



グラフィカル・ユーザー・インターフェース
構築ツール

グラフ表示支援ライブラリ

リファレンス・マニュアル

Version 1.5

はじめに

本グラフ表示支援ライブラリは、X-Mate を使用し GUI を開発される
ユーザがより便利、簡単にグラフの表示を行う為の便利ツールとして開発されました。

各グラフ部品の構造体にデータをセットし、グラフ表示支援ライブラリの関数を起動するだけで
円グラフ、折れ線グラフ、棒グラフ等、様々なグラフを簡単に表示させる事が出来ます。

また、グラフ表示支援ライブラリは X-Mate開発ライブラリを使用して実現しています。
X-Mate開発ライブラリを X-Mate と組み合わせる事で、より高度なグラフをユーザが
実現する時の手助けになるものと思っています。

本書をお読みになり、グラフ表示支援ライブラリをご利用頂ければ幸いと存じます。

目次

1 概要	(5)
2 サブルーチン	(6)
TKgropen	グラフオープン (7)
TKgrclose	グラフクローズ (9)
TKgrentry	グラフ部品の登録 (10)
TKgrdelete	グラフ部品の削除 (12)
TKgrdisplay	グラフ部品の表示 (14)
TKgrerase	グラフ部品の消去 (16)
TKgrzoom	拡大/縮小設定関数 (18)
3 部品情報	(19)
3.1 グラフ部品共通情報	(19)
折れ線グラフ	(21)
折れ線グラフ2	(23)
折れ線グラフ3	(27)
縦折れ線グラフ2	(30)
縦折れ線グラフ3	(34)
エリアグラフ	(37)
積み上げエリアグラフ	(39)
レーダーチャート	(41)
円グラフ	(44)
円グラフ2	(47)
棒グラフ	(50)
棒グラフ2	(52)
棒グラフ3	(55)
棒グラフ3A	(57)
棒グラフ4	(59)
積み上げ棒グラフ	(62)
積み上げ棒グラフ2	(65)
積み上げ棒グラフ3	(68)
積み上げ棒グラフ4	(71)
円柱グラフ	(74)
円柱グラフ2	(76)
積み上げ円柱グラフ	(78)
積み上げ円柱グラフ2	(81)
タイミングチャート	(84)
タイミングチャート2	(86)
インジケーター	(89)
4 サンプルプログラム	(92)
サンプルプログラム1 (折れ線グラフ)	(92)
サンプルプログラム2 (PRkid を使って印刷)	(98)
5 ライブラリ	(100)

1 概要

グラフ表示支援ライブラリは X-Mate を使用し GUI の開発をされるユーザがグラフの表示を行なう為の開発支援ライブラリです。

以下のグラフ表示が行えます。

- ・ 縦折れ線グラフ
- ・ 横折れ線グラフ
- ・ エリアグラフ
- ・ 積み上げエリアグラフ
- ・ レーダーチャート
- ・ 円グラフ
- ・ 縦棒グラフ
- ・ 横棒グラフ
- ・ 縦積み上げ棒グラフ
- ・ 横積み上げ棒グラフ
- ・ 縦円柱グラフ
- ・ 横円柱グラフ
- ・ 縦積み上げ円柱グラフ
- ・ 横積み上げ円柱グラフ
- ・ タイミングチャート
- ・ インジケーター

グラフの表示は、TKopen(パネルオープン)でオープンされたパネルに対し、TKgropen(グラフオープン)時にグラフの背景となる部品データ(X-Mate で作成)を表示し、個々のグラフ部品を重ねて表示する事により実現します。

2 サブルーチン

グラフ表示支援ライブラリには以下の関数が用意されています

関 数 名	機 能
TKgropen	各グラフの初期化をする。
TKgrclose	各グラフを終了する。
TKgrentry	指定されたパネルにグラフを登録し、表示する。
TKgrdelete	指定されたパネルに登録されているグラフを削除し パネルから消去する。
TKgrdisplay	指定されたパネルにグラフの表示を行なう。
TKgrerase	指定されたパネルに表示されているグラフの消去を 行なう。
TKgrzoom	指定されたグラフ部品と背景部品にパネルサイズの変 更による拡大/縮小対象部品のフラグをセット/ リセットします。

使用例について

X-Mate がインストールされたディレクトリの GRAPH にサンプルプログラムが準備されています。
参照して下さい。

機能

各グラフのオープン(初期設定)を行います。
指定されたパネルに対して背景部品を登録し表示させます。
TKgrclose と対で使用します。グラフを描画する為には、まず最初にこの関数の指定が必要です。

コーリングシーケンス

```
Grcore      *TKgropen      (core, mode, obj);

Ktcore      *core;         /* パネル管理情報          */
int          mode;         /* 登録モード              */
Ktobj       *obj;          /* グラフ背景部品データ    */
```

戻り値

・ グラフ管理情報

ライブラリが管理するグラフの情報です。
グラフに対して操作する際に使用します。

パラメータ説明

・ パネル管理情報

TKopen で返された情報を渡します。

・ 登録モード

KSINGL 単体登録モード

1部品のみの登録を行ないます。

KARRAY 複数登録モード

部品のアドレステーブルのアドレスをパラメータとして指定し、複数の部品の登録を行ないます。
テーブルエンドは NULL を設定します。

NULL 未登録モード

・ グラフ背景部品データ

グラフの背景となる部品を指定します。
グラフ表示支援ライブラリがグラフの背景となる部品を X-Mate開発ライブラリに
対して登録を行ない、表示させます。

単体登録モード時は、登録部品データのアドレスを指定します。
複数登録モード時は、部品のアドレステーブルのアドレスを指定し、
NULL までが有効データとなります。

機 能

TKgropen で作成された資源を開放します。
オープン時に指定された背景部品、描画されているグラフが残っていれば
(TKgrdelete により削除されていない時)削除しパネルから消去します。

コーリングシーケンス

```
void      TKgrclose      (grcore);  
  
Grcore    *grcore;      /*      グラフ管理情報      */
```

パラメータ説明

・ グラフ管理情報

TKgropen で返された情報を渡します。

機能

指定されたグラフ部品を登録し、表示を行ないます。
グラフ部品はライブラリ内で作成され、TKentry によって表示されます。

コーディングシーケンス

```
void      TKgrentry      (grcore, mode, grobj, bdsp);

Grcore    *grcore;      /* グラフ管理情報          */
int        mode;         /* 登録モード              */
Grobj      *grobj;       /* グラフ部品データ        */
int        bdsp;         /* 背景部品再表示指定      */
```

パラメータ説明

・ グラフ管理情報

TKgropen で返された情報を渡します。

・ 登録モード

KSINGL 単体登録モード

1部品のみの登録を行ないます。

KARRAY 複数登録モード

部品のアドレステーブルのアドレスをパラメータとして指定し、複数の
部品の登録を行ないます。
テーブルエンドは NULL を設定します。

・ グラフ部品データ

単体登録モード時は、登録部品データのアドレスを指定します。
複数登録モード時は、部品のアドレステーブルのアドレスを指定し、
NULL までが有効データとなります。

背景部品再表示指定

グラフ部品を表示した後のTKgropen 時に指定した背景部品の再表示の指定をします。

HRBDSPON	背景部品の再表示を行ないます。
HRBDSPOFF	背景部品の再表示を行ないません。

グラフ部品の削除

TKgrdelete

機能

指定された TKgrentry によって登録されたグラフ部品を削除し、パネルから消去します。

コーリングシーケンス

```
void      TKgrdelete      (grcore, mode, grobj, bdsp);

Grcore    *grcore;        /* グラフ管理情報          */
int        mode;           /* 削除モード              */
Grobj      *grobj;        /* グラフ部品データ        */
int        bdsp;           /* 背景部品再表示指定      */
```

パラメータ説明

・ グラフ管理情報

TKgropen で返された情報を渡します。

・ 削除モード

KSINGL 単体削除モード

1部品のみの削除を行ないます。

KARRAY 複数削除モード

部品のアドレステーブルのアドレスをパラメータとして指定し、複数の部品の削除を行ないます。
テーブルエンドは NULL を設定します。

KALL

各グラフに登録されている全ての部品の削除を行ないます。

・ グラフ部品データ

単体削除モード時は、削除部品データのアドレスを指定します。
複数削除モード時は、部品のアドレステーブルのアドレスを指定し、
NULL までが有効データとなります。

・ 背景部品再表示指定

グラフ部品を消去した後のTKgropen 時に指定した背景部品の再表示の
指定をします。

HRBDSPON	背景部品の再表示を行ないます。
HRBDSPOFF	背景部品の再表示を行ないません。

機能

指定された TKgrentry によって登録されたグラフ部品の表示を行ないます。
グラフ管理情報に登録されている (TKgrentry で登録した) 部品が対象になります。 TKgrerase (グラフ部品の消去) でグラフ部品を消去した後、TKgrdisplay (グラフ部品の表示) をする事によりグラフ部品の再表示となります。

コーリングシーケンス

```
void      TKgrdisplay      (grcore, mode, grobj, bdsp);

Grcore    *grcore;        /* グラフ管理情報          */
int        mode;           /* 表示モード              */
Grobj      *grobj;         /* グラフ部品データ        */
int        bdsp;           /* 背景部品再表示指定      */
```

パラメータ説明

・ グラフ管理情報

TKgropen で返された情報を渡します。

・ 表示モード

KSINGL 単体表示モード

1部品のみの表示を行ないます。

KARRAY 複数表示モード

部品のアドレステーブルのアドレスをパラメータとして指定し、複数の部品の表示を行ないます。
テーブルエンドは NULL を設定します。

KALL

各グラフに登録されている全ての部品の表示を行ないます。

・ グラフ部品データ

単体表示モード時は、表示部品データのアドレスを指定します。

複数表示モード時は、部品のアドレステーブルのアドレスを指定し、NULL までが有効データとなります。

・ 背景部品再表示指定

グラフ部品を表示した後のTKgropen 時に指定した背景部品の再表示の指定をします。

HRBDSPON	背景部品の再表示を行ないます。
HRBDSPOFF	背景部品の再表示を行ないません。

グラフ部品の消去

TKgrerase

機能

指定された TKgrentry によって登録されたグラフ部品を画面表示から消去します。(削除は行なわれません。TKgrentry で作成されたグラフ部品は残ったままです。)

グラフ管理情報に登録されている (TKgrentry で登録した) 部品が対象になります。

コーリングシーケンス

```
void      TKgrerase      (grcore, mode, grobj, bdsp);

Grcore    *grcore;      /* グラフ管理情報          */
int        mode;         /* 消去モード              */
Grobj      *grobj;       /* グラフ部品データ        */
int        bdsp;         /* 背景部品再表示指定      */
```

パラメータ説明

・ グラフ管理情報

TKgropen で返された情報を渡します。

・ 消去モード

KSINGL 単体消去モード

1部品のみの消去を行ないます。

KARRAY 複数消去モード

部品のアドレステーブルのアドレスをパラメータとして指定し、複数の部品の消去を行ないます。
テーブルエンドは NULL を設定します。

KALL

各グラフに登録されている全ての部品の消去を行ないます。

・ グラフ部品データ

単体消去モード時は、消去部品データのアドレスを指定します。

複数消去モード時は、部品のアドレステーブルのアドレスを指定し、NULL までが有効データとなります。

・ 背景部品再表示指定

グラフ部品を消去した後のTKgropen 時に指定した背景部品の再表示の指定をします。

HRBDSPON	背景部品の再表示を行ないます。
HRBDSPOFF	背景部品の再表示を行ないません。

機能

登録されているグラフと背景に対して、パネルサイズの変更による拡大/縮小の対象になるかどうかの設定を行ないます。
TKgrentry 時には拡大/縮小対象外になっています。

コーディングシーケンス

```
void      TKgrzoom      (grcore, flg, mode);

Grcore    *grcore;      /* グラフ管理情報          */
int        flg;          /* 制御フラグ              */
int        mode;        /* 拡大/縮小対応モード     */
```

パラメータ説明

・ グラフ管理情報

TKgropen で返された情報を渡します。

・ 制御フラグ

HRON フラグオン

パネルサイズに合わせて拡大/縮小の対象となります。

HROFF フラグオフ

パネルサイズに合わせて拡大/縮小の対象とはなりません。

・ 拡大/縮小対応モード

HRGRAPH

グラフ部品のみ制御フラグに従って表示を行ないます。

HRBACK

背景部品のみ制御フラグに従って表示を行ないます。

グラフ部品も背景部品も制御フラグに従って表示を行ないたい場合は
HRGRAPH と HRBACK で OR (HRGRAPH |HRBACK) をとって設定します。

3 部品情報

3.1 グラフ共通情報 (グラフヘッダ情報)

各グラフの定義データは先頭に共通のヘッダ情報を持ち、その後ろに固有のグラフ情報を持ちます。

グラフを定義する時は、必ずヘッダ情報を設定して下さい。

グラフヘッダ情報の説明を以下に示します。

```
typedef struct
{
    int      type;      /* グラフタイプ                */
    QID      id;        /* 表示チェーン ID            */
    int      flag;      /* 表示フラグ                  */
    Grdspet  *ent;      /* 表示部品アドレス          */
} Grobj;
```

・ グラフタイプ

グラフタイプを設定する事で、グラフ表示支援ライブラリがグラフ情報を判別します。

グラフのタイプを以下に示します。

モード	値
折れ線グラフ	HRline
折れ線グラフ2	HRline2
折れ線グラフ3	HRline3
縦折れ線グラフ2	HRtline2
縦折れ線グラフ3	HRtline3
エリアグラフ	HRarea1
積み上げエリアグラフ	HRarea2
レーダーチャート	HRlader
円グラフ	HRarc
円グラフ2	HRarc2
棒グラフ	HRbar
棒グラフ2	HRbar2
棒グラフ3	HRbar3
棒グラフ4	HRbar4
積み上げ棒グラフ	HRtumi
積み上げ棒グラフ2	HRtumi2
積み上げ棒グラフ3	HRtumi3
積み上げ棒グラフ4	HRtumi4
円柱グラフ	HRencyu1
円柱グラフ2	HRencyu3
積み上げ円柱グラフ	HRencyu2
積み上げ円柱グラフ2	HRencyu4
タイミングチャート	HRchart
タイミングチャート2	HRchart2
インジケータ	HRindicator

・ 表示チェーン ID

パネルに対して登録された時のチェーンIDが設定されます。
グラフ表示支援ライブラリが管理する為、初期値は NULL を指定します。

・ 表示フラグ

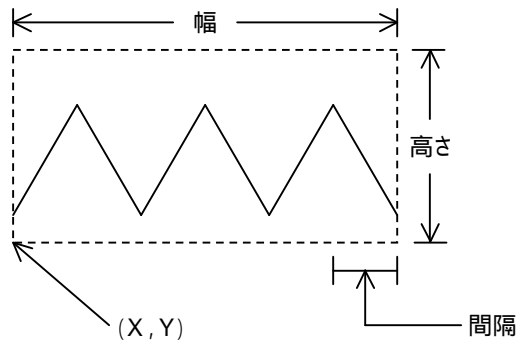
グラフ表示支援ライブラリで管理します。初期値は 1 を指定します。

・ 表示部品アドレス

グラフ表示支援ライブラリで管理します。初期値は NULL を指定します。

機 能

データにより折れ線グラフを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 原点X座標 */
    int      y;          /* 原点Y座標 */
    int      w;          /* 表示幅 */
    int      h;          /* 表示高さ */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      width;      /* 間隔 */
    int      ptn;        /* 太さ/線種 */
    char     *col;       /* 線色 */
    int      pc;         /* データ数 */
    Grptdt   *pt;        /* 表示データアドレス */
} Grline;
```

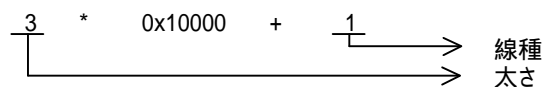
```
typedef struct          /* 表示データ構造体 */
{
    int      data;      /* データ */
} Grptdt;
```

メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	折れ線グラフの原点を示します。
w h	折れ線グラフ表示エリアの幅、高さを示します。
max min	折れ線グラフのスケールの最大値、最小値を示します。
width	各データの間隔を示します。
ptn	上位16ビット … 線の太さを示します。 下位16ビット … 線種を示します。 線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	—— ———
点線	2	- - - - -
一点破線	3	—— - - -
二点破線	4	—— — - -

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



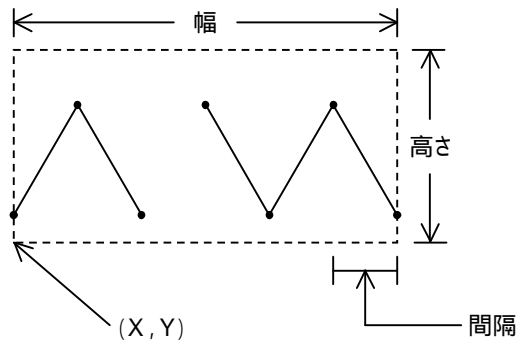
*col	折れ線グラフの線の色を示します。
pc	折れ線グラフのデータの数を示します。
*pt	折れ線グラフのデータアドレスを示します。 (実際のデータは表示データ構造体(Grptdt)の型を持つ配列データです。)

表示データ構造体(Grptdt)

data	折れ線グラフのデータを示します。
------	------------------

機 能

データにより折れ線グラフを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;    /* グラフヘッダ */
    int      x;      /* 原点X座標 */
    int      y;      /* 原点Y座標 */
    int      w;      /* 表示幅 */
    int      h;      /* 表示高さ */
    int      max;    /* スケール最大値 */
    int      min;    /* スケール最小値 */
    int      sym;    /* 座標シンボル/大きさ */
    char     *symcol; /* シンボル色 */
    int      wmode;  /* 間隔モード */
    int      width;  /* 固定間隔値 */
    int      ptn;    /* 太さ/線種 */
    char     *col;   /* 線色 */
    int      pc;     /* データ数 */
    Grptgr1  *pt;    /* 表示データアドレス */
} Grline2;
```

```
typedef struct          /*  間隔固定   表示データ構造体          */
{
    int      data;      /*  データ              */
    int      flg;       /*  フラグ              */
} Grptgr1;
```

```
typedef struct          /* 間隔可変 表示データ構造体 */
{
    int      data;      /* データ */
    int      flg;       /* フラグ */
    int      dw;        /* 可変間隔値 */
} Grptgr2;
```

メンバー説明

hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

x 折れ線グラフの原点を示します。
y

w
h

折れ線グラフ表示エリアの幅、高さを示します。

max 折れ線グラフのスケールの最大値、最小値を示します。
min

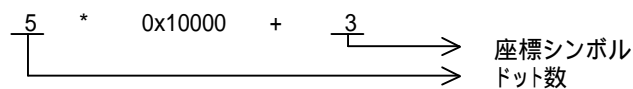
sym	上位16ビット	...	座標シンボルを示します。
	下位16ビット	...	座標シンボルの大きさをドット数で示します。

座標シンボルを以下に示します。

値	座標シンボル
1	
2	
3	x
4	
5	

注) 座標シンボルが NULL の場合、シンボルは表示されません。

< 例 > ドット数が5の座標シンボルxを表示する時には次の様に指定します。



*symcol 座標シンボルの色を示します。

wmode 間隔のモードを以下に示します。

モード	値	説 明
固定	HRFIX	間隔が固定
可変(相対)	HRPRV	間隔が前点からの相対
可変(絶対)	HRORG	間隔が原点からの絶対

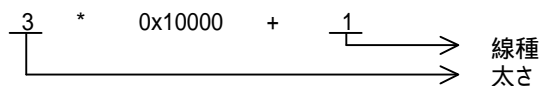
width 固定のデータの間隔を示します。
(間隔モードが固定時のみ)

ptn 上位16ビット … 線の太さを示します。
下位16ビット … 線種を示します。

線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	— — — — —
点線	2	- - - - -
一点破線	3	— — — — —
二点破線	4	— — — — —

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



*col 折れ線グラフの線の色を示します。

pc 折れ線グラフのデータの数を示します。

*pt 折れ線グラフのデータアドレスを示します。但し、間隔モードにより
表示データの構造体が異なります。
(実際のデータは表示データ構造体(Grptgr1・Grptgr2)の型を持つ
配列データです。)

間隔モード : 固定 構造体名 *Grptgr1
間隔モード : 可変 (相対、絶対) 構造体名 *Grptgr2

表示データ構造体 (Grptgr1)

data			折れ線グラフのデータを示します。
flg	0	...	表示しません。
	1	...	表示します。
	2	...	グラフ終端を示します。

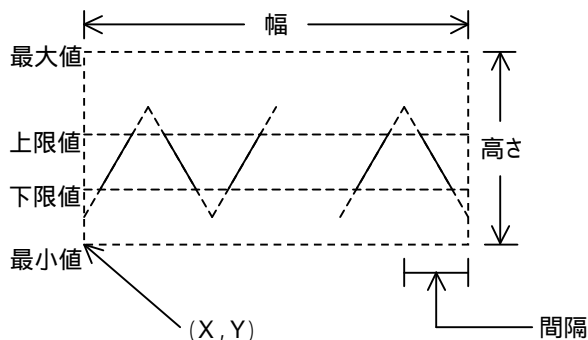
表示データ構造体 (Grptgr2)

data			折れ線グラフのデータを示します。
flg	0	...	表示しません。
	1	...	表示します。
	2	...	グラフ終端を示します。
dw			前データとの間隔値を示します。 (間隔モードが可変 (相対、絶対) 時のみ)
	相対	...	前データとの間隔値
	絶対	...	原点からの間隔値

注) データの数及び座標シンボルを変更し、再表示する様な場合は TKgrdelete、TKgrentry をする必要があります。

機能

データにより折れ線グラフを表示します。



構造体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 原点X座標 */
    int      y;          /* 原点Y座標 */
    int      w;          /* 表示幅 */
    int      h;          /* 表示高さ */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      limitup;    /* 上限値 */
    int      limitlow;   /* 下限値 */
    int      width;      /* 間隔 */
    int      ptn1;       /* 下限値以上、上限値以下の線の太さ/線種 */
    char     *col1;      /* 下限値以上、上限値以下の線色 */
    int      ptn2;       /* 上限値以上の線の太さ/線種 */
    char     *col2;      /* 上限値以上の線色 */
    int      ptn3;       /* 下限値以下の線の太さ/線種 */
    char     *col3;      /* 下限値以下の線色 */
    int      pc;         /* データ数 */
    Grptgr1  *pt;        /* 表示データアドレス */
} Grline3;
```

```
typedef struct          /* 表示データ構造体 */
{
    int      data;       /* データ */
    int      flg;        /* フラグ */
} Grptgr1;
```

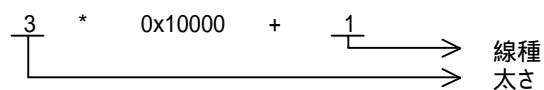
メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	折れ線グラフの原点を示します。
w h	折れ線グラフ表示エリアの幅、高さを示します。
max min	折れ線グラフのスケールの最大値、最小値を示します。
limitup limitlow	上限値、下限値を示します。
width	各データの間隔を示します。
ptn1	下限値以上、上限値以下。 上位16ビット … 線の太さを示します。 下位16ビット … 線種を示します。

線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	— — — — —
点線	2	- - - - -
一点破線	3	— — — — —
二点破線	4	— — — — —

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



*col1	下限値以上、上限値以下の線の色を示します。
ptn2	上限値以上。 ptn1と同様。
*col2	上限値以上の線の色を示します。

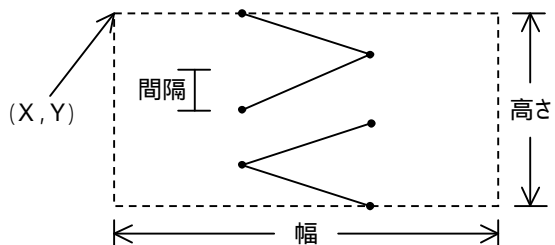
ptn3	下限値以下。 ptn1と同様。
*col3	下限値以下の線の色を示します。
pc	折れ線グラフのデータの数を示します。
*pt	折れ線グラフのデータアドレスを示します。 (実際のデータは表示データ構造体(Grptgr1)の型を持つ配列データです。)

表示データ構造体(Grptgr1)

data	折れ線グラフのデータを示します。
flg	0 … 表示しません。 1 … 表示します。 2 … グラフ終端を示します。

機 能

データにより縦折れ線グラフを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;    /* グラフヘッダ */
    int      x;      /* 原点X座標 */
    int      y;      /* 原点Y座標 */
    int      w;      /* 表示幅 */
    int      h;      /* 表示高さ */
    int      max;    /* スケール最大値 */
    int      min;    /* スケール最小値 */
    int      sym;    /* 座標シンボル/大きさ */
    char     *symcol; /* シンボル色 */
    int      wmode;  /* 間隔モード */
    int      width;  /* 固定間隔値 */
    int      ptn;    /* 太さ/線種 */
    char     *col;   /* 線色 */
    int      pc;     /* データ数 */
    Grptgr1  *pt;    /* 表示データアドレス */
} Grtline2;
```

```
typedef struct          /* 間隔固定 表示データ構造体          */
{
    int      data;      /* データ              */
    int      flg;       /* フラグ              */
} Grptgr1;
```

```
typedef struct          /* 間隔可変 表示データ構造体          */
{
    int      data;      /* データ              */
    int      flg;       /* フラグ              */
    int      dw;        /* 可変間隔値          */
} Grptgr2;
```

メンバー説明

hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

x 縦折れ線グラフの原点を示します。

y

w 縦折れ線グラフ表示エリアの幅、高さを示します。

h

max 縦折れ線グラフのスケールの最大値、最小値を示します。

min

sym 上位16ビット …… 座標シンボルを示します。
 下位16ビット …… 座標シンボルの大きさをドット数で示します。

座標シンボルを以下に示します。

値	座標シンボル
1	
2	
3	x
4	
5	

注) 座標シンボルが NULL の場合、シンボルは表示されません。

< 例 > ドット数が5の座標シンボルxを表示する時には次の様に指定します。

$$\frac{5}{\text{座標シンボル}} \times 0x10000 + \frac{3}{\text{ドット数}}$$

*symcol 座標シンボルの色を示します。

wmode 間隔のモードを以下に示します。

モード	値	説 明
固定	HRFIX	間隔が固定
可変(相対)	HRPRV	間隔が前点からの相対
可変(絶対)	HRORG	間隔が原点からの絶対

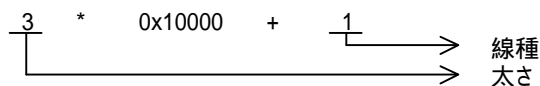
width 固定のデータの間隔を示します。
(間隔モードが固定時のみ)

ptn 上位16ビット … 線の太さを示します。
下位16ビット … 線種を示します。

線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	— — — — —
点線	2	- - - - -
一点破線	3	— — — — —
二点破線	4	— — — — —

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



*col 縦折れ線グラフの線の色を示します。

pc 縦折れ線グラフのデータの数を示します。

*pt 縦折れ線グラフのデータアドレスを示します。但し、間隔モードにより表示データの構造体が異なります。
(実際のデータは表示データ構造体(Grptgr1・Grptgr2)の型を持つ配列データです。)

間隔モード : 固定 構造体名 *Grptgr1
間隔モード : 可変 (相対、絶対) 構造体名 *Grptgr2

表示データ構造体 (Grptgr1)

data	縦折れ線グラフのデータを示します。		
flg	0	...	表示しません。
	1	...	表示します。
	2	...	グラフ終端を示します。

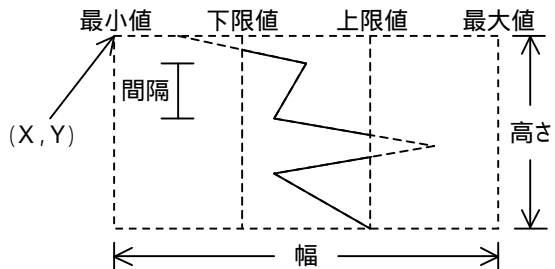
表示データ構造体 (Grptgr2)

data	縦折れ線グラフのデータを示します。		
flg	0	...	表示しません。
	1	...	表示します。
	2	...	グラフ終端を示します。
dw	前データとの間隔値を示します。 (間隔モードが可変 (相対、絶対) 時のみ)		
	相対	...	前データとの間隔値
	絶対	...	原点からの間隔値

注) データの数及び座標シンボルを変更し、再表示する様な場合は TKgrdelete、TKgrentry をする必要があります。

機 能

データにより縦折れ線グラフを表示します。

**構 造 体**

```
typedef struct
{
    Grobj    hed;           /* グラフヘッダ */
    int      x;             /* 原点X座標 */
    int      y;             /* 原点Y座標 */
    int      w;             /* 表示幅 */
    int      h;             /* 表示高さ */
    int      max;           /* スケール最大値 */
    int      min;           /* スケール最小値 */
    int      limitup;       /* 上限値 */
    int      limitlow;      /* 下限値 */
    int      width;         /* 間隔 */
    int      ptn1;          /* 下限値以上、上限値以下の線の太さ/線種 */
    char     *col1;         /* 下限値以上、上限値以下の線色 */
    int      ptn2;          /* 上限値以上の線の太さ/線種 */
    char     *col2;         /* 上限値以上の線色 */
    int      ptn3;          /* 下限値以下の線の太さ/線種 */
    char     *col3;         /* 下限値以下の線色 */
    int      pc;            /* データ数 */
    Grptgr1  *pt;           /* 表示データアドレス */
} Grtline3;

typedef struct          /* 表示データ構造体 */
{
    int      data;         /* データ */
    int      flg;          /* フラグ */
} Grptgr1;
```

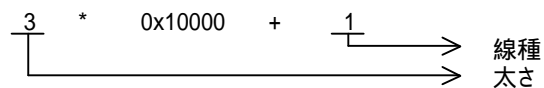
メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	縦折れ線グラフの原点を示します。
w h	縦折れ線グラフ表示エリアの幅、高さを示します。
max min	縦折れ線グラフのスケールの最大値、最小値を示します。
limitup limitlow	上限値、下限値を示します。
width	各データの間隔を示します。
ptn1	下限値以上、上限値以下。 上位16ビット … 線の太さを示します。 下位16ビット … 線種を示します。

線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	— — — — —
点線	2	- - - - -
一点破線	3	— — — — —
二点破線	4	— — — — —

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



*col1	下限値以上、上限値以下の線の色を示します。
ptn2	上限値以上。 ptn1と同様。
*col2	上限値以上の線の色を示します。

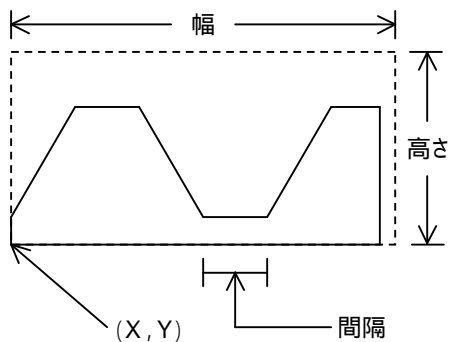
ptn3	下限値以下。 ptn1と同様。
*col3	下限値以下の線の色を示します。
pc	縦折れ線グラフのデータの数を示します。
*pt	縦折れ線グラフのデータアドレスを示します。 (実際のデータは表示データ構造体(Grptgr1)の型を持つ配列データです。)

表示データ構造体(Grptgr1)

data	縦折れ線グラフのデータを示します。
flg	0 … 表示しません。 1 … 表示します。 2 … グラフ終端を示します。

機 能

データによりエリアグラフを表示します。



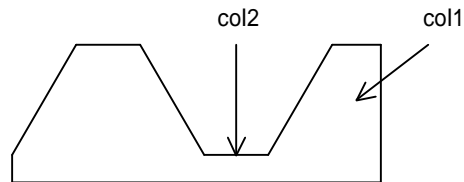
構 造 体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 原点X座標 */
    int      y;          /* 原点Y座標 */
    int      w;          /* 表示幅 */
    int      h;          /* 表示高さ */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      width;      /* 間隔 */
    char     *col1;      /* 塗りつぶし色 */
    char     *col2;      /* 枠色 */
    int      pc;         /* データ数 */
    Grptdt   *pt;        /* 表示データアドレス */
} Grarea1;
```

```
typedef struct          /* 表示データ構造体 */
{
    int      data;      /* データ */
} Grptdt;
```

メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	エリアグラフの原点を示します。
w h	エリアグラフ表示エリアの幅、高さを示します。
max min	エリアグラフのスケールの最大値、最小値を示します。
width	グラフで表示する点の間隔を示します。
*col1 *col2	エリアグラフの表示色を示します。



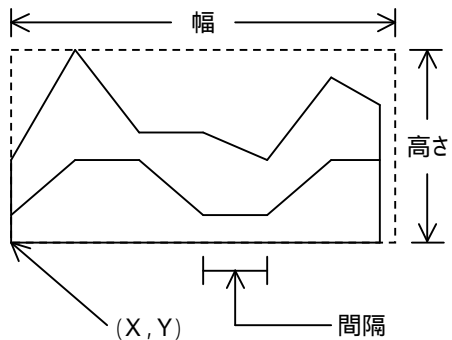
pc	エリアグラフのデータの数を示します。
*pt	エリアグラフのデータアドレスを示します。 (実際のデータは表示データ構造体 (Grptdt) の型を持つ配列データです。)

表示データ構造体 (Grptdt)

data	エリアグラフのデータを示します。
------	------------------

機 能

データにより積み上げエリアグラフを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 原点X座標 */
    int      y;          /* 原点Y座標 */
    int      w;          /* 表示幅 */
    int      h;          /* 表示高さ */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      width;      /* 間隔 */
    int      cnt;        /* 表示エリアグラフ数 */
    int      pc;         /* データ数 */
    Gradat   *dat;       /* 表示データアドレス */
} Grarea2;
```

```
typedef struct
{
    char      *col1;      /* 塗りつぶし色 */
    char      *col2;      /* 枠色 */
    Grpdt;     *dat;      /* データアドレス */
} Gradat;
```

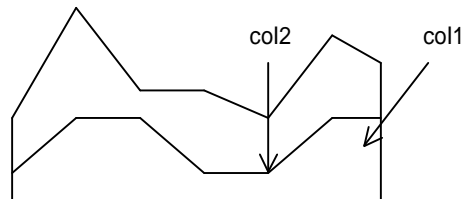
```
typedef struct          /* 表示データ構造体 */
{
    int          data;    /* データ */
} Grpdt;
```

メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	積み上げエリアグラフの原点を示します。
w h	積み上げエリアグラフ表示エリアの幅、高さを示します。
max min	積み上げエリアグラフのスケールの最大値、最小値を示します。
width	グラフで表示する点の間隔を示します。
cnt	表示する積み上げエリアグラフの数を示します。
pc	積み上げエリアグラフのデータの数を示します。
*dat	積み上げエリアグラフのデータアドレスを示します。

表示データ構造体 (Gradat)

*col1	積み上げエリアグラフの表示色を示します。
*col2	



*dat	積み上げエリアグラフのデータアドレスを示します。 (実際のデータは表示データ構造体 (Grptdt) の型を持つ配列データです。)
------	--

表示データ構造体 (Grptdt)

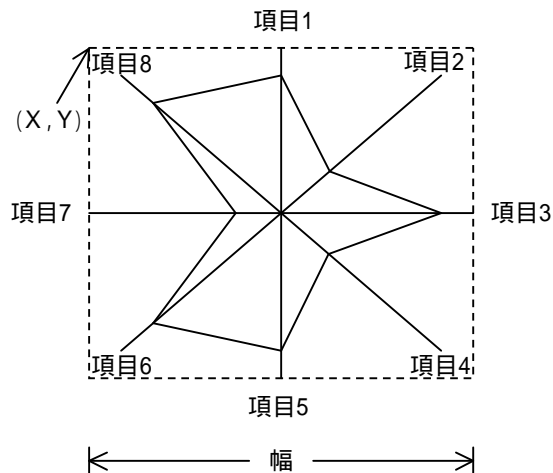
data	積み上げエリアグラフのデータを示します。
------	----------------------

レーダーチャート

HRIlader

機 能

データによりレーダーチャートを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 表示位置 X座標 */
    int      y;          /* 表示位置 Y座標 */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      w;          /* 表示幅 */
    int      ptn;        /* 太さ/線種 */
    char     *col;        /* 線色 */
    int      pc;          /* データ数 */
    Grkdat   *kdat;       /* 項目データアドレス */
    Grptdt   *dat;        /* 表示データアドレス */
} Grlader;
```

```
typedef struct
{
    char     *kdat;       /* 項目データ */
} Grkdat;
```

```
typedef struct          /* 表示データ構造体 */
{
    int      data;        /* データ */
} Grptdt;
```

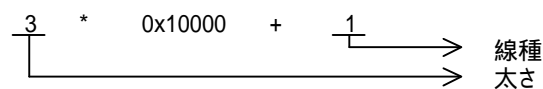

メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	レーダーチャートの表示位置を示します。
max min	レーダーチャートのスケールの最大値、最小値を示します。
w	レーダーチャートの幅を示します。 (幅 = 高さの正方形になります。)
ptn	上位16ビット ... 線の太さを示します。 下位16ビット ... 線種を示します。

線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	— — — — —
点線	2	- - - - -
一点破線	3	— — — — —
二点破線	4	— — — — —

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



*col	レーダーチャートの表示色を示します。
pc	レーダーチャートのデータの数を示します。
*kdat	レーダーチャートの項目データアドレスを示します。 (実際のデータは項目データ構造体(Grkdat)の型を持つ配列データです。)
*dat	レーダーチャートのデータアドレスを示します。 (実際のデータは表示データ構造体(Grptdt)の型を持つ配列データです。)

項目データ構造体 (Grkdat)

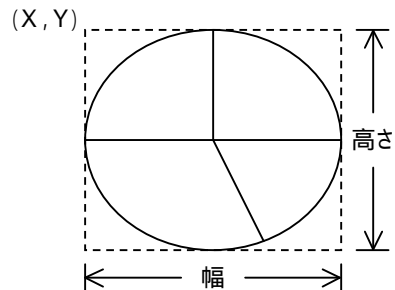
*kdat レーダーチャートの項目のアドレスを示します。

表示データ構造体 (Grptdt)

data レーダーチャートのデータを示します。

機 能

データにより円グラフを表示します。

**構 造 体**

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ          */
    int        mode;     /* モード                */
    int        x;        /* 表示位置 X座標        */
    int        y;        /* 表示位置 Y座標        */
    int        w;        /* 表示幅                */
    int        h;        /* 表示高さ              */
    int        pc;       /* データ数              */
    Grendt     *pt;      /* 表示データアドレス    */
} Grarc;
```

```
typedef struct          /* 表示データ構造体      */
{
    unsigned int data;   /* データ                */
    char        *color;  /* 表示色データ          */
} Grendt;
```

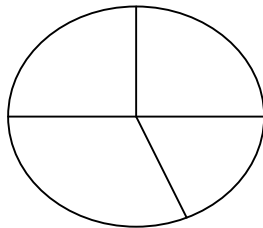
```
typedef struct
{
    unsigned int data;   /* データ                */
    char        *color;  /* 表示色データ          */
    int         flg;     /* 強調フラグ            */
} Greng;
```

メンバー説明

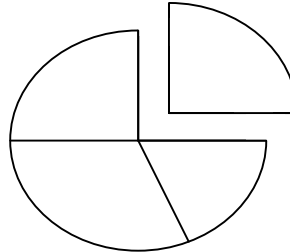
hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

mode 円グラフのモードを以下に示します。

0 … 円グラフ
1 … 強調円グラフ



円グラフ



強調円グラフ

x 円グラフの表示位置を示します。
y

w 円グラフの表示幅、高さを示します。
h

pc 円グラフのデータの数を示します。

*pt 円グラフのデータアドレスを示します。但し、モードにより表示データの
 構造体が異なります。
 (実際のデータは表示データ構造体 (Grendt・Grengt) の型を持つ
 配列データです。)

モード	(円グラフ)	構造体名	*Grendt
	(強調円グラフ)	構造体名	*Grengt

表示データ構造体 (Grendt)

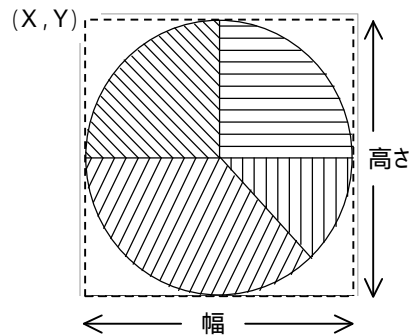
data 円グラフのデータを示します。
*color 円グラフの表示色を示します。

表示データ構造体 (Grenggr)

data 円グラフのデータを示します。
*color 円グラフの表示色を示します。
flg 0 ... 強調しません。
 1 ... 強調します。
 (強調円グラフ時のみ)

機 能

データにより円グラフを表示します。



構 造 体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ          */
    int        mode;     /* モード                */
    int        x;        /* 表示位置 X座標        */
    int        y;        /* 表示位置 Y座標        */
    int        w;        /* 表示幅                */
    int        h;        /* 表示高さ              */
    int        pc;       /* データ数              */
    Grendt2    *pt;      /* 表示データアドレス    */
} Grarc2;
```

```
typedef struct          /* 表示データ構造体      */
{
    unsigned int data;   /* データ                */
    int        ptn;     /* 網掛けパターン        */
    char       *col1;    /* 塗りつぶしフォア色    */
    char       *col2;    /* 塗りつぶしバック色    */
} Grendt2;
```

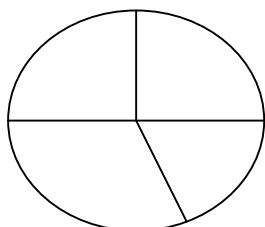
```
typedef struct
{
    unsigned int data;   /* データ                */
    int        ptn;     /* 網掛けパターン        */
    char       *col1;    /* 塗りつぶしフォア色    */
    char       *col2;    /* 塗りつぶしバック色    */
    int        flg;     /* 強調フラグ            */
} Greng2;
```

メンバー説明

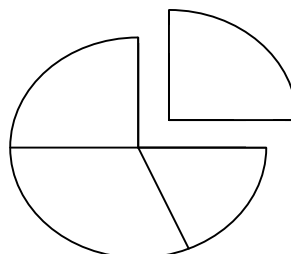
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 円グラフのモードを以下に示します。

0 …… 円グラフ
1 …… 強調円グラフ



円グラフ



強調円グラフ

x 円グラフの表示位置を示します。

y

w 円グラフの表示幅、高さを示します。

h

pc 円グラフのデータの数を示します。






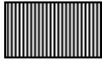
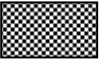
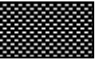
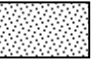
*pt 円グラフのデータアドレスを示します。但し、モードにより表示データの構造体が異なります。
(実際のデータは表示データ構造体 (Grendt2・Greng2) の型を持つ配列データです。)

モード	(円グラフ)	構造体名	*Grendt2
	(強調円グラフ)	構造体名	*Greng2

表示データ構造体 (Grendt2)

data 円グラフのデータを示します。

ptn 網掛けパターンを示します。

-1	0	1	2	3
				
4	5	6	7	
				




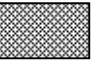





*col1 網掛けのフォア色を示します。

*col2 網掛けのバック色を示します。

表示データ構造体 (Greng2)

data 円グラフのデータを示します。

ptn 網掛けパターンを示します。

-1	0	1	2	3
				
4	5	6	7	
				

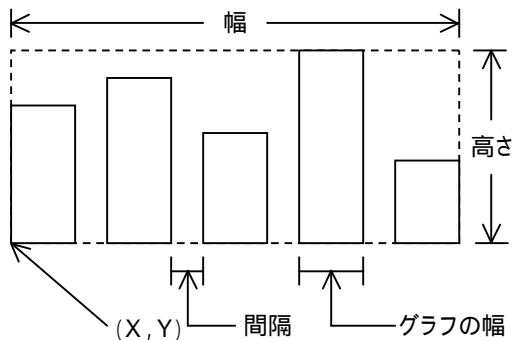
*col1 網掛けのフォア色を示します。

*col2 網掛けのバック色を示します。

flg 0 ... 強調しません。
1 ... 強調します。
(強調円グラフ時のみ)

機 能

データにより棒グラフを表示します。
(最小値から上向き又は右向きに棒グラフが表示されます。)



構 造 体

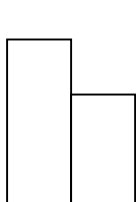
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 棒グラフの幅 */
    int        w2;       /* 棒グラフの間隔 */
    int        w3;       /* 棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    unsigned int data;   /* データ */
    char       *col;     /* 表示色 */
} Grbar;
```

メンバー説明

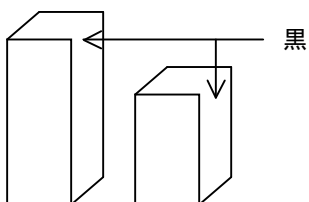
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 棒グラフのモードを以下に示します。

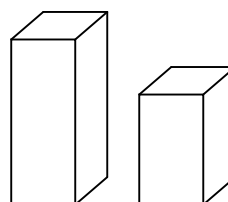
- 0 ... 縦棒グラフ
- 1 ... 縦棒グラフ (影付き)
- 2 ... 縦棒グラフ (立体)
- 3 ... 横棒グラフ
- 4 ... 横棒グラフ (影付き)
- 5 ... 横棒グラフ (立体)



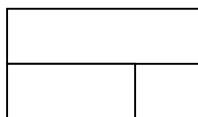
縦棒グラフ



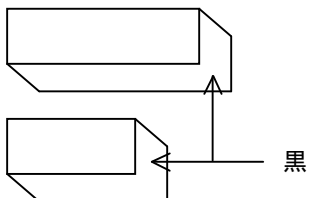
縦棒グラフ (影付き)



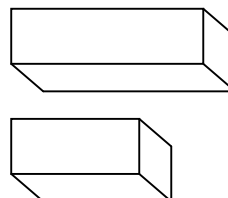
縦棒グラフ (立体)



横棒グラフ



横棒グラフ (影付き)



横棒グラフ (立体)

x 棒グラフの原点を示します。

y

w 棒グラフ表示エリアの幅、高さを示します。

h

max 棒グラフのスケールの最大値、最小値を示します。

min

w1 棒グラフの幅を示します。

w2 棒グラフの間隔を示します。

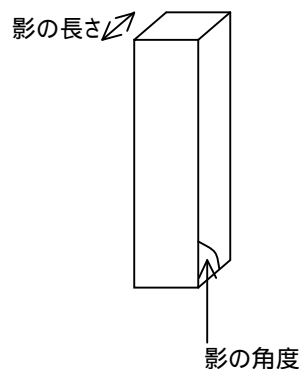
w3 棒グラフの影の長さを示します。

kaku 棒グラフの影の角度を示します。

no 何番目の棒グラフかを示します。

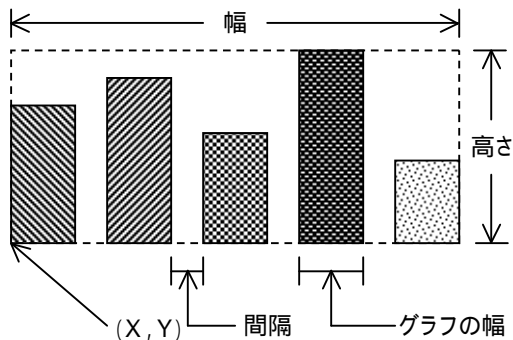
data 棒グラフのデータを示します。

*col 棒グラフの表示色を示します。



機 能

データにより棒グラフを表示します。
(最小値から上向きまたは右向きに棒グラフが表示されます。)



構 造 体

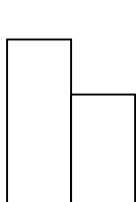
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 棒グラフの幅 */
    int        w2;       /* 棒グラフの間隔 */
    int        w3;       /* 棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    unsigned int data;   /* データ */
    int        ptn;      /* 網掛けパターン */
    char       *col1;    /* 塗りつぶしフォア色 */
    char       *col2;    /* 塗りつぶしバック色 */
} Grbar2;
```

メンバー説明

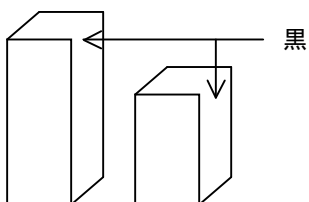
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 棒グラフのモードを以下に示します。

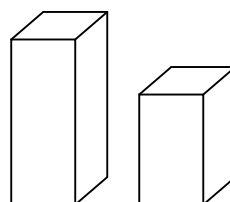
- 0 ... 縦棒グラフ
- 1 ... 縦棒グラフ (影付き)
- 2 ... 縦棒グラフ (立体)
- 3 ... 横棒グラフ
- 4 ... 横棒グラフ (影付き)
- 5 ... 横棒グラフ (立体)



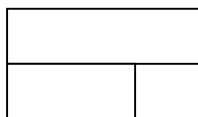
縦棒グラフ



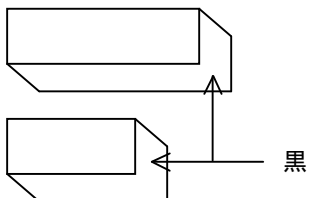
縦棒グラフ (影付き)



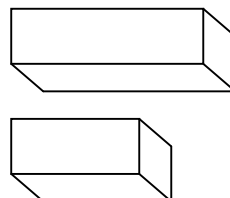
縦棒グラフ (立体)



横棒グラフ



横棒グラフ (影付き)



横棒グラフ (立体)

x 棒グラフの原点を示します。

y

w 棒グラフ表示エリアの幅、高さを示します。

h

max 棒グラフのスケールの最大値、最小値を示します。

min

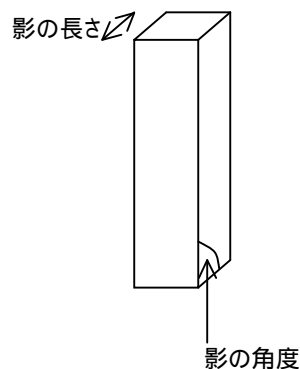
w1 棒グラフの幅を示します。

w2 棒グラフの間隔を示します。

w3 棒グラフの影の長さを示します。




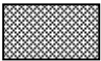





kaku 棒グラフの影の角度を示します。

no 何番目の棒グラフかを示します。



data 棒グラフのデータを示します。

ptn 網掛けパターンを示します。

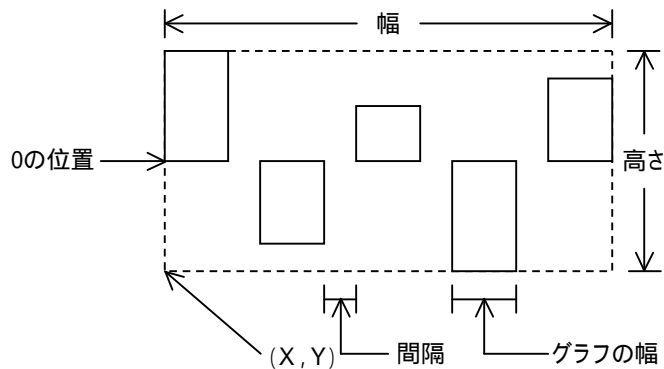
-1	0	1	2	3
				
4	5	6	7	
				

*col1 網掛けのフォア色を示します。

*col2 網掛けのバック色を示します。

機 能

データにより棒グラフを表示します。
 (0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに棒グラフが表示されます。)



構 造 体

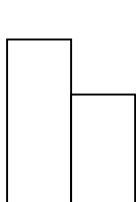
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 棒グラフの幅 */
    int        w2;       /* 棒グラフの間隔 */
    int        w3;       /* 棒グラフの影の長さ */
    double     kaku;      /* 影の角度 */
    int        no;       /* データNo. */
    int        data;     /* データ */
    char       *col;     /* 表示色 */
} Grbar3;
```

メンバー説明

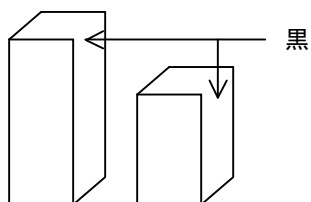
hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

mode 棒グラフのモードを以下に示します。

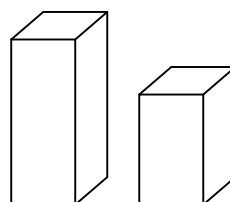
- 0 ... 縦棒グラフ
- 1 ... 縦棒グラフ (影付き)
- 2 ... 縦棒グラフ (立体)
- 3 ... 横棒グラフ
- 4 ... 横棒グラフ (影付き)
- 5 ... 横棒グラフ (立体)



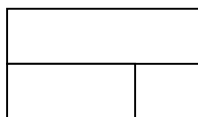
縦棒グラフ



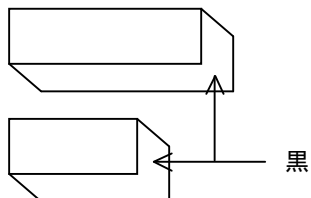
縦棒グラフ (影付き)



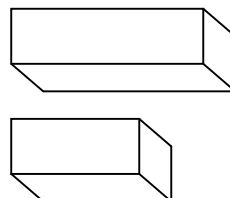
縦棒グラフ (立体)



横棒グラフ



横棒グラフ (影付き)



横棒グラフ (立体)

x 棒グラフの原点を示します。

y

w 棒グラフ表示エリアの幅、高さを示します。

h

max 棒グラフのスケールの最大値、最小値を示します。

min

w1 棒グラフの幅を示します。

w2 棒グラフの間隔を示します。

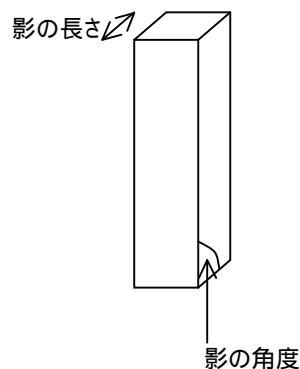
w3 棒グラフの影の長さを示します。

kaku 棒グラフの影の角度を示します。

no 何番目の棒グラフかを示します。

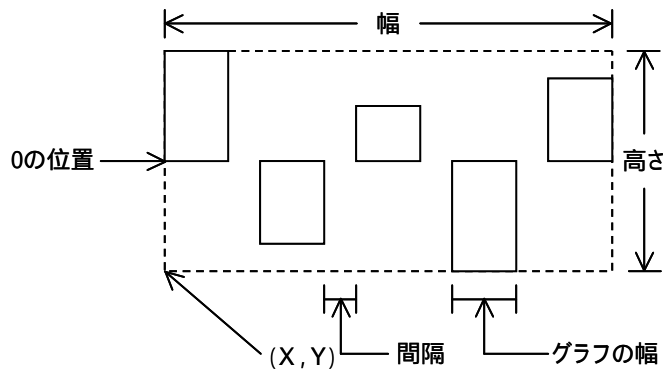
data 棒グラフのデータを示します。

*col 棒グラフの表示色を示します。



機 能

データにより棒グラフを表示します。
 (0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに棒グラフが表示されます。)

**構 造 体**

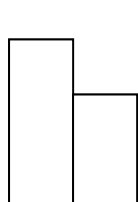
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ          */
    int        mode;     /* モード                */
    int        x;        /* 原点X座標              */
    int        y;        /* 原点Y座標              */
    int        w;        /* 表示幅                */
    int        h;        /* 表示高さ              */
    int        max;      /* スケール最大値        */
    int        min;      /* スケール最小値        */
    int        w1;       /* 棒グラフの幅          */
    int        w2;       /* 棒グラフの間隔        */
    int        w3;       /* 棒グラフの影の長さ    */
    double     kaku;     /* 影の角度              */
    int        no;       /* データNo.             */
    int        data;     /* データ                */
    char       *col;     /* 表示色                */
} Grbar3;
```

メンバー説明

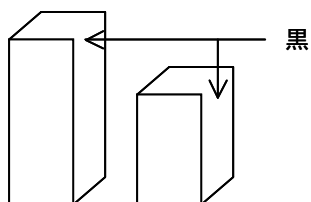
hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

mode 棒グラフのモードを以下に示します。 枠線および影は黒色

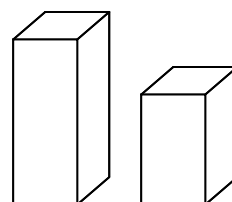
- 0 ... 縦棒グラフ (枠線なし)
- 1 ... 縦棒グラフ (枠線なし、影付き)
- 2 ... 縦棒グラフ (枠線あり、立体)
- 3 ... 横棒グラフ (枠線なし)
- 4 ... 横棒グラフ (枠線なし、影付き)
- 5 ... 横棒グラフ (枠線あり、立体)



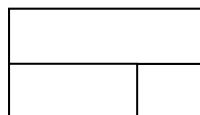
縦棒グラフ



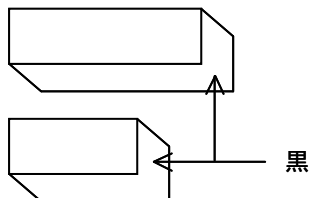
縦棒グラフ (影付き)



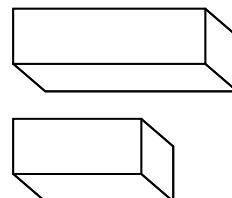
縦棒グラフ (立体)



横棒グラフ



横棒グラフ (影付き)



横棒グラフ (立体)

x 棒グラフの原点を示します。

y

w 棒グラフ表示エリアの幅、高さを示します。

h

max 棒グラフのスケールの最大値、最小値を示します。

min

w1 棒グラフの幅を示します。

w2 棒グラフの間隔を示します。

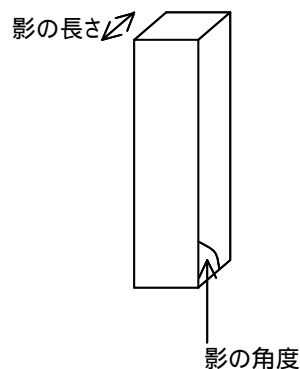
w3 棒グラフの影の長さを示します。

kaku 棒グラフの影の角度を示します。

no 何番目の棒グラフかを示します。

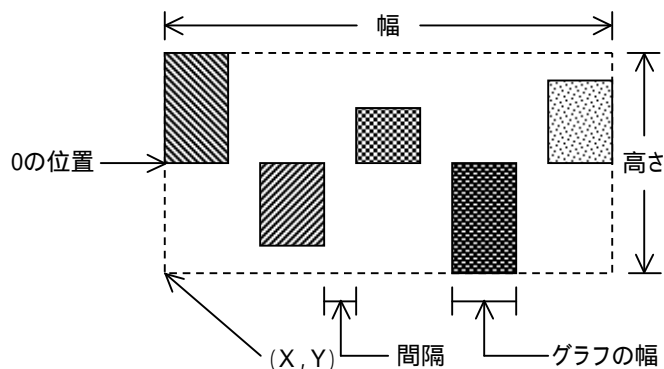
data 棒グラフのデータを示します。

*col 棒グラフの表示色を示します。



機 能

データにより棒グラフを表示します。
 (0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに棒グラフが表示されます。)



構 造 体

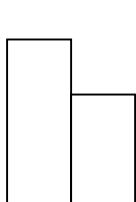
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 棒グラフの幅 */
    int        w2;       /* 棒グラフの間隔 */
    int        w3;       /* 棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    int        data;     /* データ */
    int        ptn;      /* 網掛けパターン */
    char       *col1;    /* 塗りつぶしフォア色 */
    char       *col2;    /* 塗りつぶしバック色 */
} Grbar4;
```

メンバー説明

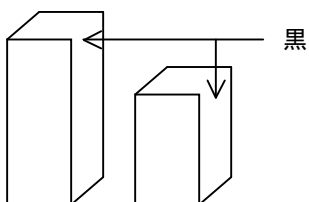
hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

mode 棒グラフのモードを以下に示します。

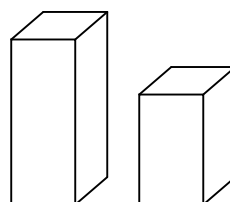
- 0 ... 縦棒グラフ
- 1 ... 縦棒グラフ (影付き)
- 2 ... 縦棒グラフ (立体)
- 3 ... 横棒グラフ
- 4 ... 横棒グラフ (影付き)
- 5 ... 横棒グラフ (立体)



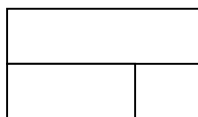
縦棒グラフ



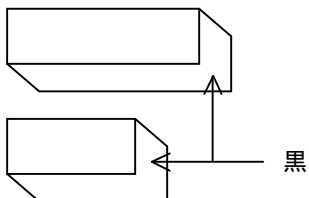
縦棒グラフ (影付き)



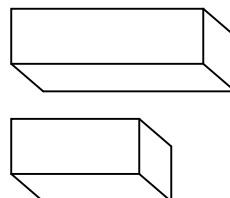
縦棒グラフ (立体)



横棒グラフ



横棒グラフ (影付き)



横棒グラフ (立体)

x 棒グラフの原点を示します。

y

w 棒グラフ表示エリアの幅、高さを示します。

h

max 棒グラフのスケールの最大値、最小値を示します。

min

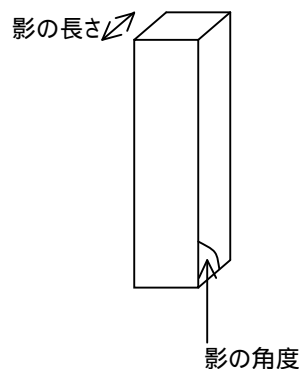
w1 棒グラフの幅を示します。

w2 棒グラフの間隔を示します。

w3 棒グラフの影の長さを示します。




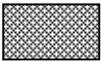





kaku 棒グラフの影の角度を示します。

no 何番目の棒グラフかを示します。



data 棒グラフのデータを示します。

ptn 網掛けパターンを示します。

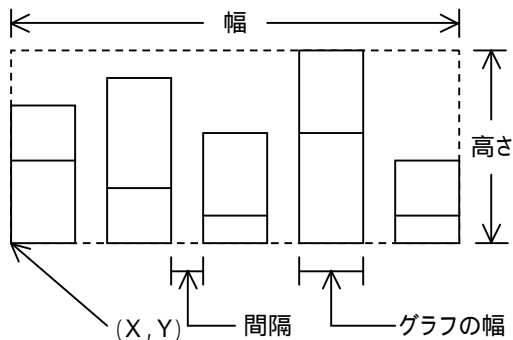
-1	0	1	2	3
				
4	5	6	7	
				

*col1 網掛けのフォア色を示します。

*col2 網掛けのバック色を示します。

機能

データにより積み上げ棒グラフを表示します。
(最小値から上向き又は右向きに積み上げ棒グラフが表示されます。)



構造体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 積み上げ棒グラフの幅 */
    int        w2;       /* 積み上げ棒グラフの間隔 */
    int        w3;       /* 積み上げ棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    int        pc;       /* データ数 */
    Grendt     *pt;      /* 表示データアドレス */
} Grtumi;
```

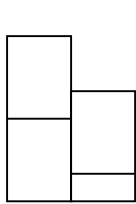
```
typedef struct /* 表示データ構造体 */
{
    unsigned int data; /* データ */
    char *color; /* 表示色データ */
} Grendt;
```

メンバー説明

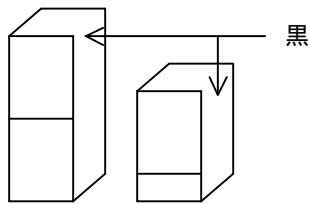
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 積み上げ棒グラフのモードを以下に示します。

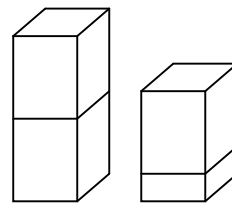
- 0 … 縦積み上げ棒グラフ
- 1 … 縦積み上げ棒グラフ (影付き)
- 2 … 縦積み上げ棒グラフ (立体)
- 3 … 横積み上げ棒グラフ
- 4 … 横積み上げ棒グラフ (影付き)
- 5 … 横積み上げ棒グラフ (立体)



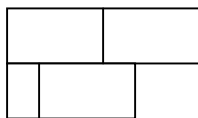
縦積み上げ棒グラフ



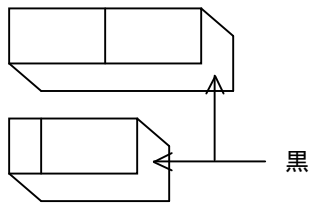
縦積み上げ棒グラフ (影付き)



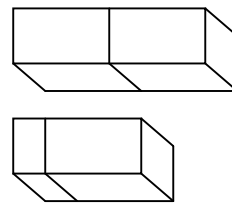
縦積み上げ棒グラフ (立体)



横積み上げ棒グラフ



横積み上げ棒グラフ (影付き)



横積み上げ棒グラフ (立体)

x 積み上げ棒グラフの原点を示します。

y

w 積み上げ棒グラフ表示エリアの幅、高さを示します。

h

max 積み上げ棒グラフのスケールの最大値、最小値を示します。

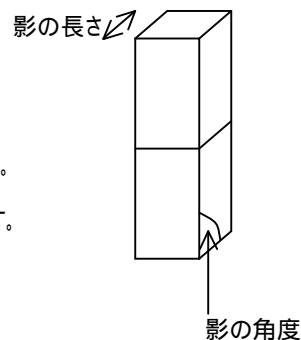
min

w1 積み上げ棒グラフの幅を示します。

w2 積み上げ棒グラフの間隔を示します。

w3 積み上げ棒グラフの影の長さを示します。

kaku 積み上げ棒グラフの影の角度を示します。



no	何番目の積み上げ棒グラフかを示します。
pc	積み上げ棒グラフのデータの数を示します。
*pt	積み上げ棒グラフのデータアドレスを示します。 (実際のデータは表示データ構造体 (Grendt) の型を持つ配列データです。)

表示データ構造体 (Grendt)

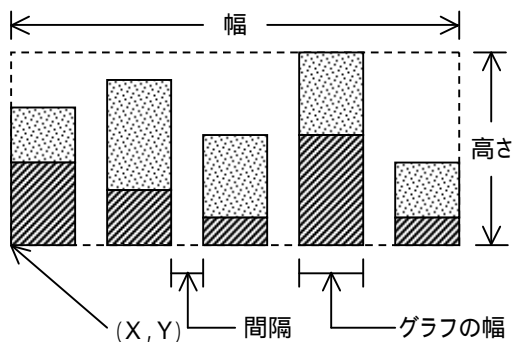
data	積み上げ棒グラフのデータを示します。
*color	積み上げ棒グラフの表示色を示します。

積み上げ棒グラフ2

HRtumi2

機 能

データにより積み上げ棒グラフを表示します。
(最小値から上向き又は右向きに積み上げ棒グラフが表示されます。)



構 造 体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 積み上げ棒グラフの幅 */
    int        w2;       /* 積み上げ棒グラフの間隔 */
    int        w3;       /* 積み上げ棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    int        pc;       /* データ数 */
    Grendt2    *pt;      /* 表示データアドレス */
} Grtumi2;
```

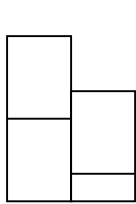
```
typedef struct      /* 表示データ構造体 */
{
    unsigned int    data;      /* データ */
    int             ptn;       /* 網掛けパターン */
    char            *col1;     /* 塗りつぶしフォア色 */
    char            *col2;     /* 塗りつぶしバック色 */
} Grendt2;
```


メンバー説明

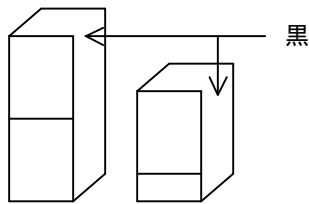
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 積み上げ棒グラフのモードを以下に示します。

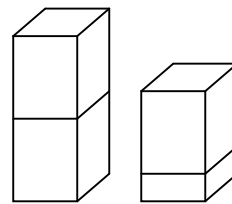
- 0 … 縦積み上げ棒グラフ
- 1 … 縦積み上げ棒グラフ (影付き)
- 2 … 縦積み上げ棒グラフ (立体)
- 3 … 横積み上げ棒グラフ
- 4 … 横積み上げ棒グラフ (影付き)
- 5 … 横積み上げ棒グラフ (立体)



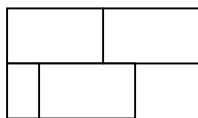
縦積み上げ棒グラフ



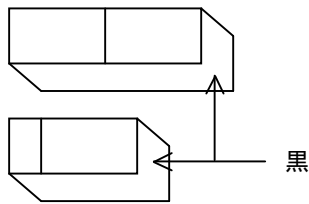
縦積み上げ棒グラフ (影付き)



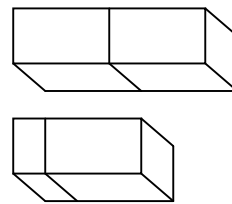
縦積み上げ棒グラフ (立体)



横積み上げ棒グラフ



横積み上げ棒グラフ (影付き)



横積み上げ棒グラフ (立体)

x 積み上げ棒グラフの原点を示します。

y

w 積み上げ棒グラフ表示エリアの幅、高さを示します。

h

max 積み上げ棒グラフのスケールの最大値、最小値を示します。

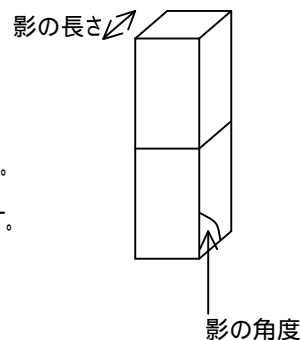
min

w1 積み上げ棒グラフの幅を示します。

w2 積み上げ棒グラフの間隔を示します。

w3 積み上げ棒グラフの影の長さを示します。






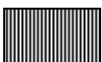



kaku 積み上げ棒グラフの影の角度を示します。



- no 何番目の積み上げ棒グラフかを示します。
- pc 積み上げ棒グラフのデータの数を示します。
- *pt 積み上げ棒グラフのデータアドレスを示します。
(実際のデータは表示データ構造体(Grendt2)の型を持つ配列データです。)

表示データ構造体 (Grendt2)

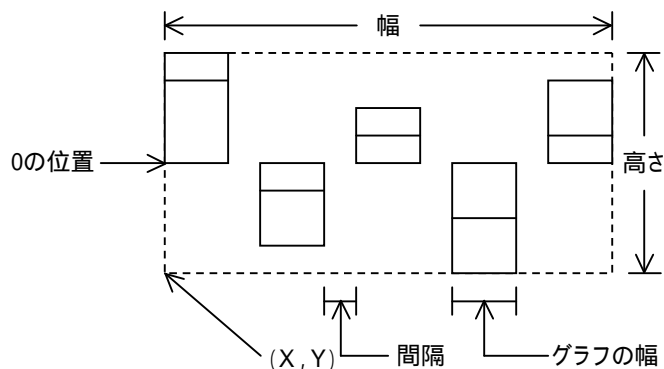
- data 積み上げ棒グラフのデータを示します。
- ptn 網掛けパターンを示します。

-1	0	1	2	3
				
4	5	6	7	
				

- *col1 網掛けのフォア色を示します。
- *col2 網掛けのバック色を示します。

機 能

データにより積み上げ棒グラフを表示します。
(0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに積み上げ棒グラフが表示されます。)



構 造 体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 積み上げ棒グラフの幅 */
    int        w2;       /* 積み上げ棒グラフの間隔 */
    int        w3;       /* 積み上げ棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    int        pc;       /* データ数 */
    Grendt3    *pt;      /* 表示データアドレス */
} Grtumi3;
```

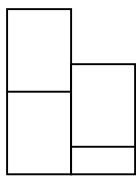
```
typedef struct          /* 表示データ構造体 */
{
    int        data;     /* データ */
    char       *color;   /* 表示色データ */
} Grendt3;
```

メンバー説明

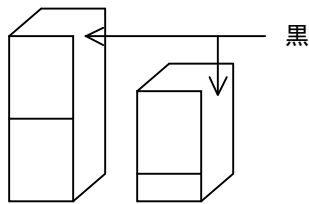
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 積み上げ棒グラフのモードを以下に示します。

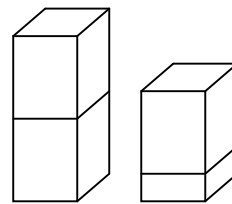
- 0 … 縦積み上げ棒グラフ
- 1 … 縦積み上げ棒グラフ (影付き)
- 2 … 縦積み上げ棒グラフ (立体)
- 3 … 横積み上げ棒グラフ
- 4 … 横積み上げ棒グラフ (影付き)
- 5 … 横積み上げ棒グラフ (立体)



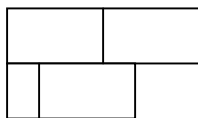
縦積み上げ棒グラフ



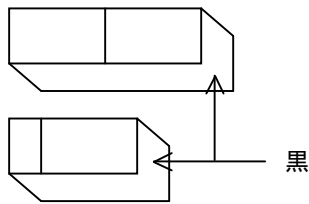
縦積み上げ棒グラフ (影付き)



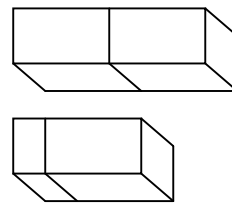
縦積み上げ棒グラフ (立体)



横積み上げ棒グラフ



横積み上げ棒グラフ (影付き)



横積み上げ棒グラフ (立体)

x 積み上げ棒グラフの原点を示します。

y

w 積み上げ棒グラフ表示エリアの幅、高さを示します。

h

max 積み上げ棒グラフのスケールの最大値、最小値を示します。

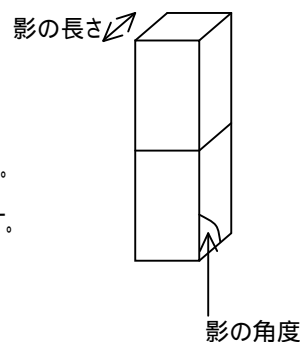
min

w1 積み上げ棒グラフの幅を示します。

w2 積み上げ棒グラフの間隔を示します。

w3 積み上げ棒グラフの影の長さを示します。

kaku 積み上げ棒グラフの影の角度を示します。



no	何番目の積み上げ棒グラフかを示します。
pc	積み上げ棒グラフのデータの数を示します。
*pt	積み上げ棒グラフのデータアドレスを示します。 (実際のデータは表示データ構造体 (Grendt3) の型を持つ配列データです。)

表示データ構造体 (Grendt3)

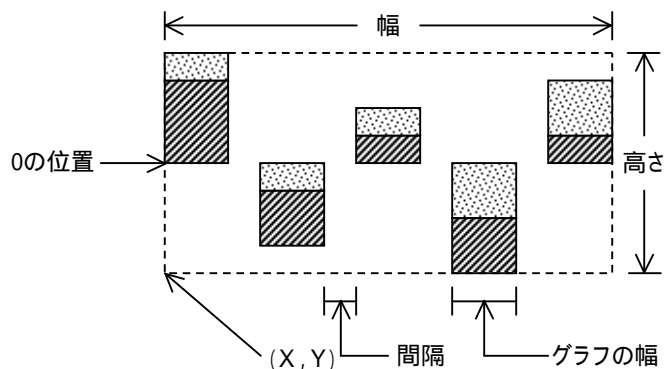
data	積み上げ棒グラフのデータを示します。
*color	積み上げ棒グラフの表示色を示します。

積み上げ棒グラフ4

HRtumi4

機 能

データにより積み上げ棒グラフを表示します。
(0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに積み上げ棒グラフが表示されます。)



構 造 体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h;        /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 積み上げ棒グラフの幅 */
    int        w2;       /* 積み上げ棒グラフの間隔 */
    int        w3;       /* 積み上げ棒グラフの影の長さ */
    double     kaku;     /* 影の角度 */
    int        no;       /* データNo. */
    int        pc;       /* データ数 */
    Grendt4    *pt       /* 表示データアドレス */
} Grtumi4;
```

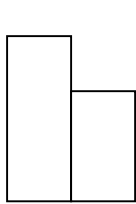
```
typedef struct          /* 表示データ構造体 */
{
    int        data;     /* データ */
    int        ptn;      /* 網掛けパターン */
    char       *col1;    /* 塗りつぶしフォア色 */
    char       *col2;    /* 塗りつぶしバック色 */
} Grendt4;
```

メンバー説明

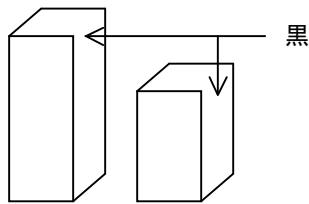
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 積み上げ棒グラフのモードを以下に示します。

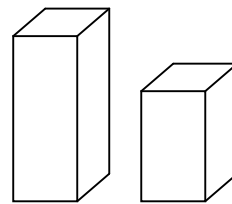
- 0 … 縦積み上げ棒グラフ
- 1 … 縦積み上げ棒グラフ (影付き)
- 2 … 縦積み上げ棒グラフ (立体)
- 3 … 横積み上げ棒グラフ
- 4 … 横積み上げ棒グラフ (影付き)
- 5 … 横積み上げ棒グラフ (立体)



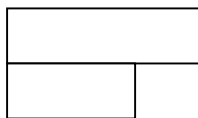
縦積み上げ棒グラフ



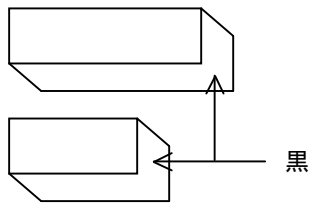
縦積み上げ棒グラフ (影付き)



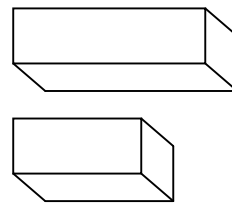
縦積み上げ棒グラフ (立体)



横積み上げ棒グラフ



横積み上げ棒グラフ (影付き)



横積み上げ棒グラフ (立体)

x 積み上げ棒グラフの原点を示します。

y

w 積み上げ棒グラフ表示エリアの幅、高さを示します。

h

max 積み上げ棒グラフのスケールの最大値、最小値を示します。

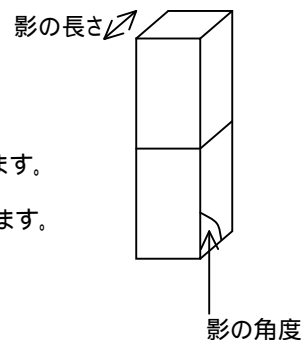
min

w1 積み上げ棒グラフの幅を示します。

w2 積み上げ棒グラフの間隔を示します。

w3 積み上げ棒グラフの影の長さを示します。






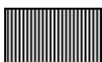



kaku 積み上げ棒グラフの影の角度を示します。



- no 何番目の積み上げ棒グラフかを示します。
- pc 積み上げ棒グラフのデータの数を示します。
- *pt 積み上げ棒グラフのデータアドレスを示します。
(実際のデータは表示データ構造体(Grendt4)の型を持つ配列データです。)

表示データ構造体 (Grendt4)

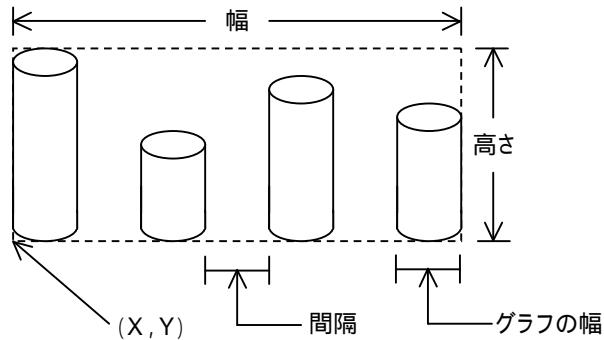
- data 積み上げ棒グラフのデータを示します。
- ptn 網掛けパターンを示します。

-1	0	1	2	3
				
4	5	6	7	
				

- *col1 網掛けのフォア色を示します。
- *col2 網掛けのバック色を示します。

機能

データにより円柱グラフを表示します。
(最小値から上向き又は右向きに円柱グラフが表示されます。)



構造体

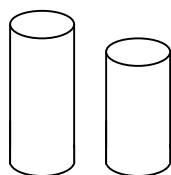
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h1;       /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 円柱グラフの幅 */
    int        w2;       /* 円柱グラフの間隔 */
    int        h2;       /* 底辺の高さ */
    int        no;       /* データNo. */
    unsigned int data;    /* データ */
    char       *col;     /* 表示色 */
} Grenc1;
```

メンバー説明

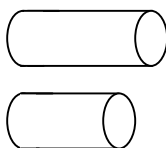
hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 円柱グラフのモードを以下に示します。

0 ... 縦円柱グラフ
1 ... 横円柱グラフ



縦円柱グラフ



横円柱グラフ

x 円柱グラフの原点を示します。

y

w 円柱グラフ表示エリアの幅、高さを示します。

h1

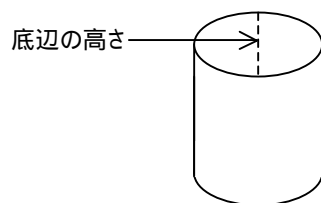
max 円柱グラフのスケールの最大値、最小値を示します。

min

w1 円柱グラフの幅を示します。

w2 円柱グラフの間隔を示します。

h2 円柱グラフの底辺の高さを示します。



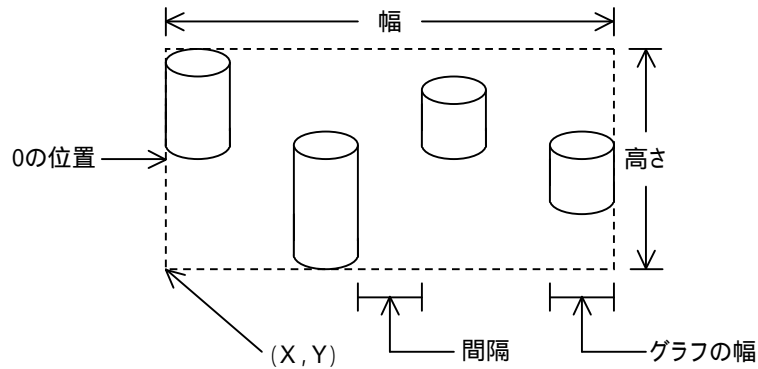
no 何番目の円柱グラフかを示します。

data 円柱グラフのデータを示します。

*col 円柱グラフの表示色を示します。

機 能

データにより円柱グラフを表示します。
 (0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに円柱グラフが表示されます。)

**構 造 体**

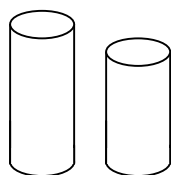
```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h1;       /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 円柱グラフの幅 */
    int        w2;       /* 円柱グラフの間隔 */
    int        h2;       /* 底辺の高さ */
    int        no;       /* データNo. */
    int        data;     /* データ */
    char       *col;     /* 表示色 */
} Grenc3;
```

メンバー説明

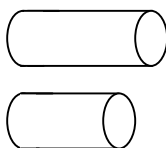
hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

mode 円柱グラフのモードを以下に示します。

0 ... 縦円柱グラフ
1 ... 横円柱グラフ



縦円柱グラフ



横円柱グラフ

x 円柱グラフの原点を示します。

y

w 円柱グラフ表示エリアの幅、高さを示します。

h1

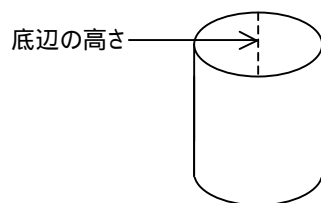
max 円柱グラフのスケールの最大値、最小値を示します。

min

w1 円柱グラフの幅を示します。

w2 円柱グラフの間隔を示します。

h2 円柱グラフの底辺の高さを示します。



no 何番目の円柱グラフかを示します。

data 円柱グラフのデータを示します。

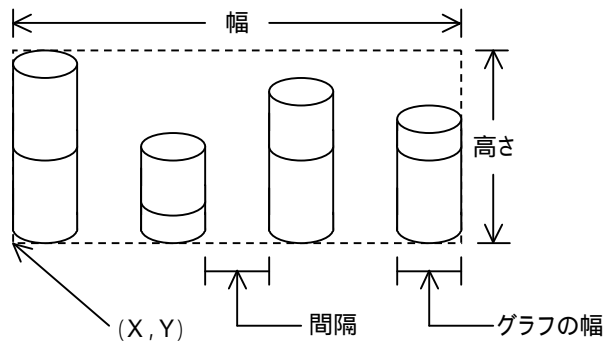
*col 円柱グラフの表示色を示します。

積み上げ円柱グラフ

HRencyu2

機能

データにより積み上げ円柱グラフを表示します。
(最小値から上向き又は右向きに積み上げ円柱グラフが表示されます。)



構造体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h1;       /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 積み上げ円柱グラフの幅 */
    int        w2;       /* 積み上げ円柱グラフの間隔 */
    int        h2;       /* 底辺の高さ */
    int        no;       /* データNo. */
    int        pc;       /* データ数 */
    Grendt     *pt;      /* 表示データアドレス */
} Grenc2;
```

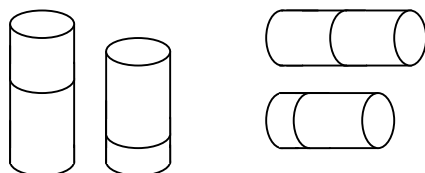
```
typedef struct      /* 表示データ構造体 */
{
    unsigned int    data; /* データ */
    char            *color; /* 表示色データ */
} Grendt;
```

メンバー説明

hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 積み上げ円柱グラフのモードを以下に示します。

0 ... 縦積み上げ円柱グラフ
1 ... 横積み上げ円柱グラフ



縦積み上げ円柱グラフ 横積み上げ円柱グラフ

x 積み上げ円柱グラフの原点を示します。
y

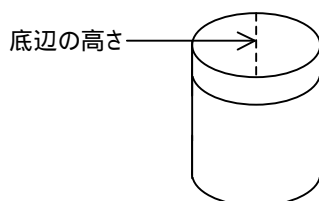
w 積み上げ円柱グラフ表示エリアの幅、高さを示します。
h1

max 積み上げ円柱グラフのスケールの最大値、最小値を示します。
min

w1 積み上げ円柱グラフの幅を示します。

w2 積み上げ円柱グラフの間隔を示します。

h2 積み上げ円柱グラフの底辺の高さを示します。



no 何番目の積み上げ円柱グラフかを示します。

pc 積み上げ円柱グラフのデータの数を示します。

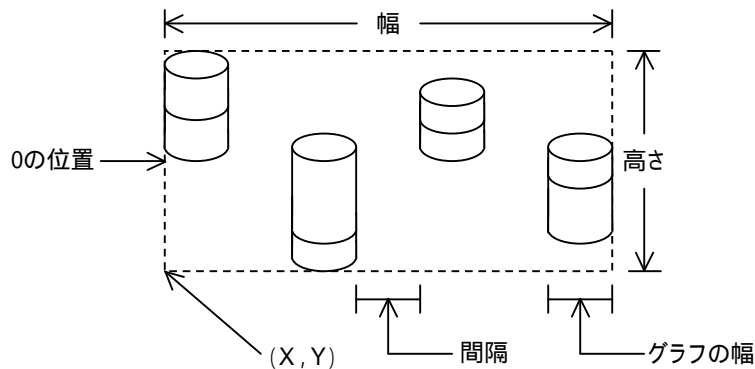
*pt 積み上げ円柱グラフのデータアドレスを示します。
(実際のデータは表示データ構造体(Grendt)の型を持つ配列データです。)

表示データ構造体 (Grendt)

data 積み上げ円柱グラフのデータを示します。
*color 積み上げ円柱グラフの表示色を示します。

機能

データにより積み上げ円柱グラフを表示します。
(0を基準に値が正の時は上向き又は右向き、値が負の時は下向き又は左向きに積み上げ円柱グラフが表示されます。)



構造体

```
typedef struct
{
    Grobj      hed;      /* グラフヘッダ */
    int        mode;     /* モード */
    int        x;        /* 原点X座標 */
    int        y;        /* 原点Y座標 */
    int        w;        /* 表示幅 */
    int        h1;       /* 表示高さ */
    int        max;      /* スケール最大値 */
    int        min;      /* スケール最小値 */
    int        w1;       /* 積み上げ円柱グラフの幅 */
    int        w2;       /* 積み上げ円柱グラフの間隔 */
    int        h2;       /* 底辺の高さ */
    int        no;       /* データNo. */
    int        pc;       /* データ数 */
    Grendt3    *pt;      /* 表示データアドレス */
} Grenc4;
```

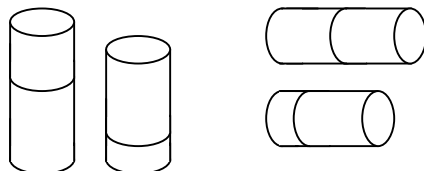
```
typedef struct          /* 表示データ構造体 */
{
    int        data;     /* データ */
    char       *color;   /* 表示色データ */
} Grendt3;
```


メンバー説明

hed グラフのヘッダ情報を示します。
(グラフ共通情報参照)

mode 積み上げ円柱グラフのモードを以下に示します。

0 … 縦積み上げ円柱グラフ
1 … 横積み上げ円柱グラフ



縦積み上げ円柱グラフ

横積み上げ円柱グラフ

x 積み上げ円柱グラフの原点を示します。
y

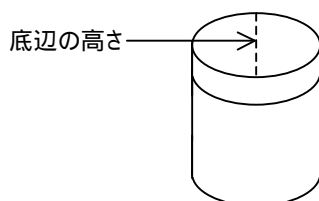
w 積み上げ円柱グラフ表示エリアの幅、高さを示します。
h1

max 積み上げ円柱グラフのスケールの最大値、最小値を示します。
min

w1 積み上げ円柱グラフの幅を示します。

w2 積み上げ円柱グラフの間隔を示します。

h2 積み上げ円柱グラフの底辺の高さを示します。



no 何番目の積み上げ円柱グラフかを示します。

pc 積み上げ円柱グラフのデータの数を示します。

*pt 積み上げ円柱グラフのデータアドレスを示します。
(実際のデータは表示データ構造体(Grendt3)の型を持つ配列データです。)

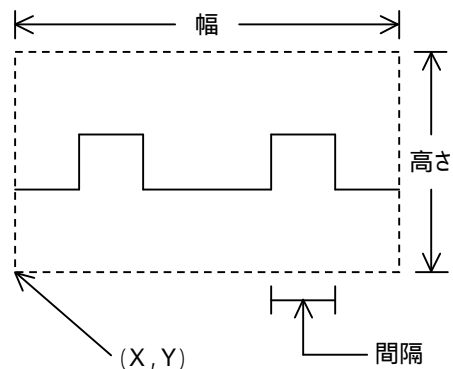
表示データ構造体 (Grendt3)

data 積み上げ円柱グラフのデータを示します。

*color 積み上げ円柱グラフの表示色を示します。

機 能

データによりタイミングチャートを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 表示位置 X座標 */
    int      y;          /* 表示位置 Y座標 */
    int      w;          /* 表示幅 */
    int      h;          /* 表示高さ */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      width;      /* 間隔 */
    int      ptn;        /* 太さ/線種 */
    char     *col;       /* 線色 */
    int      pc;         /* データ数 */
    Grptdt   *pt;        /* 表示データアドレス */
} Grchart;

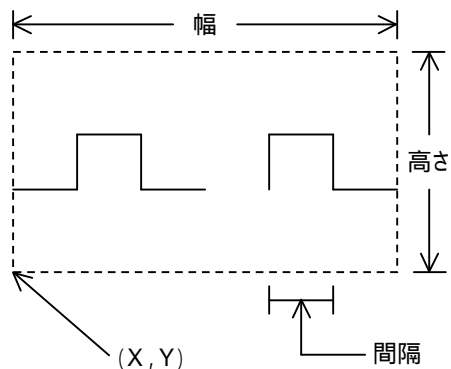
typedef struct          /* 表示データ構造体 */
{
    int      data;       /* データ */
} Grptdt;
```

メンバー説明

hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

機 能

データによりタイミングチャートを表示します。



構 造 体

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      x;          /* 原点X座標 */
    int      y;          /* 原点Y座標 */
    int      w;          /* 表示幅 */
    int      h;          /* 表示高さ */
    int      max;        /* スケール最大値 */
    int      min;        /* スケール最小値 */
    int      width;      /* 間隔 */
    int      ptn;        /* 太さ/線種 */
    char     *col;       /* 線色 */
    int      pc;         /* データ数 */
    Grptgr1  *pt;        /* 表示データアドレス */
} Grchart2;

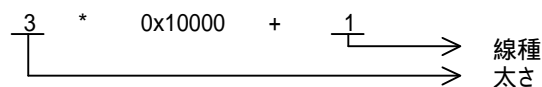
typedef struct          /* 表示データ構造体 */
{
    int      data;       /* データ */
    int      flg;        /* フラグ */
} Grptgr1;
```

メンバー説明

hed	グラフのヘッダ情報を示します。 (グラフ共通情報参照)
x y	タイミングチャートの原点を示します。
w h	タイミングチャート表示エリアの幅、高さを示します。
max min	タイミングチャートのスケールの最大値、最小値を示します。
width	各データの間隔を示します。
ptn	上位16ビット …… 線の太さを示します。 下位16ビット …… 線種を示します。 線種のモードを以下に示します。

モード	値	説 明
実線	0	—————
破線	1	—— ———
点線	2	- - - - -
一点破線	3	—— - - -
二点破線	4	—— - - -

< 例 > 太さ3ドットの破線を表示する時には次の様に指定します。



*col	タイミングチャートの線の色を示します。
pc	タイミングチャートのデータの数を示します。
*pt	タイミングチャートのデータアドレスを示します。 (実際のデータは表示データ構造体 (Grptgr1) の型を持つ配列データです。)

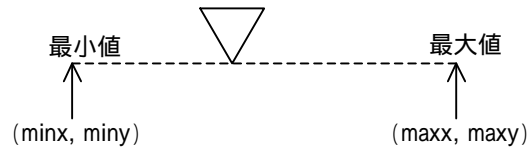
表示データ構造体 (Grptgr1)

data タイミングチャートのデータを示します。

flg 0 … 表示しません。
 1 … 表示します。
 2 … グラフ終端を示します。

機 能

データにより指定されたシンボルを表示します。

**構 造 体**

```
typedef struct
{
    Grobj    hed;        /* グラフヘッダ */
    int      type;       /* 表示タイプ */
    int      max;        /* スケール最大値 */
    int      maxx;      /* 最大値 X座標 */
    int      maxy;      /* 最大値 Y座標 */
    int      min;        /* スケール最小値 */
    int      minx;      /* 最小値 X座標 */
    int      miny;      /* 最小値 Y座標 */
    int      sym;        /* シンボル */
    Ktobj    *sym_obj;   /* ユーザ定義部品 (複合部品) */
    int      data;       /* データ */
    int      zdata;      /* ライブラリ管理データ */
} Grindicator;
```

メンバー説明

hed グラフのヘッダ情報を示します。
 (グラフ共通情報参照)

type TKgrdisplay 時の表示方法を示します。

0 …… データの位置に表示します。
 1 ~ …… 前回のデータ値から今回のデータ値まで指定された値を増加(減少)させながら表示します。

max インジケータのスケールの最大値を示します。

maxx 最大値の座標を示します。
 maxy

min インジケータのスケールの最小値を示します。

minx 最小値の座標を示します。
 miny

sym 表示するシンボルを以下に示します。

値	シンボル
0	ユーザ定義部品
1	▽
2	▼
3	△
4	▲
5	▷
6	►
7	◁
8	◄

注1) 値1~8はグラフ表示支援ライブラリが用意したデフォルトのシンボルです。
 値0のユーザ定義部品を指定した場合は必ず sym_obj を設定して下さい。

注2) 値1~8のシンボルの表示色は黒固定です。

sym_obj ユーザ定義部品指定の時複合部品を作成し、そのアドレスを示します。
 デフォルトのシンボルを指定している時は無効になります。

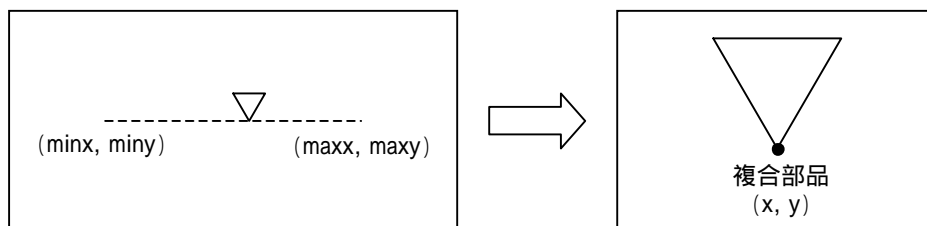
注) ユーザ定義部品は必ず複合部品を用いて作成して下さい。

data インジケータのデータを示します。

zdata ライブラリで管理する為、初期値は NULL を指定します。

注1) ユーザ定義部品を指定した時、複合部品の X, Y 座標がデータの表示位置となりますので注意してユーザ定義部品を指定して下さい。

例



注2) シンボルの内容を変更し、再表示する様な場合は、TKgrdelete、TKgrentry する必要があります。

4 サンプルプログラム

サンプルプログラム1

グラフィックを使用したプログラミング例を示します。

メインプログラム

```
#include      <TK2/TK.h>
#include      <TK2/TKusr.h>
#include      <GR.h> _____

#include      "sample.h" _____
#include      "sampledata.h" _____

Ktcore      *core;
Ktroot      *root;
Grcore      *grcore;

void      main(argc, argv)
int      argc;
char      *argv[];
{
    int      sts;

    root = (Ktroot *)TKinit();
    if( root != NULL )
    {
        core = (Ktcore *)TKopen(root,NULL,&kinf);
        if(core != NULL)
        {
            grcore = (Grcore *)TKgropen(core,KARRAY,kinf_tree); _____
            TKgrentry(grcore,KSINGL,&oresen1.hed,HRBDSPON); _____

            for(sts=0;sts!=1;)
            {
                sts = TKevent(root);
            }

            if( ( core != NULL)&&( core->stat == 1 ) )
            {
                TKgrdelete(grcore,KSINGL,&oresen1.hed,HRBDSPON); _____
                TKgrclose(grcore); _____
                TKclose(core);
                core = NULL;
            }
        }
        TKexit(root);
    }
}
```

sample.c (背景部品ファイル)

```
/* X-Mate Version 3.0.3 Copyright 1991 FUJI Data System Corp. */
/* This source file is created by X-Mate. */
/* Don't modify this. */

#include <TK2/TK.h>
#include <TK2/TKusr.h>

Ktpanel kinf =
{
    KPROC,303,349,600,400,3,ON,ON,ON,
    X-Mate User's Window,"折れ線グラフ",NULL,
    GRAY,BLACK,GRAY,XC_hand2
};
Ktrect oresen_Rect1 =
{
    KORECT,0,1,KRDWB,14,40,568,342,2,
    BACK,WHITE,BLACK
};
Ktrect oresen_Rect2 =
{
    KORECT,0,1,KRUPB,16,42,564,338,2,
    BACK,WHITE,BLACK
};
Ktrect oresen_Rect3 =
{
    KORECT,0,1,KRUPB,58,62,2,300,1,
    BACK,WHITE,BLACK
};
Ktrect oresen_Rect4 =
{
    KORECT,0,1,KRUPB,58,361,502,2,2,
    BACK,WHITE,BLACK
};
int oresen_dsp_max_dat = { 100 };
Ktidata oresen_dsp_max =
{
    KOIDATA,0,1,KIOFF,18,59,39,14,5,
    SKANA,BLACK,GRAY,WHITE,BLACK,
    KPR,KDDEC,2,(int *)&oresen_dsp_max_dat,0,
    NULL,NULL,0,NULL,0
};
int oresen_dsp_min_dat = { 0 };
Ktidata
{
    KOIDATA,0,1,KIOFF,18,346,39,14,5,
    SK oresen_dsp_min =
    KPR,KDDEC,2,(int *)&oresen_dsp_min_dat,0,
    NULL,NULL,0,NULL,0
};
```

```
Ktobj    *kinf_tree[] =  
{  
    &oresen_Rect1.hed,  
    &oresen_Rect2.hed,  
    &oresen_Rect3.hed,  
    &oresen_Rect4.hed,  
    &oresen_dsp_max.hed,  
    &oresen_dsp_min.hed,  
    NULL  
};
```

sampledata.h (クラブ部品のデータ)

```
/* 折れ線グラフ データ */
Grptdt oresen1_dat[50] =
{
    0, 50, 30, -10, 0, -70, -100, -80, 50, 20,
    60, 20, 90, -10, 100, 50, 60, 40, -20, 40,
    50, -20, 0, 80, 40, 70, -20, -40, -10, 70,
    30, 50, 30, -20, 80, 60, -30, -70, 50, 10,
    20, 60, 20, -30, -90, 0, 10, -30, 90, 0,
};
Grline oresen1 =
{
    HRLine,0,1,0, 60,360,500,300,100,-100,10,0,BLUE,50,
    (Grptdt *)oresen1_dat
};
```

説明

GR.h ファイル

グラフツールの各情報が定義されているインクルードファイルです。

"sample.h" ファイル ("XXX.h")

X-Mate で生成したグラフの背景になるパネルのヘッダ情報です。

"sampledata.h" ファイル

画面に表示するグラフの情報が定義されています。

TKgropen 処理

指定のパネルに背景部品を登録し表示を行ないます。

例では、複数登録モードの為、kinf_tree に登録している背景部品を一括登録します。

TKgrentry 処理

指定されたグラフ部品を登録し表示を行ないます。

例では、単体登録モードで oresen1 を登録しています。

TKdelete 処理

指定されたグラフ部品を削除しパネルから消去します。

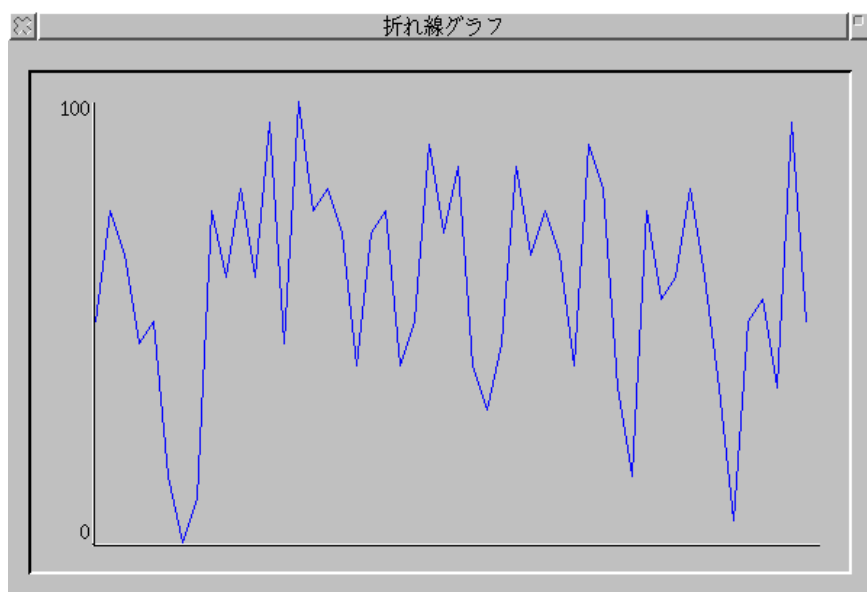
例では、単体削除モードで oresen1 を削除しています。

TKgrclose 処理

TKgropen で作成された資源を開放します。

オープン時に指定された kinf_tree に登録している背景部品を一括削除します。

画面表示例



サンプルプログラム2

PRkid を使ってグラフ部品を印刷するプログラミング例を示します。

メインプログラム

```
#include      <TK2/TK.h>
#include      <TK2/TKusr.h>
#include      <PRkid/TKPrinPS.h>
#include      <GR.h>

#include      "sample.h"
#include      "sampledata.h"

Ktcore      *core;
Ktroot      *root;
Grcore      *grcore;

void        main(argc, argv)
int         argc;
char        *argv[];
{
    int      sts,sta;

    root = (Ktroot *)TKinit();
    if( root != NULL )
    {
        core = (Ktcore *)TKopen(root,NULL,&kinf);
        if(core != NULL)
        {
            grcore = (Grcore *)TKgropen(core,KARRAY,kinf_tree);
            TKgrentry(grcore,KSINGL,&oresen1.hed,HRBDSPON);

            sta = TKpinit(NULL,KSHYOKO,60.0,60.0,0,"sample.ps");
            sta = TKpinit2(&kinf,KSPANELON,KSPNLISIZE);
            sta = TKprint(KARRAY,kinf_tree);
            sta = TKprint(KARRAY,oresen1.hed.ent);
            sta = TKpout("PostScript",KSFNODEL,1);

            for(sts=0;sts!=1;)
            {
                sts = TKevent(root);
            }
            if( ( core != NULL)&&( core->stat == 1 ) )
            {
                TKgrdelete(grcore,KSINGL,&oresen1.hed,HRBDSPON);
                TKgrclose(grcore);
                TKclose(core);
                core = NULL;
            }
        }
        TKexit(root);
    }
}
```

説 明

TKprint 処理 (背景部品)

背景部品の印刷を行ないます。

TKprint 処理 (グラフ部品)

グラフ部品の印刷を行ないます。

5 ライブラリ

グラフ表示支援ライブラリのライブラリ及びインクルードファイルは、X-Mate がインストールされたディレクトリの GRAPH に準備されています。

グラフ表示支援ライブラリ

<code>\$(XMATEHOME)/GRAPH/libTKgraph.a</code>

グラフ表示支援ライブラリ用インクルードファイル

<code>\$(XMATEHOME)/GRAPH/GR.h</code>

- ・ 本書及びプログラムは、著作権上、当社に無断で使用、複製する事は出来ません。
- ・ 本書及びプログラムの運用上のトラブルについては責任を負いかねますのでご了承願います。
- ・ 本書又は本製品の内容に御不審な点がありましたら御連絡下さい。
- ・ 本書及びプログラムは予告なしに変更する事があります。

初版発行 1993 年 8 月
第五版 2010 年 4 月

Copyright 1991 FUJI Data System Co.,Ltd.