

Mitsubishi
Energy Measuring Module EcoMonitorLight
(Model Name: EMU4-BD1-MB/EMU4-HD1-MB) and
Electronic Multimeter (Model Name: ME110SSR-MB)

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

| | | |
|--------|---|----|
| 1 | OUTLINE | 5 |
| 2 | SYSTEM CONFIGURATION | 5 |
| 3 | GOT | 5 |
| 3.1 | System Applications That Are Automatically Selected | 5 |
| 3.2 | Controller Setting of Screen Design Software | 5 |
| 3.3 | Overlap Window Setting of Screen Design Software..... | 6 |
| 4 | Measuring Terminal..... | 6 |
| 4.1 | Measuring Terminal Communication Settings | 6 |
| 4.2 | Parameter Settings for the measuring terminal | 6 |
| 5 | SCREEN SPECIFICATIONS | 8 |
| 5.1 | Display Language..... | 8 |
| 5.2 | Screen List/Transition..... | 8 |
| 5.2.1 | Screen list/transition (common) | 8 |
| 5.2.2 | Screen list/transition (individual) | 9 |
| 5.3 | Explanation of Screens..... | 10 |
| 5.3.1 | Menu (B-30001)..... | 10 |
| 5.3.2 | Current values (B-30002) | 11 |
| 5.3.3 | Electric energy graph (B-30009)..... | 12 |
| 5.3.4 | Average current gra (B-30010) | 14 |
| 5.3.5 | Parameter settings (B-30020)..... | 16 |
| 5.3.6 | Alarm history (B-30011) | 17 |
| 5.3.7 | Alarm Reset (W-30001) | 18 |
| 5.3.8 | Language Setting (W-30002)..... | 19 |
| 5.3.9 | Clock Setting (W-30003)..... | 20 |
| 5.3.10 | Confirmation dialog (W-30014)..... | 21 |
| 5.3.11 | Phase wire system (W-30015)..... | 22 |
| 5.3.12 | Type of current sensor (W-30016) | 23 |
| 5.3.13 | Current Multiplier (W-30017) | 24 |
| 5.3.14 | Voltage Multiplier (W-30018)..... | 25 |
| 5.3.15 | W · DW · Var Multiplier (W-30019) | 26 |
| 5.3.16 | Wh · Varh Multiplier (W-30020)..... | 27 |
| 5.3.17 | Valid/Invalid station setting (W-30031)..... | 28 |
| 5.3.18 | Station Number Selection (W-30030) | 29 |
| 5.4 | Device List..... | 30 |
| 5.5 | Comment List | 39 |
| 5.6 | Recipe List..... | 40 |
| 5.6.1 | Common Setting | 40 |
| 5.6.2 | Individual Setting | 40 |
| 5.7 | Script List..... | 47 |
| 5.7.1 | Project script..... | 47 |
| 5.7.2 | Screen script..... | 51 |
| 5.7.3 | Object script..... | 69 |
| 6 | About Startup System Alarm (Communication Time Out) | 88 |

REVISIONS

Sample Screen Manual

| Date | Control No.* | Description |
|--------|------------------|-----------------------|
| 2015/1 | BCN-P5999-0472 | First edition |
| 2015/3 | BCN-P5999-0472-A | Project data improved |
| | | |
| | | |
| | | |
| | | |
| | | |

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

Project Data

| Date | Project data | GT Designer3* | Description |
|--------|-------------------------------|---------------|--|
| 2015/1 | mitsubishi_EMU4_V_Ver1_E.GTX | 1.126G | First edition |
| 2015/3 | mitsubishi_EMU4_V_Ver1a_E.GTX | 1.126G | Add the detailed description of the screens. |
| | | | |
| | | | |
| | | | |
| | | | |

* 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 Mitsubishi measuring terminal, energy measuring module EcoMonitorLight: EMU4-BD1-MB/EMU4-HD1-MB (hereinafter abbreviated as model EMU4) or electronic multimeter: ME110SSR-MB (hereinafter abbreviated as model ME110) in MODBUS®/RTU(RS-485) communication. The sample screens can be used to monitor the current or power consumption.

2 SYSTEM CONFIGURATION

GOT2000

- GT27**-V(640×480)
- Interface: Standard I/F(RS-422/485)*1
- SD card *2
- Battery (GT11-50BAT type lithium battery)*3

RS-485 cable *4

Mitsubishi
Energy measuring module
EcoMonitorLight
(Model name:
EMU4-BD1-MB/EMU4-HD1-MB)
Interface: Output terminal *5
Or electronic multimeter
(Model name: ME110SSR-MB)
Interface: Output terminal *5

- *1: Terminal resistance is required. Refer to the "GOT2000 Series Connection Manual (Microcomputer, MODBUS Products, Peripherals)" for more details of terminal resistance.
- *2: The SD card is used for the project data, storage of OS, logging and recipe functions.
- *3: The battery is used for the logging function. (The battery is provided with the GOT as standard.)
- *4: For more details about the cable, please refer to the "GOT2000 Series Connection Manual (Microcomputer, MODBUS Products, Peripherals)".
- *5: Terminal resistance is required. Refer to the manual for each device for more details of terminal resistance.

3 GOT

3.1 System Applications That Are Automatically Selected

| OS type | OS name | | |
|----------------------|-----------------------------|---------------|----------------------|
| Standard Function | Standard System Application | | |
| | Standard Font | Standard Font | |
| Communication Driver | MODBUS®/RTU | | |
| Extended Function | Standard Font | | Chinese (Simplified) |
| | Outline Font | Gothic | Outline Font |
| | | | |
| | | | |
| | Document Display | | |

3.2 Controller Setting of Screen Design Software

| Item | Set value | Remarks |
|--------------------------------|-----------|---|
| Transmission Speed (BPS) | 38400 | |
| Data Bit | 8 bit | |
| Stop Bit | 1 bit | |
| Parity | Even | |
| Retry (Times) | 3 | |
| Timeout Time (Sec) | 3 | |
| Host Address | 1 | Set the MODBUS® of connecting measuring terminal. This item is fixed to 1 station in this sample. |
| Delay Time (ms) | 0 | |
| 32bit Storage | HL order | |
| FunctionCode[0F] | Used | |
| FunctionCode[10] | Used | |
| Coil read times(Points) | 2000 | |
| Input relay read times(Points) | 2000 | |

| Item | Set value | Remarks |
|--------------------------------------|-----------|---------|
| Holding register read times(Points) | 2 | |
| Input register read times(Points) | 125 | |
| Coil write times(Points) | 800 | |
| Holding register write times(Points) | 2 | |

3.3 Overlap Window Setting of Screen Design Software

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

4 Measuring Terminal

4.1 Measuring Terminal Communication Settings

(1) Energy measuring module (Model EMU4)

| Item | Set value | Remarks |
|----------------------------|-----------|---|
| MODBUS® Address | 1 | (*) |
| MODBUS® Transmission Speed | 38400 bps | |
| MODBUS® Parity | Even | Odd: Odd parity Even: Even parity Non: Non parity |
| Stop Bit | 1 bit | |

(*)MODBUS® Address is fixed to 1 station. If setting other than 1 station, this sample does not operate.

(2) Electronic multimeter (Model ME110)

| Item | Set value | Remarks |
|----------------------------|-----------|---|
| MODBUS® Address | 1 | (*) |
| MODBUS® Transmission Speed | 38.4 kbps | |
| MODBUS® Parity | Even | Odd: Odd parity Even: Even parity Non: Non parity |
| Stop Bit | 1 bit | |

(*)MODBUS® Address is fixed to 1 station. If setting other than 1 station, this sample does not operate.

4.2 Parameter Settings for the measuring terminal

The followings are the setting values at our operation check.

(1) Energy measuring module (Model EMU4)

| Item | Set value | Remarks |
|---------------------------------|-----------|---|
| Phase wire system | 3 | 1: 1P2W 2: 1P3W 3: 3P3W 4: 3P4W |
| Primary voltage (line voltage) | 220V | Valid when "phase wire system" is other than 3P4W |
| Primary voltage (phase voltage) | - | Valid when "phase wire system" is 3P4W |
| Type of current sensor | 0 | 0: Direct sensor 2: 5A sensor |
| Primary current | 1000x0.1A | |
| Demand period (Current) | 120s | |
| Demand period (Power) | 120s | |
| Current multiplier | -2(*1) | -3: x0.001 -2: x0.01 -1: x0.1 0: x1 |
| Voltage multiplier | -1(*1) | -1: x0.1 0: x1 |
| W · DW · Var multiplier | -2(*1) | -3: x0.001 -2: x0.01 |

| Item | Set value | Remarks |
|-------------------------|-----------|-----------|
| Wh · Varh multiplier | -1(*1) | -1: x0.1 |
| | | 0: x1 |
| | | 1: x10 |
| | | -2: x0.01 |
| | | -1: x0.1 |
| Frequency multiplier | x0.1 | 0: x1 |
| | | 1: x10 |
| | | 2: x100 |
| | | |
| Power factor multiplier | x0.1 | |

(*1) The multiplier is automatically changed when phase wire system, primary voltage, or primary current is changed.

(2) Electronic multimeter (Model ME110)

| Item | Set value | Remarks |
|---------------------------------|-----------|---|
| Phase wire system | 3 | 1: 1P2W |
| | | 2: 1P3W(RNT) |
| | | 3: 3P3W |
| | | 4: 3P4W |
| | | 5: 1P3W(RNS) |
| | | 6: 3P3W_3CT |
| Primary voltage (line voltage) | 6600V | Valid when "phase wire system" is other than 3P4W |
| Primary voltage (phase voltage) | - | Valid when "phase wire system" is 3P4W |
| Primary current | 50x0.1V | |
| Demand period (Current) | 10s | |
| Demand period (Power) | 10s | |
| CO2 emission coefficient | 1.5 | |
| Electric rate unit price | 1.3 | |
| Current multiplier | 1 | 1: x0.01 |
| | | 2: x0.1 |
| | | 3: x1 |
| | | 4: x10 |
| Voltage multiplier | 3 | 1: x0.1 |
| | | 2: x1 |
| | | 3: x10 |
| | | 4: x100 |
| W · DW · Var multiplier | 3 | 1: x0.0001 |
| | | 2: x0.001 |
| | | 3: x0.01 |
| | | 4: x0.1 |
| | | 5: x1 |
| | | 6: x10 |
| | | 7: x100 |
| Wh · Varh multiplier | 2 | 1: x0.01 |
| | | 2: x0.1 |
| | | 3: x1 |
| | | 4: x10 |
| | | 5: x100 |
| | | 6: x1000 |
| Frequency multiplier | x0.1 | |
| Power factor multiplier | x0.1 | |

5 SCREEN SPECIFICATIONS

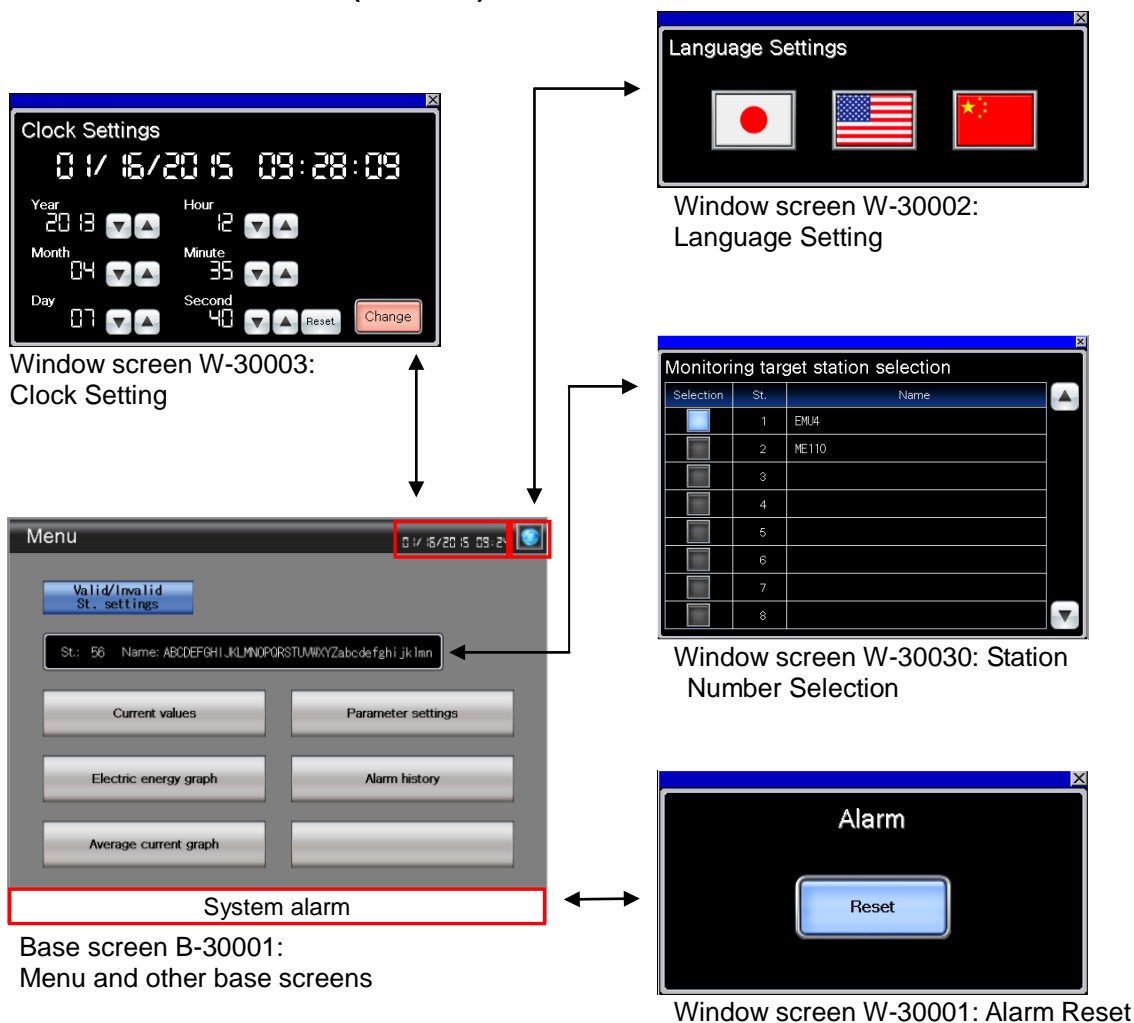
5.1 Display Language

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. 254 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 List/Transition

5.2.1 Screen list/transition (common)



5.2.2 Screen list/transition (individual)

Valid/Invalid St. settings

| Valid/Invalid | St. | Name | Model |
|---------------|-----|---|-------|
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |
| Valid | 56 | ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn | EMU4 |

Window screen
W-30031: Valid/Invalid St. settings

Menu

Valid/Invalid St. settings

St.: 56 Name: ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn

Current values

Parameter settings

Electric energy graph

Alarm history

Average current graph

Base screen B-30001: Menu

Phase wir

| | | | |
|------|---------|------|---------|
| 1P2W | Valid | 3P4W | Invalid |
| 1P3W | Invalid | | Invalid |
| 3P3W | Invalid | | Invalid |

OK Cancel

Window screen
W-30015:
Phase wire system

Type of current

| | |
|---------------|---------|
| Direct sensor | Valid |
| 5A sensor | Invalid |

OK Cancel

Window screen
W-30016:
Type of current sensor

Current Multiplier

| | | | |
|-------|---------|-----|---------|
| x0.01 | Invalid | x1 | Valid |
| x0.1 | Invalid | x10 | Invalid |

OK Cancel

Window screen
W-30017 to 30020:
Various multiplier

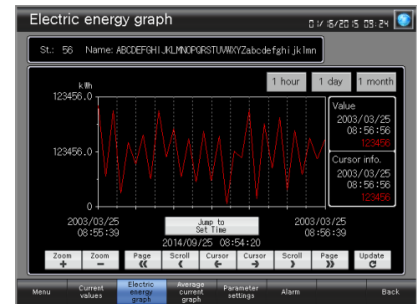
Current values

St.: 56 Name: ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn

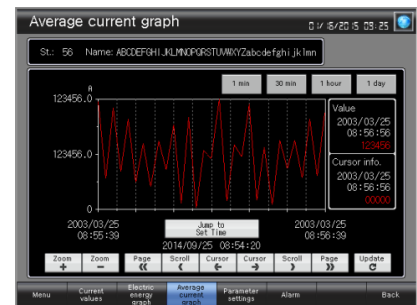
| Name | Value | Name | Value |
|-------------------|---------|-----------------|----------------|
| Current I1 | 123456A | Power | 123456W |
| Current I2 | 123456A | Power demand | 123456W |
| Current I3 | 123456A | Reactive power | 123456Var |
| Current demand I1 | 123456A | Electric energy | 123456kWh |
| Current demand I2 | 123456A | Reactive energy | 123456kVarh |
| Current demand I3 | 123456A | Frequency | 2345.6Hz |
| Voltage v1-3 | 123456V | Power factor | 12345.6% |
| Voltage v2-3 | 123456V | CO2 emission | 123456.0kg-CO2 |
| Voltage v3-1 | 123456V | Electric rate | 1234567\$ |

Menu Current values Electric energy graph Average current graph Parameter settings Alarm Back

Base screen B-30002:
Current values



Base screen B-30009:
Electric energy graph



Base screen 30010:
Average current gra

Parameter settings

St.: 56 Name: ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn

| Name | Value | Name | Value |
|--------------------------------|-----------------|-------------------------|----------|
| Phase wire system | Settings | Current Multiplier | Settings |
| Primary voltage(Line voltage) | 1234V | Voltage Multiplier | Settings |
| Primary voltage(Phase voltage) | 1234567+0.1V | W * DW * Var Multiplier | Settings |
| Type of current sensor | Direct sensor | Frequency Multiplier | x0.1 |
| Primary current | 1234+0.1A | Power factor Multiplier | x0.1 |
| Demand period (Current) | 1234s | | |
| Demand period (Power) | 1234s | | |
| CO2 emission coefficient | 1.000kg-CO2/kWh | | |
| Electric rate unit price | 34.001 /kWh | | |

Menu Current values Electric energy graph Average current graph Parameter settings Alarm Back

Base screen B-30020:
Parameter settings

Alarm history

St.: 56 Name: ABCDEFGHIJ.KLMNOPQRSTUVWXYZabcdefghi.jk.lmn

| Occurred | Message | Restored |
|-------------|---|----------|
| 01/18 09:25 | St.1 Current upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Power demand upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Voltage upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Current upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Power upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Reactive Power upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Frequency upper/lower limit alarm | 09:25 |
| 01/18 09:25 | St.1 Power factor upper/lower limit alarm | 09:25 |

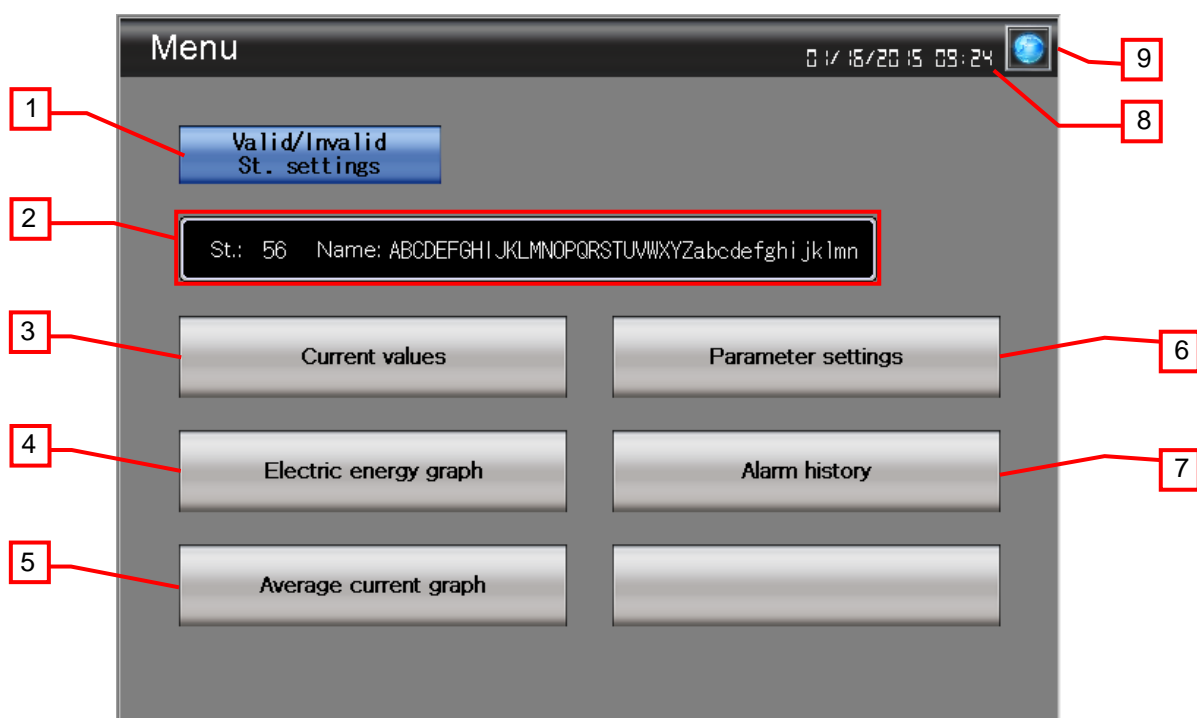
Delete One Delete All Monitor Status All Stations

Menu Current values Electric energy graph Average current graph Parameter settings Alarm Back

Base screen B-30011: Alarm history

5.3 Explanation of Screens

5.3.1 Menu (B-30001)



Outline

This is the Menu screen.

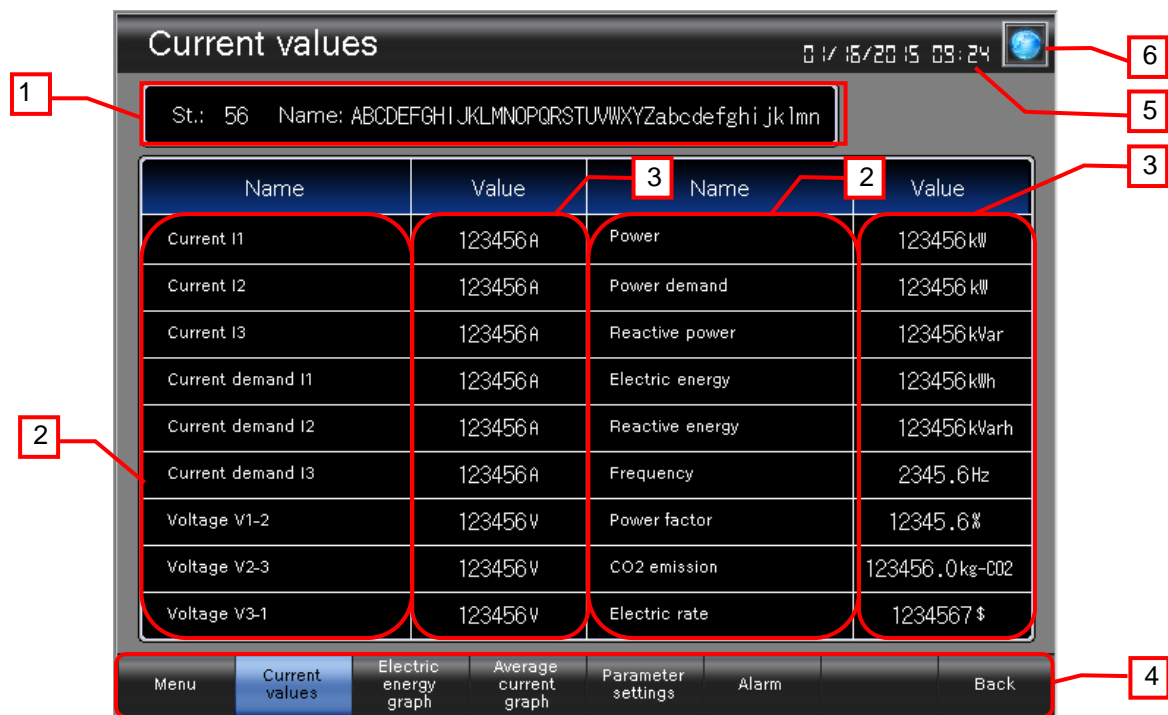
Description

1. Displays the window to switch the measuring terminal station number valid/invalid, or set the name and model.
2. Displays the station number and name of the measuring terminal currently monitored. Display the window to change the target station to be monitored by touching. Display the window to change the target station to be monitored by touching.
3. Switches to the [Current values] (1/7) screen.
4. Switches to the [Electric energy graph] screen.
5. Switches to the [Parameter settings] (1/5) screen.
6. Switches to the [Alarm history] screen.
7. Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching. This operation is the same for all base screens.
8. Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching.
9. Open the [Language Setting] window.

Remarks

- The information of 1 station is read from advanced recipe file when GOT is started and set as the target station to be monitored. For more details about scripts, please refer to "5.7 Script List".
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. [Alarm Reset] window will be displayed by touching an alarm message.

5.3.2 Current values (B-30002)



Outline

This screen monitors current values. The decimal point position of current value differs depending on the multiplier.

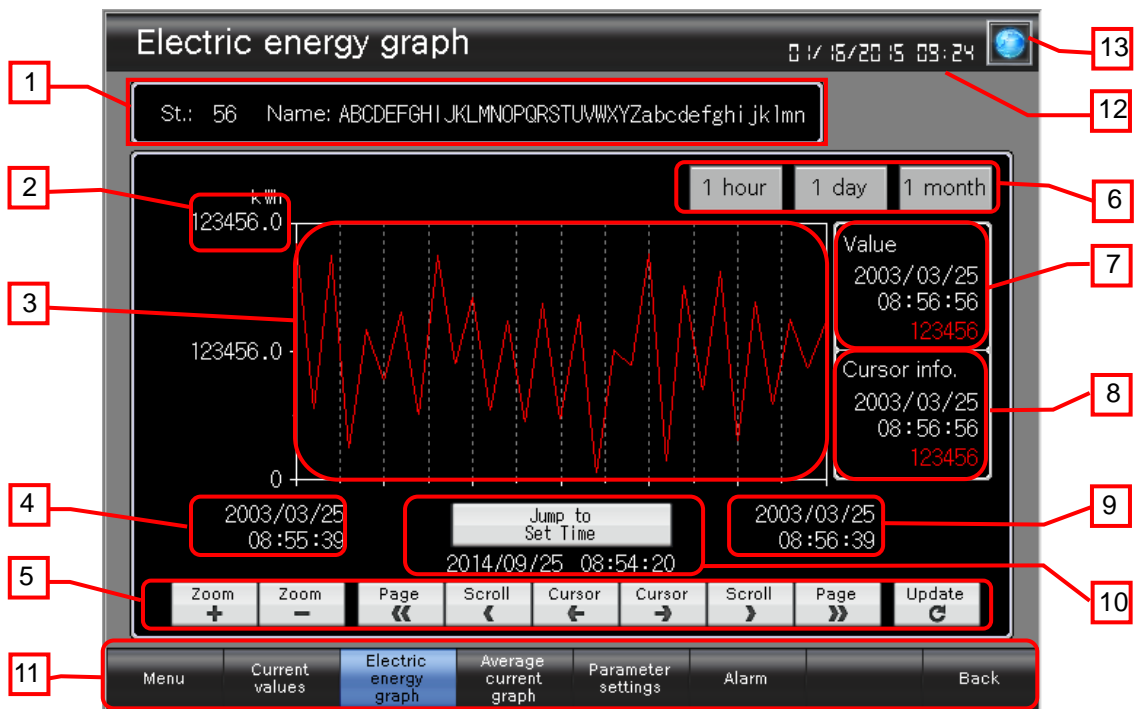
Description

1. Displays the station number and name of the measuring terminal currently monitored. Display the window to change the target station to be monitored by touching.
2. Displays the name of the measuring value currently monitored.
3. Displays the current values of the measuring terminal currently monitored.
4. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
5. Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching.
6. Open the [Language Setting] window.

Remarks

- The following are the multiplier pattern.
Model EMU4: x0.001, x0.01, x0.1, x1
Model ME110: x0.01, x0.1, x1, x10
- The multiplier of model EMU4 is automatically changed when phase wire system, primary voltage, or primary current is changed. The multiplier of model ME110 is set by the user.
- CO2 emission is calculated as follows.
CO2 emission (kg-CO2) = Electric energy (kWh) X CO2 emission coefficient (kg-CO2/kWh)
- CO2 emission is different from the display of CO2 emission (CO2 level corresponding value) on the measuring terminal.
- Electric rate is calculated as follows.
Electric rate (Yen or \$ or yuan) = Electric energy (kWh) X Electric rate unit price (Yen or \$ or yuan /kWh)
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. [Alarm Reset] window will be displayed by touching an alarm message.

5.3.3 Electric energy graph (B-30009)



Outline

This screen displays the current value of electric energy. The historical trend graph displays the past electric energy that is collected by logging function.

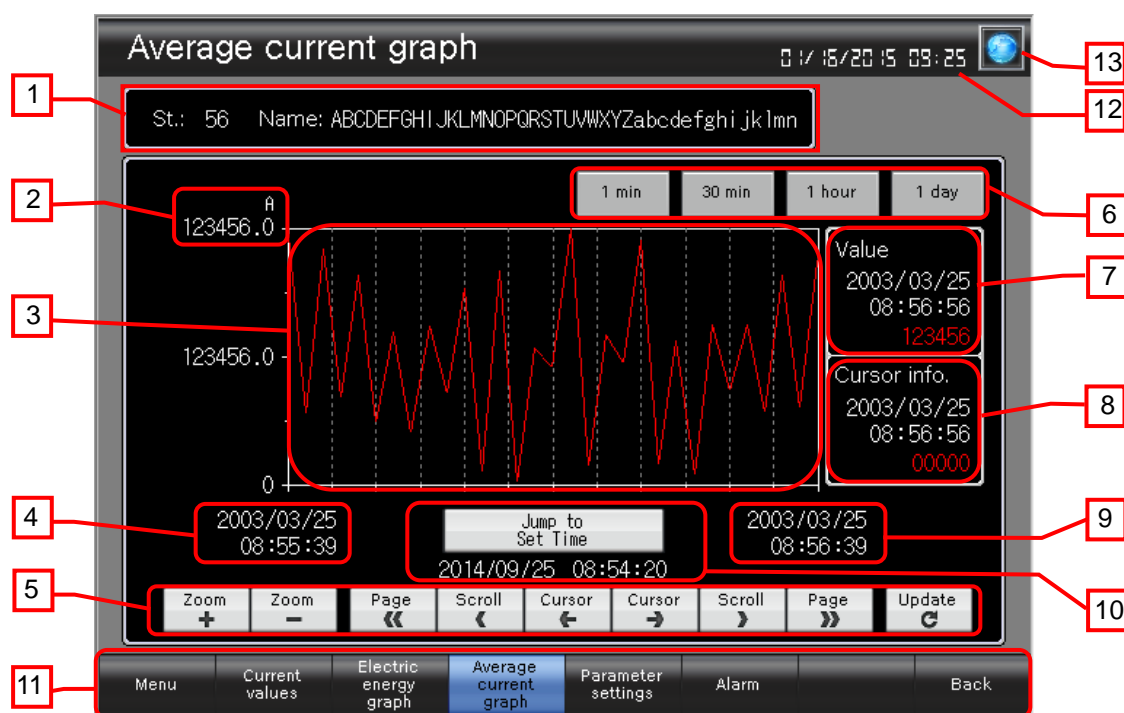
Description

- Displays the station number and name of the measuring terminal currently monitored. Display the window to change the target station to be monitored by touching.
- Displays the upper limit of the historical trend graph. Inputs the upper limit by touching the value. The input upper limit is saved by recipe function and the value is retained. The upper limit switches to the upper limit corresponding to the station number when the station number is switched.
- Displays the average current electric energy with a historical trend graph. Touch the graph to show the cursor. Data number of the horizontal axis is 31 points.
- Displays end position time in historical trend graph.
- Operates the historical trend graph.
- Switches the logging data that is displayed with a historical trend graph.
 - "1 minute": Displays the logging data for each one minute that is collected in every 00 seconds.
 - "30 minutes": Displays the logging data for each 30 minutes that is collected in every 00 minutes and 30 minutes.
 - "1 hour": Displays the logging data for each one hour that is collected in every 00 minutes.
 - "1 day": Displays the logging data for each one day that is collected in every day at 00 hour 00 minutes.
 - "1 month": Displays the logging data for each one month that is collected in every first day of each month at 00 hour 00 minutes.
- Displays the current values of average current electric energy. The decimal point position differs depending on the multiplier.
- Displays the date/time and average current of the cursor position. The decimal point position of the average current differs depending on the multiplier.
- Displays beginning position time in historical trend graph.
- Switches to the [Electric energy gra] screen. Moves the cursor to specified time.
- Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
- Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching.
- Open the [Language Setting] window.

Remarks

- The electric energy is transferred from measuring terminal to internal device of GOT by the device data transfer function. The transferred value is converted into real number by using a script and the converted value is logged. Therefore, if the value converted into real number exceeds seven digits, it may have a margin of error. For more details about scripts, please refer to "5.7 Script List".
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. [Alarm Reset] window will be displayed by touching an alarm message.

5.3.4 Average current gra (B-30010)



Outline

This screen displays the current value of average current. The historical trend graph displays the past average current that is collected by logging function.

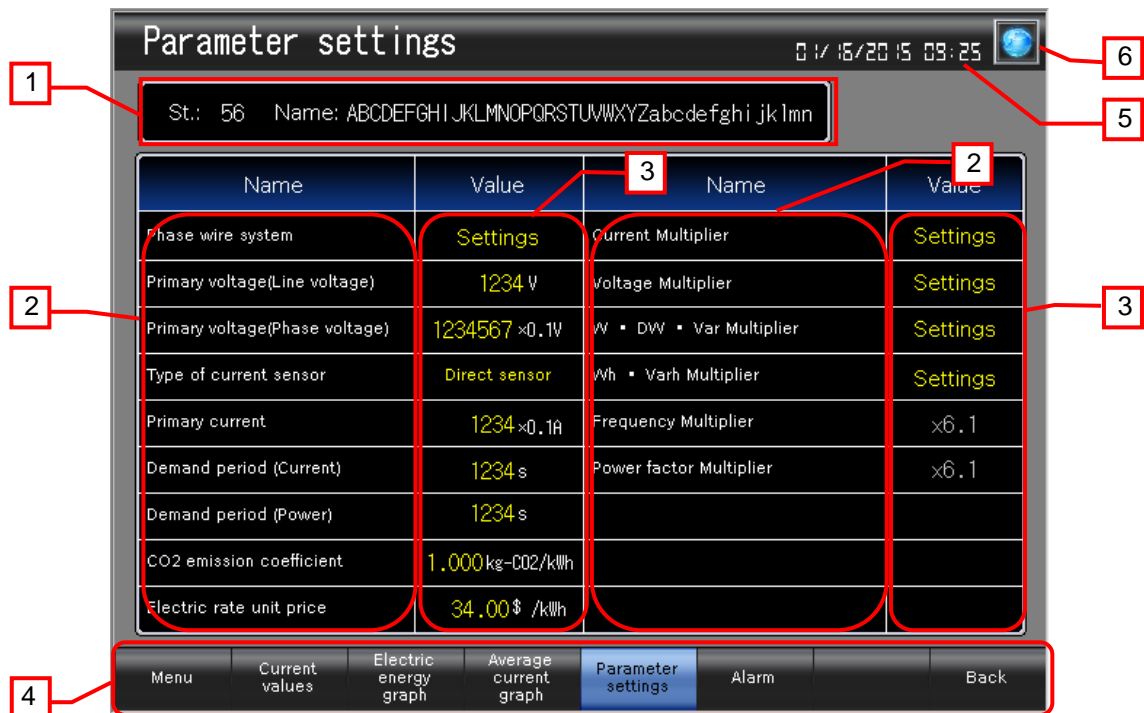
Description

- Displays the station number and name of the measuring terminal currently monitored. Display the window to change the target station to be monitored by touching.
- Displays the upper limit of the historical trend graph. Inputs the upper limit by touching the value. The input upper limit is saved by recipe function and the value is retained. The upper limit switches to the upper limit corresponding to the station number when the station number is switched.
- Displays the average current with a historical trend graph. Touch the graph to show the cursor. Data number of the horizontal axis is 31 points.
- Displays end position time in historical trend graph.
- Operates the historical trend graph.
- Switches the logging data that is displayed with a historical trend graph.
 - "1 minute": Displays the logging data for each one minute that is collected in every 00 seconds.
 - "30 minutes": Displays the logging data for each 30 minutes that is collected in every 00 minutes and 30 minutes
 - "1 hour": Displays the logging data for each one hour that is collected in every 00 minutes.
 - "1 day": Displays the logging data for each one day that is collected in every 00 minutes.
- Displays the current values of average current. The decimal point position differs depending on the multiplier.
- Displays the date/time and average current of the cursor position. The decimal point position of the average current differs depending on the multiplier.
- Displays beginning position time in historical trend graph.
- Switches to the [Electric energy gra] screen. Moves the cursor to specified time.
- Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
- Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching.
- Open the [Language Setting] window.

Remarks

- The electric energy is transferred from measuring terminal to internal device of GOT by the device data transfer function. The transferred value is converted into real number by using a script and the converted value is logged. Therefore, if the value converted into real number exceeds seven digits, it may have a margin of error. For more details about scripts, please refer to "5.6 Script List".
- As for model ME110, 0 is displayed on average current graph if the phase wire system is set to other than 3P4W.
- If a system alarm occurs, the alarm message will appear at the bottom of the screen. [Alarm Reset] window will be displayed by touching an alarm message.

5.3.5 Parameter settings (B-30020)



Outline

This window screen allows setting the parameter of EcoMonitor.

Description

1. Displays the station number and name of the measuring terminal currently monitored. Display the window to change the target station to be monitored by touching.
2. Displays the name of the setting value.
3. Displays the window to set each parameter or the key window to input value after clicking a Value.
4. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
5. Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching.
6. Open the [Language Setting] window.

Remarks

- If a system alarm occurs, the alarm message will appear at the bottom of the screen. [Alarm Reset] window will be displayed by touching an alarm message.



5.3.6 Alarm history (B-30011)



Outline

This screen displays the alarm and error status of the measuring terminal.

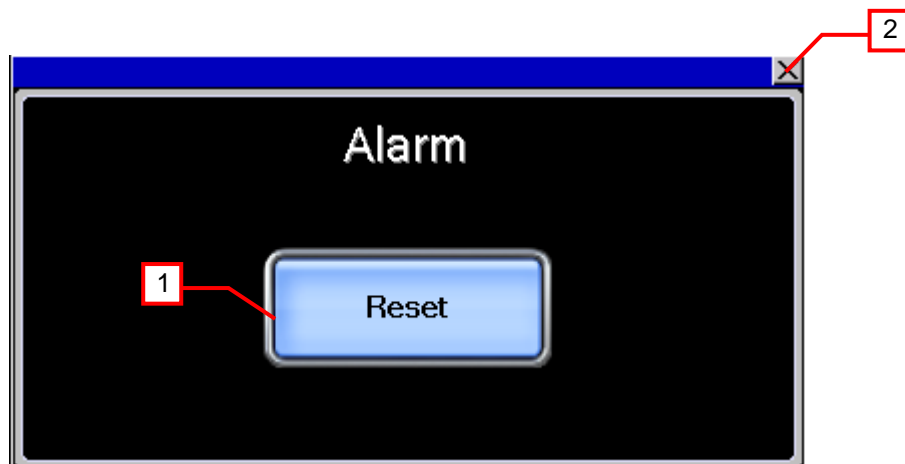
Description

1. Displays the alarm (user) with advanced user alarm display.
2. Operates the alarm (user) advanced user alarm display.
Delete One: Deletes only the selected restored alarm.
Delete All: Deletes all the restored alarms.
3. Scrolls for the alarm (user).
 Scrolls the page up and down.
 Moves the cursor up and down.
4. Switches All Stations/ Monitor Station for the alarm (user) display.
5. Switches to each screen. The blue switch indicates the currently displayed screen, thus selecting this switch will not switch the screen.
6. Displays the current date and time. Opens the [Clock Setting] or [Language Setting] window by touching.
7. Open the [Language Setting] window.

Remarks

- If a system alarm occurs, the alarm message will appear at the bottom of the screen. [Alarm Reset] window will be displayed by touching an alarm message.

5.3.7 Alarm Reset (W-30001)



Outline

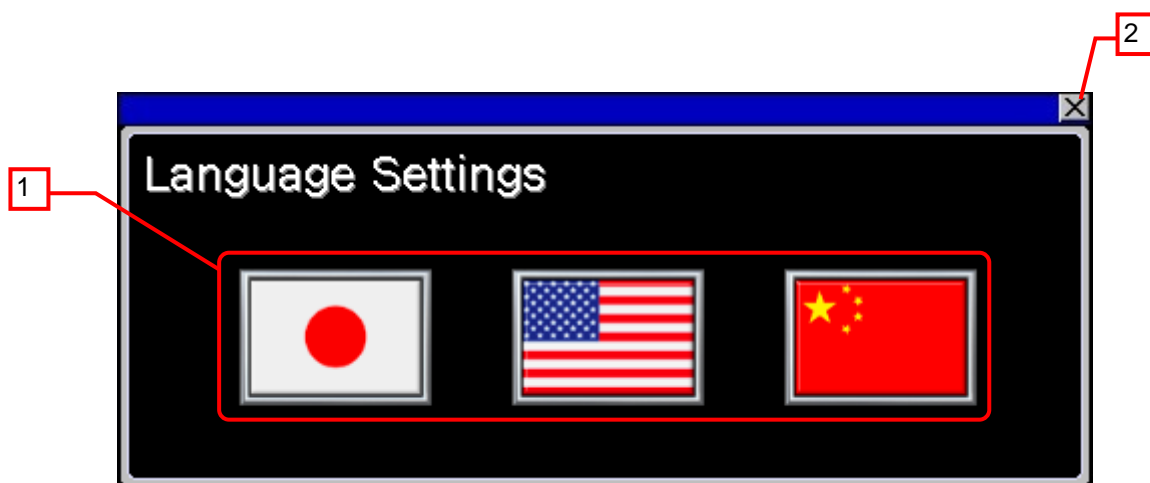
This window screen allows resetting the system alarm.

Description

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

Remarks

5.3.8 Language Setting (W-30002)



Outline

This window screen allows selecting the GOT language.

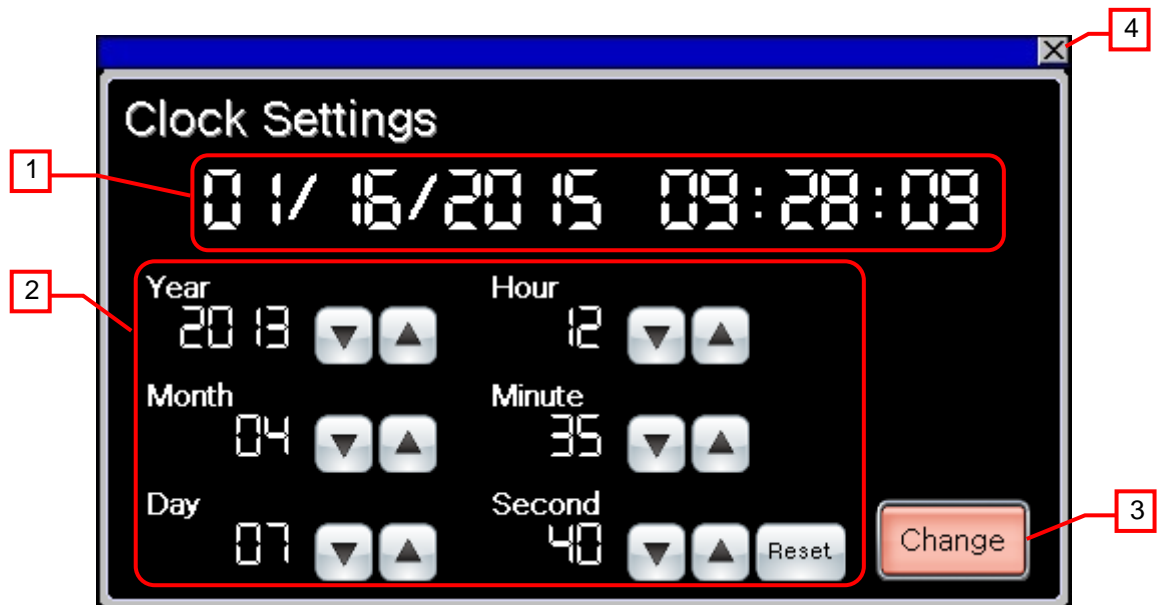
Description

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

Remarks

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



5.3.9 Clock Setting (W-30003)



Outline

This window screen allows changing the GOT clock data.

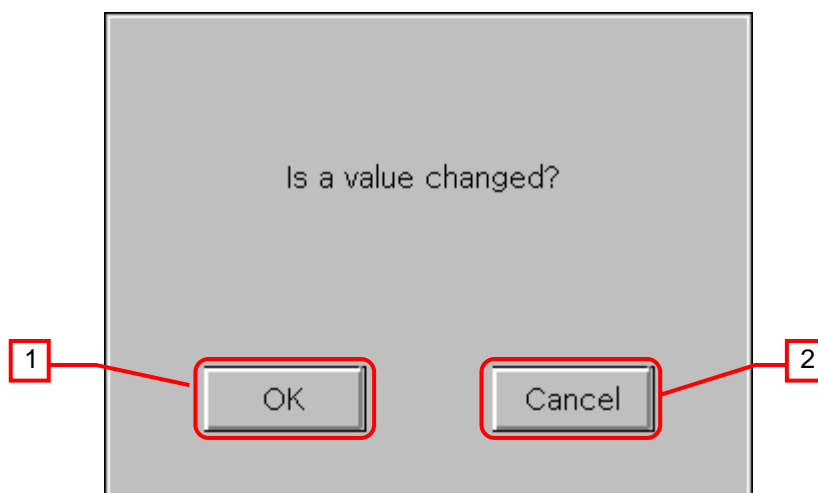
Description

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 after one second.
4. Closes the window.

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 "5.7 Script List".

5.3.10 Confirmation dialog (W-30014)



Outline

A dialog window which is displayed when determining the value set with each parameter setting window.

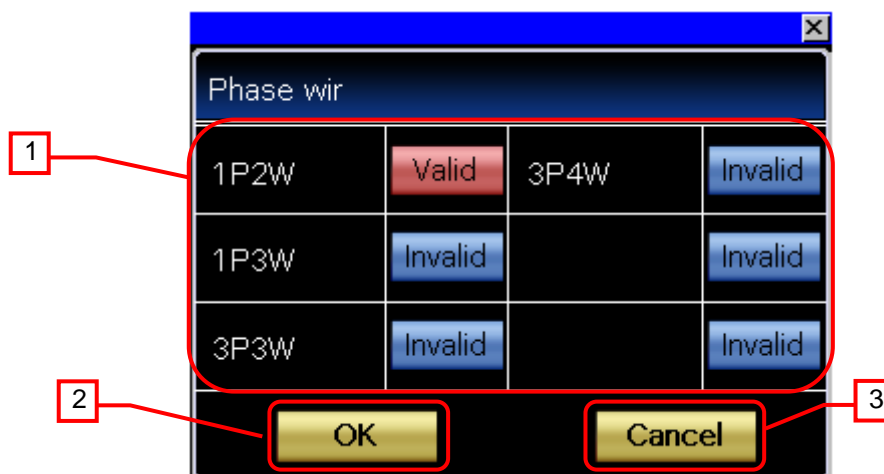
Description

1. Closes the dialog after setting the input value.
2. Closes the dialog without determining the input.

Remarks

- The following parameter values are saved to advanced recipe file to be backed up by GOT.
Common in model EMU4 and model ME110: CO₂ emission coefficient, Electric rate unit price
Only for model ME110: Various multiplier (Except for frequency and power factor)

5.3.11 Phase wire system (W-30015)



Outline

This screen sets the Phase wire system. The setting contents of phase wire system differ depending on the model.

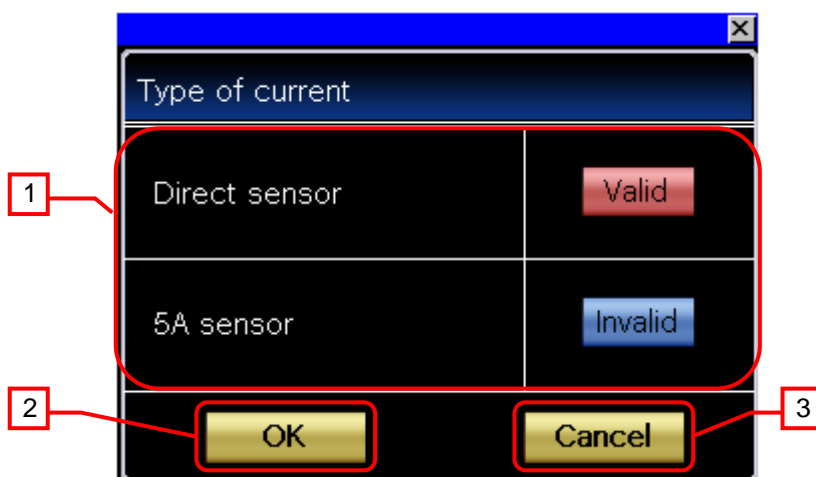
Description

1. Select the phase wire system to be set. The following phase wire systems can be set for each model.
Model EMU4: 1P2W, 1P3W, 3P3W, 3P4W
Model ME110: 1P2W, 1P3W(RNT), 3P3W, 3P4W, 1P3W(RNS), 3P3W_3CT
2. Displays the confirmation dialog window. The input is determined when touching "OK".
3. Discards the setting value and closes the window.

Remarks

- As for model ME110, it takes two or three seconds to change the phase wire system. Note that a communication error may occur if changing the setting of various parameters during this time.

5.3.12 Type of current sensor (W-30016)



Outline

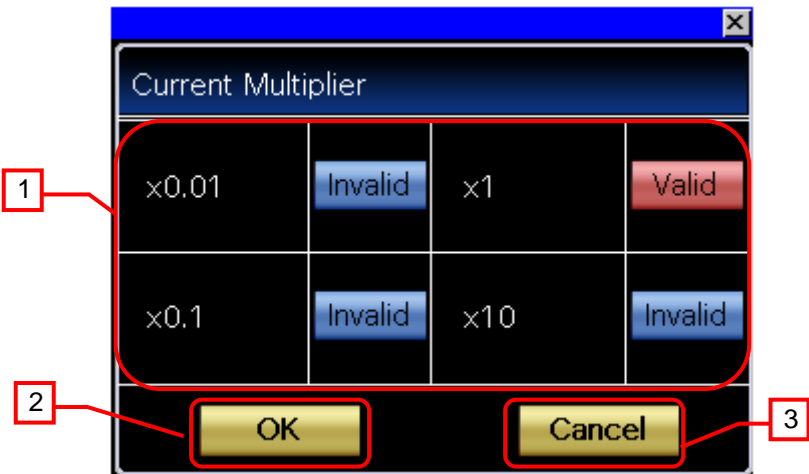
This screen sets the Type of current sensor for model EMU4.

Description

1. Selects the type of current sensor to be set.
2. Displays the confirmation dialog window. The input is determined when touching "OK".
3. Discards the setting value and closes the window.

Remarks

5.3.13 Current Multiplier (W-30017)



Outline

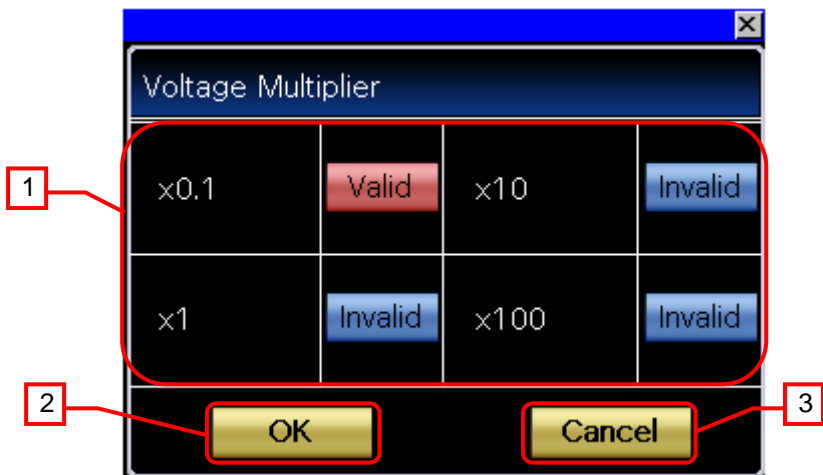
This screen sets the Current Multiplier for model ME110.

Description

- 1. Selects the multiplier to be set.
- 2. Displays the confirmation dialog window. The input is determined when touching "OK".
- 3. Discards the setting value and closes the window.

Remarks

5.3.14 Voltage Multiplier (W-30018)



Outline

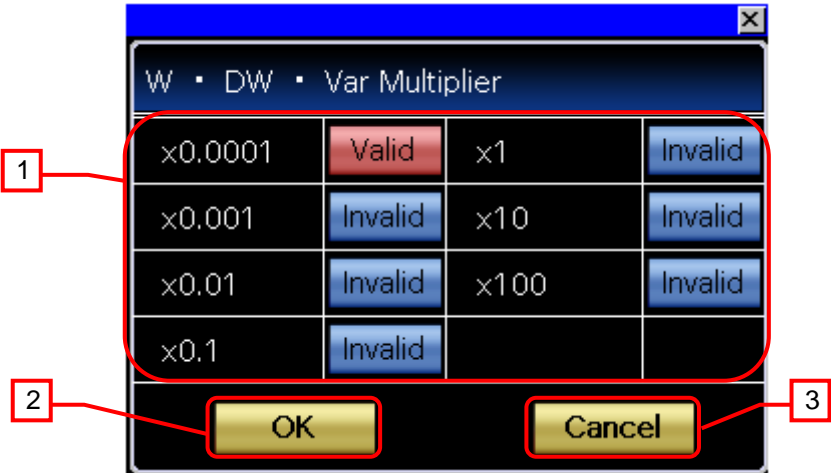
This screen sets the Voltage Multiplier for model ME110.

Description

- 1. Selects the multiplier to be set.
- 2. Displays the confirmation dialog window. The input is determined when touching "OK".
- 3. Discards the setting value and closes the window.

Remarks

5.3.15 W · DW · Var Multiplier (W-30019)



Outline

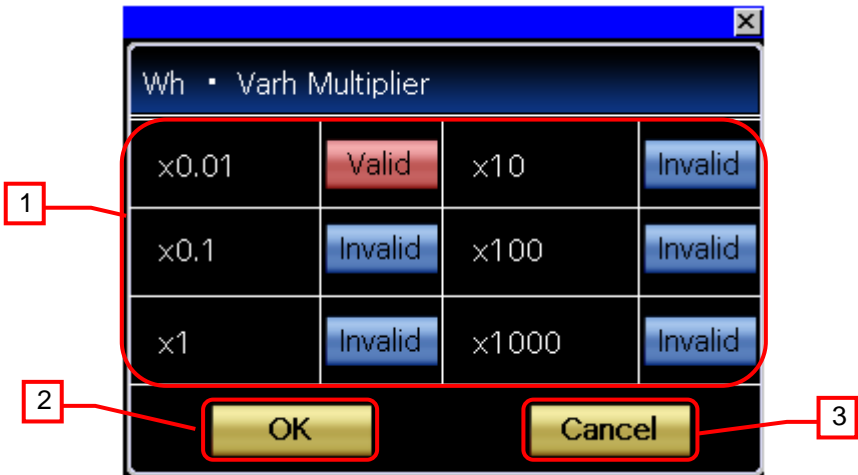
This screen sets the W · DW · Var Multiplier for model ME110.

Description

- 1. Selects the multiplier to be set.
- 2. Displays the confirmation dialog window. The input is determined when touching "OK".
- 3. Discards the setting value and closes the window.

Remarks

5.3.16 Wh · Varh Multiplier (W-30020)



Outline

This screen sets the Wh · Varh Multiplier for model ME110.

Description

- 1. Selects the multiplier to be set.
- 2. Displays the confirmation dialog window. The input is determined when touching "OK".
- 3. Discards the setting value and closes the window.

Remarks

5.3.17 Valid/Invalid station setting (W-30031)

The screenshot shows the 'Valid/Invalid St. settings' window. It contains a table with 8 rows. The first two rows are highlighted with a red box labeled '1'. The first column of the table contains buttons labeled 'Valid' or 'Invalid'. The second column contains station numbers 1 through 8. The third column contains station names 'EMU4' and 'ME110'. The fourth column contains buttons labeled 'EMU4' or '----'. A red box labeled '2' highlights the top-right corner of the window, including the title bar and a close button. A red box labeled '3' highlights the 'EMU4' button in the fourth column of the first row. A red box labeled '4' highlights the bottom-right corner of the window, including a scroll bar and a down arrow button. A red box labeled '5' highlights the close button in the top-right corner of the window.

| Valid/Invalid | St. | Name | Model |
|---------------|-----|-------|-------|
| Valid | 1 | EMU4 | EMU4 |
| Valid | 2 | ME110 | EMU4 |
| Invalid | 3 | | ---- |
| Invalid | 4 | | ---- |
| Invalid | 5 | | ---- |
| Invalid | 6 | | ---- |
| Invalid | 7 | | ---- |
| Invalid | 8 | | ---- |

Outline

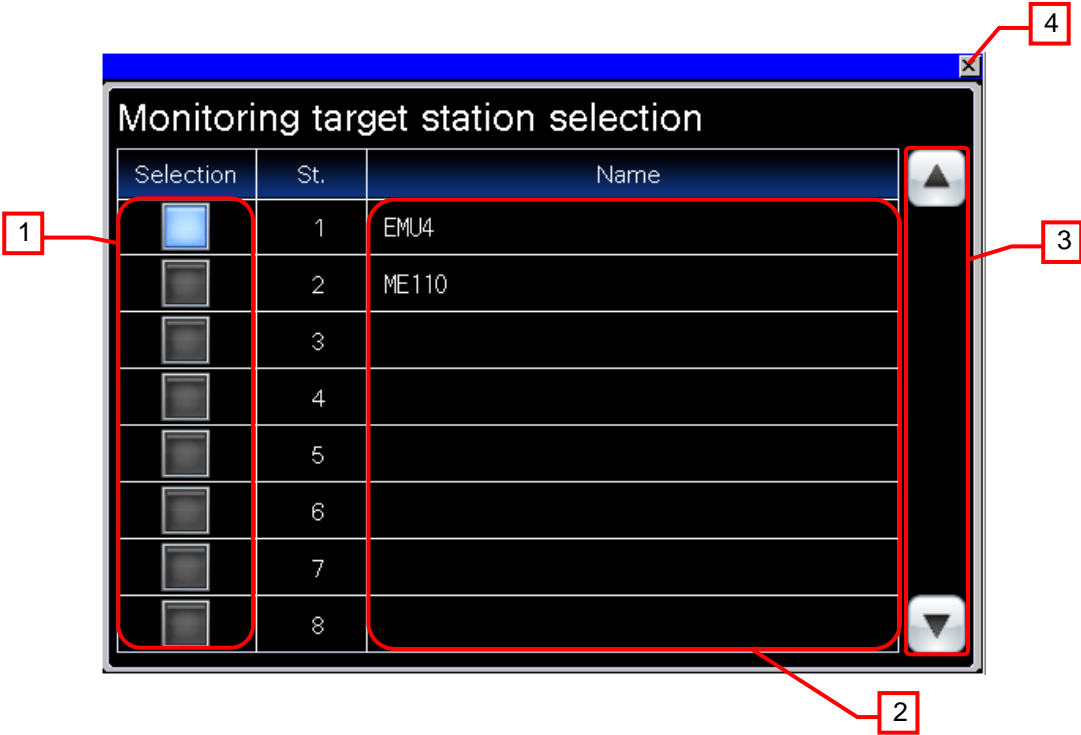
This window screen allows switching the station number of measuring terminal Valid/Invalid, setting the Name and Model.

Description

1. Switches the station number of measuring terminal Valid/Invalid.
2. Sets the name of measuring terminal whose station number is specified.
3. Selects the model of measuring terminal whose station number is specified. Select model EMU4 or model ME110.
4. Changes station No. with the switch.
5. Closes the window.

Remarks


5.3.18 Station Number Selection (W-30030)



Outline

This window screen allows selecting a station to be monitored.

Description

- 1. Switches the target station to be monitored.
- 2. Displays names of the measuring terminal specified by Valid/invalid station setting.
- 3.  Changes station No. with the switch.
- 4. Closes the window.

Remarks

5.4 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".

5.4.1 Devices of the controller

| Type | Device No. | Application | Model EMU4 | Model ME110 |
|------|----------------------------|--------------------------------------|------------|-------------|
| Bit | 1-1 to 1-31: 400525.b2 | Current demand maximum/minimum alarm | ○ | ○ |
| | 1-1 to 1-31: 400525.b3 | Power demand maximum/minimum alarm | ○ | ○ |
| | 1-1 to 1-31: 400525.b8 | Voltage maximum/minimum alarm | ○ | ○ |
| | 1-1 to 1-31: 400525.b9 | Current maximum/minimum alarm | - | ○ |
| | 1-1 to 1-31: 400525.b10 | Power maximum/minimum alarm | - | ○ |
| | 1-1 to 1-31: 400525.b11 | Reactive power maximum/minimum alarm | - | ○ |
| | 1-1 to 1-31: 400525.b12 | Frequency maximum/minimum alarm | - | ○ |
| | 1-1 to 1-31: 400525.b13 | Power factor maximum/minimum alarm | ○ | ○ |
| | 1-1 to 1-31: 400525.b14 | Harmonic voltage maximum alarm | - | ○ |
| | 1-1 to 1-31: 400525.b15 | Harmonic current maximum alarm | - | ○ |
| Word | 400513 | Phase wire system | ○ | ○ |
| | 400514 | Primary voltage (Line voltage) | ○ | ○ |
| | 400516 | Primary voltage (Phase voltage) | ○ | ○ |
| | 400520 | Primary current | ○ | ○ |
| | 400522 | Demand period (Power) | ○ | ○ |
| | 400523 | Demand period (Current) | ○ | ○ |
| | 400532 | Type of current sensor | ○ | - |
| | 400755 | Current multiplier | ○ | - |
| | 400756 | Voltage multiplier | ○ | - |
| | 400757 | W · DW · Var multiplier | ○ | - |
| | 400758 | Wh · Varh multiplier | ○ | - |
| | 400769 | Current I1 | ○ | ○ |
| | 400770 | Current I2 | ○ | ○ |
| | 400771 | Current I3 | ○ | ○ |
| | 400773 | Current value (average) | ○ | ○ |
| | 400774 | Current demand I1 | ○ | ○ |
| | 400775 | Current demand I2 | ○ | ○ |
| | 400776 | Current demand I3 | ○ | ○ |
| | 400779 | Voltage V1-2 | ○ | ○ |
| | 400780 | Voltage V2-3 | ○ | ○ |
| | 400781 | Voltage V3-1 | ○ | ○ |
| | 400790 | Power factor | ○ | ○ |
| | 400791 | Frequency | ○ | ○ |
| | 400795 | Power | ○ | ○ |
| | 400799 | Power demand | ○ | ○ |
| | 400803 | Phase wire system | ○ | ○ |
| | 401305 | Electric energy | ○ | ○ |
| | 401309 | Reactive energy | ○ | ○ |

○: With device -: Without device

5.4.2 GOT internal devices

| Type | Device No. | Application |
|------|-------------|---|
| Bit | GB40 | Script trigger (Always ON) |
| | GB61000 | Logging ID: 30001 trigger device |
| | GB61001 | Logging ID: 30002 trigger device |
| | GB61002 | Logging ID: 30003 trigger device |
| | GB61003 | Logging ID: 30004 trigger device |
| | GB61004 | Logging ID: 30005 trigger device |
| | GB61005 | Logging ID: 30006 trigger device |
| | GB61006 | Logging ID: 30007 trigger device |
| | GB61007 | Script No.30005 trigger device |
| | GB61008 | Script No.30006 trigger device |
| | GB61009 | Script No.30014 trigger device |
| | GB61010 | Script No.30017, 30023 trigger device |
| | GB61011 | Script No.30004, 30024 trigger device |
| | GB61012 | Script No.30020 trigger device |
| | GB61013 | Script No.30021 trigger device |
| | GB61014 | Script No.30025 trigger device |
| | GB61015 | Script No.30030 Recipe read flag |
| | GB61016 | Primary voltage (line voltage, phase voltage) display/hide flag |
| | GB61017 | Device data for logging transfer execution timing control flag |
| | GB61018 | Script No.30018 trigger device |
| | GB61019 | Script No.30034 trigger device |
| | GB61020 | Script No.30035 trigger device |
| | GB61021 | Valid/invalid station status flag (offset) |
| | GB61022 | Valid/invalid station status flag (offset) |
| | GB61023 | Valid/invalid station status flag (offset) |
| | GB61024 | Valid/invalid station status flag (offset) |
| | GB61025 | Valid/invalid station status flag (offset) |
| | GB61026 | Valid/invalid station status flag (offset) |
| | GB61027 | Valid/invalid station status flag (offset) |
| | GB61028 | Valid/invalid station status flag (offset) |
| | GB61052 | Monitor target station selection switch trigger |
| | GB61053 | Monitor target station selection switch trigger |
| | GB61054 | Monitor target station selection switch trigger |
| | GB61055 | Monitor target station selection switch trigger |
| | GB61056 | Monitor target station selection switch trigger |
| | GB61057 | Monitor target station selection switch trigger |
| | GB61058 | Monitor target station selection switch trigger |
| | GB61059 | Monitor target station selection switch trigger |
| | GB61060 | Station No. selection lamp (offset) |
| | GB61061 | Station No. selection lamp (offset) |
| | GB61062 | Station No. selection lamp (offset) |
| | GB61063 | Station No. selection lamp (offset) |
| | GB61064 | Station No. selection lamp (offset) |
| | GB61065 | Station No. selection lamp (offset) |
| | GB61066 | Station No. selection lamp (offset) |
| | GB61067 | Station No. selection lamp (offset) |
| | GB61091 | Script No. 30036 trigger device |
| | GB61092 | Station No. 32 lamp display/hide flag |
| | GB61093 | Script No. 30039 trigger device |
| | GD60031.b13 | GOT error reset signal |
| | GD62557.b4 | Script No.30002 trigger device |
| | GD62557.b5 | Script No.30015 trigger device |
| | GD62560.b0 | Recipe 1 write trigger device |
| Bit | GD62560.b1 | Recipe 1 read trigger device |
| | GD62562.b1 | Device data transfer ID: 193, 194 completion flag |
| | GD62562.b2 | Device data transfer ID: 195, 196 completion flag |
| | GD62562.b3 | Device data transfer ID: 197, 198 completion flag |
| | GD62562.b4 | Device data transfer ID: 199, 200 completion flag |

| Type | Device No. | Application |
|------|-------------|---|
| | GD62562.b5 | Device data transfer ID: 201, 202 completion flag |
| | GD62562.b6 | Device data transfer ID: 203, 204 completion flag |
| | GD62562.b7 | Device data transfer ID: 205, 206 completion flag |
| | GD62562.b8 | Device data transfer ID: 207, 208 completion flag |
| | GD62562.b9 | Device data transfer ID: 209, 210 completion flag |
| | GD62562.b10 | Device data transfer ID: 211, 212 completion flag |
| | GD62562.b11 | Device data transfer ID: 213, 214 completion flag |
| | GD62562.b12 | Device data transfer ID: 215, 216 completion flag |
| | GD62562.b13 | Device data transfer ID: 217, 218 completion flag |
| | GD62562.b14 | Device data transfer ID: 219, 220 completion flag |
| | GD62562.b15 | Device data transfer ID: 221, 222 completion flag |
| | GD62563.b0 | Device data transfer ID: 223, 224 completion flag |
| | GD62563.b1 | Device data transfer ID: 225, 226 completion flag |
| | GD62563.b2 | Device data transfer ID: 227, 228 completion flag |
| | GD62563.b3 | Device data transfer ID: 229, 230 completion flag |
| | GD62563.b4 | Device data transfer ID: 231, 232 completion flag |
| | GD62563.b5 | Device data transfer ID: 233, 234 completion flag |
| | GD62563.b6 | Device data transfer ID: 235, 236 completion flag |
| | GD62563.b7 | Device data transfer ID: 237, 238 completion flag |
| | GD62563.b8 | Device data transfer ID: 239, 240 completion flag |
| | GD62563.b9 | Device data transfer ID: 241, 242 completion flag |
| | GD62563.b10 | Device data transfer ID: 243, 244 completion flag |
| | GD62563.b11 | Device data transfer ID: 245, 246 completion flag |
| | GD62563.b12 | Device data transfer ID: 247, 248 completion flag |
| | GD62563.b13 | Device data transfer ID: 249, 250 completion flag |
| | GD62563.b14 | Device data transfer ID: 251, 252 completion flag |
| | GD62563.b15 | Device data transfer ID: 253, 254 completion flag |
| | GD62640.b0 | Device data transfer ID: 193 trigger device |
| | GD62642.b0 | Device data transfer ID: 194 trigger device |
| | GD62644.b0 | Device data transfer ID: 195 trigger device |
| | GD62646.b0 | Device data transfer ID: 196 trigger device |
| | GD62648.b0 | Device data transfer ID: 197 trigger device |
| | GD62650.b0 | Device data transfer ID: 198 trigger device |
| | GD62652.b0 | Device data transfer ID: 199 trigger device |
| | GD62654.b0 | Device data transfer ID: 200 trigger device |
| | GD62656.b0 | Device data transfer ID: 201 trigger device |
| | GD62658.b0 | Device data transfer ID: 202 trigger device |
| | GD62660.b0 | Device data transfer ID: 203 trigger device |
| | GD62662.b0 | Device data transfer ID: 204 trigger device |
| | GD62664.b0 | Device data transfer ID: 205 trigger device |
| | GD62666.b0 | Device data transfer ID: 206 trigger device |
| | GD62668.b0 | Device data transfer ID: 207 trigger device |
| | GD62670.b0 | Device data transfer ID: 208 trigger device |
| | GD62672.b0 | Device data transfer ID: 209 trigger device |
| | GD62674.b0 | Device data transfer ID: 210 trigger device |
| | GD62676.b0 | Device data transfer ID: 211 trigger device |
| | GD62678.b0 | Device data transfer ID: 212 trigger device |
| | GD62680.b0 | Device data transfer ID: 213 trigger device |
| Bit | GD62682.b0 | Device data transfer ID: 214 trigger device |
| | GD62684.b0 | Device data transfer ID: 215 trigger device |
| | GD62686.b0 | Device data transfer ID: 216 trigger device |
| | GD62688.b0 | Device data transfer ID: 217 trigger device |
| | GD62690.b0 | Device data transfer ID: 218 trigger device |
| | GD62692.b0 | Device data transfer ID: 219 trigger device |
| | GD62694.b0 | Device data transfer ID: 220 trigger device |
| | GD62696.b0 | Device data transfer ID: 221 trigger device |
| | GD62698.b0 | Device data transfer ID: 222 trigger device |
| | GD62700.b0 | Device data transfer ID: 223 trigger device |
| | GD62702.b0 | Device data transfer ID: 224 trigger device |
| | GD62704.b0 | Device data transfer ID: 225 trigger device |

| Type | Device No. | Application |
|------|-------------------|---|
| | GD62706.b0 | Device data transfer ID: 226 trigger device |
| | GD62708.b0 | Device data transfer ID: 227 trigger device |
| | GD62710.b0 | Device data transfer ID: 228 trigger device |
| | GD62712.b0 | Device data transfer ID: 229 trigger device |
| | GD62714.b0 | Device data transfer ID: 230 trigger device |
| | GD62716.b0 | Device data transfer ID: 231 trigger device |
| | GD62718.b0 | Device data transfer ID: 232 trigger device |
| | GD62720.b0 | Device data transfer ID: 233 trigger device |
| | GD62722.b0 | Device data transfer ID: 234 trigger device |
| | GD62724.b0 | Device data transfer ID: 235 trigger device |
| | GD62726.b0 | Device data transfer ID: 236 trigger device |
| | GD62728.b0 | Device data transfer ID: 237 trigger device |
| | GD62730.b0 | Device data transfer ID: 238 trigger device |
| | GD62732.b0 | Device data transfer ID: 239 trigger device |
| | GD62734.b0 | Device data transfer ID: 240 trigger device |
| | GD62736.b0 | Device data transfer ID: 241 trigger device |
| | GD62738.b0 | Device data transfer ID: 242 trigger device |
| | GD62740.b0 | Device data transfer ID: 243 trigger device |
| | GD62742.b0 | Device data transfer ID: 244 trigger device |
| | GD62744.b0 | Device data transfer ID: 245 trigger device |
| | GD62746.b0 | Device data transfer ID: 246 trigger device |
| | GD62748.b0 | Device data transfer ID: 247 trigger device |
| | GD62750.b0 | Device data transfer ID: 248 trigger device |
| | GD62752.b0 | Device data transfer ID: 249 trigger device |
| | GD62754.b0 | Device data transfer ID: 250 trigger device |
| | GD62756.b0 | Device data transfer ID: 251 trigger device |
| | GD62758.b0 | Device data transfer ID: 252 trigger device |
| | GD62760.b0 | Device data transfer ID: 253 trigger device |
| | GD62762.b0 | Device data transfer ID: 254 trigger device |
| | GS512.b0 | Time change signal |
| Word | GD10 | Station No. indirect specification device |
| | GD60000 | Base screen switching |
| | GD60001 | Overlap window 1 screen switching |
| | GD60004 | Overlap window 2 screen switching |
| | GD60018 | Dialog window screen switching |
| | GD60021 | Language switching |
| | GD60022 | System language switching |
| | GD60031, GD60041 | System information |
| | GD6100 to GD61039 | Station No.1 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6104 to GD61079 | Station No.2 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| Word | GD6108 to GD61119 | Station No.3 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6112 to GD61159 | Station No.4 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6116 to GD61199 | Station No.5 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6120 to GD61239 | Station No.6 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6124 to GD61279 | Station No.7 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6128 to GD61319 | Station No.8 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6132 to GD61359 | Station No.9 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6136 to GD61399 | Station No.10 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6140 to GD61439 | Station No.11 Various parameters (Refer to Station No.1 Name to Type of current sensor) |

| Type | Device No. | Application |
|------|-------------------|---|
| | GD6144 to GD61479 | Station No.12 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6148 to GD61519 | Station No.13 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6152 to GD61559 | Station No.14 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6156 to GD61599 | Station No.15 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6160 to GD61639 | Station No.16 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6164 to GD61679 | Station No.17 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6168 to GD61719 | Station No.18 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6172 to GD61759 | Station No.19 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6176 to GD61799 | Station No.20 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6180 to GD61839 | Station No.21 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6184 to GD61879 | Station No.22 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6188 to GD61919 | Station No.23 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6192 to GD61959 | Station No.24 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6196 to GD61999 | Station No.25 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6200 to GD62039 | Station No.26 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6204 to GD62079 | Station No.27 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6208 to GD62119 | Station No.28 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6212 to GD62159 | Station No.29 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6216 to GD62199 | Station No.30 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6220 to GD62239 | Station No.31 Various parameters (Refer to Station No.1 Name to Type of current sensor) |
| | GD6224 to GD62301 | Station No.1 to 31 Electric energy logging device |
| | GD6230 to GD62362 | Station No.1 to 31 Average current logging device |
| | GD6236 to GD62425 | Station No.1 to 31 Electric energy device data transfer destination |
| | GD6242 to GD62456 | Station No.1 to 31 Average current device data transfer destination |
| | GD6245 to GD62487 | Station No.1 to 31 Average current multiplier device data transfer destination |
| | GD6248 to GD62518 | Station No.1 to 31 Electric energy multiplier device data transfer destination |
| | GD62552 | Parameter offset device |
| | GD62553 | Current model storage device |
| | GD62554 | Advanced recipe common setting External control device |
| | GD62555 | Advanced recipe common setting Recipe No. storage device |
| | GD62556 | Advanced recipe common setting Record No. storage device |
| | GD62557 | Advanced recipe common setting External notification device |
| | GD62558 | Advanced recipe common setting Recipe No. notification device |
| | GD62559 | Advanced recipe common setting Record No. notification device |
| | GD62560 | Advanced recipe No.30001 Write/read trigger |
| | GD62561 | Advanced recipe No.30001 Record No. |
| | GD62562 | Device data transfer completion detection flag |
| | GD62563 | Device data transfer completion detection flag |
| | GD62564 | Electric energy graph Logging ID device |

| Type | Device No. | Application | |
|------|------------|---|------------------------------|
| | GD62565 | Electric energy graph | Graph offset device |
| | GD62566 | Electric energy graph | Upper limit device |
| | GD62567 | Electric energy graph | Upper limit device |
| | GD62568 | Electric energy graph | Cursor position time |
| | GD62569 | Electric energy graph | Cursor position time |
| | GD62570 | Electric energy graph | Cursor position time |
| Word | GD62571 | Electric energy graph | Cursor position time |
| | GD62572 | Electric energy graph | Beginning position time |
| | GD62573 | Electric energy graph | Beginning position time |
| | GD62574 | Electric energy graph | Beginning position time |
| | GD62575 | Electric energy graph | Beginning position time |
| | GD62576 | Electric energy graph | End position time |
| | GD62577 | Electric energy graph | End position time |
| | GD62578 | Electric energy graph | End position time |
| | GD62579 | Electric energy graph | End position time |
| | GD62580 | Electric energy graph | Cursor position device value |
| | GD62581 | Electric energy graph | Cursor position device value |
| | GD62582 | Average current graph | Logging ID device |
| | GD62583 | Average current graph | Graph offset device |
| | GD62584 | Average current graph | Upper limit device |
| | GD62586 | Average current graph | Cursor position time |
| | GD62587 | Average current graph | Cursor position time |
| | GD62588 | Average current graph | Cursor position time |
| | GD62589 | Average current graph | Cursor position time |
| | GD62590 | Average current graph | Beginning position time |
| | GD62591 | Average current graph | Beginning position time |
| | GD62592 | Average current graph | Beginning position time |
| | GD62593 | Average current graph | Beginning position time |
| | GD62594 | Average current graph | End position time |
| | GD62595 | Average current graph | End position time |
| | GD62596 | Average current graph | End position time |
| | GD62597 | Average current graph | End position time |
| | GD62598 | Average current graph | Cursor position device value |
| | GD62600 | Logging timer parameter | |
| | GD62601 | Phase wire system Writing work device | |
| | GD62602 | Type of current sensor Writing work device | |
| | GD62603 | Current multiplier Writing work device | |
| | GD62604 | Voltage multiplier Writing work device | |
| | GD62605 | Power multiplier Writing work device | |
| | GD62606 | Electric energy multiplier Writing work device | |
| | GD62607 | Parameter writing selection device | |
| | GD62608 | Phase wire system Display device | |
| | GD62609 | Current multiplier Display device | |
| | GD62610 | Voltage multiplier Display device | |
| | GD62611 | Power multiplier Display device | |
| | GD62612 | Electric energy multiplier Display device | |
| | GD62613 | Advanced user alarm Comment No. | |
| | GD62614 | Advanced user alarm Occurrence date | |
| | GD62616 | Advanced user alarm Occurrence time | |
| | GD62618 | Advanced user alarm Restoration date | |
| | GD62620 | Advanced user alarm Restoration time | |
| | GD62622 | Offset device for Selection station No. bit lamp | |
| | GD62623 | Offset device for the name in Station No. selection window | |
| | GD62628 | Primary voltage (line voltage) offset device | |
| | GD62630 | Primary voltage (phase voltage) offset device | |
| | GD62632 | Electric energy graph Display position time specification device (Year & Month) | |
| | GD62633 | Electric energy graph Display position time specification device (Date & Hour) | |
| | GD62634 | Electric energy graph Display position time specification device (Minute & | |

| Type | Device No. | Application |
|------|------------|--|
| | | Second) |
| | GD62635 | Average current graph Display position time specification device (Year & Month) |
| | GD62636 | Average current graph Display position time specification device (Date & Hour) |
| | GD62637 | Average current graph Display position time specification device (Minute & Second) |
| | GD62638 | Alarm display (user) All stations/monitor station display/hide device |
| | GD62640 | Device data transfer ID: 193 External control device |
| | GD62641 | Device data transfer ID: 193 External notification device |
| | GD62642 | Device data transfer ID: 194 External control device |
| | GD62643 | Device data transfer ID: 194 External notification device |
| | GD62644 | Device data transfer ID: 195 External control device |
| | GD62645 | Device data transfer ID: 195 External notification device |
| | GD62646 | Device data transfer ID: 196 External control device |
| | GD62647 | Device data transfer ID: 196 External notification device |
| | GD62648 | Device data transfer ID: 197 External control device |
| | GD62649 | Device data transfer ID: 197 External notification device |
| | GD62650 | Device data transfer ID: 198 External control device |
| | GD62651 | Device data transfer ID: 198 External notification device |
| | GD62652 | Device data transfer ID: 199 External control device |
| | GD62653 | Device data transfer ID: 199 External notification device |
| | GD62654 | Device data transfer ID: 200 External control device |
| | GD62655 | Device data transfer ID: 200 External notification device |
| | GD62656 | Device data transfer ID: 201 External control device |
| | GD62657 | Device data transfer ID: 201 External notification device |
| | GD62658 | Device data transfer ID: 202 External control device |
| | GD62659 | Device data transfer ID: 202 External notification device |
| | GD62660 | Device data transfer ID: 203 External control device |
| | GD62661 | Device data transfer ID: 203 External notification device |
| | GD62662 | Device data transfer ID: 204 External control device |
| | GD62663 | Device data transfer ID: 204 External notification device |
| | GD62664 | Device data transfer ID: 205 External control device |
| | GD62665 | Device data transfer ID: 205 External notification device |
| | GD62666 | Device data transfer ID: 206 External control device |
| | GD62667 | Device data transfer ID: 206 External notification device |
| | GD62668 | Device data transfer ID: 207 External control device |
| | GD62669 | Device data transfer ID: 207 External notification device |
| | GD62670 | Device data transfer ID: 208 External control device |
| | GD62671 | Device data transfer ID: 208 External notification device |
| | GD62672 | Device data transfer ID: 209 External control device |
| | GD62673 | Device data transfer ID: 209 External notification device |
| | GD62674 | Device data transfer ID: 210 External control device |
| | GD62675 | Device data transfer ID: 210 External notification device |
| | GD62676 | Device data transfer ID: 211 External control device |
| | GD62677 | Device data transfer ID: 211 External notification device |
| | GD62678 | Device data transfer ID: 212 External control device |
| | GD62679 | Device data transfer ID: 212 External notification device |
| | GD62680 | Device data transfer ID: 213 External control device |
| | GD62681 | Device data transfer ID: 213 External notification device |
| | GD62682 | Device data transfer ID: 214 External control device |
| | GD62683 | Device data transfer ID: 214 External notification device |
| | GD62684 | Device data transfer ID: 215 External control device |
| | GD62685 | Device data transfer ID: 215 External notification device |
| | GD62686 | Device data transfer ID: 216 External control device |
| | GD62687 | Device data transfer ID: 216 External notification device |
| | GD62688 | Device data transfer ID: 217 External control device |
| | GD62689 | Device data transfer ID: 217 External notification device |
| Word | GD62690 | Device data transfer ID: 218 External control device |
| Word | GD62691 | Device data transfer ID: 218 External notification device |

| Type | Device No. | Application |
|------|------------------|---|
| | GD62752 | Device data transfer ID: 249 External control device |
| | GD62753 | Device data transfer ID: 249 External notification device |
| | GD62754 | Device data transfer ID: 250 External control device |
| | GD62755 | Device data transfer ID: 250 External notification device |
| | GD62756 | Device data transfer ID: 251 External control device |
| | GD62757 | Device data transfer ID: 251 External notification device |
| | GD62758 | Device data transfer ID: 252 External control device |
| | GD62759 | Device data transfer ID: 252 External notification device |
| | GD62760 | Device data transfer ID: 253 External control device |
| | GD62761 | Device data transfer ID: 253 External notification device |
| | GD62762 | Device data transfer ID: 254 External control device |
| | GD62763 | Device data transfer ID: 254 External notification device |
| | GD62765 | Monitor target station selection switch |
| | GD62766 | Current model storage device (For changing timing of Station No. switching) |
| | GD63990 | Clock digital switch |
| | GD63991 | Clock digital switch |
| | GD63992 | Clock digital switch |
| | GD63993 | Clock digital switch |
| | GD63994 | Clock digital switch |
| | GD63995 | Clock digital switch |
| | GS531, GS532 | Monitor station disconnection (CH1) |
| | GS573 | MODBUS® communication control function (Common) |
| | TMP800 to TMP803 | For project and screen script operation |
| | TMP950 to TMP996 | For clock setting screen object script operation |

5.5 Comment List

| Comment group No. | Comment No. | Base/window screen No. |
|-------------------|---------------|---|
| 254 | No.1 to 319 | B-30011 (Advanced user alarm history ID 30001) |
| 255 | No.1 | B-30001 |
| | No.3, 4 | B-30002 to 30011, B-30020 |
| | No.7 | B-30001 to 30011, B-30020 |
| | No.10 | B-30020, W-30015 |
| | No.11 to 17 | B-30020 |
| | No.18 | B-30002, B-30020 |
| | No.20 to 37 | B-30002 |
| | No.40 | B-30002 |
| | No.47 | B-30001, B-30009 |
| | No.48 | B-30001, B-30010 |
| | No.49 | B-30002, B-30009, B-30010, B-30020 |
| | No.50 | B-30009, B-30010 |
| | No.57 | B-30001 |
| | No.59 | B-30001, B-30011 |
| | No.61, 62 | B-30010 |
| | No.63, 64 | B-30009, B-30010 |
| | No.68 | B-30009 |
| | No.70, 71 | B-30001 to 30011, B-30020 |
| | No.73 | B-30002, B-30020 |
| | No.75, 76 | B-30011 |
| | No.79 | B-30020, W-30016 |
| | No.80 | B-30020, W-30017 |
| | No.81 | B-30020, W-30018 |
| | No.82 | B-30020, W-30019 |
| | No.83 | B-30020, W-30020 |
| | No.84, 85 | B-30020 |
| | No.88 to 94 | W-30003 |
| | No.95 | W-30002 |
| | No.96 | W-30003 |
| | No.98 | W-30001 |
| | No.99 | W-30001, W-30003 |
| | No.101 to 110 | B-30009, B-30010 |
| | No.210 | W-30030, W-30031 |
| | No.211 | W-30031 |
| | No.212 | W-30030, W-30031 |
| | No.213 | W-30031 |
| | No.215, 216 | W-30031, W-30015 to 30020 |
| | No.217 | B-30020 |
| | No.220 | W-30030 |
| | No.221 | W-30031 |
| | No.222 | W-30030 |
| | No.223 | W-30031 |
| | No.300, 301 | W-30031 |
| | No.303 to 309 | W-30015 |
| | No.310 | B-30020, W-30016 |
| | No.311 | W-30016 |
| | No.312 to 313 | W-30019 |
| | No.314 | W-30017, W-30019, W-30020 |
| | No.315 to 317 | W-30017 to 30020 |
| | No.318 | W-30018 to 30020 |
| | No.319 | W-30020 |
| 255 | No.401 | W-30014 |
| | No.402, 403 | W-30014 to 30020 |
| | No.500 | B-30002, B-30009 to 30011, B-30020 |
| | No.501 | B-30001, B-30020 |
| | No.502 to 504 | B-30002, B-30009 to 30011, B-30020 |

5.6 Recipe List

5.6.1 Common Setting

| External notification information | |
|-----------------------------------|---------|
| External control device | GD62554 |
| Recipe No. storage device | GD62555 |
| Record No. storage device | GD62556 |
| External notification device | GD62557 |
| Recipe No. notification device | GD62558 |
| Record No. notification device | GD62559 |

5.6.2 Individual Setting

Recipe No.30001Recipe1

| Item | | Setting |
|----------------|-------------------|------------------------------------|
| Recipe file | Recipe file | Use a recipe file (read and write) |
| | File format | G2P (Binary) |
| | Drive name | A: Standard SD Card |
| | Folder name | Package1 |
| | File name | ARP30001.G2P |
| Trigger device | Write trigger 1 | GD62560.b0 |
| | Read trigger 1 | GD62560.b1 |
| | Record No. device | GD62561 |
| Block number | | |
| Record number | | |
| Block 1 | Device | GS531 |
| | Device type | Signed BIN16 |
| | Point | 2 |
| Block 2 | Device | GD61000 |
| | Device type | String |
| | Point | 20 |
| Block 3 | Device | GD61020 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 4 | Device | GD61040 |

| Item | | Setting |
|----------|-------------|--------------|
| | Device type | String |
| | Point | 20 |
| Block 5 | Device | GD61060 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 6 | Device | GD61080 |
| | Device type | String |
| | Point | 20 |
| Block 7 | Device | GD61100 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 8 | Device | GD61120 |
| | Device type | String |
| | Point | 20 |
| Block 9 | Device | GD61140 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 10 | Device | GD61160 |
| | Device type | String |
| | Point | 20 |
| Block 11 | Device | GD61180 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 12 | Device | GD61200 |
| | Device type | String |
| | Point | 20 |
| Block 13 | Device | GD61220 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 14 | Device | GD61240 |
| | Device type | String |
| | Point | 20 |
| Block 15 | Device | GD61260 |

| Item | | Setting |
|----------|-------------|--------------|
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 16 | Device | GD61280 |
| | Device type | String |
| | Point | 20 |
| Block 17 | Device | GD61300 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 18 | Device | GD61320 |
| | Device type | String |
| | Point | 20 |
| Block 19 | Device | GD61340 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 20 | Device | GD61360 |
| | Device type | String |
| | Point | 20 |
| Block 21 | Device | GD61380 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 22 | Device | GD61400 |
| | Device type | String |
| | Point | 20 |
| Block 23 | Device | GD61420 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 24 | Device | GD61440 |
| | Device type | String |
| | Point | 20 |
| Block 25 | Device | GD61460 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 26 | Device | GD61480 |

| Item | | Setting |
|----------|-------------|--------------|
| | Device type | String |
| | Point | 20 |
| Block 27 | Device | GD61500 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 28 | Device | GD61520 |
| | Device type | String |
| | Point | 20 |
| Block 29 | Device | GD61540 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 30 | Device | GD61560 |
| | Device type | String |
| | Point | 20 |
| Block 31 | Device | GD61580 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 32 | Device | GD61600 |
| | Device type | String |
| | Point | 20 |
| Block 33 | Device | GD61620 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 34 | Device | GD61640 |
| | Device type | String |
| | Point | 20 |
| Block 35 | Device | GD61660 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 36 | Device | GD61680 |
| | Device type | String |
| | Point | 20 |
| Block 37 | Device | GD61700 |

| Item | | Setting |
|----------|-------------|--------------|
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 38 | Device | GD61720 |
| | Device type | String |
| | Point | 20 |
| Block 39 | Device | GD61740 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 40 | Device | GD61760 |
| | Device type | String |
| | Point | 20 |
| Block 41 | Device | GD61780 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 42 | Device | GD61800 |
| | Device type | String |
| | Point | 20 |
| Block 43 | Device | GD61820 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 44 | Device | GD61840 |
| | Device type | String |
| | Point | 20 |
| Block 45 | Device | GD61860 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 46 | Device | GD61880 |
| | Device type | String |
| | Point | 20 |
| Block 47 | Device | GD61900 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 48 | Device | GD61920 |

| Item | | Setting |
|----------|-------------|--------------|
| | Device type | String |
| | Point | 20 |
| Block 49 | Device | GD61940 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 50 | Device | GD61960 |
| | Device type | String |
| | Point | 20 |
| Block 51 | Device | GD61980 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 52 | Device | GD62000 |
| | Device type | String |
| | Point | 20 |
| Block 53 | Device | GD62020 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 54 | Device | GD62040 |
| | Device type | String |
| | Point | 20 |
| Block 56 | Device | GD62060 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 57 | Device | GD62080 |
| | Device type | String |
| | Point | 20 |
| Block 58 | Device | GD62100 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 59 | Device | GD62120 |
| | Device type | String |
| | Point | 20 |
| Block 60 | Device | GD62140 |

| Item | | Setting |
|----------|-------------|--------------|
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 61 | Device | GD62160 |
| | Device type | String |
| | Point | 20 |
| Block 62 | Device | GD62180 |
| | Device type | Signed BIN16 |
| | Point | 20 |
| Block 63 | Device | GD62200 |
| | Device type | String |
| | Point | 20 |
| Block 64 | Device | GD62220 |
| | Device type | Signed BIN16 |
| | Point | 20 |

5.7 Script List

| Item | Settings |
|----------------|--|
| Project script | Specified |
| Screen script | B-30001 to B-30020, W-30030 to W-30031 |
| Object script | B-30002 to B-30020, W-30003 |

5.7.1 Project script

| | | | |
|--|------------------------------|--------------|-------------|
| Script No. | 30001 | Script name | Script30001 |
| Comment | Process at screen activation | | |
| Data type | Signed BIN16 | Trigger type | Rise, GB40 |
| <pre>// A process executed when a screen is activated. // Writing to a parameter [w:GD62561] = 1; // Record No. set([b:GD62560.b0]); [w:GS573] = 2; // Specify the maximum number of readable holding register</pre> | | | |

| | | | |
|---|----------------------------------|--------------|-----------------|
| Script No. | 30002 | Script name | Script30002 |
| Comment | Advanced recipe write completion | | |
| Data type | Signed BIN16 | Trigger type | Rise GD62557.b4 |
| <pre>// A process after completion of advanced recipe writing. // Search the smallest valid Station No. [u32:TMP0800] = [u32:GS531] >> 1; // Shift to right by 1bit because 0bit position is not used. [w:TMP0802] = 1; while([w:TMP0802] < 32) { if(([u32:TMP0800] & 0x00000001) == 0) { [w:GD10] = [w:TMP0802]; break; } [u32:TMP0800] = [u32:TMP0800] >> 1; [w:TMP0802] = [w:TMP0802] + 1; } // Set 1 if all station is invalid. if([w:TMP0802] == 32) { [w:GD10] = 1; } // Start the setting for screen activation set([b:GB61013]); // Clear the flag rst([b:GD62560.b0]);</pre> | | | |

| | | | |
|--|---------------|--------------|--------------------------------|
| Script No. | 30024 | Script name | Script30024 |
| Comment | Logging delay | | |
| Data type | Signed BIN16 | Trigger type | OFF SAMPLING 2 SECONDS GB61011 |
| <pre>// Start the script for logging after waiting two seconds from the screen activation. set([b:GB61011]);</pre> | | | |

| | | | |
|--|-------------------------------|--------------|------------------------------|
| Script No. | 30007 | Script name | Script30007 |
| Comment | Logging trigger OFF(No.30001) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61000 |
| // Off the logging No.30001 (Electric energy (one hour)) trigger. rst([b:GB61000]); | | | |

| | | | |
|--|-------------------------------|--------------|------------------------------|
| Script No. | 30008 | Script name | Script30008 |
| Comment | Logging trigger OFF(No.30002) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61001 |
| //Off the logging No.30002 (Electric energy (1 day)) trigger. rst([b:GB61001]); | | | |

| | | | |
|--|-------------------------------|--------------|------------------------------|
| Script No. | 30009 | Script name | Script30009 |
| Comment | Logging trigger OFF(No.30003) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61002 |
| //Off the logging No.30003 (Electric energy (1 month)) trigger. rst([b:GB61002]); | | | |

| | | | |
|---|-------------------------------|--------------|------------------------------|
| Script No. | 30010 | Script name | Script30010 |
| Comment | Logging trigger OFF(No.30004) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61003 |
| //Off the logging No.30004 (Average current (1 minute)) trigger. rst([b:GB61003]); | | | |

| | | | |
|---|-------------------------------|--------------|------------------------------|
| Script No. | 30011 | Script name | Script30011 |
| Comment | Logging trigger OFF(No.30005) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61004 |
| //Off the logging No.30005 (Average current (30 minutes)) trigger. rst([b:GB61004]); | | | |

| | | | |
|---|-------------------------------|--------------|------------------------------|
| Script No. | 30012 | Script name | Script30012 |
| Comment | Logging trigger OFF(No.30006) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61005 |
| //Off the logging No.30006 (Average current (1 hour)) trigger. rst([b:GB61005]); | | | |

| | | | |
|--|-------------------------------|--------------|------------------------------|
| Script No. | 30013 | Script name | Script30013 |
| Comment | Logging trigger OFF(No.30007) | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61006 |
| //Off the logging No.30007 (Average current (1 day)) trigger. rst([b:GB61006]); | | | |

| | | | |
|--|--|--------------|--------------|
| Script No. | 30014 | Script name | Script30014 |
| Comment | Readout to recipe after input completion | | |
| Data type | Signed BIN16 | Trigger type | Rise GB61009 |
| // Readout to advanced recipe is executed for each object after input. set([b:GD62560.b1]); | | | |

| | | | |
|--|--|--------------|----------------|
| Script No. | 30015 | Script name | Script30015 |
| Comment | Process after advanced recipe readout completion | | |
| Data type | Signed BIN16 | Trigger type | ON, GD62557.b5 |
| <pre>//Trigger OFF after advanced recipe read. //After setting valid/invalid station No., //start the script to clear logging data. if([b:GB61015] == ON) { set([b:GB61014]); rst([b:GB61015]); } rst([b:GD62560.b1]); rst([b:GD62557.b5]);</pre> | | | |

| | | | |
|---|------------------------------------|--------------|------------------------------|
| Script No. | 30018 | Script name | Script30018 |
| Comment | Device data transfer 1 trigger OFF | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GB61018 |
| <pre>// Monitor the trigger for device data transfer periodically and reset. rst([b:GD62640.b0]); rst([b:GD62644.b0]); rst([b:GD62648.b0]); rst([b:GD62652.b0]); rst([b:GD62656.b0]); rst([b:GD62660.b0]); rst([b:GD62664.b0]); rst([b:GD62668.b0]); rst([b:GD62672.b0]); rst([b:GD62676.b0]); rst([b:GD62680.b0]); rst([b:GD62684.b0]); rst([b:GD62688.b0]); rst([b:GD62692.b0]); rst([b:GD62696.b0]); rst([b:GD62700.b0]); rst([b:GD62704.b0]); rst([b:GD62708.b0]); rst([b:GD62712.b0]); rst([b:GD62716.b0]); rst([b:GD62720.b0]); rst([b:GD62724.b0]); rst([b:GD62728.b0]); rst([b:GD62732.b0]); rst([b:GD62736.b0]); rst([b:GD62740.b0]); rst([b:GD62744.b0]); rst([b:GD62748.b0]); rst([b:GD62752.b0]); rst([b:GD62756.b0]); rst([b:GD62760.b0]); rst([b:GD62642.b0]); rst([b:GD62646.b0]); rst([b:GD62650.b0]); rst([b:GD62654.b0]); rst([b:GD62658.b0]); rst([b:GD62662.b0]); rst([b:GD62666.b0]); rst([b:GD62670.b0]); rst([b:GD62674.b0]); rst([b:GD62678.b0]);</pre> | | | |

```

rst([b:GD62682.b0]);
rst([b:GD62686.b0]);
rst([b:GD62690.b0]);
rst([b:GD62694.b0]);
rst([b:GD62698.b0]);
rst([b:GD62702.b0]);
rst([b:GD62706.b0]);
rst([b:GD62710.b0]);
rst([b:GD62714.b0]);
rst([b:GD62718.b0]);
rst([b:GD62722.b0]);
rst([b:GD62726.b0]);
rst([b:GD62730.b0]);
rst([b:GD62734.b0]);
rst([b:GD62738.b0]);
rst([b:GD62742.b0]);
rst([b:GD62746.b0]);
rst([b:GD62750.b0]);
rst([b:GD62754.b0]);
rst([b:GD62758.b0]);
rst([b:GD62762.b0]);

rst([b:GB61018]);

```

| | | | |
|------------|------------------------------------|--------------|---|
| Script No. | 30026 | Script name | Script30026 |
| Comment | Device data transfer 2 trigger OFF | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second GD62642.b0 |

// Monitor the trigger for device data transfer periodically and reset.

```

rst([b:GD62642.b0]);
rst([b:GD62646.b0]);
rst([b:GD62650.b0]);
rst([b:GD62654.b0]);
rst([b:GD62658.b0]);
rst([b:GD62662.b0]);
rst([b:GD62666.b0]);
rst([b:GD62670.b0]);
rst([b:GD62674.b0]);
rst([b:GD62678.b0]);
rst([b:GD62682.b0]);
rst([b:GD62686.b0]);
rst([b:GD62690.b0]);
rst([b:GD62694.b0]);
rst([b:GD62698.b0]);
rst([b:GD62702.b0]);
rst([b:GD62706.b0]);
rst([b:GD62710.b0]);
rst([b:GD62714.b0]);
rst([b:GD62718.b0]);
rst([b:GD62722.b0]);
rst([b:GD62726.b0]);
rst([b:GD62730.b0]);
rst([b:GD62734.b0]);
rst([b:GD62738.b0]);
rst([b:GD62742.b0]);
rst([b:GD62746.b0]);
rst([b:GD62750.b0]);
rst([b:GD62754.b0]);
rst([b:GD62758.b0]);
rst([b:GD62762.b0]);

```

5.7.2 Screen script

Base screen common

| | | | |
|------------|--|--------------|-------------|
| Script No. | 30004 | Script name | Script30004 |
| Comment | Related to logging 1: Start device data transfer | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61011 |

```

//Obtain data for logging by device data transfer.

// Every one minutes
if(((w:GS652] & 0x00FF) == 0) && ([b:GB61017] == OFF))
{
    set([b:GB61017]); // Flag not to execute several times in one second.
    //Every hour
    if(((w:GS652] & 0xFF00) >> 8) == 0)
    {
        //Every day
        if((w:GS651] & 0x00FF) == 0)
        {
            //Every month
            if(((w:GS651] & 0xFF00) >> 8) == 0x0001)
            {

                //Start device data transfer
                [w:GD62600] = 5;

                //Execute device data transfer for only validate stations.
                [u32:TMP0800] = [u32:GS531] >> 1; //Because 0 bit is unused, it is shifted to 1 bit to the right.

                [w:TMP0802] = 1; //The number of loop (also as station No.)
                [w:TMP0803] = 0; //Offset to check model.
                [w:TMP0804] = 0; //Start trigger offset for Device data transfer

                while([w:TMP0802] < 32)
                {
                    if((u32:TMP0800] & 0x00000001) == 0)
                    {
                        [w:TMP0804] = ([w:TMP0802] - 1) * 4;

                        //Model check
                        if([w:GD61020[w:TMP0803]] < 1)
                        {
                            set([b:GD62640.b0[w:TMP0804]]);
                        }else{
                            set([b:GD62642.b0[w:TMP0804]]);
                        }
                    }

                    [u32:TMP0800] = [u32:TMP0800] >> 1;
                    [w:TMP0802] = [w:TMP0802] + 1;
                    [w:TMP0803] = [w:TMP0803] + 40;
                }
                set([b:GB61007]);
                return;
            }

            //Start device data transfer
            [w:GD62600] = 4;

            //Execute device data transfer for only validate stations.
            [u32:TMP0800] = [u32:GS531] >> 1; //Because 0 bit is unused, it is shifted to 1 bit to the right.

            [w:TMP0802] = 1; //The number of loop (also as station No.)
            [w:TMP0803] = 0; //Offset to check model.
            [w:TMP0804] = 0; //Start trigger offset for Device data transfer

```

```

while([w:TMP0802] < 32)
{
    if((([u32:TMP0800] & 0x00000001) == 0))
    {
        [w:TMP0804] = ([w:TMP0802] - 1) * 4;

        //Model check
        if([w:GD61020[w:TMP0803]] < 1)
        {
            set([b:GD62640.b0[w:TMP0804]]);
        }else{
            set([b:GD62642.b0[w:TMP0804]]);
        }
    }

    [u32:TMP0800] = [u32:TMP0800] >> 1;
    [w:TMP0802] = [w:TMP0802] + 1;
    [w:TMP0803] = [w:TMP0803] + 40;
}
set([b:GB61007]);
return;
}

//Start device data transfer
[w:GD62600] = 3;

//Execute device data transfer for only validate stations.
[u32:TMP0800] = [u32:GS531] >> 1;    //Because 0 bit is unused, it is shifted to 1 bit to the right.

[w:TMP0802] = 1;    //The number of loop (also as station No.)
[w:TMP0803] = 0;    //Offset to check model.
[w:TMP0804] = 0;    //Start trigger offset for Device data transfer

while([w:TMP0802] < 32)
{
    if((([u32:TMP0800] & 0x00000001) == 0))
    {
        [w:TMP0804] = ([w:TMP0802] - 1) * 4;

        //Model check
        if([w:GD61020[w:TMP0803]] < 1)
        {
            set([b:GD62640.b0[w:TMP0804]]);
        }else{
            set([b:GD62642.b0[w:TMP0804]]);
        }
    }

    [u32:TMP0800] = [u32:TMP0800] >> 1;
    [w:TMP0802] = [w:TMP0802] + 1;
    [w:TMP0803] = [w:TMP0803] + 40;
}
set([b:GB61007]);
return;

}else{
    //Every 30 minutes
    if((([w:GS652] & 0xFF00) >> 8) == 0x0030)
    {
        //Start device data transfer
        [w:GD62600] = 2;

        //Execute device data transfer for only validate stations.
        [u32:TMP0800] = [u32:GS531] >> 1;    //Because 0 bit is unused, it is shifted to 1 bit to the right.
    }
}

```

```

[w:TMP0802] = 1; //The number of loop (also as station No.)
[w:TMP0803] = 0; //Offset to check model.
[w:TMP0804] = 0; //Start trigger offset for Device data transfer

while([w:TMP0802] < 32)
{
    if([u32:TMP0800] & 0x00000001) == 0)
    {
        [w:TMP0804] = ([w:TMP0802] - 1) * 4;

        //Model check
        if([w:GD61020[w:TMP0803]] < 1)
        {
            set([b:GD62640.b0[w:TMP0804]]);
        }else{
            set([b:GD62642.b0[w:TMP0804]]);
        }
    }

    [u32:TMP0800] = [u32:TMP0800] >> 1;
    [w:TMP0802] = [w:TMP0802] + 1;
    [w:TMP0803] = [w:TMP0803] + 40;
}
set([b:GB61007]);
return;
}

//Start device data transfer
[w:GD62600] = 1;

//Execute device data transfer for only validate stations.
[u32:TMP0800] = [u32:GS531] >> 1; //Because 0 bit is unused, it is shifted to 1 bit to the right.

[w:TMP0802] = 1; //The number of loop (also as station No.)
[w:TMP0803] = 0; //Offset to check model.
[w:TMP0804] = 0; //Start trigger offset for Device data transfer

while([w:TMP0802] < 32)
{
    if([u32:TMP0800] & 0x00000001) == 0)
    {
        [w:TMP0804] = ([w:TMP0802] - 1) * 4;

        //Model check
        if([w:GD61020[w:TMP0803]] < 1)
        {
            set([b:GD62640.b0[w:TMP0804]]);
        }else{
            set([b:GD62642.b0[w:TMP0804]]);
        }
    }

    [u32:TMP0800] = [u32:TMP0800] >> 1;
    [w:TMP0802] = [w:TMP0802] + 1;
    [w:TMP0803] = [w:TMP0803] + 40;
}
set([b:GB61007]);
return;
}else{
    if([w:GS652] & 0x00FF) != 0) && ([b:GB61017] == ON))
    {
        rst([b:GB61017]); //Reset the flag after 00 sec.
    }
}

```

| | | | |
|---|---|--------------|-------------|
| <pre> } } </pre> | | | |
| Script No. | 30005 | Script name | Script30005 |
| Comment | Related to logging 2: Average current calculation | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61007 |
| <pre> //Calculate Average current data device-data-transferred. //Start after completion of all device data transfer. if([u32:GS531] == (~[u32:GD62562] - 1)) { [w:TMP0800] = 0; [w:TMP0801] = 0; [w:TMP0802] = 0; while([w:TMP0800] < 31) { if([w:GD61020[w:TMP0801]] == 0) { //EMU4 switch([s16:GD62457[w:TMP0800]]) { case -3: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]] * 0.001; break; case -2: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]] * 0.01; break; case -1: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]] * 0.1; break; case 0: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]]; break; } } }else{ //ME110 switch([w:GD61025[w:TMP0801]]) { case 1: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]] * 0.01 ; break; case 2: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]] * 0.1; break; case 3: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]]; break; case 4: [flt:GD62302[w:TMP0802]] = [s16:GD62426[w:TMP0800]] * 10; break; } } [w:TMP0800] = [w:TMP0800] + 1; [w:TMP0801] = [w:TMP0801] + 25; [w:TMP0802] = [w:TMP0802] + 2; } //Select the logging type. switch([w:GD62600]) { </pre> | | | |

```

case 1:  set([b:GB61003]); //Every one minutes
        break;

case 2:  set([b:GB61003]); //Every one minutes
        set([b:GB61004]); //Every 30 minutes
        break;

case 3:  set([b:GB61003]); //Every one minutes
        set([b:GB61004]); //Every 30 minutes
        set([b:GB61005]); //Every hour
        break;

case 4:  set([b:GB61003]); //Every one minutes
        set([b:GB61004]); //Every 30 minutes
        set([b:GB61005]); //Every hour
        set([b:GB61006]); //Every day
        break;

case 5:  set([b:GB61003]); //Every one minutes
        set([b:GB61004]); //Every 30 minutes
        set([b:GB61005]); //Every hour
        set([b:GB61006]); // Every day
        break;

default:  break;
}

[u32:GD62562] = 0;

set([b:GB61008]); //Operation script for electric energy.

rst([b:GB61007]);
}

```

| | | | |
|------------|---|--------------|-------------|
| Script No. | 30006 | Script name | Script30006 |
| Comment | Related to logging 2: Electric energy calculation | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61008 |

```
//Calculate Electric energy data device-data-transferred.//
```

```

[w:TMP0800] = 0;
[w:TMP0801] = 0;
[w:TMP0802] = 0;

//Every 1 hour/day/month logging for electric energy.
if([w:GD62600] >= 3)
{
    while([w:TMP0800] <= 31)
    {
        if([w:GD61020[w:TMP0801]] == 0)
        {
            //EMU4
            switch([w:GD62488[w:TMP0800]])
            {
                case -2: [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 0.01;
                        break;

                case -1: [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 0.1;
                        break;

                case 0:  [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]];
                        break;

                case 1:  [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 10;
                        break;
            }
        }
    }
}

```

```

        case 2:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 100;
                   break;
    }
} else {
    //ME110
    switch([w:GD61028[w:TMP0801]])
    {
        case 1:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 0.01 ;
                   break;

        case 2:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 0.1;
                   break;

        case 3:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]];
                   break;

        case 4:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 10;
                   break;

        case 5:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 100;
                   break;

        case 6:    [flt:GD62240[w:TMP0802]] = [s32:GD62364[w:TMP0800]] * 1000;
                   break;
    }
}

[w:TMP0800] = [w:TMP0800] + 1;
[w:TMP0801] = [w:TMP0801] + 25;
[w:TMP0802] = [w:TMP0802] + 2;
}

//Select the logging type.
switch([w:GD62600])
{
    case 3:    set([b:GB61000]); //Every hour
               break;

    case 4:    set([b:GB61000]); //Every hour
               set([b:GB61001]); // Every day
               break;

    case 5:    set([b:GB61000]); //Every hour
               set([b:GB61001]); // Every day
               set([b:GB61002]); // Every month
               break;

    default:   break;
}
}

set([b:GB61018]);

rst([b:GB61008]);

```

| | | | |
|------------|--------------------------------|--------------|-------------|
| Script No. | 30021 | Script name | Script30021 |
| Comment | Trans each param to work area. | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61013 |

```

//Transfer the value from recipe to work area of each parameter.

//Current multiplying factor
[w:GD62609] = [w:GD61025[w:GD62552]];

```



```

//Voltage multiplying factor
[w:GD62610] = [w:GD61026[w:GD62552]];

//Electric power multiplying factor.
[w:GD62611] = [w:GD61027[w:GD62552]];

//Electric energy multiplying factor.
[w:GD62612] = [w:GD61028[w:GD62552]];

//Graph maximum value (electric energy).
[flt:GD62566] = [flt:GD61031[w:GD62552]];

//Maximum graph value (total current)
[flt:GD62584] = ([1-248:s32:400520] * 0.1) * 1.1;

//Turn ON the internal devices (GB61021 or higher) according to the validated station No.
[u32:TMP0800] = [u32:GS531] >> 1; //Because 0 bit is unused, it is shifted to 1 bit to the right.
[w:TMP0802] = 0;
while([w:TMP0802] < 31)
{
    if([u32:TMP0800] & 0x00000001) == 1)
    {
        set([b:GB61021[w:TMP0802]]);
    }

    [u32:TMP0800] = [u32:TMP0800] >> 1;
    [w:TMP0802] = [w:TMP0802] + 1;
}

//Turn ON the bit lamp to display in Station No. selection window.
[w:TMP0803] = [w:GD10] - 1;
set([b:GB61060[w:TMP0803]]);

```

rst([b:GB61013]);

| | | | |
|------------|---------------------------------|--------------|-------------|
| Script No. | 30027 | Script name | Script30027 |
| Comment | PC(LV, PV) Disp/nonDisp setting | | |
| Data type | Signed BIN16 | Trigger type | Always |

//Change the display state of the parameters of primary current (line voltage, phase voltage) according to the conditions.

```

if([1-248:w:400513] == 4)
{
    set([b:GB61016]);
    [w:GD62628] = 2; //Line voltage offset
    [w:GD62630] = 0; //Phase voltage offset
}else{
    rst([b:GB61016]);
    [w:GD62628] = 0; //Line voltage offset
    [w:GD62630] = 2; //Phase voltage offset
}

```

| | | | |
|------------|----------------------------------|--------------|----------------------|
| Script No. | 30039 | Script name | Script30039 |
| Comment | //Change timing of St No. switch | | |
| Data type | Signed BIN16 | Trigger type | ON Sampling 1 second |

[b:GB61093]=OFF;

[s16:GD63996]=[s16:GD61020[s16:GD62552]];

//Change the timing for station No. switching for actual access according to the model.
//The following processes are designed for MEU4.
if([s16:GD61020[s16:GD62552]]==1){
 switch([w:TMP0800])

```

{
  case 1:  [w:GD10] = 1;    //Station No.1
           break;

  case 2:  [w:GD10] = 2;    //Station No.2
           break;

  case 3:  [w:GD10] = 3;    //Station No.3
           break;

  case 4:  [w:GD10] = 4;    //Station No.4
           break;

  case 5:  [w:GD10] = 5;    //Station No.5
           break;

  case 6:  [w:GD10] = 6;    //Station No.6
           break;

  case 7:  [w:GD10] = 7;    //Station No.7
           break;

  case 8:  [w:GD10] = 8;    //Station No.8
           break;

  case 9:  [w:GD10] = 9;    //Station No.9
           break;

  case 10: [w:GD10] = 10;    //Station No.10
           break;

  case 11: [w:GD10] = 11;    //Station No.11
           break;

  case 12: [w:GD10] = 12;    //Station No.12
           break;

  case 13: [w:GD10] = 13;    //Station No.13
           break;

  case 14: [w:GD10] = 14;    //Station No.14
           break;

  case 15: [w:GD10] = 15;    //Station No.15
           break;

  case 16: [w:GD10] = 16;    //Station No.16
           break;

  case 17: [w:GD10] = 17;    //Station No.17
           break;

  case 18: [w:GD10] = 18;    //Station No.18
           break;

  case 19: [w:GD10] = 19;    //Station No.19
           break;

  case 20: [w:GD10] = 20;    //Station No.20
           break;

  case 21: [w:GD10] = 21;    //Station No.21
           break;
}

```

```

case 22:  [w:GD10] = 22;      //Station No.22
         break;

case 23:  [w:GD10] = 23;      //Station No.23
         break;

case 24:  [w:GD10] = 24;      //Station No.24
         break;

case 25:  [w:GD10] = 25;      //Station No.25
         break;

case 26:  [w:GD10] = 26;      //Station No.26
         break;

case 27:  [w:GD10] = 27;      //Station No.27
         break;

case 28:  [w:GD10] = 28;      //Station No.28
         break;

case 29:  [w:GD10] = 29;      //Station No.29
         break;

case 30:  [w:GD10] = 30;      //Station No.30
         break;

case 31:  [w:GD10] = 31;      //Station No.31
         break;

default:  break;
}
}

```

Base screen 30001

| | | | |
|---|------------------------------------|--------------|-------------|
| Script No. | 30025 | Script name | Script30025 |
| Comment | Invalid station logging data clear | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61014 |
| <pre> //Clear logging data of invalid station No. [u32:TMP0800] = [u32:GS531] >> 1; //Because 0 bit is unused, it is shifted to 1 bit to the right. [w:TMP0802] = 1; [w:TMP0803] = 0; while([w:TMP0802] < 32) { if([u32:TMP0800] & 0x00000001) == 1) { [flt:GD62240[w:TMP0803]] = 0; //Electric energy [flt:GD62302[w:TMP0803]] = 0; //Average current } [u32:TMP0800] = [u32:TMP0800] >> 1; [w:TMP0802] = [w:TMP0802] + 1; [w:TMP0803] = [w:TMP0803] + 2; } rst([b:GB61014]); </pre> | | | |

Base screen 30009

| | | | |
|---|--------------------------------|--------------|-------------|
| Script No. | 30029 | Script name | Script30029 |
| Comment | Obtain Present Time | | |
| Data type | Signed BIN16 | Trigger type | Rise GB40 |
| //Store year, month, day, hour, minute, sec when screen is displayed. [w:GD62632]=[w:GS650]; [w:GD62633]=[w:GS651]; [w:GD62634]=[w:GS652]; | | | |
| Script No. | 30017 | Script name | Script30017 |
| Comment | SaveElectricErgGraphUpperLimit | | |
| Data type | Signed BIN16 | Trigger type | Rise GB40 |
| //Save the upper limit of electric energy graph [flt:GD61031[w:GD62552]] = [flt:GD62566]; set([b:GD62560.b1]); | | | |

Base screen 30010

| | | | |
|---|--------------------------------|--------------|--------------|
| Script No. | 30028 | Script name | Script30028 |
| Comment | Obtain Present Time | | |
| Data type | Signed BIN16 | Trigger type | Rise GB40 |
| //Store year, month, day, hour, minute, sec when screen is displayed. [w:GD62635]=[w:GS650]; [w:GD62636]=[w:GS651]; [w:GD62637]=[w:GS652]; | | | |
| Script No. | 30016 | Script name | Script30016 |
| Comment | SaveAvg.CurrentGraphUpperLimit | | |
| Data type | Signed BIN16 | Trigger type | Rise GB61010 |
| //Save the upper limit of average current graph [flt:GD61033[w:GD62552]] = [flt:GD62584]; set([b:GD62560.b1]); | | | |

Base screen 30020

| | | | |
|---|-----------------|--------------|-------------|
| Script No. | 30020 | Script name | Script30020 |
| Comment | Parameter write | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61012 |
| //Write parameter in EcoMonitor according to the flag. switch([w:GD62607]) { case 1: //Write the value in phase wire device (400513). [1-248:w:400513] = [w:GD62601]; break; case 2: //Write the value in sensor type device (400532). [1-248:w:400532] = [w:GD62602]; break; case 3: //Write the value in current multiplying factor (GD61011). [w:GD61025[w:GD62552]] = [w:GD62603]; [w:GD62609] = [w:GD62603]; break; case 4: //Write the value in voltage multiplying factor (GD61012). [w:GD61026[w:GD62552]] = [w:GD62604]; [w:GD62610] = [w:GD62604]; break; | | | |

```

case 5: //Write the value in electric power multiplying factor device (GD61013).
    [w:GD61027[w:GD62552]] = [w:GD62605];
    [w:GD62611] = [w:GD62605];
    break;

case 6: //Write the value in electric energy multiplying factor device (GD61014).
    [w:GD61028[w:GD62552]] = [w:GD62606];
    [w:GD62612] = [w:GD62606];
    break;

default: break;
}

set([b:GD62560.b1]);
rst([b:GB61012]);

```

Window screen 30030

| | | | |
|---|----------------------------|--------------|--------------|
| Script No. | 30037 | Script name | Script30037 |
| Comment | W-30008 When window opened | | |
| Data type | Signed BIN16 | Trigger type | Rise GB40 |
| <pre> //Reflect each setting when window opened. [w:GD62622] = 0; //Bit lamp offset [w:GD62623] = 0; //Name offset rst([b:GB61092]); //Flag not to display station No.32. //Trigger OFF only validated station No. between 1 and 8. //OFF=available ON=not available [w:TMP0800] = 0; while([w:TMP0800] < 8) { if([b:GB61021[w:TMP800]] == OFF) { rst([b:GB61052[w:TMP0800]]); //trigger OFF }else{ set([b:GB61052[w:TMP0800]]); //trigger ON } [w:TMP0800] = [w:TMP0800] + 1; } </pre> | | | |
| Script No. | 30034 | Script name | Script30034 |
| Comment | Scroll | | |
| Data type | Signed BIN16 | Trigger type | Rise GB61019 |
| <pre> //Scroll the screen. //Set bit lamp offset //Make scrolling loop if([w:GD62622] >= 8) { [w:GD62622] = [w:GD62622] - 8; }else{ [w:GD62622] = 24; } //Set name offset [w:GD62623] = [w:GD62622] * 40; //Trigger OFF only displayed and validated station No. //OFF=available ON=not available [w:TMP0800] = [w:GD62622]; [w:TMP0801] = 0; </pre> | | | |

```

while([w:TMP0801] < 8)
{
  if([w:TMP0800] != 31)
  {
    rst([b:GB61092]);
    if([b:GB61021[w:TMP800]] == OFF)
    {
      rst([b:GB61052[w:TMP0801]]); //trigger OFF
    }else{
      set([b:GB61052[w:TMP0801]]); //trigger ON
    }
  }else{
    set([b:GB61092]);
  }

  [w:TMP0800] = [w:TMP0800] + 1;
  [w:TMP0801] = [w:TMP0801] + 1;
}

```

| | | | |
|------------|--------------|--------------|--------------|
| Script No. | 30035 | Script name | Script30035 |
| Comment | Scroll | | |
| Data type | Signed BIN16 | Trigger type | Rise GB61020 |

//Scroll the screen.

```

//Set bit lamp offset
//Make scrolling loop
if([w:GD62622] < 24)
{
  [w:GD62622] = [w:GD62622] + 8;
}else{
  [w:GD62622] = 0;
}

```

```

//Set name offset
[w:GD62623] = [w:GD62622] * 40;

```

```

//Trigger OFF only displayed and validated stations No.
//OFF=available ON=not available
[w:TMP0800] = [w:GD62622];
[w:TMP0801] = 0;
while([w:TMP0801] < 8)
{
  if([w:TMP0800] != 31)
  {
    rst([b:GB61092]);

    if([b:GB61021[w:TMP800]] == OFF)
    {
      rst([b:GB61052[w:TMP0801]]); //trigger OFF
    }else{
      set([b:GB61052[w:TMP0801]]); //trigger ON
    }
  }else{
    set([b:GB61092]);
  }

  [w:TMP0800] = [w:TMP0800] + 1;
  [w:TMP0801] = [w:TMP0801] + 1;
}

```

| | | | |
|------------|-----------------------|--------------|-------------|
| Script No. | 30036 | Script name | Script30036 |
| Comment | Station No. switching | | |
| Data type | Signed BIN16 | Trigger type | ON, GB61091 |

```

//Switch station No.
//Do the necessary settings to switch the station No. as well.

[w:TMP0800] = [w:GD62765] + [w:GD62622]; //Calculate selected station No.
[w:TMP0801] = 0;

//Turn ON the bit according to the selected station No.
while([w:TMP0801] < 31)
{
    if([w:TMP0801] + 1 == [w:TMP0800])
    {
        set([b:GB61060[w:TMP0801]]);
    }else{
        rst([b:GB61060[w:TMP0801]]);
    }
    [w:TMP0801] = [w:TMP0801] + 1;
}

rst([b:GB61091]);

```

| | | | |
|------------|-----------------------|--------------|-----------------------|
| Script No. | 30038 | Script name | Script30038 |
| Comment | Station No. switching | | |
| Data type | Signed BIN16 | Trigger type | When closing a screen |

```

//Process according to the selected station No.
[w:TMP0801] = 0;

while([w:TMP0801] < 31)
{
    if([b:GB61060[w:TMP0801]] == ON)
    {
        [w:TMP0800] = [w:TMP0801] + 1;
        break;
    }
    [w:TMP0801] = [w:TMP0801] + 1;
}

//Set the offset value according to station No.
switch([w:TMP0800])
{
    case 1:    [w:GD62552] = 0;
               [w:GD62565] = 0;
               break;

    case 2:    [w:GD62552] = 40;
               [w:GD62565] = 2;
               break;

    case 3:    [w:GD62552] = 80;
               [w:GD62565] = 4;
               break;

    case 4:    [w:GD62552] = 120;
               [w:GD62565] = 6;
               break;

    case 5:    [w:GD62552] = 160;
               [w:GD62565] = 8;
               break;

    case 6:    [w:GD62552] = 200;
               [w:GD62565] = 10;
               break;

    case 7:    [w:GD62552] = 240;
               [w:GD62565] = 12;

```

```

break;

case 8:    [w:GD62552] = 280;
          [w:GD62565] = 14;
          break;

case 9:    [w:GD62552] = 320;
          [w:GD62565] = 16;
          break;

case 10:   [w:GD62552] = 360;
          [w:GD62565] = 18;
          break;

case 11:   [w:GD62552] = 400;
          [w:GD62565] = 20;
          break;

case 12:   [w:GD62552] = 440;
          [w:GD62565] = 22;
          break;

case 13:   [w:GD62552] = 480;
          [w:GD62565] = 24;
          break;

case 14:   [w:GD62552] = 520;
          [w:GD62565] = 26;
          break;

case 15:   [w:GD62552] = 560;
          [w:GD62565] = 28;
          break;

case 16:   [w:GD62552] = 600;
          [w:GD62565] = 30;
          break;

case 17:   [w:GD62552] = 640;
          [w:GD62565] = 32;
          break;

case 18:   [w:GD62552] = 680;
          [w:GD62565] = 34;
          break;

case 19:   [w:GD62552] = 720;
          [w:GD62565] = 36;
          break;

case 20:   [w:GD62552] = 760;
          [w:GD62565] = 38;
          break;

case 21:   [w:GD62552] = 800;
          [w:GD62565] = 40;
          break;

case 22:   [w:GD62552] = 840;
          [w:GD62565] = 42;
          break;

case 23:   [w:GD62552] = 880;
          [w:GD62565] = 44;

```



```

        break;

case 24:  [w:GD62552] = 920;
         [w:GD62565] = 46;
         break;

case 25:  [w:GD62552] = 960;
         [w:GD62565] = 48;
         break;

case 26:  [w:GD62552] = 1000;
         [w:GD62565] = 50;
         break;

case 27:  [w:GD62552] = 1040;
         [w:GD62565] = 52;
         break;

case 28:  [w:GD62552] = 1080;
         [w:GD62565] = 54;
         break;

case 29:  [w:GD62552] = 1120;
         [w:GD62565] = 56;
         break;

case 30:  [w:GD62552] = 1160;
         [w:GD62565] = 58;
         break;

case 31:  [w:GD62552] = 1200;
         [w:GD62565] = 60;
         break;

default:  break;
}

//Change the timing for station No. switching for actual access according to the model.
//The following processes are designed for ME110.
if([s16:GD61020[s16:GD62552]]==0){
    switch([w:TMP0800])
    {
        case 1:  [w:GD10] = 1;      //Station No.1
                 break;

        case 2:  [w:GD10] = 2;      //Station No.2
                 break;

        case 3:  [w:GD10] = 3;      //Station No.3
                 break;

        case 4:  [w:GD10] = 4;      //Station No.4
                 break;

        case 5:  [w:GD10] = 5;      //Station No.5
                 break;

        case 6:  [w:GD10] = 6;      //Station No.6
                 break;

        case 7:  [w:GD10] = 7;      //Station No.7
                 break;

        case 8:  [w:GD10] = 8;      //Station No.8
    }
}

```

```

        break;

    case 9:    [w:GD10] = 9;    //Station No.9
        break;

    case 10:   [w:GD10] = 10;   //Station No.10
        break;

    case 11:   [w:GD10] = 11;   //Station No.11
        break;

    case 12:   [w:GD10] = 12;   //Station No.12
        break;

    case 13:   [w:GD10] = 13;   //Station No.13
        break;

    case 14:   [w:GD10] = 14;   //Station No.14
        break;

    case 15:   [w:GD10] = 15;   //Station No.15
        break;

    case 16:   [w:GD10] = 16;   //Station No.16
        break;

    case 17:   [w:GD10] = 17;   //Station No.17
        break;

    case 18:   [w:GD10] = 18;   //Station No.18
        break;

    case 19:   [w:GD10] = 19;   //Station No.19
        break;

    case 20:   [w:GD10] = 20;   //Station No.20
        break;

    case 21:   [w:GD10] = 21;   //Station No.21
        break;

    case 22:   [w:GD10] = 22;   //Station No.22
        break;

    case 23:   [w:GD10] = 23;   //Station No.23
        break;

    case 24:   [w:GD10] = 24;   //Station No.24
        break;

    case 25:   [w:GD10] = 25;   //Station No.25
        break;

    case 26:   [w:GD10] = 26;   //Station No.26
        break;

    case 27:   [w:GD10] = 27;   //Station No.27
        break;

    case 28:   [w:GD10] = 28;   //Station No.28
        break;

    case 29:   [w:GD10] = 29;   //Station No.29
        break;

```

```

        case 30:  [w:GD10] = 30;      //Station No.30
                break;

        case 31:  [w:GD10] = 31;      //Station No.31
                break;

        default:  break;
    }
}

//Model
[w:GD62553] = [w:GD61020[w:GD62552]];

//Current multiplying factor
[w:GD62609] = [w:GD61025[w:GD62552]];

//Voltage multiplying factor
[w:GD62610] = [w:GD61026[w:GD62552]];

//Electric power multiplying factor.
[w:GD62611] = [w:GD61027[w:GD62552]];

//Electric energy multiplying factor.
[w:GD62612] = [w:GD61028[w:GD62552]];

//Electric energy Graph upper limit
[flt:GD62566] = [flt:GD61031[w:GD62552]];

//Average current Graph upper limit
[flt:GD62584] = [flt:GD61033[w:GD62552]];

//Added to adjust the timing of trigger for the object script.
set([b:GB61093]);

```

Window screen 30031

| | | | |
|--|----------------------------|--------------|-------------|
| Script No. | 30033 | Script name | Script30033 |
| Comment | W-30004 When window opened | | |
| Data type | Signed BIN16 | Trigger type | Rise GB40 |
| <p>Reflect each setting when window opened.</p> <p>[w:GD62622] = 0; //Bit lamp offset [w:GD62623] = 0; //Name and model offset rst([b:GB61092]); //Flag not to display the station No.32.</p> <p>//Trigger OFF only validated station No. between 1 and 8. //OFF=available ON=not available [w:TMP0800] = 0; while([w:TMP0800] < 8) { if([b:GB61021[w:TMP800]] == OFF) { rst([b:GB61052[w:TMP0800]]); //trigger OFF }else{ set([b:GB61052[w:TMP0800]]); //trigger ON } [w:TMP0800] = [w:TMP0800] + 1; } </p> | | | |

| | | | |
|--|--------------|--------------|--------------|
| Script No. | 30031 | Script name | Script30031 |
| Comment | Scroll | | |
| Data type | Signed BIN16 | Trigger type | Rise GB61019 |
| <pre>//Scroll the screen. //Set bit lamp offset //Make scrolling loop if([w:GD62622] >= 8) { [w:GD62622] = [w:GD62622] - 8; }else{ [w:GD62622] = 24; } //Set name offset [w:GD62623] = [w:GD62622] * 40; //Trigger OFF only displayed and validated stations. //OFF=available ON=not available [w:TMP0800] = [w:GD62622]; [w:TMP0801] = 0; while([w:TMP0801] < 8) { if([w:TMP0800] != 31) { rst([b:GB61092]); if([b:GB61021[w:TMP800]] == OFF) { rst([b:GB61052[w:TMP0801]]); //trigger OFF }else{ set([b:GB61052[w:TMP0801]]); //trigger ON } }else{ set([b:GB61092]); } [w:TMP0800] = [w:TMP0800] + 1; [w:TMP0801] = [w:TMP0801] + 1; } }</pre> | | | |
| Script No. | 30032 | Script name | Script30032 |
| Comment | Scroll | | |
| Data type | Signed BIN16 | Trigger type | Rise GB61020 |
| <pre>//Scroll //Set bit lamp offset //Make scrolling loop if([w:GD62622] < 24) { [w:GD62622] = [w:GD62622] + 8; }else{ [w:GD62622] = 0; } //Set name offset [w:GD62623] = [w:GD62622] * 40; //Trigger OFF only displayed and validated stations. //OFF=available ON=not available [w:TMP0800] = [w:GD62622]; [w:TMP0801] = 0; while([w:TMP0801] < 8) { if([w:TMP0800] != 31) { </pre> | | | |

```

    rst([b:GB61092]);
    if([b:GB61021[w:TMP080]] == OFF)
    {
        rst([b:GB61052[w:TMP0801]]); //trigger OFF
    }else{
        set([b:GB61052[w:TMP0801]]); //trigger ON
    }
}
}
}
[w:TMP0800] = [w:TMP0800] + 1;
[w:TMP0801] = [w:TMP0801] + 1;
}

```

| | | | |
|------------|---------------------|--------------|-----------------------|
| Script No. | 30030 | Script name | Script30030 |
| Comment | W-30004 save recipe | | |
| Data type | Signed BIN16 | Trigger type | When closing a screen |

//Save setting results which were set in W-30004 in recipe.

```

[w:TMP0800] = 0;
[u32:TMP0801] = [u32:GS531]; //Store the latest current station block.

```

//Reflect ON/OFF status which were set in W-33004 to the word device.
while([w:TMP0800] < 32)

```

{
    if([b:GB61021[w:TMP0800]] == ON)
    {
        [u32:TMP0803] = 0x00000002 << [w:TMP0800] ;
        [u32:TMP0801] = [u32:TMP0801] | [u32:TMP0803];
    }else{
        [u32:TMP0803] = 0x00000002 << [w:TMP0800] ;
        [u32:TMP0801] = [u32:TMP0801] & (~[u32:TMP0803]);
    }

    [w:TMP0800] = [w:TMP0800] + 1;
}

```

[u32:GS531] = [u32:TMP0801]; //Set GS531 to the reflected result.

```

set([b:GB61015]);
set([b:GD62560.b1]); //Read recipe

```

5.7.3 Object script

Base screen 30002

| | | | |
|----------------|-------------------------------------|--------------|--------|
| Object (name) | Numerical display (Current 1_value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

// Calculate the position of the decimal point.

```

if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400755])
    {
        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;
    }
}

```

```

        case 0:    my.decimal_point = 0;
                  break;

    }
} else {
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
                break;

        default: my.decimal_point = 2;
                break;
    }
}
redraw_object();

```

| | | | |
|----------------|-------------------------------------|--------------|--------|
| Object (name) | Numerical display (Current 2_value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400755])
    {
        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:    my.decimal_point = 0;
                  break;

    }
} else {
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
    }
}

```

| | | | |
|--|---|--------------|--------|
| <pre> break; default: my.decimal_point = 2; break; } } redraw_object(); </pre> | | | |
| Object (name) | Numerical display (Current 3 _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |
| <pre> // Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400755]) { case -3: my.decimal_point = 3; break; case -2: my.decimal_point = 2; break; case -1: my.decimal_point = 1; break; case 0: my.decimal_point = 0; break; } }else{ //ME110 switch([s16:GD61025[s16:GD62552]]) { case 1: my.decimal_point = 2; break; case 2: my.decimal_point = 1; break; case 3: my.decimal_point = 0; break; case 4: \$V = \$\$ * 10; my.decimal_point = 0; break; default: my.decimal_point = 2; break; } } redraw_object(); </pre> | | | |
| Object (name) | Numerical display (Current demand 1 _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |
| <pre> // Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400755]) { </pre> | | | |

```

        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

    }

}else{
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
                break;

        default: my.decimal_point = 2;
                break;
    }
}

redraw_object();// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400755])
    {
        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

    }

}else{
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;

```


| | | | |
|---|---|--------------|--------|
| <pre> break; case 3: my.decimal_point = 0; break; case 4: \$V = \$\$ * 10; my.decimal_point = 0; break; default: my.decimal_point = 2; break; } } redraw_object(); </pre> | | | |
| Object (name) | Numerical display (Current demand 2 _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |
| <pre> // Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400755]) { case -3: my.decimal_point = 3; break; case -2: my.decimal_point = 2; break; case -1: my.decimal_point = 1; break; case 0: my.decimal_point = 0; break; } }else{ //ME110 switch([s16:GD61025[s16:GD62552]]) { case 1: my.decimal_point = 2; break; case 2: my.decimal_point = 1; break; case 3: my.decimal_point = 0; break; case 4: \$V = \$\$ * 10; my.decimal_point = 0; break; default: my.decimal_point = 2; break; } } redraw_object(); </pre> | | | |
| Object (name) | Numerical display (Current demand 3 _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400755])
    {
        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

    }

}else{
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
                break;

        default: my.decimal_point = 2;
                break;

    }

}

redraw_object();

```

| | | | |
|----------------|--|--------------|--------|
| Object (name) | Numerical display (1 to 2 line voltage _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400756])
    {
        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

    }

}else{
    //ME110
    switch([s16:GD61026[s16:GD62552]])
    {

```

```

        case 1: my.decimal_point = 1;
            break;

        case 2: my.decimal_point = 0;
            break;

        case 3: $V = $$ * 10;
            my.decimal_point = 0;
            break;

        case 4: $V = $$ * 100;
            my.decimal_point = 0;
            break;

        default: $V = $$ * 10;
            my.decimal_point = 0;
            break;
    }
}

    redraw_object();

```

| | | | |
|----------------|--|--------------|--------|
| Object (name) | Numerical display (2 to 3 line voltage _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400756])
    {
        case -1: my.decimal_point = 1;
            break;

        case 0: my.decimal_point = 0;
            break;
    }
}
}else{
    //ME110
    switch([s16:GD61026[s16:GD62552]])
    {
        case 1: my.decimal_point = 1;
            break;

        case 2: my.decimal_point = 0;
            break;

        case 3: $V = $$ * 10;
            my.decimal_point = 0;
            break;

        case 4: $V = $$ * 100;
            my.decimal_point = 0;
            break;

        default: $V = $$ * 10;
            my.decimal_point = 0;
            break;
    }
}
redraw_object();

```

| | | | |
|---|---|--------------|--------|
| Object (name) | Numerical display (3 to 1line voltage _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |
| <pre>// Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400756]) { case -1: my.decimal_point = 1; break; case 0: my.decimal_point = 0; break; } }else{ //ME110 switch([s16:GD61026[s16:GD62552]]) { case 1: my.decimal_point = 1; break; case 2: my.decimal_point = 0; break; case 3: \$V = \$\$ * 10; my.decimal_point = 0; break; case 4: \$V = \$\$ * 100; my.decimal_point = 0; break; default: \$V = \$\$ * 10; my.decimal_point = 0; break; } } redraw_object();</pre> | | | |
| Object (name) | Numerical display (W_value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |
| <pre>// Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400757]) { case -3: my.decimal_point = 3; break; case -2: my.decimal_point = 2; break; case -1: my.decimal_point = 1; break; case 0: my.decimal_point = 0; break; case 1: \$V = \$\$ * 10;</pre> | | | |

```

        my.decimal_point = 0;
        break;
    }
}
else{
    //ME110
    switch([s16:GD61027[s16:GD62552]])
    {
        case 1: my.decimal_point = 4;
            break;

        case 2: my.decimal_point = 3;
            break;

        case 3: my.decimal_point = 2;
            break;

        case 4: my.decimal_point = 1;
            break;

        case 5: my.decimal_point = 0;
            break;

        case 6: $V = $$ * 10;
            my.decimal_point = 0;
            break;

        case 7: $V = $$ * 100;
            my.decimal_point = 0;
            break;

        default: my.decimal_point = 2;
            break;
    }
}
redraw_object();

```

| | | | |
|----------------|-------------------------------|--------------|--------|
| Object (name) | Numerical display (DW _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400757])
    {
        case -3: my.decimal_point = 3;
            break;

        case -2: my.decimal_point = 2;
            break;

        case -1: my.decimal_point = 1;
            break;

        case 0: my.decimal_point = 0;
            break;

        case 1: $V = $$ * 10;
            my.decimal_point = 0;
            break;
    }
}
}
else{

```

```
//ME110
switch([s16:GD61027[s16:GD62552]])
{
    case 1: my.decimal_point = 4;
            break;

    case 2: my.decimal_point = 3;
            break;

    case 3: my.decimal_point = 2;
            break;

    case 4: my.decimal_point = 1;
            break;

    case 5: my.decimal_point = 0;
            break;

    case 6: $V = $$ * 10;
            my.decimal_point = 0;
            break;

    case 7: $V = $$ * 100;
            my.decimal_point = 0;
            break;

    default: my.decimal_point = 2;
            break;
}
```

```
}
redraw_object();
```

| | | | |
|----------------|--------------------------------|--------------|--------|
| Object (name) | Numerical display (Var _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```
// Calculate the position of the decimal point.
```

```
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400757])
    {
        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

        case 1:   $V = $$ * 10;
                my.decimal_point = 0;
                break;
    }
}
```

```
}else{
    //ME110
    switch([s16:GD61027[s16:GD62552]])
    {
        case 1: my.decimal_point = 4;
                break;
```

```

    case 2: my.decimal_point = 3;
            break;

    case 3: my.decimal_point = 2;
            break;

    case 4: my.decimal_point = 1;
            break;

    case 5: my.decimal_point = 0;
            break;

    case 6: $V = $$ * 10;
            my.decimal_point = 0;
            break;

    case 7: $V = $$ * 100;
            my.decimal_point = 0;
            break;

    default: my.decimal_point = 2;
            break;
}

```

```

}
redraw_object();

```

| | | | |
|----------------|-------------------------------|--------------|--------|
| Object (name) | Numerical display (Wh _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.

```

```

if([s16:GD63996]==0)

```

```

{
    //EMU4
    switch([1-248:s16:400758])
    {
        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

        case 1:   $V = $$ * 10;
                my.decimal_point = 0;
                break;

        case 2:   $V = $$ * 100;
                my.decimal_point = 0;
                break;
    }
}

```

```

}else{
    //ME110
    switch([s16:GD61028[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;
    }
}

```

```

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
                break;

        case 5: $V = $$ * 100;
                my.decimal_point = 0;
                break;

        case 6: $V = $$ * 1000;
                my.decimal_point = 0;
                break;

        default: my.decimal_point = 1;
                break;
    }
}

redraw_object();

```

| | | | |
|----------------|---------------------------------|--------------|--------|
| Object (name) | Numerical display (Varh _value) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN32 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400758])
    {
        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:   my.decimal_point = 0;
                break;

        case 1:   $V = $$ * 10;
                my.decimal_point = 0;
                break;

        case 2:   $V = $$ * 100;
                my.decimal_point = 0;
                break;
    }
}
}else{
    //ME110
    switch([s16:GD61028[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
                break;
    }
}

```



```

    case 5: $V = $$ * 100;
        my.decimal_point = 0;
        break;

    case 6: $V = $$ * 1000;
        my.decimal_point = 0;
        break;

    default: my.decimal_point = 1;
        break;
}

}

    redraw_object();

```

| | | | |
|----------------|---|--------------|--------|
| Object (name) | Numerical display (CO2 emission _value) | | |
| Script user ID | 1 | | |
| Data type | Real number | Trigger type | Always |

```

//Calculate CO2 emission.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400758])
    {
        case -2: $V = $$ * 0.01 * [flt:GD61021[s16:GD62552]];
            my.decimal_point = 2;
            break;

        case -1: $V = $$ * 0.1 * [flt:GD61021[s16:GD62552]];
            my.decimal_point = 1;
            break;

        case 0:    $V = $$ * [flt:GD61021[s16:GD62552]];
            break;

        case 1:    $V = $$ * 10 * [flt:GD61021[s16:GD62552]];
            break;

        case 2:    $V = $$ * 100 * [flt:GD61021[s16:GD62552]];
            break;
    }
}
}else{
    //ME110
    switch([s16:GD61028[s16:GD62552]])
    {
        case 1: $V = $$ * 0.01 * [flt:GD61021[s16:GD62552]];
            my.decimal_point = 2;
            break;

        case 2: $V = $$ * 0.1 * [flt:GD61021[s16:GD62552]];
            my.decimal_point = 1;
            break;

        case 3: $V = $$ * [flt:GD61021[s16:GD62552]];
            break;

        case 4: $V = $$ * 10 * [flt:GD61021[s16:GD62552]];
            break;

        case 5: $V = $$ * 100 * [flt:GD61021[s16:GD62552]];
            break;

        case 6: $V = $$ * 1000 * [flt:GD61021[s16:GD62552]];

```

| | | | |
|--|--|--------------|--------|
| break; | | | |
| } | | | |
| } | | | |
| redraw_object(); | | | |
| Object (name) | Numerical display (Electric rate _value) | | |
| Script user ID | 1 | | |
| Data type | Real number | Trigger type | Always |
| //Calculate electric rate. | | | |
| if([s16:GD63996]==0) | | | |
| { | | | |
| //EMU4 | | | |
| switch([1-248:s16:400758]) | | | |
| { | | | |
| case -2: \$V = \$\$ * 0.01 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case -1: \$V = \$\$ * 0.1 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 0: \$V = \$\$ * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 1: \$V = \$\$ * 10 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 2: \$V = \$\$ * 100 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| } | | | |
| }else{ | | | |
| //ME110 | | | |
| switch([s16:GD61028[s16:GD62552]]) | | | |
| { | | | |
| case 1: \$V = \$\$ * 0.01 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 2: \$V = \$\$ * 0.1 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 3: \$V = \$\$ * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 4: \$V = \$\$ * 10 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 5: \$V = \$\$ * 100 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| case 6: \$V = \$\$ * 1000 * [flt:GD61023[s16:GD62552]]; | | | |
| break; | | | |
| } | | | |
| } | | | |
| redraw_object(); | | | |

Base screen 30009

| | | | |
|--|---|--------------|--------|
| Object (name) | Numerical display (Current value _No.1) | | |
| Script user ID | 1 | | |
| Data type | Unsigned BIN32 | Trigger type | Always |
| <pre>// Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400758]) { case -2: my.decimal_point = 2; break; case -1: my.decimal_point = 1; break; case 0: my.decimal_point = 0; break; case 1: \$V = \$\$ * 10; my.decimal_point = 0; break; case 2: \$V = \$\$ * 100; my.decimal_point = 0; break; } } }else{ //ME110 switch([s16:GD61028[s16:GD62552]]) { case 1: my.decimal_point = 2; break; case 2: my.decimal_point = 1; break; case 3: my.decimal_point = 0; break; case 4: \$V = \$\$ * 10; my.decimal_point = 0; break; case 5: \$V = \$\$ * 100; my.decimal_point = 0; break; case 6: \$V = \$\$ * 1000; my.decimal_point = 0; break; } } redraw_object();</pre> | | | |
| Object (name) | Numerical display (Cursor information_No.1) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN16 | Trigger type | Always |
| <pre>// Calculate the position of the decimal point. if([s16:GD63996]==0) {</pre> | | | |

```

//EMU4
switch([1-248:s16:400758])
{
    case -2: my.decimal_point = 2;
            break;

    case -1: my.decimal_point = 1;
            break;

    case 0:   my.decimal_point = 0;
            break;

    case 1:   my.decimal_point = 0;
            break;

    case 2:   my.decimal_point = 0;
            break;
}
}else{
//ME110
switch([s16:GD61028[s16:GD62552]])
{
    case 1: my.decimal_point = 2;
            break;

    case 2: my.decimal_point = 1;
            break;

    case 3: my.decimal_point = 0;
            break;

    case 4: my.decimal_point = 0;
            break;

    case 5: my.decimal_point = 0;
            break;

    case 6: my.decimal_point = 0;
            break;
}
}
redraw_object();

```

Base screen 30010

| | | | |
|---|--|--------------|--------|
| Object (name) | Numerical display (Current value_No.1) | | |
| Script user ID | 1 | | |
| Data type | Unsigned BIN32 | Trigger type | Always |
| // Calculate the position of the decimal point. if([s16:GD63996]==0) { //EMU4 switch([1-248:s16:400755]) { case -3: my.decimal_point = 3; break; case -2: my.decimal_point = 2; break; case -1: my.decimal_point = 1; break; } } | | | |

```

        case 0:    my.decimal_point = 0;
                  break;

    }
} else {
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: $V = $$ * 10;
                my.decimal_point = 0;
                break;
    }
}
redraw_object();

```

| | | | |
|----------------|---|--------------|--------|
| Object (name) | Numerical display (Cursor information_No.1) | | |
| Script user ID | 1 | | |
| Data type | Signed BIN16 | Trigger type | Always |

```

// Calculate the position of the decimal point.
if([s16:GD63996]==0)
{
    //EMU4
    switch([1-248:s16:400755])
    {
        case -3: my.decimal_point = 3;
                break;

        case -2: my.decimal_point = 2;
                break;

        case -1: my.decimal_point = 1;
                break;

        case 0:    my.decimal_point = 0;
                  break;
    }
} else {
    //ME110
    switch([s16:GD61025[s16:GD62552]])
    {
        case 1: my.decimal_point = 2;
                break;

        case 2: my.decimal_point = 1;
                break;

        case 3: my.decimal_point = 0;
                break;

        case 4: my.decimal_point = 0;
                break;
    }
}
redraw_object();

```

Base screen 30020

| | | | |
|---|---|--------------|--------|
| Object (name) | Comment display(Word)(Phase wire system) | | |
| Script user ID | 1 | | |
| Data type | Unsigned BIN16 | Trigger type | Always |
| <pre>//Change displayed comment according to the model. if([w:GD62553] >= 1) { [w:GD62608] = [1-248:w:400513] + 10; }else{ [w:GD62608] = [1-248:w:400513]; }</pre> | | | |

Window screen 30003

| | | | |
|---|--------------------------|--------------|-----------|
| Object (name) | Numerical display (Year) | | |
| Script user ID | 1 | | |
| Data type | Unsigned BIN16 | Trigger type | Rise GB40 |
| <pre>// 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 Year 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</pre> | | | |

| | | | |
|---|----------------------------|--------------|--------|
| Object (name) | Numerical display (Month) | | |
| Script user ID | 2 | | |
| Data type | Unsigned BIN16 | Trigger type | Always |
| // 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 (name) | Numerical display (Day) | | |
| Script user ID | 3 | | |
| Data type | Unsigned BIN16 | Trigger type | Always |
| // Year & Month Setting [w:GS513] = ([w:TMP980] << 8) + [w:TMP981]; // Set Year & Month to Change Time Device | | | |
| Object (name) | Numerical display (Hour) | | |
| Script user ID | 4 | | |
| Data type | Unsigned BIN16 | Trigger type | Always |
| // Date & Time Setting [w:GS514] = ([w:TMP982] << 8) + [w:TMP983]; // Set Date & Time to Change Time Device | | | |
| Object (name) | Numerical display (Minute) | | |
| Script user ID | 5 | | |
| Data type | Unsigned BIN16 | Trigger type | Always |
| // Minute & Second Setting [w:GS515] = ([w:TMP984] << 8) + [w:TMP985]; // Set Minute & Second to Change Time Device | | | |
| Object (name) | Numerical display (Second) | | |
| Script user ID | 6 | | |
| Data type | Unsigned BIN16 | Trigger type | Always |
| // 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; | | | |

6 About Startup System Alarm (Communication Time Out)

Because station No. that is specified in User alarm monitor is accessed right after startup, a system alarm (communication time out) is displayed in case a station is not connected to a controller. Although stations that do not exist are specified as invalid stations in the Valid/invalid station setting, a system alarm cannot be inhibited because controllers are accessed prior to interception when starting up.

By following a procedure 1) below with GT Designer3, an unnecessary system alarm can be inhibited right after startup.

- 1) Delete the device setting for non-existent station No. in User alarm monitor [30001 alarm1].

By following the table below, delete the device setting for unconnected station No.

| No. | Device No. | Station No. |
|-----|-----------------|-------------|
| 1 | 1-1 400525. b2 | 1 |
| 2 | 1-1 400525. b3 | 1 |
| 3 | 1-1 400525. b8 | 1 |
| 4 | 1-1 400525. b9 | 1 |
| 5 | 1-1 400525. b10 | 1 |
| 6 | 1-1 400525. b11 | 1 |
| 7 | 1-1 400525. b12 | 1 |
| 8 | 1-1 400525. b13 | 1 |
| 9 | 1-1 400525. b14 | 1 |
| 10 | 1-1 400525. b15 | 1 |
| 11 | 1-2 400525. b2 | 2 |
| 12 | 1-2 400525. b3 | 2 |
| 13 | 1-2 400525. b8 | 2 |
| 14 | 1-2 400525. b9 | 2 |
| 15 | 1-2 400525. b10 | 2 |
| 16 | 1-2 400525. b11 | 2 |
| 17 | 1-2 400525. b12 | 2 |
| 18 | 1-2 400525. b13 | 2 |
| 19 | 1-2 400525. b14 | 2 |
| 20 | 1-2 400525. b15 | 2 |
| 21 | 1-3 400525. b2 | 3 |
| 22 | 1-3 400525. b3 | 3 |
| 23 | 1-3 400525. b8 | 3 |
| 24 | 1-3 400525. b9 | 3 |
| 25 | 1-3 400525. b10 | 3 |
| 26 | 1-3 400525. b11 | 3 |
| 27 | 1-3 400525. b12 | 3 |
| 28 | 1-3 400525. b13 | 3 |
| 29 | 1-3 400525. b14 | 3 |
| 30 | 1-3 400525. b15 | 3 |
| 31 | 1-4 400525. b2 | 4 |
| 32 | 1-4 400525. b3 | 4 |
| 33 | 1-4 400525. b8 | 4 |
| 34 | 1-4 400525. b9 | 4 |
| 35 | 1-4 400525. b10 | 4 |
| 36 | 1-4 400525. b11 | 4 |
| 37 | 1-4 400525. b12 | 4 |
| 38 | 1-4 400525. b13 | 4 |
| 39 | 1-4 400525. b14 | 4 |
| 40 | 1-4 400525. b15 | 4 |
| 41 | 1-5 400525. b2 | 5 |
| 42 | 1-5 400525. b3 | 5 |

| | | |
|----|-----------------|----|
| 43 | 1-5 400525. b8 | 5 |
| 44 | 1-5 400525. b9 | 5 |
| 45 | 1-5 400525. b10 | 5 |
| 46 | 1-5 400525. b11 | 5 |
| 47 | 1-5 400525. b12 | 5 |
| 48 | 1-5 400525. b13 | 5 |
| 49 | 1-5 400525. b14 | 5 |
| 50 | 1-5 400525. b15 | 5 |
| 51 | 1-6 400525. b2 | 6 |
| 52 | 1-6 400525. b3 | 6 |
| 53 | 1-6 400525. b8 | 6 |
| 54 | 1-6 400525. b9 | 6 |
| 55 | 1-6 400525. b10 | 6 |
| 56 | 1-6 400525. b11 | 6 |
| 57 | 1-6 400525. b12 | 6 |
| 58 | 1-6 400525. b13 | 6 |
| 59 | 1-6 400525. b14 | 6 |
| 60 | 1-6 400525. b15 | 6 |
| 61 | 1-7 400525. b2 | 7 |
| 62 | 1-7 400525. b3 | 7 |
| 63 | 1-7 400525. b8 | 7 |
| 64 | 1-7 400525. b9 | 7 |
| 65 | 1-7 400525. b10 | 7 |
| 66 | 1-7 400525. b11 | 7 |
| 67 | 1-7 400525. b12 | 7 |
| 68 | 1-7 400525. b13 | 7 |
| 69 | 1-7 400525. b14 | 7 |
| 70 | 1-7 400525. b15 | 7 |
| 71 | 1-8 400525. b2 | 8 |
| 72 | 1-8 400525. b3 | 8 |
| 73 | 1-8 400525. b8 | 8 |
| 74 | 1-8 400525. b9 | 8 |
| 75 | 1-8 400525. b10 | 8 |
| 76 | 1-8 400525. b11 | 8 |
| 77 | 1-8 400525. b12 | 8 |
| 78 | 1-8 400525. b13 | 8 |
| 79 | 1-8 400525. b14 | 8 |
| 80 | 1-8 400525. b15 | 8 |
| 81 | 1-9 400525. b2 | 9 |
| 82 | 1-9 400525. b3 | 9 |
| 83 | 1-9 400525. b8 | 9 |
| 84 | 1-9 400525. b9 | 9 |
| 85 | 1-9 400525. b10 | 9 |
| 86 | 1-9 400525. b11 | 9 |
| 87 | 1-9 400525. b12 | 9 |
| 88 | 1-9 400525. b13 | 9 |
| 89 | 1-9 400525. b14 | 9 |
| 90 | 1-9 400525. b15 | 9 |
| 91 | 1-10 400525. b2 | 10 |
| 92 | 1-10 400525. b3 | 10 |
| 93 | 1-10 400525. b8 | 10 |
| 94 | 1-10 400525. b9 | 10 |

| | | |
|-----|------------------|----|
| 95 | 1-10 400525. b10 | 10 |
| 96 | 1-10 400525. b11 | 10 |
| 97 | 1-10 400525. b12 | 10 |
| 98 | 1-10 400525. b13 | 10 |
| 99 | 1-10 400525. b14 | 10 |
| 100 | 1-10 400525. b15 | 10 |
| 101 | 1-11 400525. b2 | 11 |
| 102 | 1-11 400525. b3 | 11 |
| 103 | 1-11 400525. b8 | 11 |
| 104 | 1-11 400525. b9 | 11 |
| 105 | 1-11 400525. b10 | 11 |
| 106 | 1-11 400525. b11 | 11 |
| 107 | 1-11 400525. b12 | 11 |
| 108 | 1-11 400525. b13 | 11 |
| 109 | 1-11 400525. b14 | 11 |
| 110 | 1-11 400525. b15 | 11 |
| 111 | 1-12 400525. b2 | 12 |
| 112 | 1-12 400525. b3 | 12 |
| 113 | 1-12 400525. b8 | 12 |
| 114 | 1-12 400525. b9 | 12 |
| 115 | 1-12 400525. b10 | 12 |
| 116 | 1-12 400525. b11 | 12 |
| 117 | 1-12 400525. b12 | 12 |
| 118 | 1-12 400525. b13 | 12 |
| 119 | 1-12 400525. b14 | 12 |
| 120 | 1-12 400525. b15 | 12 |
| 121 | 1-13 400525. b2 | 13 |
| 122 | 1-13 400525. b3 | 13 |
| 123 | 1-13 400525. b8 | 13 |
| 124 | 1-13 400525. b9 | 13 |
| 125 | 1-13 400525. b10 | 13 |
| 126 | 1-13 400525. b11 | 13 |
| 127 | 1-13 400525. b12 | 13 |
| 128 | 1-13 400525. b13 | 13 |
| 129 | 1-13 400525. b14 | 13 |
| 130 | 1-13 400525. b15 | 13 |
| 131 | 1-14 400525. b2 | 14 |
| 132 | 1-14 400525. b3 | 14 |
| 133 | 1-14 400525. b8 | 14 |
| 134 | 1-14 400525. b9 | 14 |
| 135 | 1-14 400525. b10 | 14 |
| 136 | 1-14 400525. b11 | 14 |
| 137 | 1-14 400525. b12 | 14 |
| 138 | 1-14 400525. b13 | 14 |
| 139 | 1-14 400525. b14 | 14 |
| 140 | 1-14 400525. b15 | 14 |
| 141 | 1-15 400525. b2 | 15 |
| 142 | 1-15 400525. b3 | 15 |
| 143 | 1-15 400525. b8 | 15 |
| 144 | 1-15 400525. b9 | 15 |
| 145 | 1-15 400525. b10 | 15 |
| 146 | 1-15 400525. b11 | 15 |

| | | |
|-----|------------------|----|
| 147 | 1-15 400525. b12 | 15 |
| 148 | 1-15 400525. b13 | 15 |
| 149 | 1-15 400525. b14 | 15 |
| 150 | 1-15 400525. b15 | 15 |
| 151 | 1-16 400525. b2 | 16 |
| 152 | 1-16 400525. b3 | 16 |
| 153 | 1-16 400525. b8 | 16 |
| 154 | 1-16 400525. b9 | 16 |
| 155 | 1-16 400525. b10 | 16 |
| 156 | 1-16 400525. b11 | 16 |
| 157 | 1-16 400525. b12 | 16 |
| 158 | 1-16 400525. b13 | 16 |
| 159 | 1-16 400525. b14 | 16 |
| 160 | 1-16 400525. b15 | 16 |
| 161 | 1-17 400525. b2 | 17 |
| 162 | 1-17 400525. b3 | 17 |
| 163 | 1-17 400525. b8 | 17 |
| 164 | 1-17 400525. b9 | 17 |
| 165 | 1-17 400525. b10 | 17 |
| 166 | 1-17 400525. b11 | 17 |
| 167 | 1-17 400525. b12 | 17 |
| 168 | 1-17 400525. b13 | 17 |
| 169 | 1-17 400525. b14 | 17 |
| 170 | 1-17 400525. b15 | 17 |
| 171 | 1-18 400525. b2 | 18 |
| 172 | 1-18 400525. b3 | 18 |
| 173 | 1-18 400525. b8 | 18 |
| 174 | 1-18 400525. b9 | 18 |
| 175 | 1-18 400525. b10 | 18 |
| 176 | 1-18 400525. b11 | 18 |
| 177 | 1-18 400525. b12 | 18 |
| 178 | 1-18 400525. b13 | 18 |
| 179 | 1-18 400525. b14 | 18 |
| 180 | 1-18 400525. b15 | 18 |
| 181 | 1-19 400525. b2 | 19 |
| 182 | 1-19 400525. b3 | 19 |
| 183 | 1-19 400525. b8 | 19 |
| 184 | 1-19 400525. b9 | 19 |
| 185 | 1-19 400525. b10 | 19 |
| 186 | 1-19 400525. b11 | 19 |
| 187 | 1-19 400525. b12 | 19 |
| 188 | 1-19 400525. b13 | 19 |
| 189 | 1-19 400525. b14 | 19 |
| 190 | 1-19 400525. b15 | 19 |
| 191 | 1-20 400525. b2 | 20 |
| 192 | 1-20 400525. b3 | 20 |
| 193 | 1-20 400525. b8 | 20 |
| 194 | 1-20 400525. b9 | 20 |
| 195 | 1-20 400525. b10 | 20 |
| 196 | 1-20 400525. b11 | 20 |
| 197 | 1-20 400525. b12 | 20 |
| 198 | 1-20 400525. b13 | 20 |

| | | |
|-----|------------------|----|
| 199 | 1-20 400525. b14 | 20 |
| 200 | 1-20 400525. b15 | 20 |
| 201 | 1-21 400525. b2 | 21 |
| 202 | 1-21 400525. b3 | 21 |
| 203 | 1-21 400525. b8 | 21 |
| 204 | 1-21 400525. b9 | 21 |
| 205 | 1-21 400525. b10 | 21 |
| 206 | 1-21 400525. b11 | 21 |
| 207 | 1-21 400525. b12 | 21 |
| 208 | 1-21 400525. b13 | 21 |
| 209 | 1-21 400525. b14 | 21 |
| 210 | 1-21 400525. b15 | 21 |
| 211 | 1-22 400525. b2 | 22 |
| 212 | 1-22 400525. b3 | 22 |
| 213 | 1-22 400525. b8 | 22 |
| 214 | 1-22 400525. b9 | 22 |
| 215 | 1-22 400525. b10 | 22 |
| 216 | 1-22 400525. b11 | 22 |
| 217 | 1-22 400525. b12 | 22 |
| 218 | 1-22 400525. b13 | 22 |
| 219 | 1-22 400525. b14 | 22 |
| 220 | 1-22 400525. b15 | 22 |
| 221 | 1-23 400525. b2 | 23 |
| 222 | 1-23 400525. b3 | 23 |
| 223 | 1-23 400525. b8 | 23 |
| 224 | 1-23 400525. b9 | 23 |
| 225 | 1-23 400525. b10 | 23 |
| 226 | 1-23 400525. b11 | 23 |
| 227 | 1-23 400525. b12 | 23 |
| 228 | 1-23 400525. b13 | 23 |
| 229 | 1-23 400525. b14 | 23 |
| 230 | 1-23 400525. b15 | 23 |
| 231 | 1-24 400525. b2 | 24 |
| 232 | 1-24 400525. b3 | 24 |
| 233 | 1-24 400525. b8 | 24 |
| 234 | 1-24 400525. b9 | 24 |
| 235 | 1-24 400525. b10 | 24 |
| 236 | 1-24 400525. b11 | 24 |
| 237 | 1-24 400525. b12 | 24 |
| 238 | 1-24 400525. b13 | 24 |
| 239 | 1-24 400525. b14 | 24 |
| 240 | 1-24 400525. b15 | 24 |
| 241 | 1-25 400525. b2 | 25 |
| 242 | 1-25 400525. b3 | 25 |
| 243 | 1-25 400525. b8 | 25 |
| 244 | 1-25 400525. b9 | 25 |
| 245 | 1-25 400525. b10 | 25 |
| 246 | 1-25 400525. b11 | 25 |
| 247 | 1-25 400525. b12 | 25 |
| 248 | 1-25 400525. b13 | 25 |
| 249 | 1-25 400525. b14 | 25 |
| 250 | 1-25 400525. b15 | 25 |

| | | |
|-----|------------------|----|
| 251 | 1-26 400525. b2 | 26 |
| 252 | 1-26 400525. b3 | 26 |
| 253 | 1-26 400525. b8 | 26 |
| 254 | 1-26 400525. b9 | 26 |
| 255 | 1-26 400525. b10 | 26 |
| 256 | 1-26 400525. b11 | 26 |
| 257 | 1-26 400525. b12 | 26 |
| 258 | 1-26 400525. b13 | 26 |
| 259 | 1-26 400525. b14 | 26 |
| 260 | 1-26 400525. b15 | 26 |
| 261 | 1-27 400525. b2 | 27 |
| 262 | 1-27 400525. b3 | 27 |
| 263 | 1-27 400525. b8 | 27 |
| 264 | 1-27 400525. b9 | 27 |
| 265 | 1-27 400525. b10 | 27 |
| 266 | 1-27 400525. b11 | 27 |
| 267 | 1-27 400525. b12 | 27 |
| 268 | 1-27 400525. b13 | 27 |
| 269 | 1-27 400525. b14 | 27 |
| 270 | 1-27 400525. b15 | 27 |
| 271 | 1-28 400525. b2 | 28 |
| 272 | 1-28 400525. b3 | 28 |
| 273 | 1-28 400525. b8 | 28 |
| 274 | 1-28 400525. b9 | 28 |
| 275 | 1-28 400525. b10 | 28 |
| 276 | 1-28 400525. b11 | 28 |
| 277 | 1-28 400525. b12 | 28 |
| 278 | 1-28 400525. b13 | 28 |
| 279 | 1-28 400525. b14 | 28 |
| 280 | 1-28 400525. b15 | 28 |
| 281 | 1-29 400525. b2 | 29 |
| 282 | 1-29 400525. b3 | 29 |
| 283 | 1-29 400525. b8 | 29 |
| 284 | 1-29 400525. b9 | 29 |
| 285 | 1-29 400525. b10 | 29 |
| 286 | 1-29 400525. b11 | 29 |
| 287 | 1-29 400525. b12 | 29 |
| 288 | 1-29 400525. b13 | 29 |
| 289 | 1-29 400525. b14 | 29 |
| 290 | 1-29 400525. b15 | 29 |
| 291 | 1-30 400525. b2 | 30 |
| 292 | 1-30 400525. b3 | 30 |
| 293 | 1-30 400525. b8 | 30 |
| 294 | 1-30 400525. b9 | 30 |
| 295 | 1-30 400525. b10 | 30 |
| 296 | 1-30 400525. b11 | 30 |
| 297 | 1-30 400525. b12 | 30 |
| 298 | 1-30 400525. b13 | 30 |
| 299 | 1-30 400525. b14 | 30 |
| 300 | 1-30 400525. b15 | 30 |
| 301 | 1-31 400525. b2 | 31 |
| 302 | 1-31 400525. b3 | 31 |

| | | |
|-----|------------------|----|
| 303 | 1-31 400525. b8 | 31 |
| 304 | 1-31 400525. b9 | 31 |
| 305 | 1-31 400525. b10 | 31 |
| 306 | 1-31 400525. b11 | 31 |
| 307 | 1-31 400525. b12 | 31 |
| 308 | 1-31 400525. b13 | 31 |
| 309 | 1-31 400525. b14 | 31 |
| 310 | 1-31 400525. b15 | 31 |