|  |  |
| --- | --- |
| AN204388 | |
| FM4S6E2DH（GAT-D）入門   |  | | --- | | 関連部品ファミリー：S6E2DHシリーズ | | |
| このガイドの目的は、サイプレスによって指定されたS6E2DHのグラフィック解像度の簡単な紹介を提供することです。 |

目次

[1 序文 1](#_Toc492390575)

[1.1 このガイドについて 1](#_Toc492390576)

[2 概要概要 1](#_Toc492390577)

[3 GAT-Dツールの段階的な利用 2](#_Toc492390578)

[3.1 プロジェクトの新規作成 3](#_Toc492390579)

[3.2 画像のインポート 4](#_Toc492390580)

[3.3 ページの作成 6](#_Toc492390581)

[3.4 フローデザインテーブルの作成 17](#_Toc492390582)

[3.5 ソース生成 21](#_Toc492390583)

[3.6 生成されたソースをIARプロジェクトに追加する 22](#_Toc492390584)

[3.7 ビルド後にダウンロードして実行 24](#_Toc492390585)

[4 GAT-Dの適用 26](#_Toc492390586)

[4.1 イベント管理 26](#_Toc492390587)

[4.2 ユーザーのソースコード 27](#_Toc492390588)

[4.3 ディスプレイシミュレーション 29](#_Toc492390589)

[4.4 スプライト特殊効果 30](#_Toc492390590)

[4.5 タッチパネルの解像度 31](#_Toc492390591)

[4.6 ソース生成オプション 32](#_Toc492390592)

[5 GAT-D API 34](#_Toc492390593)

[5.1 インターフェイスの説明 35](#_Toc492390594)

[5.2 ソースに参加する 37](#_Toc492390595)

[6 プログラム 38](#_Toc492390596)

[6.1 デバッグモード 38](#_Toc492390597)

[6.2 製品モード 39](#_Toc492390598)

[7 サンプルプロジェクトの説明 39](#_Toc492390599)

[7.1 建設 39](#_Toc492390600)

[7.2 機能概要 41](#_Toc492390601)

[改訂履歴 42](#_Toc492390602)

# 序文

## このガイドについて

このガイドの目的は、サイプレスによって指定されたS6E2DHのグラフィック解像度の簡単な紹介を提供することです。このガイドの説明：

1. S6E2DH（GAT-D）用のグラフィックオーサリングツールの操作方法。
2. GAT-Dの高度なアプリケーション。
3. 生成されたソースとグラフィックスドライバーを組み込みソフトウェアに統合する方法。
4. 内部フラッシュと外部フラッシュにプログラムします。
5. サンプルの構築。
6. このガイドは、IAR SystemsABのEWARMを使用した例で説明されています。他のIDEについては、適切に参照してください。

# 概要概要

次の図は、各コンポーネントの正確な関係を示しています。

図1。概要概要

PC環境

GAT-Dツール

画像

.Bmp

.Jpeg

.Png

.Gif

.etc.

画像

.Bmp

.Jpeg

.Png

.Gif

.etc.

プロジェクトの作成

画像変換

ディスプレイ構成

ディスプレイシミュレーション

ソース生成

IAR EW IDE

生成されたソース

GAT-D API.a

.a

GAT-DAPIライブラリ

グラフィックドライバー

basic\_graphics.a

util.a

その他の関数ソース

ユーザー編集

フラッシュイメージ（.Mhx）

組み込みデバイス

外部フラッシュ

*プログラム*

S6E2DH内部フラッシュ

*ビルド/プログラム*

GAT-DツールとGAT-DAPIライブラリは、S6E2DH-グラフィック解像度-Ver1.0.zipにあります。

GAT-Dツール：S6E2DH-グラフィック解像度-Ver1.0 \ 01。GAT-D Tool\01. Installer \ GAT-D\_xxx.exe

GAT-D APIライブラリ：S6E2DH-グラフィック解像度-Ver1.0 \ 02。GAT-DAPIライブラリ\ 01。IAR\01. bin\GAT-D API.a

グラフィックドライバー：S6E2D-2DGraphicsDriver-v1.00.798-IAR.zipのパッケージ内

S6E2D-2DGraphicsDriver-v1.00.798-IAR \ 01\_bin \ arm\_none\_iar \ release \ basic\_graphics.a

S6E2D-2DGraphicsDriver-v1.00.798-IAR \ 01\_bin \ arm\_none\_iar \ release \ util.a

# GAT-Dツールの段階的な利用

次の手順では、GAT-Dツールを使用して、S6E2DHシリーズベースのLCDデバイスに表示するビデオ（一連のフレーム）を配置する方法について説明します。

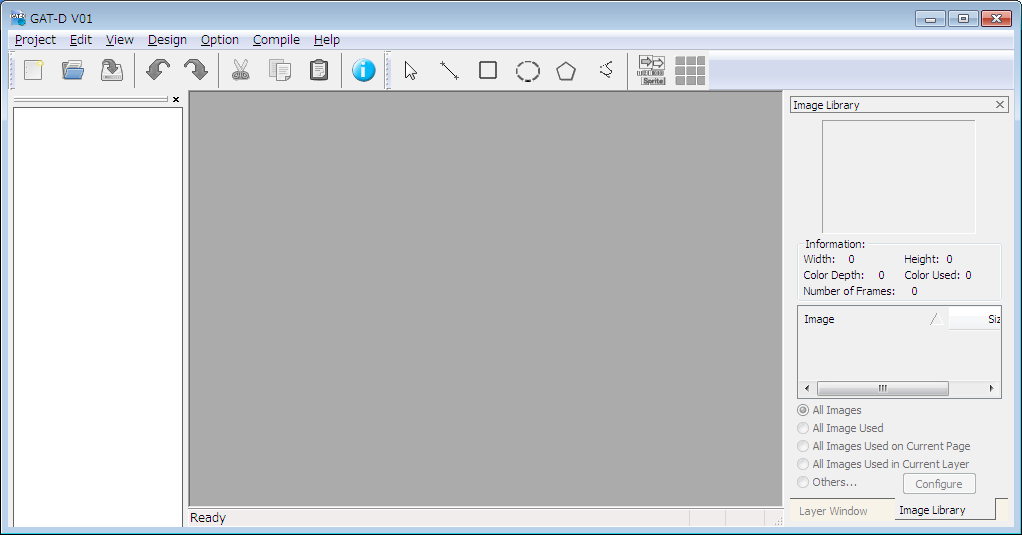
1. 新しいプロジェクトを作成する
2. 画像をインポートする
3. ページの作成
4. フローデザインテーブルの作成
5. ソース生成
6. 生成されたソースをIARプロジェクトに追加します
7. ビルド後にダウンロードして実行する

また、想定される条件：

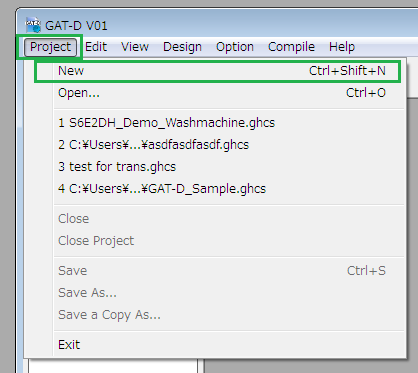
1. GAT-Dツールがインストールされています
2. IAR Embedded Workbenchがインストールされています（ver.7.10以降）
3. 用意された画像素材
4. S6E2DHシリーズベースのLCDデバイスとICE（I-Jet推奨）を用意

## プロジェクトの新規作成

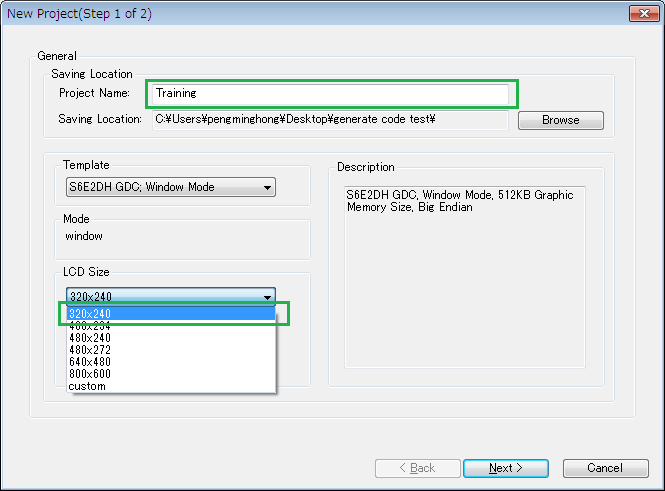
ステップ1：GAT-Dツールを開くと、ツールは次のように開きます。

図2。GAT-Dツールウィンドウを開きました

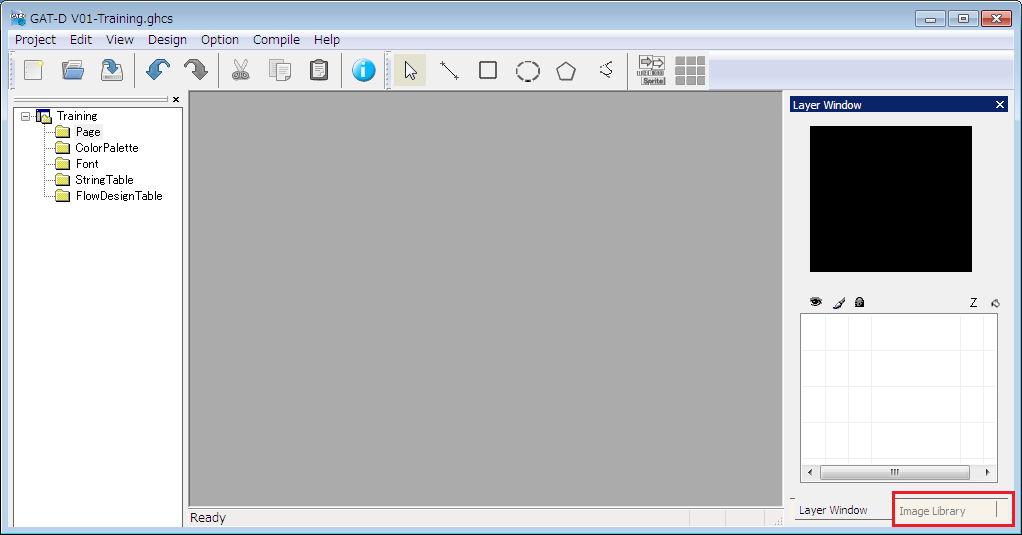
ステップ2：「プロジェクト」メニューをクリックし、次のように「新規」をクリックします

図3。新しいプロジェクトメニュー

ステップ3：名前を入力し、LCDサイズを選択します。

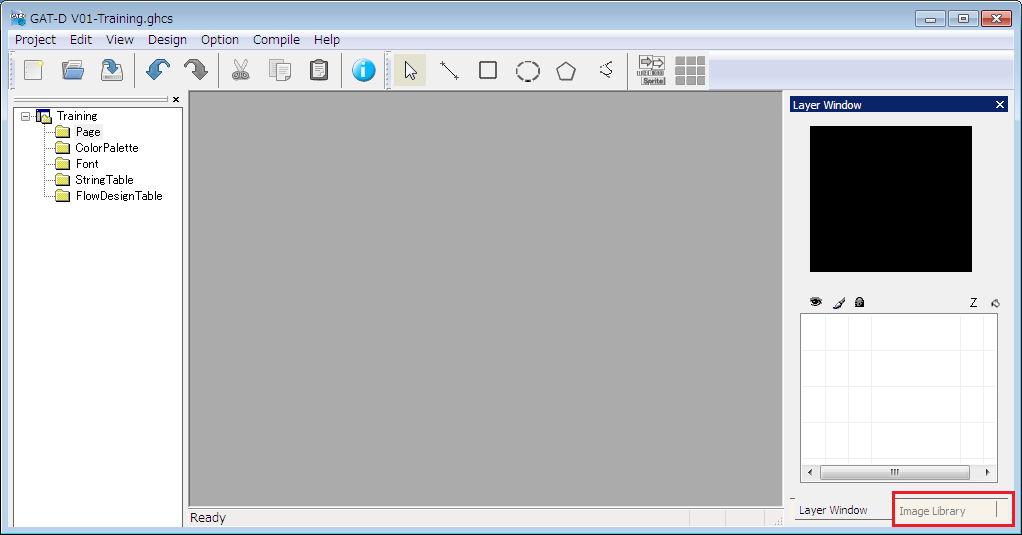
図4。新規プロジェクトステップ1/2

ステップ4：「次へ」をクリックしてから「完了」をクリックすると、次のように新しいプロジェクトが作成されます。

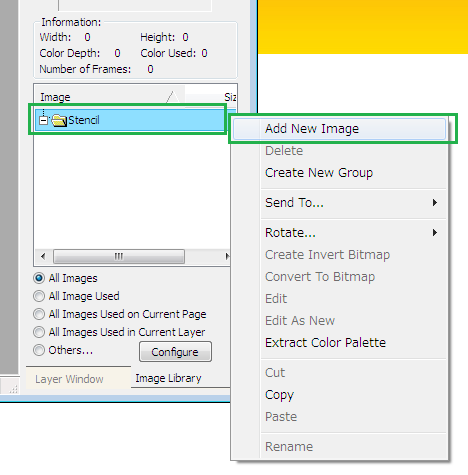
図5。新しい計画

## 画像のインポート

ステップ1：「画像ライブラリ」をクリックします

図6。[画像ライブラリ]タブ

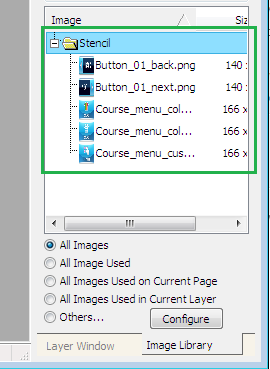
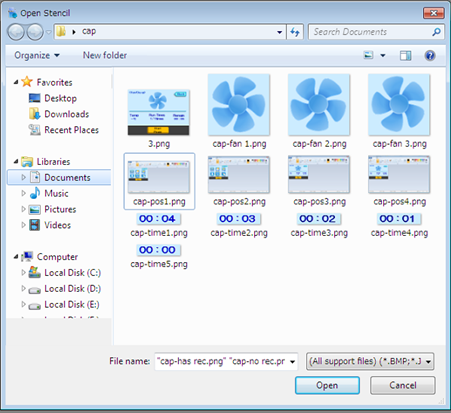
ステップ2：「画像ライブラリ」の「ステンシル」を右クリックし、メニューの「新しい画像を追加」をクリックします。

図7。新しい画像メニューを追加

ステップ3：パスに移動して画像を選択すると、画像は次のように表示されます。

注：画像のファイルタイプは、bmp、png、jpeg、gif、psdをサポートしています。

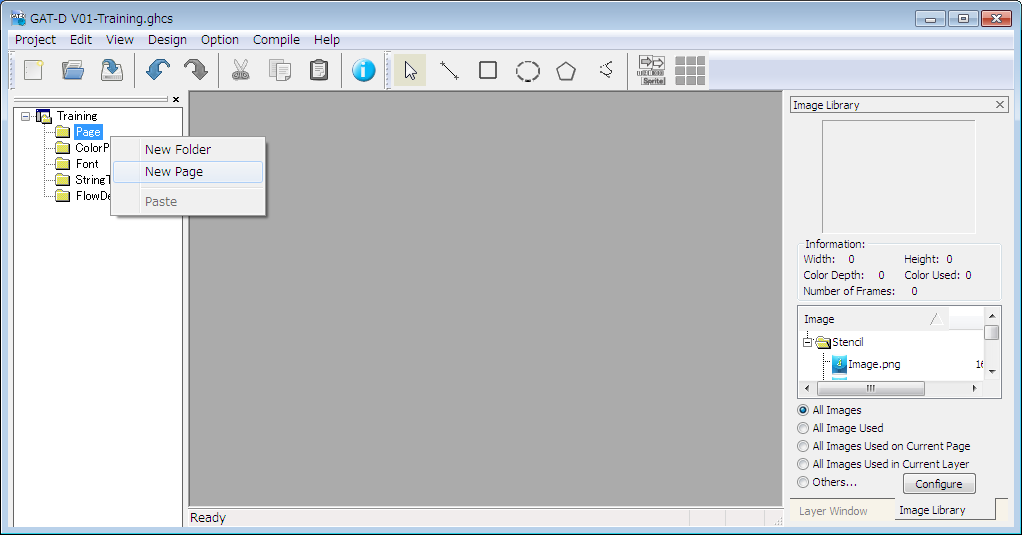
図8。画像リストを追加



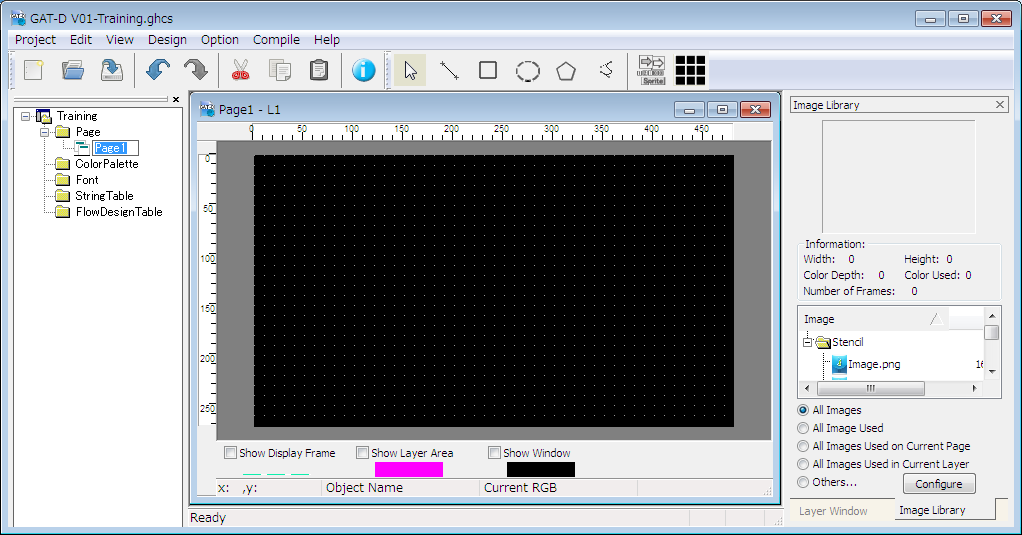
## ページの作成

### 新しいページを作成する

ステップ1：「ページ」フォルダを右クリックし、メニューの「新しいページ」をクリックします。

図9。新しいページメニュー

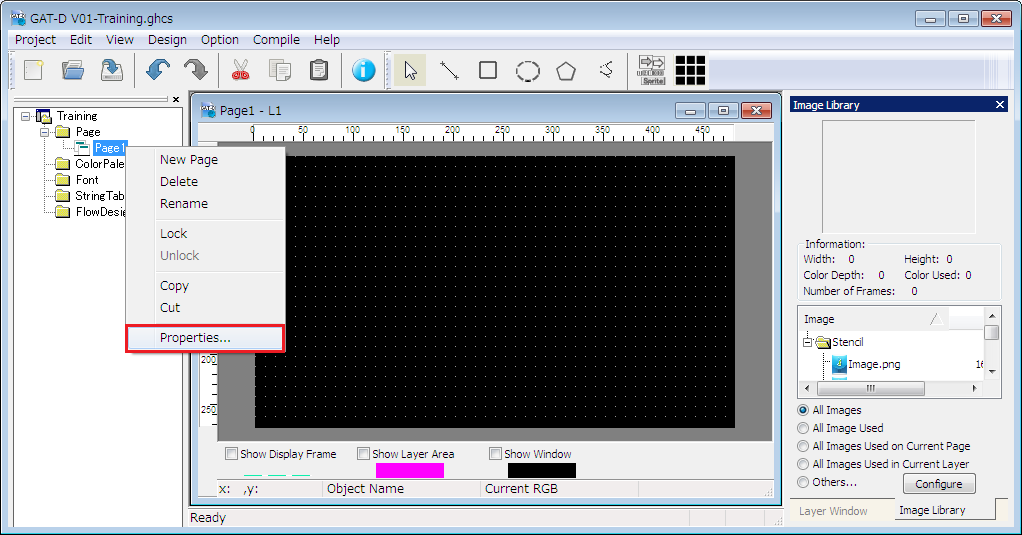
次に、次のように新しいページが作成されます。

図10。新しいページを作成しました

### ページ構成

ステップ1：「Page1」を右クリックし、メニューの「プロパティ…」をクリックします

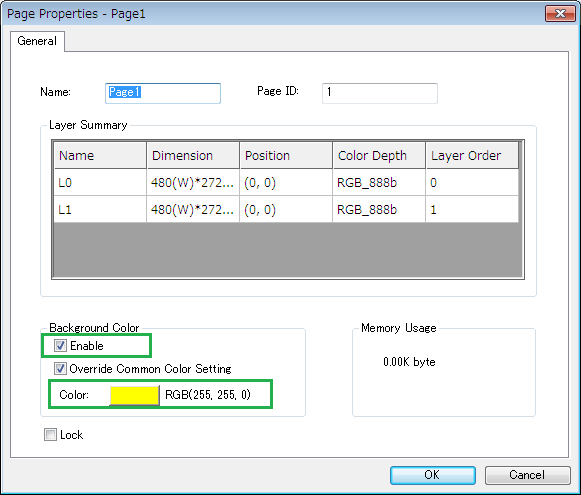
図11。ページプロパティメニュー



ステップ2：「有効にする」にチェックを入れます。ページの背景色が変更される可能性があります。

RGBカラーを選択します。ページに何もない場合、LCDは背景色を表示します。

図12。ページプロパティウィンドウ

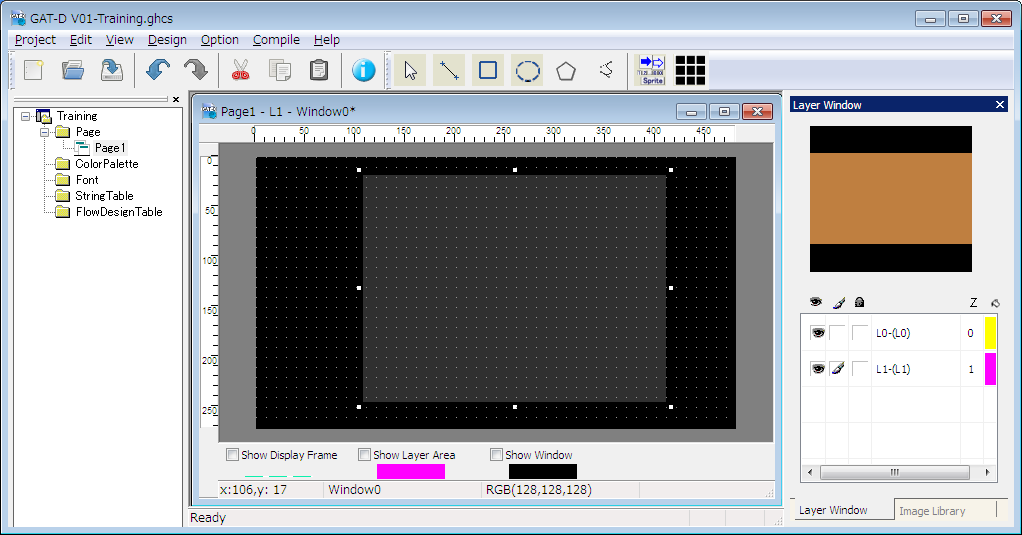


### ページにウィンドウを追加

ステップ1：描画されたツールバーからウィンドウを選択し、左クリックして作業領域に移動します。ウィンドウは次のように追加されます。

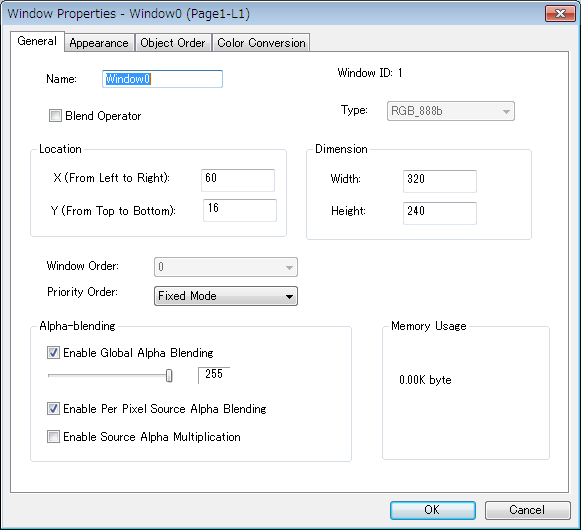
レイヤーに複数のウィンドウを追加する場合は、レイヤーオプションをマルチウィンドウレイヤーに設定する必要があります。そうしないと、レイヤーに1つのウィンドウしか追加できません。

図13。ウィンドウを追加



ステップ2：作業領域のウィンドウをダブルクリックすると、ウィンドウ構成ウィンドウが表示されます。それらを構成します。

図14。ウィンドウプロパティ



1つのウィンドウに複数の画像を追加する場合は、「ブレンド演算子」をチェックする必要があります

図3-14は、「ブレンド演算子」に設定されているウィンドウと設定されていないウィンドウの画像の違いを示しています。

図15。ブレンドオペレーター

外部フラッシュ

「ブレンドオペレータ」に設定されたウィンドウ内の画像

ウィンドウ内の画像が「ブレンドオペレータ」に設定されていない

S6E2DH MCU

GDCブレンドエンジン

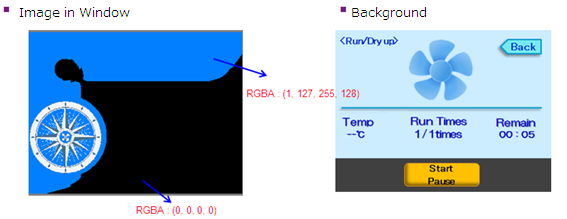
VRAM

GDCディスプレイエンジン

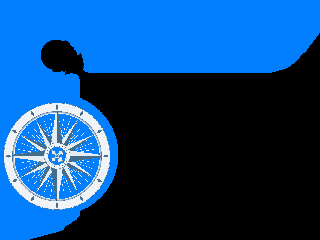
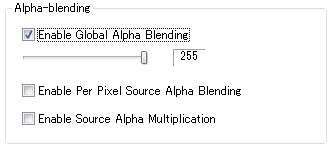
LCD

「アルファブレンディング」の設定は、ウィンドウ上の画像と背景のブレンド結果に影響します。

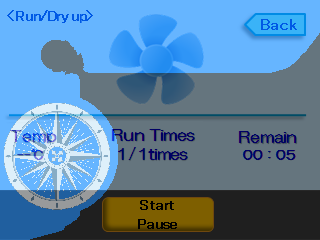
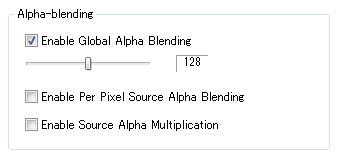
図16。ウィンドウアルファブレンドイラスト



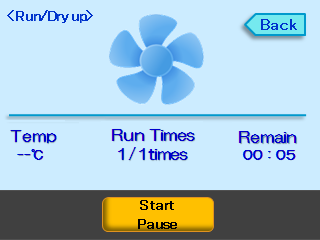
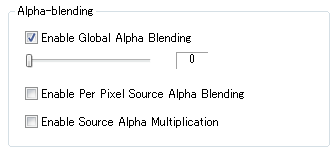
**イラストA：**



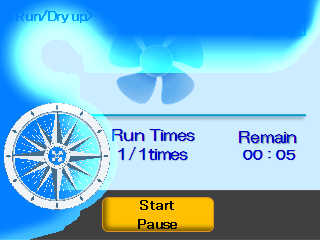
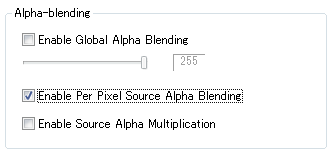
**イラストB：**



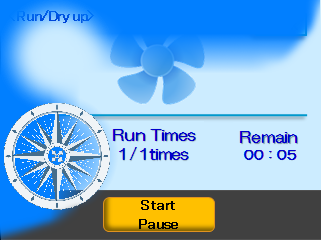
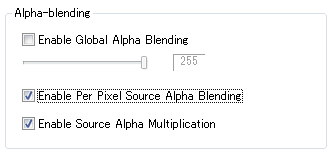
**イラストC：**



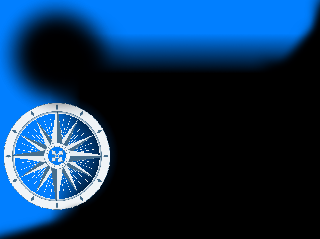
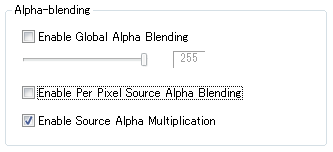
**イラストD：**



**イラストE：**



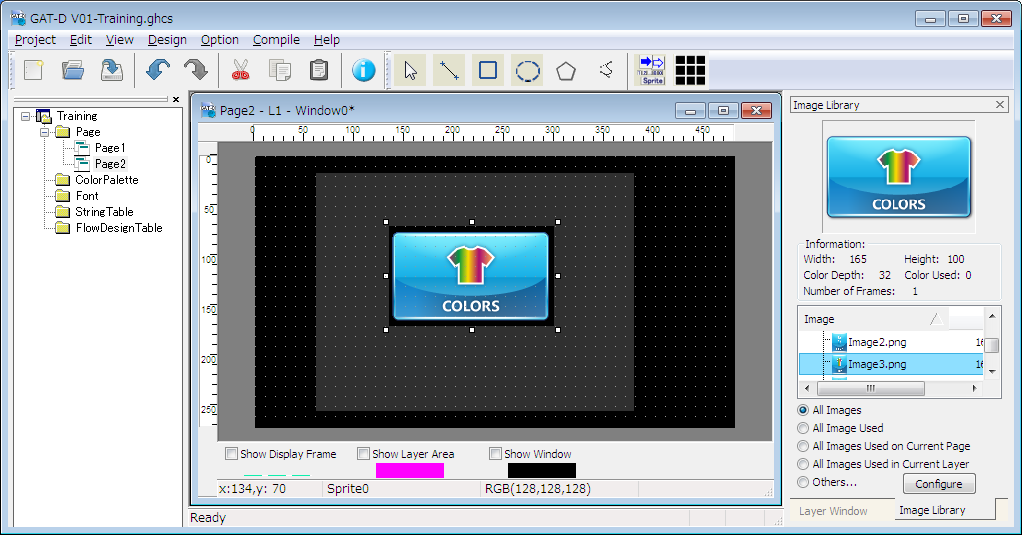
**イラストF：**



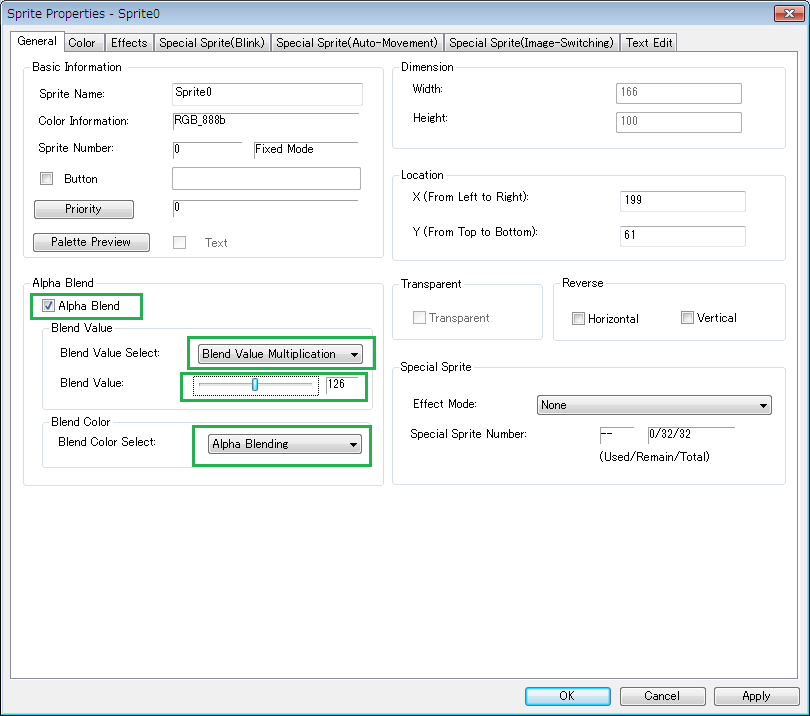
上の図は、S6E2DH-グラフィック解像度-Ver1.0 \ 03のサンプルのページ1からのものです。サンプル\ 01。GAT-D\_Tool\01. Training\Training.zip.

### ウィンドウに画像を追加

ステップ1：「画像ライブラリ」から画像を選択し、選択したウィンドウにドラッグアンドドロップすると、次のように画像が追加されます。

図17。ウィンドウに画像を追加

ステップ2：画像をダブルクリックすると、画像設定ウィンドウが表示されます。それらを構成します

図18。画像プロパティウィンドウ

アルファブレンド構成の動作については、次の図を参照してください。

図19。アルファブレンドイラスト

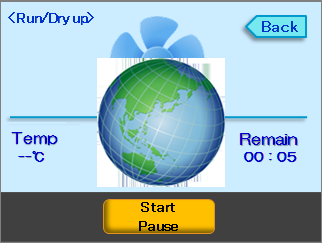
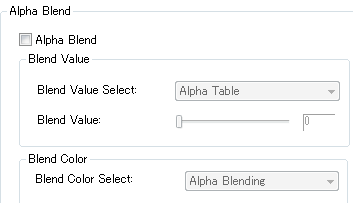
**使用した画像：**

RGBA : (255, 255, 255, 0)

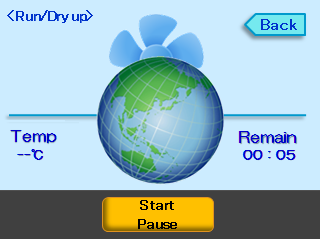
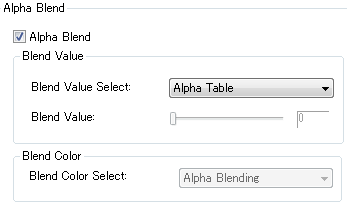


不透明

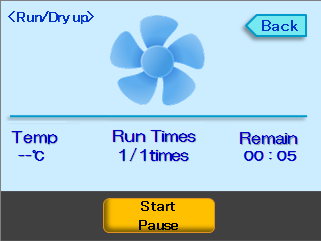
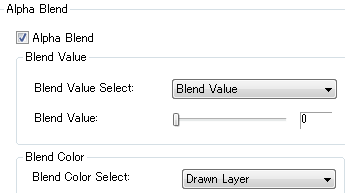
イラストA：



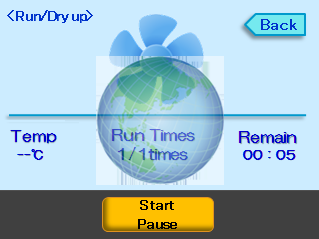
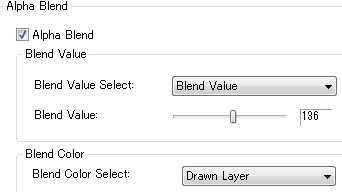
**イラストB：**



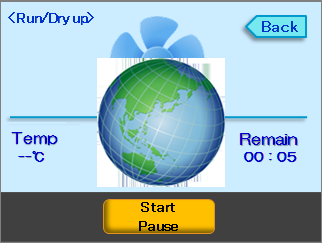
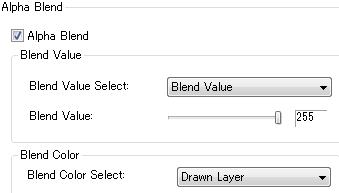
**イラストC：**



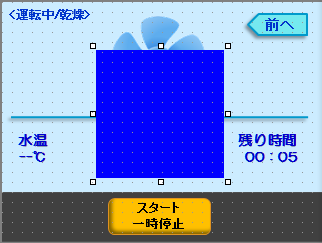
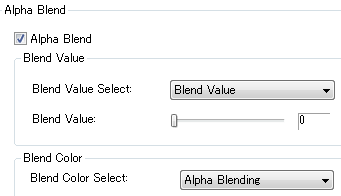
イラストD：



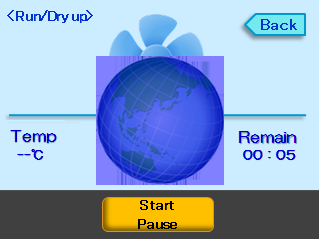
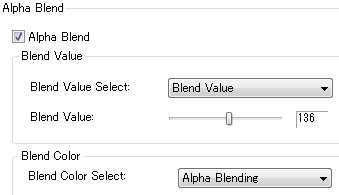
**イラストE：**



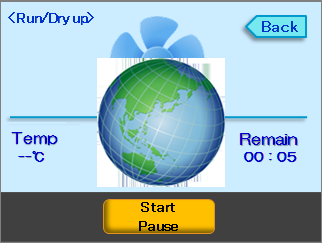
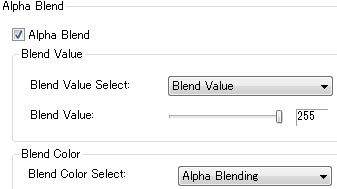
**イラストF：**



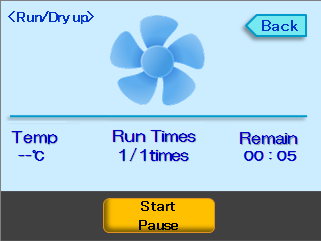
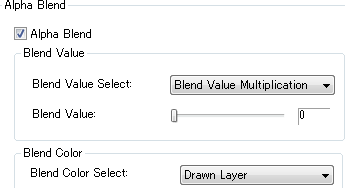
**イラストG：**



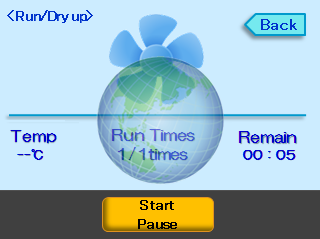
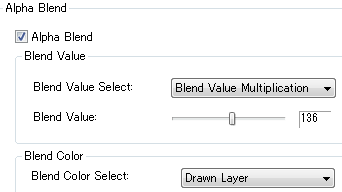
**イラストH：**



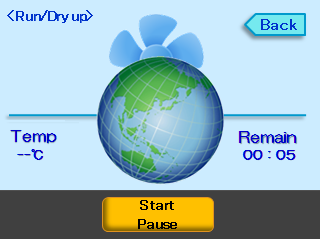
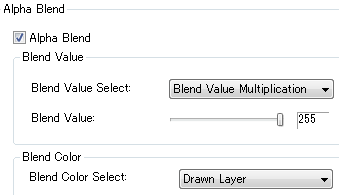
**イラストI：**



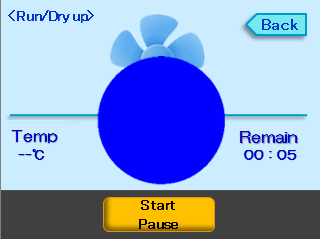
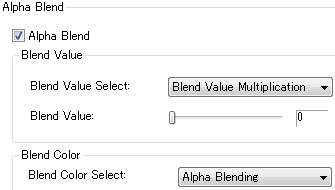
**イラストJ：**



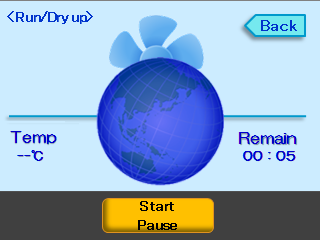
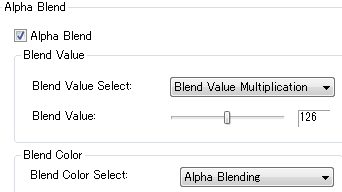
**イラストK：**



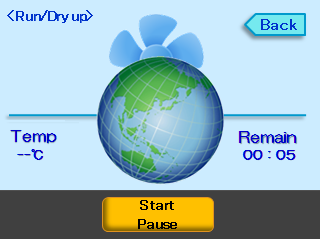
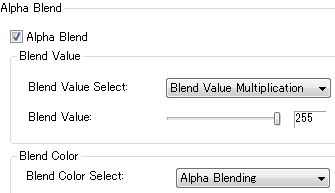
**イラストL：**



**イラストM：**



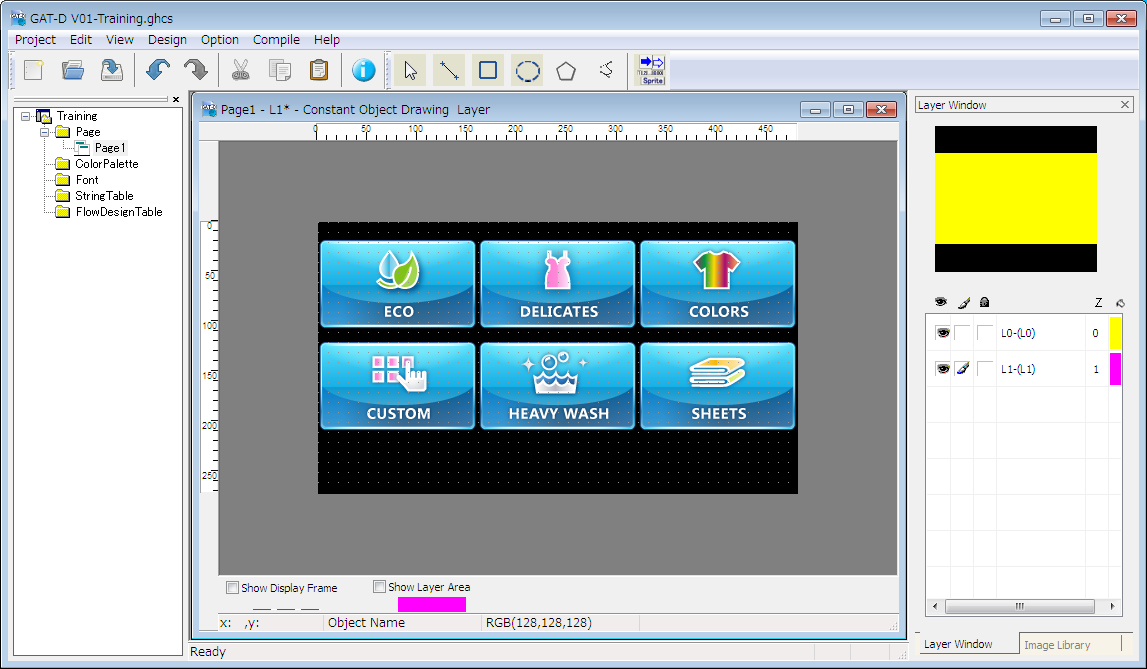
**イラストN：**



上の図は、S6E2DH-グラフィック解像度-Ver1.0 \ 03のサンプルの4ページからのものです。サンプル\ 01。GAT-D\_Tool \ 01。Training\Training.zip.

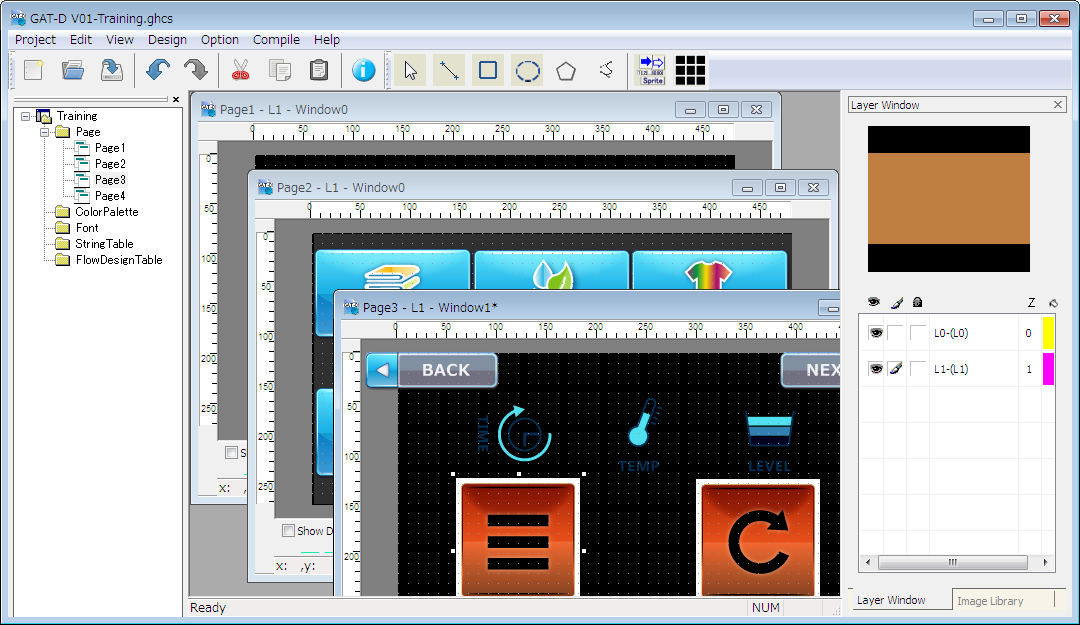
ステップ3：必要なイメージを追加して構成します。

注：ページの構成後、“View” -> “Page Preview Window” からプレビューできます。

図20。編集されたページビュー

### すべてのページを作成

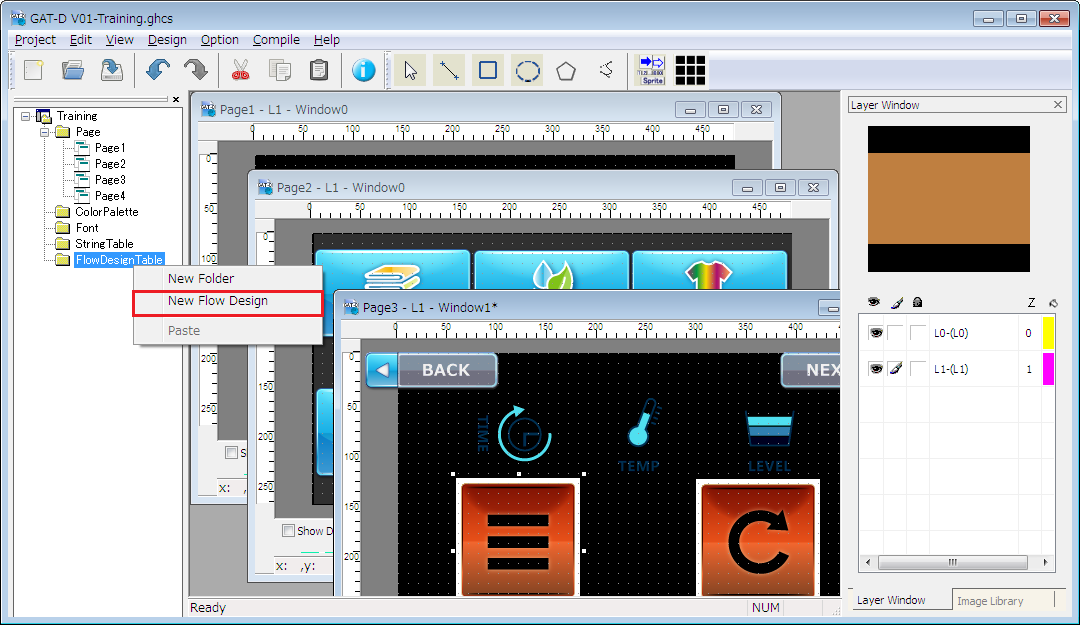
必要に応じて、さらにページを作成します。

図21。すべてのページビューを編集

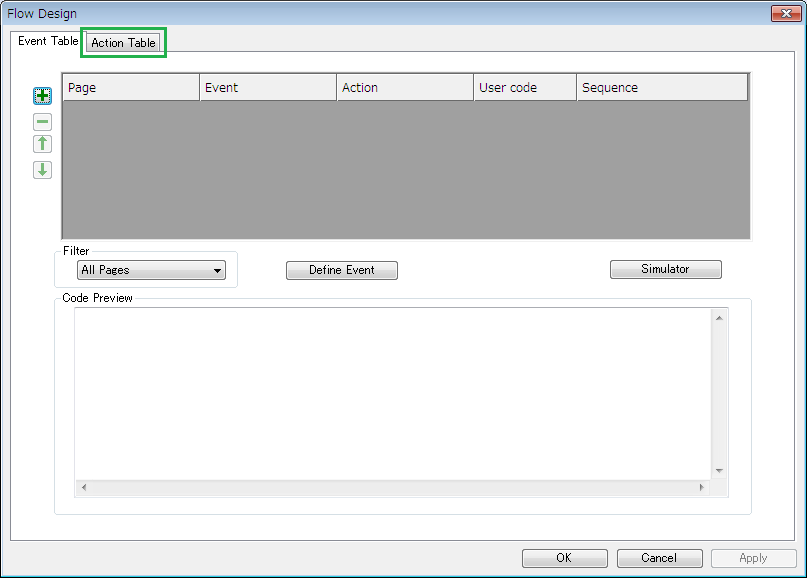
## フローデザインテーブルの作成

### フロー設計を追加する

ステップ1：「FlowDesignTable」フォルダーを右クリックし、メニューの「新しいフローデザイン」をクリックします。

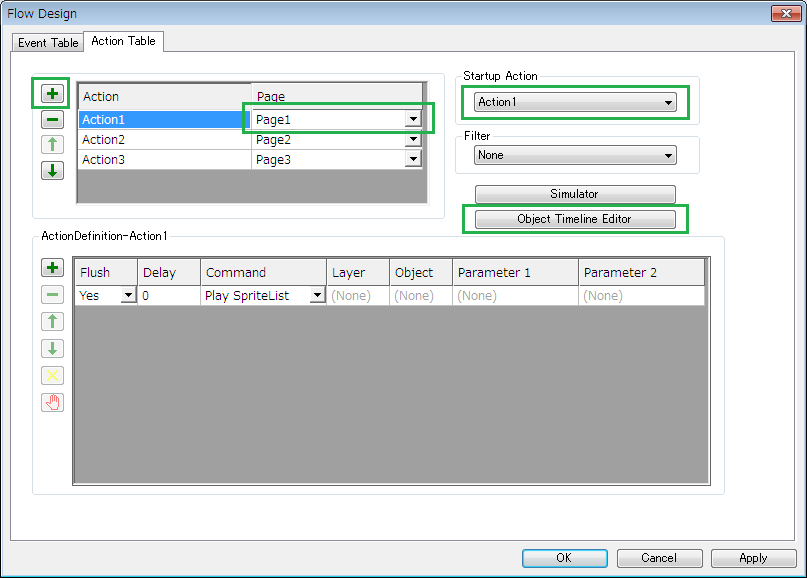
図22。新しいフロー設計メニューを追加

ステップ2：新しいフローデザインをダブルクリックすると、ウィンドウが次のように表示されます。ディスプレイの起動のみの場合はイベントを使用しないため、[アクションテーブル]を直接クリックします。

図23。フローデザインウィンドウアクションテーブルタブ

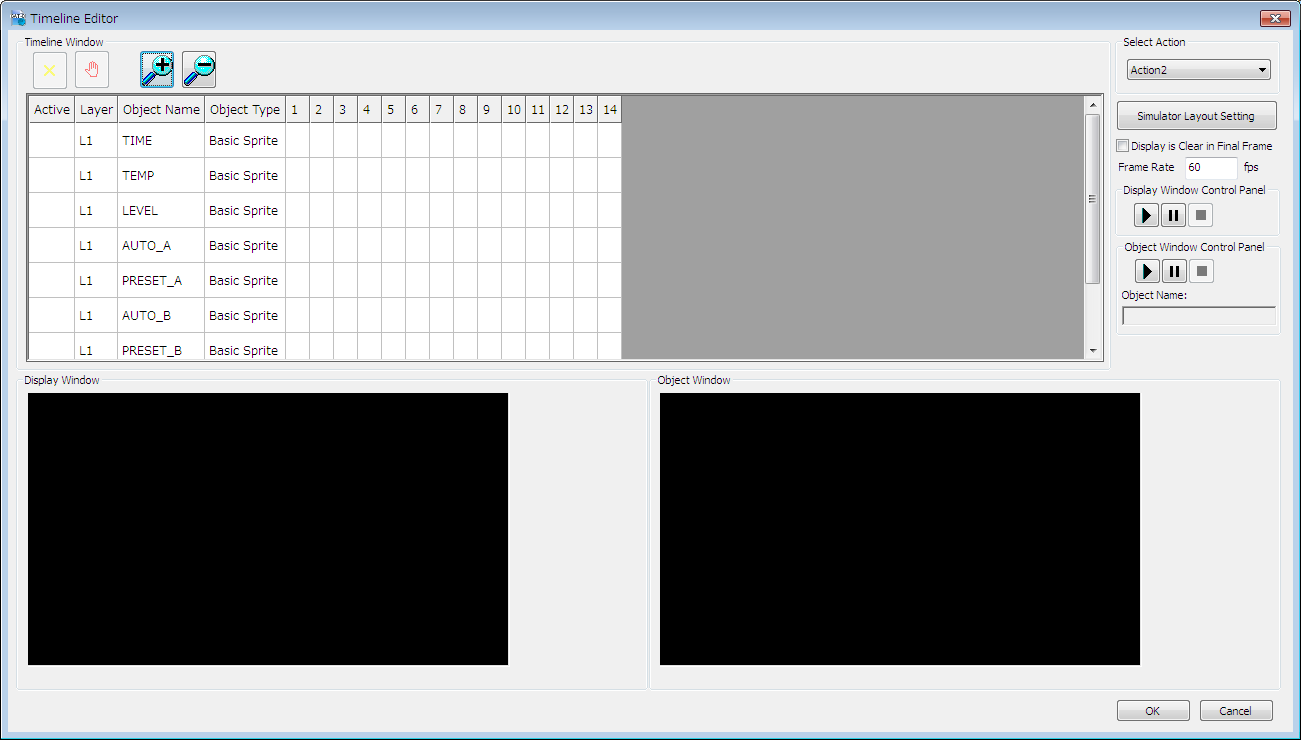
ステップ3：「+」をクリックしてアクションを追加し、表示するアクションのページを選択します。

1. アクションは小さなビデオ（一連のフレーム）のようなもので、

図24。アクションビューを追加

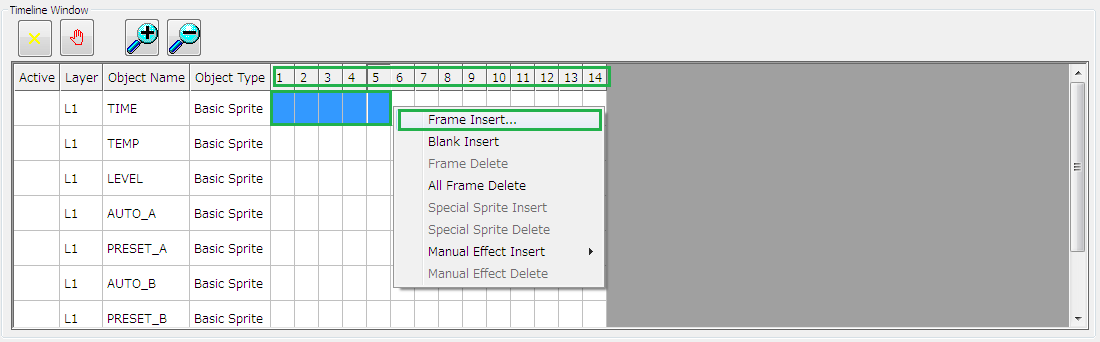
ステップ4：電源投入後に表示するアクションを決定する「スタートアップアクション」を選択します。

ステップ5：「オブジェクトタイムラインエディタ」をクリックすると、「タイムラインエディタ」ウィンドウが表示されます。

図25。タイムラインエディタウィンドウ

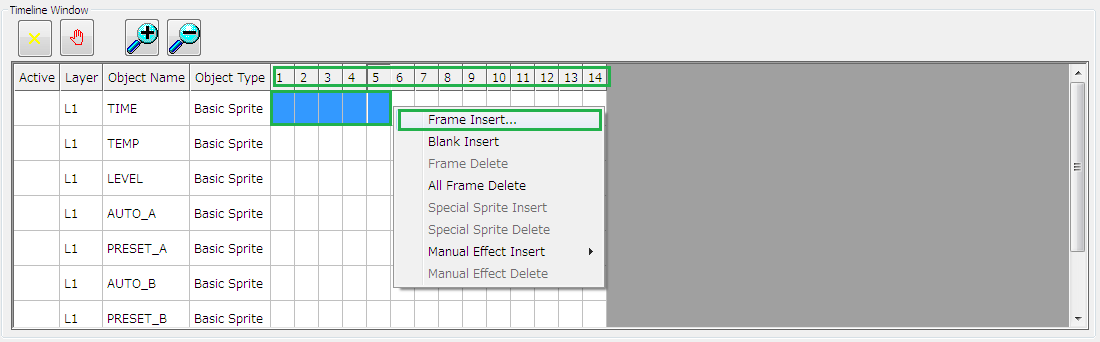
ステップ6：画像を表示するフレームを選択し、ダブルクリックして[フレーム挿入]を選択します。

1. ヘッド番号はフレームです。

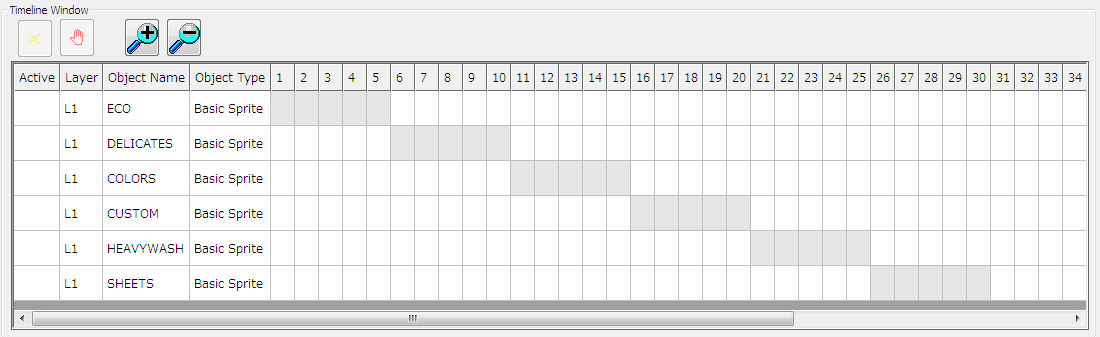
図26。フレーム挿入メニュー

ステップ6：画像を表示するフレームを選択し、ダブルクリックして[フレーム挿入…]を選択します

1. ヘッド番号はフレームです。

図27。フレーム挿入メニュー

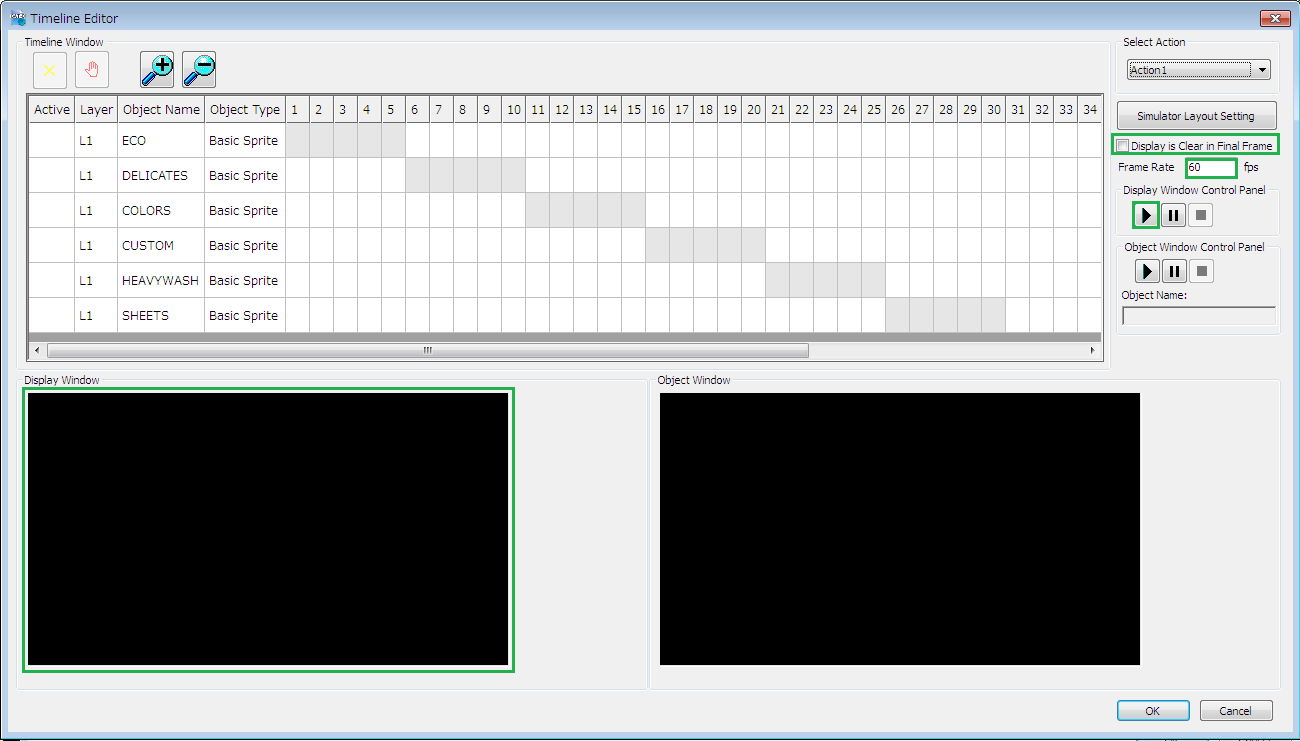
ステップ7：各画像にフレームを挿入します。各画像を5フレーム表示してから次に表示すると、タイムラインは次のようになります。

図28。挿入されたフレームビュー

ステップ8：アクションの再生後に何も表示されない場合は、「最終フレームで表示がクリア」をチェックします。

S6E2DHシリーズのフレームレートは60fpsな​​ので、「フレームレート」を60fpsに変更します。

「▶」をクリックすると、表示ウィンドウでのアクションをシミュレートできます。

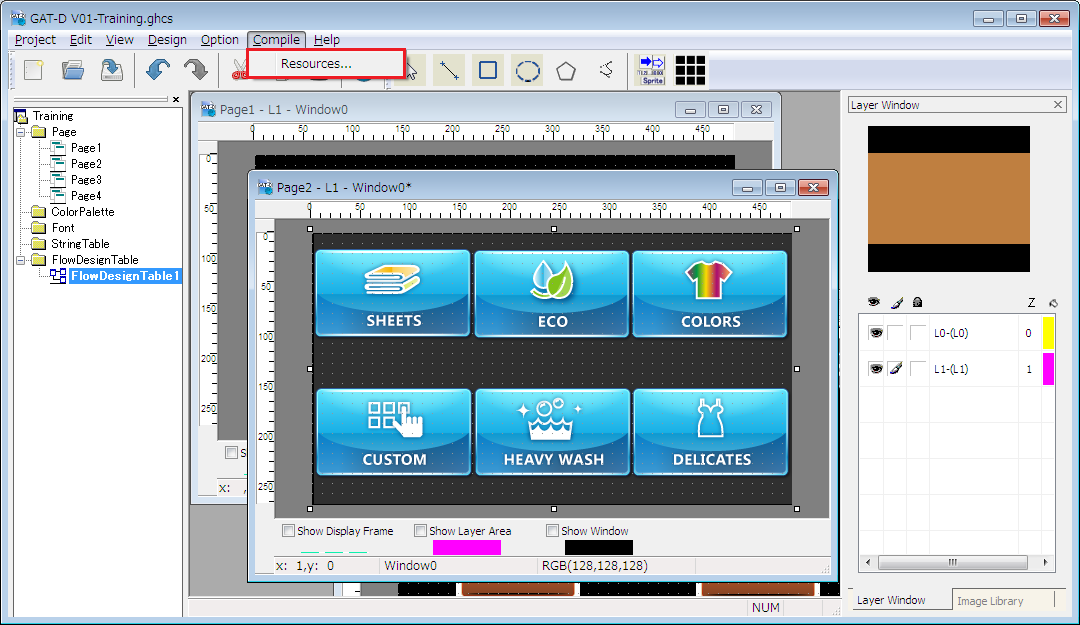
図29。表示オプション

### すべてのフロー設計を追加

さらにフローデザインを追加することもできますが、1つのプログラムに対して1つのフローデザインで十分です。

## ソース生成

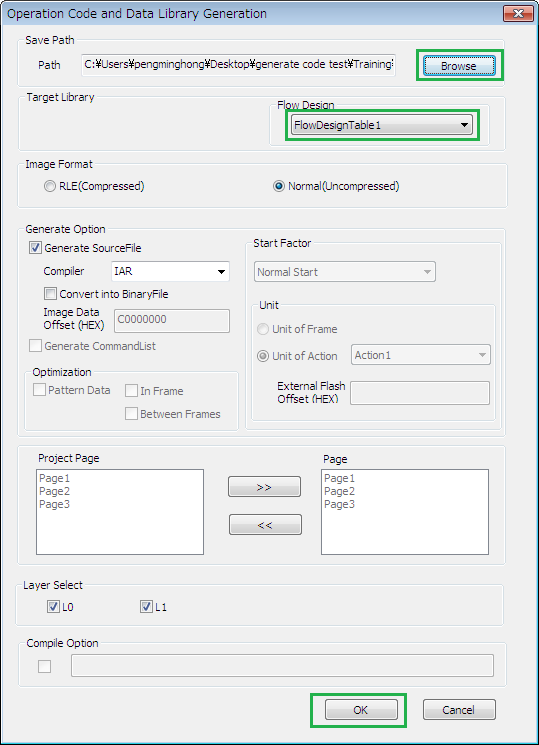
ステップ1：「コンパイル」をクリックし、「リソース…」をクリックして「オペコードとデータライブラリの生成」ウィンドウを開きます。

図30。ソース生成メニュー

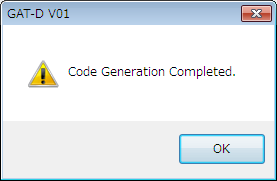
ステップ2：「参照」をクリックして、ソースが生成されたパスを選択します。

フロー設計がある場合はそれを選択します。

「OK」をクリックして、他のオプションのデフォルトでソースを生成します。

図31。ソース生成オプションウィンドウ

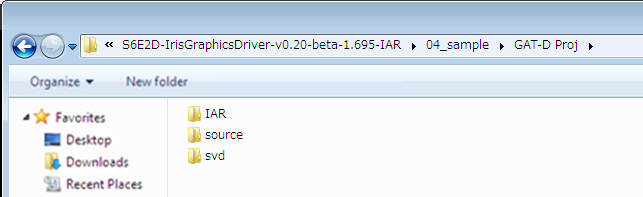
ステップ3：フォローウィンドウはソース生成の成功を意味し、生成されたソースはステップ2で選択したパスで見つけることができます。

図32。コード生成完了ウィンドウ

## 生成されたソースをIARプロジェクトに追加する

ステップ1：グラフィックスドライバーパッケージS6E2D-2DGraphicsDriver-v1.00.798-IAR.zipを入手し、任意のパスに解凍します。

S6E2DH-グラフィック解像度-Ver1.0-IAR \ 03を見つけます。サンプル\ 02。IAR \ GAT-D Proj.zipを作成し、グラフィックスドライバーの抽出パスの04\_sampleに抽出します。

図33。サンプルパス

ステップ2：IARプロジェクトを開きます。

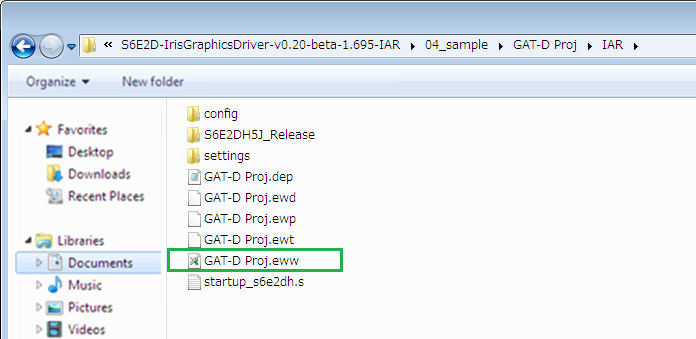
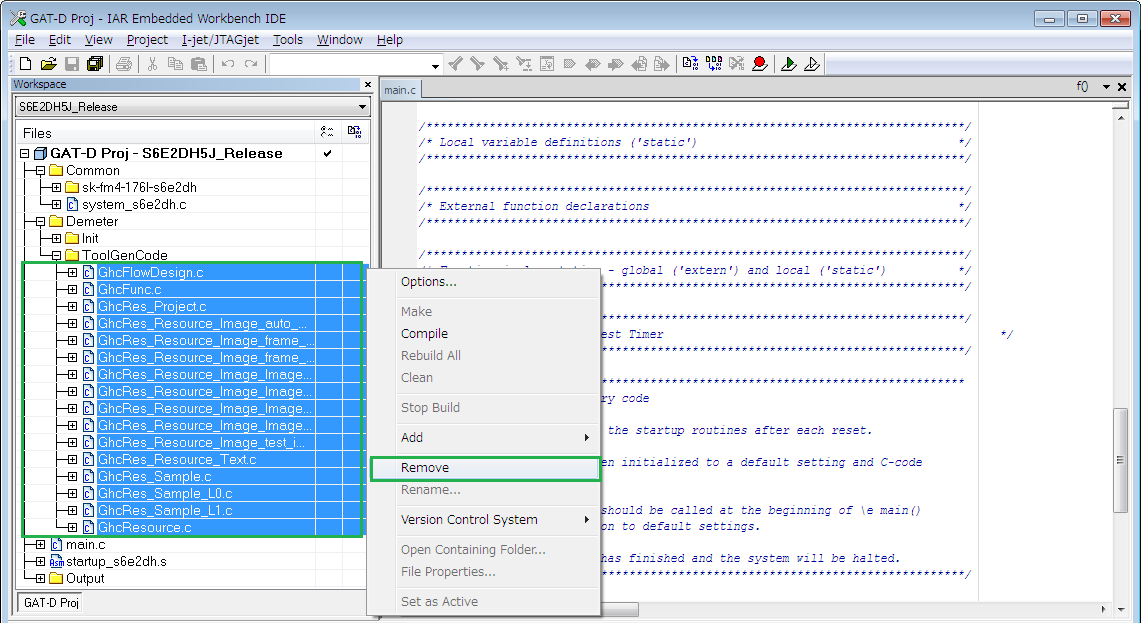
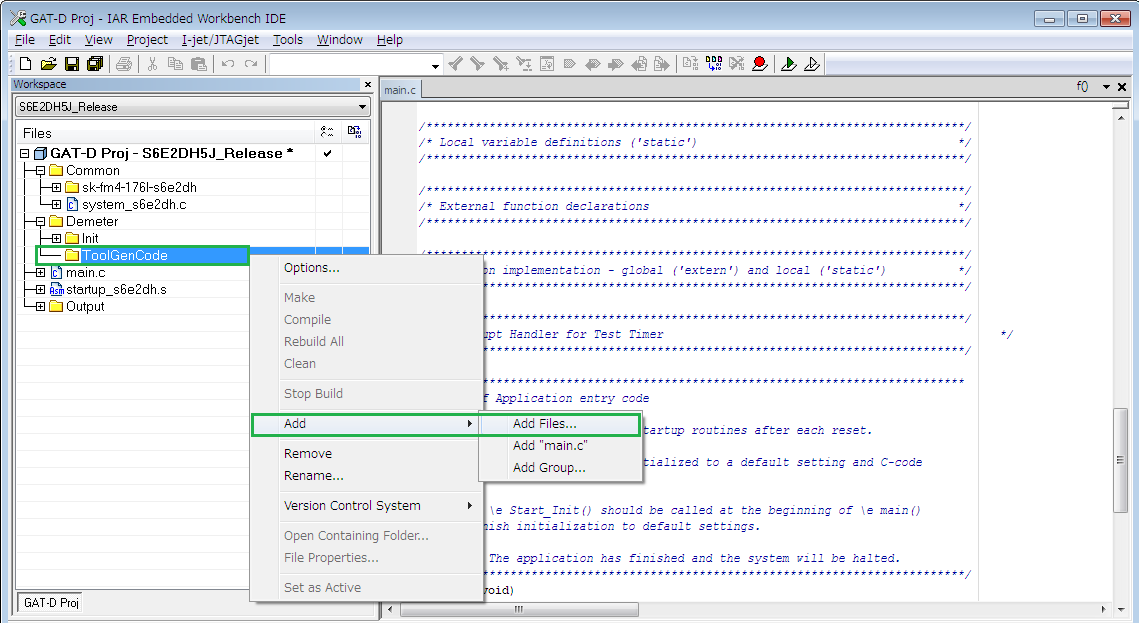
図34。サンプルプロジェクトファイル

図35。プロジェクト削除メニュー

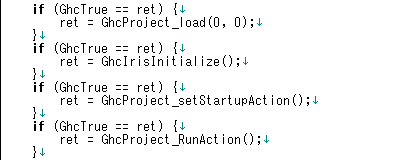
手順4：パス内のすべてのソースを削除します：S6E2D-2DGraphicsDriver-v1.00.798-IAR.zip \ 04\_sample \ GAT-D Proj \ source \ Demeter \ GenCode次に、新しく生成されたすべてのソースをパスにコピーします。

図36。ソース生成パス

ステップ5：S6E2D-2DGraphicsDriver-v1.00.798-IAR \ 04\_sample \ GAT-D Proj \ source \ Demeter \ GenCodeから新しく生成されたすべてのソース（\* .cファイルのみがOK）をIARプロジェクトに追加します。

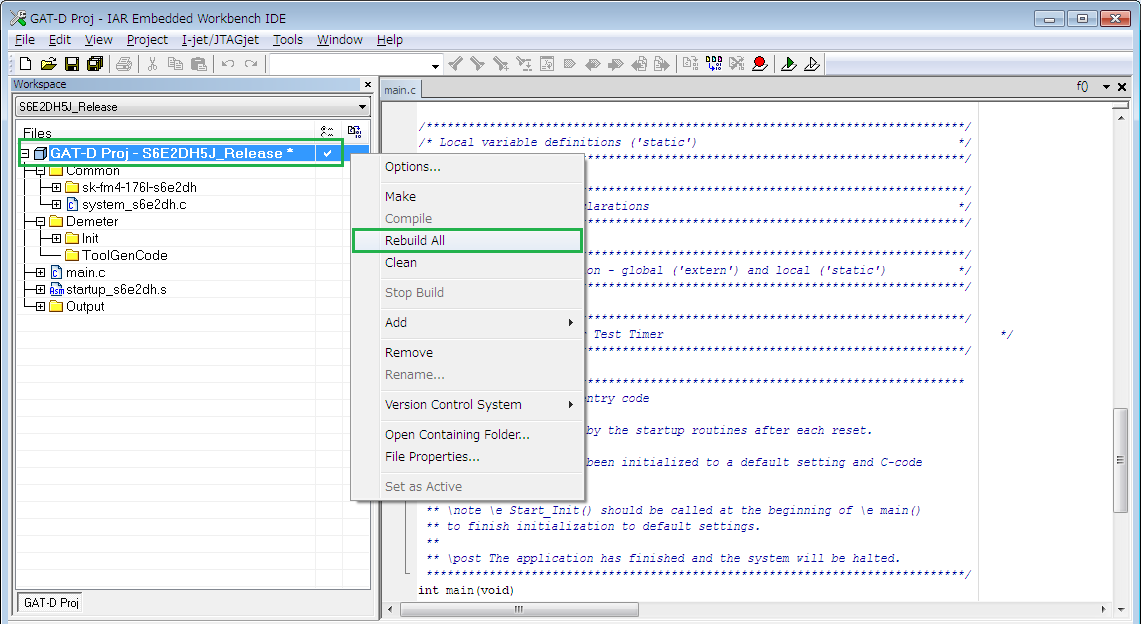
図37。新しいソース追加メニュー

1. APIの最小呼び出しはmain.cに実装されています。

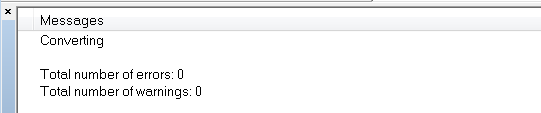
図38。APIの最小呼び出し

## ビルド後にダウンロードして実行

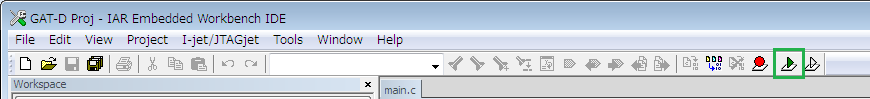
ステップ1：IARプロジェクトグループを右クリックし、メニューで[すべて再構築]を選択してプロジェクトをビルドします。

図39。メニューの再構築

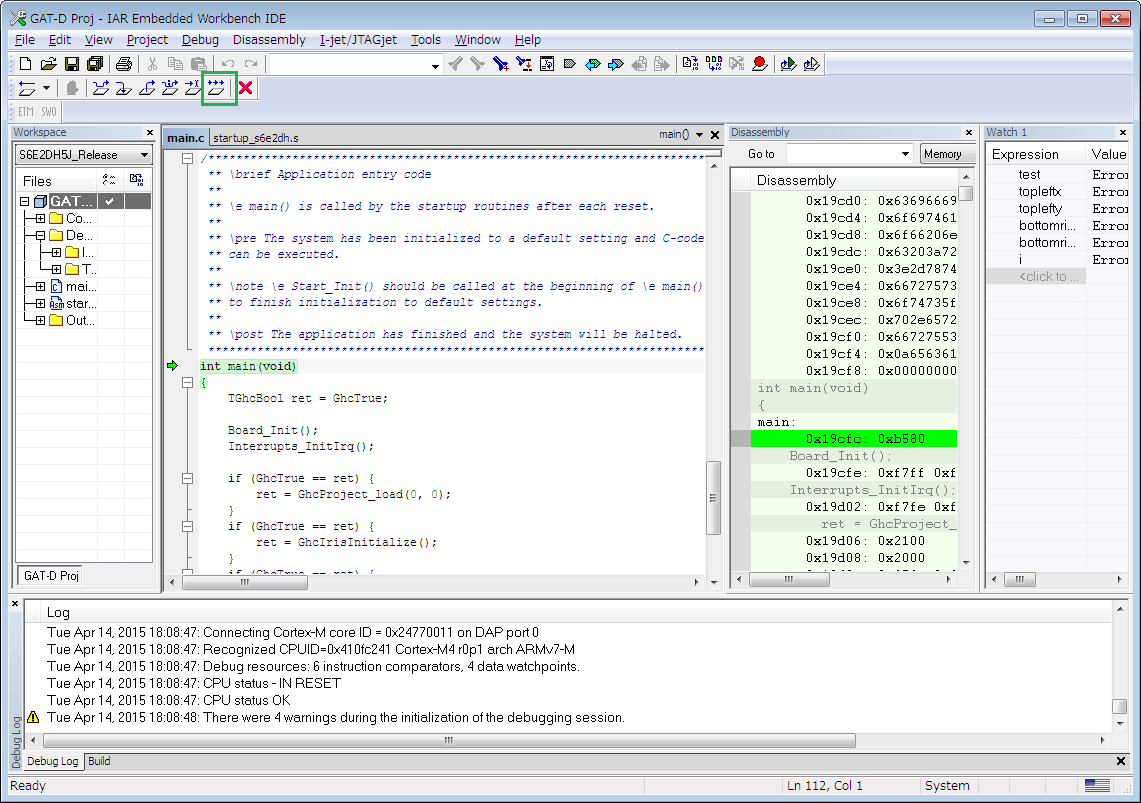
ステップ2：エラーがないことを確認します。

図40。ビルド結果

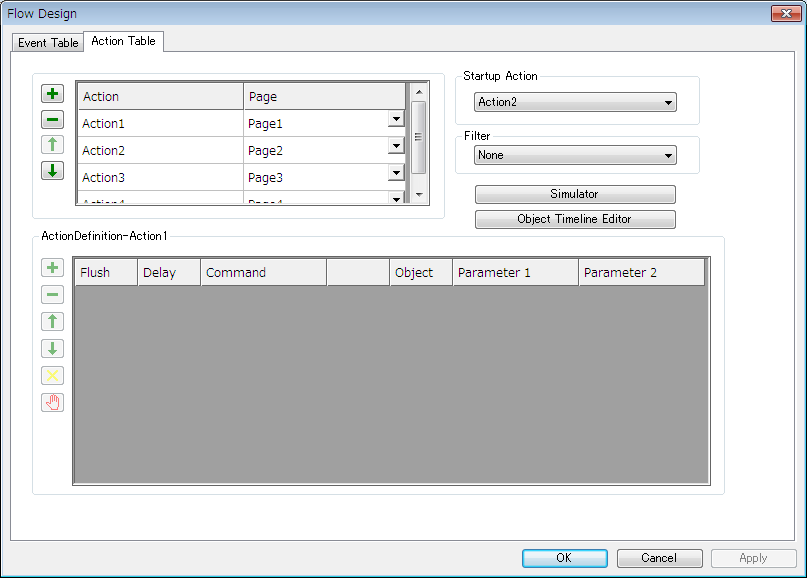
ステップ3：IAR EWIDEの[ダウンロードしてデバッグ]ボタンをクリックしてプログラムをダウンします

図41。デバッグボタン

ステップ4：「実行」ボタンをクリックして実行します。

図42。実行ボタン

手順5：LCDの表示が「アクションテーブル」で選択したアクションと同じであることを確認します。

図43。スタートアップアクションビュー

# GAT-Dの適用

## イベント管理

グラフィックベースの組み込みソフトウェアの一般的なモデルを説明した図に従ってください。

図44。組み込みソフトウェアモデル

組み込みソフトウェアモデル

ページ1

ページ2

Page3

Page4

Even1/Action2

Event3/ Action4

Event1/Action2

Event4/Action5

Event5/Action3

Event2/Action3

Event7/Action1

Startup/Action1

1\*:

Event1/Action2

GAT-Dを使用すると、上記の機能はPC側で記述でき、組み込み側で作業する代わりに、ソースを自動的に生成できます。これにより、組み込み側ではイベント確認を実装するだけで済み、1つのイベントがトリガーされた場合、GAT-D APIを呼び出します：GhcProject\_sendEvent\_Ex（eGEN\_xxxx）参照されたアクションが表示されます。

xxxx：イベント名。詳細については、5.1.6も参照してください。

以下はサンプルです。

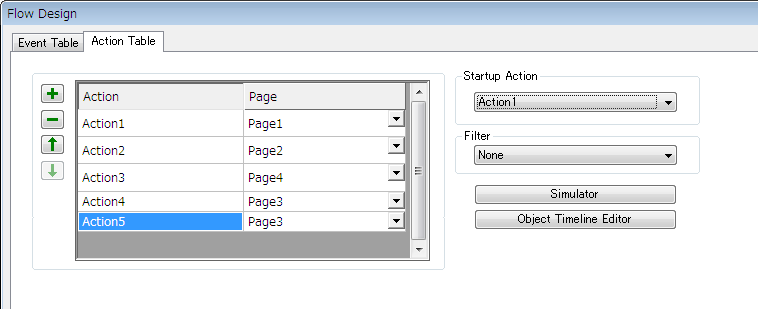
図45。アクションテーブルのサンプル

図46。イベントテーブルのサンプル

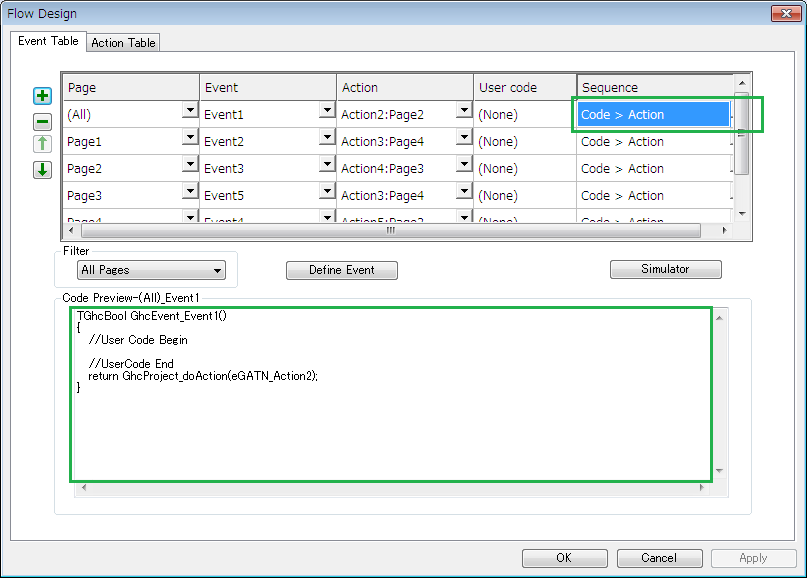
上の図は、S6E2DHのサンプルのフローデザイン-グラフィック解像度-Ver1.0 \ 03からのものです。サンプル\ 01。GAT-D\_Tool\01. Training\Training.zip.

イベントタイプには、システム、ハードウェア、オブジェクトの3種類があります。PC側では実際に違いはありません。

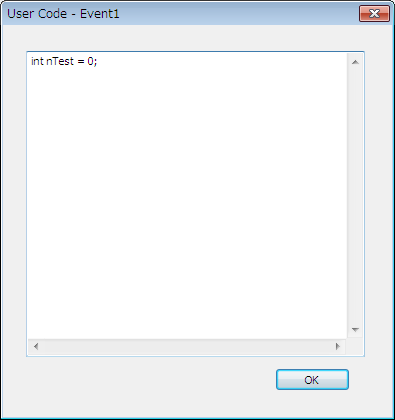
## ユーザーのソースコード

GAT-Dは、イベント管理に加えて、ユーザーが標準のCソースコードを入力する場所を提供します。

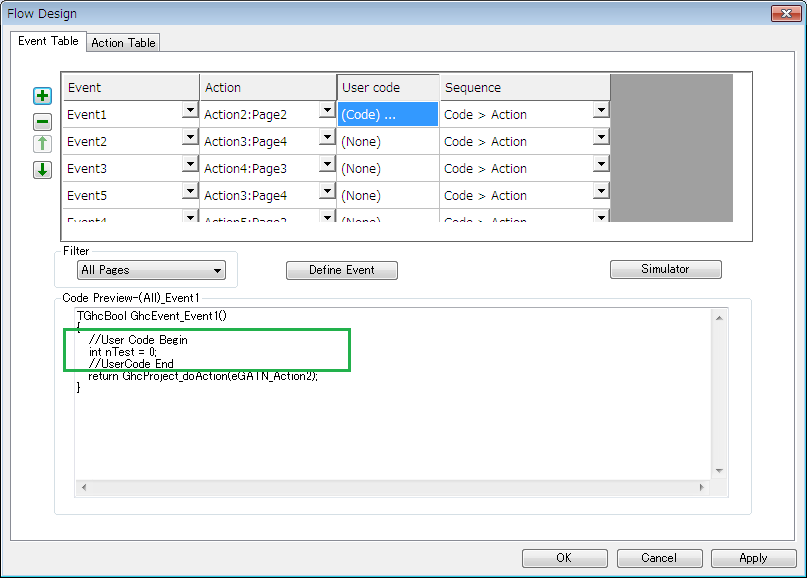
次のように：

図47。ユーザーのコードビュー

「（なし）」をダブルクリックすると、編集ウィンドウが表示されます。標準のCコードを次のように入力します。

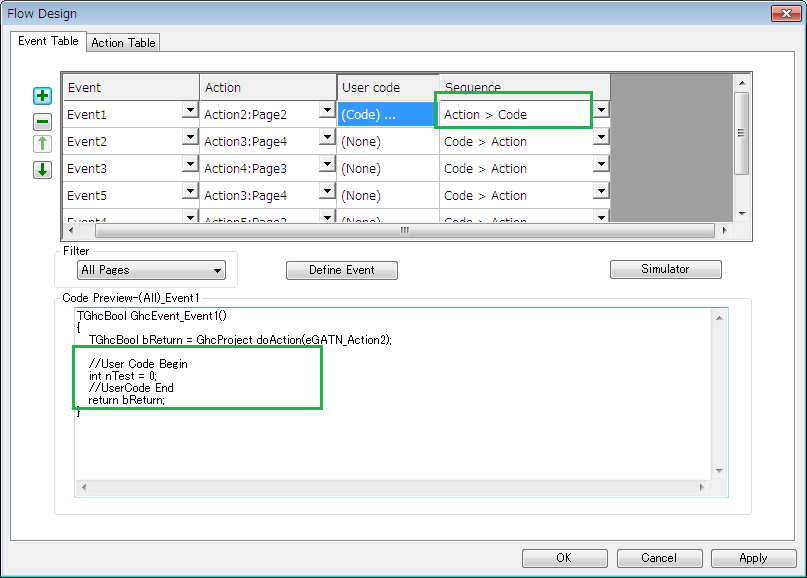
図48。ユーザーのコード編集ウィンドウ

入力されたコードは自動的に追加されます：

図49。編集されたコードビュー

「シーケンス」を「Code > Action」に選択すると、アクション表示前にコードが実装されます。

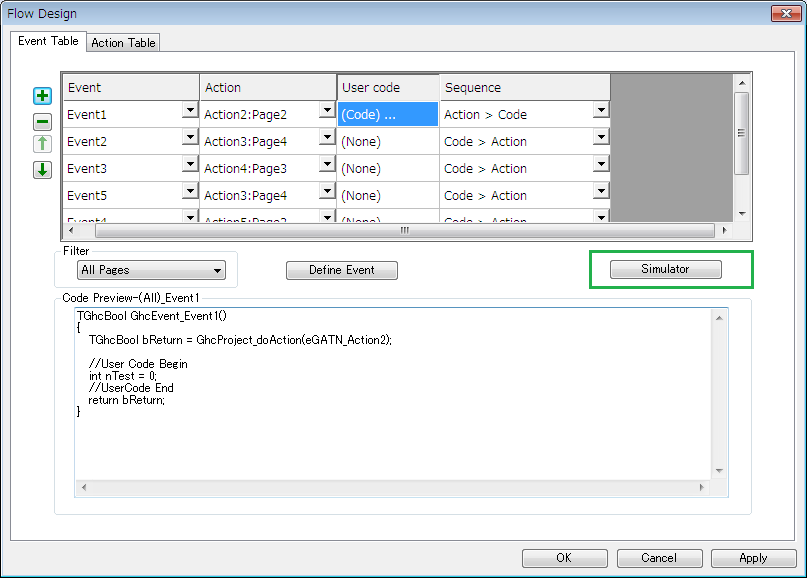
アクション後の表示を実装する場合は、次のように「Action > Code」を選択できます。

図50。ユーザーのコード順序

## ディスプレイシミュレーション

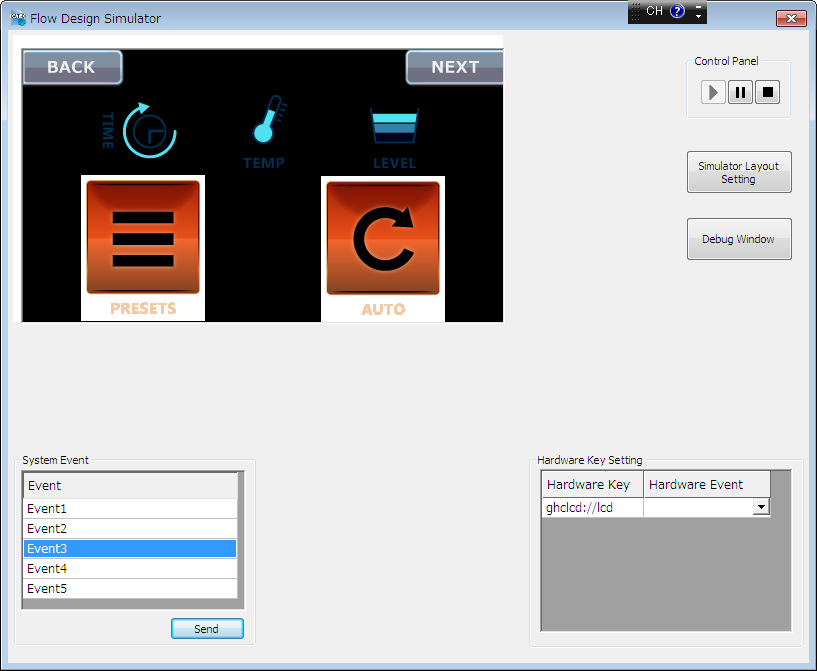
イベントテーブルとアクションテーブルを編集した後、GAT-Dで動作をシミュレートできます。

「シミュレーション」をクリックすると、「フローデザインシミュレータ」ウィンドウが表示されます。

図51。シミュレーションボタン

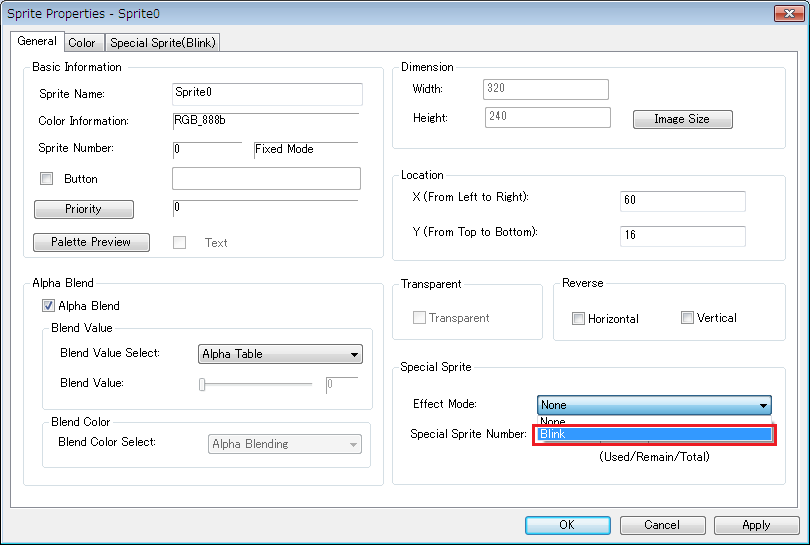
「▶」をクリックすると「スタートアップアクション」が再生され、イベントを選択すると「送信」をクリックすると、イベントテーブルの説明に従ってアクションが再生されます。

図52。シミュレーションウィンドウ



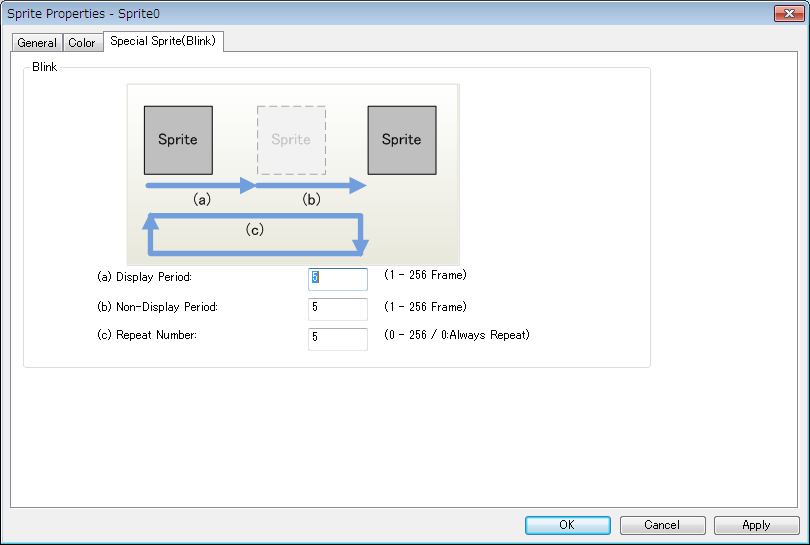
## スプライト特殊効果

GAT-D V01はスプライトの特殊効果をサポートしています。図に従って、どこに設定するかを示します。

図53。スプライト特殊効果

### 点滅

特殊スプライト効果モードを設定した後、次の図に従って特殊効果を設定できます。

図54。スプライトブリンク

点滅効果の場合、オブジェクトのタイムラインを編集して表示し、繰り返し表示しないことで実現できますが、ここで設定した特殊効果により、埋め込み側のメモリを節約し、オブジェクトのタイムラインを編集する時間を節約できます。特殊スプライト効果は際限なく点滅します。

## タッチパネルの解像度

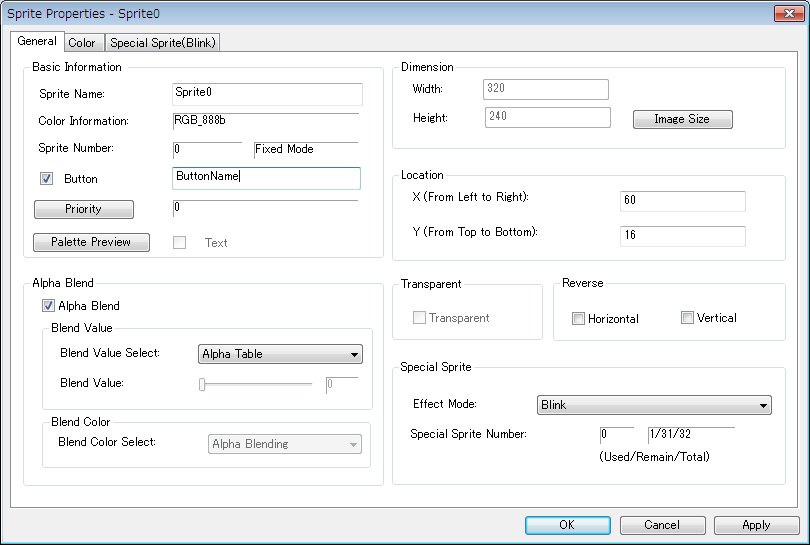
S6E2DHデバイスは一般的にLCDを装備し、主にタッチパネルも装備しています。最も一般的なタッチパネル操作は押しボタンです。ボタンが押されているかどうかを判断するには、押された座標とボタン領域を取得する必要があります。つまり、各ページのすべてのボタン領域を管理します。通常、ボタンは単一のオブジェクトとして設計されており、オブジェクトの座標はすでにGAT-Dによって管理されています。したがって、GAT-Dは特定のボタンの座標を提供できます。

また、GAT-Dは現在表示されているページを表示することもできるため、機能ソフトウェアは表示の状態遷移を管理する必要がありません。

プッシュイベントが発生した場合は、最初に現在表示されているページを取得し、次にページ上のボタンの座標を取得します。次に、プッシュされた座標を使用して、それが真のボタンプッシュであるかどうかをクリアできます。プッシュイベントが参照される表示である場合は、GAT-D APIを呼び出します。GhcProject\_sendEvent\_Ex（）は実際のアクションを表示できます。See 4.1.

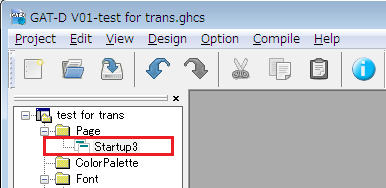
### ボタン名

タッチパネルを搭載したLCDデバイスをサポートするために、組み込み側のGAT-D APIは、ユーザーにAPI GhcFunc\_getButtonAreaByIndex（）を提供します。ボタンエリアの座標は、APIを呼び出すことで取得できます。画像やオブジェクトがボタンで、APIを呼び出して座標領域を取得したい場合は、PC側で「ボタン」をチェックして「ボタン名」を入力する必要があります。「ボタン名」は、APIの呼び出しで使用される標準のC式に適用する必要があります。詳細については、5.1.7も参照してください。

図55。ボタンオプション

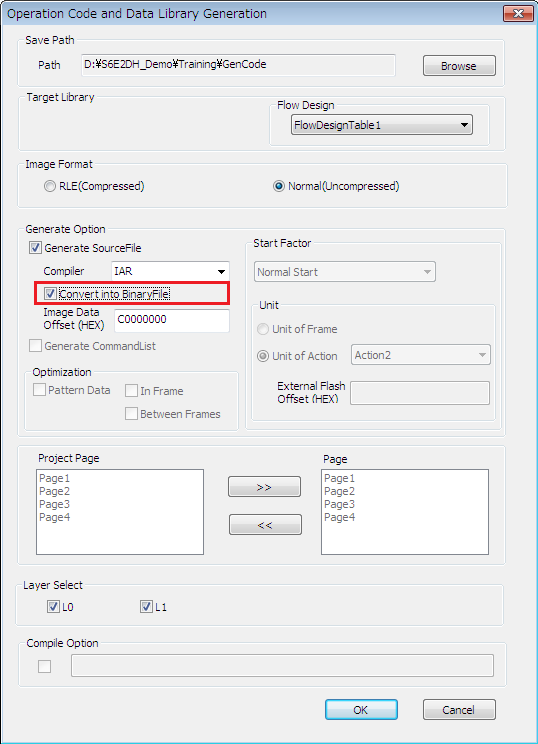
### ページ名

GAT-D APIは、GhcFunc\_getCurrentPage（）のAPIをユーザーに提供して、現在表示されているページを埋め込み側で取得します。ページはページ名の定義とともに返されるため、ページ名は標準のC式に適用する必要があります。詳細については、5.1.9も参照してください。

図56。ページ名

## ソース生成オプション

S6E2DHシリーズは、フラッシュ画像を内部フラッシュと外部フラッシュにプログラムする「USBダイレクトプログラマー」を提供します。これにより、GAT-Dの画像データを外部フラッシュに直接プログラムできます。GAT-Dは、「USBダイレクトプログラマー」に適用される画像データをフラッシュ画像ファイルに生成できます。「BinaryFileに変換」をチェックすると、GAT-Dはすべての画像データを.mhxファイルに生成します。

図57。ソース生成オプション

「ImageDataOffset（Hex）」は外部フラッシュの開始アドレスです。値はデフォルトで0xC0000000に設定されています。これは、グラフィックスドライバーが使用する外部フラッシュの開始アドレスです。

開始アドレスは、ダウンロード時に「USBダイレクトプログラマー」の設定と同じである必要があります。

図58。プログラマーの設定ビュー

# GAT-D API

次の図は、インターフェイスがどこにあるか、およびソフトウェアの概要を説明しています。

図59。APIの位置

組み込みデバイス

グラフィックドライバー

.

ハードウェア（S6E2DH）

その他のドライバー

ソースの生成

GAT-DAPIライブラリ

その他のライブラリ

その他のアプリケーション

グラフィックアプリケーション

その他の関数ソース

インターフェース

2\*

1\*

1 \*および2 \*：ユーザー関数ソースもハードウェアまたはグラフィックスドライバーにアクセスできますが、推奨されません

次の表は、GAT-DAPIライブラリとGeneratesourceのインターフェースを説明しています。

|  |  |  |
| --- | --- | --- |
| 番号 | 試作 | 注: |
| 1 | TGhcBool GhcProject\_load(TGhcU32 a\_mcuMemorySize, TGhcU32 a\_displayListSize) | 4.1.1を参照してください |
| 2 | TGhcBool GhcIrisInitialize(void) | 4.1.2を参照してください |
| 3 | TGhcBool GhcProject\_setStartupAction() | 4.1.3を参照してください |
| 4 | TGhcBool GhcProject\_RunAction() | 4.1.4を参照してください |
| 5 | TGhcBool GhcProject\_StopRuningAction(void) | 4.1.5を参照してください |
| 6 | void GhcProject\_sendEvent\_Ex(EGhcEventName eEventName) | 4.1.6を参照してください |
| 7 | TGhcInteger GhcFunc\_getButtonAreaByIndex(TGhcInteger a\_Index, TGhcS32\* a\_pBtnTpLft\_X, TGhcS32\* a\_pBtnTpLft\_Y, TGhcS32\* a\_pBtnBtmRt\_X, TGhcS32\* a\_pBtnBtmRt\_Y) | 4.1.7を参照してください |
| 8 | TGhcInteger GhcFunc\_getPageOfButtonByIndex(TGhcInteger a\_Index, TGhcU32\* a\_pPageId) | 4.1.8を参照してください |
| 9 | TGhcInteger GhcFunc\_getCurrentPage(TGhcU32\* a\_pCurrentPageId) | 4.1.9を参照してください |

## インターフェイスの説明

### GhcProject\_load

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | 生成されたソースにGAT-Dプロジェクトデータをロードする | | |
| 項目 | 分類 | 名 | 変更内容 |
| TGhcU32 1\* | a\_mcuMemorySize | 未使用、0に設定 |
| TGhcU32 | a\_displayListSize | 未使用、0に設定 |
| 戻り値 | TGhcBool 2\* | | 真：成功; False：失敗 |
| 注: | 1 \*：TGhcU32はunsignedintのtypedefです  2 \*：TGhcBoolはboolのtypedefです | | |

### GhcIrisInitialize

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | グラフィックドライバの初期化 | | |
| 項目 | 分類 | 名 | 変更内容 |
| Void | - | - |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: |  | | |

### GhcProject\_setStartupAction

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | 起動時に再生されるアクションを設定します | | |
| 項目 | 分類 | 名 | 変更内容 |
| Void | - | - |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: | 設定するアクションはGAT-Dツールで選択します。  詳しくはGAT-Dのユーザーズマニュアルをご覧ください。  GhcProject\_RunAction（）を呼び出して表示します。 | | |

### GhcProject\_RunAction

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | スタートアップアクションを再生する | | |
| 項目 | 分類 | 名 | 変更内容 |
| Void | - | - |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: | GhcProject\_setStartupAction（）の呼び出し後に呼び出す必要があります | | |

### GhcProject\_StopRuningAction

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | アクションが再生されている場合は、再生を停止します。 | | |
| 項目 | 分類 | 名 | 変更内容 |
| Void | - | - |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: | この呼び出しは、アクションが実行されているのとは異なる別のルーチンに配置する必要があります。 | | |

### GhcProject\_sendEvent\_Ex

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | 特定のイベントを参照するアクションを再生します | | |
| 項目 | 分類 | 名 | 変更内容 |
| EGhcEventName | eEventName | 与えられたイベント名1 \* |
| 戻り値 | Void | | - |
| 注: | アクションとイベント間の参照は、GAT-Dツールのイベントテーブルで設定されます。詳細については、4.1イベントも参照してください。  1 \*：イベント名の前に「eGEN\_」を追加する必要があります。 | | |

### GhcFunc\_getButtonAreaByIndex

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | 指定されたボタン名でボタンの面積を取得します | | |
| 項目 | 分類 | 名 | 変更内容 |
| TGhcInteger | a\_Index | 与えられたボタン名2 \* |
| TGhcS32\* 1\* | a\_pBtnTpLft\_X | 左上隅のx座標を保存するアドレス |
| TGhcS32\* | a\_pBtnTpLft\_Y | 左上隅のy座標を保存するアドレス |
| TGhcS32\* | a\_pBtnBtmRt\_X | 右下隅のx座標を保存するアドレス |
| TGhcS32\* | a\_pBtnBtmRt\_Y | 右下隅のy座標を保存するアドレス |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: | 1 \*：TGhcS32はsignedintのtypedefです。  2 \*：オブジェクトのボタンオプションを設定するときに入力した名前と同じ | | |

### GhcFunc\_ getPageOfButtonByIndex

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | 指定されたボタン名でボタンを押すページIDを取得します | | |
| 項目 | 分類 | 名 | 変更内容 |
| TGhcInteger | a\_Index | 与えられたボタン名1 \* |
| TGhcU32 | a\_pPageId | ページ2のIDを保存するアドレス\* |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: | 1 \*：オブジェクトのボタンオプションを設定するときに入力した名前と同じ  2 \*：IDはGAT-Dツールのページ名で定義されています。 | | |

### GhcFunc\_getCurrentPage

|  |  |  |  |
| --- | --- | --- | --- |
| 端子名 | LCDに表示されている現在のページIDを取得します | | |
| 項目 | 分類 | 名 | 変更内容 |
| TGhcU32 | a\_pCurrentPageId | ページ1のIDを保存するアドレス\* |
| 戻り値 | TGhcBool | | 真：成功; False：失敗 |
| 注: | 1 \*：IDはGAT-Dツールのページ名で定義されています。 | | |

## ソースに参加する

次の図は、インターフェイスを使用してユーザーのソースに参加する方法を説明しています。

図60。ユーザーのソースに参加する方法

組み込みソフトウェア

GhcProject\_load

起動

Calling

メインルーチン

Calling

Calling

Calling

Graphics

GhcIrisInitialize

GhcProject\_setStartupAction

GhcProject\_RunAction

GhcProject\_sendEvent\_Ex

GhcFunc\_getCurrentPage

GhcFunc\_getButtonAreaByIndex

Calling

Event ?

Other routine

Start 2\*

Calling

Loop buttons

Calling

Throw Event

有り

No

1\*

1\*

GhcProject\_StopRuningAction

Calling

Push?

有り

No

3\*

3\*

1 \*：呼び出しは、アクションの再生が完了するまで戻らないか、無限ループアクションの場合は戻らない。この場合、唯一のメソッドは、割り込みルーチンでGhcproject\_StopRuningAction（）を呼び出すことです。

2 \*：タッチパネル検出ルーチンでの動作と想定します。

3 \*：これらのソースはGAT-Dユーザーによって実装されます。

# プログラム

## デバッグモード

デバッグ中に、ハイパーバスベースの外部フラッシュを使用する場合、内部フラッシュと外部フラッシュをプログラムする方法は2つあります。HSSPIベースの外部フラッシュを使用する場合は、方法Aになります。

方法A：

図‑1デバッグのプログラム方法A

PC環境

GAT-Dツール

IAR EW IDE

生成されたソース

その他

フラッシュイメージ（.mhx）

組み込みデバイス

外部フラッシュ

*プログラム*

S6E2DH内部フラッシュ

*ビルド/プログラム*

S6E2DHPCプログラマーを使用する

デバッガーを使用する

方法B：

図‑2デバッグのプログラム方法B

PC環境

GAT-Dツール

IAR EW IDE

生成されたソース

その他

フラッシュイメージ（.mhx）

組み込みデバイス

S6E2DH内部フラッシュ

*ビルド/プログラム*

デバッガーを使用する

1. 画像はCソースに生成され、外部フラッシュにもプログラムされます。

外部フラッシュ

binファイルを生成するかどうかを示すGAT-Dツールをツールで選択できます。詳しくはGAT-Dのユーザーズマニュアルをご覧ください。

## 製品モード

次の図は、開発後に製品をプログラムする方法を示しています。

図3。製品のプログラム

PC環境

GAT-Dツール

IAR EW IDE

生成されたソース

その他

フラッシュイメージ（.mhx）

組み込みデバイス

外部フラッシュ

*プログラム*

S6E2DH内部フラッシュ

*ビルド/プログラム*

S6E2DHPCプログラマーを使用する

S6E2DHPCプログラマーを使用する

S6E2DH「USBダイレクトプログラマー」は、内部フラッシュと外部フラッシュ（HSSPIおよびハイパーバス）の両方をサポートします。詳しくは「USBダイレクトプログラマ」のユーザーズマニュアルをご覧ください。

ハイパーバスベースのフラッシュのプログラマー：USBDIRECT\_V01L16c03\_for\_STK。S6E2D-2DGraphicsDriver-v1.00.798-IAR.zipの00\_s6e2dh\_demeter\_sw\_framework \ progtoolにあります。

# サンプルプロジェクトの説明

## 建設

サンプルプロジェクトは03にあります。サンプル\ 02。IAR\ 02. S6E2DH\_Demo\_Washmachine \ S6E2DH\_Demo\_Washmachine Project.zipですが、コンパイルしないとコンパイルは成功しません。

04\_S6E2D-2DGraphicsDriver-v1.00.798-IAR.zipの抽出パスのサンプル。

次の図は、サンプルプロジェクトの構築を示しています。

図4。サンプルプロジェクトの構築

PC環境

GAT-Dツール

IAR EW IDE

生成されたソース

(1\*)

GAT-D API.a

.a

GAT-DAPIライブラリ

グラフィックドライバー

basic\_graphics.a

util.a

その他の関数ソース

ユーザー編集

サンプルプロジェクト

コンパイル

リンク

source.a

1 \*：外部フラッシュに画像を含めるか含めないかはOKです

## 機能概要

パス内：03。サンプル\ 01。GAT-Dツールは、パス：03でGAT-DおよびS6E2DH\_Demo\_WashmachineProject.zipを使用したサンプルです。サンプル\ 02。IAR\02. S6E2DH\_Demo\_Washmachineは、ツールサンプルによって生成された組み込みソフトウェアサンプル使用生成ソースです。

注：デバッグ方法は方法Aを使用しています（ソース生成中にBinaryFileに変換するをチェック）

これらのサンプルの要約機能は次のとおりです。

図5。サンプル関数

S6E2DH洗濯機

スパンションロゴ

Action1最終フレーム

設定

Action2最終フレーム

選択肢

Action3最終フレーム

洗浄

アクション4の繰り返し

次へ/アクション2

Next/ Action3

Back/Action2

Back/Action1

Next/Action4

Next/Action1

Startup/Action1

Startup/Action1: Event/LCD Display Action

保持

LCDディスプレイ

選択肢

Action3最終フレーム

2\*:

1\*:

1\*:

2\*:

スパンションロゴ：次のイベントはタッチスパンションロゴです

設定：戻るイベントはLCDの「戻る」ボタンを押すことです

次のイベントはLCDの「次へ」ボタンを押すことです

選択：戻るイベントは、LCDの「戻る」ボタンを押すことです

次のイベントはLCDの「次へ」ボタンを押すことです

洗濯：次のイベントはLCDの「停止」ボタンを押すことです

改訂履歴

ドキュメントタイトル：AN204388-FM4 S6E2DH（GAT-D）入門

＜＜★Document Number: 002-04388

| 版 | Engineering Change Notification (技術変更届) | Orig。変化の | 発行日 | 変更内容 |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |