

Web Datalogger Unit用

モニタ部品 説明書

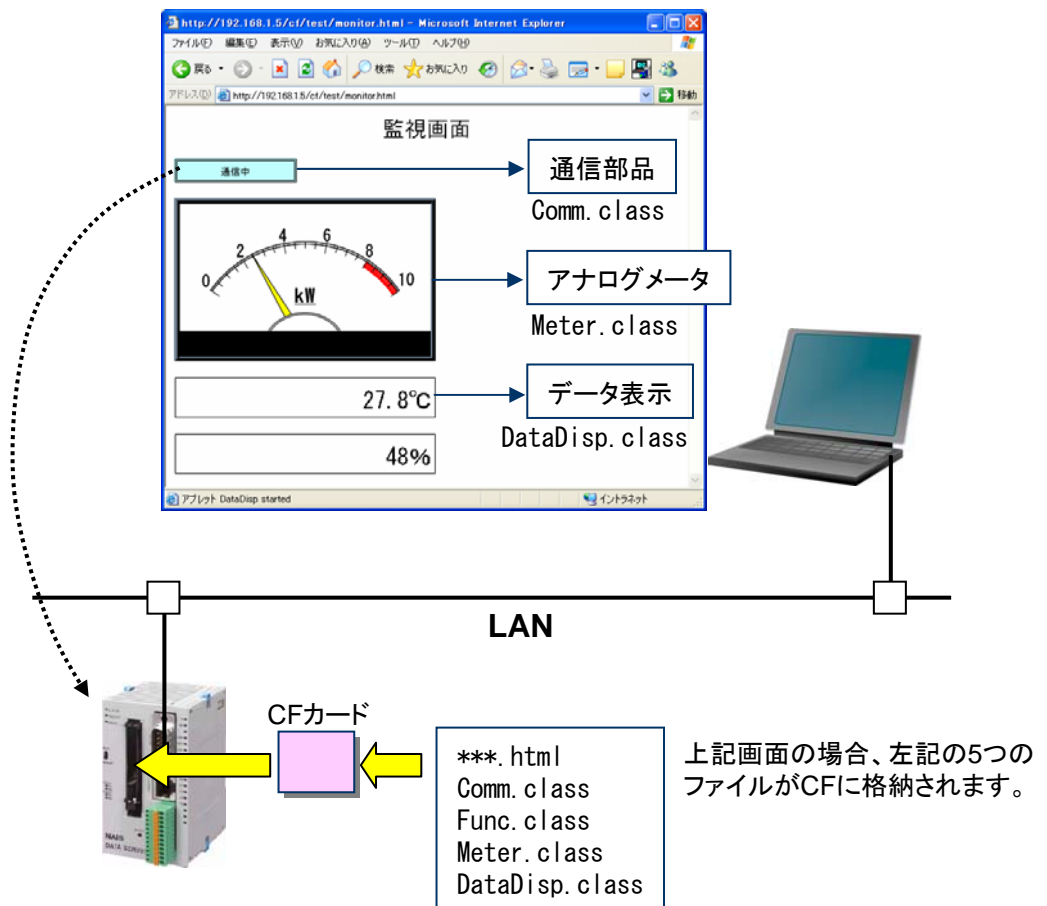
【ご注意】

このモニタ部品は、Web Datalogger Unit(AFL1200)用です。
AFL1000では使用できません。

モニタ部品の概要

モニタ部品 (Javaアプレット) をHTMLで記述して、画面 (Webページ) を作成することにより、DLUのデータをWebブラウザでモニタすることができます。
作成したWebページやモニタ部品は、CFカードにコピーしてDLUに装着します。
DLUとの通信は、通信部品が自動でおこないますので、通信プログラムなどを作成する必要はありません。

●モニタ部品使用例



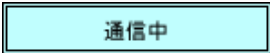



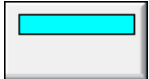
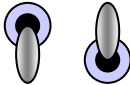





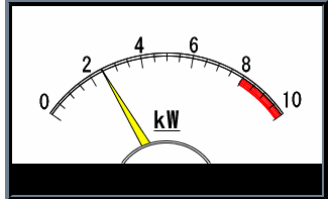
■通信部品について

- ・通信部品は1つのWebページに必ず1つ必要となります。
- ・各部品 (上記のメータやデータ表示) は、通信部品が取得したデータを参照します。

■Funcクラスについて

HTML中に記述することはありませんが、Func.classが必ず必要となります。
他のモニタ部品 (アプレット) と同じ場所にコピーしておきます。

モニタ部品の一覧

部品名	外観(例)	動作内容
通信部品		DLUとの通信を実行します。
ランプ		ビットデバイスのON/OFF情報を表示します。
ピクチャーランプ	 ONの画像  OFFの画像	ビットデバイスのON/OFF情報を予め用意した絵で表示します。
スイッチ		ビットデバイスのON/OFFを操作できます。
ピクチャースイッチ	 スイッチの画像	ビットデバイスのON/OFFを予め用意した絵で操作できます。
ブザー		ビットデバイスのONを検知して、予め用意したサウンドファイル (*.au)を鳴らすことができます。
データ表示		DLUのデータをテキストで表示します。(データの後に任意の文字を付加できます)
データ設定		アナログ出力や外部機器のデータをセットできます。
スライダー		アナログ出力や外部機器のデータをスライダーを使ってセットできます。
メッセージ表示		データ値によって任意のメッセージを表示できます。(256種類登録できます)
プログレスバー		データをプログレスバーで%表示します。
メータ		データをメータ形式で表示します。
トレンドグラフ		現在値データを折れ線グラフ形式で表示します。

DLUのメモリエリア

■DLUのメモリエリア

DLUは、以下のデバイス(メモリエリア)をサポートしています。
これらのデバイスを参照するため、HTMLに以下の記述をおこないます。

(例) **param name="ref_data" value="DT0"**

名称		使用できる点数・範囲	機能
リレー	外部入力(X)	本体入力: X0~X3 増設ユニット1: X20~X3F 増設ユニット2: X40~X5F 増設ユニット3: X60~X7F	外部からの入力でON/OFF
	外部出力(Y)	本体出力: Y0 増設ユニット1: Y20~Y3F 増設ユニット2: Y40~Y5F 増設ユニット3: Y60~Y7F	外部にON/OFFを出力
	内部リレー(R)	256点 (R0~R15F)	DLUの内部でON/OFFさせる。 トリガ接点として使用可能。
レジスタ	データレジスタ(DT)	8000ワード (DT0~DT7999)	DLUが蓄積しているデータの現在値が格納される。
	特殊データレジスタ(DT)	DT90054: 分・秒 DT90055: 日・時 DT90056: 年・月 DT90057: 曜日	DLUに設定された時刻が格納されている。 書き込み可能。

■DLUが蓄積しているデータが格納されるデータレジスタ(DT)

ファイルNo.	登録No.	DT
1	1~250	0~499
2	1~250	500~999
3	1~250	1000~1499
4	1~250	1500~1999
5	1~250	2000~2499
6	1~250	2500~2999
7	1~250	3000~3499
8	1~250	3500~3999
9	1~250	4000~4499
10	1~250	4500~4999
11	1~250	5000~5499
12	1~250	5500~5999
13	1~250	6000~6499
14	1~250	6500~6999
15	1~250	7000~7499
16	1~250	7500~7999

【注意事項】

DLUは、蓄積データを格納するエリアとして、常に2ワード占有します。
蓄積デバイスが接点、16ビットデータ、32ビットデータいずれの場合も2ワード使用します。

よって、250の登録に対してデータレジスタは500ワード割り付けられています。
HTMLで指定するときは、DTの先頭番号を指定します。

例)

・ファイルNo.1のNo.1に登録されたデータ
param name ="ref_data" value="DT0"

・ファイルNo.1のNo.10に登録されたデータ
param name ="ref_data" value="DT20"

・ファイルNo.10のNo.200に登録されたデータ
param name ="ref_data" value="DT4898"

■計算式

$$([\text{ファイルNo.}] - 1) \times 500 + ([\text{登録No.}] - 1) \times 2$$

■データの型について

取得したデータを表示するために、データの型を別途指定する必要があります。
データの型は以下の通りです。

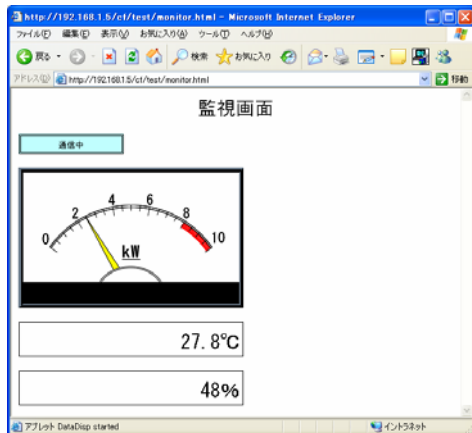
使用するモニタ部品によって、指定できる型とできない型があります。
詳細は、それぞれのモニタ部品の説明を参照してください。

パラメータ(data_type)	データ型
S16	符号付き16ビット整数
US16	符号なし16ビット整数
HEX4	HEX4桁
BIN16	2進数(16ビット)
ASC	文字
S32	符号付き32ビット整数
US32	符号なし32ビット整数
HEX8	HEX8桁
BIN32	2進数(32ビット)
FLT	実数(単精度浮動小数点)

HTMLへの記述について

- ・作成するWebページ(HTMLファイル)にモニタ部品 (Javaアプレット)を記述するには、`<applet>`タグを使用します。
- 1つのWebページには、必ず通信アプレットを1つ記述する必要があります。

(例)



```
<HTML>
<HEAD>
<TITLE>テスト</TITLE>
</HEAD>
```

```
<!--通信用アプレットのみ「name = "Comm"」の記述が必要です。
ダウンロード元のIPアドレスは自動で取得されます。
通信間隔やタイムアウトの設定が可能です。-->
<applet name="Comm" code="Comm.class" width=150 height=30 >
  <param name="renew_timing" value="1000">
</applet>
```

```
<!--実際に部品を使う際の記述-->
<applet code="Meter.class" width=320 height=200>
  <param name="ref_data" value="DT0">
  <param name="data_type" value="FLT">
  <param name="min" value="0">
  <param name="max" value="10">
  <param name="pmax" value="8">
  <param name="unit_name" value="kW">
  <param name="count_num" value="5">
</applet>
<applet code="DataDisp.class" width=320 height=50>
  <param name="ref_data" value="DT2">
  <param name="data_type" value="FLT">
  <param name="unit_name" value="°C">
</applet>
<applet code="DataDisp.class" width=320 height=50>
  <param name="ref_data" value="DT4">
  <param name="data_type" value="S16">
  <param name="unit_name" value="%">
</applet>
</HTML>
```


モニタ部品 パラメータ仕様

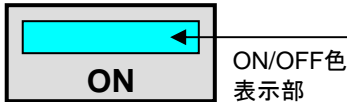
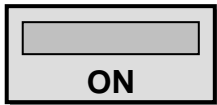
部品名: スイッチ

アプレット名: Switch.class

概要

DLUのビットデバイスをON/OFFする。

イメージ



1. セット: 接点をONするコマンドを送信。
2. リセット: 接点をOFFするコマンドを送信。
3. オルタネート: 現在の状態を読み込んで反転させるコマンドを送信。

パラメータ

項目	パラメータ名	設定内容	初期値
参照データ	ref_data	P.4を参照	—
参照ビット	ref_bit	0~15	0
種類	type	set: セット reset: リセット alt: オルタネート push1: マウスクリックでset, 離すとreset push2: マウスクリックでreset, 離すとset	alt
ON時の色	color_on	RGB値で指定	0x00FFFF
OFF時の色	color_off	RGB値で指定	ボタンと同色
ボタンのテキスト	text		—
参照モード	ref_mode	off: 参照しない, on: 参照する	on

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・接点デバイス(X, Y, R, T, C, L)以外が指定されると、0:OFF, 0以外:ONと判断する。
- ・ref_modeがonの場合、設定直後とrenew_timing毎に確認する。
- ・ref_modeがoffの場合はcolor_on, color_offは無効。(この部分はボタンと同色になる)

モニタ部品 パラメータ仕様

部品名: ピクチャースイッチ

アプレット名: PictSwitch.class

概要

DLUのビットデバイスをON/OFFする。
スイッチの絵(定常状態時とマウスクリック時)は別途用意する。

イメージ

【例】



(定常時)



(クリック時)

1. セット: 接点をONするコマンドを送信
(lockがonの時は、解除時にOFFするコマンドを送信)
2. リセット: 接点をOFFするコマンドを送信
(lockがonの時は、解除時にONするコマンドを送信)

上記の場合、定常時は左側の絵が表示され、マウスでクリックすると右側の絵になる。
マウスを離すと、絵は左側に戻る。
lockがonの場合、クリックするたびに絵が変わる。

パラメータ

項目	パラメータ名	設定内容	初期値
設定データ	ref_data	P.4を参照	—
参照ビット	ref_bit	0~15	0
種類	type	set: セット reset: リセット alt: オルタネート push1: マウスクリックでset, 離すとreset push2: マウスクリックでreset, 離すとset	alt
クリック時の画像	pict_click	ファイル名	—
定常時の画像	pict_normal	ファイル名	—
ON時の画像	pict_on	pict_click, pict_normal	pict_click
OFF時の画像	pict_off	pict_click, pict_normal	pict_normal
参照モード	ref_mode	off: 参照しない, on: 参照する	off

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・接点デバイス(X, Y, R, T, C, L)以外が指定されると、0: OFF, 0以外: ONと判断する。

モニタ部品 パラメータ仕様

部品名: ブザー

アプレット名: Buzzer.class

概要

DLUビットデバイスのONにより、あらかじめ用意されたサウンドファイルを再生する。

イメージ

パラメータ

項目	パラメータ名	設定内容	初期値
参照データ	ref_data	P.4を参照	—
参照ビット	ref_bit	0~15	0
サウンドファイル	sound	*.au	—
再生モード	play_mode	1: 対象接点ONで鳴動し、一定時間経過後に自動停止 2: 対象接点ON中に常時鳴動	1
1回の再生時間	sound_time	サウンドファイルを1回再生する時間	1000(ms)
トータル再生時間	play_time	再生モード「1」における自動停止までの時間	5000(ms)

備考

- ・使用できる音のファイルは、拡張子が「au」のファイルのみ。
- ・sound_timeとplay_timeは100ms単位で指定する。
- ・接点デバイス(X, Y, R, T, C, L)以外が指定されると、0:OFF, 0以外:ONと判断する。

モニタ部品 パラメータ仕様

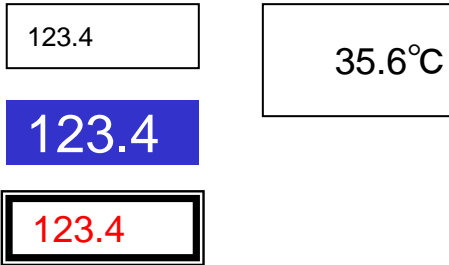
部品名: データ表示

アプレット名: DataDisp.class

概要

DLUのデータをテキストで表示する。

イメージ



パラメータ

項目	パラメータ名	設定内容	初期値
参照データ	ref_data	P.4を参照	—
データの型	data_type	P.5を参照(ASC以外の全てのデータ型に対応)	S16
換算値	conv_value	-9999.99~9999.99	1
背景色	color_back	背景色	0x000000
小数点以下の桁数	point	0~9(実数時に有効)	0
テキストの横位置	text_hpos	左:left, 中央:center, 右:right	right
テキストの縦位置	text_vpos	上:top, 中央:center, 下:bottom	center
単位	unit_name	「°C」や「kw」など自由に設定可能	—
外枠	frame_type	0:枠線無し, 1:一重線, 2:二重線	1

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・各種フォントパラメータも変更可能。(font_color, font_style, font_size)

モニタ部品 パラメータ仕様

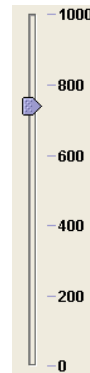
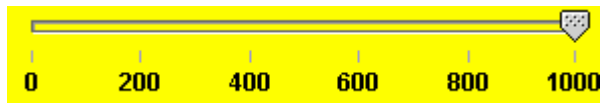
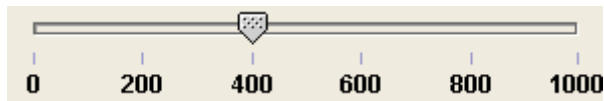
部品名: スライダー

アプレット名: Slider.class

概要

スライダーを使ってDLUのレジスタへ任意の値をセットする。
(大まかな設定に使用)

イメージ



パラメータ

項目	パラメータ名	設定内容	初期値
設定データ	ref_data	P.4を参照	—
データの型	data_type	P.5を参照(整数に対応) S16,US16,S32,US32	S16
表示方向	direction	horizontal: 横方向, vertical: 縦方向	horizontal
最小値, 最大値	min,max	整数で設定する。	—
メモリの刻み数	count_num	上図の例だと count_num = 5	1
最小/最大の表示選択	range_disp	off: 表示しない, on: 表示する	on
参照モード	ref_mode	off: 参照しない, on: 参照する	off
背景色	color_back	RGB値で指定	システムカラー
範囲越え時の背景色	color_over	RGB値で指定	システムカラー

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・min,maxの大小関係が不正、または指定されたデータ型の範囲外な場合、「range:error」と表示される。
- ・ref_modeがonの時でもスライダーのノブを操作しているときは参照を行わない。

モニタ部品 パラメータ仕様

部品名: メッセージ表示

アプレット名: Message.class

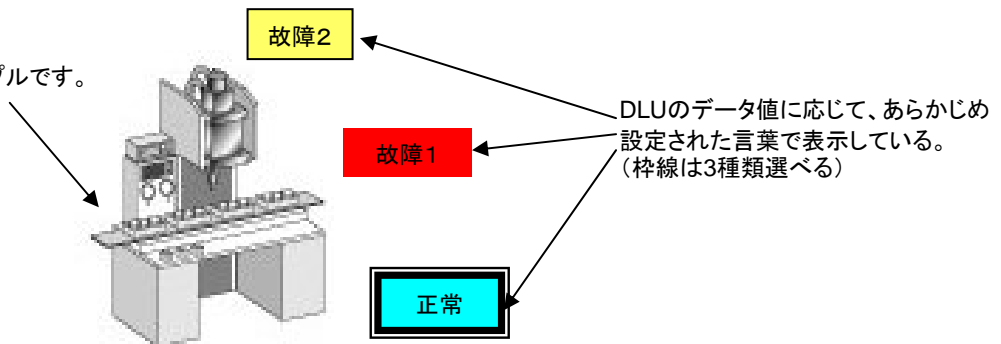
概要

DLUから取得したデータを、あらかじめ設定された言葉で表示する。

(例). DT0 == 1の時:故障1, 0の時(または1以外の時):正常運転中

イメージ

【例】
この絵はサンプルです。



パラメータ

項目	パラメータ名	設定内容	初期値
参照データ	ref_data	P.4を参照	—
データの型	data_type	P.5を参照(整数と実数に対応) S16,US16,S32,US32,FLT	S16
デフォルトメッセージ	msg_default		—
デフォルト背景色	color_default	RGB値で指定	0xFFFFFFFF
表示パターン1~256	data1~256	例: <param name="data1" value="1">	—
表示メッセージ	msg1~256	例: <param name="msg1" value="表示1">	—
テキスト色	color_msg1~ 256	例: <param name="color_msg1" value="0x000000">	0x000000
表示背景色	color_back1~256	例: <param name="color1" value="0xFFFFF0">	0xFFFFFFFF
テキストの横位置	text_hpos	左:left, 中央:center, 右:right	left
テキストの縦位置	text_vpos	上:top, 中央:center, 下:bottom	center
外枠	frame_type	0:枠線無し, 1:一重線, 2:二重線	1

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・data(またはtext)に257以上を記述しても無視される。(エラーにはならない)
- ・dataのvalueに指定されたデータ型に合わない数値が指定された場合、無視される。(設定がないという扱い)

モニタ部品 パラメータ仕様

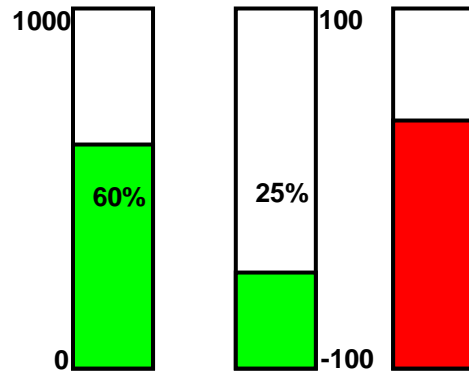
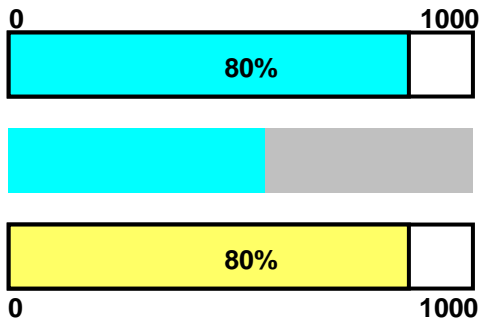
部品名: プログレスバー

アプレット名: Progress.class

概要

DLUから取得したデータが、最小値と最大値が決められたエリアに対してどの位置にあるかを視覚的に(%)確認する。

イメージ



パラメータ

項目	パラメータ名	設定内容	初期値
参照データ	ref_data	P.4を参照	—
データの型	data_type	P.5を参照(整数と実数に対応) S16,US16,S32,US32,FLT	S16
換算値	conv_value	-9999.99~9999.99	1
表示方向	direction	horizontal: 横方向, vertical: 縦方向	horizontal
最小値, 最大値	min,max	小数点以下2桁まで指定可能	—
警告範囲	pmin,pmax	小数点以下2桁まで指定可能 (pmin以下, pmax以上で有効)	なし(指定がなければ警告しない)
最小/最大の表示選択	range_disp	off: 表示しない, on: 表示する	on
最小/最大の表示位置	range_pos	upper(left): 上(または左), lower(right): 下(または右)	horizontal: upper vertical: left
%の表示形式	percent_disp	off: 表示しない, on: 表示する	on
フレーム(枠)の表示	frame_disp	off: 表示しない, on: 表示する	on
色(現在値)	color_current	RGB値で設定	0x00FFFF
色(背景)	color_back	RGB値で設定	0xFFFFFFFF
色(範囲超え時)	color_over	RGB値で設定	0xFF0000

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・min,maxの大小関係が不正、または指定されたデータ型の範囲外な場合、「range:error」と表示される。

モニタ部品 パラメータ仕様

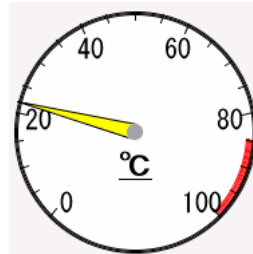
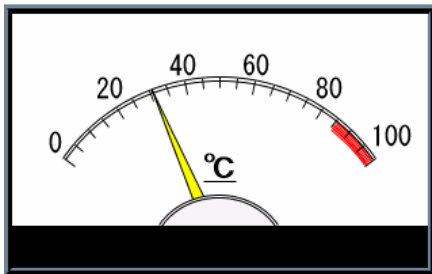
部品名: メーター

アプレット名: Meter.class

概要

DLUから取得したデータが、最小値と最大値が決められたエリアに対してどの位置にあるかをメーター形式で確認する。

イメージ



丸形(circle)はアプレットのサイズ(width, height)に従う。
例) width=100, height=100 ⇒ 真円(縦横比が1:1)

縦横比は1:1.6で最適化されています。

パラメータ

項目	パラメータ名	設定内容	初期値
参照データ	ref_data	P.4を参照	—
データの型	data_type	P.5を参照(整数と実数に対応) S16,US16,S32,US32,FLT	S16
換算値	conv_value	-9999.99~9999.99	1
表示形式	style	rect: 四角形, circle: 丸形	rect
最小値, 最大値	min,max	小数点以下2桁まで指定可能	—
警告範囲	pmin,pmax	小数点以下2桁まで指定可能 (pmin以下, pmax以上で有効)	なし(指定がなければ警告しない)
表示単位	unit_name	mA等(文字列をそのまま表示。 半角で4文字程度の長さが丁度よい)	—
メモリの刻み数	count_num	上図の例だと count_num = 5	5
色(背景)	color_back	RGB値で指定	0xFFFFFFFF
色(範囲超え時)	color_over	RGB値で指定	0xFFFFFFFF
ゼロの位置(丸時有効)	zero_pos	0: 垂直, 1: ななめ	0

備考

- ・ref_dataが不正な場合はアプレットロード時に「ref_data:error」と表示される。
- ・min,maxの大小関係が不正、または指定されたデータ型の範囲外な場合、「range:error」と表示される。

モニタ部品 パラメータ仕様

部品名: トレンドグラフ

アプレット名: TrendGraph.class

概要

DLUから取得したデータ(現在値)をグラフ上に表示する。
最大8つまでのデータが登録可能。

イメージ

■詳細は次ページ

下記パラメータ内「参照データ」の「設定内容」に関しては、「ユニットのメモリ構成(2)」を参照。

パラメータ

項目	パラメータ名	設定内容	初期値
データ名	data1_name	表示する名前(全角8文字以内)	—
参照データ	ref_data1	P.4を参照	—
データの型	data1_type	P.5を参照(整数と実数に対応) S16,US16,S32,US32,FLT	S16
換算値	conv_value	-9999.99~9999.99	1
線種	data1_line	1~8	1
線の色	data1_color	RGB値で指定	0x000000
最小値, 最大値	min,max	小数点以下2桁まで指定可能	0.0, 1000.0
縦軸の表示単位	unit_name	kw/h等(文字列をそのまま表示)	—
横軸の分割数	x_division		10
縦軸の分割数	y_division		10
自動停止回数	stop_count	stop_count回サンプリングしたら自動停止	0(停止しない)
グラフの更新間隔	renew_timing	グラフが更新される(描画される)間隔	1000

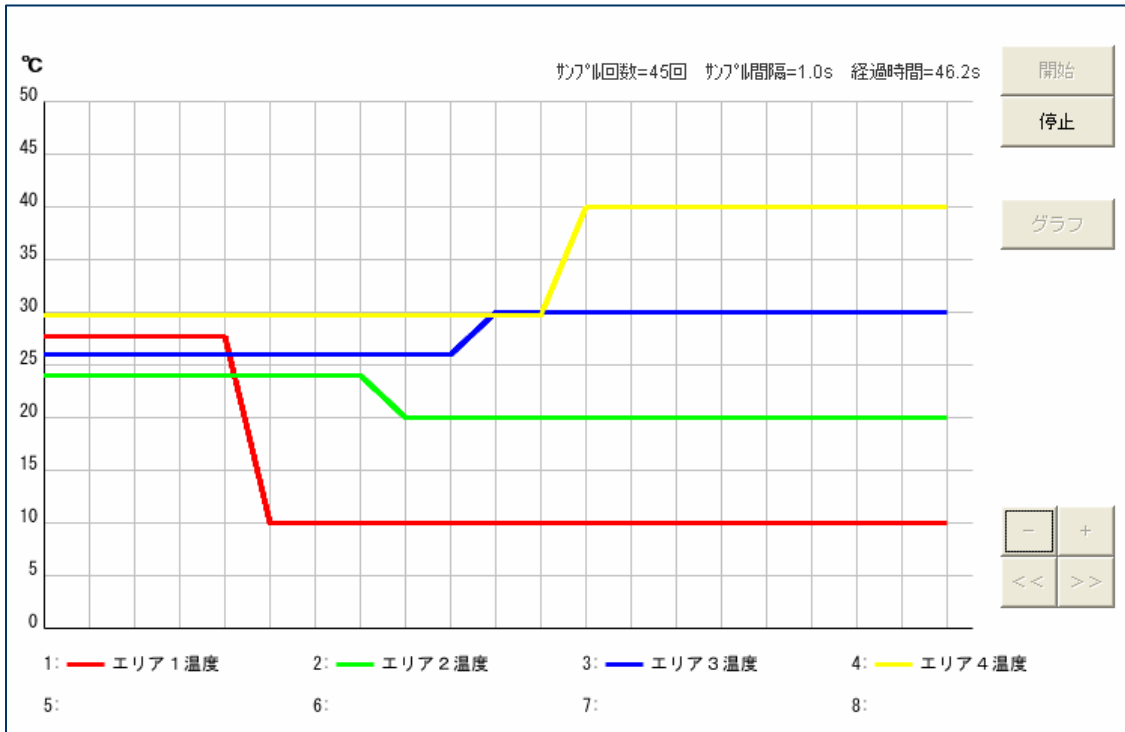
備考

- ・data1_* は、data1~data8まで指定可能。
- ・デバイスタイプとデバイスNo.があれば、それ以外は指定が無くても表示される。
- ・なお、全くどのパラメータに指定が無くてもグラフは表示される。(グラフ上で登録可能)

■トレンドグラフのアプレット(TrendGraph.class)詳細

トレンドグラフは一定間隔で現在値を読み込み表示する

アプレットの外観



●各ボタンの働き

- ・開始: モニタを開始する。
- ・停止: モニタを停止する。
- ・グラフ: 表示範囲やグリッドの数等を設定する。「分割数」で割り切れない場合、縦軸には小数点以下2桁まで表示する)
- ・縮小／拡大: サンプルングを停止した後、画面の拡大／縮小を設定する。
- ・「<</>>」: サンプルング停止後、データをさかのぼって確認する。

●その他特長

- ・「開始」ボタンが押されたら、「停止」以外は無効なボタンとなる。
- ・1データに対して120個のデータまで復元可能。(表示を戻すことが出来る)

注)

通信クラスの「renew_timing」は、あくまでも通信クラスがDLUからデータを取得する間隔で、トレンドグラフの更新間隔は別途設定する必要があります。

例えば、通信クラスのrenew_timingが100msで、トレンドグラフのrenew_timingが1秒の場合、データは100ms間隔で取得してありますが、グラフの表示が更新されるのは1秒間隔となります。よって、トレンドグラフの更新間隔が通信クラスの更新間隔より短い場合、トレンドグラフのデータは何回か同じ値を表示することになります。

モニタ部品 パラメータ仕様

部品名:

アプレット名:

概要

各アプレットに共通なフォントの設定をおこないます

イメージ

■Javaでサポートされる標準的なフォント(論理フォント)
→実際はOSの物理フォントにマッピングされる。

Serif PLAIN

SansSerif PLAIN

Monospaced PLAIN → 等幅フォント

Dialog PLAIN

DialogInput PLAIN

・BOLDやITALICもサポートされます

SansSerif ITALIC

SansSerif BOLD

SansSerif ITALIC+BOLD

パラメータ

項目	パラメータ名	設定内容	初期値
フォントタイプ	font_face	Serif, SansSerif, Monospaced Dialog, DialogInput	Monospaced
フォントサイズ	font_size	数字で指定	12
フォントスタイル	font_style	BOLD(太字), ITALIC(斜文字) 両方指定する場合は「;(セミコロン)」 で区切る。(順不同) 例: <param name="font_style" value="BOLD;ITALIC">	PLAIN
フォントカラー	font_color	RGB値で設定	0x000000
			-

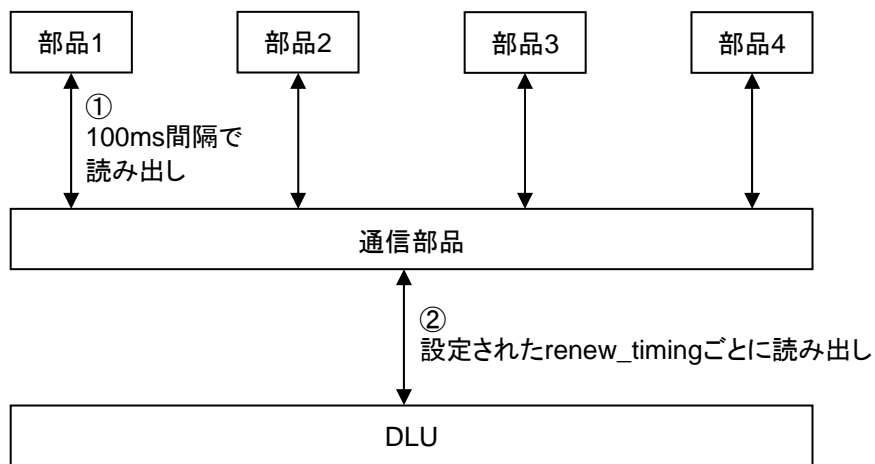
備考

- ・上記フォントの指定には大文字, 小文字の区別はありません。
- ・プロポーションアルフォントは表示がずれたりする可能性が高いため、推奨しません。(表示は可能)
- ・Windows上では「MS 明朝」や「MS ゴシック」も表示できます。

注意事項

モニタ部品とDLUとの通信は、以下のように実行されます。
1画面に存在するモニタ部品の点数が多いほど、リアルタイム性は悪くなります。
これは、通信部品の「renew_timing」を短くすることにより多少は改善されますが、よりリアルタイム性を追求される場合は、1画面に記述するモニタ部品をできるだけ少なくしてください。
多くのモニタ部品を使用し通信間隔を短くすると、モニタに使用しているパソコンのCPUの使用率が高くなります。

(例)1つのWebページに4つのモニタ部品が存在するケース



①各部品は最初に、読み出すデバイス種別(ref_dataで記述)を通信部品に登録し、以降は100ms間隔で通信部品からデータを読み出す。

②通信部品は、各部品から登録されたデバイスをDLUから1つずつ読み出す。
(読み出す間隔は「renew_timing」で記述)

上記により、例えばrenew_timingが1000ms(1秒)の場合、モニタ部品1のデータが更新されるのに、DLU内でのデータ変化に対して最大4秒 + α の遅れが生じる事になります。

この場合、renew_timingを100msにすれば、遅れは最大400ms + α となります。