

組込み向けGUIツールキット

+G



GUIアプリを思いのままに



PlusG は組込み Linux 上で動作するグラフィカルなアプリケーションを作成するライブラリとツールのキットです。

ライブラリはコアエンジンに「DirectFB」を使用しています。「DirectFB」は CPU パワーを必要とする X-Window システムなどを使わずに直接フレームバッファに描画するライブラリで、CPU パワーの低い組込み機器でもストレスなく動作させることができます。

ウィンドウシステムやボタン、画像、タイマー機能などを備えたイベントドリブン型 GUI プログラムを簡単に作成することが可能です。また、ボタンやエディットボックスなどのオブジェクトはユーザオリジナル画像 (png) から生成しますのでデザインも思いのままです。

ライブラリの他、開発ツールとして、プロジェクト管理機能付き高機能エディタ (統合開発環境 SWEET)、ダイアログエディタ、カラーパレットエディタ (減色、共通パレット) が付属しています。

PlusG と PlusG が必要とするライブラリは全てソースコードが付いていますので Linux が動く環境であればアーキテクチャに依存することなくいろいろな環境で使用可能です。

特徴

Windows 上での開発・実行が可能

コンパイルや動作確認など開発は全て Windows 上で行うことが出来ます。

LinuxOS やビルド環境を Windows の仮想 PC ソフトウェア上で行うことにより、全ての開発を Windows パソコン 1 台で行うことができます。ですから、ターゲット機 (試作機) の完成を待つことなく、アプリケーションの先行開発も可能です。また、ターゲット機が 1 台でも複数のメンバで開発できます。

ダイアログなどのデザイン、プログラミング作業、その他、付属のツールは全て Windows アプリケーションです。

※「仮想 PC ソフトウェア」は VirtualBox を採用しています。

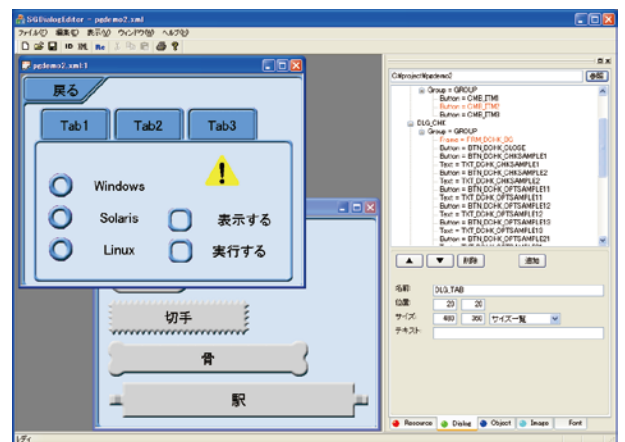
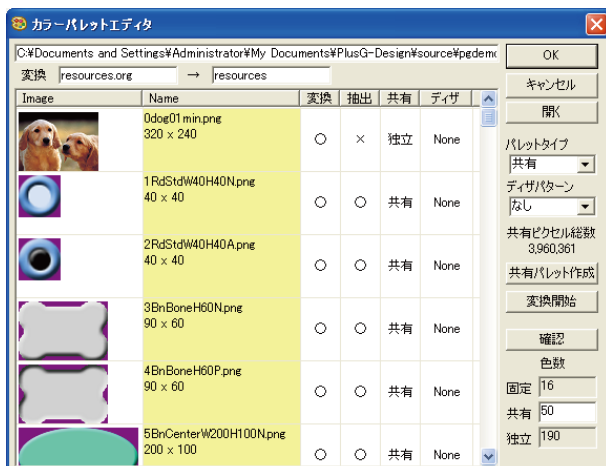
※VirtualBox とは Intel x86 の仮想化ソフトウェアで、サンマイクロシステムズ社が提供するソフトウェアです。

※開発環境をインストールした VirtualBox 用の仮想ハードディスクファイルを提供しますので WindowsPC があればすぐに開発が開始できます。

自由なデザイン

ウィンドウフレーム、ボタン、エディットボックスなどのオブジェクトはユーザが作成したオリジナル画像から生成されますので、アプリケーションの全てのデザインが自由に設計できます。作成したオブジェクトは付属のダイアログエディタでデザインできますのでマウス操作で簡単にダイアログが作成できます。

また、オブジェクト画像は実行時にロードするので、アプリケーションのスキンをいくつか用意し動的に切り替えることも簡単に行えます。



C++ ソースコード自動生成

付属のダイアログエディタは、作成したオブジェクトを配置しダイアログをデザインするツールです。

このソフトウェアはオブジェクトのレイアウト情報を元に C++ のソースコードを生成します。生成されたソースコードには、ボタンのクリック、タイマーの発生などイベント処理用の空関数が多数自動作成されますので、このイベント関数に、アクション、ページ遷移などを書き加えコンパイルすれば GUI アプリケーションの完成です。

マルチアーキテクチャ

PlusG はソースコード提供ですので、Linux が動作する環境とフレームバッファがあれば動作します。パソコン版 (x86)、SuperH、ARM 等アーキテクチャを選びません。

256色サポート

PlusG ライブラリは 32・24・16Bit/Pixel のカラーに対応していますが、CPU パワーの低い組込み機器で動作することを前提に開発されているので、8Bit256 色のインデックスカラー表示しかサポートしていない機器にも対応しています。

付属のカラーパレット作成ツールでフルカラーソースの高品質な減色、共通パレットの作成など、市販のグラフィックエディタでは難しいパレット作成が簡単にできます。

このエディタは、減色による画像の劣化を最小限に抑える為の「ディザ」や「誤差拡散」などの機能も備えています。

※256色 (8ビット / ピクセル) 環境で 1 画面に表示する画像 (ボタンやフレーム、写真など) は全て同一のパレットを使用する必要があります。したがって全ての画像を高品質に表示するには、256 のパレットの一部を共通領域、一部を独立領域とするなどの工夫が必要です。

機能説明

メッセージ（イベント通知）

PlusG を使って作成するアプリケーションはメッセージドリブン型です。PlusG のダイアログクラスにはいくつかのメッセージ処理用のコールバック関数(Virtual)が用意されています。オブジェクトやデバイス、タイマーなどからイベントを受け取ると、それらの関数をコールバックします。例えば、代表的なイベント通知関数に次のものがあります。

オープン時に一度だけ呼ばれる（初期化用）関数
virtual void PostInitial(SGHandle handledlg,int id);

表示の更新が必要な時に呼ばれる関数
virtual void Draw(SGRect region);

ボタンオブジェクトがクリックされ時に呼ばれる関数
virtual void ClickButton(SGHandle handleobj,int act);

設定したインターバルタイマー イベントが発生した時呼ばれる関数
virtual void Timer(TimeVal time);

ダイアログが閉じた時に呼ばれる関数
virtual void Closed(SGHandle handledlg,int id,int param);

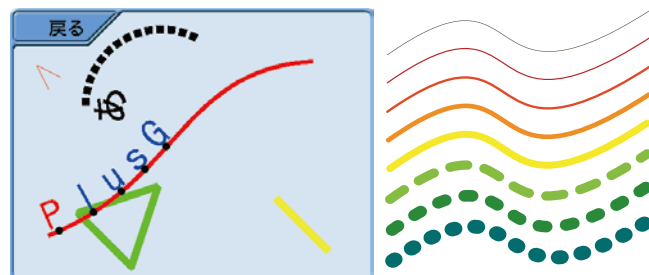
※本解説では「メッセージドリブン」という表現を採用していますが、一般的には「イベントドリブン」「イベント駆動」という呼び方の方が多く使われているようです。

2次元ベクトルデータの描画

キャンバスオブジェクトは2次元ベクトルデータの描画機能を備えています。

2次元ベクトルデータを基本とした有名なファイルフォーマットに PostScript や PDF がありますが、キャンバスオブジェクトの描画機能はこの PostScript や PDF と同じような機能をサポートしています。線は直線、ベジェ曲線、円弧、属性は実線、破線、太さ、色など、演算は縮小、拡大、回転、移動と豊富な機能を備えています。

また TrueType フォントなどアウトラインフォントにも対応しています。※描画エンジンとして Cairo を使用しています。



パレット管理方法

8Bit/Pixel のフレームバッファを使用する際は256色パレットを使用します。

1つのデバイスには1つのパレットしか使用することができませんので、画面上に表示される全てのオブジェクトは同一のパレットを使用する必要があります。

PlusG では256あるパレットを特色、共通色（抽出）、固有色と3つのグループに分け、使用するリソース画像から使用される全ての色を抽出し、使用頻度の高い重要な色を抽出し、自動的にパレットを作成する機能があります。

この機能により256色のフレームバッファで高品質に画像（写真）を表示することが可能になっています。



オリジナル画像



190色に減色



オブジェクトと写真が同時に表示可能

256色パレット	特色	基本色など16色
	共通	全てのオブジェクトから抽出した50色
	固有	写真など固有の画像などから抽出した190色
	※表示する写真によって自動的に入れ替わる領域	

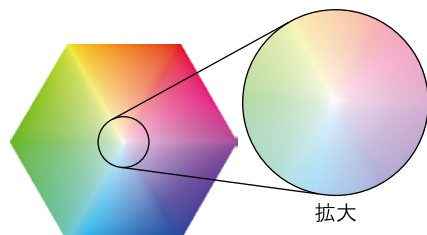
ディザリング

少ない色数で多くの色を表現する手法としてディザリングという方法があります。

写真画像には少なくとも数千、場合によっては数百万の色が使われています。減色したパレットで写真画像を表示する場合そのまま表示すると著しく劣化する場合があります。

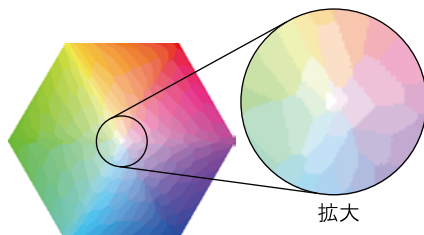
このような場合にディザリングを行うと擬似的に表現色数が増え品質が向上します。

PlusG に付属のパレットエディタは高品質な減色機能に加え、ディザリング機能も備えています。



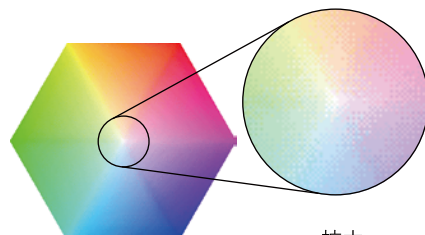
オリジナル画像

拡大



190色に減色

拡大



190色に減色+ディザ加工

拡大

フォントについて

PlusG ライブラリは TrueType や Type1 などのアウトラインフォントをサポートしています。

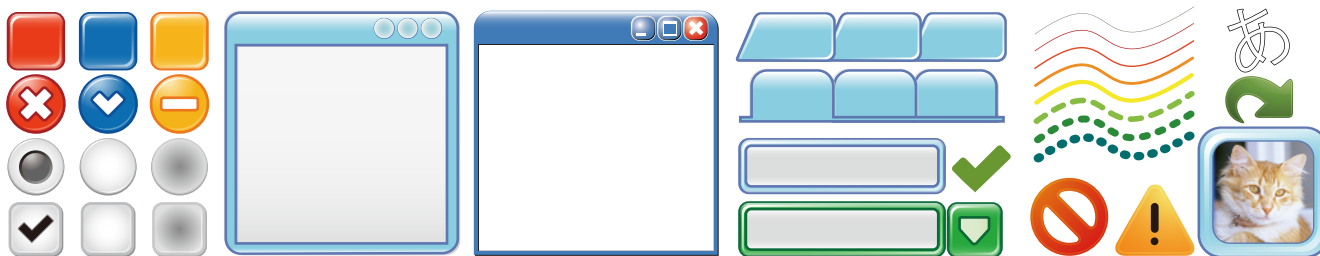
※FreeType2 フォントライブラリを使用しています。

文字コードについて

PlusG ライブラリ内部で使用する文字コードは UTF8 です。

ソースファイルなどは UTF8 を使用しています。

オブジェクトの種類



ボタン

押しボタン、ラジオボタン、チェックボックス、タブページなどの動作を行います。

フレーム

ウィンドウやダイアログの枠、タブページの枠、リストやメニュー表示の枠などの表示を行います。

アイコン・ピクチャー

アイコンとピクチャーは非常によく似たオブジェクトです。
イラスト画像や写真画像の表示を行います。
ピクチャーはパレットの管理が可能です。

キーボード

キーボードや外部スイッチなどが押されたことを通知するオブジェクトです。

エディットボックス

文字列の編集（挿入、削除、キャレット移動など）、管理を行います。

テキスト

文字列を表示します。書体、サイズ、色、寄せ、マージンなどの属性を持っています。

キャンバス

直線、曲線、文字、塗りつぶしなど Draw 系のデータを処理します。
ベクトルデータを表示することが可能です。

グループ

複数のオブジェクトをグループ管理するオブジェクトです。
グループ管理することで、一括に移動や消去などの属性変更などができます。

構成

開発は仮想 PC の中で行います。

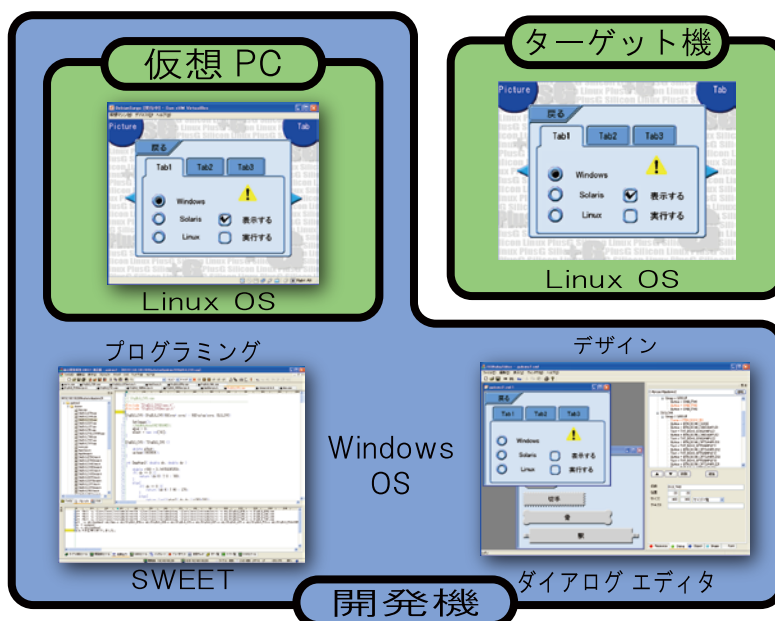
プログラミングやデザインなどの各エディタは Windows アプリケーションですが、編集するファイルは仮想 PC の内部にあります。

仮想 PC は生成・編集されたソースコードのビルドを行い、仮想 PC 上で実行、動作検証を行います。また、(同一ソースを)クロスビルドすることでターゲット機で動作するバイナリの生成も行います。

SWEET エディタには、仮想 PC でのビルド・実行とターゲット機でのビルド・実行の切り替えをボタンひとつで切り替える機能がっています。

Windows 開発機、Linux 仮想 PC、Linux ターゲット機はそれぞれ LAN でつながっています。

※Telnet、NFS、Samba などを使用します。



開発手順

1. リソースイメージの作成

Photoshop など画像編集ソフトを使ってボタンなどの部品のデザインを行います。各部品は、状態の違いにより数パターンの画像が必要です。例えば、ボタンの場合は、通常、押下、選択、押下+選択、無効、フォーカスと6種類登録可能で必要に応じて作成します。また、ボタンとエディットボックスは、高さの伸長ができませんので、高さの変化に応じた画像を用意します。

2. リソースイメージの登録と配置（デザイン）

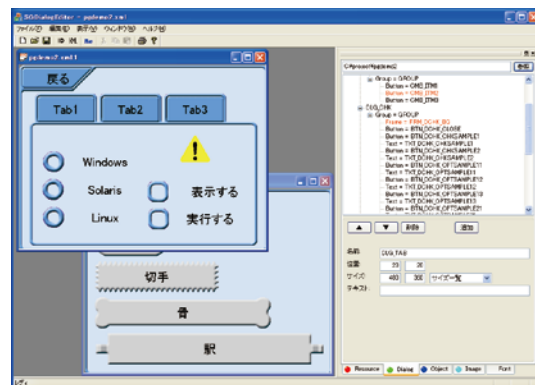
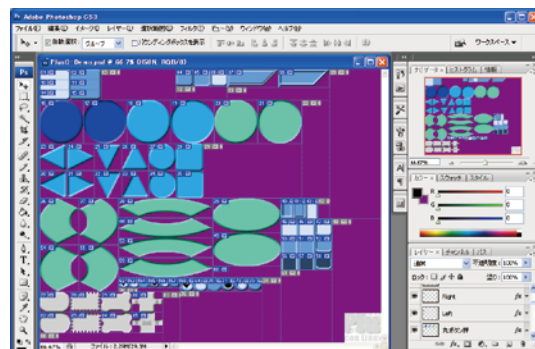
PlusG 付属のダイアログエディタを使い、作成した画像をオブジェクトとして登録します。

登録したオブジェクトを使用しダイアログを作成します。位置、サイズはマウス操作で簡単に編集できます。設計は必要な全てのダイアログの数だけ行います。

3. スケルトン ソースコードの出力

ダイアログエディタを使い、登録オブジェクトや、作成したダイアログのレイアウト情報を元に C++ のソースコードを生成します。

生成されたソースコードには、ボタンのクリック、タイマーの発生などイベント処理用の空関数が多数自動作成されます。



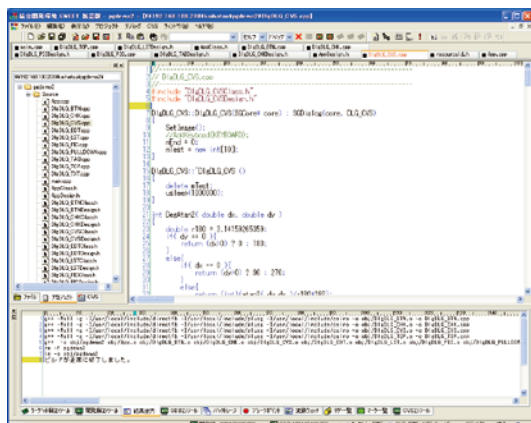
4. プログラミング

PlusG 付属のソースコードエディタを使いコーディングを行います。

SWEET は当社が販売するプロジェクト管理機能付きテキスト エディタです。

SWEET は複数ソースファイルのプロジェクトとしての管理機能や、Makefile の自動作成、Telnet、ソフトウェアデバッガなど、組込みアプリケーション作成に必要な機能が全て揃っています。

自動生成されたソースコードのイベント処理用空関数に必要なコードを記述します。

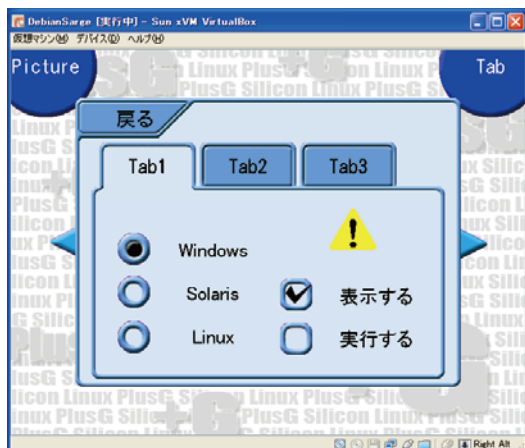


5. セルフビルド

通常のアプリケーション開発同様、ビルド、テスト、デバッグを行います。

SWEET は Makefile の作成、GCC コンパイラの実行、GDB を使ったステップ実行などが出来ます。

④、⑤を繰り返し行いダイアログなどのユーザインターフェース部分を完成させます。



6. クロスビルド（ターゲット機による実行）

外部からの信号の取込や、速度など実機でしか行えない部分のコーディングと検証を行います。

ライセンス

PlusG のライセンスは、開発プロジェクト単位になっており、開発者の人数、使用するマシン数に制限はありません。もちろんランタイムライセンスも不要ですので出荷台数にも影響しません。

ソースコード開示について

PlusG が必要とする全てのライブラリは非 GPL ライセンスですので、出荷の際にお客様の作成したアプリケーションのソースコードを開示する必要はありません。

ソフトウェア名	ライセンス
cairo	LGPL 1.2 or MPL 1.1
directfb	LGPL
expat	フリー
freetype	フリー
fontconfig	フリー
libstdc++6	例外付 GPL
glibc	LGPL
libpng	フリー
zlib	フリー

安心サポート

初期導入サポートとして導入時にオンサイトで開発環境の構築、アプリケーション開発手順、ライブラリについての解説などを行います。導入後、開発期間サポートとして、関数の使い方はもちろん、お客様のアプリケーション機能に応じたサンプルプログラムの作成や、お客様のターゲットボード用の開発環境の構築に関するサポートや Linux に関する一般的な質問などにも対応致します。

クラス一覧

SGApplication	アプリケーションの基本クラス
SGGraphics	PlusG の中心的クラス (主に DirectFB へのアクセス)
SGCore	PlusG クラス間通信を提供
SGImage	ウィジェットの画像を保持する基本
SGButtonImage	SGButtonObject にボタン画像を提供
SGEditboxImage	SGEditboxObject にエディットボックス画像を提供
SGFrameImage	SGFrameObject にフレーム画像を提供
SGImageList	SGImage の集合クラス
SGObject	ウィジェットオブジェクトの基底
SGButtonObject	ボタンウィジェットを提供
SGTextObject	テキストウィジェットを提供
SGEditboxObject	エディットボックスウィジェットを提供
SGFrameObject	フレームウィジェットを提供
SGIconObject	アイコンウィジェットを提供
SGPictureObject	ピクチャーウィジェットを提供
SGKeyboardObject	キーボードインターフェースを提供
SGCanvasObject	キャンバスウィジェットを提供
SGGroupObject	オブジェクトのグループを提供
SGCanvasDraw	ドロー系の図形描画

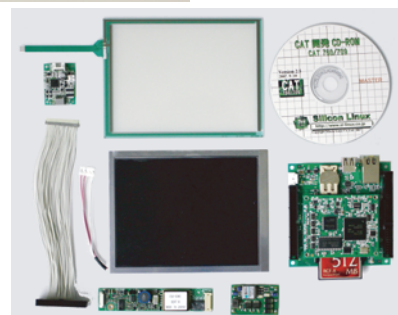
受託開発

アプリケーションの開発、外部 H/W との連携、通信などの開発、Linux デバイスドライバの作成、Linux の移植など、一部または全部をお手伝いさせて頂くことも可能です。

導入キット

SH4 搭載 Linux ボード、TFT 液晶、タッチパネル、ハーネス、電源、を専用ケースに収めた「PlusG 評価用 液晶キット」が用意されています。

キットにはソースコード付きサンプルプログラムがインストール済みです。サンプルを参考に導入からすぐに開発が始められます。



SGEventData	イベント情報を保持
SGFont	フォントイメージ属性を保持
SGFontData	フォント属性 (種類、サイズ) を保持
SGFontList	SGFontData の集合クラス
SGSharedSurface	描画イメージの共有を提供
SGSurface	描画イメージを保持
SGDialog	ダイアログを提供
SGCaret	キャレット
SGDisplayLayer	画面インターフェースを提供
SGImageProvider	オブジェクトに画像を提供
SGPalette	カラーパレットを保持
SGTextAttribute	文字列属性を保持
FdData	デバイスのファイルディスクリプタを保持
KeySymbolName	キー定数とキー文字の対応を保持する構造体
SGRect	矩形の頂点情報を保持
SGSound	イベントサウンドを提供
SGWindow	ウィンドウ機能を提供
SGWindowDataList	SGDialog の集合クラス