

グラフィックスライブラリ ユーザーズガイド

2010年10月25日 第11版

MiSPO

株式会社ミスポ

はじめに

本製品はリアルタイム OS「NORTi」用のグラフィックスライブラリです。

グラフィックスライブラリには開発キットに付属する評価版と、単品で販売されている製品版の2種類があり、本書は、両方のユーザーズガイドを兼ねています。この2つの版の違いはグラフィックスクントローラドライバソースコードの有無です。評価版グラフィックスライブラリには、グラフィックスクントローラドライバがライブラリに内蔵されており、設定値が開発キットの評価ボード仕様に合わせて固定となっています。一方、製品版グラフィックスライブラリにはドライバソースコードが付属しており、ユーザーのボードに合わせてカスタマイズしてアプリケーションにリンクすることにより、グラフィックライブラリをユーザーのボードに移植することができます。

グラフィックスライブラリで使用できるフォントは以下の3種類です。

- ・日本規格協会 JIS X 9051-84 準拠「表示装置用 16 ドット字形」
- ・日本規格協会 JIS X 9052-83 準拠「ドットプリンタ用 24 ドット字形」
- ・「Web Browser for NORTi」内蔵 14 ドットフォント

日本規格協会 JIS フォントはグラフィックスライブラリに内蔵されており、ライセンスされたプロジェクトの範囲でユーザーのボードへの搭載を自由に行えます。「Web Browser for NORTi」内蔵のフォントをユーザーのボードへ搭載するためには、別途「Web Browser for NORTi」のロイヤリティが発生します。

NORTi は株式会社ミスポの登録商標です。

本書で使用するコンパイラ名、CPU 名、その他製品名は、各メーカーの商標です。

本書に記載されている内容は、予告無く変更されることがあります。

目次

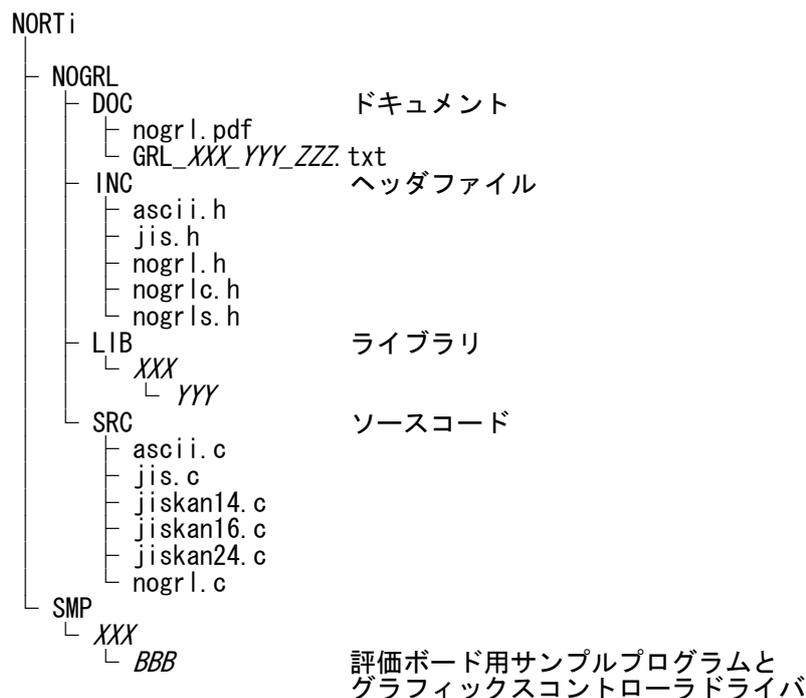
第1章 概要.....	1
1.1 フォルダおよびファイル構成.....	1
ドキュメント.....	1
ヘッダファイル.....	2
ライブラリ.....	3
ソースコード.....	4
1.2 主な仕様.....	5
座標系.....	5
画面の回転.....	6
色.....	9
線種.....	10
描画の流れ.....	10
画面管理.....	11
使用するリソース.....	12
コンフィグレーション.....	13
ダブル・バッファリング.....	14
第2章 API仕様.....	15
grl_ini.....	15
grl_ext.....	15
grl_set_pal.....	16
grl_get_pal.....	17
grl_cre_scr.....	18
grl_set_col, grl_get_col.....	19
grl_set_ptn, grl_get_ptn.....	19
grl_set_pos, grl_get_pos.....	20
grl_set_fnt, grl_get_fnt.....	21
grl_dra_dot.....	22
grl_dra_lin.....	22
grl_dra_lto.....	23
grl_dra_rec.....	23
grl_fil_rec.....	24
grl_dra_cir.....	25
grl_cpy_vrm.....	26
grl_cpy_img.....	27
grl_clr_scr.....	27

grl_dra_txt.....	28
grl_get_tid.....	29
grl_chg_pri.....	29
grl_set_buf.....	30
grl_set_vrm.....	30
grl_cpy_buf.....	31
grl_wai_end.....	32
付録	33
フォント.....	33
JIS X 9051 準拠 表示装置用 16 ドット字形データ.....	34
JIS X 9052 準拠 ドットプリンタ用 24 ドット字形データ.....	38

第 1 章 概要

1.1 フォルダおよびファイル構成

本ライブラリのフォルダ構成は、次の通りです。



上記の *XXX* は対応プロセッサ名略称 (例: SH, ARM, etc.)、*YYY* は対応コンパイラ名略称 (例: SHC7, SHC8, SHC9, etc.)、*BBB* は評価ボード名 (例: MS7720, MS7727CP, MS7760CP etc.) です。実際のフォルダ名は、DOC フォルダの補足説明書を参照してください。

ドキュメント

nogrl.pdf

グラフィックスライブラリ ユーザーズガイド (本書) です。ファイル構成、API 仕様、制限事項など、各処理系 (プロセッサやコンパイラ) に依存しない共通の事柄を説明しています。

GRL_XXX_YYY_ZZZ.txt

グラフィックスライブラリ 補足説明書です。処理系に依存する部分の説明、修正履歴を記述しています。ファイル名の *XXX*、*YYY* の部分は処理系によって異なります。*ZZZ* はバージョンです。

ヘッダファイル

`nogr1.h`

グラフィックスライブラリヘッダファイルです。グラフィックスライブラリを利用する全てのソースファイルで`#include` してください。

`nogr1c.h`

グラフィックスライブラリ コンフィグレーションヘッダファイルです。アプリケーションの1つのソースファイルでのみ`#include` してください。`#include` 文の上に、本ヘッダファイル内に定義されるマクロの値を定義することにより、コンフィグレーションを行うことができます。

`nogr1s.h`

グラフィックスライブラリ内部定義ヘッダファイルです。通常アプリケーションで`#include` する必要はありません。

`ascii.h`, `jis.h`

ASCII コード、JIS コード、シフト JIS コード関連ヘッダファイルです。各コードを相互に変換する関数を利用する場合、アプリケーションのソースファイルに`#include` してください。

ライブラリ

グラフィックスライブラリは、プロセッサ、コンパイラ、エンディアン、デバッグ情報の有無によって異なるライブラリがそれぞれ用意されています。ご使用の開発環境に必要なライブラリをアプリケーションにリンクしてください。

ファイル名の命名規則は次の通りです。

nogdXXXX.XXX デバッグ情報付きライブラリ
noglXXXX.XXX デバッグ情報なしライブラリ

XXXX.XXXの部分はプロセッサやエンディアンによって異なります。例えば、ルネサス SuperH/SHC 用のライブラリファイル名は次のようになります。

デバッグ情報付きライブラリ

nogdsh3b.lib SH-3、ビッグエンディアン
nogdsh3l.lib SH-3、リトルエンディアン
nogdsh4b.lib SH-4、ビッグエンディアン
nogdsh4l.lib SH-4、リトルエンディアン

デバッグ情報なしライブラリ

noglsh3b.lib SH-3、ビッグエンディアン
noglsh3l.lib SH-3、リトルエンディアン
noglsh4b.lib SH-4、ビッグエンディアン
noglsh4l.lib SH-4、リトルエンディアン

ソースコード

グラフィックスライブラリはライブラリ化されていますので、通常は以下のソースファイルをコンパイルしてアプリケーションにリンクする必要はありません。

`nogr1.c`

グラフィックスライブラリのソースコード

`ascii.c`

ASCII コードを JIS コード、シフト JIS コードに変換する関数

`jis.c`

JIS コードとシフト JIS コードを相互に変換関数

`jiskan14.c`, `jiskan16.c`, `jiskan24.c`

JIS フォントデータとフォント描画関数

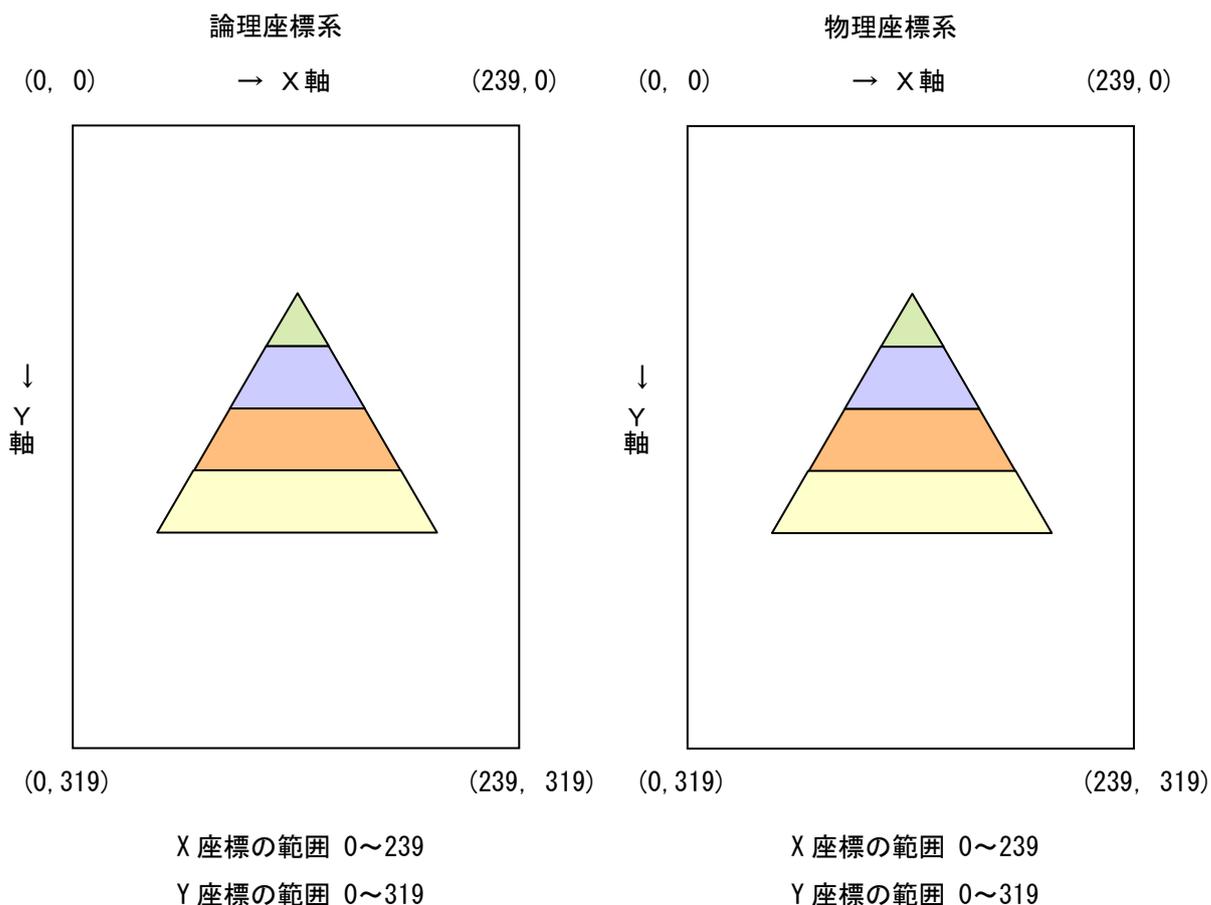
1.2 主な仕様

座標系

既定の設定では、論理座標系と物理座標系は同一です。LCD の左上を原点として右方向が X 軸正方向、下方向が Y 軸正方向となります。ここでは、論理座標系はアプリケーションがグラフィックスライブラリ API によって描画を行うときに使用する座標系を、物理座標系はグラフィックスライブラリが VRAM に描画するとき使用する座標系を意味します。

グラフィックスライブラリはその初期化処理においてグラフィックスコントローラのディスプレイデバイス情報取得関数より物理的な画面の幅と高さを取得し、その範囲内で描画を実行します。各 API の引数で座標値や幅、高さを指定する場合は、描画対象が画面の幅と高さの範囲に収まる値となるように考慮してください。

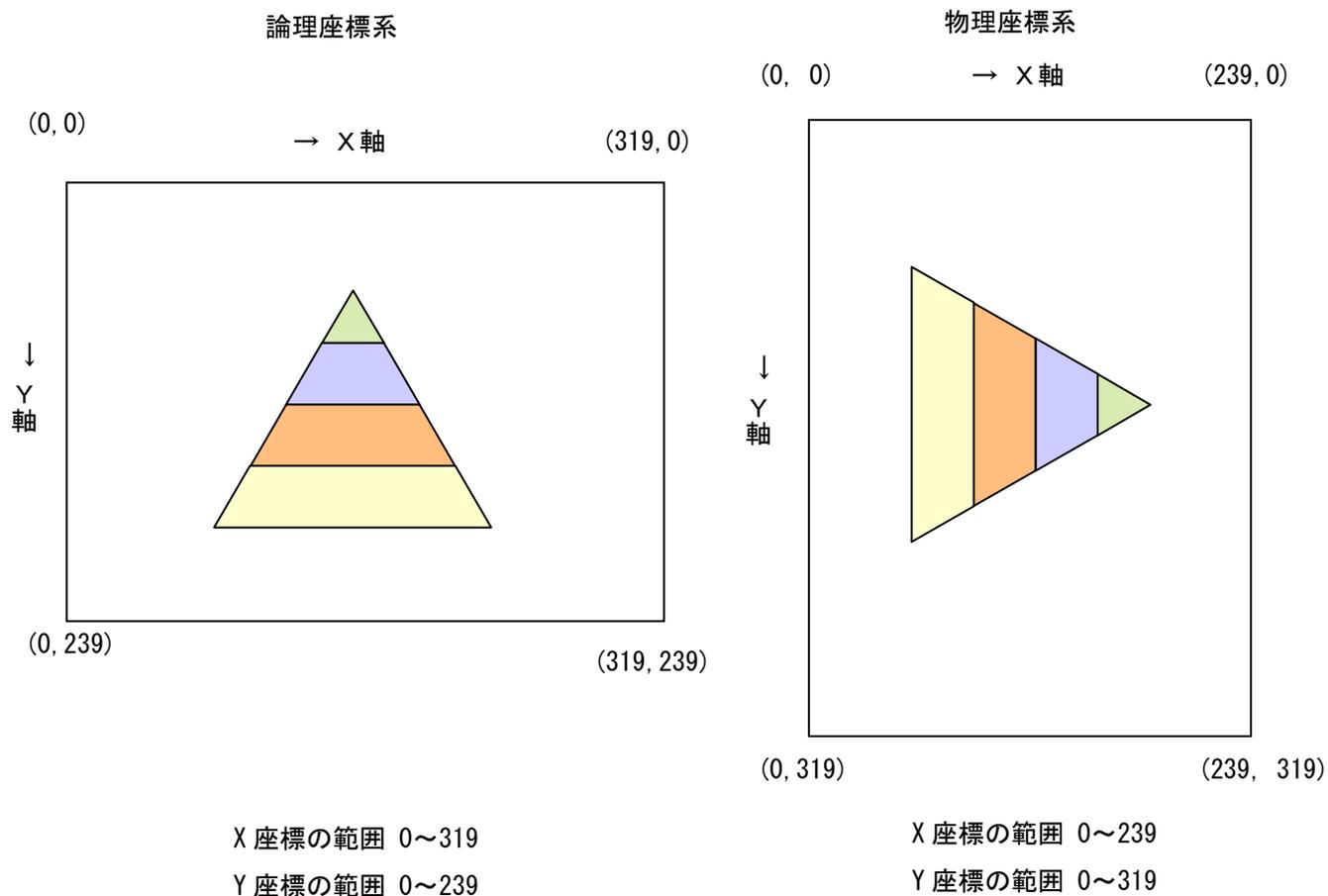
例) 縦型 QVGA LCD (240×320) の場合



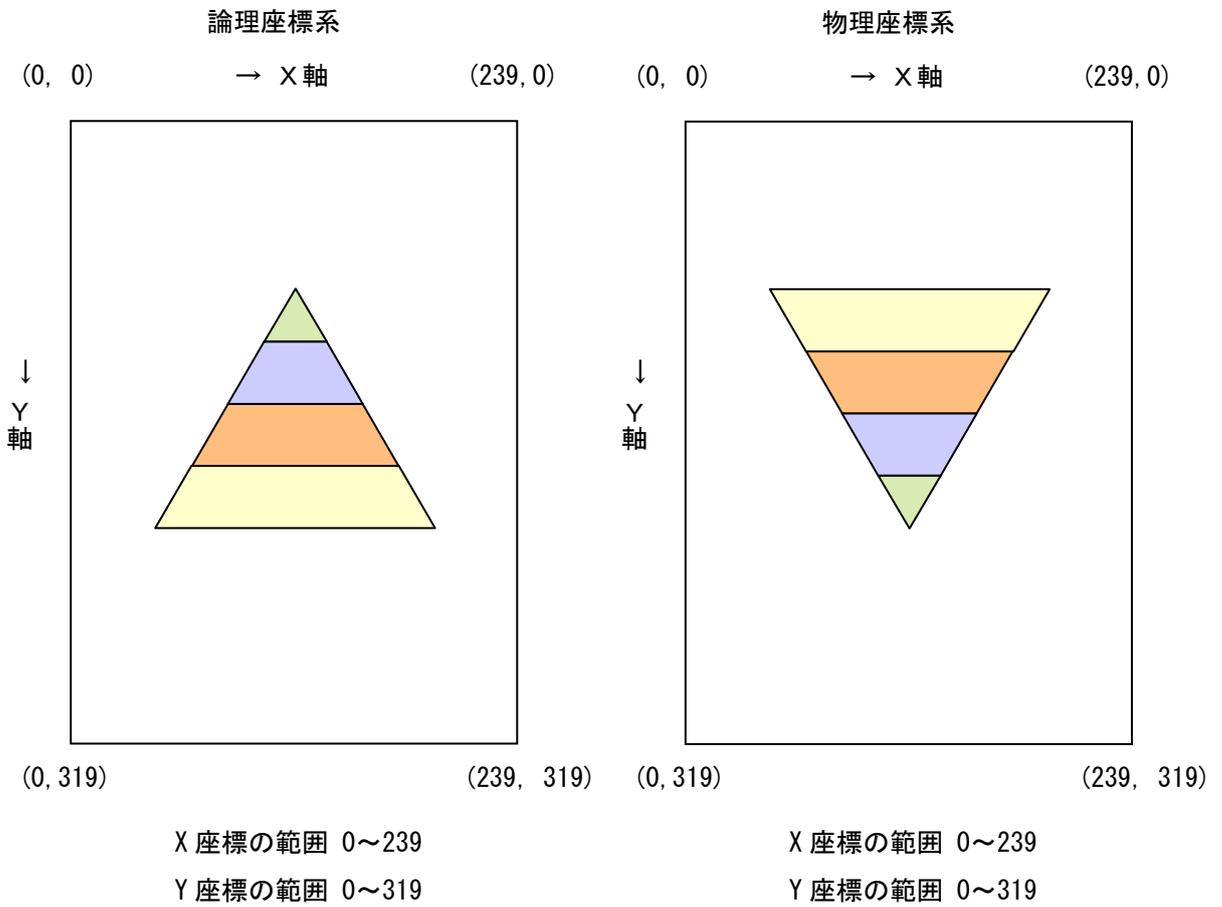
画面の回転

論理座標系を右回りに 90 度、180 度、270 度回転させて物理座標系にマッピングすることができます。
この機能により、縦長 LCD を横倒しに設置して横長 LCD として使用する事などができます。

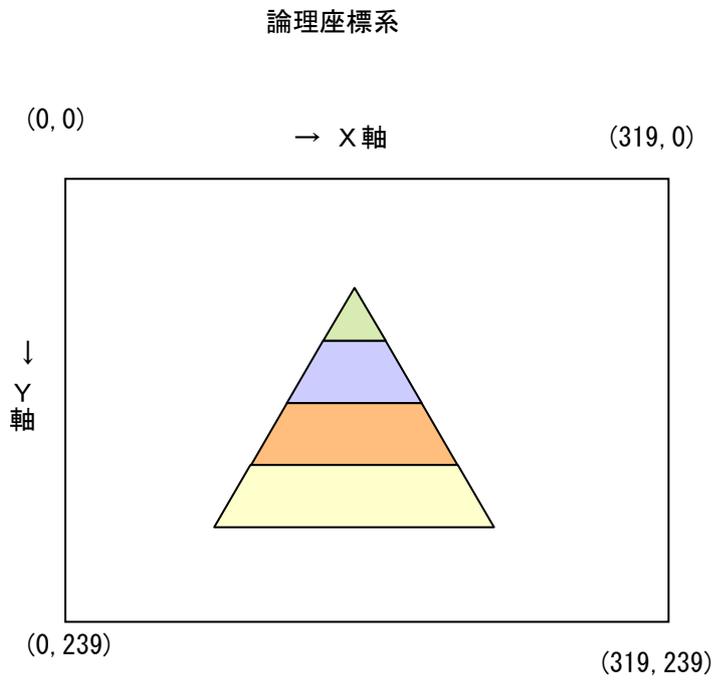
例) 90 度回転した場合



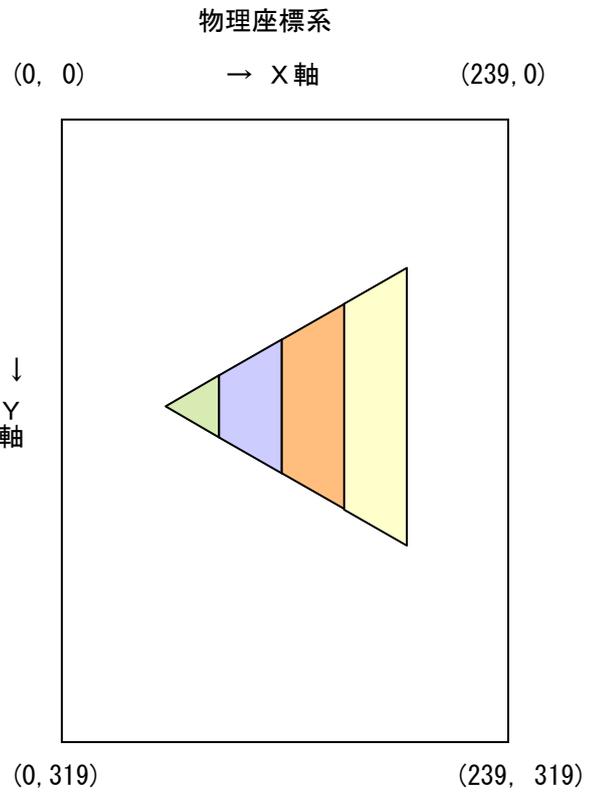
例) 180度回転した場合



例) 270 度回転した場合



X 座標の範囲 0~319
Y 座標の範囲 0~239



X 座標の範囲 0~239
Y 座標の範囲 0~319

色

16bit カラーを指定する場合は以下の RGB565 フォーマットを使用します。

MSB		LSB
RED 5bit	GREEN 6bit	BLUE 5bit

例)

色	RGB565	ビット
白	0xFFFF	11111 111111 11111
黄色	0xFFE0	11111 111111 00000
シアン	0x07FF	00000 111111 11111
緑	0x07E0	00000 111111 00000
マゼンタ	0xF81F	11111 000000 11111
赤	0xF800	11111 000000 00000
青	0x001F	00000 000000 11111
黒	0x0000	00000 000000 00000

パレットカラーを指定する場合はカラーパレットに格納されている各色のインデックス値を指定します。

例)

色	インデックス値	カラーパレットの例 (RGB888)
白	0	0xFFFFFFFF
黄色	1	0xFFFF00
シアン	2	0x00FFFF
緑	3	0x00FF00
マゼンタ	4	0xFF00FF
赤	5	0xFF0000
青	6	0x0000FF
黒	7	0x000000

線種

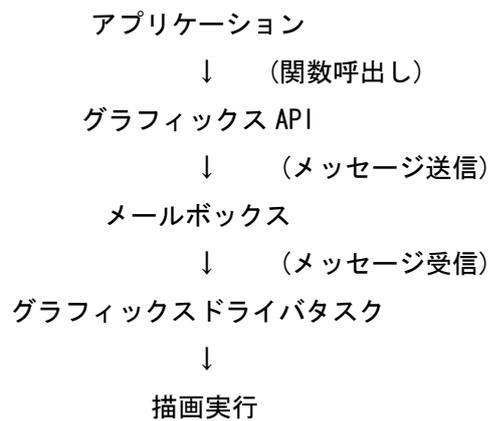
線種は 16bit のビットパターンで指定します。

例)

実線	0xFFFF	—————
点線	0xAAAA
破線	0xCCCC	- - - - -

描画の流れ

グラフィックスライブラリ内部の描画実行部分は独立したタスクとなっています。アプリケーションからコールされるグラフィックス API は、描画要求メッセージをメールボックスへ送信するだけで、直ぐにリターンします。グラフィックスドライバタスクは、メールボックスから描画要求メッセージを受信すると順次描画を実行します。



画面管理

マルチタスクに対応するため、グラフィックス API では画面 ID による管理を行っています。アプリケーションでは、最初に画面 ID を指定して画面を生成してください。生成された画面毎に画面管理ブロックが存在し、そこに描画に必要な属性が保持されます。タスク毎に異なる画面 ID を指定することによって、複数のタスクからの異なる属性による描画を可能としています。保持される属性には例えば次のようなものがあります。

ペンカラー（点・線・文字の色）

線種

ペン位置（次の線描画の始点）

画面生成後、アプリケーションでは画面 ID を指定してグラフィックス API を呼び出してください。グラフィックスライブラリは指定された ID の画面属性に基づいて描画を実行します。

例)

```
TASK Task1(void)
{
    ID scrid = 1;      /* 画面 ID */

    /* グラフィックスの初期化(1つのタスクから1回だけ実行) */
    grl_ini();

    /* 画面の生成 */
    grl_cre_scr(scrid, NULL);

    /* 座標(0,0)-(100,100)を結ぶ白い実線を描画 */
    grl_set_col(scrid, 0xffff);
    grl_set_ptn(scrid, 0xffff);
    grl_dra_lin(scrid, 0, 0, 100, 100);

    /* 座標(100,100)-(100,200)を結ぶ黄色い点線を描画 */
    grl_set_col(scrid, 0xffe0);
    grl_set_ptn(scrid, 0xaaaa);
    grl_dra_lto(scrid, 100, 200)
        :
}
}
```

使用するリソース

グラフィックスライブラリが使用する OS のリソースの数は `nogr1c.h` の中に `GRL_N...` で始まるマクロ名で定義されています。グラフィックスライブラリを使用するアプリケーションでは、以下のようにカーネルコンフィグレーションでグラフィックスライブラリの ID 使用数を加算してください。

例)

```
#include "nogr1c.h"

#define TSKID_MAX 5+GRL_NTSK /* タスク ID 使用数 */
#define SEMID_MAX 2+GRL_NSEM /* セマフォ ID 使用数 */
#define FLGID_MAX 1+GRL_NFLG /* イベントフラグ ID 使用数 */
#define MBXID_MAX 4+GRL_NMBX /* メールボックス ID 使用数 */
#define MBFID_MAX 1+GRL_NMBF /* メッセージバッファ ID 使用数 */
#define PORID_MAX 1+GRL_NPOR /* ランデブ用ポート ID 使用数 */
#define MPLID_MAX 1+GRL_NMPL /* 可変長メモリプール ID 使用数 */
#define MPFID_MAX 1+GRL_NMPF /* 固定長メモリプール ID 使用数 */
#define DTQID_MAX 1+GRL_NDTQ /* データキュー ID 使用数 */
#define MTXID_MAX 1+GRL_NMTX /* ミューテックス ID 使用数 */
#define ISRID_MAX 1+GRL_NISR /* 割り込みサブルーチン ID 使用数 */
#define CYCNO_MAX 1+GRL_NCYC /* 周期起動ハンドラ ID 使用数 */
#define ALMNO_MAX 1+GRL_NALM /* アラームハンドラ ID 使用数 */

#include "nocfg4.h"
```

※ 上記の例で、5, 2, 1, 4...はアプリケーションが使用する各リソース数の例です。

コンフィグレーション

グラフィックスドライバタスクの優先度とスタックサイズ、生成できる画面の最大数、キューイングできる描画メッセージの最大数は `nogr1c.h` に定義されています。各マクロの既定値と意味は次の通りです。

マクロ名	既定値	意味
<code>GRL_TSK_PRI</code>	0	ドライバタスク優先度※
<code>GRL_TSK_SSZ</code>	1024	ドライバタスクスタックサイズ
<code>GRL_SCRID_MAX</code>	8	生成できる画面の最大数
<code>GRL_MPF_QCNT</code>	16	キューイングできる描画メッセージの最大数

※0 はグラフィックスライブラリの初期化を行ったアプリケーションタスクと同じ優先度

アプリケーションのコンフィグレーションファイルで `#include "nogr1c.h"` の前にこれらのマクロの値を定義することにより、これらの値をユーザーがカスタマイズすることができます。

例)

```
#define GRL_TSK_PRI      4      ドライバタスクプライオリティを 4 に設定
#define GRL_SCRID_MAX   10     生成できる画面の最大数を 10 に設定

#include "nogr1c.h"
```

ダブル・バッファリング

ダブル・バッファリングとは、VRAM を 2 画面分用意して非表示の VRAM に描画を行い、描画が完了した時点で表示中の VRAM と切り替えることにより、ちらつきの無い、滑らかな表示を行う手法のことです。グラフィックスライブラリ API を使用してダブル・バッファリングを行う手順は次の通りです。

- ① `grl_set_buf` によって現在表示していない VRAM 領域を描画領域として選択します。
- ② 選択した描画領域に対して描画を行います。
- ③ 描画が完了したら `grl_set_vrm` によって描画領域を表示します。
- ④ ①～③を繰り返します。

例)

```

for (i = 0; i < 1000; i++) {
    bnum ^= 1;
    grl_set_buf(SCRID, bnum);           ①描画バッファを変更

    grl_set_col(SCRID, COLOR_BLACK);
    grl_set_fnt(SCRID, FONT_GAIJI24, 24, 24, 0);
    for (y = 0; y < DISP_H; y += 24) {
        grl_dra_txt(SCRID, 0, y, buf[bnum]);   ②前回描画した数字を黒で消去
    }

    ito0u(buf[bnum], i, DISP_W / 24);
    grl_set_col(SCRID, COLOR_RED);
    grl_set_fnt(SCRID, FONT_GAIJI24, 24, 24, 0);
    for (y = 0; y < DISP_H; y += 24) {
        grl_dra_txt(SCRID, 0, y, buf[bnum]);   ②数字を赤で描画
    }
    grl_set_vrm(SCRID, bnum);           ③LCD に表示
}

```

第 2 章 API 仕様

grl_ini

機能	グラフィックスライブラリの初期化	
形式	ER grl_ini(void)	
説明	グラフィックコントローラの初期化、オブジェクト(タスク、メモリプール、メールボックス)の生成、グラフィックドライバタスクの起動を行います。	
戻値	E_OK	正常終了
	E_OBJ	既に初期化済み
	E_TMOUT	LCD コントローラドライバの初期化失敗
	その他	オブジェクト生成失敗時の NORTi エラーコード

grl_ext

機能	グラフィックスライブラリの終了	
形式	ER grl_ext(void)	
説明	グラフィックコントローラの終了処理、オブジェクトの削除、グラフィックドライバタスクを終了と削除を行います。	
戻値	E_OK	正常終了
	E_NOEXS	グラフィックスライブラリが初期化されていない

grl_set_pal

機能 カラーパレットの設定

形式 ER grl_set_pal (VP pal, UH num)

pal パレットデータ先頭アドレス

num パレットデータに格納されているエントリ数

説明 パレットデータをグラフィックスコントローラカラーパレットに設定します。

pal にはグラフィックスコントローラが対応するカラーフォーマットのパレットエントリを格納したパレットデータの先頭アドレスを指定します。

例) RGB888 パレットデータエントリフォーマット

MSB				LSB
全て 0	R:8 ビット	G:8 ビット	B:8 ビット	

num にはパレットデータに格納されているエントリ数を指定します。

戻値 E_OK 正常終了

grl_get_pal

機能 カラーパレットの取得

形式 ER grl_get_pal (VP pal, UH num)

pal パレットデータを格納する先頭アドレス

num パレットデータを格納するエントリ数

説明 グラフィックスコントローラのカラーパレットに設定されているパレットデータを取得します。 pal にはグラフィックスコントローラが対応するカラーフォーマットのパレットエントリを格納するパレットデータの先頭アドレスを指定します。

例) RGB888 パレットデータエントリフォーマット

MSB				LSB
全て0	R:8ビット	G:8ビット	B:8ビット	

num にはパレットデータに格納するエントリ数を指定します。

戻値 E_OK 正常終了

grl_cre_scr

機能 画面の生成

形式 ER grl_cre_scr (ID scrid, T_CSCR *pk_cscr)

scrid 画面 ID

pk_cscr 画面生成情報パケットへのポインタ

説明 画面を生成します。具体的には、画面 ID で識別される画面管理ブロックに画面属性が保持され、各描画 API の実行が可能となります。

画面生成情報パケットの構造は次の通りです。

```
typedef struct t_cscr
```

```
{   ATR scratr;           画面属性
    const char *name;     画面名へのポインタ
} T_CSCR;
```

scratr には以下のいずれかを設定してください。

TSA_GRL_ROT0 画面を回転しない

TSA_GRL_ROT90 画面を右回りに 90 度回転して表示

TSA_GRL_ROT180 画面を右回りに 180 度回転して表示

TSA_GRL_ROT270 画面を右回りに 270 度回転して表示

name は現在未使用です。NULL を設定してください。

pk_cscr に NULL を設定した場合、既定の属性として TSA_GRL_ROT0 が設定されます。

その他の画面属性初期値は次の通りです。

ペンカラー 白

線種 実線

ペン位置 (0, 0)

戻値 E_OK 正常終了

E_ID 画面 ID が範囲外

E_OBJ 画面が既に生成されている

grl_set_col, grl_get_col

機能 ペンカラーの設定・取得

形式 ER grl_set_col(ID scrid, UH col)
ER grl_get_col(ID scrid, UH *col)
scrid 画面 ID
col ペンカラー

説明 scrid で指定される画面のペンカラーを設定・取得します。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR NULL ポインタを指定

grl_set_ptn, grl_get_ptn

機能 線種の設定・取得

形式 ER grl_set_ptn(ID scrid, UH ptn)
ER grl_get_ptn(ID scrid, UH *ptn)
scrid 画面 ID
ptn 線種

説明 scrid で指定される画面の線種を設定・取得します。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR NULL ポインタを指定

grl_set_pos, grl_get_pos

機能 ペン位置の設定・取得

形式 ER grl_set_pos(ID scrid, H x, H y)
ER grl_get_pos(ID scrid, H *x, H *y)
scrid 画面 ID
x ペン位置の X 座標
y ペン位置の Y 座標

説明 scrid で指定される画面のペン位置を設定・取得します。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値が範囲外または NULL ポインタを指定

grl_set_fnt, grl_get_fnt

機能 フォント種別、サイズ、属性の設定・取得

形式 ER grl_set_fnt(ID scrid, VP font, UH w, UH h, UH attr)
 ER grl_get_fnt(ID scrid, VP *font, UH *w, UH *h, UH *attr)

scrid 画面 ID
 font フォント種別
 w 1 文字の幅
 h 1 文字の高さ
 attr フォント属性

説明 scrid で指定される画面のフォント種別、サイズ、属性を設定・取得します。

font に指定できるフォント種別は次の 3 種類です。

FONT_WEB14 14×14 ドット・日本語フォント

FONT_JIS16 16×16 ドット・ゴシック体日本語フォント

FONT_JIS24 24×24 ドット・明朝体日本語フォント

FONT_WEB14 を使用する場合は、「Web Browser for NORTi」ライブラリをアプリケーションにリンクしてください。

1 文字の幅と高さ w と h に指定できるのはフォント本来のサイズとその 2 倍、または 0 です。0 を指定した場合はフォント本来のサイズに設定されます。

例) FONT_JIS16 の場合

w = 0 , h = 0 の場合 16×16 ドット

w = 16, h = 16 の場合 16×16 ドット

w = 32, h = 32 の場合 32×32 ドット

その他の場合 非表示

フォント属性は将来の拡張用です。通常 0 を設定してください。

設定されなかったフォントのデータはアプリケーションにリンクされませんので、無駄なコードサイズを節約することができます。

戻値 E_OK 正常終了
 E_ID 画面 ID が範囲外
 E_NOEXS 画面が未生成
 E_PAR font に NULL ポインタを設定

grl_dra_dot

機能 点の描画

形式 ER grl_dra_dot(ID scrid, H x, H y)

scrid 画面 ID

x X 座標

y Y 座標

説明 座標(x, y)の位置に画面のペンカラーで点を描画します。正常終了した場合、画面のペン位置は座標(x, y)となります。

戻値 E_OK 正常終了

E_ID 画面 ID が範囲外

E_NOEXS 画面が未生成

E_PAR 座標値が範囲外

grl_dra_lin

機能 線の描画

形式 ER grl_dra_lin(ID scrid, H x1, H y1, H x2, H y2)

scrid 画面 ID

x1 始点の X 座標

y1 始点の Y 座標

x2 終点の X 座標

y2 終点の Y 座標

説明 始点(x1, y1)、終点(x2, y2)を端点とする直線を、画面のペンカラー、線種で描画します。正常終了した場合、画面のペン位置は座標(x2, y2)となります。

戻値 E_OK 正常終了

E_ID 画面 ID が範囲外

E_NOEXS 画面が未生成

E_PAR 座標値が範囲外

grl_dra_lto

機能 ペン位置からの線の描画

形式 ER grl_dra_lto(ID scrid, H x, H y)

scrid 画面 ID
x 終点の X 座標
y 終点の Y 座標

説明 画面のペン位置を始点、(x, y)を終点とする直線を、画面のペンカラー、線種で描画します。正常終了した場合、画面のペン位置は座標(x, y)となります。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値が範囲外

grl_dra_rec

機能 矩形の描画

形式 ER grl_dra_rec(ID scrid, H x, H y, UH w, UH h)

scrid 画面 ID
x 左上頂点の X 座標
y 左上頂点の Y 座標
w 幅
h 高さ

説明 左上頂点の座標(x, y)、幅 w、高さ h の矩形を、画面のペンカラー、線種で描画します。画面のペン位置は変化しません。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値または幅、高さが範囲外

grl_fil_rec

機能 矩形の描画(塗りつぶし)

形式 ER grl_fil_rec(ID scrid, H x, H y, UH w, UH h, UH col)

scrid	画面 ID
x	左上頂点の X 座標
y	左上頂点の Y 座標
w	幅
h	高さ
col	塗りつぶし色

説明 左上頂点の座標(x, y)、幅 w、高さ h の矩形を、画面のペンカラー、線種で描画し、矩形内を色 col で塗りつぶします。画面のペン位置は変化しません。

戻値

E_OK	正常終了
E_ID	画面 ID が範囲外
E_NOEXS	画面が未生成
E_PAR	座標値または幅、高さが範囲外

grl_dra_cir

機能 円の描画

形式 ER grl_dra_cir (ID scrid, H x, H y, UH r)

scrid 画面 ID
x 中心点の X 座標
y 中心点の Y 座標
r 半径

説明 中心点の座標 (x, y)、半径 r の円を、画面のペンカラー、線種で描画します。画面のペン位置は変化しません。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値または半径が範囲外

grl_cpy_vrm

機能 VRAM 上のデータコピー

形式 ER grl_cpy_vrm(ID scrid, H dx, H dy, H sx, H sy, UH w, UH h)

scrid 画面 ID
dx コピー先左上頂点 X 座標
dy コピー先左上頂点 Y 座標
sx コピー元左上頂点 X 座標
sy コピー元左上頂点 Y 座標
w コピー元幅
h コピー元高さ

説明 VRAM 上で、左上座標 (sx, sy)、幅 w、高さ h の矩形の位置にあるデータを、左上座標 (dx, dy)、
の位置にコピーします。画面のペン位置は変化しません。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値または幅、高さが範囲外

grl_cpy_img

機能 画像データを VRAM 上へコピー

形式 ER grl_cpy_img(ID scrid, H dx, H dy, UH w, UH h, UH *img)

scrid 画面 ID
x コピー先左上頂点 X 座標
y コピー先左上頂点 Y 座標
w 画像データ幅
h 画像データ高さ
img 画像データ先頭アドレス

説明 幅 w、高さ h の画像データを VRAM 上の左上座標 (x, y)、幅 w、高さ h の位置にコピーします。画面のペン位置は変化しません。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値または幅、高さが範囲外または NULL ポインタを指定

grl_clr_scr

機能 画面全体の塗りつぶし

形式 ER grl_clr_scr(ID scrid, UH col)

scrid 画面 ID
col 塗りつぶし色

説明 画面全体を色 fcol で塗りつぶします。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成

grl_dra_txt

機能 文字列の描画

形式 ER grl_dra_txt(ID scrid, H x, H y, char *txt)

scrid 画面 ID
x 文字列左上 X 座標
y 文字列左上 Y 座標
txt 文字列へのポインタ

説明 txt で指定された文字列を、画面のペンカラー、フォント種別、フォントサイズで描画します。x, y は先頭文字の左上の座標を指定してください。文字列 txt は最大 256 バイト + null 文字 '¥0' で指定できます。半角 ASCII コード文字の表示は現在未対応です。アルファベットや英数字は全角文字を指定してください。文字列中に半角 ASCII コード文字が含まれる場合は、半角サイズの中抜き四角(□)を描画します。画面のペン位置は変化しません。

戻値 E_OK 正常終了
E_ID 画面 ID が範囲外
E_NOEXS 画面が未生成
E_PAR 座標値が範囲外または NULL ポインタを指定
E_OBJ 画面のフォントが未設定

grl_get_tid

機能	ドライバタスク ID の取得	
形式	ID grl_get_tid(void)	
説明	ドライバタスクの ID を返します。	
戻値	正の値	ドライバタスク ID
	0	ドライバタスクが未生成

grl_chg_pri

機能	ドライバタスクプライオリティの設定	
形式	ER grl_chg_pri(PRI tskpri)	
	tskpri	ドライバタスクプライオリティ
説明	グラフィックスライブラリドライバタスクのプライオリティを設定します。	
戻値	E_OK	正常終了
	その他	NORTi システムコール chg_pri の戻値

grl_set_buf

機能	描画を行う VRAM 領域の設定
形式	ER grl_set_buf(ID scrid, UH bnum) scrid 画面 ID bnum VRAM 番号
説明	bnum の値によって、グラフィックスライブラリが描画を行う VRAM 領域の開始アドレスを以下のように設定します。設定は scrid 以外の全ての画面 ID に対しても適用されます。 0 グラフィックスコントローラの初期化で VRAM 領域の先頭として設定したアドレス 0 以外 0 のときのアドレス+(1 ラインのバイト数×画面の高さ)のアドレス
戻値	E_OK 正常終了 E_ID 画面 ID が範囲外 E_NOEXS 画面が未生成

grl_set_vrm

機能	画面に表示する VRAM 領域の設定
形式	ER grl_set_vrm(ID scrid, UH vnum) scrid 画面 ID vnum VRAM 番号
説明	vnum の値によって、画面に表示する VRAM 領域の開始アドレスを以下のように設定します。設定は scrid 以外の全ての画面 ID に対しても適用されます。 0 グラフィックスコントローラの初期化で VRAM 領域の先頭として設定したアドレス 0 以外 0 のときのアドレス+(1 ラインのバイト数×画面の高さ)のアドレス
戻値	E_OK 正常終了 E_ID 画面 ID が範囲外 E_NOEXS 画面が未生成

grl_cpy_buf

機能 VRAM 領域間のデータコピー

形式 ER grl_cpy_buf(ID scrid, UH bnum)

scrid 画面 ID

bnum VRAM 番号

説明 bnum の値によって、VRAM 間のデータコピーを以下のように行います。

0 VRAM1 のデータを VRAM0 にコピーします。

0 以外 VRAM0 のデータを VRAM1 にコピーします。

VRAM0 と VRAM1 とは以下の VRAM 領域を指します。

VRAM0 grl_set_vrm(scrid, 0) で画面に表示される VRAM 領域

VRAM1 grl_set_vrm(scrid, 1) で画面に表示される VRAM 領域

戻値 E_OK 正常終了

E_ID 画面 ID が範囲外

E_NOEXS 画面が未生成

grl_wai_end

機能	描画の完了待ち
形式	ER grl_wai_end(void)
説明	グラフィックスドライバタスクへ起床を要求するメッセージを送り、待ち状態へ移行します。グラフィックスドライバタスクでは、先にキューイングされていた描画メッセージを処理した後で、この起床要求を実行するので、発行済みの描画 API が実際に処理されたことの待ちに使えます。
戻値	E_OK 正常終了 E_CTX 非タスクコンテキストで、または、ディスパッチ禁止状態で待ち実行 E_RLWAI 待ち状態を強制解除された(待ちの間に rel_wai を受け付け)
補足	内部では待ちに slp_tsk を使用していますので、この API を発行したタスクに対して、他から wup_tsk を発行しないようにしてください。

付録

フォント

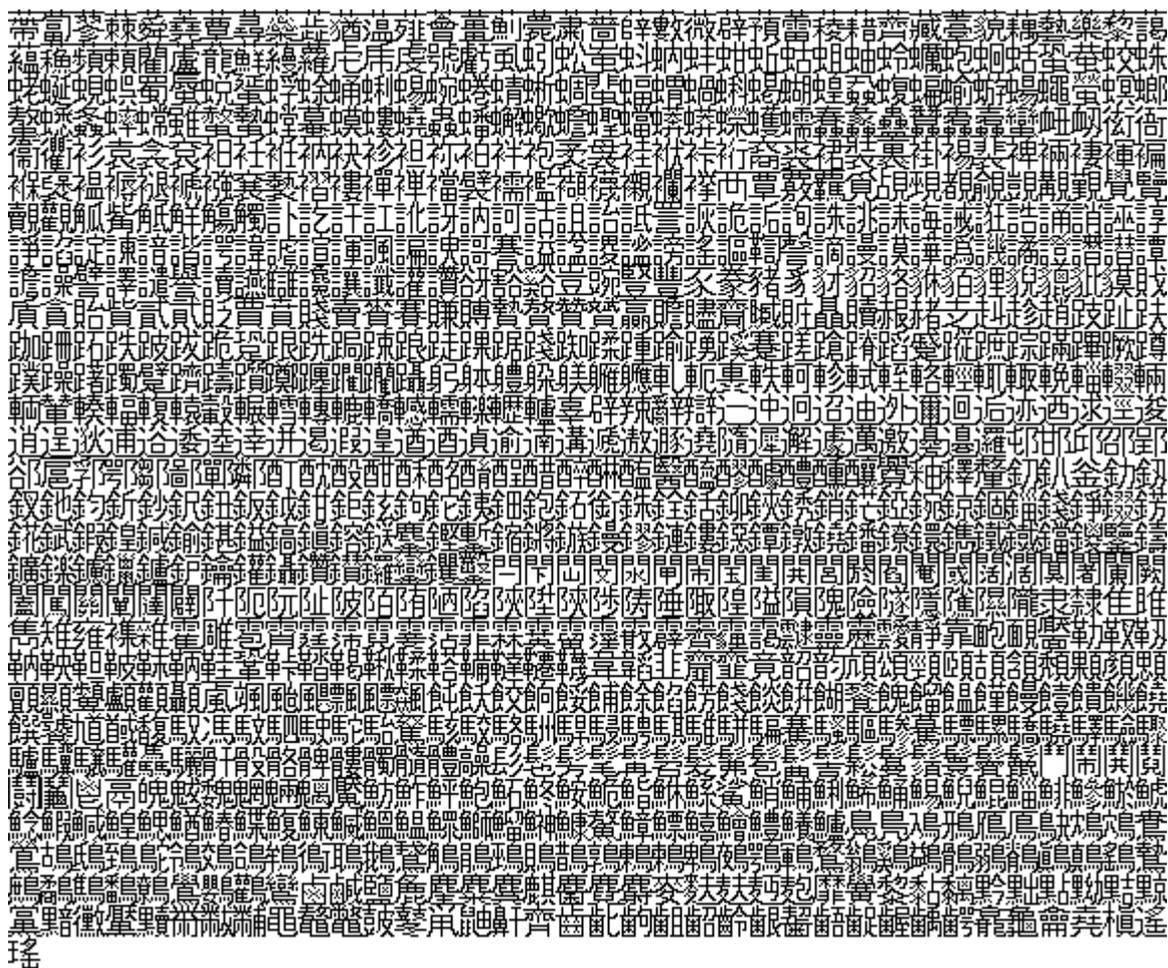
グラフィックスライブラリは以下のフォントを内蔵しています。

JIS X 9051 準拠 表示装置用 16 ドット字形データ

文字表	JIS X 0208-1983
文字数	6877 文字 (非漢字 524 文字 + 第一水準漢字 2965 文字 + 第二水準漢字 3388 文字)
文字サイズ	16×16 ドット
字体	ゴシック体

JIS X 9052 準拠 ドットプリンタ用 24 ドット字形データ

文字表	JIS X 0208-1983
文字数	6877 文字 (非漢字 524 文字 + 第一水準漢字 2965 文字 + 第二水準漢字 3388 文字)
文字サイズ	24×24 ドット
字体	明朝体



謝車遮蛇邪借勺尺杓灼爵酌积錫若寂弱惹主取守手朱殊
狩珠種腫趣邪借酒首儒受祝寿授樹綬需囚収周宗就州修愁拾
洲秀秋終繡習臭舟菟衆夙宿潤盾純巡遵醇順処初術述俊峻春瞬
戎柔汁洪獸縱旬楯殉淳準潤盾純巡遵醇順処初術述俊峻春瞬
緒署書薯諸助叙女序徐恕鋤除傷償勝捷昇昌昭晶將焦觸
獎妾娼消涉湘象賞穰蒸讓心震人仁刃塵千翠寸世瀨畝是凄請
樵沼証詔詳狀侵唇辛進針水炊睡澄摺聖脊責赤川戰踐選遷礎
訟擾条杖伸信親厨筵星隻節煎然阻窺槍槽增憎藏贈村遜怠待
辱尻新親厨筵星隻節煎然阻窺槍槽增憎藏贈村遜怠待
臣須酢趨雛据杉栖戚斥昔蟬仙線糲雙争瘦相促意態卓啄宅托
誨数枢政脆窃前煽全遡鼠槽增憎藏贈村遜怠待意態卓啄宅托
嵩成稅脆窃前煽全遡鼠槽增憎藏贈村遜怠待意態卓啄宅托
性齐設染漸訴巢騷像增憎藏贈村遜怠待意態卓啄宅托
静折洗善蘇早鎗統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
撰浅前善蘇早鎗統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
鮮素組早鎗統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
搔操鎗統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
送遭鎗統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
賊族統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
舵橈統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
逮隊橈統駟代台第醍辰奪脫巽豎迪棚谷狸鱈樽誰丹
濁諾荈探日歎淡湛炭智池知稚置致蚰遲馳築畜竹筑蓄逐秩
坦担荈探日歎淡湛炭智池知稚置致蚰遲馳築畜竹筑蓄逐秩
男談着中仲宙忠抽昼柱徵徵挑暢朝潮牒町眺聽脹腸蝶調
茶嫡着中仲宙忠抽昼柱徵徵挑暢朝潮牒町眺聽脹腸蝶調
兆凋喋籠帖帳疔弔張周彫綴鏗椿潰坪壺孀紬瓜吊釣鶴亭低
謀超跳鈔長頂鳥勅拂直朕沈珍賃鎮陳津墜椎槌追鎚痛通
塚姆捆規佃漬柘土鳶綴鏗椿潰坪壺孀紬瓜吊釣鶴亭低
偵刺貞呈堤定帝底庭廷弟悌抵挺提梯汀碇禎程締艇訂諦
蹄逋邸鄭釘鼎泥摘擢敵滴的笛適鏞弱哲徹撤轍迭鉄典填
天展店添纏甜貼轉顛点伝殿灑田電兎吐堵塗妬屠徒斗杜
渡登菟賭途都鍍砥砺努度土奴怒倒党冬凍刀唐塔塘套宕

島嶋悼投搭東桃棣棟盜淘湯涛灯燈當痘禱等答筒糖統到
 董蕩藤討膳豆踏逃透鑽陶頭騰鬪働動同堂導撞撞洞瞳童
 峒萄道銅峙鵠屯停敦沌豚遁頓吞雲鈍奈那内乍風雍謎灘
 舌寅西靜繩驟南楠軟難汝二尼式念捻撚燃粘乃晒之禁囊
 菲任妊忍認濡禰祢寧葱猫熱年念捻撚燃粘乃晒之禁囊
 濃納能腦臆牌背肺輩配倍培媒梅煤煤須買壳賠陪這蠅秤矧
 排敗杯盃博拍柏泊白涪柏泅泅版犯班畔繁股藩販範采煩煩
 萩伯剥博拍柏泊白涪柏泅泅版犯班畔繁股藩販範采煩煩
 肇善廬幡肌焄晶八鉢泅泅版犯班畔繁股藩販範采煩煩
 判半反叛蕃避非飛樋餓品彬斌瀕貧寶譜負賦赴阜附侮
 晚番肥被誹肘弼必畢筆逼蒜蛙斧普路奮粉糞紛霧篔簹暮
 罷彦膝麥肘弼必畢筆逼蒜蛙斧普路奮粉糞紛霧篔簹暮
 豹廟描富富布府怖扶敷葦路奮粉糞紛霧篔簹暮
 夫撫武舞葡燕部封楓風拚焚奮粉糞紛霧篔簹暮
 仏物鮒閉陛保鋪鋪圍捕步傍剖坊妨帽忘忙房暴望某棒冒紡
 柄並蔽閉陛保鋪鋪圍捕步傍剖坊妨帽忘忙房暴望某棒冒紡
 勉婉弁鞭奉寶鋒飽鳳鵬乏亡僕卜墨撲朴牧睦穆鉤勃沒殆堀
 包呆報豐邦鋒飽鳳鵬乏亡僕卜墨撲朴牧睦穆鉤勃沒殆堀
 褒訪謀貌貿盆摩磨魔麻埋妹味枚每哩禳娘冥名命明盟諸木
 肪膨翻凡盆摩磨魔麻埋妹味枚每哩禳娘冥名命明盟諸木
 奔本抹末沫迄僂爾磨魔麻埋妹味枚每哩禳娘冥名命明盟諸木
 俣又抹末沫迄僂爾磨魔麻埋妹味枚每哩禳娘冥名命明盟諸木
 姪北滅免棉綿緬面麵摸模茂妄孟毛猛盲網耗蒙儲木默約藥
 空勿餅尤戾初昔問悶紋門匆也治佑優勇耶野宥幽憂預傭幼
 訛躍靖柳莖由祐裕誘遊邑郵雄融夕子余蓉酪亂卵嵐欄劉瞭稜
 湧涌猶猷攏曜楊樣洋浴裸來菜賴雷離亮僚兩凌寮料梁涼獺
 庸揚搖攏曜楊樣洋浴裸來菜賴雷離亮僚兩凌寮料梁涼獺
 欲沃浴豈翼淀羅螺裸來菜賴雷離亮僚兩凌寮料梁涼獺
 覽利吏履李龍侶慮旅虞了亮僚兩凌寮料梁涼獺
 留硫粒隆童龍侶慮旅虞了亮僚兩凌寮料梁涼獺
 良諒遼量陵領力綠倫厘林淋淋臨臨麟麟麟麟麟麟麟麟麟麟
 令伶例洽勵嶺伶玲玲玲玲玲玲玲玲玲玲玲玲玲玲玲玲玲玲

グラフィックスライブラリ ユーザーズガイド

株式会社ミスポ	http://www.mispo.co.jp/
一般のお問い合わせ	sales@mispo.co.jp
技術サポートご依頼	norti@mispo.co.jp
