

**S1C31 Family**  
**emWin**  
**アプリケーションノート**

#### 評価ボード・キット、開発ツールご使用上の注意事項

---

1. 本評価ボード・キット、開発ツールは、お客様での技術的評価、動作の確認および開発のみに用いられることを想定し設計されています。それらの技術評価・開発等の目的以外には使用しないで下さい。本品は、完成品に対する設計品質に適合していません。
2. 本評価ボード・キット、開発ツールは、電子エンジニア向けであり、消費者向け製品ではありません。お客様において、適切な使用と安全に配慮願います。弊社は、本品を用いることで発生する損害や火災に対し、いかなる責も負いかねます。通常の使用においても、異常がある場合は使用を中止して下さい。
3. 本評価ボード・キット、開発ツールに用いられる部品は、予告無く変更されることがあります。

本資料のご使用につきましては、次の点にご留意願います。  
本資料の内容については、予告無く変更することがあります。

---

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。
2. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これらに起因する第三者の知的財産権およびその他の権利侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の知的財産権およびその他の権利の実施権の許諾を行うものではありません。
3. 特性値の数値の大小は、数直線上の大小関係で表しています。
4. 製品および弊社が提供する技術を輸出等するにあたっては「外国為替および外国貿易法」を遵守し、当該法令の定める手続きが必要です。大量破壊兵器の開発等およびその他の軍事用途に使用する目的をもって製品および弊社が提供する技術を費消、再販または輸出等しないでください。
5. 本資料に掲載されている製品は、生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。
6. 本資料に掲載されている会社名、商品名は、各社の商標または登録商標です。

# 目次

1. 概要.....	2
1.1 動作環境.....	2
2. emWin のセットアップ.....	3
2.1 emWin パッケージの構成.....	3
2.2 ターゲットハードウェアと開発環境の準備.....	4
2.3 既存のサンプルプロジェクトをビルドする.....	4
3. S1C31 Family の emWin 設定.....	6
3.1 LCDConf.c / LCDConf.h.....	6
3.1.1 ディスプレイパネルタイプ.....	6
3.1.2 色変換.....	6
3.1.3 ディスプレイドライバ.....	6
3.2 GUIConf.c / GUIConf.h.....	6
3.3 Epson_C31.c / Epson_C31.h.....	7
4. サンプルソフトウェア.....	8
4.1 c31_demo1.....	8
4.2 c31_demo2.....	8
4.3 c31_demo3.....	8
4.4 c31_basic_performance.....	8
4.5 c31_sample.....	8
5. プロジェクトの作成.....	9
5.1 既存のプロジェクトをコピーする.....	9
5.2 新規にプロジェクトを作成する.....	10
6. S1C31 Family で emWin を使用する場合の補足情報.....	23
6.1 ディスプレイのアップデート.....	23
6.2 2D ライブラリ.....	23
6.3 ウィジェット.....	23
6.4 GUI-Builder.....	25
6.5 ビットマップの作成.....	25
6.6 既存の emWin サンプルコードの使用.....	30
改訂履歴表.....	31

## 1. 概要

---

### 1. 概要

本ドキュメントは S1C31 Family 対応の emWin の使い方とサンプルソフトの動作について記載しています。emWin は、Segger Microcontroller 社が所有しているグラフィックソフトウェアと GUI（グラフィカルユーザーインターフェイス）製品です。

本パッケージに含まれている emWin は、S1C31D01 用に調整されています。

#### 1.1 動作環境

S1C31D01 で emWin プログラムを作成して実行するには、以下の環境が必要です。

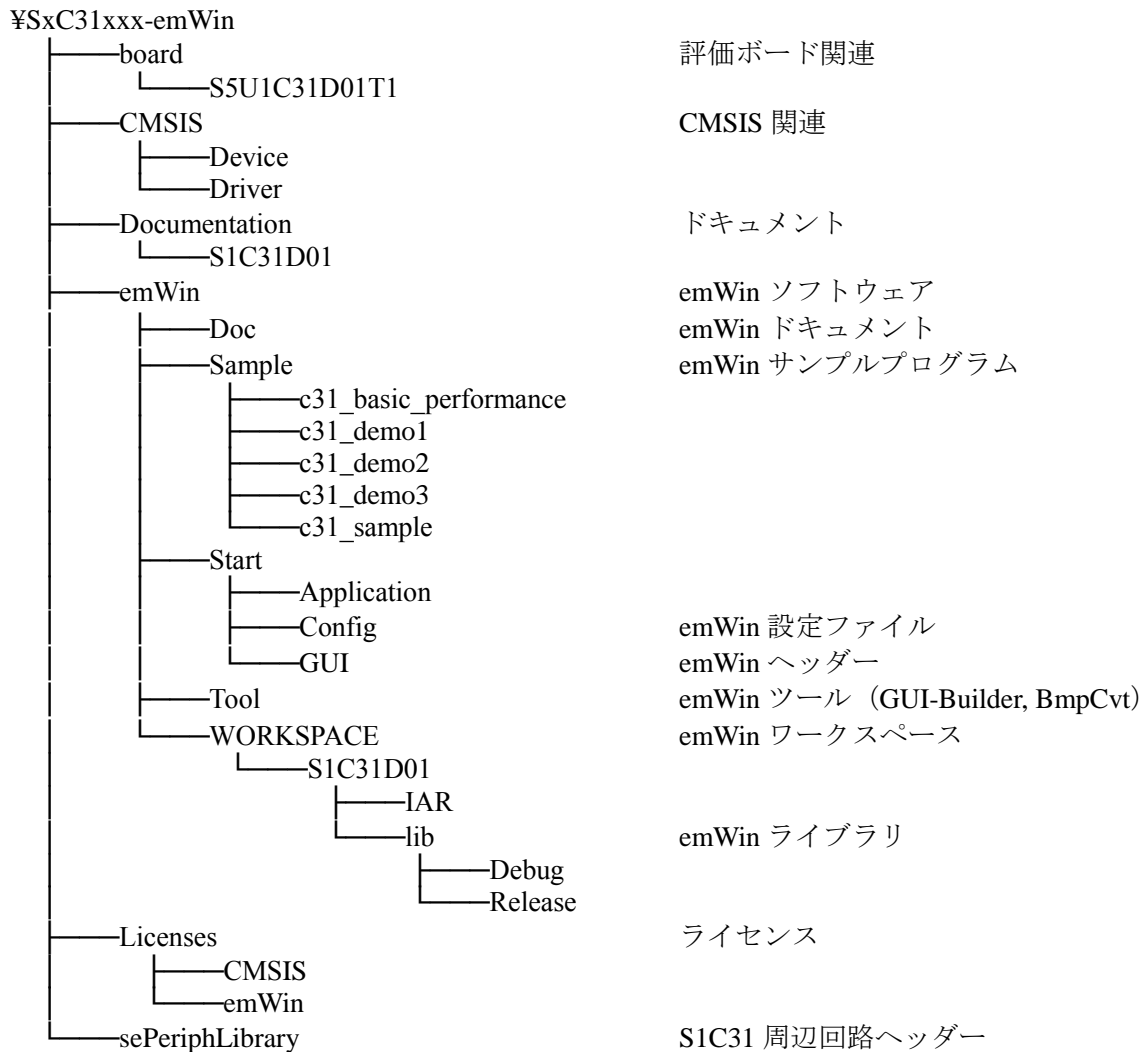
- S1C31D01 評価ボード  
評価ボードの入手方法については、最寄りのエプソンセールスオフィスにお問い合わせください。
- ハードウェアデバッガインターフェース  
Segger J-Link Plus が推奨されます。
- ARM 用 IAR Embedded Workbench  
本パッケージは IAR Ver.7.80 で確認しています。
- S1C31D01 用 emWin パッケージ（本パッケージ）  
本パッケージの emWin は Ver.5.40g です。

## 2. emWin のセットアップ

本章では、弊社より提供する emWin パッケージと評価ボードを使用したソフトウェアのビルド・デバッグ方法について説明します。

### 2.1 emWin パッケージの構成

emWin パッケージのディレクトリ構成は以下のようになっています。



## 2. emWin のセットアップ

---

### 2.2 ターゲットハードウェアと開発環境の準備

IAR 開発環境のインストールと評価ボードとの接続は以下のドキュメントを参照して下さい。

- S1C31 Family ソフトウェア開発セットアップガイド
- S1C31D01 周辺回路サンプルソフトウェアマニュアル
- S5U1C31D01 マニュアル

### 2.3 既存のサンプルプロジェクトをビルドする

ハードウェアとソフトウェアをインストールして設定したら、サンプルプロジェクトをビルドして S1C31D01 にロードし、システムが正しく機能していることを確認してください。ここでは、c31\_demo1 プロジェクトのビルドについて説明します。

#### 1) emWin ワークスペースを開く

IAR EWARM IDE を起動し、[ファイル]>[開く]>[ワークスペース]をクリックして表示されるファイル選択ダイアログで、emWin パッケージの解凍先フォルダ内にある以下のファイルを選択します。

¥SxC31xx-emWin¥emWin¥WORKSPACE¥S1C31D01¥IAR¥emWin.eww

#### 2) サンプルプロジェクトをビルドする...

“c31\_demo1” プロジェクトを右クリックし、[アクティブに設定]を選択してアクティブにします。次に、プロジェクトを右クリックして[全てを再ビルド]を選択してビルドします。

プロジェクトを選択した後、F8 キーを押して “Debug – All” を選択して “全てをビルド” ボタンをクリックしてプロジェクトをビルドする方法もあります。

メッセージウィンドウの[ビルド]タブにエラーがないことを確認します。

#### 3) ファイルを S1C31D01 にダウンロード

“Ctrl+D” を押すか、[プロジェクト]>[ダウンロードしてデバッグ]を選択します。

ダウンロードが完了したら、F5 キーを押してプログラムを開始します。

(注) 初めて S1C31D01 にプロジェクトをダウンロードすると、次のダイアログが表示されます。

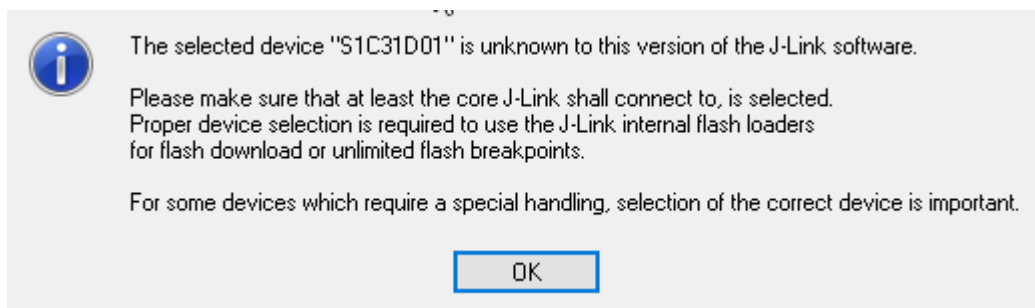


図 2.1 MCU 未選択警告ダイアログ

## 2. emWin のセットアップ

OK ボタンをクリックし、次のダイアログで “Cortex-M0 +” を選択します。

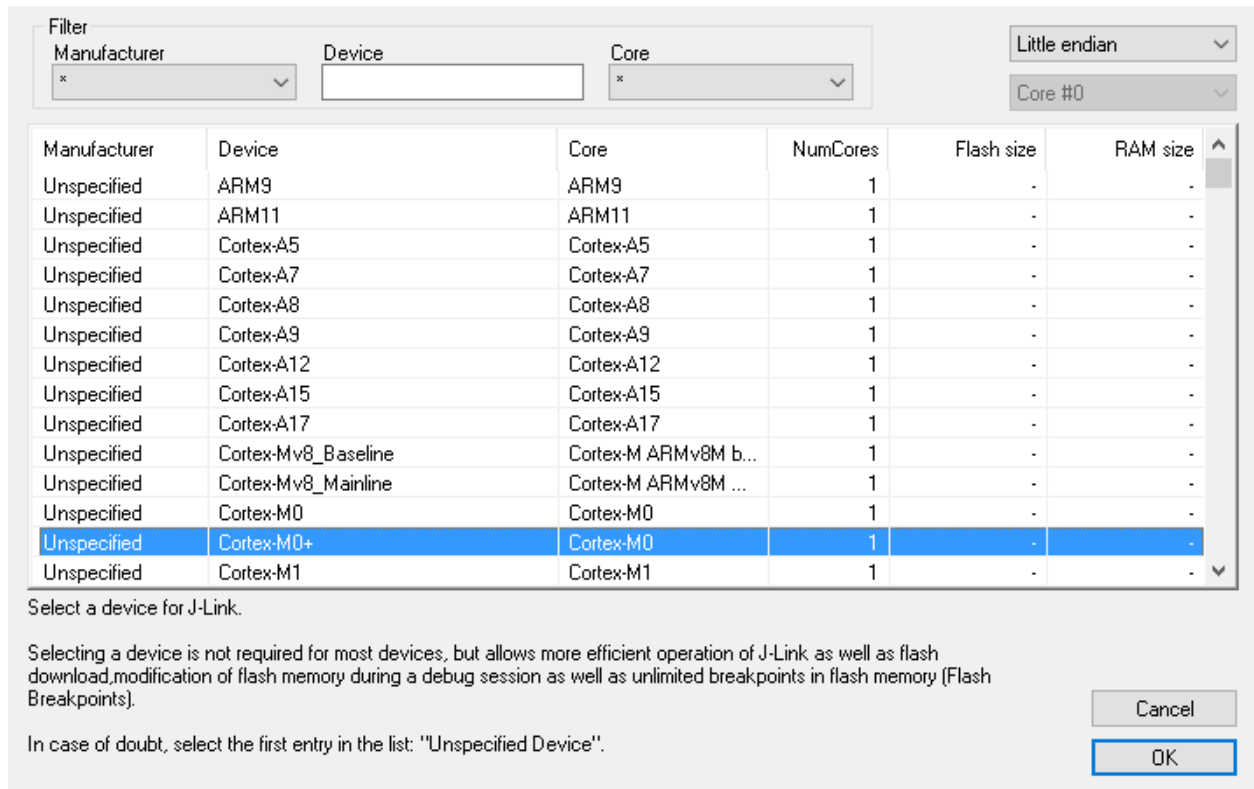


図 2.2 MCU 選択ダイアログ

- 4) ディスプレイに表示されるデモプログラムをチェックします。  
ディスプレイの表示は、選択したデモプログラムによって異なります。

## 3. S1C31 Family の emWin 設定

---

### 3. S1C31 Family の emWin 設定

本章では S1C31 Family で emWin を使用する場合の設定ファイルについて説明します。

すべての emWin プロジェクトには、“config” プロジェクトグループが含まれています。このグループには、添付されたプロジェクトを構成するか、または emWin ハードウェアインターフェースのサポートを提供する以下のファイルが含まれています。

#### 3.1 LCDConf.c / LCDConf.h

ファイル LCDConf.c は、表示パネルの種類、色変換方法、および表示ドライバが選択されている場所です。

##### 3.1.1 ディスプレイパネルタイプ

現在、LCDConf.c には 7 つのディスプレイパネルタイプが定義されています。パネルタイプを選択すると、パネルのプロパティ（幅、高さ、色の濃さ）が設定され、適切なパネル初期化シーケンスが設定されます。

現在サポートされているパネルは以下の通りです。

LPM010R030B	- 180x180, 6 ビットカラー, 0.99 インチ, 円形
LPM012M134B	- 240x240, 6 ビットカラー, 1.19 インチ, 円形
LPM014T262C	- 205x148, 6 ビットカラー, 1.39 インチ, 長方形
LPM027M128A	- 400x240, SPI, 3 ビットカラー, 2.7 インチ, 長方形
LS010B7DH01	- 128x128, SPI, 1 ビット BW, 1.0 インチ, 円形
LS012B7DH02	- 240x240, SPI, 1 ビット BW, 1.2 インチ, 円形
LS013B4DN04	- 96x96, SPI, 1 ビット BW, 1.3 インチ, 正方形
LS013B7DH06	- 128x128, SPI, 3 ビットカラー, 1.33 インチ, 正方形

リストからパネル定義を 1 つだけ選択し、コメントを外しておきます。

##### 3.1.2 色変換

選択できる色変換タイプは以下の通りです。

COLOR_CONVERSION GUICC_2222	- 6 ビットカラー (RrGgBb) と 2 ビットアルファをサポート
COLOR_CONVERSION GUICC_111	- 1 ビットカラー (モノクロ) をサポート

色変換タイプはパネル定義に連動して選択されますので変更する必要はありません。

##### 3.1.3 ディスプレイドライバ

ディスプレイドライバがここで選択されます。現時点では、以下の一つしか定義されていないので変更する必要はありません。

GUIDRV\_EPSON\_C31

#### 3.2 GUIConf.c / GUIConf.h

これらのファイルで使用される設定については、emWin のマニュアルを参照してください。



### 3.3 Epson\_C31.c / Epson\_C31.h

これらのファイルは、GUIDRV\_EPSON\_MCU が LCDConf.c で定義されていて、構成設定を含まない場合に使用されます。

## 4. サンプルソフトウェア

---

### 4. サンプルソフトウェア

本章では、emWin パッケージに含まれているサンプルソフトウェアについて説明します。

#### 4.1 c31\_demo1

このサンプルプログラムでは、テキスト描画やポリゴンの使用、アイコンやチェックボックスの基本的なウィジェットの使用などの 2D 描画を表示します。

#### 4.2 c31\_demo2

このサンプルプログラムは、emWin ウィジェットを使用する方法を示しています。表示更新のために emWin ウィジェットをサブクラス化する方法の例については、ソースコードを参照してください。

#### 4.3 c31\_demo3

このサンプルプログラムは、bin2c.exe と bmpcvt.exe ツールを使用して作成されたさまざまなイメージを表示します。

#### 4.4 c31\_basic\_performance

このサンプルプログラムは、emWin サンプルプログラムの BASIC\_Performance プロジェクトの移植で、素数の数を計算する秒数(ms)を表示します。

#### 4.5 c31\_sample

このサンプルプログラムはエプソンのロゴとテキストで “Welcome” を表示します。  
新規プロジェクトを作成する場合に、この最小限のプロジェクトを使用してください。

---

## 5. プロジェクトの作成

本章では、新規にプロジェクトを作成する手順について説明します。プロジェクトを作成するには、既存のプロジェクトをコピーするやり方とワークスペースとプロジェクトを最初から作成する方法があります。

### 5.1 既存のプロジェクトをコピーする

既存のプロジェクトをコピーすると、新しいプロジェクトを簡単に作成できます。"c31\_sample"という名前のプロジェクトは、この目的のために最低限必要な emWin プロジェクトです。以下の手順に従って、“c31\_sample” から新規プロジェクトを作成します。

- 1) サンプルプロジェクトディレクトリを新しいディレクトリにコピーします。

ファイルエクスプローラで、“¥SxC31xx-emWin¥emWin¥Sample¥” に移動します。

プロジェクトディレクトリ “c31\_sample” を右クリックし、コピーを選択します。

“Sample” ディレクトリを右クリックし、貼り付けを選択します。

最後に、ディレクトリ "c31\_sample – コピー"の名前を新しいプロジェクト名（例：myproj）に変更します。

新しいプロジェクトディレクトリにある以下のファイルとディレクトリは不要ですから削除して下さい。

```
c31_sample.dep
c31_sample.ewd
Debug
settings
```

- 2) myproj ディレクトリ内のファイルの名前を新しいプロジェクト名に変更します。

ファイルエクスプローラで、新しく作成したディレクトリ

“¥SxC31xxx-emWin¥emWin¥Sample¥ myproj” に移動します。

以下のファイルの名前を変更します。

```
c31_sample.ewp ⇒ myproj.ewp
c31_sample.c ⇒ myproj.c
```

- 3) プロジェクトの既存の参照を変更します。

テキストエディタで“myproj.ewp”を開き、以下のように新しいプロジェクト名に変更して下さい。

```
c31_sample ⇒ myproj
```

- 4) IAR Embedded Workbench を起動し、ワークスペース “¥SxC31xxx-emWin¥emWin¥WORKSPACE ¥S1C31D01¥IAR¥emWin.eww” を開きます。

## 5. プロジェクトの作成

---

- 5) プロジェクト “myproj” プロジェクトファイルを IAR にインポートします。  
[プロジェクト]>[既存プロジェクトの追加]を選択します。  
プロジェクト “¥SxC31xxx-emWin¥emWin¥sample¥myproj¥myproj.ewp” を選択します。  
[開く]ボタンをクリックしてプロジェクトを開きます。
- 6) プロジェクトをビルドします。  
ワークスペースウィンドウで、“myproj – Debug” を右クリックし、[すべてを再ビルド]を選択します。
- 7) デバッガの設定を行います。  
“myproj” プロジェクトを右クリックして[オプション]を選択します。  
オプション設定ウィンドウのカテゴリの中から[デバッガ]を選択します。  
設定タブのドライバを “J-Lin/ J-Trace” に設定します。
- 8) プロジェクトを実行します。  
キーボードから、“Ctrl+D” を入力してプロジェクトをダウンロードし、デバッグを開始します。

(注) 新しいプロジェクトと、ワークスペースで実行されていないプロジェクトでは、「2.3 既存のサンプルプロジェクトをビルドする」で説明したMCU選択ダイアログシーケンスが発生します。

- 9) プロジェクトを変更します。  
プロジェクトビルドを検証した後、ロードして実行することで独自の要件に合わせてカスタマイズを続けることができます。

### 5.2 新規にプロジェクトを作成する

次の手順では、最初からワークスペースを作成してプロジェクトを追加する方法を示します。この例ではワークスペースに “emExample”、プロジェクトに “emEx1” という名前を付けます。開始する前に、ワークスペースファイルを保存する “emExample” という名前のディレクトリとプロジェクトファイルを保存する “emEx1” という名前のディレクトリを作成する必要があります。プロジェクトディレクトリ “emEx1” は “emExample” のサブディレクトリになります。

- 1) IAR IDE で開いているワークスペースをすべて閉じます。
- 2) 新しいワークスペースを作成して保存します。  
メニューから[ファイル]>[新規作成]>[ワークスペース]を選択します。  
メニューから[ファイル]>[ワークスペースの名前を付けて保存]を選択します。  
[ワークスペースの名前を付けて保存]ダイアログボックスで、新しく作成した “emExample” ディレクトリに移動し、ワークスペース名 “emExample” を入力して[保存]ボタンをクリックします。

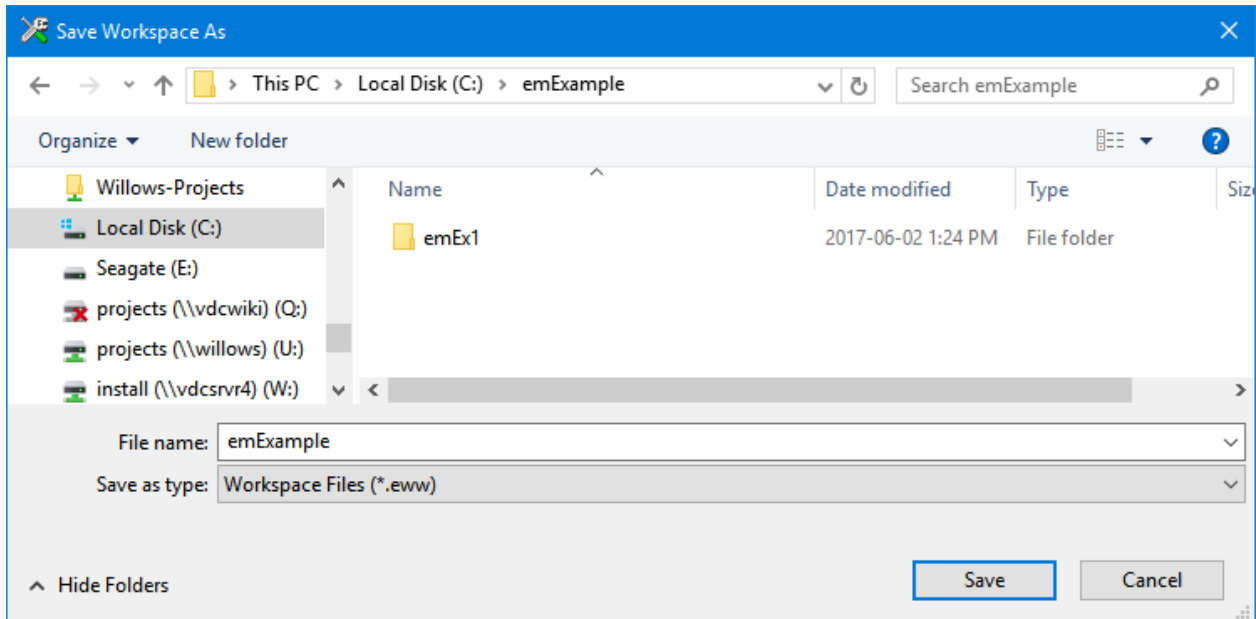


図 5.1 新規ワークスペースの保存

- 3) 新しく作成したワークスペースにプロジェクトを追加します。  
メニューから[プロジェクト]>[新規プロジェクトの作成]を選択します。  
[空のプロジェクト]テンプレートを選択し、[OK]をクリックします。

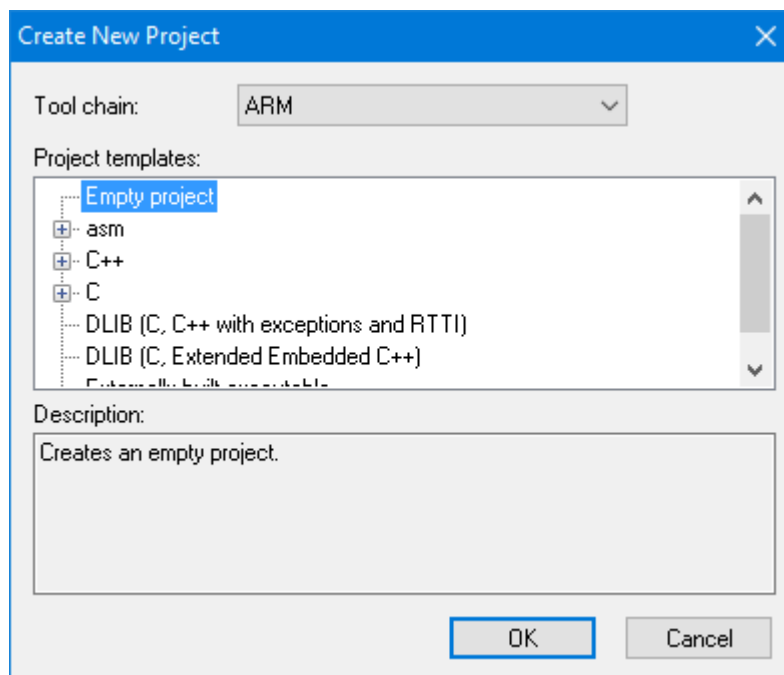


図 5.2 空プロジェクトの作成

## 5. プロジェクトの作成

---

[名前を付けて保存]ダイアログボックスで、“emExample¥emEx1”ディレクトリに移動します  
ファイル名に“emEx1”と入力します。

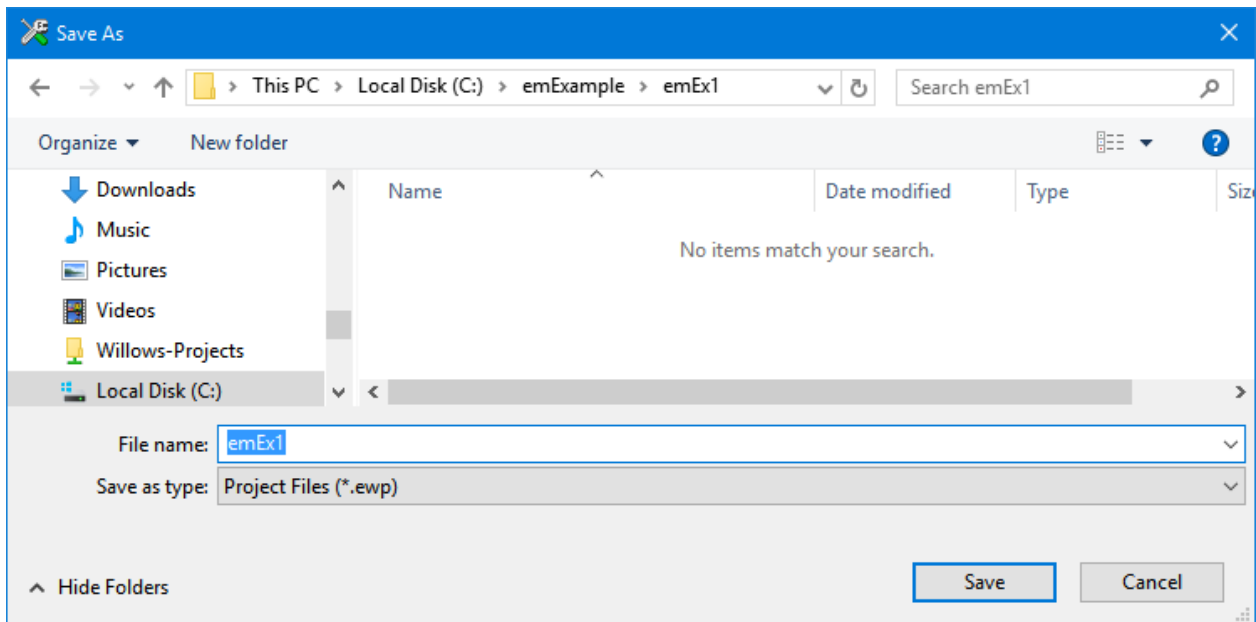


図 5.3 新規プロジェクトの作成

## 4) 新しいプロジェクトの設定を行います。

プロジェクトの設定を開始するには、新しいプロジェクトを右クリックし、[オプション]を選択します。S1C31D01 プログラムに必要な設定手順は次のとおりです。

## 【一般オプション】 カテゴリ

[ターゲット]タブを選択します。

デバイスで Epson S1C31D01 を選択します。

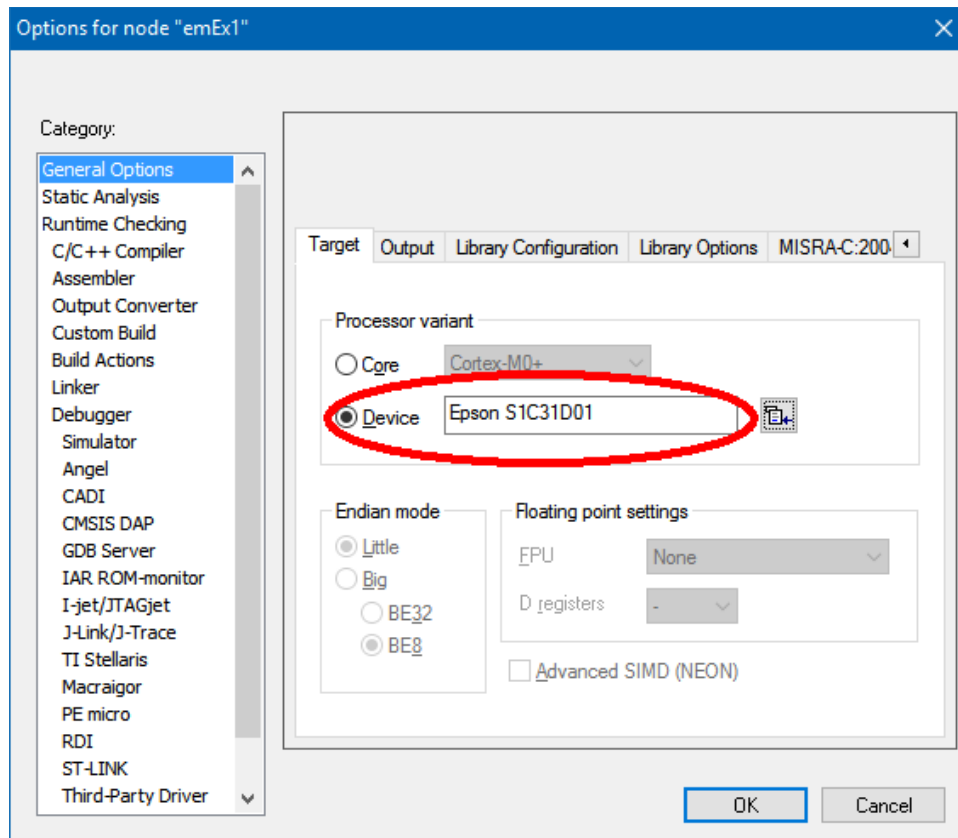


図 5.4 一般オプションの設定

## 5. プロジェクトの作成

### 【C/C++コンパイラ】 カテゴリ

[プリプロセッサ]タブで、必要なディレクトリが含まれていることを確認します。

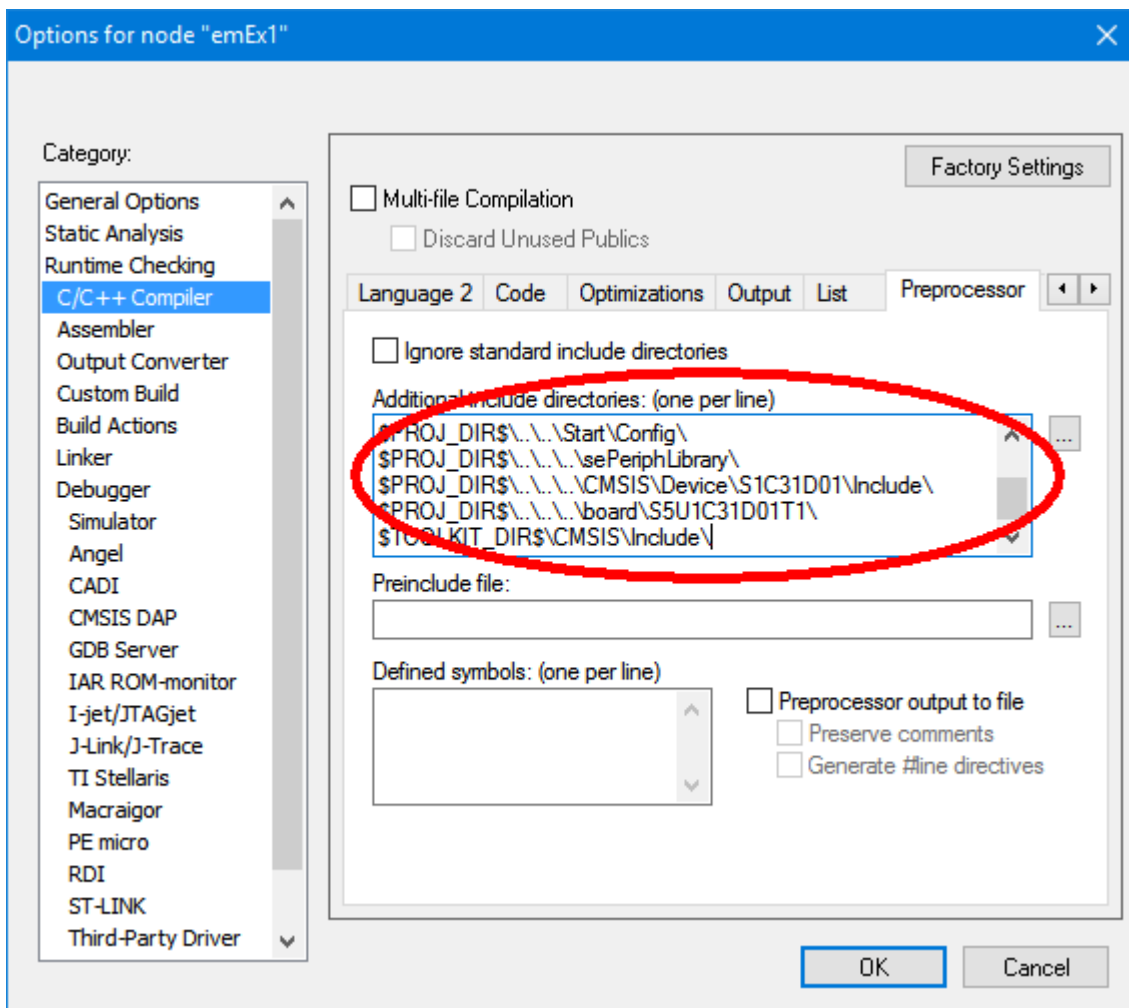


図 5.5 C/C++コンパイラオプションの設定

サンプルプログラムで使用されるインクルードされたディレクトリのリストは次のとおりです。

```
$PROJ_DIR$\\..\\..\\Start\\GUI\\Core\\
$PROJ_DIR$\\..\\..\\Start\\GUI\\Widget\\
$PROJ_DIR$\\..\\..\\Start\\GUI\\WM\\
$PROJ_DIR$\\..\\..\\Start\\Config\\
$PROJ_DIR$\\..\\..\\sePeriphLibrary\\
$PROJ_DIR$\\..\\..\\CMSIS\\Device\\S1C31D01\\Include\\
$PROJ_DIR$\\..\\..\\board\\S5U1C31D01T1\\
$TOOLKIT_DIR$\\CMSIS\\include\\
```



【リンカ】 カテゴリ

【設定】 タブで、デフォルトのリンカ制御ファイルを使用します。

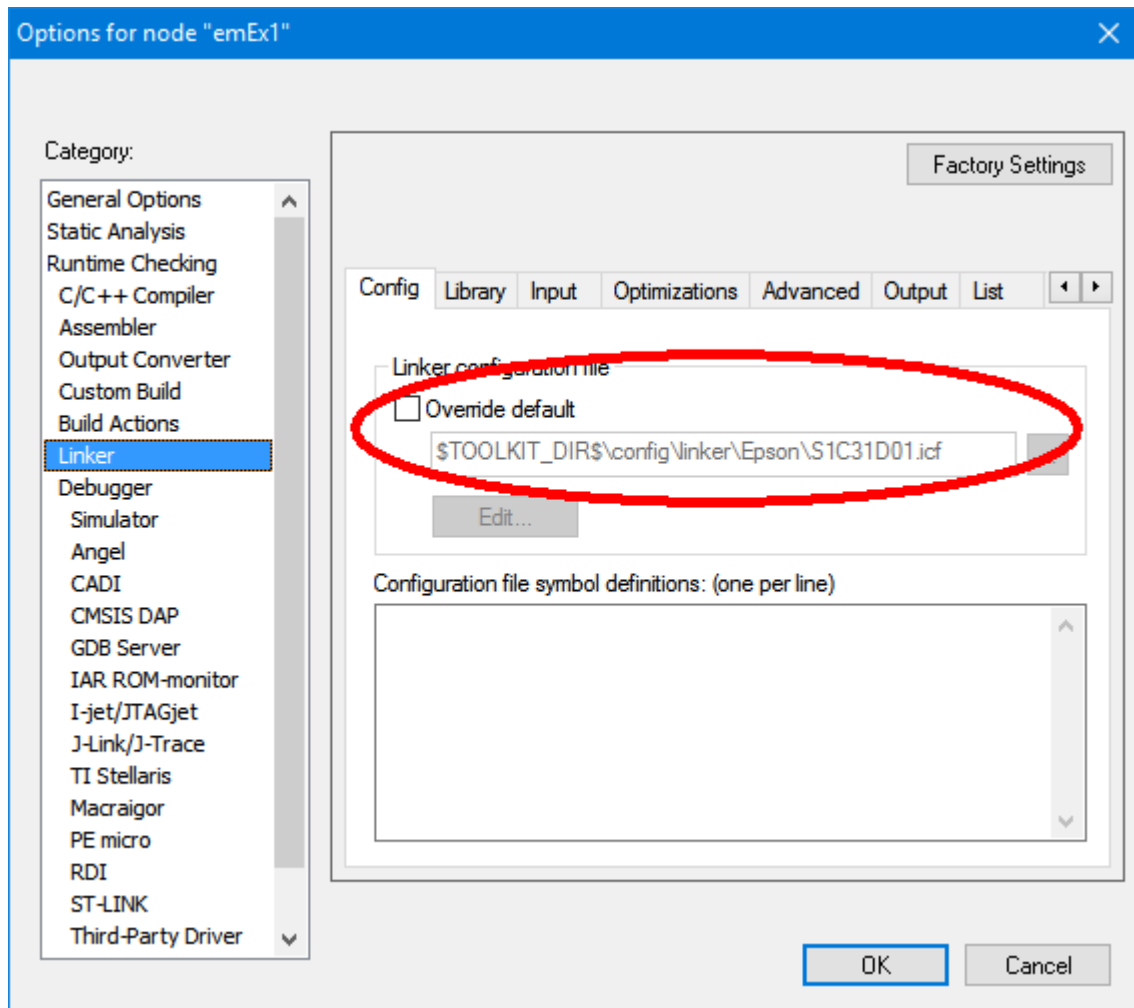


図 5.6 リンカオプション（設定）の設定

## 5. プロジェクトの作成

【ライブラリ】タブで、このビルドに使用するライブラリを指定します。

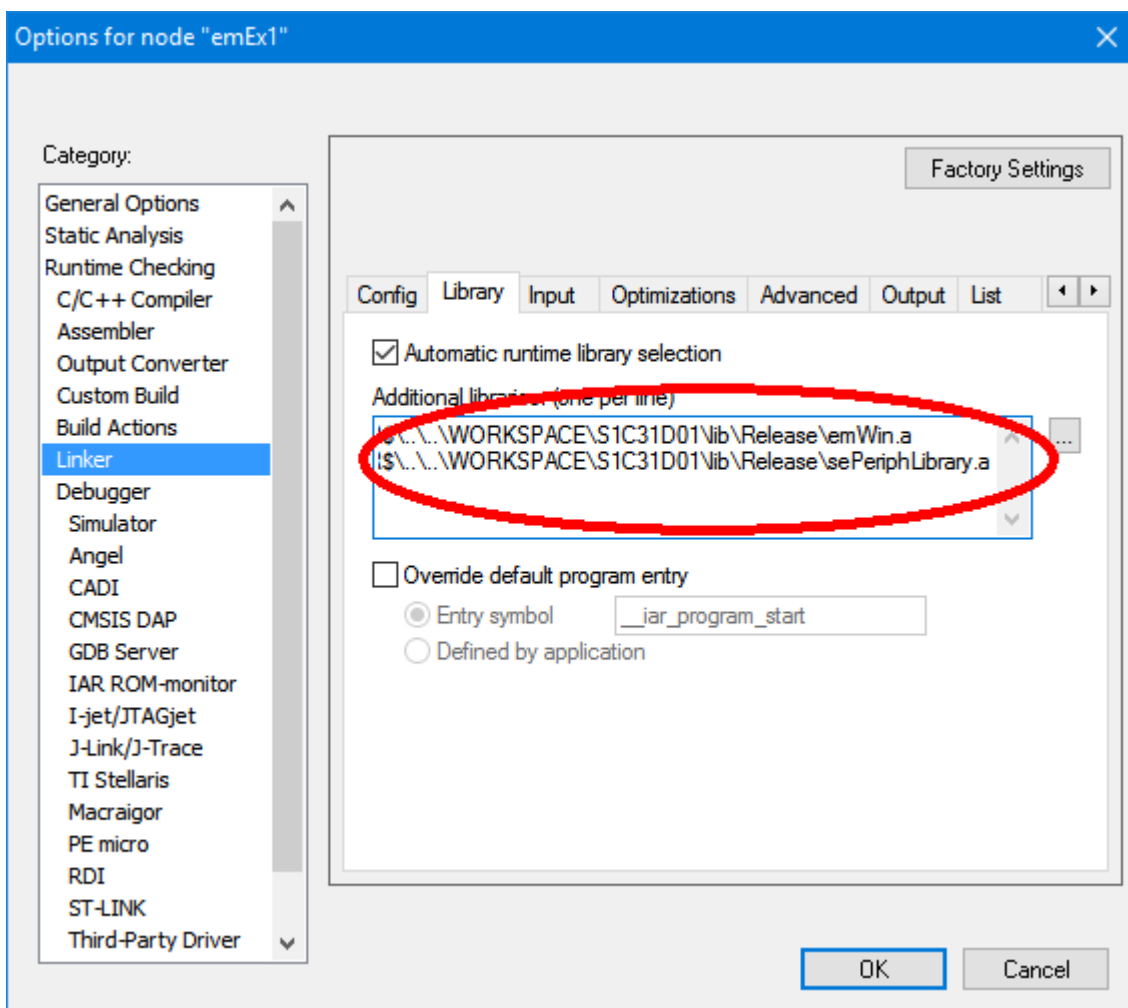


図 5.7 リンカオプション（ライブラリ）の設定

サンプルプログラムで使用されるライブラリは以下の通りです。

```
$PROJ_DIR$\\..\\..\\WORKSPACE\\S1C31D01\\lib\\Release\\emWin.a
```

```
$PROJ_DIR$\\..\\..\\WORKSPACE\\S1C31D01\\lib\\Release\\sePeriphLibrary.a
```

または、デバッグバージョン用のライブラリを指定する場合は以下の通りです。

```
$PROJ_DIR$\\..\\..\\WORKSPACE\\S1C31D01\\lib\\Debug\\emWin.a
```

```
$PROJ_DIR$\\..\\..\\WORKSPACE\\S1C31D01\\lib\\Debug\\sePeriphLibrary.a
```

【デバッガ】 カテゴリ

【設定】 タブのドライバで J-Link/J-Trace を選択します。

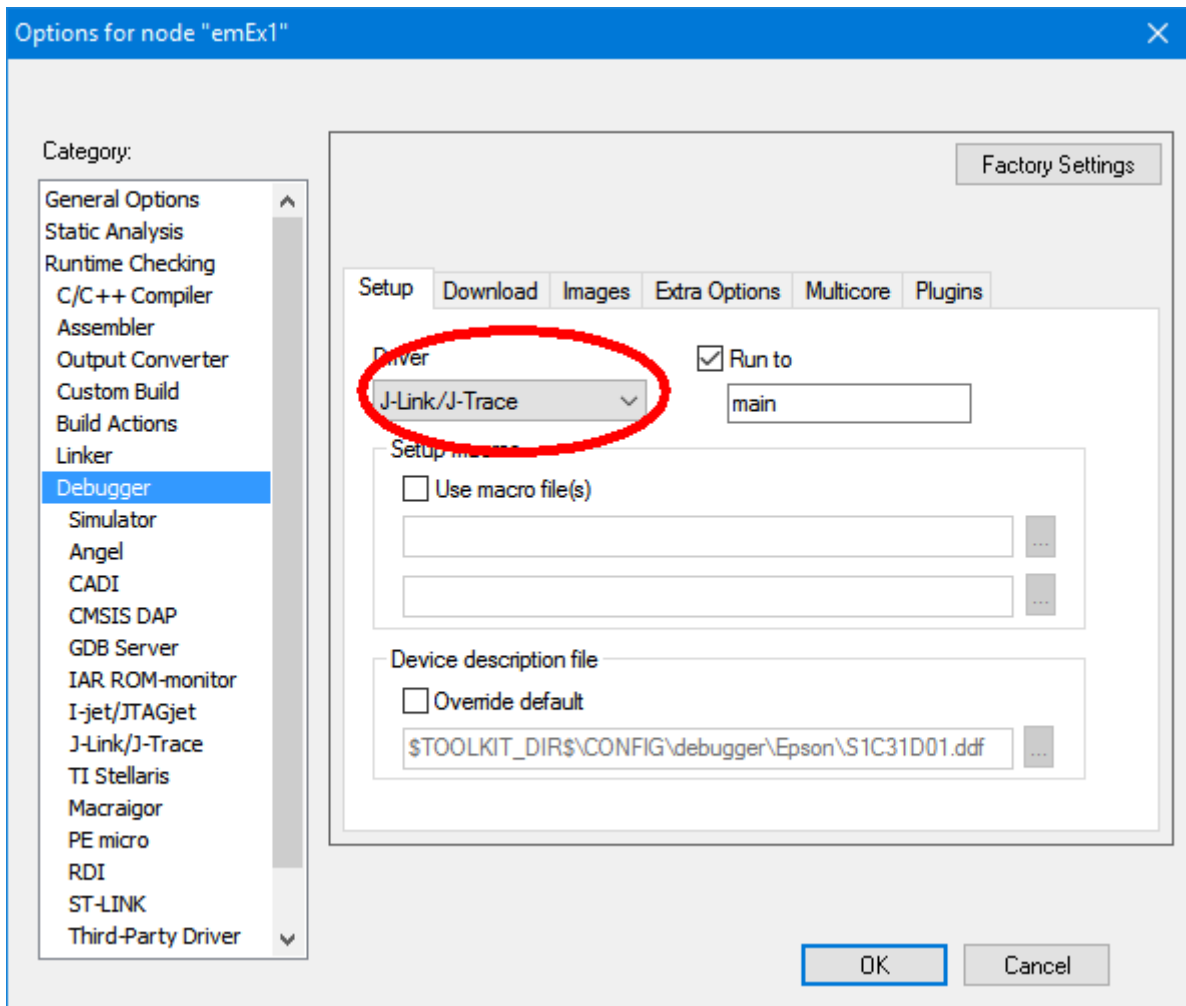


図 5.8 デバッガ（設定）の設定

## 5. プロジェクトの作成

【ダウンロード】タブでフラッシュローダの設定を行います。

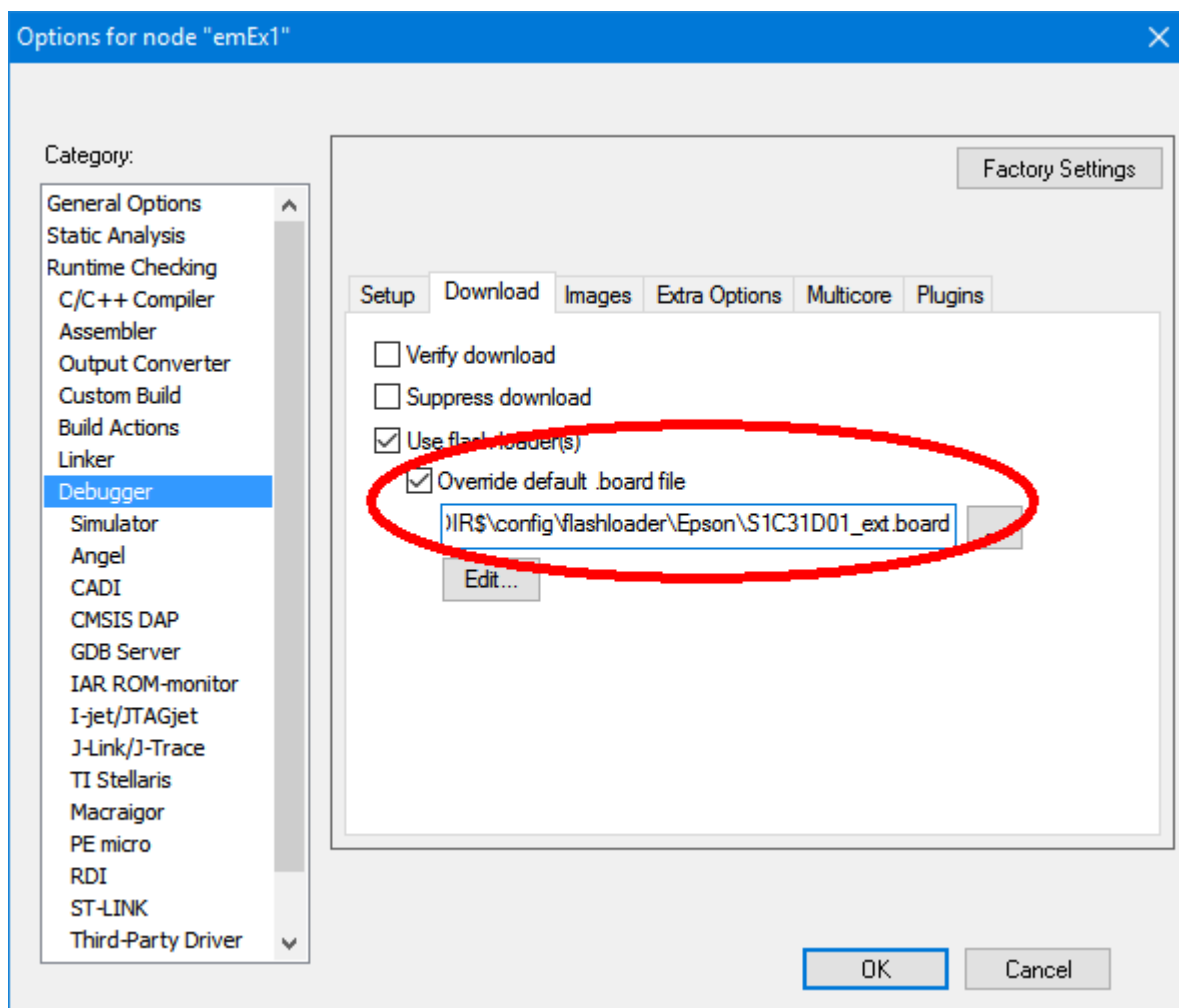


図 5.9 デバッガ（ダウンロード）の設定

ブリッジボードを使用していない場合は、次の項目を選択します。

```
$TOOLKIT_DIR¥config¥flashloader¥Epson¥S1C31D01_int.board
```

S5U1C31001L1100 ブリッジボードを使用している場合は、次の項目を選択します。

```
$TOOLKIT_DIR¥config¥flashloader¥Epson¥S1C31D01_ext.board
```

### 5) スタートアップファイルを追加します。

すべてのプロジェクトに必要な共通のファイルがあります。これらのファイルは、ARM コアおよび S1C31D01 ペリフェラルの起動と構成を担当します。ファイルは3つのグループに分かれて配置されます。

ファイルの各グループを追加するプロセスは同じです。

- グループを作成する
- グループにファイルを追加する

これらのファイルを追加するには、プロジェクトを右クリックし、[追加]>[グループの追加]を選択します。

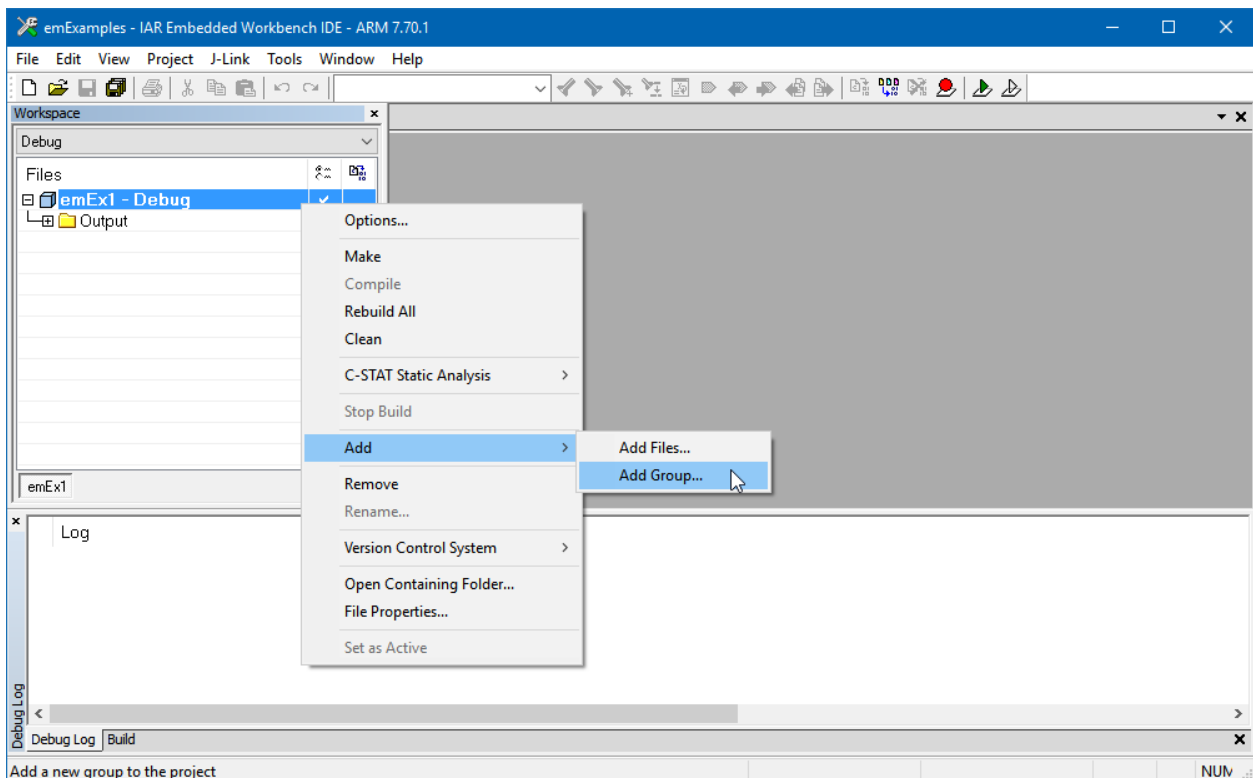


図 5.10 グループの追加

## 5. プロジェクトの作成

最初のグループに“board”という名前を付け、ボードサポートファイルを追加します。

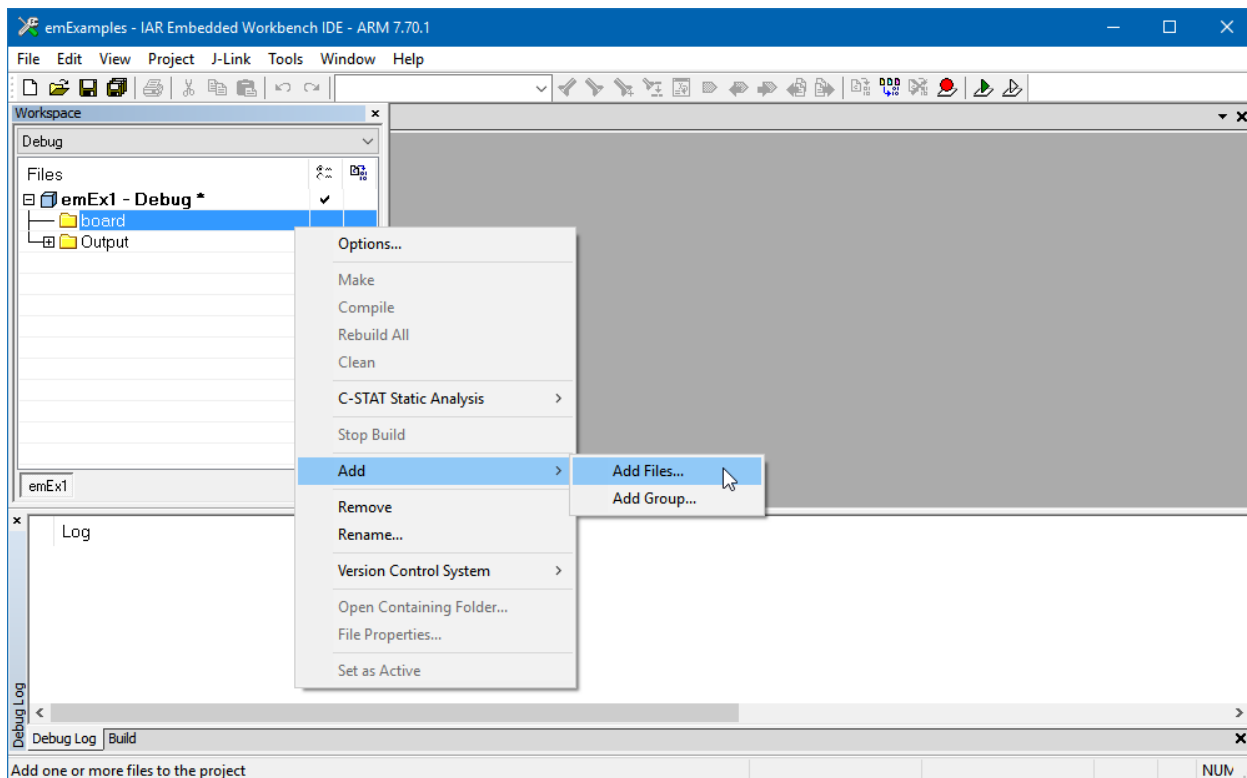


図 5.11 ファイルの追加

以下のファイルを見つけてボードグループに追加します。

```
¥SxC31xxx-emWin¥board¥S5U1C31D01T1¥board.c
¥SxC31xxx-emWin¥board¥S5U1C31D01T1¥board.h
¥SxC31xxx-emWin¥board¥S5U1C31D01T1¥settings.h
¥SxC31xxx-emWin¥board¥S5U1C31D01T1¥IAR¥read.c
¥SxC31xxx-emWin¥board¥S5U1C31D01T1¥IAR¥write.c
```

- 6) 同じ手順を使用してグループ“CMSIS”を作成し、以下のファイルを追加します。

```
¥SxC31xxx-emWin¥CMSIS¥Device¥S1C31D01¥Source¥system_S1C31D01.c
¥SxC31xxx-emWin¥CMSIS¥Device¥S1C31D01¥Source¥IAR¥startup_S1C31D01.s
```

- 7) 最後に、グループ“config”を作成し、以下のファイルを追加します。

```
¥SxC31xxx-emWin¥emWin¥Start¥Config¥Epson_C31.c
¥SxC31xxx-emWin¥emWin¥Start¥Config¥Epson_C31.h
¥SxC31xxx-emWin¥emWin¥Start¥Config¥GUICConf.c
¥SxC31xxx-emWin¥emWin¥Start¥Config¥GUICConf.h
¥SxC31xxx-emWin¥emWin¥Start¥Config¥LCDConf.c
```

¥SxC31xxx-emWin¥emWin¥Start¥Config¥LCDConf.h  
 ¥SxC31xxx-emWin¥emWin¥Start¥Config¥SIMConf.c  
 ¥SxC31xxx-emWin¥emWin¥Start¥Config¥GUI\_X\_Ex.c  
 ¥SxC31xxx-emWin¥emWin¥Start¥Config¥GUIDRV\_EPSON\_MCU.h

8) 完了したら、プロジェクトは以下ようになります：

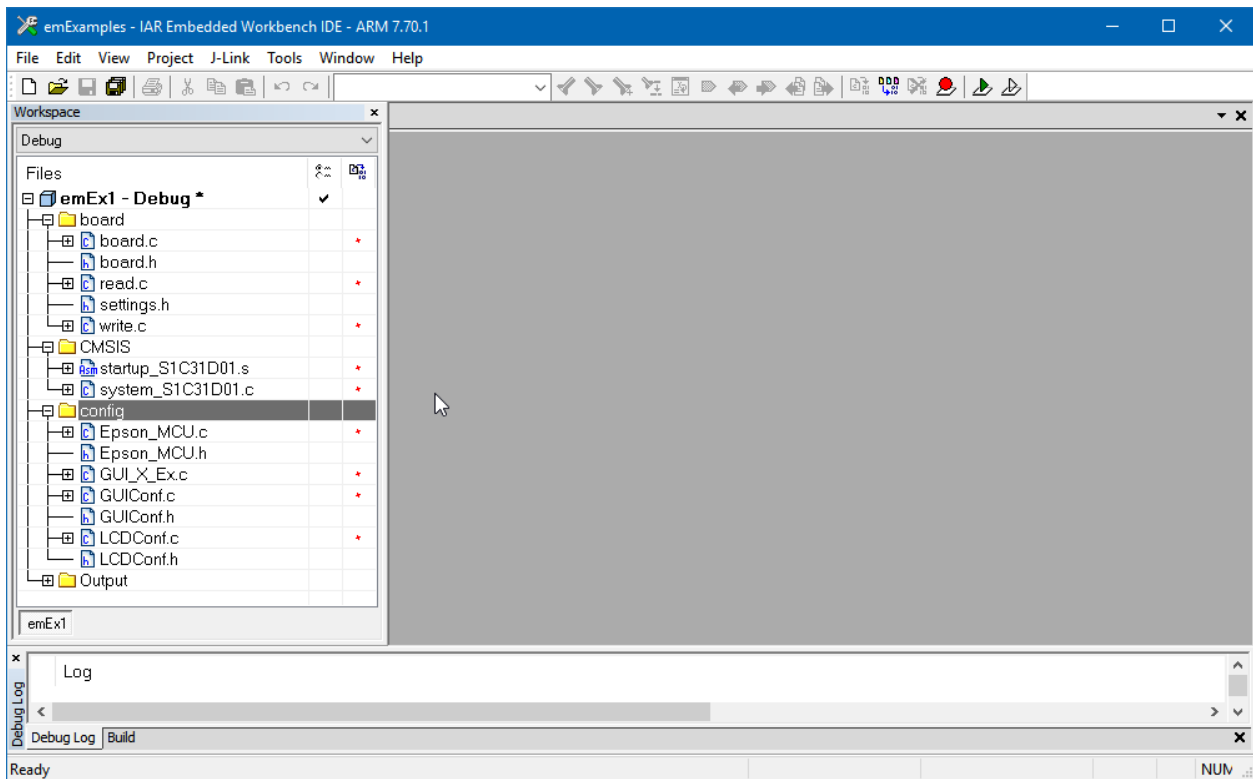


図 5.12 プロジェクトの最終形態

9) ソースファイルをプロジェクトに追加します。

ファイルエクスプローラで“¥emExamples¥emEx1”ディレクトリに移動します。

“¥emExamples¥emEx1”ディレクトリにCソースファイルを作成します。

この例では、“¥SxC31xxx-emWin¥emWin¥Sample¥c31\_sample”ディレクトリから“c31\_sample.c”ソースコードをコピーし、“emEx1.c”という名前に変更しました。

IAR IDE で emEx1 プロジェクトを右クリックし、[追加] >[ファイルの追加]を選択します。

[ファイルの追加]ダイアログボックスでCソースファイルを探して選択し、[開く]ボタンをクリックします

10) 新しいプロジェクトをビルドする。

プロジェクト名を右クリックし、[すべてを再ビルド]を選択します。プロジェクトはエラーなしで再ビルドする必要があります

## 5. プロジェクトの作成

---

11) デバッグ用にダウンロードする。

“Ctrl+D” を押すか、メニューから[プロジェクト]>[ダウンロードしてデバッグ]を選択します。

F5 キーを押してサンプルプログラムを開始するか、IAR デバッグコントロールを使用してプログラムをステップ実行します。



### 6. S1C31 Family で emWin を使用する場合の補足情報

本章では、開発者が S1C31 Family で emWin を使用する際に役立つ情報を記述します。

#### 6.1 ディスプレイのアップデート

ほとんどのグラフィカルシステムでは、新しいデータがディスプレイメモリに書き込まれるとすぐにディスプレイが更新されます。

S1C31 Family での動作は少し異なります。描画はディスプレイメモリバッファに行われますが、パネル更新シーケンスが開始されるまでパネルは更新されません。

表示パネルを更新するプロセスはアプリケーションに委ねられ、最適な時に行われます。

アプリケーションがパネル更新を制御しているため、emWin がメモリデバイス呼び出す必要はありません。デバイスインタフェースは安全なメモリ領域に描画し、すべての描画が完了した後にディスプレイを更新します。これは、S1C31 Family のハードウェアインタフェースが既にシステムに任せていることです。

#### 6.2 2D ライブラリ

emWin 2D ライブラリは、いつでも呼び出すことができる描画ルーチンのコレクションです。同様に、ディスプレイ更新ルーチン `seMDC_PanelUpdate ()` は、ディスプレイを更新するためにいつでも呼び出すことができます。

アプリケーションは、パネルを更新するタイミングを完全に制御しているため、ディスプレイ上に不適切な画像や不完全な画像は描画されないはずですが

#### 6.3 ウィジェット

ウィジェットは、ユーザーインターフェイスの作成を簡素化するためのコントロールとビジュアルエフェクトを追加するために定義されたグラフィカルアイテムです。ウィジェットには、ボタン、チェックボックス、フレームウィンドウ、ノブ、リストボックスなどの項目が含まれます。

ウィジェットの 1 つの特徴は、`WM_PAINT` メッセージを受信したときに描画することです。ほとんどのシステムでは、ウィジェットが描画されるとすぐにディスプレイに表示されます。S1C31 Family のディスプレイアーキテクチャでは、ウィジェットまたは他の GUI エlement がディスプレイバッファに書き込んでもディスプレイにすぐには表示されません。

ウィジェットがディスプレイ上で確実に更新されるように、S1C31 Family 用に書かれたプログラムは、使用するウィジェットごとにメッセージチェーンに接続し、`WM_PAINT` メッセージ用の `seMDC_PanelUpdate ()` を呼び出す必要があります。

プログラマは、ほとんどのウィジェットのウィンドウ・メッセージ・ハンドラをオーバーライドして、期待通りに描画する必要があります。

ウィジェットが `WM_PAINT` メッセージを受け取ると、ウィジェットは必要に応じてその外観を更新します。 `WM_PAINT` の後、システムは `WM_POST_PAINT` メッセージを発行します。

## 6. S1C31 Family で emWin を使用する場合の補足情報

---

オーバーライドの作成は以下のように行います。

```
void aFunction (void)
{
    WM_HWIN hCal;
    ...
    // Create a calendar.
    // Note the callback for the calendar to the _cbCalendar() routine
    hCal = CALENDAR_Create (WM_HBKWIN, 0, 0, 2016, 11, 30, 1, ID_CALENDAR, 0) ;
    WM_SetCallback (hCal, _cbCalendar) ;
    // Show the calendar, pause, then delete the calendar
    WM_InvalidateWindow (hCal) ;
    GUI_Delay (5000) ;
    WM_DeleteWindow (hCal) ;
    ...
}
```

カレンダーコールバックルーチンはすべてのメッセージをデフォルトのカレンダーコールバックに渡し、メッセージが WM\_POST\_PAINT の場合はパネル更新ルーチンが呼び出されます。

```
//-----
// _cbCalendar
// The common dialogs handle most message internally. However updating the
// display requires a panel update sequence and that is where this routine
// comes in.
//
// _cbCalendar connects to the CALENDAR message chain and adds functionality to
// the WM_POST_PAINT message. All other messages are processed by the default
// calendar handler.
//-----

static void _cbCalendar (WM_MESSAGE * pMsg)
{
    GUI_RECT invalidRect;
    CALENDAR_Callback (pMsg) ;
    // WM_POST_PAINT is sent after WM_PAINT completes.
    // This is an excellent time to update the panel.
    if (pMsg-> MsgId == WM_POST_PAINT) {
        WM_GetInvalidRect (pMsg-> hWin, &invalidRect) ;
        seMDC_PanelUpdate (invalidRect.y0, invalidRect.y1) ;
    }
}
```

同様にして、他のすべてのウィジェットの更新機能をオーバーライドします。デフォルトのウィジェットコールバックは、WIDGETNAME\_Callback という形式になります。したがって、MULTIEDIT ウィジェットではデフォルトのコールバックは MULTIEDIT\_Callback () で、TREEVIEW ウィジェットではデフォルトのコールバックは TREEVIEW\_Callback () です。

### 6.4 GUI-Builder

emWin には GUI-Builder というツールが含まれていて、emWin ウィジェットを使ってグラフィカルにユーザーインターフェイスを作成することができます。

demo2\_dialog2.c ファイルのサンプルコードは、GUI-Builder によって生成されたダイアログボックスの使用法を示しています。このツールを使用すると、ウィンドウ化されたユーザーインターフェイスの作成が簡単になりますが、生成されたコードをそのまま使用できるとは限りません。

demo2\_dialog2.c ファイルに含まれるソースコードは、GUI-Builder ツールによって生成されました。デモ作業を行うために必要なコードはすべて、下記のコメントで括られた領域に追加されています。

```
// USER START (Optionally insert ...)  
...  
// USER END
```

サンプルには、フレームウィンドウ、グラフ表示、および2つのボタンが含まれています。サンプルにはマウス入力がないので、ボタンは機能しておらず、単にデフォルトのボタンコードで描画され、WM\_POST\_PAINT メッセージ中にダイアログコールバックルーチンによって表示されます。

### 6.5 ビットマップの作成

ビットマップをセーブおよびロードするファイルシステムがなければ、画像を作成して文字の配列としてセーブする必要があります。

イメージファイルを文字配列に変換するには2つの方法があります。両方の方法には利点とトレードオフがあります。含まれているサンプルコードは、両方の方法を使用したイメージ処理をしています。

プロジェクト c31\_demo2 では、demo2\_images.c の画像が Bin2C.exe で作成されました。ファイル内の最初のイメージは、.BMP ファイルのヘッダー情報、0x42 (b) および 0x4D (m) です。残りの画像は.GIF 画像から得られたものです。文字配列を簡単に調べると、.GIF イメージの特徴的な 0x47、0x49、0x46 (GIF) が表示されます。

プロジェクト c31\_demo3 では、epsonlogo.c ファイルの Epson ロゴイメージを BmpCvt プログラムを使用して作成しました。イメージ配列内のデータには、ファイルヘッダー情報は含まれていません。

#### 【Bin2C】

Bin2C.exe は、Segger 社の Web サイト ([https://www.segger.com/downloads/free\\_tools](https://www.segger.com/downloads/free_tools)) からダウンロードできます。

プログラム Bin2C.exe は、任意のバイナリファイルを文字の配列に変換します。プログラムを使用してバイナリ形式のイメージファイル (.BMP、.JPG、.GIF、.PNG) を文字配列に変換することができます。

バイナリイメージの文字配列への直接変換の利点は、イメージファイルと同じ特性を保持することであり、イメージファイルが圧縮されている場合はサイズも小さくなります。

## 6. S1C31 Family で emWin を使用する場合の補足情報

この利点は、画像を復号化するためのルーチンを含む必要性によって相殺されます。不可逆圧縮な画像フォーマット（例えば.JPG ファイル）では、復号された画像の色を予想される色と一致させることが困難になります。

Bin2C を使用してイメージを作成する方法は以下の通りです。

- 1) アプリケーションを開きます。



図 6.1 Bin2C 起動画面

- 2) [Select file ...]ボタンをクリックし、変換したいファイルがあるディレクトリを参照し、ファイルを選択して[Open]ボタンをクリックします。

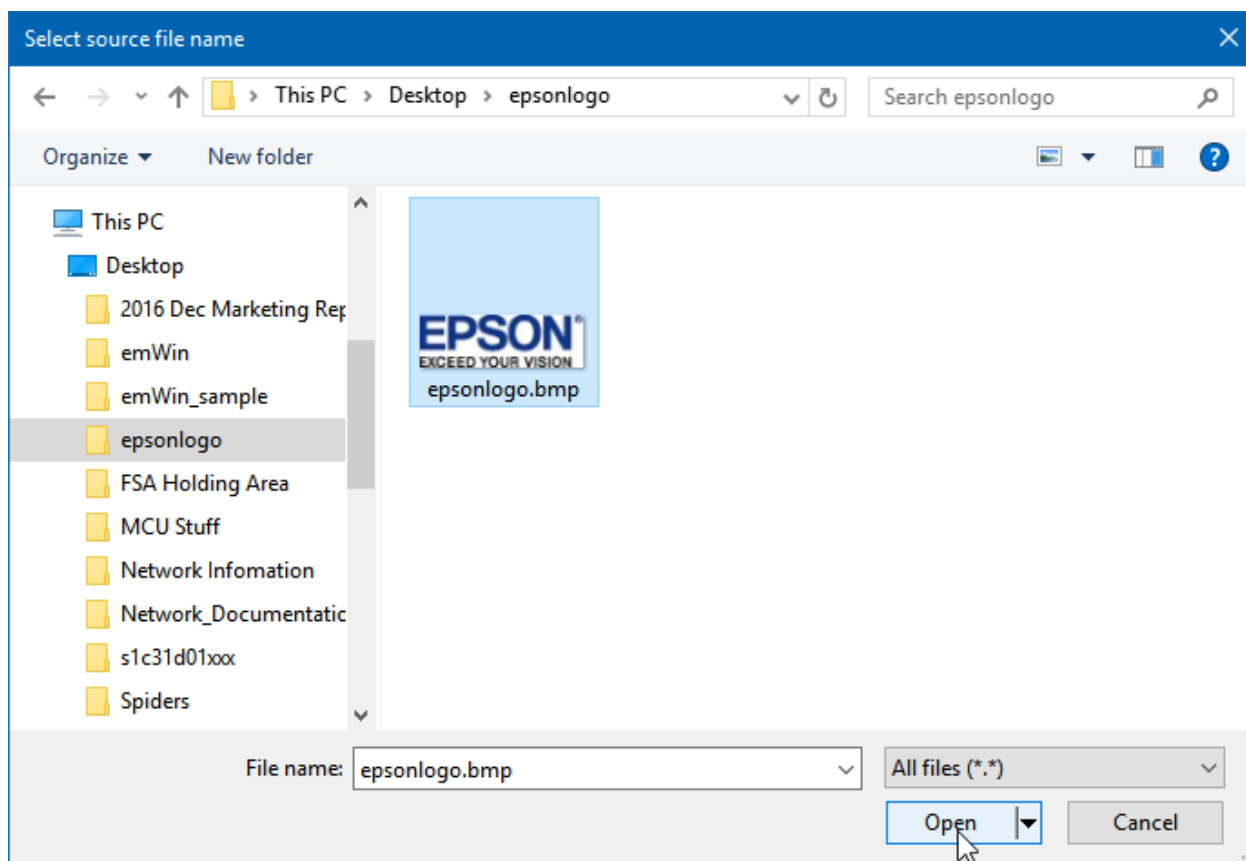


図 6.2 ファイル選択ダイアログ

## 6. S1C31 Family で emWin を使用する場合の補足情報

---

- 3) “Convert” ボタンをクリックすると、ファイルの ASCII 表現がバイナリイメージと同じディレクトリに保存されます。

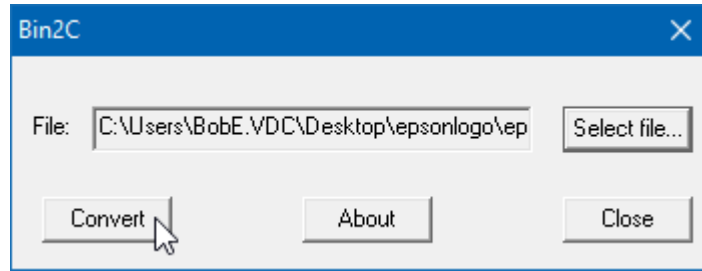


図 6.3 画像変換ダイアログ

## 6. S1C31 Family で emWin を使用する場合の補足情報

### 【BmpCvt】

BmpCvt.exe で作成された画像は、カラーパレットと画像の各ピクセルを表す値の配列で構成されています。このようにして作成された画像は、Bin2C.exe で作成された画像よりもわずかに大きいかもしれませんが、画像デコードルーチンに必要なコードスペースはなく、画像のデコードオーバーヘッドがないため、画像の読み込み速度がやや速くなります。

emWin アプリケーションで使用するビットマップイメージを作成する方法は以下の通りです。

- 1) まず、イメージを作成し、色深度を 1 ピクセルあたり 8 ビットに減らします。



“¥SxC31xx-emWin¥emWin¥Sample¥c31\_demo3¥artwork” ディレクトリにある eponlogo.bmp を参照してください。

- 2) プログラム BmpCvt.exe を起動し、イメージを BmpCvt ウィンドウにドラッグします。

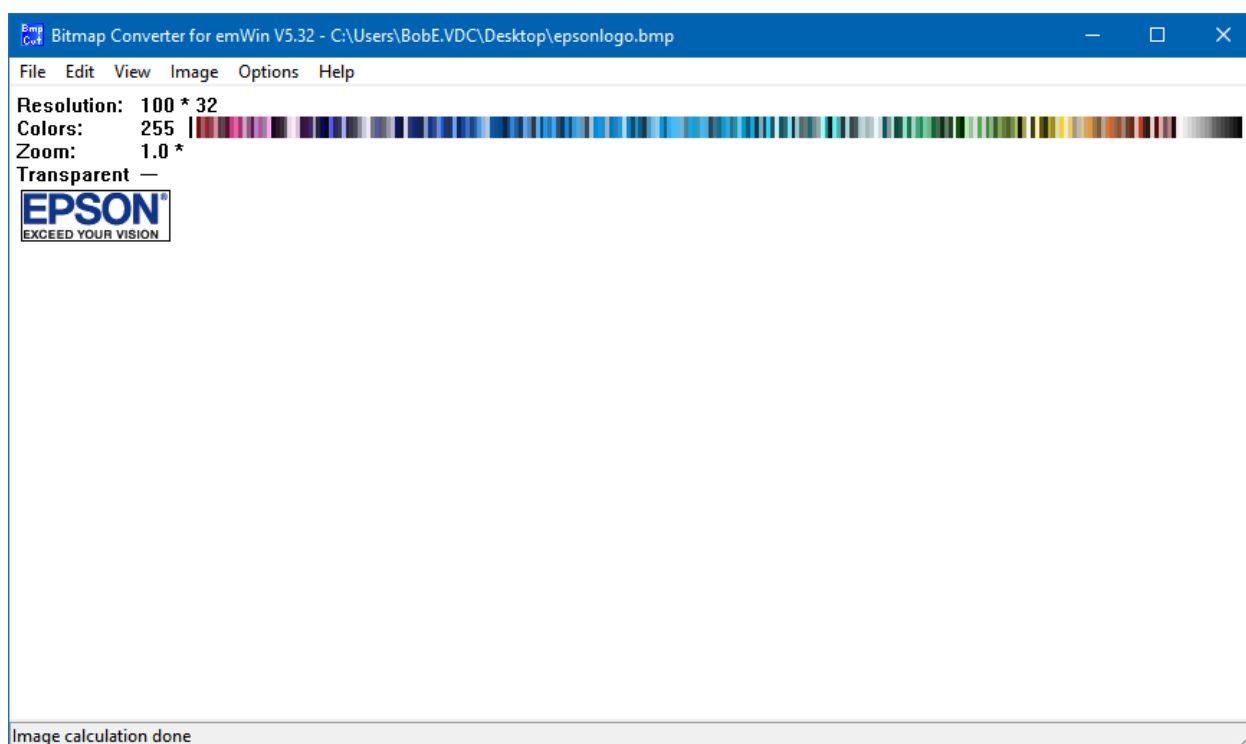


図 6.4 イメージファイル読み込み

## 6. S1C31 Family で emWin を使用する場合の補足情報

- 3) [Image]>[Convert to]>[6bit color (222)]を使用してイメージを6ビットカラーに変換します。

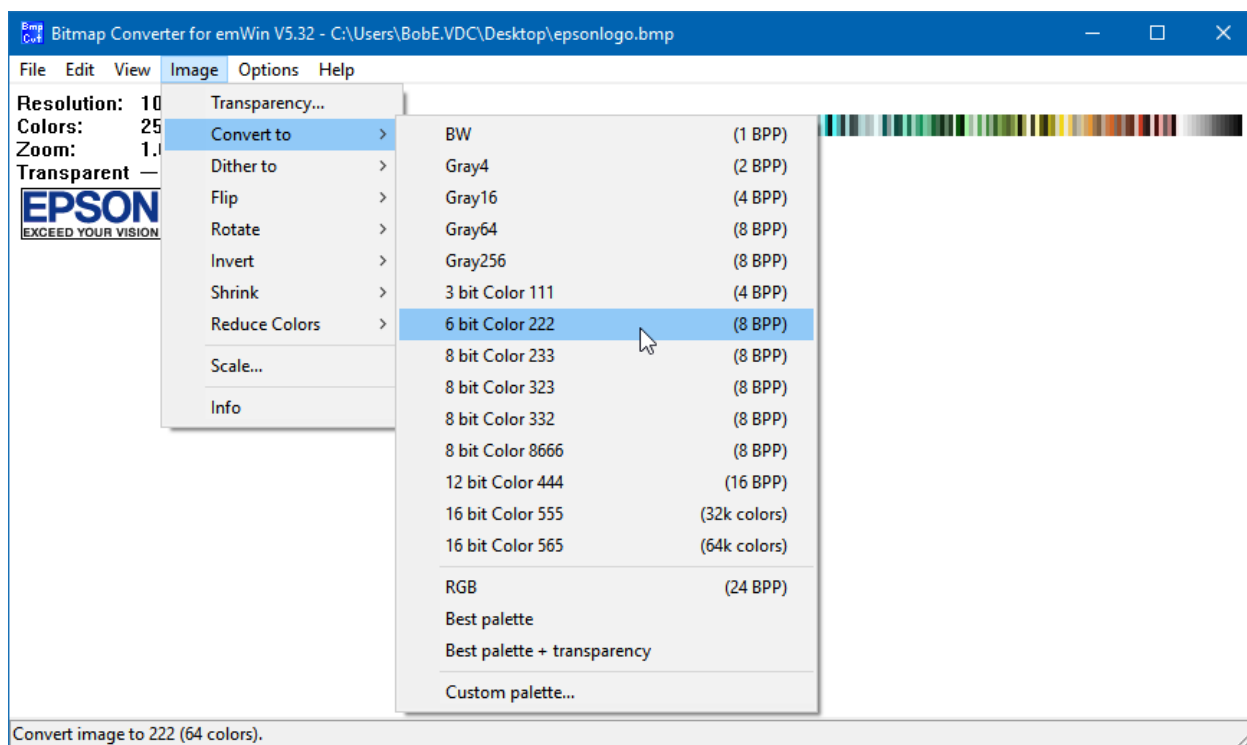


図 6.5 イメージ減色操作

- 4) [File]>[Save as...]を選択して、画像を文字配列としてセーブします。

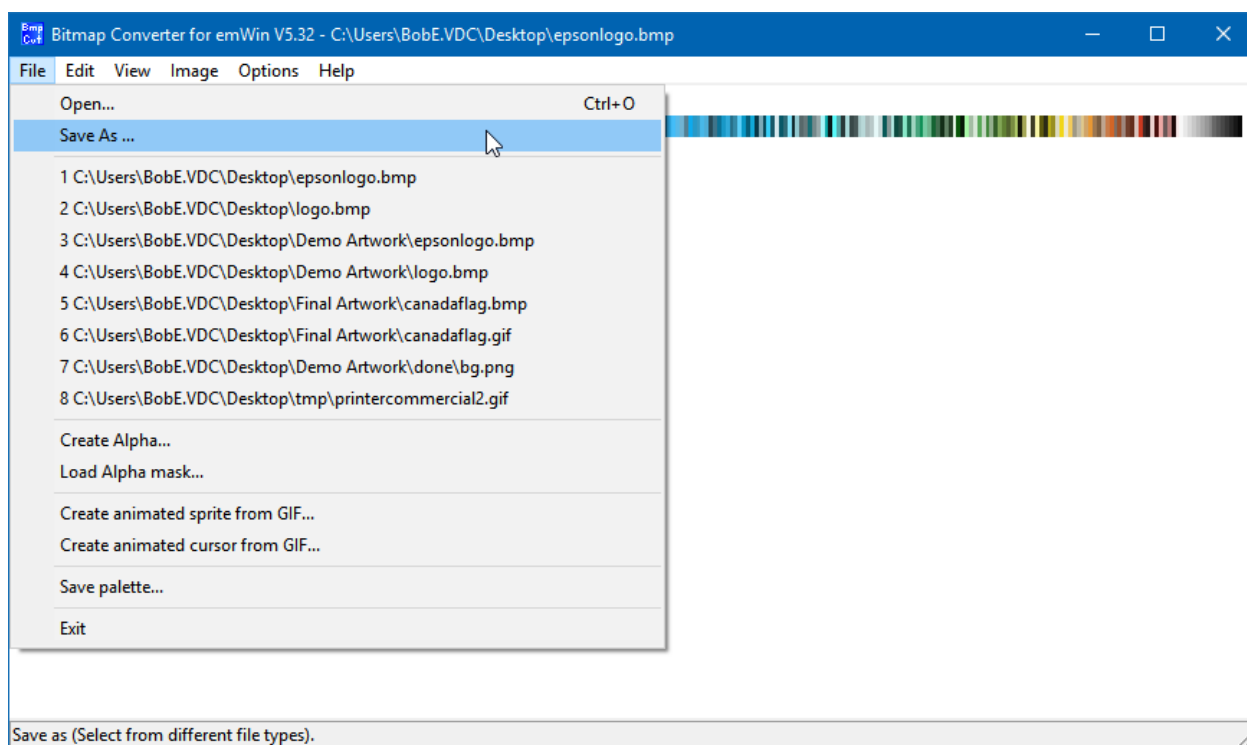


図 6.6 イメージ保存操作

## 6. S1C31 Family で emWin を使用する場合の補足情報

- 5) “Save as type :” を “C"ビットマップファイル (\*.c)” に設定し、目的のファイルに任意の名前を付けます。 [Save]ボタンをクリックして画像ファイルをセーブします。

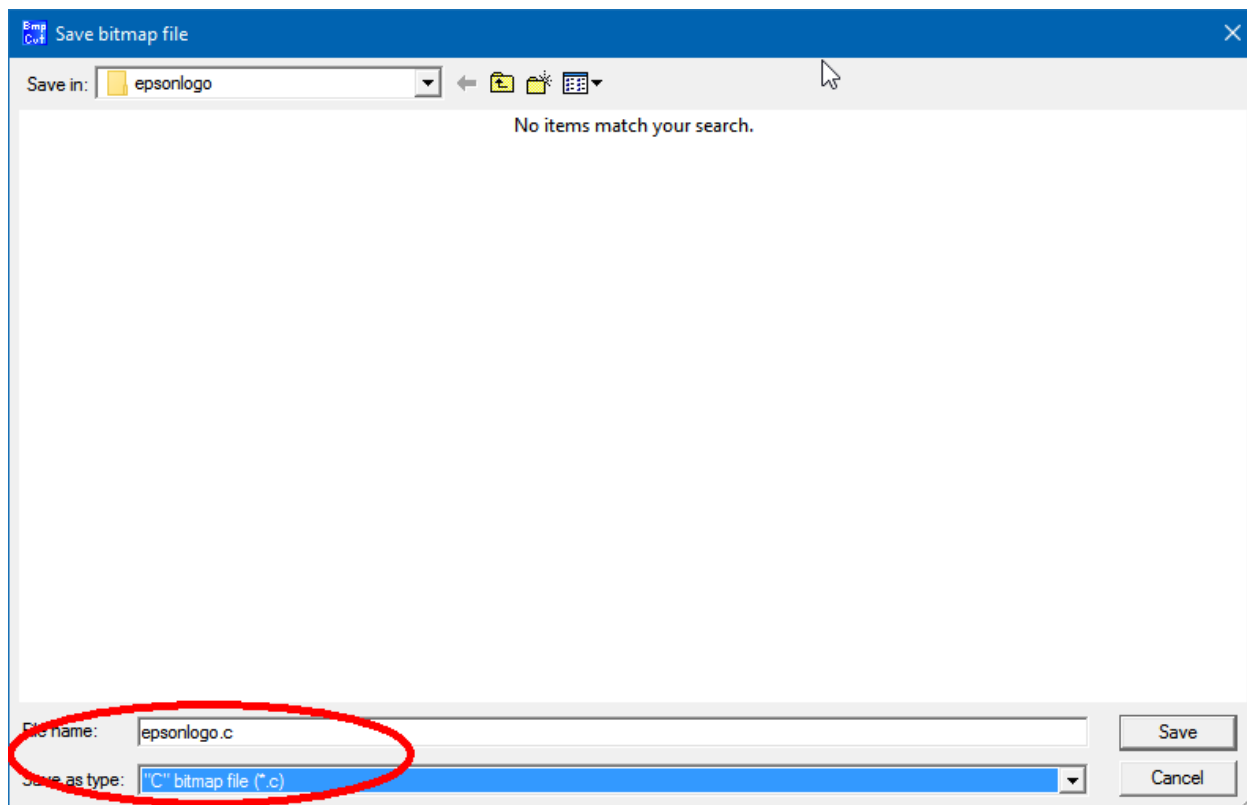


図 6.7 イメージ保存操作（保存方式選択）

### 6.6 既存の emWin サンプルコードの使用

Segger 社は、emWin GUI 用のサンプルコードの大量のコレクションを提供しています。ほとんどのサンプルコードは S1C31D01 プラットフォームで動作しますが、調整や変更が必要になる場合もあります。

Segger 社のサンプルコードは、Windows ベースのエミュレータで使用するために作成されています。これにより、余分なオーバーヘッドが追加され、S1C31D01 で使用するためには削除する必要があります。



改訂履歴表

付-1

Rev. No.	日付	ページ	種別	改訂内容 (旧内容を含む) および改訂理由
Rev 1.0	2017/10/11	全ページ	新規	新規制定

## セイコーエプソン株式会社

営業本部 デバイス営業部

---

東京 〒191-8501 東京都日野市日野 421-8  
TEL (042) 587-5313 (直通) FAX (042) 587-5116

大阪 〒541-0059 大阪市中央区博労町 3-5-1 御堂筋グランタワー15F  
TEL (06) 6120-6000 (代表) FAX (06) 6120-6100

---

ドキュメントコード : 413564000  
2017年 11月 作成