三菱電機株式会社
エネルギー計測ユニット EcoMonitorLight
(形名：EMU4-BD1-MB/EMU4-HD1-MB)、
電子式マルチ指示計器(形名：ME110SSR-MB)

サンプル画面説明書

三菱電機株式会社

サンプルのご利用について

サンプル用の画面データ、取扱説明書などのファイルは、以下の各項に同意の上でご利用いただくものとします。

(1) 当社製品をご使用中またはご使用検討中のお客様がご利用の対象となります。

(2) 当社が提供するファイルの知的財産権は、当社に帰属するものとします。

(3) 当社が提供するファイルは、改竄、転載、譲渡、販売を禁止します。
但し、内容の一部または全てをお客様作成の機器やシステム内の当社製品上でご利用いただく場合はその限りではありません。また、当社製品をご利用いただいたお客様作成の仕様書、設計書、組み込み製品の取扱説明書などへの転載、複製、引用、レイアウトの変更についてもその限りではありません。

(4) 当社が提供するファイルやそのファイルから抽出されるデータを利用することによって生じた如何なる損害も当社は補償をいたしません。お客様の責任においてご利用ください。

(5) 当社が提供するファイルに利用条件などが添付されている場合は、その条件にも従ってください。

(6) 予告なしに当社が提供するファイルの削除や内容の変更を行うことがあります。

(7) 当社が提供するファイルのご使用に際しては、対応するマニュアルおよびマニュアルで紹介している関連マニュアルをよくお読みいただくと共に、安全に対して充分に注意を払って正しい取扱いをしてください。

目次

目次 3

改訂履歴 4

1. 概要 5

2. システム構成 5

3. GOT について 5

3.1 インストールが必要な OS 5

3.2 作画ソフトウェアの接続機器設定 6

3.3 プロジェクトデータ・OS の格納先について 6

4. 計測端末について 6

4.1 計測端末の通信設定 6

4.2 計測端末のパラメータ設定 7

5. 画面仕様 9

5.1 表示言語 9

5.2 画面遷移 9

5.3 画面説明 11

5.3.1 メニュー(B-30001) 11
5.3.2 現在値モニタ(1/7) (B-30002) 12
5.3.3 現在値モニタ(2/7)(B-30003) 13
5.3.4 現在値モニタ(3/7)(B-30004) 14
5.3.5 現在値モニタ(4/7)(B-30005) 15
5.3.6 現在値モニタ(5/7)(B-30006) 16
5.3.7 現在値モニタ(6/7)(B-30007) 17
5.3.8 現在値モニタ(7/7)(B-30008) 18
5.3.9 電気量グラフ(B-30009) 19
5.3.10 総合電流グラフ(B-30010) 20
5.3.11 パラメータ設定(1/5)(B-30020) 21
5.3.12 パラメータ設定(2/5)(B-30021) 22
5.3.13 パラメータ設定(3/5)(B-30022) 23
5.3.14 パラメータ設定(4/5)(B-30023) 24
5.3.15 パラメータ設定(5/5)(B-30024) 25
5.3.16 アラーム履歴(B-30011) 26
5.3.17 アラームリセット(W-30001) 27
5.3.18 時計＆言語設定(W-30002) 28
5.3.19 有効/無効局設定(W-30003) 29
5.3.20 入力確認ダイアログ(W-30014) 30
5.3.21 相線式(W-30015) 31
5.3.22 センサ種別(W-30016) 32
5.3.23 電流、電流デマンド　乗率(W-30017) 33
5.3.24 電圧　乗率(W-30018) 34
5.3.25 電力、電力デマンド、無効電力　乗率(W-30019) 35
5.3.26 電力量、無効電力量　乗率(W-30020) 36

5.4 使用デバイス一覧 37

5.5 コメント一覧 41

5.6 スクリプト一覧 44

改訂履歴

サンプル画面説明書

改訂日付	管理番号*	改訂内容
2013/4/4	SDM-0038	初版

＊ 管理番号は、右下に記載しています。

プロジェクトデータ

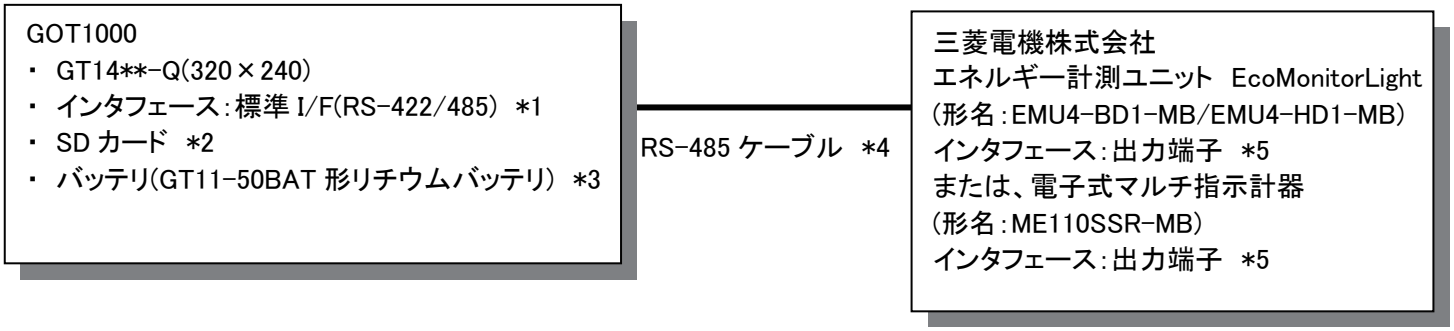
改訂日付	プロジェクトデータ	GT Designer3*	改訂内容
2012/4/4	SDS-0038.GTW	1.68W	初版

＊ プロジェクトデータ作成時に使用した作画ソフトウェアのバージョンです。記載したバージョンと同等、またはそれ以降のバージョンの作画ソフトウェアを使用してください。

1. 概要

GOT1000 と三菱電機株式会社　計測端末　エネルギー計測ユニット EcoMonitorLight：EMU4-BD1-MB/EMU4-HD1-MB(以下、機種 EMU4 と省略)または電子式マルチ指示計器：ME110SSR-MB(以下、機種 ME110 と省略)を MODBUS®/RTU(RS-485)で接続し、電流、電力、電力量のモニタなどを行うサンプル画面の説明書です。

2. システム構成



*1：終端抵抗が必要となります。終端抵抗については、「GOT1000シリーズ接続マニュアル(マイコン・MODBUS®・周辺機器接続編)」を参照してください。
*2：SDカードは、プロジェクトデータ・OSの格納およびロギング機能、拡張レシピ機能で使用しています。
*3：バッテリは、ロギング機能で使用しています。(バッテリはGOT本体に標準装備しています。)
*4：ケーブルの詳細については、「GOT1000シリーズ接続マニュアル(マイコン・MODBUS®・周辺機器接続編)」を参照してください。
*5：終端抵抗が必要となります。終端抵抗については、各機器のマニュアルを参照してください。

3. GOT について

3.1 インストールが必要な OS

OS の種類	OS の名称	
基本機能	基本 OS	
	基本 OS システム画面情報	
	基本 OS システム画面データ	
	標準フォント	日本語(欧州対応)
	TrueType 数字フォント	7 セグ
通信ドライバ	MODBUS®/RTU	
拡張機能	ストローク基本フォント	ストロークフォントサポート機能
		ストローク基本フォント(日本)
	デバイスデータ転送	
オプション機能	標準フォント(中国：簡体)	
	拡張レシピ	
	ロギング	
	オブジェクトスクリプト	

3.2 作画ソフトウェアの接続機器設定

項 目	設定値	備 考
ボーレート(BPS)	38400 bps	
データ長	8 bit	
ストップビット	1 bit	
パリティ	偶数	
リトライ回数(回)	3	
通信タイムアウト時間(秒)	3	
自局アドレス	1	接続されている計測端末の MODBUS®アドレスを設定します。本サンプルでは、1 局固定とします。
送信ディレイ時間(ms)	0	
32 ビット格納順序	HL 順	

3.3 プロジェクトデータ・OS の格納先について

本サンプルのプロジェクトデータ・OS は、GOT の C ドライブ(内蔵フラッシュメモリ)に収まりきらないサイズであるため、A ドライブ(標準 SD カード)に書き込みを行う必要があります。

A ドライブへの書き込み方法については、「GT Designer3 Version1 画面設計マニュアル(共通編)」を参照してください。

4. 計測端末について

4.1 計測端末の通信設定

(1) エネルギー計測ユニット(機種 EMU4)

項 目	設定値	備 考
MODBUS®アドレス	1	(*)
MODBUS®ボーレート	38400 bps	
MODBUS®パリティ	Even	Odd:奇数パリティ
		Even:偶数パリティ
		Non:パリティなし
ストップビット	1 bit	

(*)MODBUS®アドレスは 1 局固定とします。1 局以外を設定した場合、本サンプルは動作しません。

(2) 電子式マルチ指示計器(機種 ME110)

項 目	設定値	備 考
MODBUS®アドレス	1	(*)
MODBUS®ボーレート	38.4kbps	
MODBUS®パリティ	Even	Odd:奇数パリティ
		Even:偶数パリティ
		Non:パリティなし
ストップビット	1 bit	

(*)MODBUS®アドレスは 1 局固定とします。1 局以外を設定した場合、本サンプルは動作しません。

4.2 計測端末のパラメータ設定

弊社で動作確認した際の設定値は下記となります。

(1) エネルギー計測ユニット(機種 EMU4)

項 目	設定値	備 考
相線式	3	1:1P2W
		2:1P3W
		3:3P3W
		4:3P4W
一次電圧(線間電圧)	220V	「相線式」が 3P4W 以外の際に有効
一次電圧(相電圧)	−	「相線式」が 3P4W の際に有効
センサ種別	0	0:ダイレクトセンサ
		2:5A センサ
一次電流	1000x0.1A	
電流デマンド時限	120s	
電力デマンド時限	120s	
電流・電流デマンド 乗率	−2(*1)	−3:×0.001
		−2:×0.01
		−1:×0.1
		0:×1
電圧 乗率	−1(*1)	−1:×0.1
		0:×1
電力・電力デマンド・無効電力 乗率	−2(*1)	−3:×0.001
		−2:×0.01
		−1:×0.1
		0:×1
		1:×10
電力量・無効電力量 乗率	−1(*1)	−2:×0.01
		−1:×0.1
		0:×1
		1:×10
		2:×100
周波数 乗率	×0.1	
力率 乗率	×0.1	

(*1)相線式、一次電圧、一次電流を変更すると、乗率は自動的に変更されます。

(2) 電子式マルチ指示計器(機種 ME110)

項　目	設定値	備　考
相線式	3	1:1P2W
		2:1P3W(RNT)
		3:3P3W
		4:3P4W
		5:1P3W(RNS)
		6:3P3W_3CT
一次電圧(線間電圧)	6600V	「相線式」が 3P4W 以外の際に有効
一次電圧(相電圧)	−	「相線式」が 3P4W の際に有効
一次電流	50x0.1V	
電流デマンド時限	10s	
電力デマンド時限	10s	
CO2 排出係数	1.5	
電気料金単価	1.3	
電流・電流デマンド　乗率	1	1:×0.01
		2:×0.1
		3:×1
		4:×10
電圧　乗率	3	1:×0.1
		2:×1
		3:×10
		4:×100
電力・電力デマンド・無効電力　乗率	3	1:×0.0001
		2:×0.001
		3:×0.01
		4:×0.1
		5:×1
		6:×10
		7:×100
電力量・無効電力量　乗率	2	1:×0.01
		2:×0.1
		3:×1
		4:×10
		5:×100
		6:×1000
周波数　乗率	×0.1	
力率　乗率	×0.1	

5. 画面仕様

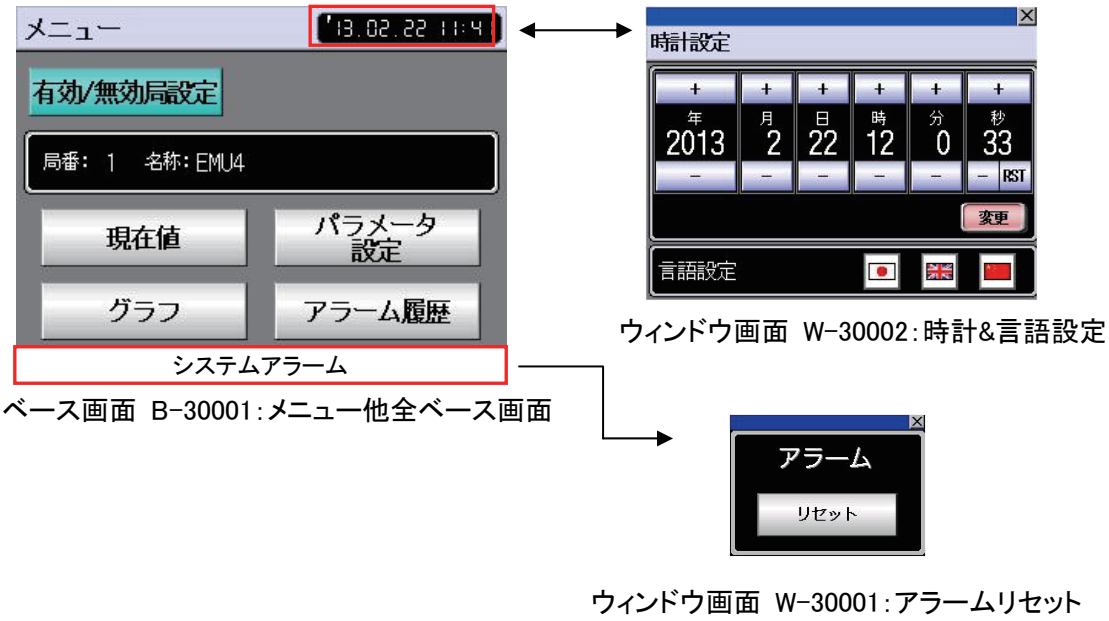
5.1 表示言語

画面上に表示する文字列は、日本語・英語・中国語(簡体)の 3 言語で切り換え表示できます。各言語の文字列は、コメントグループ No.254、255 の列 No.1～3 に下記のように登録しています。言語切り換えデバイスに列 No.を格納すると列 No.に対応した言語を表示します。

列 No.	言語
1	日本語
2	英語
3	中国語(簡体)

5.2 画面遷移

5.2.1 画面遷移(共通)

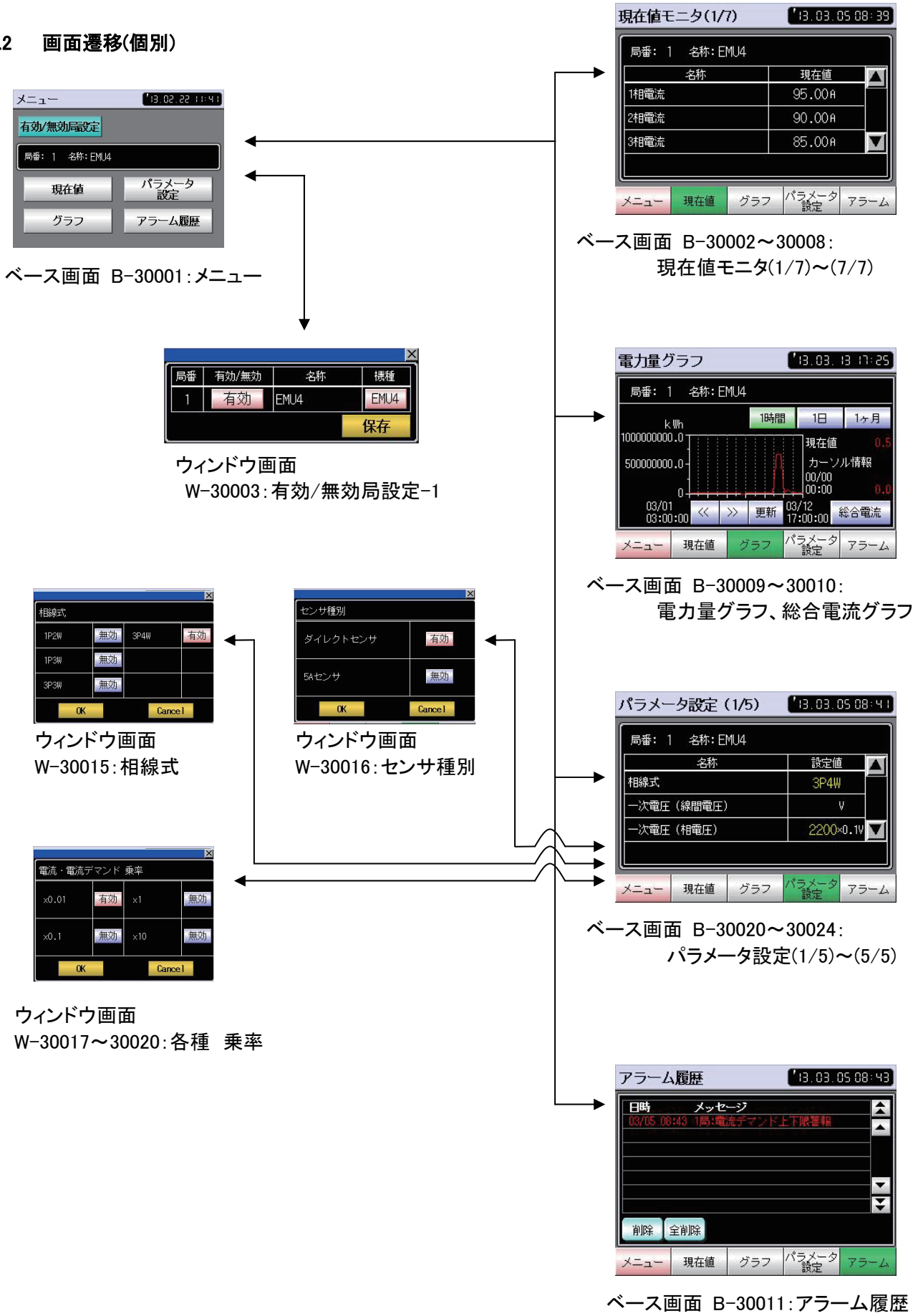


ベース画面 B-30001:メニュー他全ベース画面

ウィンドウ画面 W-30002:時計&言語設定

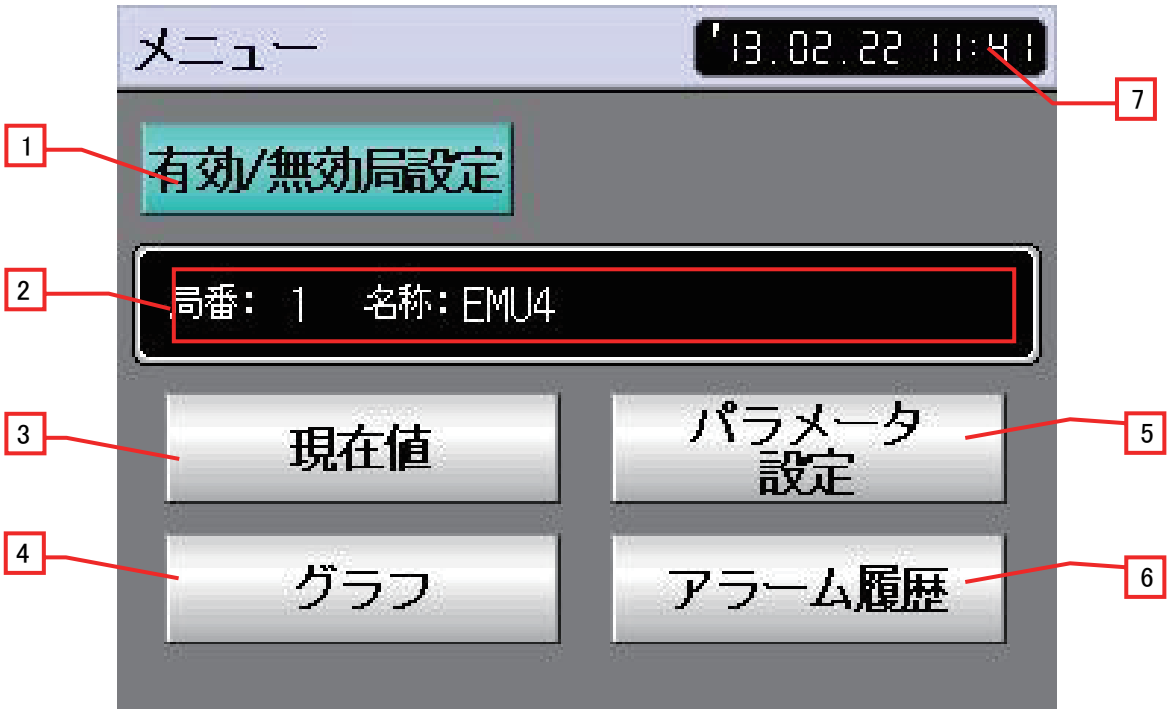
ウィンドウ画面 W-30001:アラームリセット

5.2.2　画面遷移(個別)


メニュー
有効/無効局設定
局番: 1　名称: EMU4
現在値
パラメータ設定
グラフ
アラーム履歴
ベース画面 B-30001:メニュー
現在値モニタ(1/7)
局番: 1　名称: EMU4
名称
現在値
1相電流
95.00A
2相電流
90.00A
3相電流
85.00A
メニュー
現在値
グラフ
パラメータ設定
アラーム
ベース画面 B-30002～30008:
現在値モニタ(1/7)～(7/7)
局番
有効/無効
名称
機種
1
有効
EMU4
EMU4
保存
ウィンドウ画面
W-30003:有効/無効局設定-1
電力量グラフ
局番: 1　名称: EMU4
1時間
1日
1ヶ月
kWh
現在値
カーソル情報
更新
総合電流
ベース画面 B-30009～30010:
電力量グラフ、総合電流グラフ
相線式
1P2W
無効
3P4W
有効
1P3W
無効
3P3W
無効
OK
Cancel
ウィンドウ画面
W-30015:相線式
センサ種別
ダイレクトセンサ
有効
5Aセンサ
無効
OK
Cancel
ウィンドウ画面
W-30016:センサ種別
パラメータ設定 (1/5)
局番: 1　名称: EMU4
名称
設定値
相線式
3P4W
一次電圧 (線間電圧)
V
一次電圧 (相電圧)
2200×0.1V
メニュー
現在値
グラフ
パラメータ設定
アラーム
ベース画面 B-30020～30024:
パラメータ設定(1/5)～(5/5)
電流・電流デマンド 乗率
x0.01
有効
x1
無効
x0.1
無効
x10
無効
OK
Cancel
ウィンドウ画面
W-30017～30020:各種　乗率
アラーム履歴
日時
メッセージ
03/05 08:43 1局:電流デマンド上下限警報
削除
全削除
メニュー
現在値
グラフ
パラメータ設定
アラーム
ベース画面 B-30011:アラーム履歴

5.3 画面説明

5.3.1 メニュー(B-30001)



概要

メニュー画面です。

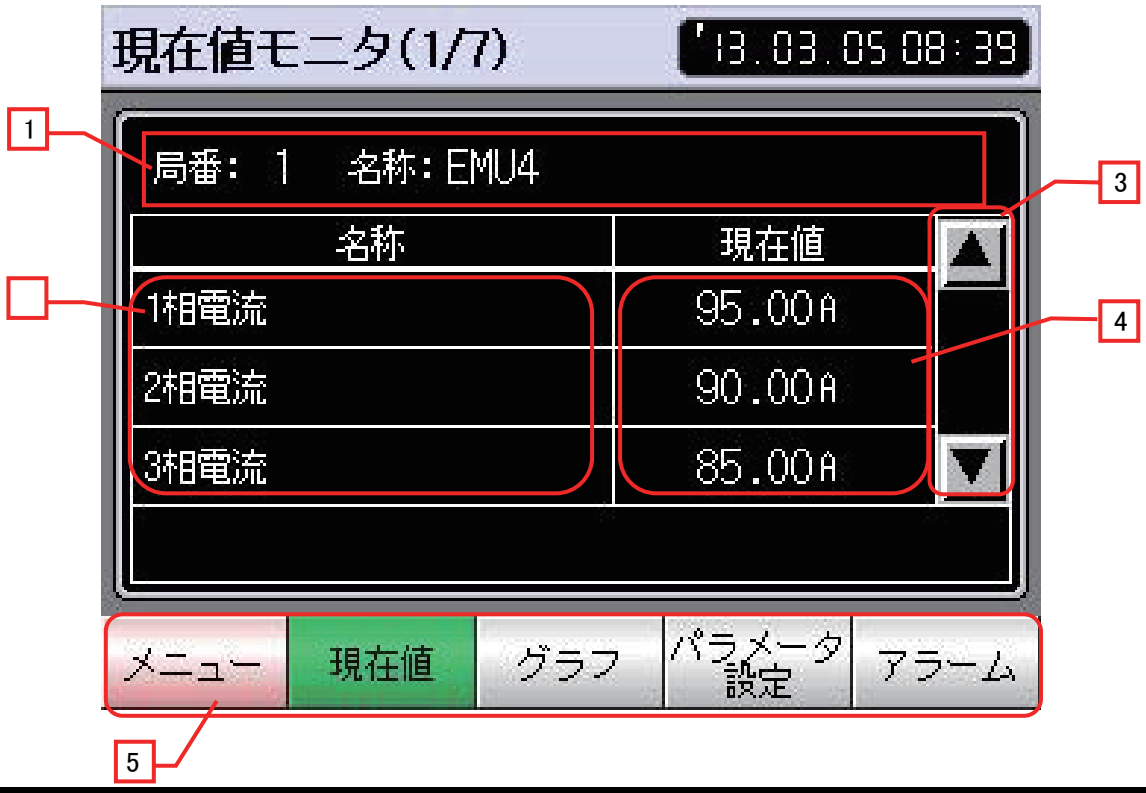
詳細

1. 計測端末の局番の有効/無効切り換えや、名称・機種の設定をするためのウィンドウを表示します。
2. 現在モニタ中の計測端末の局番と名称を表示します。
3. 現在値モニタ(1/7)画面に切り換えます。
4. 電力量グラフ画面に切り換えます。
5. パラメータ設定(1/5)画面に切り換えます。
6. アラーム履歴画面に切り換えます。
7. 現在の日時を表示します。押下すると、時計および言語設定ウィンドウを表示します。この動作は、全ベース画面で統一しています。

備考

・GOT 起動時に、プロジェクトスクリプトにて拡張レシピファイルから 1 局の情報を読み出し、モニタ対象局として設定しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.2　現在値モニタ(1/7)(B-30002)



概要

1 相電流、2 相電流、3 相電流の現在値をモニタします。現在値の小数点位置は、乗率により異なります。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(7/7)画面に切り換えます。
   ▼:現在値モニタ(2/7)画面に切り換えます。
4. 1 相電流、2 相電流、3 相電流の現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・乗率のパターンは下記のとおりです。
　機種 EMU4:×0.001、×0.01、×0.1、×1
　機種 ME110:×0.01、×0.1、×1、×10
・機種 EMU4 の乗率は、相線式、一次電圧、一次電流を変更すると自動的に変更されます。機種 ME110 の乗率は、ユーザで設定します。
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.3　現在値モニタ(2/7)(B-30003)

概要

1 相電流デマンド、2 相電流デマンド、3 相電流デマンドの現在値をモニタします。現在値の小数点位置は、乗率により異なります。

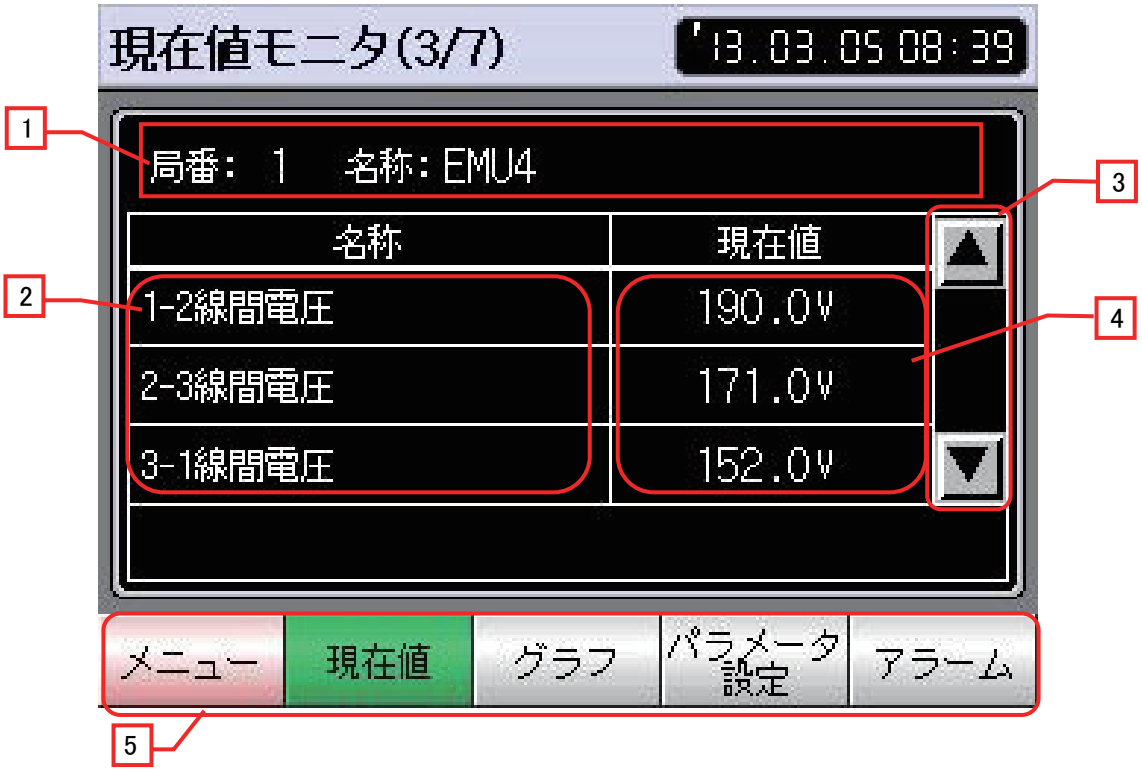
詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(1/7)画面に切り換えます。
   ▼:現在値モニタ(3/7)画面に切り換えます。
4. 1 相電流デマンド、2 相電流デマンド、3 相電流デマンドの現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・乗率のパターンは下記のとおりです。
　機種 EMU4:×0.001、×0.01、×0.1、×1
　機種 ME110:×0.01、×0.1、×1、×10
・機種 EMU4 の乗率は、相線式、一次電圧、一次電流を変更すると自動的に変更されます。機種 ME110 の乗率は、ユーザで設定します。
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.4　現在値モニタ(3/7)(B-30004)



概要

1-2 線間電圧、2-3 線間電圧、3-1 線間電圧の現在値をモニタします。現在値の小数点位置は、乗率により異なります。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(2/7)画面に切り換えます。
   ▼:現在値モニタ(4/7)画面に切り換えます。
4. 1-2 線間電圧、2-3 線間電圧、3-1 線間電圧の現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・乗率のパターンは下記のとおりです。
　機種 EMU4:×0.1、×1
　機種 ME110:×0.1、×1、×10、×100
・機種 EMU4 の乗率は、相線式、一次電圧、一次電流を変更すると自動的に変更されます。機種 ME110 の乗率は、ユーザで設定します。
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.5　現在値モニタ(4/7)(B-30005)

概要

電力、電力デマンド、無効電力の現在値をモニタします。現在値の小数点位置は、乗率により異なります。

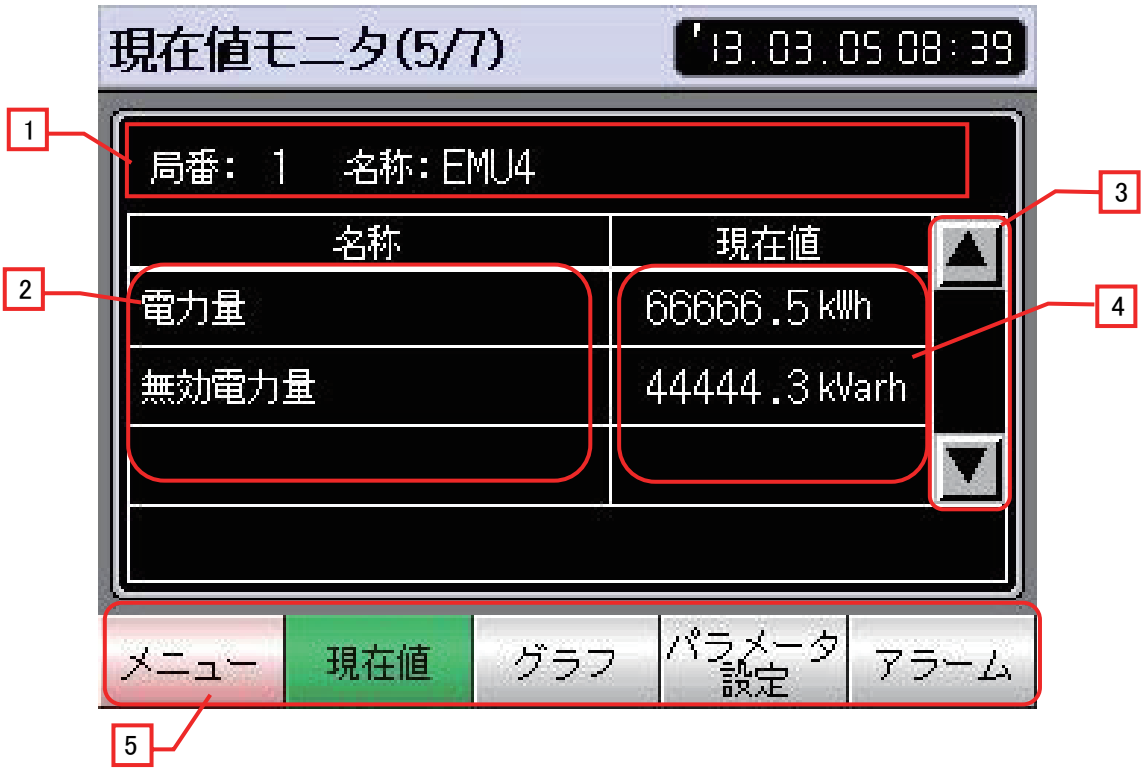
詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(3/7)画面に切り換えます。
   ▼:現在値モニタ(5/7)画面に切り換えます。
4. 電力、電力デマンド、無効電力の現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・乗率のパターンは下記のとおりです。
　機種 EMU4:×0.001、×0.01、×0.1、×1、×10
　機種 ME110:×0.0001、×0.001、×0.01、×0.1、×1、×10、×100
・機種 EMU4 の乗率は、相線式、一次電圧、一次電流を変更すると自動的に変更されます。機種 ME110 の乗率は、ユーザで設定します。
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.6　現在値モニタ(5/7)(B-30006)



概要

電力量、無効電力量の現在値をモニタします。現在値の小数点位置は、乗率により異なります。

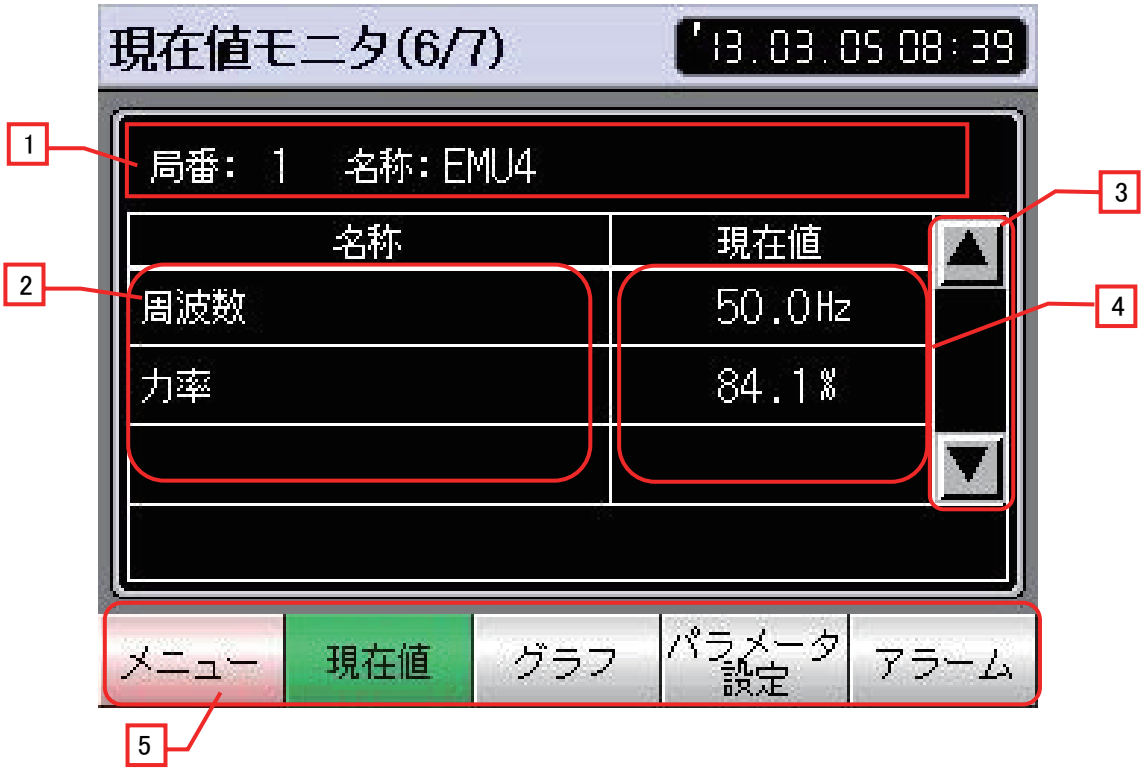
詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(4/7)画面に切り換えます。
   ▼:現在値モニタ(6/7)画面に切り換えます。
4. 電力量、無効電力量の現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・乗率のパターンは下記のとおりです。
　機種 EMU4: ×0.01、×0.1、×0.1、×1、×10、×100
　機種 ME110: ×0.01、×0.1、×1、×10、×100、×1000
・機種 EMU4 の乗率は、相線式、一次電圧、一次電流を変更すると自動的に変更されます。機種 ME110 の乗率は、ユーザで設定します。
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.7　現在値モニタ(6/7)(B-30007)



概要

周波数、力率の現在値をモニタします。周波数、力率共に、小数点位置は 1 桁で固定です。

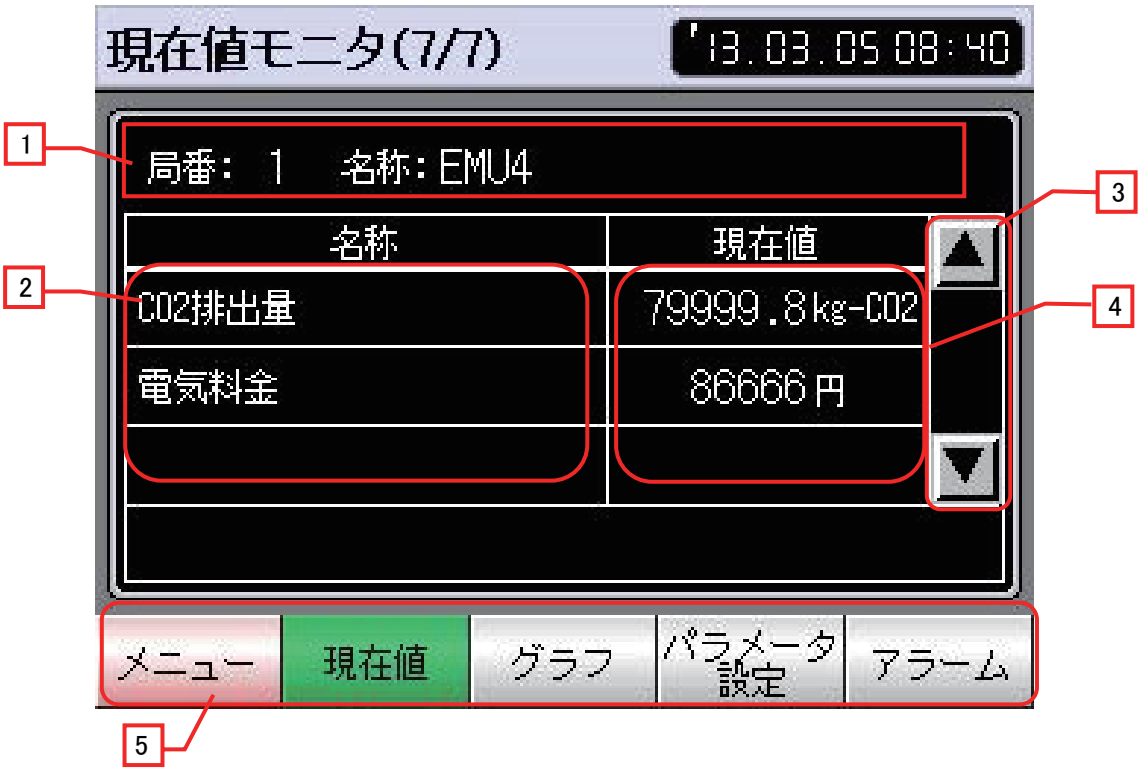
詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(5/7)画面に切り換えます。
   ▼:現在値モニタ(7/7)画面に切り換えます。
4. 周波数、力率の現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.8　現在値モニタ(7/7)(B-30008)



概要

CO2 排出量、電気料金の現在値をモニタします。CO2 排出量の小数点位置は、1 桁で固定です。

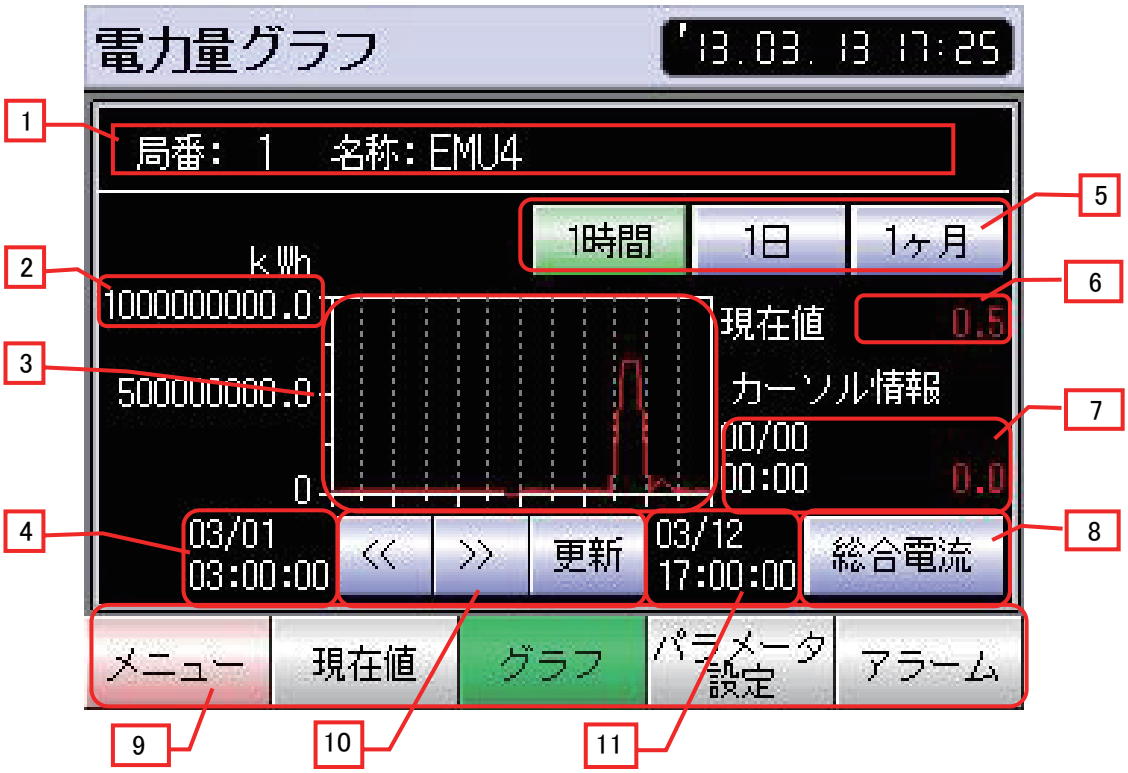
詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. モニタしている計測値の名称を表示します。
3. 現在値モニタ画面を切り換えます。
   ▲:現在値モニタ(6/7)画面に切り換えます。
   ▼:現在値モニタ(1/7)画面に切り換えます。
4. CO2 排出量、電気料金の現在値を表示します。
5. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・CO2 排出量は、下記の計算に基づき、算出しています。
　CO2 排出量(kg-CO2)=電力量(kWh)XCO2 排出係数(kg-CO2/kWh)
・CO2 排出量は、計測端末側の CO2 排出量(CO2 量換算値)表示とは異なります。
・電気料金は、下記の計算に基づき、算出しています。
　電気料金(円 or ＄ or 元)=電気量(kWh)X 電気料金単価(円 or ＄ or 元 /kWh)
・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.9　電気量グラフ(B-30009)



概要

電力量の現在値を表示し、ロギング機能で収集した過去の電力量をヒストリカルトレンドグラフで表示します。

詳細

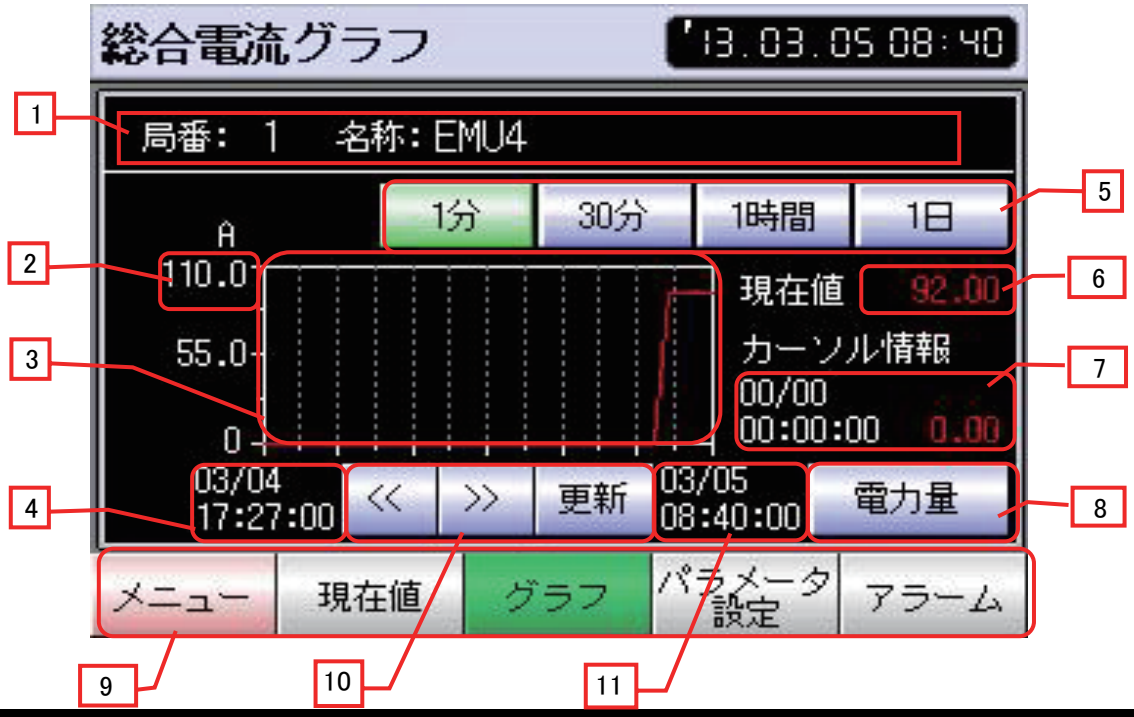
1. 現在モニタ中の計測端末の局番と名称を表示します。
2. ヒストリカルトレンドグラフの上限値を表示します。数値を押下することで、上限値の入力をします。入力した上限値は拡張レシピ機能で保存され、値を保持します。上限値は、局番を切り換えると、局番に対応した上限値へ切り換わります。
3. 電力量をヒストリカルトレンドグラフで表示します。グラフ上を押下すると、カーソルを表示します。横軸のデータ数は 31 点です。
4. ヒストリカルトレンドグラフの表示終了位置時刻を表示します。
5. ヒストリカルトレンドグラフで表示するロギングデータを切り換えます。
   「1 時間」:毎 00 分に収集した、1 時間ごとのロギングデータを表示します。
   「1 日」:毎 00 分に収集した、1 日ごとのロギングデータを表示します。
   「1 ヶ月」:毎 00 分に収集した、1ヶ月ごとのロギングデータを表示します。
6. 電力量の現在値を表示します。小数点位置は、乗率により異なります。
7. カーソル位置の日時と、電力量を表示します。電力量の小数点位置は、乗率により異なります。
8. 「総合電流グラフ」画面に切り換えます。
9. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。
10. ヒストリカルトレンドグラフを操作します。
    「<<」「>>」:グラフを左右にスクロールします。
    更新:カーソルを消去し、最新のデータを表示します。
11. ヒストリカルトレンドグラフの表示開始位置時刻を表示します。

備考

・デバイスデータ転送機能で計測端末から電力量を、GOT の内部デバイスに転送しています。転送した値を、スクリプトを使用して実数に変換し、変換した値をロギングしています。そのため、実数に変換した値が 7 桁を超える場合には、誤差の発生する場合があります。スクリプトの詳細は「5.6　スクリプト一覧」を参照してください。

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.10 総合電流グラフ(B-30010)



概要

総合電流の現在値を表示し、ロギング機能で収集した過去の総合電流をヒストリカルトレンドグラフで表示します。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. ヒストリカルトレンドグラフの上限値を表示します。数値を押下することで、上限値の入力をします。入力した上限値は拡張レシピ機能で保存され、値を保持します。上限値は、局番を切り換えると、局番に対応した上限値へ切り換わります。
3. 総合電流をヒストリカルトレンドグラフで表示します。グラフ上を押下すると、カーソルを表示します。横軸のデータ数は 31 点です。
4. ヒストリカルトレンドグラフの表示終了位置時刻を表示します。
5. ヒストリカルトレンドグラフで表示するロギングデータを切り換えます。
   「1 分」:毎 00 秒に収集した、1分ごとのロギングデータを表示します。
   「30 分」:毎 00 分と毎 30 分に収集した、30 分ごとのロギングデータを表示します。
   「1 時間」:毎 00 分 1 に収集した、時間ごとのロギングデータを表示します。
   「1 日」:毎 00 時 00 分に収集した、1 日ごとのロギングデータを表示します。
6. 総合電流の現在値を表示します。小数点位置は、乗率により異なります。
7. カーソル位置の日時と、総合電流を表示します。総合電流の小数点位置は、乗率により異なります。
8. 「電力量グラフ」画面に切り換えます。
9. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。
10. ヒストリカルトレンドグラフを操作します。
    「<<」「>>」:グラフを左右にスクロールします。
    更新:カーソルを消去し、最新のデータを表示します。
11. ヒストリカルトレンドグラフの表示開始位置時刻を表示します。

備考

- デバイスデータ転送機能で計測端末から総合電流を、GOT の内部デバイスへ転送しています。転送した値を、スクリプトを使用して実数に変換し、変換した値をロギングしています。そのため、実数に変換した値が 7 桁を超える場合、誤差の発生する可能性があります。スクリプトの詳細は「5.6 スクリプト一覧」を参照してください。
- 機種 ME110 は、相線式が 3P4W 以外に設定されている場合、総合電流グラフは 0 表示となります。
- システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.11 パラメータ設定(1/5)(B-30020)

概要

相線式、一次電圧(線間電圧)、一次電圧(相電圧)の設定をします。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. 設定値の名称を表示します。
3. パラメータ設定画面を切り換えます。
   ▲:パラメータ設定(5/5)画面に切り換えます。
   ▼:パラメータ設定(2/5)画面に切り換えます。
4. 設定値を押下すると、相線式を設定するための、ウィンドウを表示します。
5. 一次電圧(線間電圧)を設定します。一次電圧(線間電圧)は、相線式が 3P4W 以外に設定されている場合のみ、設定可能です。3P4W が設定されている場合は、非表示となります。
6. 一次電圧(相電圧)を設定します。一次電圧(相電圧)は、相線式が 3P4W に設定されている場合のみ、設定可能です。3P4W 以外が設定されている場合は、非表示となります。
7. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.12 パラメータ設定(2/5)(B-30021)

概要

センサ種別、一次電流、電流デマンド時限の設定をします。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. 設定値の名称を表示します。
3. パラメータ設定画面を切り換えます。
   ▲:パラメータ設定(1/5)画面に切り換えます。
   ▼:パラメータ設定(3/5)画面に切り換えます。
4. 設定値を押下すると、センサ種別を設定するためのウィンドウを表示します。センサ種別は、接続機種が機種 EMU4 の場合のみ、設定可能です。機種 ME110 の場合は、非表示となります。
5. 一次電流を設定します。設定できる値は、選択した機種、相線式により異なるため、詳細は各機種のマニュアルを参照してください。
6. 電流デマンド時限を設定します。
7. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.13 パラメータ設定(3/5)(B-30022)

概要

電力デマンド時限、CO2 排出係数、電気料金単価の設定をします。CO2 排出係数の小数点位置は 3 桁、電気料金単価の小数点位置は 2 桁で固定です。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. 設定値の名称を表示します。
3. パラメータ設定画面を切り換えます。
   ▲:パラメータ設定(2/5)画面に切り換えます。
   ▼:パラメータ設定(4/5)画面に切り換えます。
4. 電力デマンド時限を設定します。
5. CO2 排出係数を設定します。
6. 電気料金単価を設定します。
7. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

- CO2 排出係数は、計測端末側の固有パラメータではありません。
- 電気料金単価は、計測端末側の固有パラメータではありません。
- システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.14 パラメータ設定(4/5)(B-30023)

概要

機種 ME110 用の電流・電流デマンド 乗率、電圧 乗率、電力・電力デマンド・無効電力 乗率を設定します。
機種 EMU4 の場合は、設定値は非表示となります。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. 設定値の名称を表示します。
3. パラメータ設定画面を切り換えます。
   ▲:パラメータ設定(3/5)画面に切り換えます。
   ▼:パラメータ設定(5/5)画面に切り換えます。
4. 設定値を押下すると、電流・電流デマンド 乗率を設定するためのウィンドウを表示します。
5. 設定値を押下すると、電圧 乗率を設定するためのウィンドウを表示します。
6. 設定値を押下すると、電力・電力デマンド・無効電力 乗率を設定するためのウィンドウを表示します。
7. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.15 パラメータ設定(5/5)(B-30024)

概要

機種 ME110 用の電力量・無効電力量 乗率、周波数 乗率、力率 乗率を設定します。機種 EMU4 の場合は、設定値は非表示となります。

詳細

1. 現在モニタ中の計測端末の局番と名称を表示します。
2. 設定値の名称を表示します。
3. パラメータ設定画面を切り換えます。
   ▲:パラメータ設定(4/5)画面に切り換えます。
   ▼:パラメータ設定(1/5)画面に切り換えます。
4. 設定値を押下すると、電力量・無効電力量 乗率を設定するためのウィンドウ画面が開きます。
5. 周波数 乗率を表示します。周波数 乗率は変更できません。
6. 力率 乗率を表示します。力率 乗率は変更できません。
7. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.16 アラーム履歴(B-30011)

概要

計測端末の警報と異常ステータスを表示します。

詳細

1. 拡張ユーザアラーム表示でアラームを表示します。
2. 拡張ユーザアラーム表示を操作します。
   削除:選択された復旧済みアラームを削除します。
   全削除:復旧済みアラームをすべて削除します。
   ⏫ ⏬ :上下にページスクロールします。
   ▲ ▼ :カーソルを上下に移動します。
3. 各画面に切り換えます。緑色のスイッチは、現在表示中画面のため画面は切り換わりません。

備考

・システムアラームが発生した場合、画面下にアラームメッセージを表示します。アラームメッセージを押下すると、アラームリセットウィンドウが表示されます。

5.3.17 アラームリセット(W-30001)

概要

システムアラームをリセットします。

詳細

1. システムアラームをリセットし、1秒後にウィンドウ画面を閉じます。

備考

5.3.18 時計＆言語設定(W-30002)

概要

GOT の時計データの変更と、表示する言語を選択します。

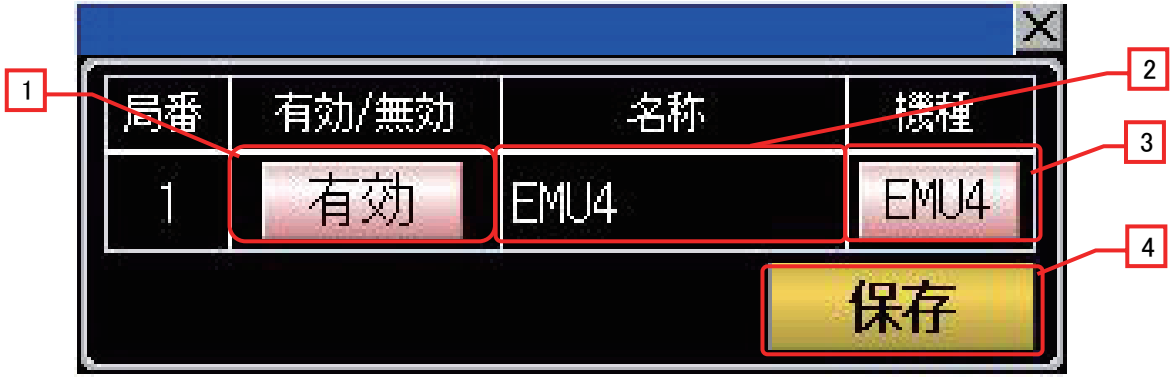
詳細

1. 変更したい日時を [+] [−] スイッチで設定します。[+] [−] スイッチは、長押しすると連続で増減します。RST スイッチは、秒をリセットします。
2. 変更する日時を表示します。
3. 設定した日時を GOT の時計データに反映し、1秒後にウィンドウ画面を閉じます。
4. 言語を切り換え、ウィンドウ画面を閉じます。

備考

・変更する日時の初期値は、ウィンドウ画面を表示した時の日時です。
・変更する日時の年・月・日・時・分・秒の数値表示にオブジェクトスクリプトを設定しています。スクリプトの詳細については、「5.6 スクリプト一覧」を参照してください。

5.3.19 有効/無効局設定(W-30003)



概要

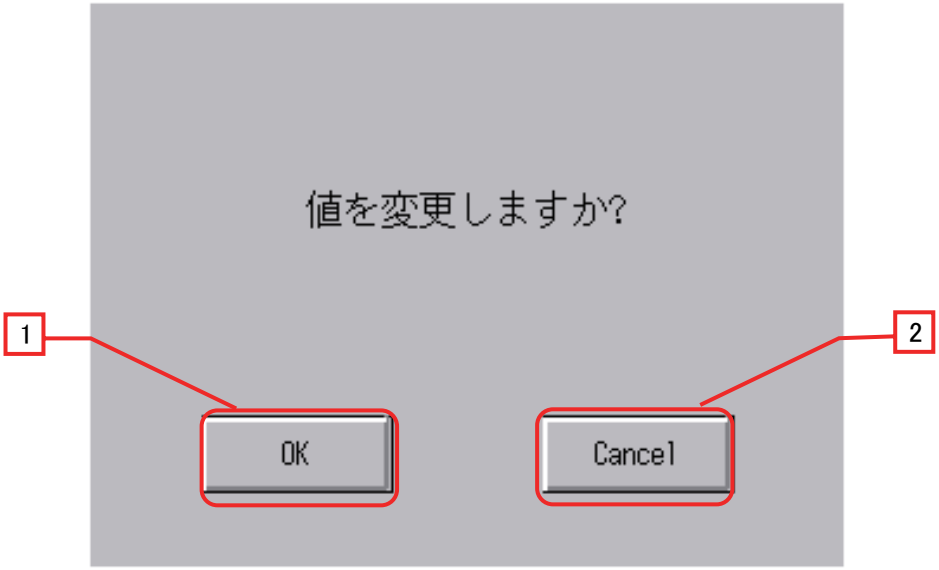
計測端末の局番の有効/無効切り換え、名称の設定、機種の設定を行います。

詳細

1. 計測端末の局番の有効/無効を切り換えます。
2. 指定した局番の計測端末の名称を設定します。
3. 指定した局番の計測端末の機種を選択します。機種 EMU4 または、機種 ME110 を選択します。
4. 設定した内容を拡張レシピファイルへ保存します。

備考

5.3.20 入力確認ダイアログ(W-30014)



概要

各パラメータ設定用ウィンドウ画面で設定した値を確定する際に表示するダイアログウィンドウです。

詳細

1. 入力した値を設定し、ダイアログを閉じます。
2. 入力は確定されず、ダイアログを閉じます。

備考

・以下のパラメータの値は、GOT 側で停電保持するため、拡張レシピファイルに保存します。
機種 EMU4、機種 ME110 共通:CO2 排出係数、電気料金単価
機種 ME110 のみ:各種乗率(周波数、力率除く)

5.3.21 相線式(W-30015)

概要

相線式を設定します。相線式は、機種により、設定できる内容が異なります。

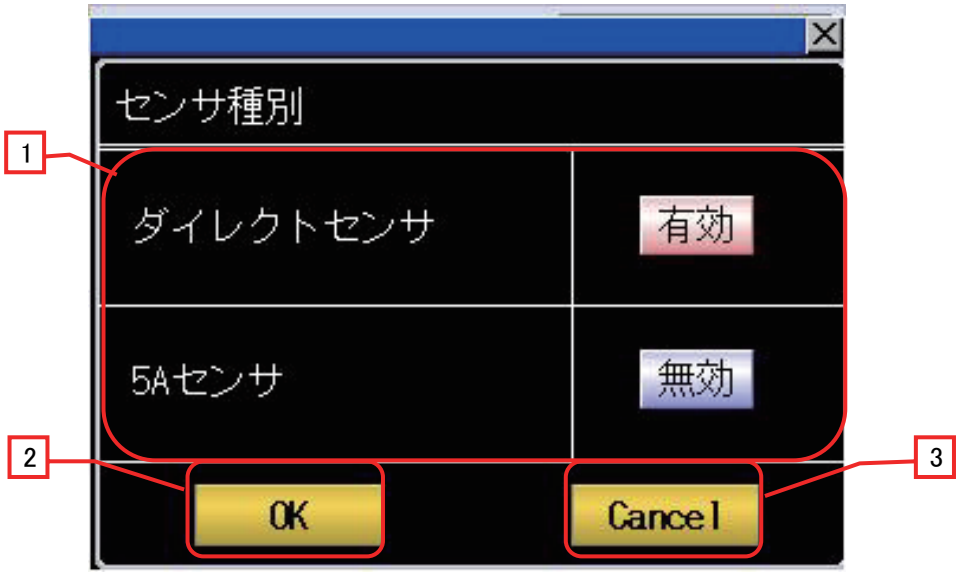
詳細

1. 設定する相線式を選択します。機種ごとの設定できる相線式は下記のとおりです。
   機種 EMU4:1P2W、1P3W、3P3W、3P4W
   機種 ME110:1P2W、1P3W(RNT)、3P3W、3P4W、1P3W(RNS)、3P3W_3CT
2. 確認ダイアログウィンドウが表示され、OK を押下すると、入力が確定します。
3. 設定した値を破棄し、ウィンドウを閉じます。

備考

・機種 ME110 の場合、相線式の変更処理に 2～3 秒かかります。この間に各種パラメータを設定変更すると、通信エラーの発生する場合があるためご注意ください。

5.3.22 センサ種別(W-30016)



概要

機種 EMU4 用のセンサ種別を設定します。

詳細

1. 設定するセンサ種別を選択します。
2. 確認ダイアログウィンドウが表示され、OK を押下すると、入力が確定します。
3. 設定した値を破棄し、ウィンドウを閉じます。

備考

5.3.23 電流、電流デマンド　乗率(W-30017)

概要

機種 ME110 用の電流・電流デマンド　乗率を設定します。

詳細

1. 設定する乗率を選択します。
2. 確認ダイアログウィンドウが表示され、OK を押下すると、入力が確定します。
3. 設定した値を破棄し、ウィンドウを閉じます。

備考

5.3.24 電圧　乗率(W-30018)

概要

機種 ME110 用の電圧　乗率を設定します。

詳細

1. 設定する乗率を選択します。
2. 確認ダイアログウィンドウが表示され、OK を押下すると、入力が確定します。
3. 設定した値を破棄し、ウィンドウを閉じます。

備考

5.3.25 電力、電力デマンド、無効電力　乗率(W-30019)

概要

機種 ME110 用の電力・電力デマンド・無効電力　乗率を設定します。

詳細

1. 設定する乗率を選択します。
2. 確認ダイアログウィンドウが表示され、OK を押下すると、入力が確定します。
3. 設定した値を破棄し、ウィンドウを閉じます。

備考

5.3.26 電力量、無効電力量　乗率(W-30020)

概要

機種 ME110 用の電力量、無効電力量　乗率を設定します。

詳細

1. 設定する乗率を選択します。
2. 確認ダイアログウィンドウが表示され、OK を押下すると、入力が確定します。
3. 設定した値を破棄し、ウィンドウを閉じます。

備考

5.4 使用デバイス一覧

画面上のスイッチやランプなどに設定されている一部のデバイスは、スクリプトなどの共通設定にも設定されている場合があります。これらのデバイスを一括で変更する場合には[一括変更]の使用を推奨します。[一括変更]の詳細については、「GT Designer3 Version1 画面設計マニュアル(共通編)」を参照してください。

5.4.1 接続機器のデバイス

タイプ	デバイス番号	用途	機種 EMU4	機種 ME110
ビット	400525.b2	電流デマンド上下限警報	○	○
	400525.b3	電力デマンド上下限警報	○	○
	400525.b8	電圧上下限警報	○	○
	400525.b9	電流上下限警報	－	○
	400525.b10	電力上下限警報	－	○
	400525.b11	無効電力上下限警報	－	○
	400525.b12	周波数上下限警報	－	○
	400525.b13	力率上下限警報	○	○
	400525.b14	高調波電圧上限警報	－	○
	400525.b15	高調波電流上限警報	－	○
ワード	400513	相線式	○	○
	400514	一次電圧(線間電圧)	○	○
	400516	一次電圧(相電圧)	○	○
	400520	一次電流	○	○
	400522	電力デマンド時限	○	○
	400523	電流デマンド時限	○	○
	400532	センサ種別	○	－
	400755	電流、電流デマンド 乗率	○	－
	400756	電圧 乗率	○	－
	400757	電力、電力デマンド、無効電力 乗率	○	－
	400758	電力量、無効電力量 乗率	○	－
	400769	1 相電流	○	○
	400770	2 相電流	○	○
	400771	3 相電流	○	○
	400773	電流現在値(平均)	○	○
	400774	1 相電流デマンド	○	○
	400775	2 相電流デマンド	○	○
	400776	3 相電流デマンド	○	○
	400779	1-2 線間電圧	○	○
	400780	2-3 線間電圧	○	○
	400781	3-1 線間電圧	○	○
	400790	力率	○	○
	400791	周波数	○	○
	400795	電力	○	○
	400799	電力デマンド	○	○
	400803	無効電力	○	○
	401305	電力量	○	○
	401309	無効電力量	○	○

○:デバイス有り、－:デバイス無し

5.4.2　GOT の内部デバイス

タイプ	デバイス番号	用途
ビット	GB40	スクリプトトリガ(常時 ON)
	GB41	デバイスデータ転送トリガ OFF トリガ(常時 OFF)
	GD600031.b13	GOT エラーリセット信号
	GS512.b0	時刻変更信号
	GB61000	ロギング ID:30001　トリガデバイス
	GB61001	ロギング ID:30002　トリガデバイス
	GB61002	ロギング ID:30003　トリガデバイス
	GB61003	ロギング ID:30004　トリガデバイス
	GB61004	ロギング ID:30005　トリガデバイス
	GB61005	ロギング ID:30006　トリガデバイス
	GB61006	ロギング ID:30007　トリガデバイス
	GB61007	スクリプト No.30004　起動トリガ
	GB61008	スクリプト No.30005　起動トリガ
	GB61009	書き込み完了デバイス
	GB61010	書き込み完了デバイス(電力量グラフ上限値)
	GB61011	スクリプト No.30004　トリガデバイス
	GB61012	スクリプト No.30020　トリガデバイス
	GB61013	スクリプト No.30021　トリガデバイス
	GB61014	スクリプト No.30025　トリガデバイス
	GB61015	スクリプト No.30025　起動フラグ
	GB61016	一次電圧(線間電圧、相電圧)表示/非表示フラグ
	GB61017	ロギング用デバイスデータ転送実施タイミング制御フラグ
	GS512.b0	時計変更信号
ワード	GD10	局番間接指定デバイス
	GD60000	ベース画面切り換え
	GD60001	オーバーラップウィンドウ 1 画面切り換え
	GD60004	オーバーラップウィンドウ 2 画面切り換え
	GD60018	ダイアログウィンドウ　画面切り換え
	GD60021	言語切り換え
	GD60022	システム言語切り換え
	GD60031、GD60041	システム情報
	GD61000	局番 1　名称
	GD61001	局番 1　名称
	GD61002	局番 1　名称
	GD61003	局番 1　名称
	GD61004	局番 1　名称
	GD61005	局番 1　名称
	GD61006	局番 1　機種
	GD61007	局番 1 CO2 排出係数
	GD61008	局番 1 CO2 排出係数
	GD61009	局番 1　電気料金単価
	GD61010	局番 1　電気料金単価
	GD61011	局番 1　電流・電流デマンド　乗率
	GD61012	局番 1　電圧　乗率
	GD61013	局番 1　電力・電力デマンド・無効電力　乗率
	GD61014	局番 1　電力量・無効電力量　乗率
	GD61015	局番 1　力率　乗率
	GD61016	局番 1　周波数　乗率

タイプ	デバイス番号	用途
ワード	GD61017	局番 1 電力量グラフ最大値
	GD61018	局番 1 電力量グラフ最大値
	GD61019	局番 1 総合電流グラフ最大値
	GD61020	局番 1 総合電流グラフ最大値
	GD61025～GD61045	局番 2 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61050～GD61070	局番 3 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61075～GD61095	局番 4 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61100～GD61120	局番 5 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61125～GD61145	局番 6 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61150～GD61170	局番 7 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61175～GD61195	局番 8 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61200～GD61220	局番 9 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61225～GD61245	局番 10 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61250～GD61270	局番 11 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61275～GD61295	局番 12 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61300～GD61320	局番 13 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61325～GD61345	局番 14 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61350～GD61370	局番 15 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61375～GD61395	局番 16 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61400～GD61420	局番 17 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61425～GD61445	局番 18 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61450～GD61470	局番 19 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61475～GD61495	局番 20 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61500～GD61520	局番 21 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61525～GD61545	局番 22 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61550～GD61570	局番 23 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61575～GD61595	局番 24 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61600～GD61620	局番 25 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61625～GD61645	局番 26 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61650～GD61670	局番 27 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61675～GD61695	局番 28 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61700～GD61720	局番 29 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61725～GD61745	局番 30 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61750～GD61770	局番 31 各種パラメータ（局番1 名称～センサ種別を参照）
	GD61775～GD61836	局番1～31 電力量ロギングデバイス
	GD61837～GD61898	局番1～31 総合電流ロギングデバイス
	GD61899～GD61960	局番1～31 電力量 デバイスデータ転送先
	GD61961～GD61991	局番1～31 総合電流 デバイスデータ転送先
	GD61992～GD62022	局番1～31 総合電流 乗率 デバイスデータ転送先
	GD62023～GD62053	局番1～31 電力量 乗率 デバイスデータ転送先
	GD62100、GD62101	現在値モニタ 1 相電流
	GD62102、GD62103	現在値モニタ 2 相電流
	GD62104、GD62105	現在値モニタ 3 相電流
	GD62106、GD62107	現在値モニタ 1 相電流デマンド
	GD62108、GD62109	現在値モニタ 2 相電流デマンド
	GD62110、GD62111	現在値モニタ 3 相電流デマンド

タイプ	デバイス番号	用途
ワード	GD62112、GD62113	現在値モニタ 1-2 線間電圧
	GD62114、GD62115	現在値モニタ 2-3 線間電圧
	GD62116、GD62117	現在値モニタ 3-1 線間電圧
	GD62118、GD62119	現在値モニタ 電力
	GD62120、GD62121	現在値モニタ 電力デマンド
	GD62122、GD62123	現在値モニタ 無効電力
	GD62124、GD62125	現在値モニタ 電力量
	GD62126、GD62127	現在値モニタ 無効電力量
	GD62130、GD62131	現在値モニタ CO2 排出量
	GD62132、GD62133	現在値モニタ 電気料金
	GD62139	パラメータオフセットデバイス
	GD62140	現在機種格納デバイス
	GD62141	拡張レシピ共通設定 外部制御デバイス
	GD62142	拡張レシピ共通設定 レシピ No.格納デバイス
	GD62143	拡張レシピ共通設定 レコード No.格納デバイス
	GD62144	拡張レシピ共通設定 外部通知デバイス
	GD62145	拡張レシピ共通設定 レシピ No.通知デバイス
	GD62146	拡張レシピ共通設定 レコード No.通知デバイス
	GD62147	拡張レシピ No.30001 書き込み/読み出しトリガ
	GD62148	拡張レシピ No.30001 レコード No.
	GD62149	デバイスデータ転送 ID:201 外部制御デバイス
	GD62150	デバイスデータ転送 ID:201 外部通知デバイス
	GD62151	電力量グラフ ロギング ID デバイス
	GD62152	電力量グラフ グラフオフセットデバイス
	GD62153、GD62154	電力量グラフ 上限値デバイス
	GD62155	電力量グラフ カーソル位置時刻
	GD62156	電力量グラフ カーソル位置時刻
	GD62157	電力量グラフ カーソル位置時刻
	GD62158	電力量グラフ カーソル位置時刻
	GD62159	電力量グラフ 表示開始位置時刻
	GD62160	電力量グラフ 表示開始位置時刻
	GD62161	電力量グラフ 表示開始位置時刻
	GD62162	電力量グラフ 表示開始位置時刻
	GD62163	電力量グラフ 表示終了位置時刻
	GD62164	電力量グラフ 表示終了位置時刻
	GD62165	電力量グラフ 表示終了位置時刻
	GD62166	電力量グラフ 表示終了位置時刻
	GD62167、GD62168	電力量グラフ カーソル位置デバイス値
	GD62169	総合電流グラフ ロギング ID デバイス
	GD62170	総合電流グラフ グラフオフセットデバイス
	GD62171、GD62172	総合電流グラフ 上限値デバイス
	GD62173	総合電流グラフ カーソル位置時刻
	GD62174	総合電流グラフ カーソル位置時刻
	GD62175	総合電流グラフ カーソル位置時刻
	GD62176	総合電流グラフ カーソル位置時刻
	GD62177	総合電流グラフ 表示開始位置時刻
	GD62178	総合電流グラフ 表示開始位置時刻
	GD62179	総合電流グラフ 表示開始位置時刻
	GD62180	総合電流グラフ 表示開始位置時刻

タイプ	デバイス番号	用途
ワード	GD62181	総合電流グラフ　表示終了位置時刻
	GD62182	総合電流グラフ　表示終了位置時刻
	GD62183	総合電流グラフ　表示終了位置時刻
	GD62184	総合電流グラフ　表示終了位置時刻
	GD62185、GD62186	総合電流グラフ　カーソル位置デバイス値
	GD62187	ロギングタイマーパラメータ
	GD62188	相線式　書き込みワークデバイス
	GD62189	センサ種別　書き込みワークデバイス
	GD62190	電流乗率　書き込みワークデバイス
	GD62191	電圧乗率　書き込みワークデバイス
	GD62192	電力乗率　書き込みワークデバイス
	GD62193	電力量乗率　書き込みワークデバイス
	GD62194	パラメータ書き込み処理フラグ
	GD62195	相線式　表示デバイス
	GD62196	電流乗率　表示デバイス
	GD62197	電圧乗率　表示デバイス
	GD62198	電力乗率　表示デバイス
	GD62199	電力量乗率　表示デバイス
	GD62200	拡張ユーザアラーム　コメント番号
	GD62201、GD62202	拡張ユーザアラーム　発生日付
	GD62203、GD62204	拡張ユーザアラーム　発生時刻
	GD62205、GD62206	拡張ユーザアラーム　復旧日付
	GD62207、GD62208	拡張ユーザアラーム　復旧時刻
	GD62209	デバイスデータ転送 ID:202　外部制御デバイス
	GD62210	デバイスデータ転送 ID:202　外部通知デバイス
	GD62211、GD62212	電力量グラフ　現在値
	GD62213、GD62214	総合電流グラフ　現在値
	GD62219	一次電圧(線間電圧)　オフセットデバイス
	GD62220	一次電圧(相電圧)　オフセットデバイス
	GD63990～GD63995	時計のデジスイッチ
	GS513～GS516	変更時刻
	GS650～GS652	現在時刻
	GS531、GS532	モニタ局切断(CH1)
	GS573	MODBUS®通信コントロール機能(共通)
	TMP800～TMP803	プロジェクト、画面スクリプト演算用
	TMP950～TMP996	時計設定画面　オブジェクトスクリプト演算用

5.5 コメント一覧

コメントグループ No.	コメント No.	ベース/ウィンドウ画面 No.
254	No.1～319	B-30011(拡張ユーザアラーム履歴 ID 30001)
255	No.1	B-30001～30011、B-30020～30024
	No.2	B-30001～30011、B-30020～30024
	No.3	B-30002～30011、B-30020～30024
	No.4	B-30001～30011、B-30020～30024
	No.5	B-30009
	No.7	B-30001～30011、B-30020～30024
	No.8	B-30009、B-30010

コメントグループ No.	コメント No.	ベース/ウィンドウ画面 No.
255	No.9	B-30011
	No.10	B-30020、W-30015
	No.11	B-30020
	No.12	B-30021
	No.13	B-30021
	No.14	B-30022
	No.15	B-30020
	No.16	B-30022
	No.17	B-30022
	No.18	B-30002～30008、B-30020～30024
	No.19	B-30020～30024
	No.20	B-30002
	No.21	B-30002
	No.22	B-30002
	No.23	B-30004
	No.24	B-30004
	No.25	B-30004
	No.26	B-30005
	No.27	B-30005
	No.28	B-30006
	No.29	B-30006
	No.30	B-30007
	No.31	B-30007
	No.32	B-30003
	No.33	B-30003
	No.34	B-30003
	No.35	B-30005
	No.36	B-30008
	No.37	B-30008
	No.40	B-30002
	No.41	B-30003
	No.42	B-30004
	No.43	B-30005
	No.44	B-30006
	No.45	B-30007
	No.46	B-30008
	No.47	B-30009
	No.48	B-30010
	No.49	B-30002～30010
	No.50	B-30009、B-30010
	No.51	B-30020
	No.52	B-30021
	No.53	B-30022
	No.54	B-30023
	No.55	B-30024
	No.57	B-30001
	No.59	B-30001
	No.60	B-30010
	No.61	B-30010

コメントグループ No.	コメント No.	ベース/ウィンドウ画面 No.
255	No.62	B-30010
	No.63	B-30009、B-30010
	No.64	B-30009、B-30010
	No.68	B-30009
	No.70	B-30001～30010、B-30020～30024
	No.71	B-30001～30010、B-30020～30024
	No.73	B-30008、B-30022
	No.75	B-30011
	No.76	B-30011
	No.79	B-30021、W-30016
	No.80	B-30023、W-30017
	No.81	B-30023、W-30018
	No.82	B-30023、W-30019
	No.83	B-30024、W-30020
	No.84	B-30024
	No.85	B-30024
	No.88	W-30002
	No.89	W-30002
	No.90	W-30002
	No.91	W-30002
	No.92	W-30002
	No.93	W-30002
	No.94	W-30002
	No.95	W-30002
	No.96	W-30002
	No.98	W-30001
	No.99	W-30001
	No.210	W-30003
	No.211	W-30003
	No.212	W-30003
	No.213	W-30003
	No.214	W-30003
	No.215	W-30003、W-30015～30020
	No.216	W-30003、W-30015～30020
	No.217	B-30020、B-30023、B-30024
	No.300	W-30003
	No.301	W-30003
	No.303	B-30020、W-30015
	No.304	B-30020、W-30015
	No.305	B-30020、W-30015
	No.306	B-30020、W-30015
	No.307	B-30020、W-30015
	No.308	B-30020
	No.309	B-30020
	No.310	B-30021、W-30016
	No.311	B-30021、W-30016
	No.312	B-30023、W-30019
	No.313	B-30023、W-30019
	No.314	B-30023、B-30024、W-30017～30020

コメントグループ No.	コメント No.	ベース/ウィンドウ画面 No.
255	No.315	B-30023、B-30024、W-30017～30020
	No.316	B-30023、B-30024、W-30017～30020
	No.317	B-30023、B-30024、W-30017～30020
	No.318	B-30023、B-30024、W-30018～30020
	No.319	B-30024、W-30020
	No.400	W-30014
	No.401	W-30014～30020
	No.402	W-30014～30020

5.6 スクリプト一覧

項　目	設定
プロジェクトスクリプト	有り
画面スクリプト	有り：B-30001、B-30002、B-30003、B-30004、B-30005、B-30006、B-30007、B-30008 、B-30009 、B-30010 、B-30011 、B-30020 、B-30021 、B-30022 、B-30023、B-30024
オブジェクトスクリプト	有り：B-30002、B-30003、B-30004、B-30005、B-30006、B-30008、B-30009、B-30010、W-30002

5.6.1 プロジェクトスクリプト

スクリプト No.	30001	スクリプト名	Script30001
コメント	画面起動時の処理		
データ形式	符号付き BIN16	トリガ種別	立上り　GB40

```
//画面起動時に実行する処理です

//パラメータ書き込み
[w:GD62148] = 1; //レコード No.
set([b:GD62147.b0]);

[w:GS573] = 2;//保持レジスタ読み出し最大数指定
```

スクリプト No.	30002	スクリプト名	Script30002
コメント	拡張レシピ書込み完了		
データ形式	符号付き BIN16	トリガ種別	ON 中　GD62144.b4

```
//拡張レシピの書き込み完了後の処理です。

//フラグのクリア
rst([b:GD62147.b0]);

//一番若い有効局番を探します。
[u32:TMP0800] = [u32:GS531] >> 1;    //0bit 目は未使用なので、あらかじめ 1bit 分右へシフトします。
[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;

}

//全局無効の場合は、1 を設定します。
if([w:TMP0802] == 32)
{
    [w:GD10] = 1;
}

//画面起動時設定起動
set([b:GB61013]);

rst([b:GD62144.b4]);
```

スクリプト No.	30003	スクリプト名	Script30003
コメント	オフセットを設定		
データ形式	符号付き BIN16	トリガ種別	常時

```
//現在の局番から、オフセットを求めます。
[w:GD62139] = ([w:GD10] - 1) * 25;
```

スクリプト No.	30004	スクリプト名	Script30004
コメント	ロギング関連 1:デバイスデータ転送開始		
データ形式	符号付き BIN16	トリガ種別	ON 中　GB61011

```
//ロギングをする対象のデータをデバイスデータ転送で取得します。

//1 分毎
if(((([w:GS652] & 0x00FF) == 0) && ([b:GB61017] == OFF))
{
    set([b:GB61017]);//1 秒で複数回実行されないよう、フラグを立てます。
    //1 時間毎
    if((([w:GS652] & 0xFF00) >> 8) == 0)
    {
        //1 日毎
        if(([w:GS651] & 0x00FF) == 0)
        {
            //1 ヶ月毎
            if((([w:GS651] & 0xFF00) >> 8)     == 0x0001)
            {
                //デバイスデータ転送開始
                [w:GD62187] = 5;
                if([w:GD62140] < 1)
                {
                    set([b:GD62149.b0]);
                }else{
                    set([b:GD62209.b0]);
                }
                return;
            }

            //デバイスデータ転送開始
            [w:GD62187] = 4;
            if([w:GD62140] < 1)
            {
                set([b:GD62149.b0]);
            }else{
                set([b:GD62209.b0]);
```

```
            }
            return;
        }

        //デバイスデータ転送開始
        [w:GD62187] = 3;
        if([w:GD62140] < 1)
        {
            set([b:GD62149.b0]);
        }else{
            set([b:GD62209.b0]);
        }
        return;

    }else{
        //30 分毎
        if((([w:GS652] & 0xFF00) >> 8) == 0x0030)
        {
            //デバイスデータ転送開始
            [w:GD62187] = 2;
            if([w:GD62140] < 1)
            {
                set([b:GD62149.b0]);
            }else{
                set([b:GD62209.b0]);
            }
            return;
        }
    }

    //デバイスデータ転送開始
    [w:GD62187] = 1;
    if([w:GD62140] < 1)
    {
        set([b:GD62149.b0]);
    }else{
        set([b:GD62209.b0]);
    }
    return;
}else{
    if((([w:GS652] & 0x00FF) != 0) && ([b:GB61017] == ON))
    {
        rst([b:GB61017]); //00 秒を過ぎたら、フラグをリセットします。
    }
}
```

スクリプト No.	30007	スクリプト名	Script30007
コメント	ロギングトリガ OFF(No.30001)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61000

//ロギング No.30001(電力量(1 時間))のトリガを OFF します。
rst([b:GB61000]);			

スクリプト No.	30008	スクリプト名	Script30008
コメント	ロギングトリガ OFF(No.30002)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61001

//ロギング No.30002(電力量(1 日))のトリガを OFF します。
rst([b:GB61001]);			

スクリプト No.	30009	スクリプト名	Script30009
コメント	ロギングトリガ OFF(No.30003)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61002

//ロギング No.30003(電力量(1 ヶ月))のトリガを OFF します。
rst([b:GB61002]);			

スクリプト No.	30010	スクリプト名	Script30010
コメント	ロギングトリガ OFF(No.30004)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61003

//ロギング No.30004(総合電流(1 分))のトリガを OFF します。
rst([b:GB61003]);			

スクリプト No.	30011	スクリプト名	Script30011
コメント	ロギングトリガ OFF(No.30005)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61004

//ロギング No.30005(総合電流(30 分))のトリガを OFF します。
rst([b:GB61004]);			

スクリプト No.	30012	スクリプト名	Script30012
コメント	ロギングトリガ OFF(No.30006)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61005

//ロギング No.30006(総合電流(1 時間))のトリガを OFF します。
rst([b:GB61005]);			

スクリプト No.	30013	スクリプト名	Script30013
コメント	ロギングトリガ OFF(No.30007)		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GB61006

//ロギング No.30007(総合電流(1 日))のトリガを OFF します。
rst([b:GB61006]);			

スクリプト No.	30014	スクリプト名	Script30014
コメント	入力完了後、レシピへ読み出す		
データ形式	符号付き BIN16	トリガ種別	立上り GB61009

//各種オブジェクトで、入力を実施後、拡張レシピへの読み出しを実施します。
set([b:GD62147.b1]);			

スクリプト No.	30015	スクリプト名	Script30015
コメント	拡張レシピ読み出し完了後処理		
データ形式	符号付き BIN16	トリガ種別	ON 中 GD62144.b5

//拡張レシピの読み出し完了後、トリガを OFF します。
//局番の有効/無効の設定後は、無効になった局番の
//ロギングデータをクリアするためのスクリプトを起動します。			

```
if([b:GB61015] == ON)
{
    set([b:GB61014]);
    rst([b:GB61015]);
}

rst([b:GD62147.b1]);
rst([b:GD62144.b5]);
```

スクリプト No.	30018	スクリプト名	Script30018
コメント	デバイスデータ転送 1 トリガ OFF		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GD62149.b0

```
//デバイスデータ転送のトリガを定期的に監視して、リセットします。

rst([b:GD62149.b0]);
```

スクリプト No.	30024	スクリプト名	Script30024
コメント	ロギングディレイ		
データ形式	符号付き BIN16	トリガ種別	OFF 中周期 2 秒 GB61011

```
//画面起動時より 2 秒待ってから、ロギング用スクリプトを起動します。
set([b:GB61011]);
```

スクリプト No.	30026	スクリプト名	Script30026
コメント	デバイスデータ転送 2 トリガ OFF		
データ形式	符号付き BIN16	トリガ種別	ON 中周期 1 秒 GD62209.b0

```
//デバイスデータ転送のトリガを定期的に監視して、リセットします。

rst([b:GD62209.b0]);
```

5.6.2 画面スクリプト

ベース画面 共通

スクリプト No.	30005	スクリプト名	Script30005
コメント	ロギング関係 2:総合電流の演算		
データ形式	符号付き BIN16	トリガ種別	ON 中 GB61007

```
//デバイスデータ転送した総合電流のデータに演算をします。

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

while([w:TMP0800] < 31)
{
    if([w:GD61006[w:TMP0801]] == 0)
    {
        //EMU4
        switch([s16:GD61992[w:TMP0800]])
        {
            case -3: [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]] * 0.001;
                     break;

            case -2: [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]] * 0.01;
                     break;

            case -1: [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]] * 0.1;
                     break;

            case 0:      [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]];
```

```
                    break;
        }

    }else{
        //ME110
        switch([w:GD61011[w:TMP0801]])
        {
            case 1:     [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]] * 0.01 ;
                    break;

            case 2:     [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]] * 0.1;
                    break;

            case 3:     [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]];
                    break;

            case 4:     [flt:GD61837[w:TMP0802]] = [s16:GD61961[w:TMP0800]] * 10;
                    break;
        }

    }

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

}

//どのロギングをするか選択
switch([w:GD62187])
{
    case 1:     set([b:GB61003]);//1 分毎
            break;

    case 2:     set([b:GB61003]);//1 分毎
            set([b:GB61004]);//30 分毎
            break;

    case 3:     set([b:GB61003]);//1 分毎
            set([b:GB61004]);//30 分毎
            set([b:GB61005]);//1 時間毎
            break;

    case 4:     set([b:GB61003]);//1 分毎
            set([b:GB61004]);//30 分毎
            set([b:GB61005]);//1 時間毎
            set([b:GB61006]);//1 日毎
            break;

    case 5:     set([b:GB61003]);//1 分毎
            set([b:GB61004]);//30 分毎
            set([b:GB61005]);//1 時間毎
            set([b:GB61006]);//1 日毎
            break;

    default:    break;
```

```
}

set([b:GB61008]);//電力量の演算スクリプト起動

rst([b:GB61007]);
```

スクリプト No.	30006	スクリプト名	Script30006
コメント	ロギング関係 2:電力量の演算		
データ形式	符号付き BIN16	トリガ種別	ON 中　GB61008

```
//デバイスデータ転送した電力量のデータに演算をします。

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

//電力量は、1 時間、1 日、1 ヶ月ごとのみ、ロギング
if([w:GD62187] >= 3)
{
    while([w:TMP0800] <= 31)
    {
        if([w:GD61006[w:TMP0801]] == 0)
        {
            //EMU4
            switch([w:GD62023[w:TMP0800]])
            {
                case -2: [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 0.01;
                         break;

                case -1: [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 0.1;
                         break;

                case 0:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]];
                         break;

                case 1:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 10;
                         break;

                case 2:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 100;
                         break;
            }
        }else{
            //ME110
            switch([w:GD61014[w:TMP0801]])
            {
                case 1:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 0.01 ;
                         break;

                case 2:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 0.1;
                         break;

                case 3:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]];
                         break;

                case 4:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 10;
                         break;

                case 5:      [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 100;
```

```
                    break;

            case 6:     [flt:GD61775[w:TMP0802]] = [s32:GD61899[w:TMP0800]] * 1000;
                    break;
        }

    }

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

}

//どのロギングをするか選択
switch([w:GD62187])
{
    case 3:     set([b:GB61000]);//1 時間毎
            break;

    case 4:     set([b:GB61000]);//1 時間毎
            set([b:GB61001]);//1 日毎
            break;

    case 5:     set([b:GB61000]);//1 時間毎
            set([b:GB61001]);//1 日毎
            set([b:GB61002]);//1 ヶ月毎
            break;

    default:    break;
}
}

rst([b:GB61008]);
```

スクリプト No.	30019	スクリプト名	Script30019
コメント	現在の機種を格納		
データ形式	符号付き BIN16	トリガ種別	常時

```
//現在の機種を格納します。
[w:GD62140] = [w:GD61006[w:GD62139]];
```

スクリプト No.	30021	スクリプト名	Script30021
コメント	各種パラメータをワークエリアに転送		
データ形式	符号付き BIN16	トリガ種別	ON 中　GB61013

```
//各種パラメータのワークエリアに、レシピから読み込んだ値を転送します。

//電流　乗率
[w:GD62196] = [w:GD61011[w:GD62139]];

//電圧　乗率
[w:GD62197] = [w:GD61012[w:GD62139]];

//電力　乗率
[w:GD62198] = [w:GD61013[w:GD62139]];

//電力量　乗率
[w:GD62199] = [w:GD61014[w:GD62139]];
```

```
//グラフ上限値(電力量)
[flt:GD62153] = [flt:GD61017[w:GD62139]];

//グラフ上限値(総合電流)
[flt:GD62171] = ([00-FF:s32:400520] * 0.1) * 1.1;

rst([b:GB61013]);
```

スクリプト No.	30027	スクリプト名	Script30027
コメント	一次電流(線間電圧、相電圧)表示・非表示設定		
データ形式	符号付き BIN16	トリガ種別	常時

```
//パラメータの一次電流(線間電圧、相電圧)を条件によって表示/非表示を切り換えます。

if([00-FF:w:400513] == 4)
{
    set([b:GB61016]);
    [w:GD62219] = 2; //線間電圧オフセット
    [w:GD62220] = 0; //相電圧オフセット
}else{
    rst([b:GB61016]);
    [w:GD62219] = 0; //線間電圧オフセット
    [w:GD62220] = 2; //相電圧オフセット
}
```

ベース画面 30001

スクリプト No.	30025	スクリプト名	Script30025
コメント	無効局ロギングデータクリア		
データ形式	符号なし BIN16	トリガ種別	ON 中　GB61014

```
//無効になった局番のロギングデータをクリアします。

[u32:TMP0800] = [u32:GS531] >> 1;    //0bit 目は未使用なので、あらかじめ 1bit 分右へシフトします。
[w:TMP0802] = 1;
[w:TMP0803] = 0;

while([w:TMP0802] < 32)
{
    if(([u32:TMP0800] & 0x00000001) == 1)
    {
        [flt:GD61775[w:TMP0803]] = 0;   //電力量
        [flt:GD61837[w:TMP0803]] = 0;   //総合電流
    }

    [u32:TMP0800] = [u32:TMP0800] >> 1;
    [w:TMP0802] = [w:TMP0802] + 1;
    [w:TMP0803] = [w:TMP0803] + 2;
}

rst([b:GB61014]);
```

ベース画面 30009

スクリプト No.	30017	スクリプト名	Script30017
コメント	電力量グラフ　上限値保存		
データ形式	符号付き BIN16	トリガ種別	立上り　GB61010

```
//電力量グラフの上限値を保存します。
[flt:GD61017] = [flt:GD62153];

set([b:GD62147.b1]);
```

ベース画面 30010

スクリプト No.	30023	スクリプト名	Script30023
コメント	総合電流グラフ　上限値保存		
データ形式	符号付き BIN16	トリガ種別	立上り　GB61010

```
//総合電流グラフの上限値を保存します。
[flt:GD61019] = [flt:GD62171];

set([b:GD62147.b1]);
```

ベース画面 30020、30021、30022、30023、30024

スクリプト No.	30020	スクリプト名	Script30020
コメント	パラメータ書き込み		
データ形式	符号付き BIN16	トリガ種別	ON 中　GB61012

```
//フラグに従い、パラメータを EcoMonitor に書き込みます。

switch([w:GD62194])
{
    case 1:  //相線式のデバイス(400513)に、値を書き込みます。
            [0-FF:w:400513] = [w:GD62188];
            break;

    case 2:  //センサ種別のデバイス(400532)に、値を書き込みます。
            [0-FF:w:400532] = [w:GD62189];
            break;

    case 3:  //電流　乗率のデバイス(GD61011)に、値を書き込みます。
            [w:GD61011[w:GD62139]] = [w:GD62190];
            [w:GD62196] = [w:GD62190];
            break;

    case 4:  //電圧　乗率のデバイス(GD61012)に、値を書き込みます。
            [w:GD61012[w:GD62139]] = [w:GD62191];
            [w:GD62197] = [w:GD62191];
            break;

    case 5:  //電力　乗率のデバイス(GD61013)に、値を書き込みます。
            [w:GD61013[w:GD62139]] = [w:GD62192];
            [w:GD62198] = [w:GD62192];
            break;

    case 6:  //電力量　乗率のデバイス(GD61014)に、値を書き込みます。
            [w:GD61014[w:GD62139]] = [w:GD62193];
            [w:GD62199] = [w:GD62193];
            break;
```

```
    default:  break;
}

set([b:GD62147.b1]);
rst([b:GB61012]);
```

5.6.3 オブジェクトスクリプト

ベース画面 30002

オブジェクト(名称)	数値表示(1 相電流_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62100] = [0-FF:s16:400769];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62100] = [0-FF:s16:400769];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62100] = [0-FF:s16:400769];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62100] = [0-FF:s16:400769];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62100] = [0-FF:s16:400769];
             my.decimal_point = 2;
             break;

        case 2:  [s32:GD62100] = [0-FF:s16:400769];
             my.decimal_point = 1;
             break;

        case 3:  [s32:GD62100] = [0-FF:s16:400769];
             my.decimal_point = 0;
             break;

        case 4:  [s32:GD62100] = [0-FF:s16:400769] * 10;
             my.decimal_point = 0;
             break;
    }
```

```
}
redraw_object();
```

オブジェクト(名称)	数値表示(2 相電流_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62102] = [0-FF:s16:400770];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62102] = [0-FF:s16:400770];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62102] = [0-FF:s16:400770];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62102] = [0-FF:s16:400770];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62102] = [0-FF:s16:400770];
               my.decimal_point = 2;
               break;

        case 2:  [s32:GD62102] = [0-FF:s16:400770];
               my.decimal_point = 1;
               break;

        case 3:  [s32:GD62102] = [0-FF:s16:400770];
               my.decimal_point = 0;
               break;

        case 4:  [s32:GD62102] = [0-FF:s16:400770] * 10;
               my.decimal_point = 0;
               break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(3 相電流_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62104] = [0-FF:s16:400771];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62104] = [0-FF:s16:400771];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62104] = [0-FF:s16:400771];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62104] = [0-FF:s16:400771];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62104] = [0-FF:s16:400771];
             my.decimal_point = 2;
             break;

        case 2:  [s32:GD62104] = [0-FF:s16:400771];
             my.decimal_point = 1;
             break;

        case 3:  [s32:GD62104] = [0-FF:s16:400771];
             my.decimal_point = 0;
             break;

        case 4:  [s32:GD62104] = [0-FF:s16:400771] * 10;
             my.decimal_point = 0;
             break;
    }

}
redraw_object();
```

ベース画面 30003

オブジェクト(名称)	数値表示(1 相電流デマンド_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62106] = [0-FF:s16:400774];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62106] = [0-FF:s16:400774];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62106] = [0-FF:s16:400774];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62106] = [0-FF:s16:400774];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62106] = [0-FF:s16:400774];
               my.decimal_point = 2;
               break;

        case 2:  [s32:GD62106] = [0-FF:s16:400774];
               my.decimal_point = 1;
               break;

        case 3:  [s32:GD62106] = [0-FF:s16:400774];
               my.decimal_point = 0;
               break;

        case 4:  [s32:GD62106] = [0-FF:s16:400774] * 10;
               my.decimal_point = 0;
               break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(2 相電流デマンド_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62108] = [0-FF:s16:400775];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62108] = [0-FF:s16:400775];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62108] = [0-FF:s16:400775];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62108] = [0-FF:s16:400775];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62108] = [0-FF:s16:400775];
             my.decimal_point = 2;
             break;

        case 2:  [s32:GD62108] = [0-FF:s16:400775];
             my.decimal_point = 1;
             break;

        case 3:  [s32:GD62108] = [0-FF:s16:400775];
             my.decimal_point = 0;
             break;

        case 4:  [s32:GD62108] = [0-FF:s16:400775] * 10;
             my.decimal_point = 0;
             break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(3 相電流デマンド_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62110] = [0-FF:s16:400776];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62110] = [0-FF:s16:400776];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62110] = [0-FF:s16:400776];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62110] = [0-FF:s16:400776];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62110] = [0-FF:s16:400776];
              my.decimal_point = 2;
              break;

        case 2:  [s32:GD62110] = [0-FF:s16:400776];
              my.decimal_point = 1;
              break;

        case 3:  [s32:GD62110] = [0-FF:s16:400776];
              my.decimal_point = 0;
              break;

        case 4:  [s32:GD62110] = [0-FF:s16:400776] * 10;
              my.decimal_point = 0;
              break;
    }

}
redraw_object();
```

ベース画面 30004

オブジェクト(名称)	数値表示(1-2 線間電圧_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400756])
    {
        case -1: [s32:GD62112] = [0-FF:s16:400779];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62112] = [0-FF:s16:400779];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61012[s16:GD62139]])
    {
        case 1:  [s32:GD62112] = [0-FF:s16:400779];
                my.decimal_point = 1;
                break;

        case 2:  [s32:GD62112] = [0-FF:s16:400779];
                my.decimal_point = 0;
                break;

        case 3:  [s32:GD62112] = [0-FF:s16:400779] * 10;
                my.decimal_point = 0;
                break;

        case 4:  [s32:GD62112] = [0-FF:s16:400779] * 100;
                my.decimal_point = 0;
                break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(2-3 線間電圧_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400756])
    {
        case -1: [s32:GD62114] = [0-FF:s16:400780];
                 my.decimal_point = 1;
```

```
                break;

        case 0:     [s32:GD62114] = [0-FF:s16:400780];
                my.decimal_point = 0;
                break;
    }

}else{
    //ME110
    switch([s16:GD61012[s16:GD62139]])
    {
        case 1:  [s32:GD62114] = [0-FF:s16:400780];
            my.decimal_point = 1;
            break;

        case 2:  [s32:GD62114] = [0-FF:s16:400780];
            my.decimal_point = 0;
            break;

        case 3:  [s32:GD62114] = [0-FF:s16:400780] * 10;
            my.decimal_point = 0;
            break;

        case 4:  [s32:GD62114] = [0-FF:s16:400780] * 100;
            my.decimal_point = 0;
            break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(3-1 線間電圧_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400756])
    {
        case -1: [s32:GD62116] = [0-FF:s16:400781];
                my.decimal_point = 1;
                break;

        case 0:     [s32:GD62116] = [0-FF:s16:400781];
                my.decimal_point = 0;
                break;

    }

}else{
    //ME110
    switch([s16:GD61012[s16:GD62139]])
    {
        case 1:  [s32:GD62116] = [0-FF:s16:400781];
            my.decimal_point = 1;
            break;
```

```
        case 2:  [s32:GD62116] = [0-FF:s16:400781];
                my.decimal_point = 0;
                break;

        case 3:  [s32:GD62116] = [0-FF:s16:400781] * 10;
                my.decimal_point = 0;
                break;

        case 4:  [s32:GD62116] = [0-FF:s16:400781] * 100;
                my.decimal_point = 0;
                break;
    }

}
redraw_object();
```

ベース画面 30005

オブジェクト(名称)	数値表示(電力_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400757])
    {
        case -3: [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 3;
                break;

        case -2: [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 2;
                break;

        case -1: [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 1;
                break;

        case 0:     [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 0;
                break;

        case 1:     [s32:GD62118] = [0-FF:s16:400795] * 10;
                my.decimal_point = 0;
                break;
    }

}else{
    //ME110
    switch([s16:GD61013[s16:GD62139]])
    {
        case 1:  [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 4;
                break;
```

```
        case 2:  [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 3;
                break;

        case 3:  [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 2;
                break;

        case 4:  [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 1;
                break;

        case 5:  [s32:GD62118] = [0-FF:s16:400795];
                my.decimal_point = 0;
                break;

        case 6:  [s32:GD62118] = [0-FF:s16:400795] * 10;
                my.decimal_point = 0;
                break;

        case 7:  [s32:GD62118] = [0-FF:s16:400795] * 100;
                my.decimal_point = 0;
                break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(電力デマンド_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400757])
    {
        case -3: [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 3;
                break;

        case -2: [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 2;
                break;

        case -1: [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 1;
                break;

        case 0:      [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 0;
                break;

        case 1:      [s32:GD62120] = [0-FF:s16:400799] * 10;
                my.decimal_point = 0;
```

```
                break;
    }

}else{
    //ME110
    switch([s16:GD61013[s16:GD62139]])
    {
        case 1:  [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 4;
                break;

        case 2:  [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 3;
                break;

        case 3:  [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 2;
                break;

        case 4:  [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 1;
                break;

        case 5:  [s32:GD62120] = [0-FF:s16:400799];
                my.decimal_point = 0;
                break;

        case 6:  [s32:GD62120] = [0-FF:s16:400799] * 10;
                my.decimal_point = 0;
                break;

        case 7:  [s32:GD62120] = [0-FF:s16:400799] * 100;
                my.decimal_point = 0;
                break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(無効電力_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400757])
    {
        case -3: [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 3;
                break;

        case -2: [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 2;
                break;

        case -1: [s32:GD62122] = [0-FF:s16:400803];
```

```
                my.decimal_point = 1;
                break;

        case 0:     [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 0;
                break;

        case 1:     [s32:GD62122] = [0-FF:s16:400803] * 10;
                my.decimal_point = 0;
                break;
    }

}else{
    //ME110
    switch([s16:GD61013[s16:GD62139]])
    {
        case 1:  [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 4;
                break;

        case 2:  [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 3;
                break;

        case 3:  [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 2;
                break;

        case 4:  [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 1;
                break;

        case 5:  [s32:GD62122] = [0-FF:s16:400803];
                my.decimal_point = 0;
                break;

        case 6:  [s32:GD62122] = [0-FF:s16:400803] * 10;
                my.decimal_point = 0;
                break;

        case 7:  [s32:GD62122] = [0-FF:s16:400803] * 100;
                my.decimal_point = 0;
                break;
    }

}
redraw_object();
```

ベース画面 30006

オブジェクト(名称)	数値表示(電力量_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
```

```
    //EMU4
    switch([0-FF:s16:400758])
    {
        case -2: [s32:GD62124] = [0-FF:s32:401305];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62124] = [0-FF:s32:401305];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62124] = [0-FF:s32:401305];
                 my.decimal_point = 0;
                 break;

        case 1:      [s32:GD62124] = [0-FF:s32:401305] * 10;
                 my.decimal_point = 0;
                 break;

        case 2:      [s32:GD62124] = [0-FF:s32:401305] * 100;
                 my.decimal_point = 0;
                 break;
    }

}else{
    //ME110
    switch([s16:GD61014[s16:GD62139]])
    {
        case 1:  [s32:GD62124] = [0-FF:s32:401305];
             my.decimal_point = 2;
             break;

        case 2:  [s32:GD62124] = [0-FF:s32:401305];
             my.decimal_point = 1;
             break;

        case 3:  [s32:GD62124] = [0-FF:s32:401305];
             my.decimal_point = 0;
             break;

        case 4:  [s32:GD62124] = [0-FF:s32:401305] * 10;
             my.decimal_point = 0;
             break;

        case 5:  [s32:GD62124] = [0-FF:s32:401305] * 100;
             my.decimal_point = 0;
             break;

        case 6:  [s32:GD62124] = [0-FF:s32:401305] * 1000;
             my.decimal_point = 0;
             break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(無効電力量_数値)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400758])
    {
        case -2: [s32:GD62126] = [0-FF:s32:401309];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62126] = [0-FF:s32:401309];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62126] = [0-FF:s32:401309];
                 my.decimal_point = 0;
                 break;

        case 1:      [s32:GD62126] = [0-FF:s32:401309] * 10;
                 my.decimal_point = 0;
                 break;

        case 2:      [s32:GD62126] = [0-FF:s32:401309] * 100;
                 my.decimal_point = 0;
                 break;
    }

}else{
    //ME110
    switch([s16:GD61014[s16:GD62139]])
    {
        case 1:  [s32:GD62126] = [0-FF:s32:401309];
             my.decimal_point = 2;
             break;

        case 2:  [s32:GD62126] = [0-FF:s32:401309];
             my.decimal_point = 1;
             break;

        case 3:  [s32:GD62126] = [0-FF:s32:401309];
             my.decimal_point = 0;
             break;

        case 4:  [s32:GD62126] = [0-FF:s32:401309] * 10;
             my.decimal_point = 0;
             break;

        case 5:  [s32:GD62126] = [0-FF:s32:401309] * 100;
             my.decimal_point = 0;
             break;

        case 6:  [s32:GD62126] = [0-FF:s32:401309] * 1000;
             my.decimal_point = 0;
```

```
            break;
    }

}
redraw_object();
```

ベース画面 30008

オブジェクト(名称)	数値表示(CO2 排出量_数値)		
スクリプトユーザ ID	1		
データ形式	実数	トリガ種別	常時

```
//CO2 排出量を計算します。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400758])
    {
        case -2: [flt:GD62130] = ([0-FF:s32:401305] * 0.01) * [flt:GD61007[w:GD62139]];
                 my.decimal_point = 2;
                 break;

        case -1: [flt:GD62130] = ([0-FF:s32:401305] * 0.1) * [flt:GD61007[w:GD62139]];
                 my.decimal_point = 1;
                 break;

        case 0:      [flt:GD62130] = [0-FF:s32:401305] * [flt:GD61007[w:GD62139]];
                 break;

        case 1:      [flt:GD62130] = ([0-FF:s32:401305] * 10) * [flt:GD61007[w:GD62139]];
                 break;

        case 2:      [flt:GD62130] = ([0-FF:s32:401305] * 100) * [flt:GD61007[w:GD62139]];
                 break;
    }

}else{
    //ME110
    switch([s16:GD61014[s16:GD62139]])
    {
        case 1:  [flt:GD62130] = ([0-FF:s32:401305] * 0.01) * [flt:GD61007[w:GD62139]];
                my.decimal_point = 2;
                break;

        case 2:  [flt:GD62130] = ([0-FF:s32:401305] * 0.1) * [flt:GD61007[w:GD62139]];
                my.decimal_point = 1;
                break;

        case 3:  [flt:GD62130] = [0-FF:s32:401305] * [flt:GD61007[w:GD62139]];
                break;

        case 4:  [flt:GD62130] = ([0-FF:s32:401305] * 10) * [flt:GD61007[w:GD62139]];
                break;

        case 5:  [flt:GD62130] = ([0-FF:s32:401305] * 100) * [flt:GD61007[w:GD62139]];
                break;
```

```
        case 6:  [flt:GD62130] = ([0-FF:s32:401305] * 1000) * [flt:GD61007[w:GD62139]];
                 break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(電気料金_数値)		
スクリプトユーザ ID	1		
データ形式	実数	トリガ種別	常時

```
//電気料金を計算します。
[flt:GD62132] = [0-FF:s32:401305] * [flt:GD61009[w:GD62139]];

if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400758])
    {
        case -2: [flt:GD62132] = ([0-FF:s32:401305] * 0.01) * [flt:GD61009[w:GD62139]];
                 break;

        case -1: [flt:GD62132] = ([0-FF:s32:401305] * 0.1) * [flt:GD61009[w:GD62139]];
                 break;

        case 0:      [flt:GD62132] = [0-FF:s32:401305] * [flt:GD61009[w:GD62139]];
                 break;

        case 1:      [flt:GD62132] = ([0-FF:s32:401305] * 10) * [flt:GD61009[w:GD62139]];
                 break;

        case 2:      [flt:GD62132] = ([0-FF:s32:401305] * 100) * [flt:GD61009[w:GD62139]];
                 break;
    }

}else{
    //ME110
    switch([s16:GD61014[s16:GD62139]])
    {
        case 1:  [flt:GD62132] = ([0-FF:s32:401305] * 0.01) * [flt:GD61009[w:GD62139]];
                 break;

        case 2:  [flt:GD62132] = ([0-FF:s32:401305] * 0.1) * [flt:GD61009[w:GD62139]];
                 break;

        case 3:  [flt:GD62132] = [0-FF:s32:401305] * [flt:GD61009[w:GD62139]];
                 break;

        case 4:  [flt:GD62132] = ([0-FF:s32:401305] * 10) * [flt:GD61009[w:GD62139]];
                 break;

        case 5:  [flt:GD62132] = ([0-FF:s32:401305] * 100) * [flt:GD61009[w:GD62139]];
                 break;

        case 6:  [flt:GD62132] = ([0-FF:s32:401305] * 1000) * [flt:GD61009[w:GD62139]];
                 break;
    }
```

```
}
redraw_object();
```

ベース画面 30009

オブジェクト(名称)	数値表示(現在値_No.1)		
スクリプトユーザ ID	1		
データ形式	符号なし BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400758])
    {
        case -2: [s32:GD62211] = [0-FF:s32:401305];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62211] = [0-FF:s32:401305];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62211] = [0-FF:s32:401305];
                 break;

        case 1:      [s32:GD62211] = [0-FF:s32:401305] * 10;
                 break;

        case 2:      [s32:GD62211] = [0-FF:s32:401305] * 100;
                 break;
    }

}else{
    //ME110
    switch([s16:GD61014[s16:GD62139]])
    {
        case 1:  [s32:GD62211] = [0-FF:s32:401305];
             my.decimal_point = 2;
             break;

        case 2:  [s32:GD62211] = [0-FF:s32:401305];
             my.decimal_point = 1;
             break;

        case 3:  [s32:GD62211] = [0-FF:s32:401305];
             my.decimal_point = 0;
             break;

        case 4:  [s32:GD62211] = [0-FF:s32:401305] * 10;
             break;

        case 5:  [s32:GD62211] = [0-FF:s32:401305] * 100;
             break;

        case 6:  [s32:GD62211] = [0-FF:s32:401305] * 1000;
```

```
                break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(カーソル情報_No.1)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN16	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF: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:GD61014[s16:GD62139]])
    {
        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();
```

ベース画面 30010

オブジェクト(名称)	数値表示(現在値_No.1)		
スクリプトユーザ ID	1		
データ形式	符号なし BIN32	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF:s16:400755])
    {
        case -3: [s32:GD62213] = [0-FF:s16:400773];
                 my.decimal_point = 3;
                 break;

        case -2: [s32:GD62213] = [0-FF:s16:400773];
                 my.decimal_point = 2;
                 break;

        case -1: [s32:GD62213] = [0-FF:s16:400773];
                 my.decimal_point = 1;
                 break;

        case 0:      [s32:GD62213] = [0-FF:s16:400773];
                 my.decimal_point = 0;
                 break;

    }

}else{
    //ME110
    switch([s16:GD61011[s16:GD62139]])
    {
        case 1:  [s32:GD62213] = [0-FF:s16:400773];
               my.decimal_point = 3;
               break;

        case 2:  [s32:GD62213] = [0-FF:s16:400773];
               my.decimal_point = 2;
               break;

        case 3:  [s32:GD62213] = [0-FF:s16:400773];
               my.decimal_point = 1;
               break;

        case 4:  [s32:GD62213] = [0-FF:s16:400773];
               my.decimal_point = 0;
               break;
    }

}
redraw_object();
```

オブジェクト(名称)	数値表示(カーソル情報_No.1)		
スクリプトユーザ ID	1		
データ形式	符号付き BIN16	トリガ種別	常時

```
//小数点の位置を算出しています。
if([s16:GD61006[s16:GD62139]]==0)
{
    //EMU4
    switch([0-FF: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:GD61011[s16:GD62139]])
    {
        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();
```

ウィンドウ画面 30002

オブジェクト(名称)	数値表示(年)		
スクリプトユーザ ID	1		
データ形式	符号なし BIN16	トリガ種別	立上り GB40

```
//時計データより本日の年月を取得
[w:TMP0] = [w:GS650] & 0xF000;//設定用時計データより年の下 2 桁の 10 の位を取得
[w:TMP10] = [w:TMP0] >> 12;//桁合せ
[w:TMP20] = [w:TMP10] * 10;//BCD->BIN
[w:TMP1] = [w:GS650] & 0x0F00;//設定用時計データより年の下 2 桁の 1 の位を取得
[w:TMP11] = [w:TMP1] >> 8;//BCD->BIN
[w:TMP100] = 2000 + [w:TMP20] +  [w:TMP11];//TMP100 に年を BIN でセット
[w:GD63990] = [w:TMP100];//年をセット

[w:TMP2] = [w:GS650] & 0x00F0;//設定用時計データより月の 10 の位を取得
[w:TMP12] = [w:TMP2] >> 4;//桁合せ
[w:TMP22] = [w:TMP12] * 10;//BCD->BIN
[w:TMP3] = [w:GS650] & 0x000F;//設定用時計データより月の 1 の位を取得
[w:TMP101] = [w:TMP22] +  [w:TMP3];//TMP101 に月を BIN でセット
[w:GD63991] = [w:TMP101];//月をセット

[w:TMP4] = [w:GS651] & 0xF000;//設定用時計データより日の下 2 桁の 10 の位を取得
[w:TMP14] = [w:TMP4] >> 12;//桁合せ
[w:TMP24] = [w:TMP14] * 10;//BCD->BIN
[w:TMP5] = [w:GS651] & 0x0F00;//設定用時計データより日の下 2 桁の 1 の位を取得
[w:TMP15] = [w:TMP5] >> 8;//BCD->BIN
[w:TMP102] =[w:TMP24] +  [w:TMP15];//TMP102 に日を BIN でセット
[w:GD63992] = [w:TMP102];//日をセット

[w:TMP6] = [w:GS651] & 0x00F0;//設定用時計データより時の 10 の位を取得
[w:TMP16] = [w:TMP6] >> 4;//桁合せ
[w:TMP26] = [w:TMP16] * 10;//BCD->BIN
[w:TMP7] = [w:GS651] & 0x000F;//設定用時計データより時の 1 の位を取得
[w:TMP103] = [w:TMP26] +  [w:TMP7];//TMP103 に時を BIN でセット
[w:GD63993] = [w:TMP103];//時をセット

[w:TMP8] = [w:GS652] & 0xF000;//設定用時計データより分の下 2 桁の 10 の位を取得
[w:TMP18] = [w:TMP8] >> 12;//桁合せ
[w:TMP28] = [w:TMP18] * 10;//BCD->BIN
[w:TMP9] = [w:GS652] & 0x0F00;//設定用時計データより分の下 2 桁の 1 の位を取得
[w:TMP19] = [w:TMP9] >> 8;//BCD->BIN
[w:TMP104] =[w:TMP28] +  [w:TMP19];//TMP104 に分を BIN でセット
[w:GD63994] = [w:TMP104];//分をセット

[w:TMP1000] = [w:GS652] & 0x00F0;//設定用時計データより秒の 10 の位を取得
[w:TMP1010] = [w:TMP1000] >> 4;//桁合せ
[w:TMP1020] = [w:TMP1010] * 10;//BCD->BIN
[w:TMP1001] = [w:GS652] & 0x000F;//設定用時計データより秒の 1 の位を取得
[w:TMP105] = [w:TMP1020] +  [w:TMP1001];//TMP105 に秒を BIN でセット
[w:GD63995] = [w:TMP105];//秒をセット
```

オブジェクト(名称)	数値表示(月)		
スクリプトユーザ ID	2		
データ形式	符号なし BIN16	トリガ種別	常時

```
// BIN -> BCD 変換
```

```
[w:TMP200] = [w:GD63990] - 2000;// 年の下 2 桁

[w:TMP201] = (([w:TMP200] / 10) << 4) + ([w:TMP200] % 10);  // 年 BIN -> BCD
[w:TMP202] = (([w:GD63991] / 10) << 4) + ([w:GD63991] % 10);   // 月 BIN -> BCD
[w:TMP203] = (([w:GD63992] / 10) << 4) + ([w:GD63992] % 10);   // 日 BIN -> BCD
[w:TMP204] = (([w:GD63993] / 10) << 4) + ([w:GD63993] % 10);   // 時 BIN -> BCD
[w:TMP205] = (([w:GD63994] / 10) << 4) + ([w:GD63994] % 10);   // 分 BIN -> BCD
[w:TMP206] = (([w:GD63995] / 10) << 4) + ([w:GD63995] % 10);   // 秒 BIN -> BCD
```

オブジェクト(名称)	数値表示(日)		
スクリプトユーザ ID	3		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 年月設定

[w:GS513] = ([w:TMP201] << 8) + [w:TMP202];  // 変更時刻デバイスに年月セット
```

オブジェクト(名称)	数値表示(時)		
スクリプトユーザ ID	4		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 日時設定

[w:GS514] = ([w:TMP203] << 8) + [w:TMP204];  // 変更時刻デバイスに日時セット
```

オブジェクト(名称)	数値表示(分)		
スクリプトユーザ ID	5		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 分秒設定

[w:GS515] = ([w:TMP205] << 8) + [w:TMP206];  // 変更時刻デバイスに分秒セット
```

オブジェクト(名称)	数値表示(秒)		
スクリプトユーザ ID	6		
データ形式	符号なし BIN16	トリガ種別	常時

```
// 曜日設定

[w:TMP211] = [w:GD63990];  //年(BIN)
[w:TMP212] = [w:GD63991];  //月(BIN)
[w:TMP213] = [w:GD63992];  //日(BIN)

if(([w:TMP212] == 1) || ([w:TMP212] == 2)){//1･2 月の場合のみ前年の 13･14 月として計算するための補正処理
    [w:TMP211] =[w:TMP211] - 1; //年から 1 を減算
    [w:TMP212] =[w:TMP212] + 12;//月に 12 を加算
}

[w:TMP220] = [w:TMP211]/4;//ツェラーの公式に必要な項を作成
[w:TMP221] = [w:TMP211]/100;//ツェラーの公式に必要な項を作成
[w:TMP222] = [w:TMP211]/400;//ツェラーの公式に必要な項を作成
[w:TMP223] = (13*[w:TMP212]+8)/5;//ツェラーの公式に必要な項を作成

//ツェラーの公式で曜日算出して変更時刻デバイスに曜日をセット
[w:GS516] = ([w:TMP211]+[w:TMP220]-[w:TMP221]+[w:TMP222]+[w:TMP223]+[w:TMP213])%7;
```