD41316
	Offset	None
Block 2	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD31568.b0
	Offset	None

ID: 204 Transfer 4

Item		Settings
Device Data Transfer Trigger	Trigger Type	Rise
	External Control Device	GD31570
	Trigger Device	GD31570.b0
	Transfer Inverting Flag Device	GD31570.b1
External Notification Information	☑ External Notification Device	GD31571
	Device Data Transfer Notification Signal	GD31571.b0
	Device Data Transfer Error Notification Signal	GD31571.b15
Device	Block Number	3

Item		Settings
Block 1	Device Type	Signed BIN16
	Points	19
	Source Device	GD53605
	Destination Device	U03-G1
	Offset	Destination GD31567
Block 2	Device Type	Bit
	Points	1
	Source Device	GB40
	Destination Device	Y0040
	Offset	None
Block 3	Device Type	Bit
	Points	1
	Source Device	GB41
	Destination Device	GD31570.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	Package1
	File Name	ARP30001.G1P
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	Package1
	File Name	ARP30002.G1P
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 B-30019, B-30100 to B-30106
Object script	Specified: B-30006 to B-30019, 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
set([b:Y41]); //Execute Parameter Batch Read			

Script No.	30029	Script name	Script30029
Comment	Alarm, Error Info Acquisition		
Data type	Signed BIN16	Trigger type	Sampling, 1 Sec

//Observe Error and Alarm
if(([w:GD40032] != [w:U03-G8192]) || ([w:GD40547] != [w:U03-G9984]))
{
  set([b:GD31568.b0]);
}			

Script No.	30033	Script name	Script30033
Comment	Status Read		
Data type	Signed BIN16	Trigger type	Fall, GD31568.b0

//Read Statuses
set([b:GD31562.b0]);			

Script No.	30002	Script name	Script30002
Comment	Device Data Transfer Flag Clear		
Data type	Signed BIN16	Trigger type	ON Sampling, 3 Sec, GD31562.b0

//Clear Flag for Device Data Transfer
 	
rst([b:GD31562.b0]);			

Script No.	30011	Script name	Script30011
Comment	Device Data Transfer Flag Clear		
Data type	Signed BIN16	Trigger type	ON Sampling, 3 Sec, GD31564.b0

//Clear Flag for Device Data Transfer
 	
rst([b:GD31564.b0]);			

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] = 3; //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]);
```

5.9.2 Screen script

Base screen 30001

Script No.	30035	Script name	Script30035
Comment	Device Data Transfer 3 Start		
Data type	Signed BIN16	Trigger type	ON, Y41

```
//Transfer parameters to internal devices using Device Data Transfer.
if([b:X41] == ON)
{
        set([b:GD31568.b0]);
        rst([b:Y41]);
}
```

Script No.	30041	Script name	Script30041
Comment	NW Map, Sensor List Disp.Control		
Data type	Signed BIN16	Trigger type	ON, GB40

```
//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:GD35000],128);
        fmov([w:TMP0000],[w:GD31617],48);
        set([b:GB30027]);
}
```

Script No.	30074	Script name	Script30074
Comment	Sensor Automatic Detection		
Data type	Signed BIN16	Trigger type	ON, Y0031

```
//Automatically Detect Sensor
if([b:X41] == ON)
{
        //If the parameter batch read flag is on, reset the auto detect flag and start the device data transfer.
        if([b:GB30044] == ON)
        {
                rst([b:Y31]);
                set([b:GD31568.b0]);
                rst([b:GB30044]);
        }
}else{
        //Set the flag when starting the parameter batch read.
        if([b:GB30044] == OFF)
        {
                set([b:GB30044]);
        }
}
```

Base screen 30002

Script No.	30023	Script name	Script30023
Comment	B-30002,30005 Scr.Start Process		
Data type	Signed BIN16	Trigger type	ON, GB40

```
//Read Statuses or Parameters
if([b:GB30007] == OFF)
{
	if(([w:U03-G8192] != 0) || ([w:U03-G9984] != 0))
	{
		set([b:GD31568.b0]);
	}else{
		set([b:GD31562.b0]);
	}
	set([b:GB30007]);
}
```

Script No.	30019	Script name	Script30019
Comment	Script No.30005 Start		
Data type	Signed BIN16	Trigger type	OFF, GD31563.b0

```
//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:GD31526] = 0;
[w:TMP0001] = 0;
[w:TMP0004] = 0;
[w:TMP0001] = [w:GD31525];
[w:TMP0009] = 0;

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

while([w:GD31526] < 32)
{
	//Check Connection Status
	if(([w:GD40290] != 0) && ([w:TMP0001] < [w:GD40290]))
	{
		[w:GD31600] = [w:GD40291[w:TMP0001]];	//Connection ID
		[w:TMP0001] = [w:TMP0001] + 1;

		if(([w:GD31600] & 0x0200) == 0)
		{
			set([b:GB30600]);//Output Bit
			[w:GD31601] = ([w:GD31600] & 0x0200) >> 9;
		}else{
			set([b:GB30601]);//Input Bit
			[w:GD31601] = ([w:GD31600] & 0x0200) >> 9;
		}

		//Device Parameter
		[w:TMP0004] = [w:GD31600] & 0x00FF;	//Offset to Obtain Device Parameter
		if([b:GB30600] == ON)
		{
			//Output
			[w:TMP0005] = [w:GD40804[w:TMP0004]];
		}else{
```

```
                        //Input
                        [w:TMP0005] = [w:GD41060[w:TMP0004]];
                }
                [w:GD31604] = [w:TMP0005] - 12288;

                //I/O Points Pattern
                [w:TMP0006] = [w:GD31604] + 43;  //Parameter Offset
                [w:TMP0007] = [w:GD41316[w:TMP0006]];
                [w:GD31605] = ([w:TMP0007] & 0x00C0) >> 6;          //0:  Input,  1:  Output,  2:
Combined
                [w:GD31606] = ([w:TMP0007] & 0x003F) + 1;           //Points

                //Module Model Number
                [w:TMP0008] = [w:GD31604] + 44;  //Parameter Offset
                [w:GD31607] = [w:GD41316[w:TMP0008]];


                //Alarm Information
                if([w:GD40547] != 0)
                {
                        [w:TMP0004] = 0;
                        while([w:TMP0004] < [w:GD40547])
                        {
                                if([w:GD40548[w:TMP0004]] == [w:GD31600])
                                {
                                        [w:GD31602] = 1; //Alarm Occurrence
                                        break;
                                }
                                [w:TMP0004] = [w:TMP0004] + 1;
                        }
                }

                //Error Information
                if([w:GD40032] != 0)
                {
                        [w:TMP0004] = 0;
                        while([w:TMP0004] < [w:GD40032])
                        {
                                if([w:GD40033[w:TMP0004]] == [w:GD31600])
                                {
                                        [w:GD31603] = 1; //Error Occurrence
                                        break;
                                }
                                [w:TMP0004] = [w:TMP0004] + 1;
                        }
                }

                //Display
                if([w:GD31603] == 1)
                {

                        //Error Occurrence
                        switch([w:GD31605])
                        {
                                case 0:  [w:GD35000[w:GD31526]]  =  30030  +  [w:TMP0009];
        //Input
                                                break;

                                case 1:  [w:GD35000[w:GD31526]]  =  30021  +  [w:TMP0009];
        //Output
                                                break;
```

```
                                case 2:  [w:GD35000[w:GD31526]]  =  30039  +  [w:TMP0009];
        //Combined
                                                        break;
                        }


                        [w:GD35032[w:GD31526]] = 0;
                        [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;  //ID
                        [w:GD35096[w:GD31526]] = [w:GD31607];    //Module Model No.
                }else{
                        if([w:GD31602] == 1)
                        {
                                //Alarm Occurrence
                                switch([w:GD31605])
                                {
                                        case 0:  [w:GD35000[w:GD31526]]     =     30027     +
[w:TMP0009];     //Input
                                                                break;

                                        case 1:  [w:GD35000[w:GD31526]]     =     30018     +
[w:TMP0009];     //Output
                                                                break;

                                        case 2:  [w:GD35000[w:GD31526]]     =     30036     +
[w:TMP0009];     //Combined
                                                                break;

                                }
                                [w:GD35032[w:GD31526]] = 0;
                                [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;  //ID
                                [w:GD35096[w:GD31526]] = [w:GD31607];    //Module Model No.
                        }else{
                                //正常
                                switch([w:GD31605])
                                {
                                        case 0:  [w:GD35000[w:GD31526]]     =     30024     +
[w:TMP0009];     //Input
                                                                break;

                                        case 1:  [w:GD35000[w:GD31526]]     =     30015     +
[w:TMP0009];     //Output
                                                                break;

                                        case 2:  [w:GD35000[w:GD31526]]     =     30033     +
[w:TMP0009];     //Combined
                                                                break;

                                }
                                [w:GD35032[w:GD31526]] = 0;
                                [w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;  //ID
                                [w:GD35096[w:GD31526]] = [w:GD31607];    //Module Model No.
                        }
                }

                [w:GD31526] = [w:GD31526] + 1;
                rst([b:GB30600]); //Output Bit
                rst([b:GB30601]); //Input Bit

                //Work Area Clear
                [w:TMP0004] = 0;
```

```
                fmov([w:TMP0004],[w:GD31600],5);
        }else{
                //If data does not exist, do not display.
                [w:GD35000[w:GD31526]] = 0xFFFF;
                [w:GD35032[w:GD31526]] = 0xFFFF;
                [w:GD35064[w:GD31526]] = 0xFFFF;
                [w:GD35096[w:GD31526]] = 0xFFFF;

                [w:GD31526] = [w:GD31526] + 1;
                rst([b:GB30600]); //Output Bit
                rst([b:GB30601]); //Input Bit
        }
}

//Trigger Reset
rst([b:GB30000]);
```

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]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD31000],512);      //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004],[w:GD31520],3);        //Clear Cursor Position Information(B-30003,4)
fmov([w:TMP0004],[w:GD31600],8);        //Work Area Clear
[w:GD32008] = 0; //Cursor Position Information Clear(B-30003,4)

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

Script No.	30006	Script name	Script30006
Comment	Scroll Up		
Data type	Signed BIN16	Trigger type	Rise, GB30004

```
//Scroll List Up
if([w:GD31525] != 0)
{
        [w:GD31525] = [w:GD31525] - 32;
        set([b:GB30000]);
}
```

Script No.	30007	Script name	Script30007
Comment	Scroll Down		
Data type	Signed BIN16	Trigger type	Rise, GB30005

```
//Scroll List Down
if((([w:GD31525] + 32) < [w:U03-G8960]) && ([w:U03-G8960] > 32))
{
        [w:GD31525] = [w:GD31525] + 32;
        set([b:GB30000]);
}
```

Script No.	30010	Script name	Script30010
Comment	B-30002 Detail Screen Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30006

```
//Determine detail screen to display.
[w:TMP0000] = [w:GD31523] & 0x00FF;

if(([w:GD31524] >= 30015) && ([w:GD31524] <= 30023))
{
```

```
        //Output
        [w:GD31567] = [w:GD40804[w:TMP0000]];   //Offset
        [w:TMP0001] = [w:TMP0000];
}else{
        //Input
        [w:GD31567] = [w:GD41060[w:TMP0000]];   //Offset
        [w:TMP0001] = [w:TMP0000] + 0x0200;
}

[w:U03-G10320] = 0;         //Read Individual Parameter
[w:U03-G10321] = [w:TMP0001];     //Read Target ID

//Select base screen to display.
switch([w:GD35096[w:GD35128]])
{
        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;

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

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 0x2900:    [w:GD62000] = 30018;
                                    break;

        case 0xA000:    [w:GD62000] = 30019;
                                    break;

        case 0xA001:    [w:GD62000] = 30019;
                                    break;

        case 0xA300:    [w:GD62000] = 30019;
                                    break;

        case 0xA301:    [w:GD62000] = 30019;
                                    break;

        case 0xA600:    [w:GD62000] = 30019;
                                    break;

        case 0xA601:    [w:GD62000] = 30019;
                                    break;

        default:                    break;
}


rst([b:GB30006]);
```

Base screen 30003

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

```
if([b:GB30003] == OFF)
{

        fmov([w:TMP0000],[w:GD31600],10);
        [w:TMP0110] = 0; //Initialize Temporary Area for Comparison
        [w:TMP0111] = 0; //Initialize Temporary Area for Comparison
        set([b:GD31562.b0]);        //Obtain Parameters and Display Parts
        set([b:GB30003]);//Turn on the flag not to work when displaying the screen for the second time or later.
}
```

Script No.	30008	Script name	Script30008
Comment	Script No.30003 Start		
Data type	Signed BIN16	Trigger type	OFF, GD31563.b0

```
//After reading the status, start Script No.30003.
if([b:GB30016] == OFF)
{
        set([b:GB30001]);
        set([b:GB30016]);
}
```

Script No.	30003	Script name	Script30003
Comment	B-30003 Parts Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30001

```
//Display Parts
[w:TMP0020] = 0;

while(([w:TMP0020] < [w:GD40289]) && ([w:GD40289] > 0))
{
        //Obtain Connection ID
```

```
[w:GD31600] = [w:GD40291[w:TMP0020]];
[w:TMP0020] = [w:TMP0020] + 1;

if(([w:GD31600] & 0x0200) == 0)
{
	set([b:GB30600]);//Output Bit
}else{
	set([b:GB30601]);//Input Bit
}

//Alarm Information
if([w:GD40547] != 0)
{
	[w:TMP0004] = 0;
	while([w:TMP0004] < [w:GD40547])
	{
		if([w:GD40548[w:TMP0004]] == [w:GD31600])
		{
			[w:GD31602] = 1; //Alarm Occurrence
			break;
		}
		[w:TMP0004] = [w:TMP0004] + 1;
	}
}

//Error Information
if([w:GD40032] != 0)
{
	[w:TMP0004] = 0;
	while([w:TMP0004] < [w:GD40032])
	{
		if([w:GD40033[w:TMP0004]] == [w:GD31600])
		{
			[w:GD31603] = 1; //Error Occurrence
			break;
		}
		[w:TMP0004] = [w:TMP0004] + 1;
	}
}

//Device Parameter
[w:TMP0004] = [w:GD31600] & 0x00FF;	//Offset to Obtain Device Parameter

if([b:GB30600] == ON)
{
	//Output
	[w:TMP0005] = [w:GD40804[w:TMP0004]];
	[w:TMP0012] = [w:TMP0004];
}else{
	//Input
	[w:TMP0005] = [w:GD41060[w:TMP0004]];
	[w:TMP0012] = [w:TMP0004] + 0x0100;
}
[w:GD31604] = [w:TMP0005] - 0x3000;

//I/O Points Pattern
[w:TMP0006] = [w:GD31604] + 43;  //Parameter Offset
[w:TMP0007] = [w:GD41316[w:TMP0006]];
[w:GD31605] = ([w:TMP0007] & 0x00C0) >> 6;	//0: Input, 1: Output, 2: Combined
[w:GD31606] = ([w:TMP0007] & 0x003F) + 1;	//Points
```

```
	//Module Model Number
	[w:TMP0006] = [w:GD31604] + 44;
	[w:GD35130[w:TMP0012]] = [w:GD41316[w:TMP0006]];

	//ON/OFF Information
	[w:TMP0008] = [w:GD31600] & 0x00FF;
	if([w:GD31605] == 0)		//Input Case
	{
		[w:TMP0004] = 0;

		while([w:TMP0004] < [w:GD31606])
		{
			[w:TMP0009] = [w:TMP0008] / 16;
			[w:TMP0010] = [w:TMP0008] % 16;

			[w:GD33020] = 0x0001;
			[w:GD33020] = [w:GD33020] << [w:TMP0010];

			if([w:GD31603] == 1)		//Check If Error Occurred
			{
				[w:GD31256[w:TMP0008]] = 3;		//Error Occurrence & Input ON
			}else{
				if([w:GD31602] == 1)		//Check If Alarms Occurred
				{
					[w:GD31256[w:TMP0008]] = 2;		//Alarm	Occurrence
& Input ON
				}else{
					[w:GD31256[w:TMP0008]] = 1;		//Normal
				}
			}

			[w:TMP0008] = [w:TMP0008] + 1;
			[w:TMP0004] = [w:TMP0004] + 1;
		}
	}else{
		if([w:GD31605] == 1)		//Output Case
		{
			[w:TMP0004] = 0;

			while([w:TMP0004] < [w:GD31606])
			{
				[w:TMP0009] = [w:TMP0008] / 16;
				[w:TMP0010] = [w:TMP0008] % 16;

				[w:GD33020] = 0x0001;
				[w:GD33020] = [w:GD33020] << [w:TMP0010];

				if([w:GD31603] == 1)		//Check If Error Occurred
				{
					[w:GD31000[w:TMP0008]] = 3;		//Error Occurrence &
Output ON
				}else{
					if([w:GD31602] == 1)		//Check If Alarms Occurred
					{
						[w:GD31000[w:TMP0008]] = 2;		//Alarm
Occurrence & Output ON
					}else{
						[w:GD31000[w:TMP0008]] = 1;		//Normal
					}
				}
```

```
                                [w:TMP0008] = [w:TMP0008] + 1;
                                [w:TMP0004] = [w:TMP0004] + 1;
                        }
                }else{
                        if([w:GD31605] == 2)        //Combined Case
                        {
                                [w:TMP0004] = 0;
                                [w:TMP0011] = [w:GD31606] / 2;    //If combined, points are
reduced by half.
                                while([w:TMP0004] < [w:TMP0011])
                                {
                                        [w:TMP0009] = [w:TMP0008] / 16;
                                        [w:TMP0010] = [w:TMP0008] % 16;

                                        [w:GD33020] = 0x0001;
                                        [w:GD33020] = [w:GD33020] << [w:TMP0010];

                                        //Output
                                        if([w:GD31603] == 1)        //Check If Error Occurred
                                        {
                                                [w:GD31000[w:TMP0008]] = 3;       //Error
Occurrence & Output ON
                                        }else{
                                                if([w:GD31602] == 1)        //Check    If    Alarms
Occurred
                                                {
                                                        [w:GD31000[w:TMP0008]]      =      2;
        //Alarm Occurrence & Output ON
                                                }else{
                                                        [w:GD31000[w:TMP0008]]      =      1;
        //Normal
                                                }
                                        }

                                        //Input
                                        if([w:GD31603] == 1)        //Check If Error Occurred
                                        {
                                                [w:GD31256[w:TMP0008]] = 3;       //Error
Occurrence & Input ON
                                        }else{
                                                if([w:GD31602] == 1)        //Check    If    Alarms
Occurred
                                                {
                                                        [w:GD31256[w:TMP0008]]      =      2;
        //Alarm Occurrence & Input ON
                                                }else{
                                                        [w:GD31256[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:GD31600],5);

}

//Display Cursor
if([w:GD62000] == 30004)
{
        //Output
        [w:GD31520] = 0; //Left Right
        [w:GD31521] = 0; //Up Down
        [w:GD31522] = [w:GD31520] + [w:GD31521];
        [w:GD32008] = [w:GD31522];          //Cursor Position Display

        if([w:GD31000[w:GD31522]] < 4)
        {
                [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
        }
}else{
        //Input
        [w:GD31520] = 256;          //Left Right
        [w:GD31521] = 0; //Up Down
        [w:GD31522] = [w:GD31520] + [w:GD31521];
        [w:GD32008] = [w:GD31522] - 256; //Cursor Position Display

        if([w:GD31000[w:GD31522]] < 4)
        {
                [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
        }
}

rst([b:GB30001]);
```

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]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD31000],512);          //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004],[w:GD31520],3);            //Clear Cursor Position Information(B-30003,4)
fmov([w:TMP0004],[w:GD31600],8);            //Work Area Clear
[w:GD32008] = 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 Detail Screen Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30002

```
//Display Detail Information Screen of Cursor Position

//Obtain Temporary ID
if([w:GD31522] < 256)
{
        [w:TMP0000] = [w:GD31522];          //Output ID
}else{
```

```
	[w:TMP0000] = [w:GD31522] - 256; //Input ID
}

if([w:GD31000[w:GD31522]] > 4)	//Judge whether a module exists in the cursor position.
{
	if([w:GD31522] < 256)
	{
		//Output
		[w:TMP0002] = [w:TMP0000];
		while([w:TMP0002] >= 0)
		{
			if([w:GD31000[w:TMP0002]] != 0)
			{
				//Output
				[w:TMP0001] = 0;
				while([w:TMP0001] < [w:GD40290]) //Repeat the Number of
Connection ID Counts
				{
					if([w:GD40291[w:TMP0001]] == [w:TMP0002])
	//Judge ID
					{
						set([b:GB30009]); //ID Match Flag
						break;
					}else{
						[w:TMP0001] = [w:TMP0001] + 1;
					}
				}

				//Combined Output Side Case
				if([b:GB30009] == OFF)
				{
					[w:TMP0001] = 0;
					while([w:TMP0001] < [w:GD40290]) //Repeat the Number
of Connection ID Counts
					{
						if([w:GD40291[w:TMP0001]] == ([w:TMP0002] +
0x0200))//Judge ID
						{

							//Judge whether the matched ID is that
of a combined module or not.
							[w:TMP0010] =
[w:GD41060[w:TMP0002]];	//Parameter Top Address
							[w:TMP0011] = [w:TMP0010] - 0x3000;

							[w:TMP0012] =
[w:GD41359[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;
						}else{
							[w:TMP0001] = [w:TMP0001] + 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:GD65535] = [w:TMP0002];
                                [w:TMP0001] = 0;
                                while([w:TMP0001] < [w:GD40290])//Repeat the Number of Connection ID
Counts
                                {
                                        if([w:GD40291[w:TMP0001]]    ==    ([w:TMP0002]    +    0x0200))
        //Judge ID
                                        {
                                                set([b:GB30009]);//ID Match Flag
                                                break;
                                        }else{
                                                [w:TMP0001] = [w:TMP0001] + 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:GD31522] < 256)
                        {
                                if([b:GB30020] == OFF)
                                {
                                        //Output
                                        [w:GD31567] = [w:GD40804[w:TMP0002]];    //Offset
                                        [w:TMP0003] = [w:TMP0002];
                                        [w:TMP0004] = [w:TMP0002];
                                }else{
                                        //If combined, refer to the input side.
                                        [w:GD31567] = [w:GD41060[w:TMP0002]];    //Offset
                                        [w:TMP0003] = [w:TMP0002] + 0x0200;
```

```
                    [w:TMP0004] = [w:TMP0002] + 0x0100;
            }
}else{
            //Input
            [w:GD31567] = [w:GD41060[w:TMP0002]];    //Offset
            [w:TMP0003] = [w:TMP0002] + 0x0200;
            [w:TMP0004] = [w:TMP0002] + 0x0100;
}

[w:U03-G10320] = 0;         //Read Individual Parameter
[w:U03-G10321] = [w:TMP0003];     //Read Target ID

//Select base screen to display.
switch([w:GD35130[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;

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

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 0x2900:    [w:GD62000] = 30018;
                                                    break;

                        case 0xA000:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA001:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA300:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA301:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA600:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA601:    [w:GD62000] = 30019;
                                                    break;

                        default:                    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:GD31562.b0] == OFF) && ([b:GB30001] == OFF))
        {
        //Save Current Cursor Position
        [w:TMP0122] = [w:GD31522];

        //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:GD31520] = (16 - ([w:TMP0124] + 1)) + 256;

        //Y Coordinate
        [w:TMP0125] = [w:TMP0121] - Input_Y;
        [w:TMP0126] = [w:TMP0125] / Object_Y;
        [w:GD31521] = [w:TMP0126] * 16;
```

```
	//Offset
	[w:GD31522] = [w:GD31520] + [w:GD31521];

	//If the offset position does not change, displayed parts do not change.
	if([w:TMP0122] != [w:GD31522])
	{
		//Cursor Display
		if([w:GD31000[w:GD31522]] < 4)
		{
			[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
		}

		//Cursor Delete
		if([w:GD31000[w:TMP0122]] < 4)
		{
			[w:GD31000[w:TMP0122]] = 0;
		}else{
			[w:GD31000[w:TMP0122]] = [w:GD31000[w:TMP0122]] - 4;
		}
	}
	}

	[w:GD32008] = [w:GD31522] - 256; //Cursor Position Display
}
```

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

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:GD31522];		//Evacuate Previous Cursor Position

if([w:GD31520] == 256)
{
	[w:GD31520] = 271;
}else{
	[w:GD31520] = [w:GD31520] -1;
}

[w:TMP0000] = [w:GD31520];

[w:GD31522] = [w:TMP0000] + [w:GD31521];		//Offset

if([w:GD31000[w:TMP0020]]	>= 4)
{
	[w:GD31000[w:TMP0020]]	= [w:GD31000[w:TMP0020]]	- 4;
}else{
	[w:GD31000[w:TMP0020]]	= 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;

[w:GD32008] = [w:GD31522] - 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:GD31522];          //Evacuate Previous Cursor Position

if([w:GD31521] == 240)
{
        [w:GD31521] = 0;
}else{
        [w:GD31521] = [w:GD31521] + 16;
}

[w:GD31522] = [w:GD31520] + [w:GD31521];            //Offset

if([w:GD31000[w:TMP0020]]   >= 4)
{
        [w:GD31000[w:TMP0020]]   = [w:GD31000[w:TMP0020]]   - 4;
}else{
        [w:GD31000[w:TMP0020]]   = 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;

[w:GD32008] = [w:GD31522] - 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:GD31522];          //Evacuate Previous Cursor Position

if([w:GD31521] == 0)
{
        [w:GD31521] = 240;
}else{
        [w:GD31521] = [w:GD31521] - 16;
}

[w:GD31522] = [w:GD31520] + [w:GD31521];            //Offset

if([w:GD31000[w:TMP0020]]   >= 4)
{
        [w:GD31000[w:TMP0020]]   = [w:GD31000[w:TMP0020]]   - 4;
}else{
        [w:GD31000[w:TMP0020]]   = 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;

[w:GD32008] = [w:GD31522] - 256; //Cursor Position Display
```

*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 Screen Initialization		
Data type	Signed BIN16	Trigger type	ON, GB40

```
if([b:GB30003] == OFF)
{

        fmov([w:TMP0000],[w:GD31600],10);
```

```
	[w:TMP0110] = 0; //Initialize Temporary Area for Comparison
	[w:TMP0111] = 0; //Initialize Temporary Area for Comparison
	set([b:GD31562.b0]);	//Obtain Parameters and Display Parts
	set([b:GB30003]);//Turn on the flag not to work when displaying the screen for the second time or later.
}
```

Script No.	30008	Script name	Script30008
Comment	Script No.30003 Start		
Data type	Signed BIN16	Trigger type	OFF, GD31563.b0

```
//After reading the status, start Script No.30003.
if([b:GB30016] == OFF)
{
	set([b:GB30001]);
	set([b:GB30016]);
}
```

Script No.	30003	Script name	Script30003
Comment	B-30003 Parts Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30001

```
//Display Parts
[w:TMP0020] = 0;

while(([w:TMP0020] < [w:GD40289]) && ([w:GD40289] > 0))
{
	//Obtain Connection ID
	[w:GD31600] = [w:GD40291[w:TMP0020]];
	[w:TMP0020] = [w:TMP0020] + 1;

	if(([w:GD31600] & 0x0200) == 0)
	{
		set([b:GB30600]);//Output Bit
	}else{
		set([b:GB30601]);//Input Bit
	}

	//Alarm Information
	if([w:GD40547] != 0)
	{
		[w:TMP0004] = 0;
		while([w:TMP0004] < [w:GD40547])
		{
			if([w:GD40548[w:TMP0004]] == [w:GD31600])
			{
				[w:GD31602] = 1; //Alarm Occurrence
				break;
			}
			[w:TMP0004] = [w:TMP0004] + 1;
		}
	}

	//Error Information
	if([w:GD40032] != 0)
	{
		[w:TMP0004] = 0;
		while([w:TMP0004] < [w:GD40032])
		{
			if([w:GD40033[w:TMP0004]] == [w:GD31600])
			{
				[w:GD31603] = 1; //Error Occurrence
				break;
			}
```

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

        //Device Parameter
        [w:TMP0004] = [w:GD31600] & 0x00FF;       //Offset to Obtain Device Parameter

        if([b:GB30600] == ON)
        {
                //Output
                [w:TMP0005] = [w:GD40804[w:TMP0004]];
                [w:TMP0012] = [w:TMP0004];
        }else{
                //Input
                [w:TMP0005] = [w:GD41060[w:TMP0004]];
                [w:TMP0012] = [w:TMP0004] + 0x0100;
        }
        [w:GD31604] = [w:TMP0005] - 0x3000;

        //I/O Points Pattern
        [w:TMP0006] = [w:GD31604] + 43;  //Parameter Offset
        [w:TMP0007] = [w:GD41316[w:TMP0006]];
        [w:GD31605] = ([w:TMP0007] & 0x00C0) >> 6;          //0: Input, 1: Output, 2: Combined
        [w:GD31606] = ([w:TMP0007] & 0x003F) + 1;           //Points

        //Module Model Number
        [w:TMP0006] = [w:GD31604] + 44;
        [w:GD35130[w:TMP0012]] = [w:GD41316[w:TMP0006]];

        //ON/OFF Information
        [w:TMP0008] = [w:GD31600] & 0x00FF;
        if([w:GD31605] == 0)        //Input Case
        {
                [w:TMP0004] = 0;

                while([w:TMP0004] < [w:GD31606])
                {
                        [w:TMP0009] = [w:TMP0008] / 16;
                        [w:TMP0010] = [w:TMP0008] % 16;

                        [w:GD33020] = 0x0001;
                        [w:GD33020] = [w:GD33020] << [w:TMP0010];

                        if([w:GD31603] == 1)        //Check If Error Occurred
                        {
                                [w:GD31256[w:TMP0008]] = 3;       //Error Occurrence & Input ON
                        }else{
                                if([w:GD31602] == 1)        //Check If Alarms Occurred
                                {
                                        [w:GD31256[w:TMP0008]] = 2;       //Alarm    Occurrence
& Input ON
                                }else{
                                        [w:GD31256[w:TMP0008]] = 1;       //Normal
                                }
                        }

                        [w:TMP0008] = [w:TMP0008] + 1;
                        [w:TMP0004] = [w:TMP0004] + 1;
                }
        }else{
                if([w:GD31605] == 1)        //Output Case
```

```
		{
			[w:TMP0004] = 0;

			while([w:TMP0004] < [w:GD31606])
			{
				[w:TMP0009] = [w:TMP0008] / 16;
				[w:TMP0010] = [w:TMP0008] % 16;

				[w:GD33020] = 0x0001;
				[w:GD33020] = [w:GD33020] << [w:TMP0010];

				if([w:GD31603] == 1)		//Check If Error Occurred
				{
					[w:GD31000[w:TMP0008]] = 3;		//Error Occurrence &
Output ON
				}else{
					if([w:GD31602] == 1)		//Check If Alarms Occurred
					{
						[w:GD31000[w:TMP0008]] = 2;		//Alarm
Occurrence & Output ON
					}else{
						[w:GD31000[w:TMP0008]] = 1;		//Normal
					}
				}

				[w:TMP0008] = [w:TMP0008] + 1;
				[w:TMP0004] = [w:TMP0004] + 1;
			}
		}else{
			if([w:GD31605] == 2)		//Combined Case
			{
				[w:TMP0004] = 0;
				[w:TMP0011] = [w:GD31606] / 2;	//If combined, points are
reduced by half.
				while([w:TMP0004] < [w:TMP0011])
				{
					[w:TMP0009] = [w:TMP0008] / 16;
					[w:TMP0010] = [w:TMP0008] % 16;

					[w:GD33020] = 0x0001;
					[w:GD33020] = [w:GD33020] << [w:TMP0010];

					//Output
					if([w:GD31603] == 1)		//Check If Error Occurred
					{
						[w:GD31000[w:TMP0008]] = 3;		//Error
Occurrence & Output ON
					}else{
						if([w:GD31602] == 1)		//Check If Alarms
Occurred
						{
							[w:GD31000[w:TMP0008]] = 2;
		//Alarm Occurrence & Output ON
						}else{
							[w:GD31000[w:TMP0008]] = 1;
		//Normal
						}
					}

					//Input
					if([w:GD31603] == 1)		//Check If Error Occurred
```

```
                                        {
                                                [w:GD31256[w:TMP0008]] = 3;       //Error
Occurrence & Input ON
                                        }else{
                                                if([w:GD31602] == 1)       //Check    If    Alarms
Occurred
                                                {
                                                        [w:GD31256[w:TMP0008]]      =      2;
        //Alarm Occurrence & Input ON
                                                }else{
                                                        [w:GD31256[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:GD31600],5);

}

//Display Cursor
if([w:GD62000] == 30004)
{
        //Output
        [w:GD31520] = 0; //Left Right
        [w:GD31521] = 0; //Up Down
        [w:GD31522] = [w:GD31520] + [w:GD31521];
        [w:GD32008] = [w:GD31522];          //Cursor Position Display

        if([w:GD31000[w:GD31522]] < 4)
        {
                [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
        }
}else{
        //Input
        [w:GD31520] = 256;          //Left Right
        [w:GD31521] = 0; //Up Down
        [w:GD31522] = [w:GD31520] + [w:GD31521];
        [w:GD32008] = [w:GD31522] - 256; //Cursor Position Display

        if([w:GD31000[w:GD31522]] < 4)
        {
                [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
        }
}

rst([b:GB30001]);
```

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:GD31522];          //Previous Cursor Position

if([w:GD31520] ==    15)
{
          [w:GD31520] = 0;
}else{
          [w:GD31520] = [w:GD31520] + 1;
}

[w:TMP0000] = [w:GD31520];

[w:GD31522] = [w:TMP0000] + [w:GD31521];              //Offset

if([w:GD31000[w:TMP0020]]    >= 4)
{
          [w:GD31000[w:TMP0020]]    = [w:GD31000[w:TMP0020]]    - 4;
}else{
          [w:GD31000[w:TMP0020]]    = 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;

[w:GD32008] = [w:GD31522];          //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:GD31522];          //Evacuate Previous Cursor Position

if([w:GD31520] ==    0)
{
          [w:GD31520] = 15;
}else{
          [w:GD31520] = [w:GD31520] -1;
}

[w:TMP0000] = [w:GD31520];

[w:GD31522] = [w:TMP0000] + [w:GD31521];              //Offset

if([w:GD31000[w:TMP0020]]    >= 4)
{
          [w:GD31000[w:TMP0020]]    = [w:GD31000[w:TMP0020]]    - 4;
}else{
          [w:GD31000[w:TMP0020]]    = 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 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:GD31522];          //Evacuate Previous Cursor Position

if([w:GD31521] == 240)
```

```
{
        [w:GD31521] = 0;
}else{
        [w:GD31521] = [w:GD31521] + 16;
}

[w:GD31522] = [w:GD31520] + [w:GD31521];            //Offset

if([w:GD31000[w:TMP0020]]   >= 4)
{
        [w:GD31000[w:TMP0020]]   = [w:GD31000[w:TMP0020]]   - 4;
}else{
        [w:GD31000[w:TMP0020]]   = 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;

[w:GD32008] = [w:GD31522];          //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:GD31522];          //Evacuate Previous Cursor Position

if([w:GD31521] == 0)
{
        [w:GD31521] = 240;
}else{
        [w:GD31521] = [w:GD31521] - 16;
}

[w:GD31522] = [w:GD31520] + [w:GD31521];            //Offset

if([w:GD31000[w:TMP0020]]   >= 4)
{
        [w:GD31000[w:TMP0020]]   = [w:GD31000[w:TMP0020]]   - 4;
}else{
        [w:GD31000[w:TMP0020]]   = 0;
}
[w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
```

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]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD31000],512);          //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004],[w:GD31520],3);            //Clear Cursor Position Information(B-30003,4)
fmov([w:TMP0004],[w:GD31600],8);            //Work Area Clear
[w:GD32008] = 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 Detail Screen Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30002

```
//Display Detail Information Screen of Cursor Position

//Obtain Temporary ID
if([w:GD31522] < 256)
{
	[w:TMP0000] = [w:GD31522];	//Output ID
}else{
	[w:TMP0000] = [w:GD31522] - 256; //Input ID
}

if([w:GD31000[w:GD31522]] > 4)	//Judge whether a module exists in the cursor position.
{
	if([w:GD31522] < 256)
	{
		//Output
		[w:TMP0002] = [w:TMP0000];
		while([w:TMP0002] >= 0)
		{
			if([w:GD31000[w:TMP0002]] != 0)
			{
				//Output
				[w:TMP0001] = 0;
				while([w:TMP0001] < [w:GD40290]) //Repeat the Number of Connection ID Counts
				{
					if([w:GD40291[w:TMP0001]] == [w:TMP0002])	//Judge ID
					{
						set([b:GB30009]);//ID Match Flag
						break;
					}else{
						[w:TMP0001] = [w:TMP0001] + 1;
					}
				}

				//Combined Output Side Case
				if([b:GB30009] == OFF)
				{
					[w:TMP0001] = 0;
					while([w:TMP0001] < [w:GD40290]) //Repeat the Number of Connection ID Counts
					{
						if([w:GD40291[w:TMP0001]] == ([w:TMP0002] + 0x0200))//Judge ID
						{

							//Judge whether the matched ID is that of a combined module or not.
							[w:TMP0010] = [w:GD41060[w:TMP0002]];	//Parameter Top Address
							[w:TMP0011] = [w:TMP0010] - 0x3000;

							[w:TMP0012] = [w:GD41359[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;
                                                }else{
                                                        [w:TMP0001] = [w:TMP0001] + 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:GD65535] = [w:TMP0002];
                        [w:TMP0001] = 0;
                        while([w:TMP0001] < [w:GD40290])//Repeat the Number of Connection ID
Counts
                        {
                                if([w:GD40291[w:TMP0001]]    ==    ([w:TMP0002]    +    0x0200))
        //Judge ID
                                {
                                        set([b:GB30009]);//ID Match Flag
                                        break;
                                }else{
                                        [w:TMP0001] = [w:TMP0001] + 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:GD31522] < 256)
                {
                        if([b:GB30020] == OFF)
```

```
		{
			//Output
			[w:GD31567] = [w:GD40804[w:TMP0002]];	//Offset
			[w:TMP0003] = [w:TMP0002];
			[w:TMP0004] = [w:TMP0002];
		}else{
			//If combined, refer to the input side.
			[w:GD31567] = [w:GD41060[w:TMP0002]];	//Offset
			[w:TMP0003] = [w:TMP0002] + 0x0200;
			[w:TMP0004] = [w:TMP0002] + 0x0100;
		}
	}else{
		//Input
		[w:GD31567] = [w:GD41060[w:TMP0002]];	//Offset
		[w:TMP0003] = [w:TMP0002] + 0x0200;
		[w:TMP0004] = [w:TMP0002] + 0x0100;
	}

	[w:U03-G10320] = 0;		//Read Individual Parameter
	[w:U03-G10321] = [w:TMP0003];	//Read Target ID

	//Select base screen to display.
	switch([w:GD35130[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;

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

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 0x2900:    [w:GD62000] = 30018;
                                                    break;

                        case 0xA000:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA001:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA300:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA301:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA600:    [w:GD62000] = 30019;
                                                    break;

                        case 0xA601:    [w:GD62000] = 30019;
                                                    break;

                        default:                    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:GD31562.b0] == OFF) && ([b:GB30001] == OFF))
        {
                //Save Current Cursor Position
                [w:TMP0122] = [w:GD31522];

                //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:GD31520] = 16 - ([w:TMP0124] + 1);

                    //Y Coordinate
                    [w:TMP0125] = [w:TMP0121] - Input_Y;
                    [w:TMP0126] = [w:TMP0125] / Object_Y;
                    [w:GD31521] = [w:TMP0126] * 16;

                    //Offset
                    [w:GD31522] = [w:GD31520] + [w:GD31521];

                    //If the offset position does not change, displayed parts do not change.
                    if([w:TMP0122] != [w:GD31522])
                    {
                              //Cursor Display
                              if([w:GD31000[w:GD31522]] < 4)
                              {
                                        [w:GD31000[w:GD31522]] = [w:GD31000[w:GD31522]] + 4;
                              }

                              //Cursor Delete
                              if([w:GD31000[w:TMP0122]] < 4)
                              {
                                        [w:GD31000[w:TMP0122]] = 0;
                              }else{
                                        [w:GD31000[w:TMP0122]] = [w:GD31000[w:TMP0122]] - 4;
                              }
                    }
          }

          [w:GD32008] = [w:GD31522];          //Cursor Position Display
}
```

*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

```
//Read Statuses or Parameters
if([b:GB30007] == OFF)
{
	if(([w:U03-G8192] != 0) || ([w:U03-G9984] != 0))
	{
		set([b:GD31568.b0]);
	}else{
		set([b:GD31562.b0]);
	}
	set([b:GB30007]);
}
```

Script No.	30016	Script name	Script30016
Comment	Script No.30015 Start		
Data type	Signed BIN16	Trigger type	OFF, GD31563.b0

```
//After reading the status, start Script No.30015.
if([b:GB30016] == OFF)
{
	set([b:GB30008]);
	set([b:GB30016]);
}
```

Script No.	30015	Script name	Script30015
Comment	B-30005 Model and Others		
Data type	Signed BIN16	Trigger type	ON, GB30008

```
//Display Objects
[w:GD31526] = 0;
[w:TMP0004] = 0;
[w:TMP0006] = 1;
[w:TMP0100] = [w:GD31529];

while([w:GD31526] < 15)
{
	//Check Connection Status
	if(([w:GD40290] != 0) && ([w:TMP0100] < [w:GD40290]))
	{
		[w:GD31600] = [w:GD40291[w:TMP0100]];	//Connection ID
		[w:TMP0100] = [w:TMP0100] + 1;

		if(([w:GD31600] & 0x0200) == 0)
		{
			set([b:GB30600]);//Output Bit
			[w:GD31601] = ([w:GD31600] & 0x0200) >> 9;
		}else{
			set([b:GB30601]);//Input Bit
			[w:GD31601] = ([w:GD31600] & 0x0200) >> 9;
		}

		//Offset
		[w:TMP0000] = [w:GD31600] & 0x00FF;

		if(([w:GD31600] & 0x0200) == 0)
		{
			//Output
			[w:TMP0001] = [w:GD40804[w:TMP0000]];
		}else{
			//Input
```

```
                        [w:TMP0001] = [w:GD41060[w:TMP0000]];
                }
                [w:GD31604] = [w:TMP0001] - 12288;

                //Module Type
                [w:TMP0002] = [w:GD31604] + 43;
                [w:GD31617[w:GD31526]] = ([w:GD41316[w:TMP0002]] & 0x00C0) >> 6;

                //Module Model No.
                [w:TMP0005] = [w:GD31604] + 44;
                [w:GD35096[w:GD31526]] = [w:GD41316[w:TMP0005]];

                //Alarm Information
                if([w:GD40547] != 0)
                {
                        [w:TMP0004] = 0;
                        while([w:TMP0004] < [w:GD40547])
                        {
                                if([w:GD40548[w:TMP0004]] == [w:GD31600])
                                {
                                        [w:GD31602] = 1; //Alarm Occurrence
                                        if(([w:U03-G10256] != 304) && (([w:U03-G10256] < 200)
|| ([w:U03-G10256] > 202)))
                                        {
                                                [w:GD31633[w:GD31526]] = [w:U03-G10256];
                                        }else{
                                                [w:TMP0003] = ([w:TMP0001] - 0x3000) + 39;
                                                [w:GD31633[w:GD31526]]                          =
[w:GD41316[w:TMP0003]];
                                        }
                                        break;
                                }
                                [w:TMP0004] = [w:TMP0004] + 1;
                        }
                }

                //Error Information
                if([w:GD40032] != 0)
                {
                        [w:TMP0004] = 0;
                        while([w:TMP0004] < [w:GD40032])
                        {
                                if([w:GD40033[w:TMP0004]] == [w:GD31600])
                                {
                                        [w:GD31603] = 1; //Error Occurrence
                                        if(([w:U03-G10256] == 304) || (([w:U03-G10256] >= 200)
&& ([w:U03-G10256] <= 202)))
                                        {
                                                [w:GD31633[w:GD31526]] = [w:U03-G10256];
                                        }else{
                                                [w:TMP0003] = ([w:TMP0001] - 0x3000) + 39;
                                                [w:GD31633[w:GD31526]]                          =
[w:GD41316[w:TMP0003]];
                                        }
                                        break;
                                }
                                [w:TMP0004] = [w:TMP0004] + 1;
                        }
                }

                //Display
```

```
		if([w:GD31603] == 1)
		{
			[w:GD31665[w:GD31526]] = 3;	//Error Occurrence
			[w:GD35032[w:GD31526]] = [w:GD31601];	//I/O
			[w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;	//ID
			[w:GD31649[w:GD31526]] = [w:GD31600];
			[w:GD35643] = [w:GD35643] | ([w:TMP0006] << [w:GD31526]);
	//Display Conditions

		}else{
			if([w:GD31602] == 1)
			{
				[w:GD31665[w:GD31526]] = 2;	//Alarm Occurrence
				[w:GD35032[w:GD31526]] = [w:GD31601];	//I/O
				[w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;	//ID
				[w:GD31649[w:GD31526]] = [w:GD31600];
				[w:GD35643] = [w:GD35643] | ([w:TMP0006] << [w:GD31526]);
	//Display Conditions
			}else{
				[w:GD31665[w:GD31526]] = 1;	//Normal
				[w:GD35032[w:GD31526]] = [w:GD31601];	//I/O
				[w:GD35064[w:GD31526]] = [w:GD31600] & 0x00FF;	//ID
				[w:GD31649[w:GD31526]] = [w:GD31600];
				[w:GD31633[w:GD31526]] = 0;	//ステータス
				[w:GD35643] = [w:GD35643] | ([w:TMP0006] << [w:GD31526]);
	//Display Conditions
			}
		}

		[w:GD31526] = [w:GD31526] + 1;
		rst([b:GB30600]); //Output Bit
		rst([b:GB30601]); //Input Bit

		//Work Area Clear
		[w:TMP0004] = 0;
		fmov([w:TMP0004],[w:GD31600],5);

	}else{
		//If data does not exist, do not display.
		[w:GD31665[w:GD31526]] = 0xFFFF;
		[w:GD35032[w:GD31526]] = 0xFFFF;
		[w:GD35064[w:GD31526]] = 0xFFFF;
		[w:GD35096[w:GD31526]] = 0xFFFF;
		[w:GD31617[w:GD31526]] = 0xFFFF;
		[w:GD31649[w:GD31526]] = 0x7FFE;
		[w:GD31633[w:GD31526]] = 24219;
		[w:GD35643] = [w:GD35643] ^ ([w:TMP0006] << [w:GD31526]);	//Display
Conditions
		[w:GD31526] = [w:GD31526] + 1;
		rst([b:GB30600]); //Output Bit
		rst([b:GB30601]); //Input Bit
	}
}

//Trigger Reset
rst([b:GB30008]);
```

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]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD31000],512);          //Initialize Object Values(B-30003, 4)
fmov([w:TMP0004],[w:GD31520],3);            //Clear Cursor Position Information(B-30003,4)
fmov([w:TMP0004],[w:GD31600],8);            //Work Area Clear
[w:GD32008] = 0; //Cursor Position Information Clear(B-30003,4)

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

Script No.	30024	Script name	Script30024
Comment	Previous Page		
Data type	Signed BIN16	Trigger type	Rise, GB30014

```
//Scroll List Up
if([w:GD31529] > 0)
{
        [w:GD31529] = [w:GD31529] - 15;
        set([b:GB30008]);
}
```

Script No.	30025	Script name	Script30025
Comment	Next Page		
Data type	Signed BIN16	Trigger type	Rise, GB30015

```
//Scroll List Down
if((([w:GD31529] + 15) < [w:U03-G8960]) && ([w:U03-G8960] > 15))
{
        [w:GD31529] = [w:GD31529] + 15;
        set([b:GB30008]);
}
```

Script No.	30034	Script name	Script30034
Comment	B-30005 Detail Screen Display		
Data type	Unsigned BIN16	Trigger type	ON, GB30018

```
//Determine detail screen to display.
[w:TMP0000] = [w:GD31523] & 0x00FF;

if((([w:GD31523] & 0x0200) >> 8) == 0)
{
        //Output
        [w:GD31567] = [w:GD40804[w:TMP0000]];   //Offset
}else{
        //Input
        [w:GD31567] = [w:GD41060[w:TMP0000]];   //Offset
}

[w:U03-G10320] = 0;         //Read Individual Parameter
[w:U03-G10321] = [w:GD31523];     //Read Target ID

//Select base screen to display.
switch([w:GD35096[w:GD31528]])
{
        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;

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

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 0x2900:    [w:GD62000] = 30018;
                                    break;

case 0xA000:    [w:GD62000] = 30019;
                                    break;

case 0xA001:    [w:GD62000] = 30019;
                                    break;

case 0xA300:    [w:GD62000] = 30019;
                                    break;

case 0xA301:    [w:GD62000] = 30019;
                                    break;

case 0xA600:    [w:GD62000] = 30019;
                                    break;

case 0xA601:    [w:GD62000] = 30019;
                                    break;
```

```
        default:                    break;
}

rst([b:GB30018]);
```

Base screen 30006 to 30019

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:GD40032] != 0)
        {
                [w:TMP0000] = 0;
                while([w:TMP0000] < [w:GD40032])
                {
                        if([w:GD40033[w:TMP0000]] == [w:U03-G10321])
                        {
                                set([b:GB30028]);//Error Occurrence Flag
                                break;
                        }
                        [w:TMP0000] = [w:TMP0000] + 1;
                }
        }

        if([b:GB30028] == ON)
        {
                //If errors occurred, read the most recently read settings.
                set([b:GD31564.b0]);
        }else{
                //If errors are not occurred, write the latest information.
                set([b:Y40]);
        }
        [w:GD31681] = 1; //Specify Start Line of Processing Method
        set([b:GB30017]);
}
```

Script No.	30014	Script name	Script30014
Comment	Parameter Access Error		
Data type	Signed BIN16	Trigger type	ON, Y40

```
//Processing after Reading Parameters Individually
if([w:U03-G10320] == 0)
{
        set([b:GB30024]);//Read Flag
}else{
        set([b:GB30025]);//Write Flag
}

if([b:X42] == ON)
{
                [w:GD62007] = 30004;
                rst([b:GB30024]);
                rst([b:GB30025]);
                rst([b:Y40]);
}else{
        if([b:X41] == ON)
        {
                rst([b:GB30024]);
                rst([b:GB30025]);
```

```
                set([b:GB30026]);
                rst([b:Y40]);
        }
}
```

Script No.	30026	Script name	Script30026
Comment	Module Model No. Correct		
Data type	Signed BIN16	Trigger type	OFF, GD31565.b0

```
if([b:GB30021] == OFF)
{
        //Convert Module Model Number Into Decimal Number
        [w:TMP0000] = [w:GD53629] & 0x000F;       //Ones Digit
        [w:TMP0001] = (([w:GD53629] & 0x00F0) >> 4) * 10;  //Tenths Digit
        [w:TMP0002] = (([w:GD53629] & 0x0F00) >> 8) * 100;//Hundreds Digit
        [w:TMP0003] = (([w:GD53629] & 0xF000) >> 12) * 1000;         //Thousands Digit

        [w:GD31607] = [w:TMP0000] + [w:TMP0001] + [w:TMP0002] + [w:TMP0003];

        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

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

rst([b:GB30016]);
rst([b:GB30017]);
rst([b:GB30021]);

rst([b:GB30028]);

[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD31000],512);        //Initialize Object Values(B-30002)
fmov([w:TMP0004],[w:GD31600],17);         //Work Area Clear

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

Script No.	30038	Script name	Script30038
Comment	Module Detail Info Display		
Data type	Signed BIN16	Trigger type	ON, GB30019

```
//Display Detail Information

//Work Area Clear
[w:TMP0004] = 0;
fmov([w:TMP0004],[w:GD31600],7);

//Alarm Information
if([w:GD40547] != 0)
{
        [w:TMP0004] = 0;
        while([w:TMP0004] < [w:GD40547])
        {
                if([w:GD40548[w:TMP0004]] == [w:GD53604])
                {
                        [w:GD31602] = 1; //Alarm Occurrence
```

```
                        if(([w:U03-G10256] != 305) && (([w:U03-G10256] != 304) &&
(([w:U03-G10256] < 200) || ([w:U03-G10256] > 202))))
                        {
                                [w:GD31608] = [w:U03-G10256];
                        }else{
                                [w:GD31608] = [w:GD53624];
                        }
                        break;
                }
                [w:TMP0004] = [w:TMP0004] + 1;
        }
}

//Error Information
if([w:GD40032] != 0)
{
        [w:TMP0004] = 0;
        while([w:TMP0004] < [w:GD40032])
        {
                if([w:GD40033[w:TMP0004]] == [w:GD53604])
                {
                        [w:GD31603] = 1; //Error Occurrence
                        if(([w:U03-G10256] != 305) && (([w:U03-G10256] == 304) ||
(([w:U03-G10256] >= 200) && ([w:U03-G10256] <= 202))))
                        {
                                [w:GD31608] = [w:U03-G10256];
                        }else{
                                [w:GD31608] = [w:GD53624];
                        }
                        break;
                }
                [w:TMP0004] = [w:TMP0004] + 1;
        }
}

//Device Parameter
//I/O Points Pattern
[w:GD31605] = ([w:GD53628] & 0x00C0) >> 6;          //0: Input, 1: Output, 2: Combined
[w:GD31606] = ([w:GD53628] & 0x003F) + 1;           //Points

//ON/OFF Information
[w:TMP0008] = 0;
[w:TMP0000] = [w:GD53604] & 0x00FF;
if([w:GD31605] == 0)        //Input Case
{
        [w:TMP0004] = 0;

        while([w:TMP0004] < [w:GD31606])
        {
                [w:TMP0009] = [w:TMP0000] / 16;
                [w:TMP0010] = [w:TMP0000] % 16;

                [w:GD31682[w:TMP0008]] = [w:TMP0009];
                [w:TMP0012] = 0x0001;
                [w:GD31990[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

                if([w:GD31603] == 1)        //Check If Error Occurred
                {
                        [w:GD31609[w:TMP0008]] = 3;       //Error Occurrence
                }else{
                        if([w:GD31602] == 1)        //Check If Alarms Occurred
```

```
                        {
                                [w:GD31609[w:TMP0008]] = 2;       //Alarm Occurrence
                        }else{
                                [w:GD31609[w:TMP0008]] = 1;       //Normal
                                [w:GD31608] = 0;
                        }
                }

                [w:TMP0000] = [w:TMP0000] + 1;
                [w:TMP0008] = [w:TMP0008] + 1;
                [w:TMP0004] = [w:TMP0004] + 1;
        }

}else{
        if([w:GD31605] == 1)        //Output Case
        {
                [w:TMP0004] = 0;

                while([w:TMP0004] < [w:GD31606])
                {
                        [w:TMP0009] = [w:TMP0000] / 16;
                        [w:TMP0010] = [w:TMP0000] % 16;

                        [w:GD31682[w:TMP0008]] = [w:TMP0009] + 4096;
                        [w:TMP0012] = 0x0001;
                        [w:GD31990[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

                        if([w:GD31603] == 1)        //Check If Error Occurred
                        {
                                [w:GD31609[w:TMP0008]] = 3;       //Error Occurrence
                        }else{
                                if([w:GD31602] == 1)        //Check If Alarms Occurred
                                {
                                        [w:GD31609[w:TMP0008]] = 2;       //Alarm Occurrence
                                }else{
                                        [w:GD31609[w:TMP0008]] = 1;       //Normal
                                        [w:GD31608] = 0;
                                }
                        }

                        [w:TMP0000] = [w:TMP0000] + 1;
                        [w:TMP0008] = [w:TMP0008] + 1;
                        [w:TMP0004] = [w:TMP0004] + 1;
                }

        }else{
                if([w:GD31605] == 2)        //Combined Case
                {
                        [w:TMP0004] = 0;
                        [w:TMP0011] = [w:GD31606] / 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:GD31682[w:TMP0008]] = [w:TMP0009] + 4096;
                                [w:TMP0012] = 0x0001;
                                [w:GD31990[w:TMP0008]] = [w:TMP0012] << [w:TMP010];
```

```
                                    if([w:GD31603] == 1)        //Check If Error Occurred
                                    {
                                            [w:GD31609[w:TMP0008]] = 3;       //Error Occurrence
                                    }else{
                                            if([w:GD31602] == 1)        //Check If Error Occurred
                                            {
                                                    [w:GD31609[w:TMP0008]] = 2;       //Alarm
Occurrence
                                            }else{
                                                    [w:GD31609[w:TMP0008]] = 1;       //Normal
                                                    [w:GD31608] = 0;
                                            }
                                    }

                                    //Input
                                    [w:GD31683[w:TMP0008]] = [w:TMP0009];
                                    [w:TMP0012] = 0x0001;
                                    [w:GD31991[w:TMP0008]] = [w:TMP0012] << [w:TMP010];

                                    if([w:GD31603] == 1)        //Check If Error Occurred
                                    {
                                            [w:GD31610[w:TMP0008]] = 3;       //Error Occurrence
                                    }else{
                                            if([w:GD31602] == 1)        //Check If Alarms Occurred
                                            {
                                                    [w:GD31610[w:TMP0008]] = 2;       //Alarm
Occurrence
                                            }else{
                                                    [w:GD31610[w:TMP0008]] = 1;       //Normal
                                                    [w:GD31608] = 0;
                                            }
                                    }

                                    [w:TMP0000] = [w:TMP0000] + 1;
                                    [w:TMP0008] = [w:TMP0008] + 1;
                                    [w:TMP0004] = [w:TMP0004] + 1;
                            }
                    }
            }
}

rst([b:GB30019]);
```

Script No.	30039	Script name	Script30039
Comment	Parameter Re-read		
Data type	Signed BIN16	Trigger type	OFF, GD31563.b0

```
//Re-read Parameters
if([b:GB30016] == OFF)
{
        set([b:GD31564.b0]);
        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([w:U03-G10320] == 0)
{
        set([b:GD31564.b0]);
}else{
        [w:U03-G10320] = 0;
```

```
        set([b:Y40]);
}

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:GD40032] != 0)
{
        [w:TMP0000] = 0;
        while([w:TMP0000] < [w:GD40032])
        {
                if([w:GD40033[w:TMP0000]] == [w:U03-G10321])
                {
                        set([b:GB30028]);//Error Occurrence Flag
                        break;
                }
                [w:TMP0000] = [w:TMP0000] + 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:U03-G10320] = 0;
        [w:U03-G10321] = [w:GD53604];
        set([b:Y0040]);
}

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:GD40032] != 0)
{
        [w:TMP0000] = 0;
        while([w:TMP0000] < [w:GD40032])
        {
                if([w:GD40033[w:TMP0000]] == [w:U03-G10321])
                {
                        set([b:GB30028]);//Error Occurrence Flag
                        break;
                }
                [w:TMP0000] = [w:TMP0000] + 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:U03-G10320] = 1;
        [w:U03-G10321] = [w:GD53604];
        set([b:GD31570.b0]);
}

rst([b:GB30030]);
```

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
	[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.	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.	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.	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] = 280;
                        }else{
                                [w:GD54006] = 268;
                        }
                }else{
                        //CC-Link
                        if([w:GD54088] == 100)
                        {
                                [w:GD54006] = 281;
                        }else{
```

```
                                        [w:GD54006] = 268;
                                }
                        }

                        [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] = 226;		//ID or Station
			[w:GD54036] = 232;		//All IDs or All Stations
			[w:GD54037] = 233;		//ID Specification or Station Specification
			[w:GD54038] = 285;		//Not Specified or Station Sub
			[w:GD54039] = 0; //Not Specified or Station Sub Specified
			break;

	//CC-Link
	case 2:	[w:GD54035] = 225;//ID or Station
			[w:GD54036] = 242;//All IDs or All Stations
			[w:GD54037] = 243;		//ID Specification or Station Specification
			[w:GD54038] = 227;		//Not Specified or Station Sub
			[w:GD54039] = 244;		//Not Specified or Station Sub Specified
			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;
}
```

Script No.	30053	Script name	Script30053
Comment	B-30102 I/O Judgment		
Data type	Signed BIN16	Trigger type	Rise/Fall, GD54086.b2

```
//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] & 0x00FF;		//Output
}
```

Base screen 30103

Script No.	30055	Script name	Script30055
Comment	B-30103,30106 Scr.Start Process		
Data type	Signed BIN16	Trigger type	ON, GB40

```
//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]);
}
```

Script No.	30060	Script name	Script30060
Comment	B-30103 Normal End		
Data type	Signed BIN16	Trigger type	Rise, SM1437

```
///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) && (([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
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.	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)
{
		//Do no restore data from the backup that was not ended 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] = 226;          //ID or Station
                              [w:GD54071] = 232;          //All IDs or All Stations
                              [w:GD54072] = 233;          //ID Specification or Station Specification
                              [w:GD54073] = 285;          //Not Specified or Station Sub
                              [w:GD54074] = 0; //Not Specified or Station Sub Specified
                              break;
```

```
        //CC-Link
        case 2:  [w:GD54070] = 225;         //ID or Station
                        [w:GD54071] = 242;          //All IDs or All Stations
                        [w:GD54072] = 243;          //ID Specification or Station Specification
                        [w:GD54073] = 227;          //Not Specified or Station Sub
                        [w:GD54074] = 244;          //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

```
//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]);
}
```

Script No.	30044	Script name	Script30044
Comment	B-30106 Normal End		
Data type	Signed BIN16	Trigger type	Rise, SM1440

```
//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
```

Script No.	30045	Script name	Script30045
Comment	B-30106 Abnormal End		
Data type	Signed BIN16	Trigger type	ON, SM1441

```
//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]; //Restoration Failure Message
                        [w:GD62007] = 30101;//Notification Dialog
                }
        }else{
                if([w:SD1453] != 0)
                {
                        [w:GD54006] = [w:SD1453]; //Restoration Failure Message
                        [w:GD62007] = 30101;//Notification Dialog
                }else{
```

```
                    [w:GD54006] = 32004; //Restoration Failure Message
                    [w:GD62007] = 30101;//Notification Dialog
                }
        }

        set([b:GB54009]);
}
```

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]);
```

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;
```

5.9.3 Object script

Base screen 30006 to 30019

Object	Word lamp *1		
Script user ID	1		
Data type	Unsigned BIN16	Trigger type	Ordinary

```
//Display ON/OFF Status
[w:GD32000] = $$ & [w:GD31990];
```

*1: Set for the word lamps that indicate the ON/OFF of I/O. The device number changes depending on the number of I/O points.

Base screen 30013, Base screen 30018

Object	Level *1		
Script user ID	2		
Data type	Signed BIN16	Trigger type	Ordinary

```
//Display ON/OFF Threshold Value
screen_draw(0);
redraw_object();

[w:TMP0200] = (object_width / 100) * [w:GD53605];
[w:TMP0201] = (object_width / 100) * [w:GD53606];

d_line([w:TMP0200],0,[w:TMP0200],object_height,0,1,224);      //OFF to ON Threshold Value
d_line([w:TMP0201],0,[w:TMP0201],object_height,0,1,3);                  //ON to OFF 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 30010 to 30012, Base screen 30014, Base screen 30015, Base screen 30017

Object	Level *1		
Script user ID	2		
Data type	Signed BIN16	Trigger type	Ordinary

```
//Display Threshold Value, Alarm Judgment(Hi), (Lo)
screen_draw(0);
redraw_object();

[w:TMP0200] = (object_width / 100) * [w:GD53605];
[w:TMP0201] = (object_width / 100) * [w:GD53607];
[w:TMP0202] = (object_width / 100) * [w:GD53608];

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 30018

Object	Numerical input *1		
Script user ID	3		
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 30101

Object	Word lamp		
Script user ID	1		
Data type	Signed BIN16	Trigger type	Ordinary

```
//Script to Display Comments
if([w:GD54101[w:GD54001]] != 0)
{
        [w:GD54076] = (([w:GD54101[w:GD54001]] * 10) + $$) + 221;
}else{
        [w:GD54076] = 0;
}
```

The above script is set for all word lamps of the “Execution unit”. The specified device varies depending on the object.

Window screen 30001

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 Hour 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

Sybol Name	Device and Constant	Remarks
Input_X	42	Upper-left-most X Coordinate of Input Display Area
Input_Y	155	Upper-left-most Y Coordinate of Input Display Area
Frame_X	618	Lower-right-most X Coordinate of I/O Display Area
Frame_Y	427	Lower-right-most Y Coordinate of I/O Display Area
Object_X	36	Word Lamp Width
Object_Y	17	Word Lamp Height

5.9.5 Object script symbol

Sybol Name	Device and Constant	Remarks
object_width	124	Level Width
object_height	25	Level Height

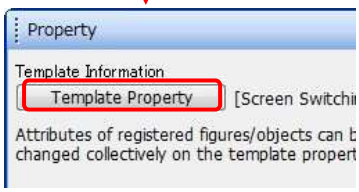
6. TEMPLATES

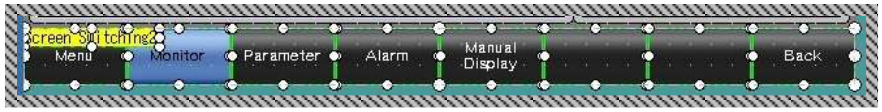
Templates are a group of figures and objects. Related settings are grouped into template attributes and registered, so the devices and colors, etc. can be easily changed in a batch. For more details about changing the attribute settings, please refer to the "GT Designer3 (GOT2000) Help".

The template information is only displayed on the screen design software's editing screen. It is not displayed on the GOT display screen.

Example: Changing a font

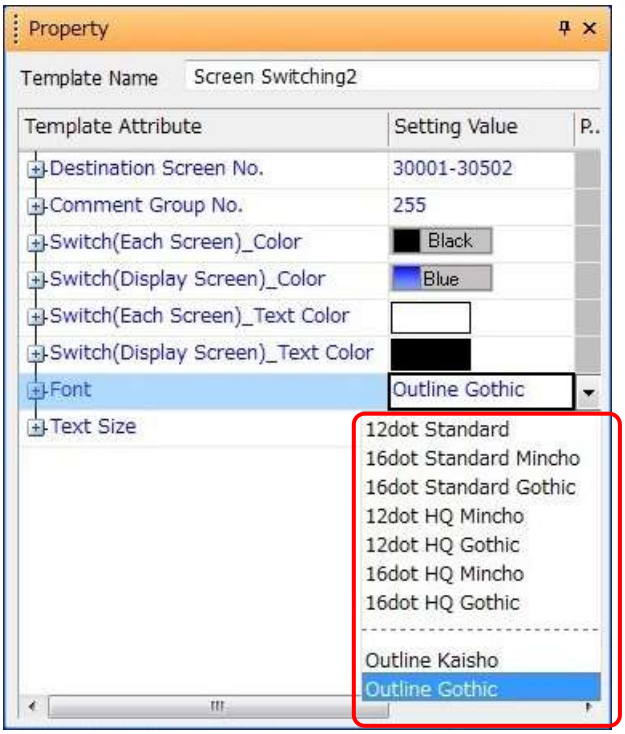
(1) Select [Template Information], and click on [Template Property] (or double-click [Template Information]).





The figures and objects that are registered in the template are changed to the selected state.

(2) Click on [Font], and select the new font.

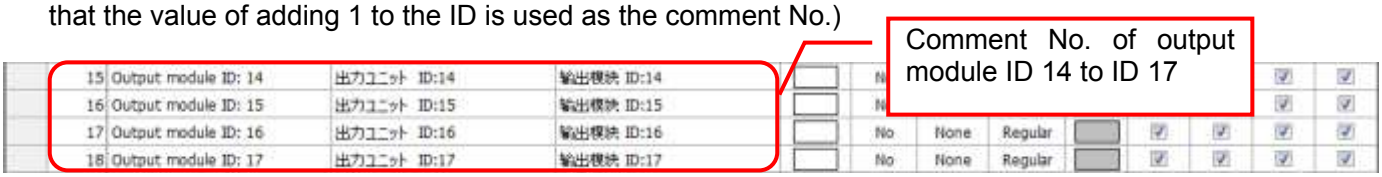


7. OTHERS

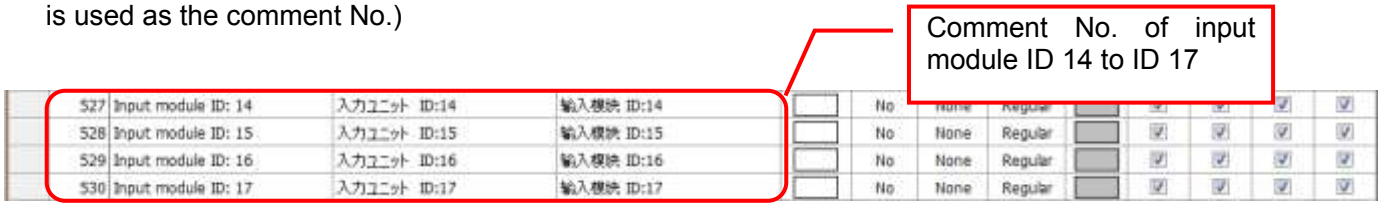
7.1 User-Defined Name Registration

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.)



- 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.)



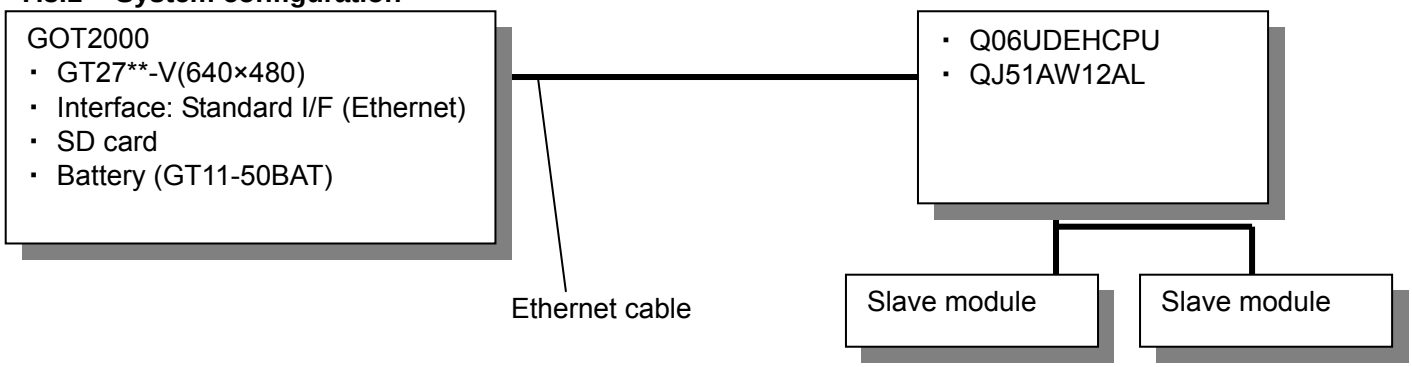
7.2 Changing System Configuration

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 [Sensor Auto Detect] switch for a second or reboot the GOT.

7.3 Changing CPU

To use the MELSEC-Q Series CPU, change the system configuration and the Ethernet setting.

7.3.1 System configuration



7.3.2 Ethernet Setting of Screen Design Software

	Host	Net No.	Station	Unit Type	IP Address	Port No.	Communication
1	*	1	1	QnUDE(H)	192.168.3.39	5006	UDP

7.4 Changing Start I/O Number

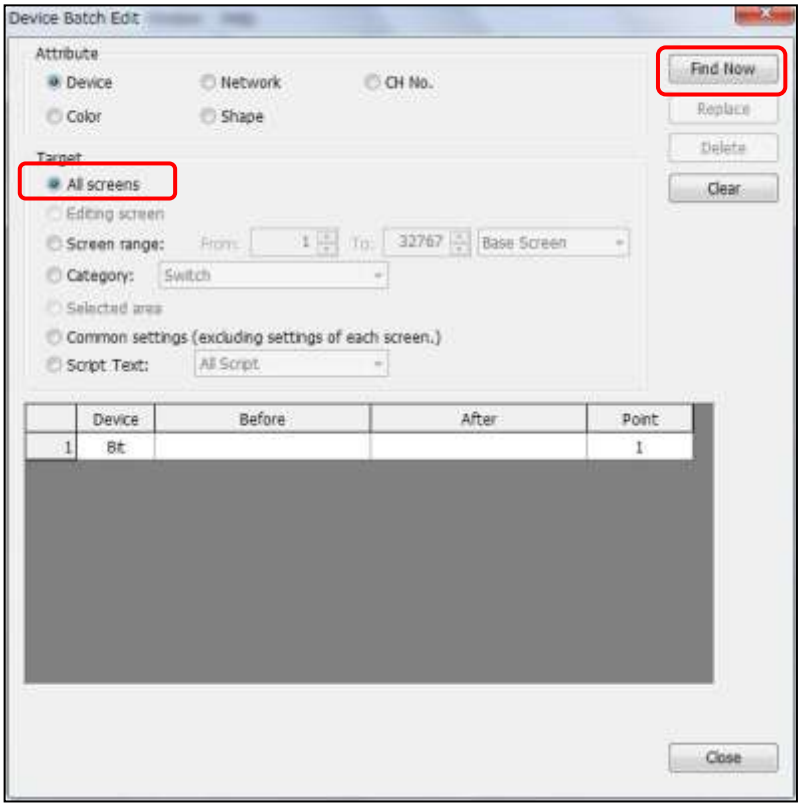
Follow the procedure below to change the start I/O number of the module to a value other than 30H.
(Example: Changing the start I/O number from 30H to 50H)

7.4.1 To change devices set in the screen

To change devices set in the screen, select [All Screens] for the target of the Device Batch Edit.
The devices to change by selecting [All Screens] are U03-G0 to U03-G10321, X0041, Y0030 to Y0041.

(1) Select [Search/Replace] - [Batch Edit] - [Device] menu.

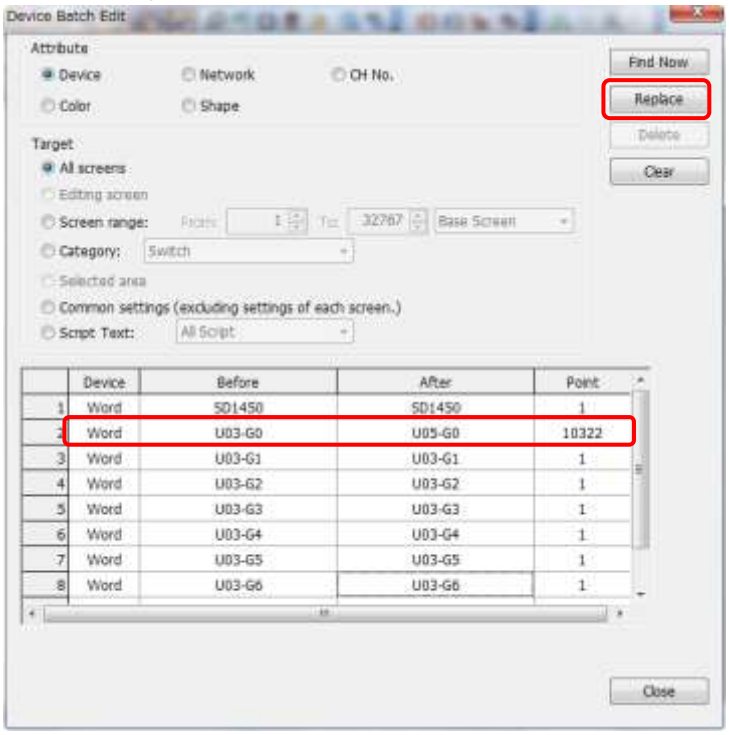
(2) In the displayed setting dialog, select [All Screens], and click [Find Now].



(3) Set the [After] device and [Point], and execute the batch edit.

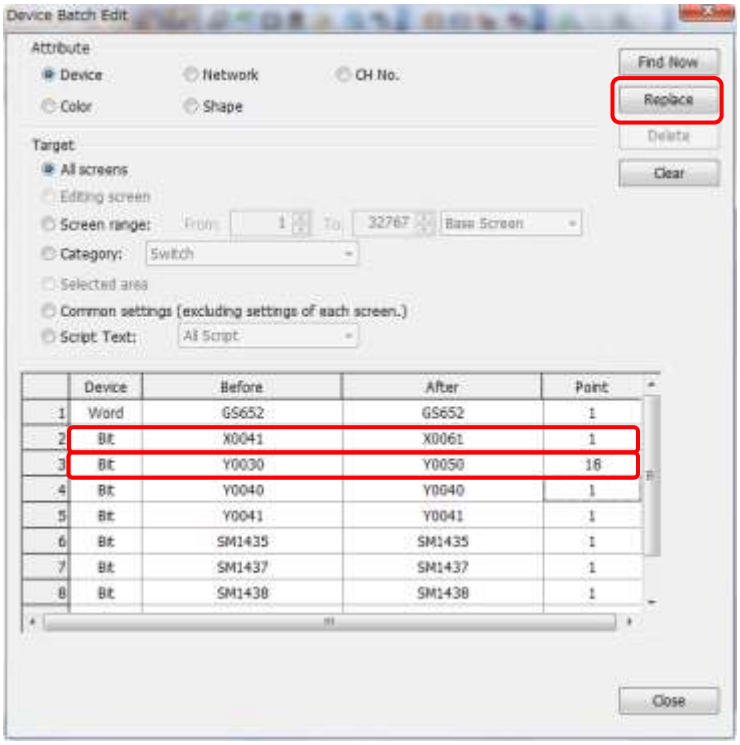
・Changing the start I/O number of the buffer memory

Set [Before] to U03-G0, [After] to U05-G0, and [Point] to 10322, and click [Replace]. U03-G0 to U03-G10321 will be changed to U05-G0 to U05-G10321.



・Changing the start I/O number of the I/O signal

To change the input signal (X device), set [Before] to X0041, [After] to X0061, and [Point] to 1, and click [Replace]. X0041 will be changed to X0061. To change the output signal (Y device), set [Before] to Y0030, [After] to Y0050, and [Point] to 18, and click [Replace]. Y0030 to Y0041 will be changed to Y0050 to Y0061.



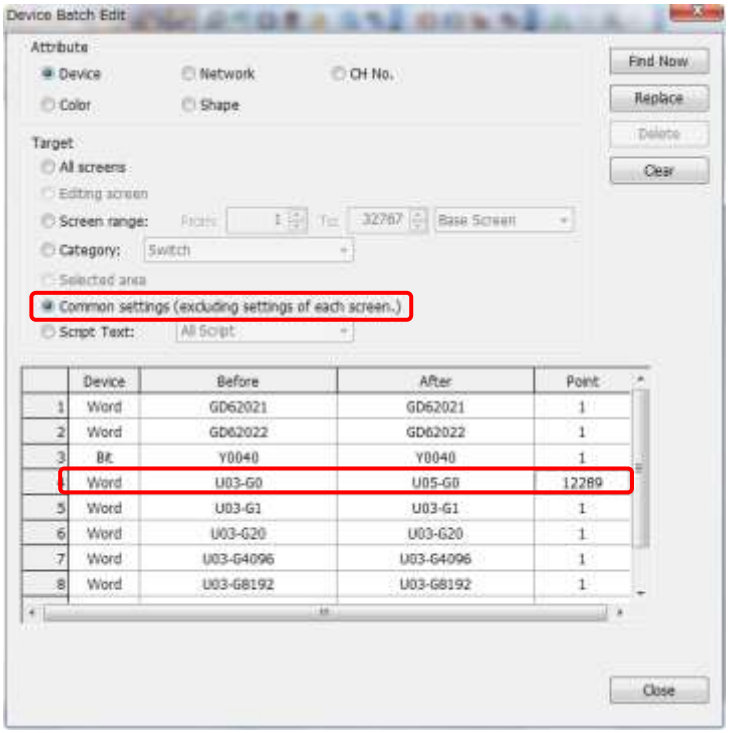
7.4.2 To change devices set in [Common]

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 7.4.1.

The devices to change by selecting [Common Settings] are U03-G0 to U03-G12288, Y0040.

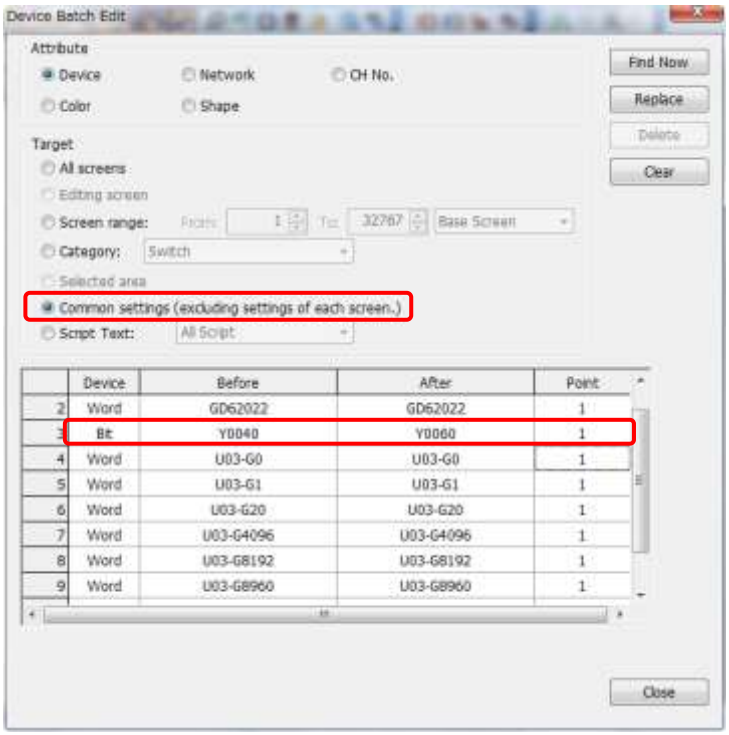
・Changing the start I/O number of the buffer memory

Set [Before] to U03-G0, [After] to U05-G0, and [Point] to 12289, and click [Replace]. U03-G0 to U03-G12288 will be changed to U05-G0 to U05-G12288.



・Changing the start I/O number of the I/O signal

Set [Before] to Y0040, [After] to Y0060, and [Point] to 1, and click [Replace]. Y0040 will be changed to Y0060.



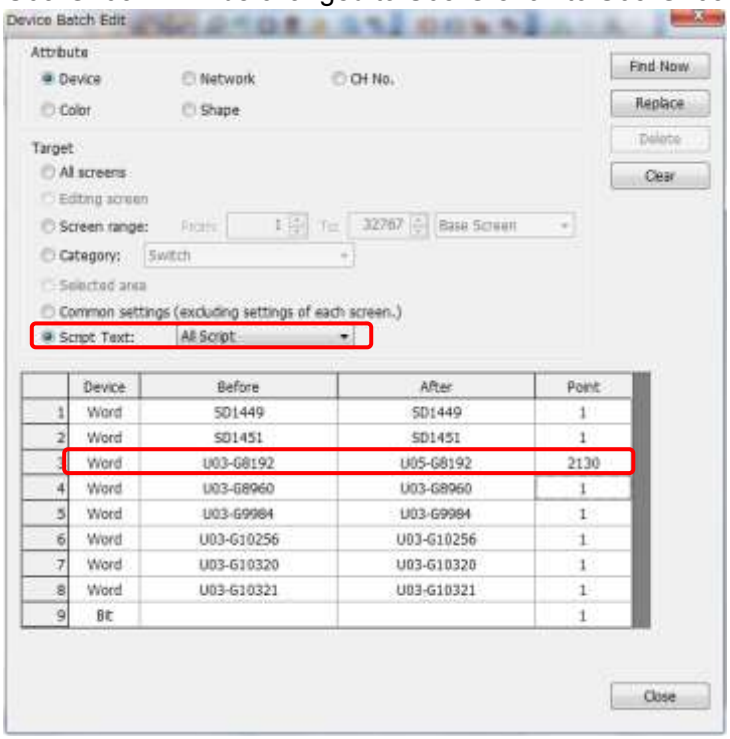
7.4.3 To change devices set in [Script Text]

To change devices set in [Script Text], select [Script Text] for the target of the Device Batch Edit. Follow the same procedures as in 7.4.1.

The devices to change by selecting [Script Text] are U03-G8192 to U03-G10321, X0041 to X0042, Y0040 to Y0041.

・Changing the start I/O number of the buffer memory

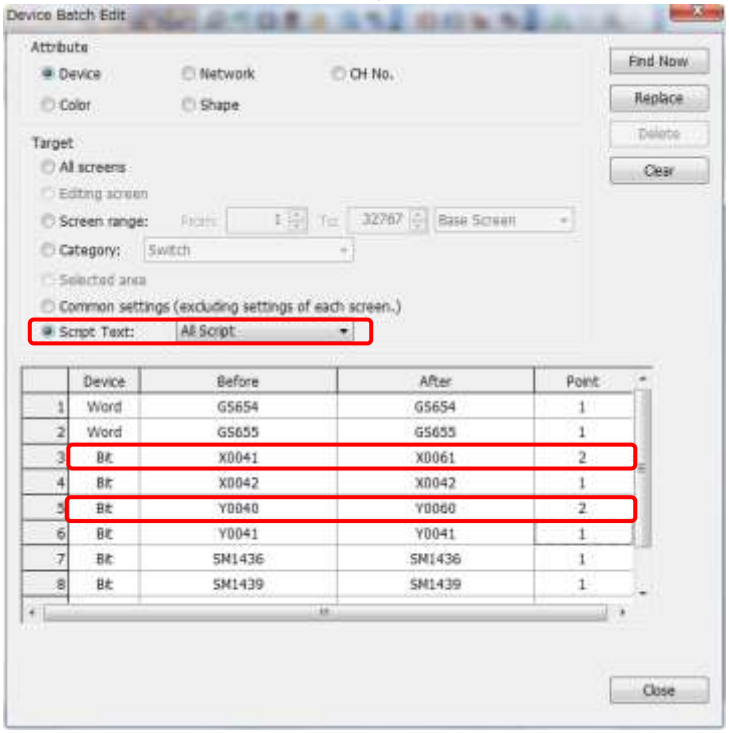
Set [Before] to U03-G8192, [After] to U05-G8192, and [Point] to 2130, and click [Replace]. U03-G8192 to U03-G10321 will be changed to U05-G 8192 to U05-G10321.



・Changing the start I/O number of the I/O signal

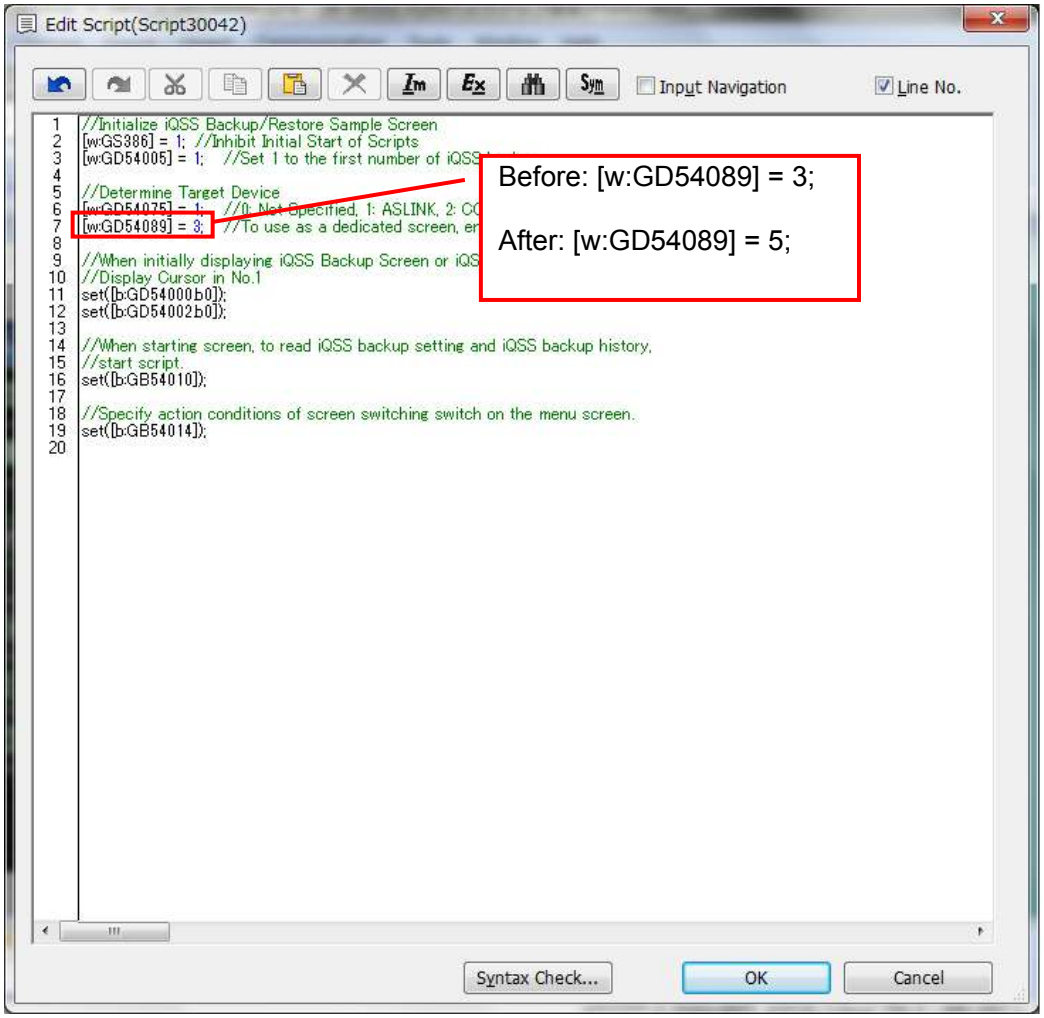
To change the input signal (X device), set [Before] to X0041, [After] to X0061, and [Point] to 2, and click [Replace]. X0041 to X0042 will be changed to X0061 to X0062.

To change the output signal (Y device), set [Before] to Y0040, [After] to Y0060, and [Point] to 2, and click [Replace]. Y0040 to Y0041 will be changed to Y0060 to Y0061.



7.4.4 To change project script

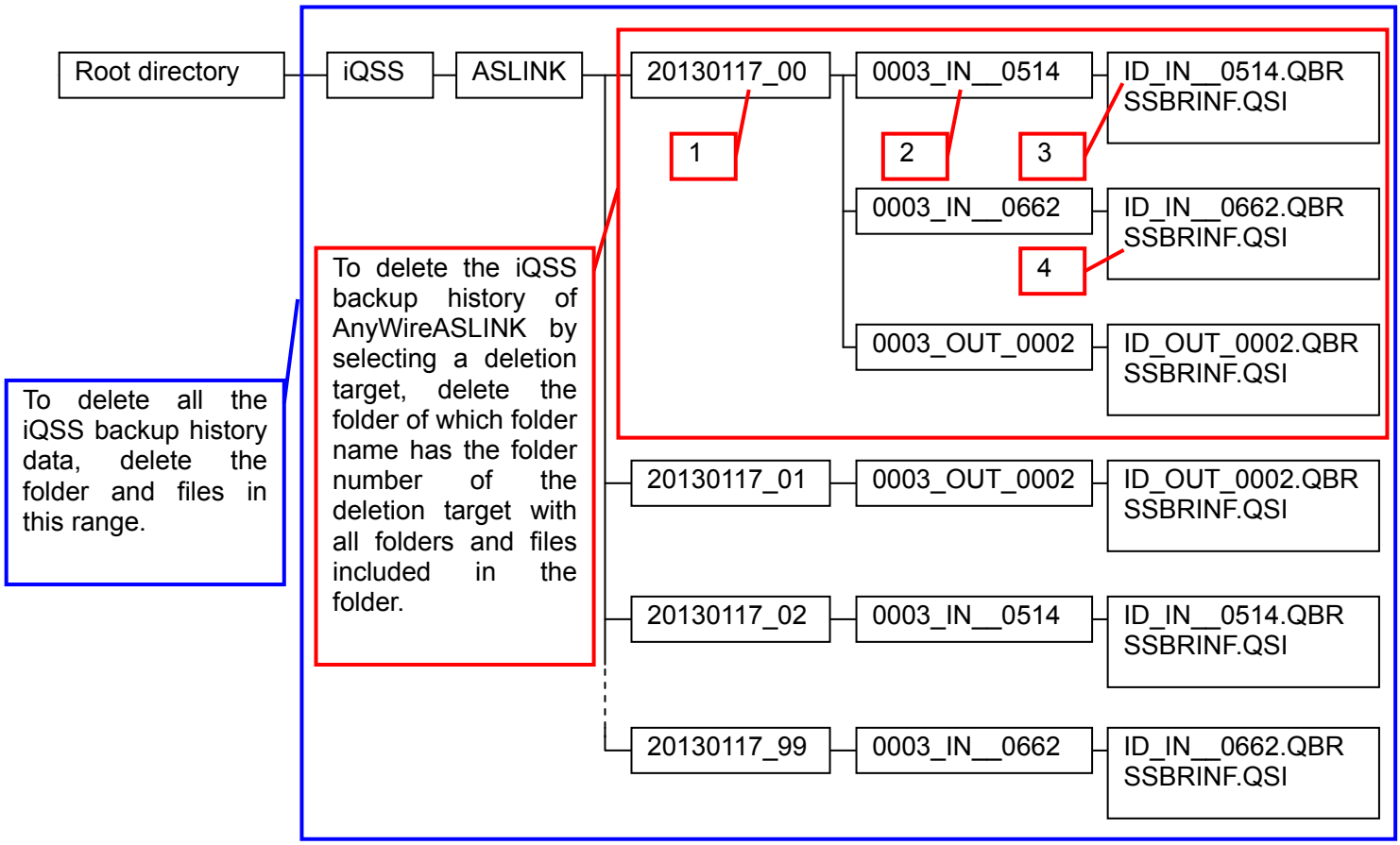
In the project script No.30042, the start I/O number is specified. Follow the procedure below to modify data.



7.5 iQSS Backup Folder Configuration

7.5.1 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 “7.5.2 Folder name, file name details”.



7.5.2 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.