

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

H8S/2456

EXDMAC と A/D による QVGA LCD とタッチパネル用 GUI アプリケーション

要旨

H8S/2454, H8S/2456, H8S/2426 および H8S/2424 の EXDMAC と A/D モジュールによる QVGA LCD 表示とタッチパネル制御の GUI アプリケーション例を示します。

動作確認デバイス

H8S/2456, H8S/2426, H8S/2454, H8S/2424

目次

1. 説明	2
2. PC 側ツールのインストール.....	4
3. GUI 画面の WYSIWYG デザイン	5
4. Windows 上で実行可能な EXE ファイルの生成	19
5. H8S/2456 実機環境での動作.....	25

1. 説明

PC で動作する GUI 開発ツールを用いて H8S/2456 用の QVGA LCD 表示とタッチパネルのソフトウェアを作成することができます。

ツールの特徴は次のとおりです。

- (1) Visual Studio のフォームデザイナを使って GUI 画面を WYSIWYG でデザインできます。(図 1)
- (2) Visual Studio で作成した画面デザインから C 言語プログラムを自動生成できます。
- (3) 生成したプログラムを PC 上でビルドして動作確認できます。(図 2)
- (4) PC 上で動作確認ができたプログラムをターゲットプロセッサ用にビルドしなおすだけで移植できます。(図 3)

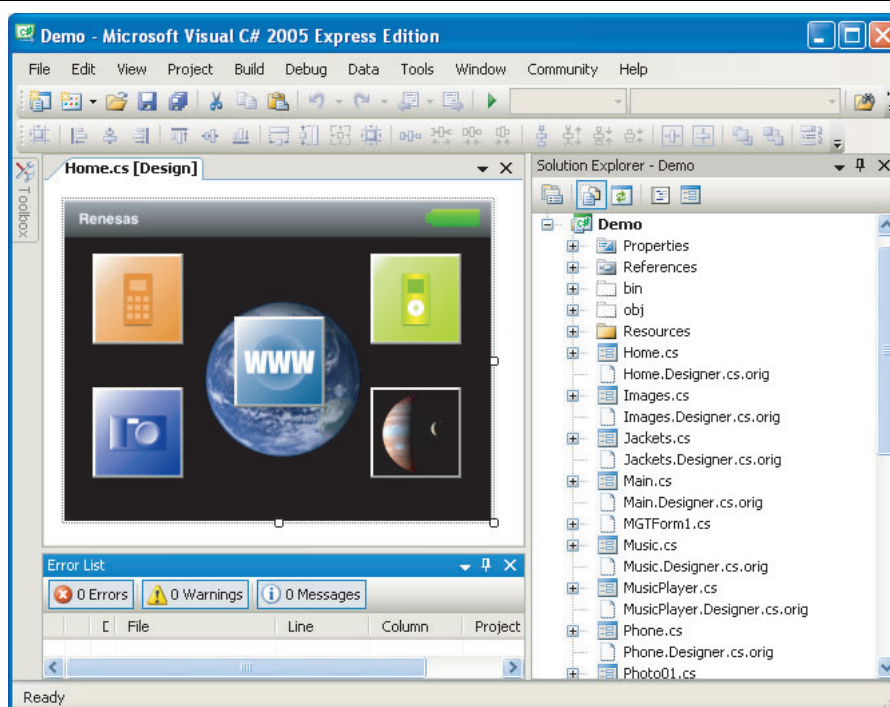


図 1 GUI 画面のデザイン例



図 2 PC 上の動作確認



図 3 ターゲットプロセッサ上の動作確認

2. PC 側ツールのインストール

GUI 画面の WYSIWYG デザインと GUI プログラム生成ツール (guigen) が自動生成した C プログラムを PC 上でビルドして動作確認するため、下記のツールのインストールが必要となります。

「Visual C# 2005 Express Edition」および「Visual C++ 2005 Express Edition」をインストールしてください。

Visual Studio 2005 Service Pack 1 を入手して、インストールした Visual C++ 2005 Express Edition と Visual C# 2005 Express Edition を最新の状態にアップデートしてください。

画面デザイン用のカスタムコントロール (MGTCControls) のインストールと登録を行ってください。

画面デザイン用のカスタムフォームテンプレート (MGTFrm.zip) をインストールしてください。

詳細は別紙の GUI 開発ツールユーザーズマニュアルの第 3 章を参照ください。

3. GUI 画面の WYSIWYG デザイン

Visual C# 2005 Express Edition で GUI 画面の WYSIWYG デザインのみを行います。

画面デザインから自動生成された C# のコードを GUI プログラム生成ツール (guigen) で C ソースに変換してこれを Visual C++ 2005 Express Edition 上でビルドを行いますので、C# 上ではビルドは行いません。

3.1 使用可能な GUI 部品について

GUI 画面の WYSIWYG デザインで使用可能な GUI 部品には制限があります。画面デザイン用のカスタムコントロール (MGTCControls) で追加した部品のみが利用できます。

詳細は別紙の GUI 開発ツールユーザズマニュアルの[4.3. GUI 部品 (コントロール)], 第 5 章 GUI 部品を参照ください。

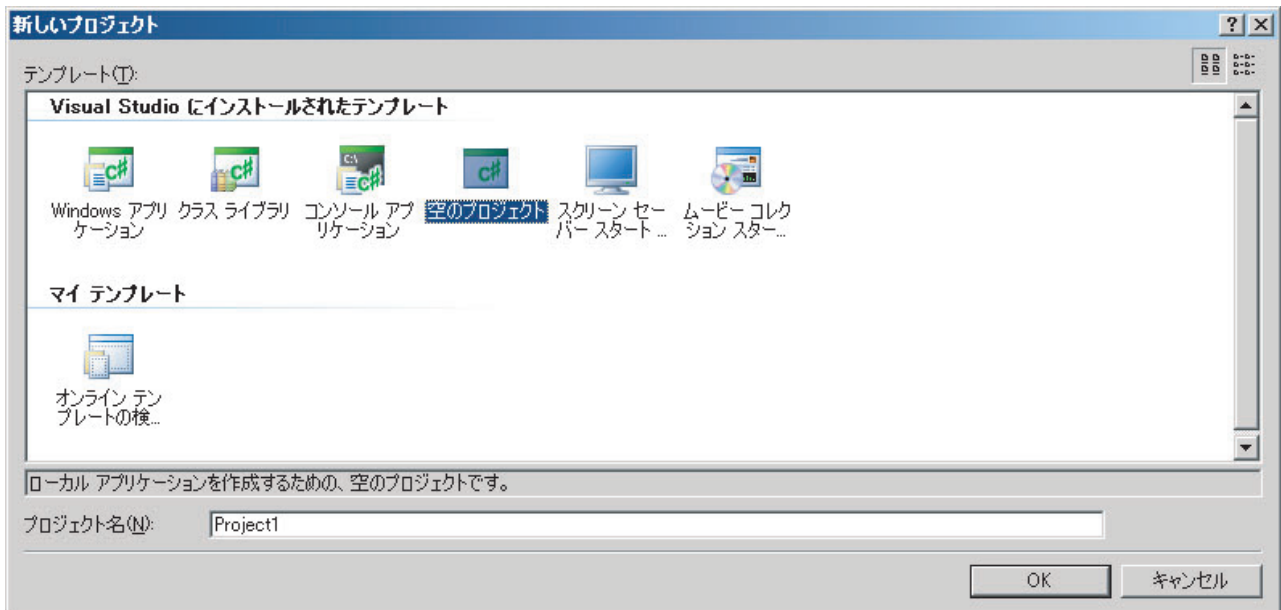
3.2 新しいプロジェクトの作成

次の手順で、新しいプロジェクトを作成します。

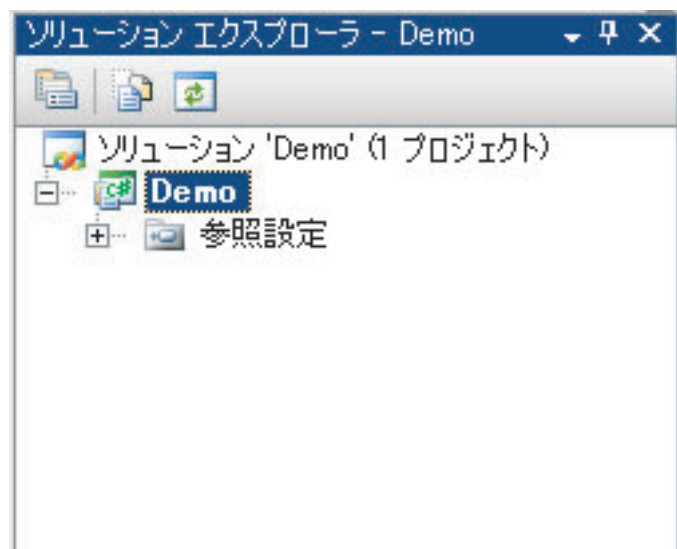
- (1) Microsoft Visual C# 2005 Express Edition を起動します。
- (2) 「ファイル」メニューから「新しいプロジェクト...」を選択します。



- (3) 「新しいプロジェクト」ダイアログが開くので、「空のプロジェクト」を選択し、「プロジェクト名」欄にプロジェクトの名前を入力して、「OK」ボタンを押します。



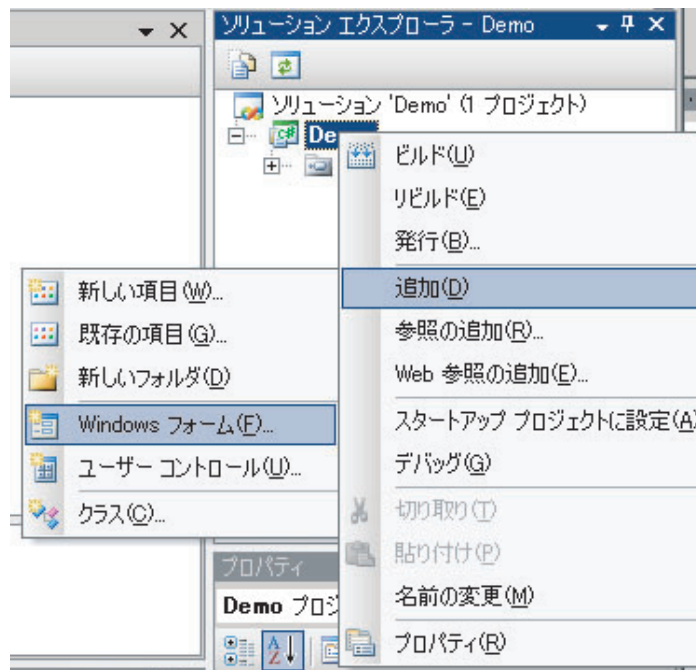
- (4) 新しいプロジェクトが作成され「ソリューションエクスプローラ」ウィンドウに表示されます。



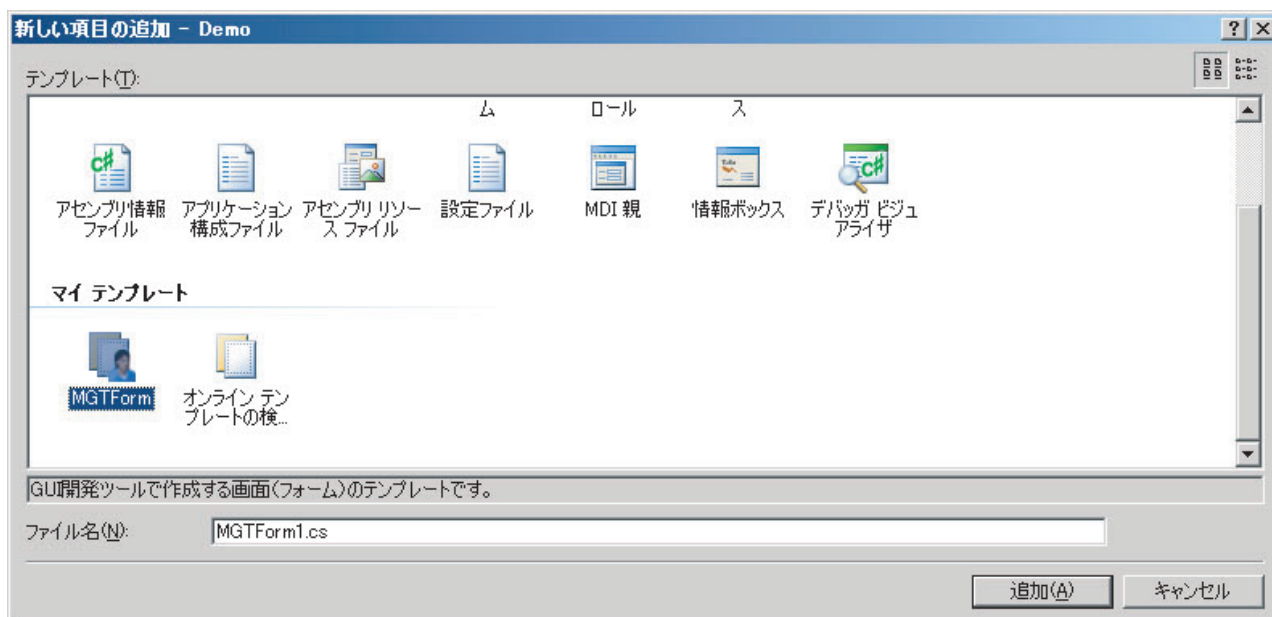
3.3 画面の追加

次の手順で、インストールしておいたカスタムフォームテンプレート (MGTFORM) を使って、プロジェクトに新しい画面 (フォーム) を追加します。

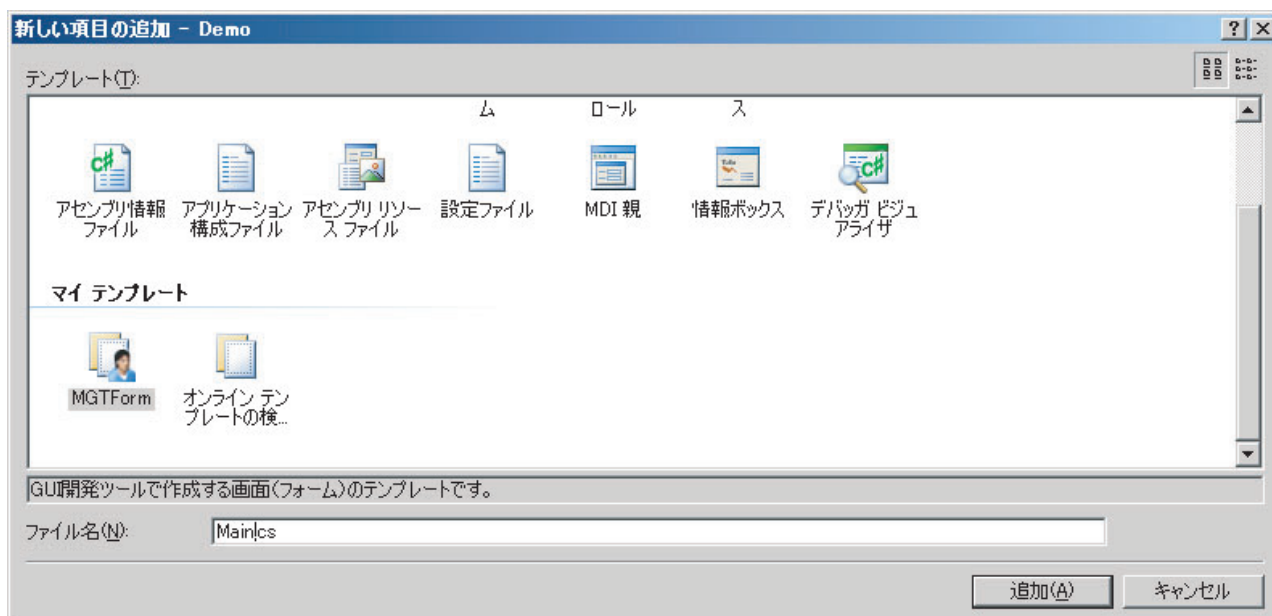
- (1) 「ソリューションエクスプローラ」ウィンドウのツリー表示で、画面を追加したいプロジェクト名をクリックして選択します。
- (2) 「プロジェクト」メニューから「Windows フォーム(F)...」 「追加(D)」を選択します。



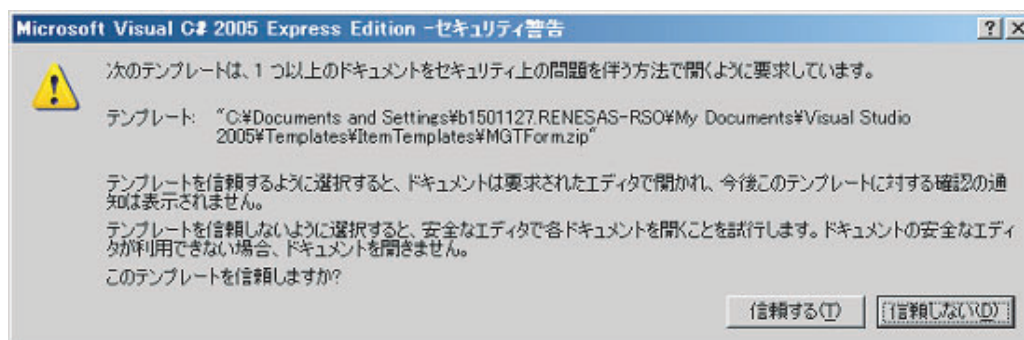
- (3) 「新しい項目の追加」ダイアログが開くので、「テンプレート」欄の下側に表示されている「MGTFORM」アイコンをクリックして選択します。なお、Visual Studio の Windows フォームをそのまま使うことはできません。画面をデザインする場合は、あらかじめインストールした GUI 開発ツール専用のカスタムフォームテンプレート (MGTFORM) を使用してください。



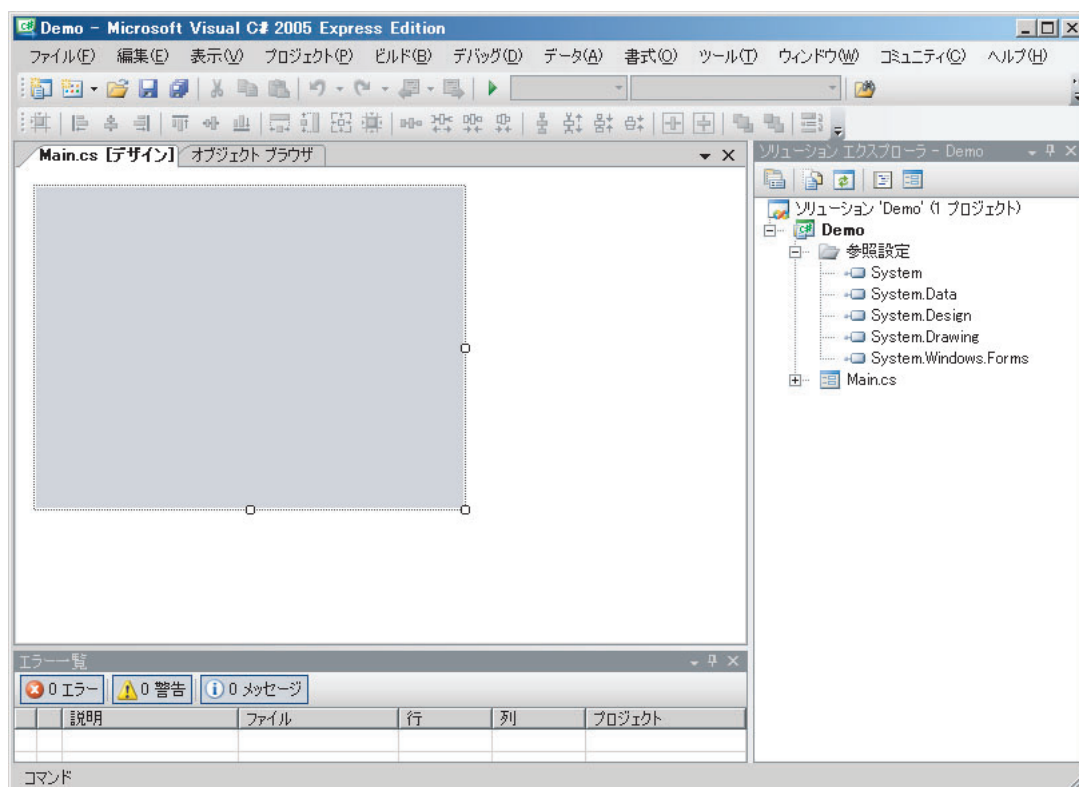
- (4) 「ファイル名」欄に画面の名前を入力して、「追加」ボタンを押します。
 なお、次のいずれかの名前をもつフォーム (画面) を初期画面として作成する必要があります。
 MAIN Main main



以下の Window が表示されますので、「信頼する(T)」ボタンを押します。



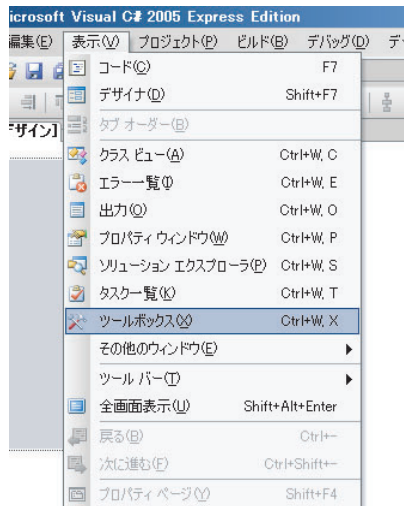
- (5) 新しい画面が作成され「ソリューションエクスプローラ」ウィンドウで選択されているプロジェクトの下に表示されます。同時に、その画面のデザイナー画面が開きます。



3.4 GUI 部品の追加

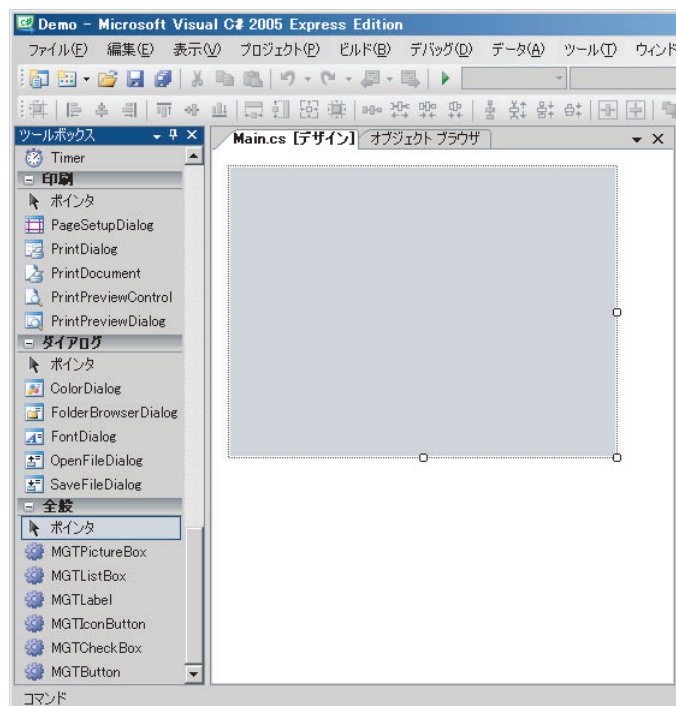
次の手順で、フォーム (画面) に GUI 部品を追加します。

- (1) 「ツールボックス」画面を開き、「全般」タブの下にあるカスタムコントロール (MGTCControls) から目的の GUI 部品をクリックして選択します。



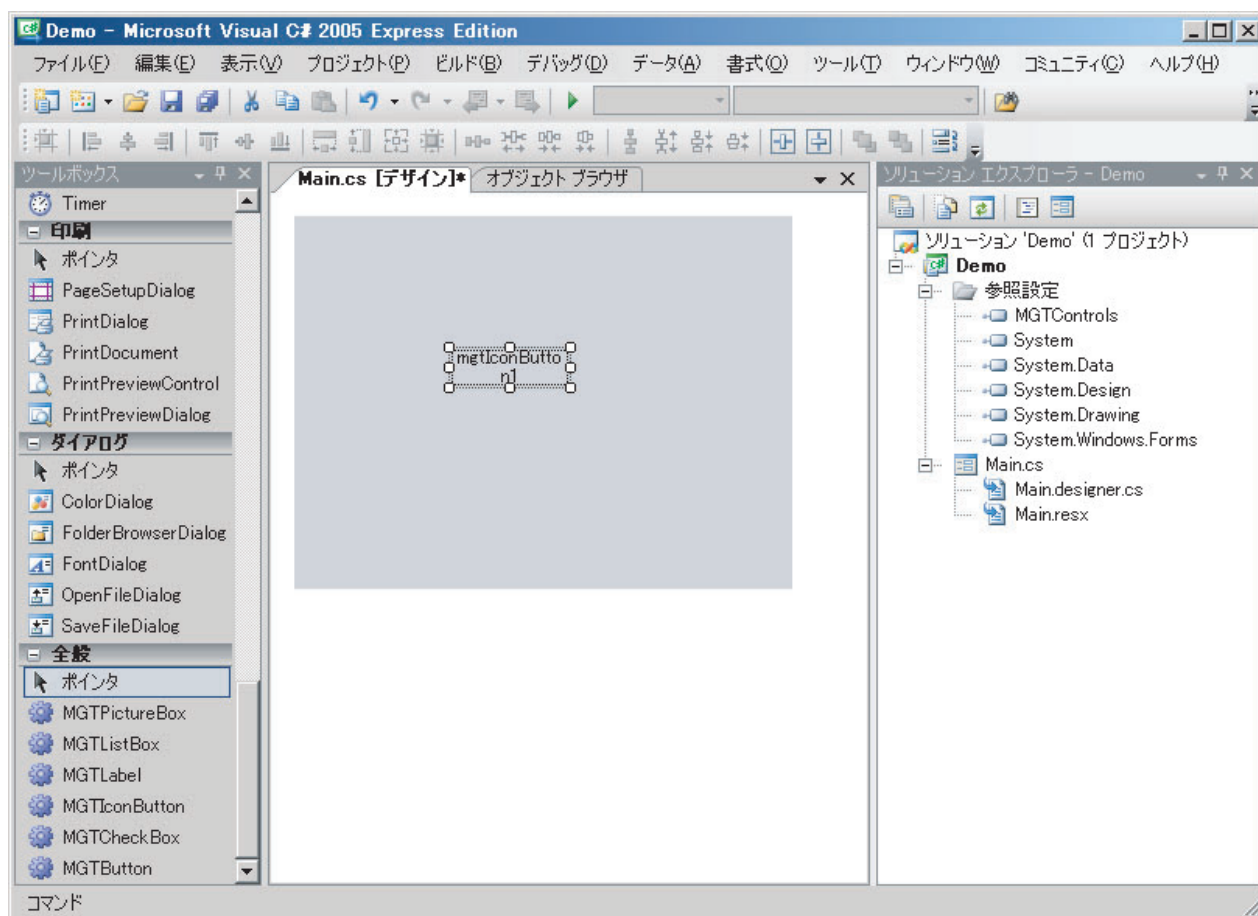
インストールが正しくされていれば、ツールボックスには次の GUI 部品が登録されています。

- MGTLLabel
- MGTButton
- MGTIconButton
- MGTCheckBox
- MGTListBox
- MGTPictureBox



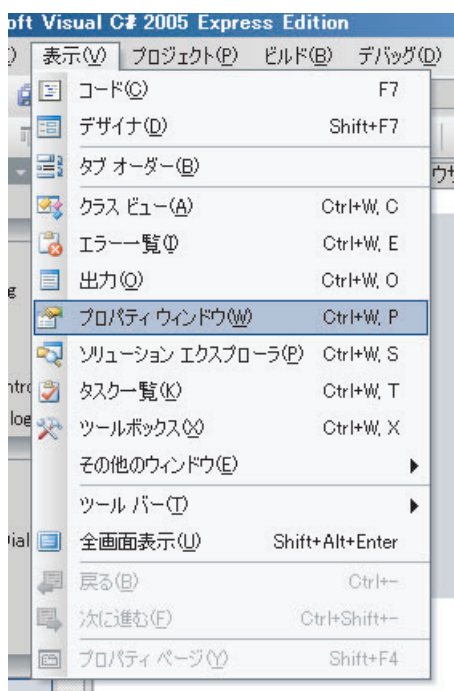
- (2) 選択した GUI 部品をデザイナー画面にドラッグ&ドロップします。
- (3) 選択した GUI 部品が画面に追加されます。

以下は画面上に IconButton を追加した例です。



フォーム (画面) または GUI 部品にはプロパティを設定することができます。プロパティウィンドウを開いた状態でフォーム (画面) または GUI 部品を選択すると、プロパティの確認と変更を行うことができます。

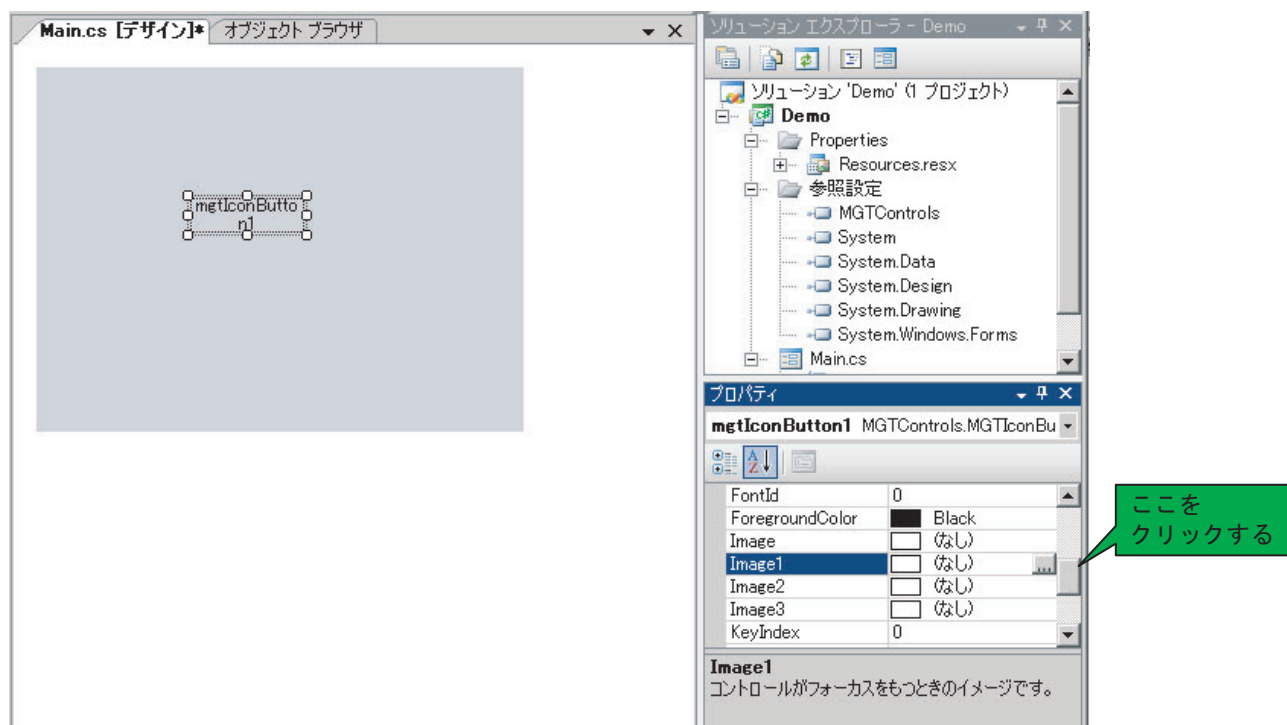
プロパティの詳細については別紙の GUI 開発ツールユーザーズマニュアルの「第 6 章 GUI 部品のプロパティ」を参照ください。



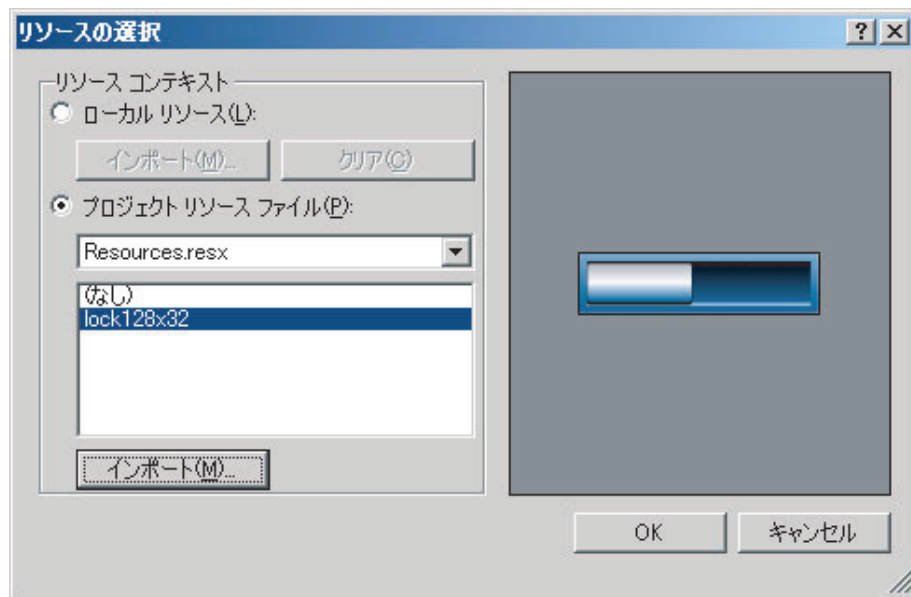
以下は IconButton のプロパティの例です。この例では次のプロパティを指定しています。

- ボタンを押していない時と押した時の画像の指定 (Image, Image1, Image2)
- ボタンのサイズの指定 (Size)
- ボタンの位置の指定 (Location)
- テキストの削除 (Text)

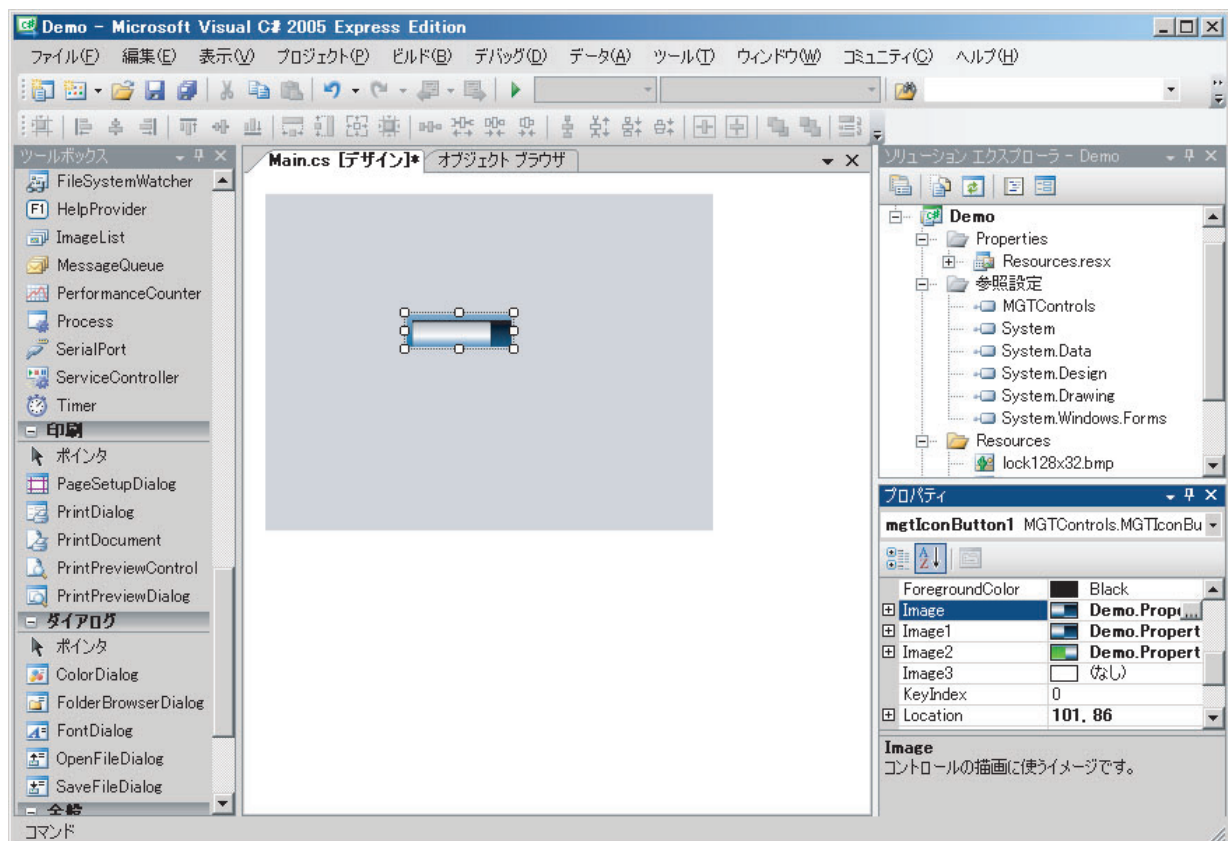
以下は Image1 の画像を指定の例です。同様の方法で Image, Image2 の画像も指定できます。



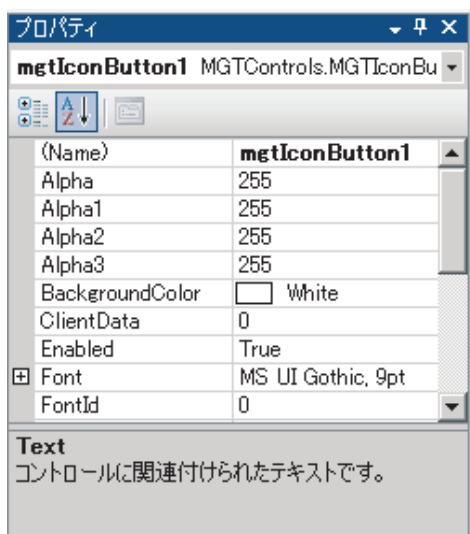
Resources フォルダの画像を選択後に「OK」ボタンで決定します。Resources フォルダに存在しない画像を選択する場合には「Import」ボタンをクリックして画像を選択後に「Open」ボタンで決定します。



以下は Image, Image1, Image2, Size, Location, Text のプロパティを指定した例です。



フォーム (画面) に GUI 部品をドラッグ&ドロップすると、名前が自動で付きますが、Name プロパティで変更可能です。



以下の例では Name プロパティを unlock に変更しています。この名前は C# のデザインを GUI プログラム生成ツール (guigen) で C ソースに変換すると、gui.out.h の中で以下のように定義されます。

```
#define MGT_O_Main_unlock 20
```

定義のルールは別紙の GUI 開発ツールユーザズマニュアルの[4.1. オブジェクト ID]を参照ください。

この定義は C ソース中でイベントの判定 (ボタン検出) で使われます。



3.4.1 OnClickGoto イベントによるフォーム (画面) の切り替え

OnClickGoto プロパティに飛び先のフォーム (画面) 名を指定すると、ボタンが押された時に指定したフォーム (画面) に切り替わります。このイベント処理は C# が自動生成したコードから GUI プログラム生成ツール (guigen) で C ソースを生成するため、ユーザはコードを記述する必要がありません。

OnClickGoto プロパティに指定した飛び先のフォーム (画面) を 3.3 章の方法で追加するだけで OK です。

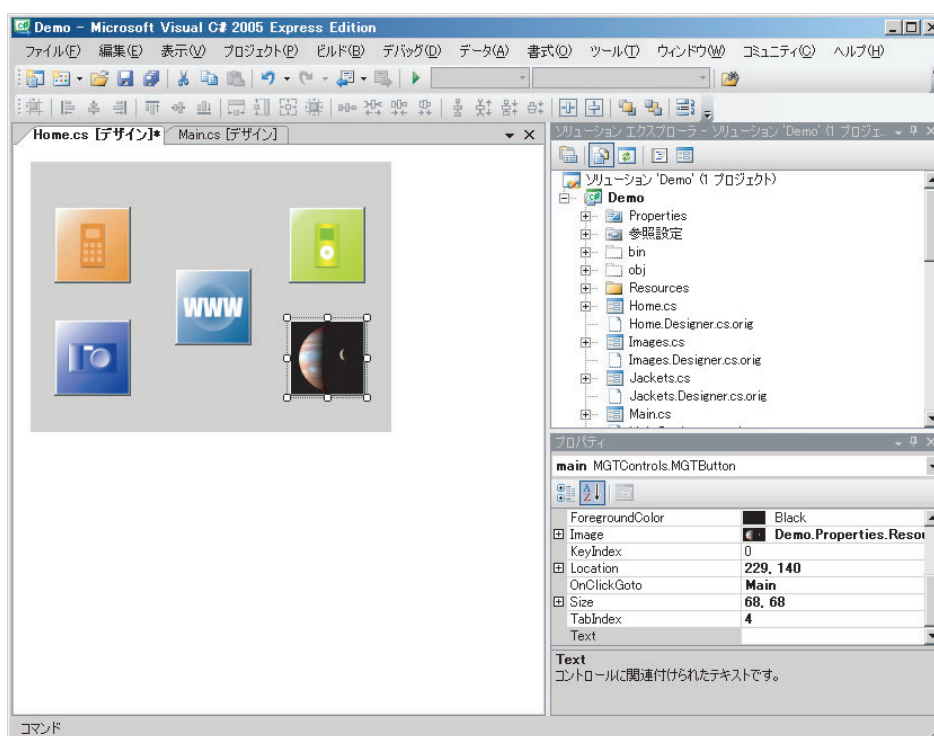
以下に Main フォームの unlock ボタンの OnClickGoto プロパティに Home フォームを指定した場合の例を示します。



飛び先の Home フォーム上には 5 つの MGTButton を配置しています。MGTButton は IconButton と同様な機能を持っています。

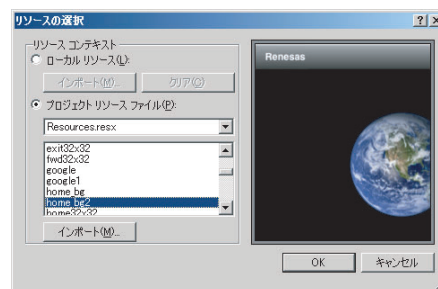
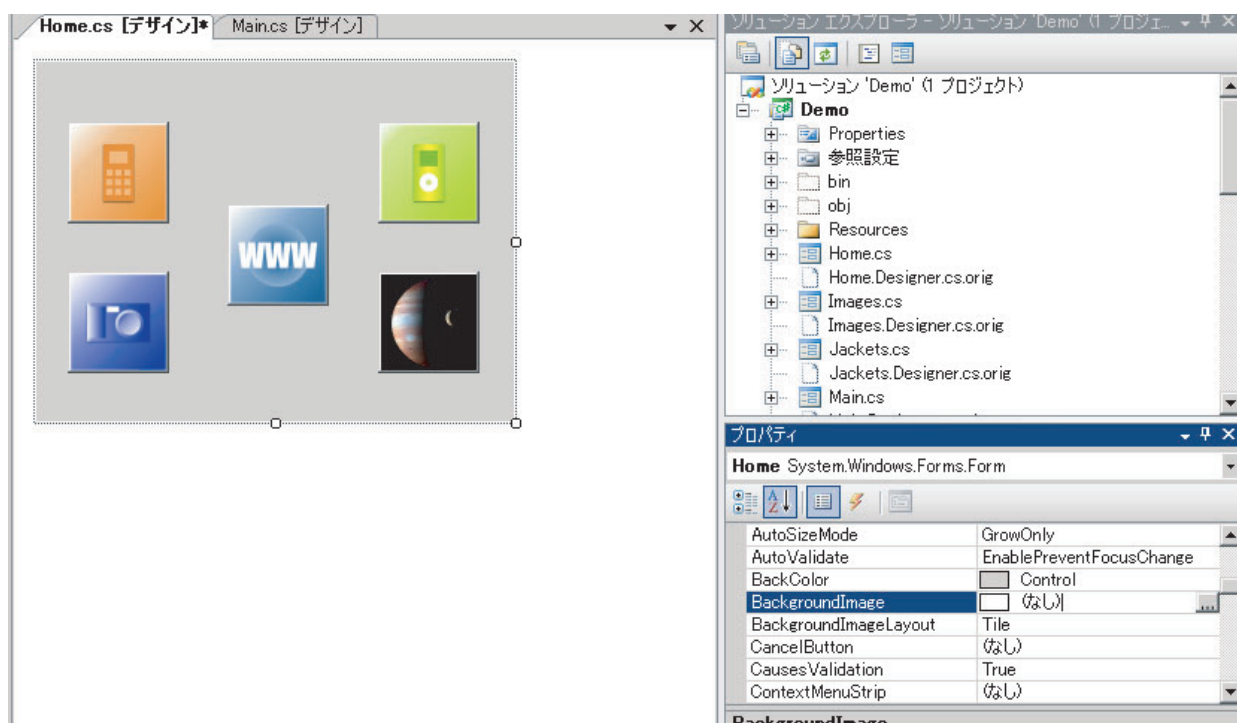
Home フォームの例では各ボタンの Name, Image, Location, OnClickGoto, Size, Text のプロパティを設定しています。これらの動作は Main フォームの IconButton と同様のため、説明を省略します。

詳細は Solution Explorer 上で Home フォーム (Home.cs) を選択して、各 GUI 部品のプロパティを参照ください。



3.5 フォーム (画面) の BackgroundImage の指定

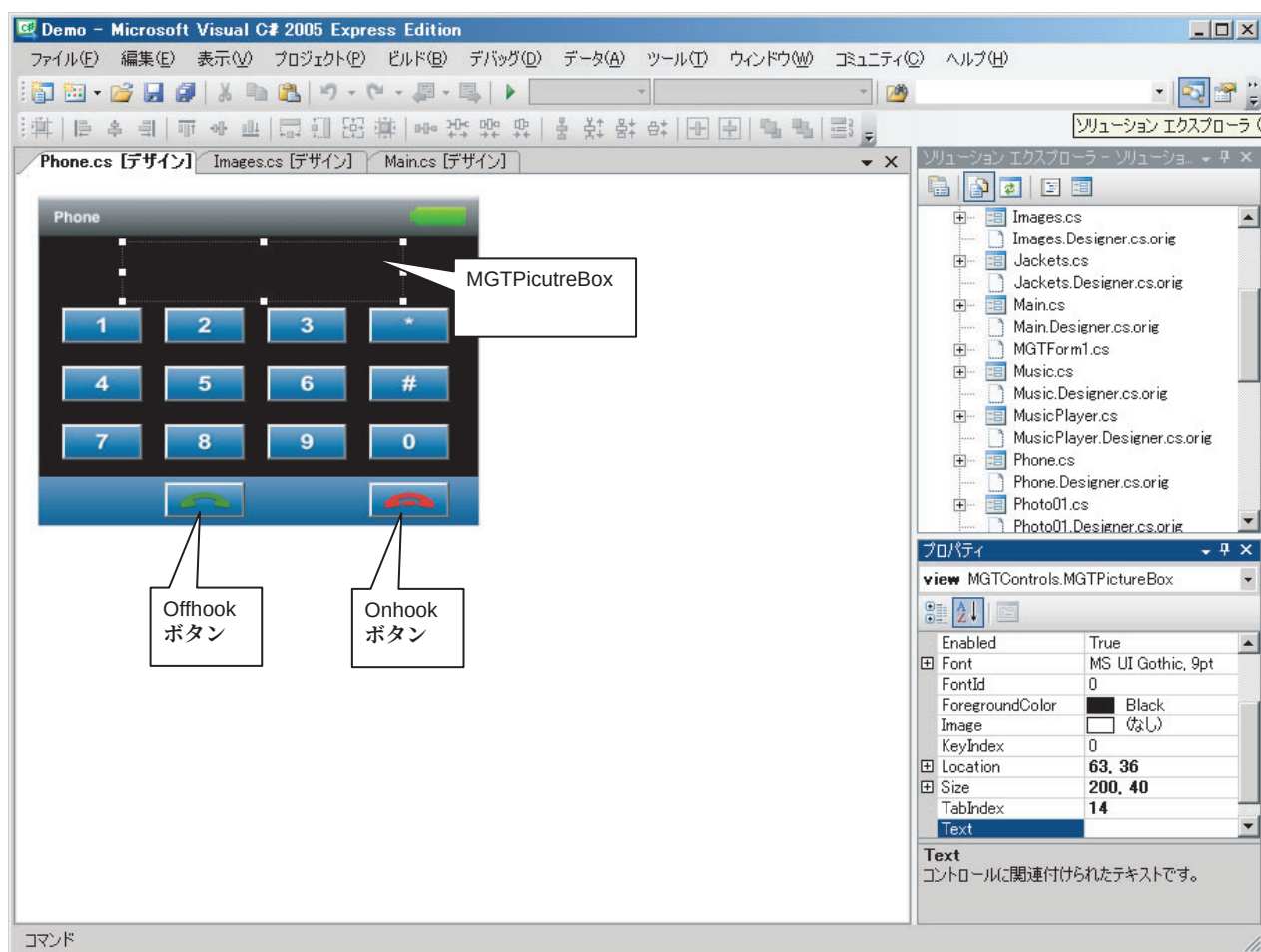
ボタンの画像指定と同じ方法でフォーム (画面) の BackgroundImage プロパティで背景を指定することができます。



3.6 PictureBox とボタンイベントの実装例

以下に Phone フォームに実装した例を示します。

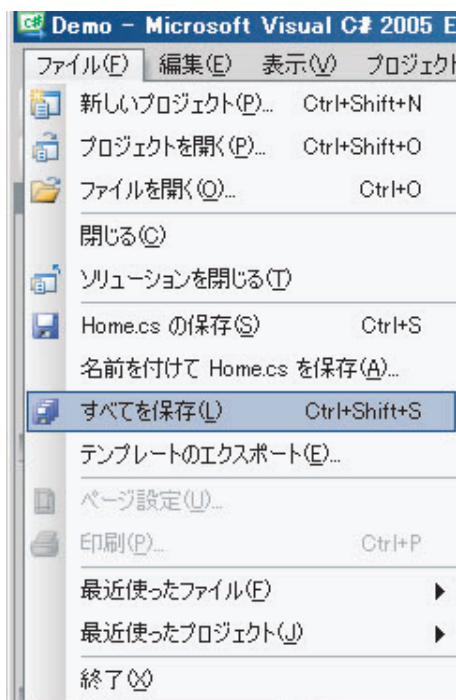
- (1) ボタンはすべて MGTButton が使われています。OnClickGoto プロパティの飛び先は登録しておりません。
- (2) アニメーション表示するため、MGTPictureBox が使われています。
- (3) アニメーション表示が停止している時に Offhook ボタンを押すと一定の周期で MGTPictureBox の絵が切り替わります。この処理はユーザが作成する main.c プログラムで Offhook ボタンの検出イベント (MGT_HOOK_ONCLICK) と画面の周期処理イベント (MGT_HOOK_POLLING) の組み合わせで実現しています。
- (4) アニメーション表示の際に Onhook ボタンを押すとアニメーション表示が停止します。停止状態で更に Onhook ボタンを押すと Home フォーム (画面) に遷移します。Onhook ボタンには 2 つの機能が割り当てられているため、C# の OnClickGoto プロパティを使ったフォームの遷移を行うことができません。これはユーザが作成する main.c プログラムの Offhook ボタンの検出イベント (MGT_HOOK_ONCLICK) 内で状態を判断して、mgt_transit()関数で Home フォーム (画面) に遷移しています。



Photo～Photo05, Web01～Web03 は Home フォーム (画面) と同じ仕組みで実装していますので、説明は省略します。詳細はサンプルコードを参照ください。

3.7 デザインの保存

フォーム (画面) デザインが完了したら「すべてを保存 (L)」ですべてのファイルを保存します。VisualC#を終了するには「終了 (x)」を選択してください。



なお, VisualC#上ではビルドを行いません。生成した C# のソースと画像データは GUI プログラム生成ツール (guigen.exe) で C ソースとヘッダファイルに変換されます。

ビルドは VisualC++ のコマンドライン上で行います。

3.8 デザインの修正

VisualC#の終了後に再度デザインを修正する場合は保存先のフォルダ下の***.sln (Microsoft Visual Studio Solution) をクリックすると作成したデザインが立ち上がります。

3.9 VisualC#のより詳しい使い方について

VisualC#のより詳しい使い方を知りたい方は VisualC# の Help メニュー, Microsoft のホームページあるいは市販の解説本等を参考にしてください。

4. Windows 上で実行可能な EXE ファイルの生成

Windows 上で実行可能な EXE ファイルを生成するためには下記の 2 つの内容をユーザが用意する必要があります。

(1) Visual C# で GUI 画面の WYSIWYG デザインを行い C# のソースと画像データを自動生成します。

(2) C 言語の userevent.c ソース(*1)にイベント処理を追加します。

この 2 つをサンプルのプロジェクトフォルダにコピーして VisualC++ のコマンドラインから MAKE.BAT を実行すると Windows で動作する EXE ファイルが生成されます。

【注】 *1 別紙の GUI 開発ツールユーザズマニュアルでは main.c にユーザのイベント追加するように説明されておりますが、Windows 用の main.c をそのまま H8S/2456 の High-performance Embedded Workshop 環境にコピーするといろいろと不都合が生じますので、Windows 用、H8S/2456 用に同等の機能を持った main.c のサンプルを用意しております。ユーザが作成するイベントコードは main.c には記述せずに userevent.c ソース内に追加するようにしてください。

表 1 サンプルプロジェクトフォルダファイル一覧

ディレクトリ	ファイル名	分類	説明
bin	guigen.exe		GUI プログラム生成ツール
bin	w9xpopen.exe		
bin	MSVCR71.dll		
include	lcd.h	ヘッダファイル	修正不要
include	mgt.h	ヘッダファイル	修正不要
lib	LCD.lib	ライブラリ	ビルド時に指定する
lib	MGT.lib	ライブラリ	ビルド時に指定する
lib	SDL.lib	ライブラリ	ビルド時に指定する
lib	SDLmain.lib	ライブラリ	ビルド時に指定する
lib	WGP16.lib	ライブラリ	ビルド時に指定する
lib	LCD.dll	DLL	生成した EXE の実行時に必要となる
lib	SDL.dll	DLL	生成した EXE の実行時に必要となる
lib	WGP16.dll	DLL	生成した EXE の実行時に必要となる
SampleProject	userevent.c	C ファイル	必要に応じてユーザがイベント処理を追加してください
SampleProject	MAKE.BAT	バッチ	VisualC++ コマンドラインから実行すると Windows で動作する EXE ファイルを生成する
SampleProject	Makefile	Makefile	VisualC++ の NMAKE ファイル
SampleProject	main.c	C ファイル	メインプログラムです。修正不要です。
SampleProject	gui.out.h	ヘッダファイル	GUI プログラム生成ツールが自動生成する
SampleProject	gui.out.c	C ファイル	GUI プログラム生成ツールが自動生成する
SampleProject	gui.out.img????	画像データ (末尾の????は数字)	GUI プログラム生成ツールが自動生成する
SampleProject ¥GUIbuilder_demo _C#_design		C# プロジェクト	VisualC# で生成したフォームデザインのディレクトリの内容をここにコピーする

4.1 Makefile の修正

下記に SampleProject ¥GUIbuilder_demo_C#_design の下に C#のソースファイル (***.cs) が存在している場合の Makefile の例を示します。パスが異なる場合は

```

$(GUIGEN) --output . GUIbuilder_demo_C#_design
のパスを修正する必要があります。
-----

# -*- mode: makefile; coding: sjis-dos -*-
# Sample NMAKE Makefile

LD = link
RM = -DEL
CP = -COPY
GUIGEN = ..¥bin¥guigen

CFLAGS = /c /O2 /FD /EHsc /MD /nologo ¥
        /D "__WIN32__" /D "NDEBUG" /D "_WINDOWS" /D "_UNICODE" /D "UNICODE" ¥
        /I ..¥include

#LDFLAGS = /SUBSYSTEM:WINDOWS /MACHINE:X86 /NOLOGO
LDFLAGS = /SUBSYSTEM:console /MACHINE:X86 /NOLOGO

LIBDIR = ..¥lib
LIBS = MGT.lib WGP16.lib LCD.lib SDL.lib SDLmain.lib

TARGET = GUIbuilder_demo.exe

all: $(TARGET)

$(TARGET): main.obj userevent.obj gui.out.obj WGP16.dll LCD.dll SDL.dll
    $(LD) $(LDFLAGS) /OUT:$(TARGET) main.obj userevent.obj gui.out.obj ¥
    /LIBPATH:$(LIBDIR) $(LIBS)

WGP16.dll LCD.dll SDL.dll:
    $(CP) ..¥lib¥*.dll .

main.obj:main.c
userevent.obj: userevent.c
gui.out.obj:gui.out.c

gui.out.c: guigen

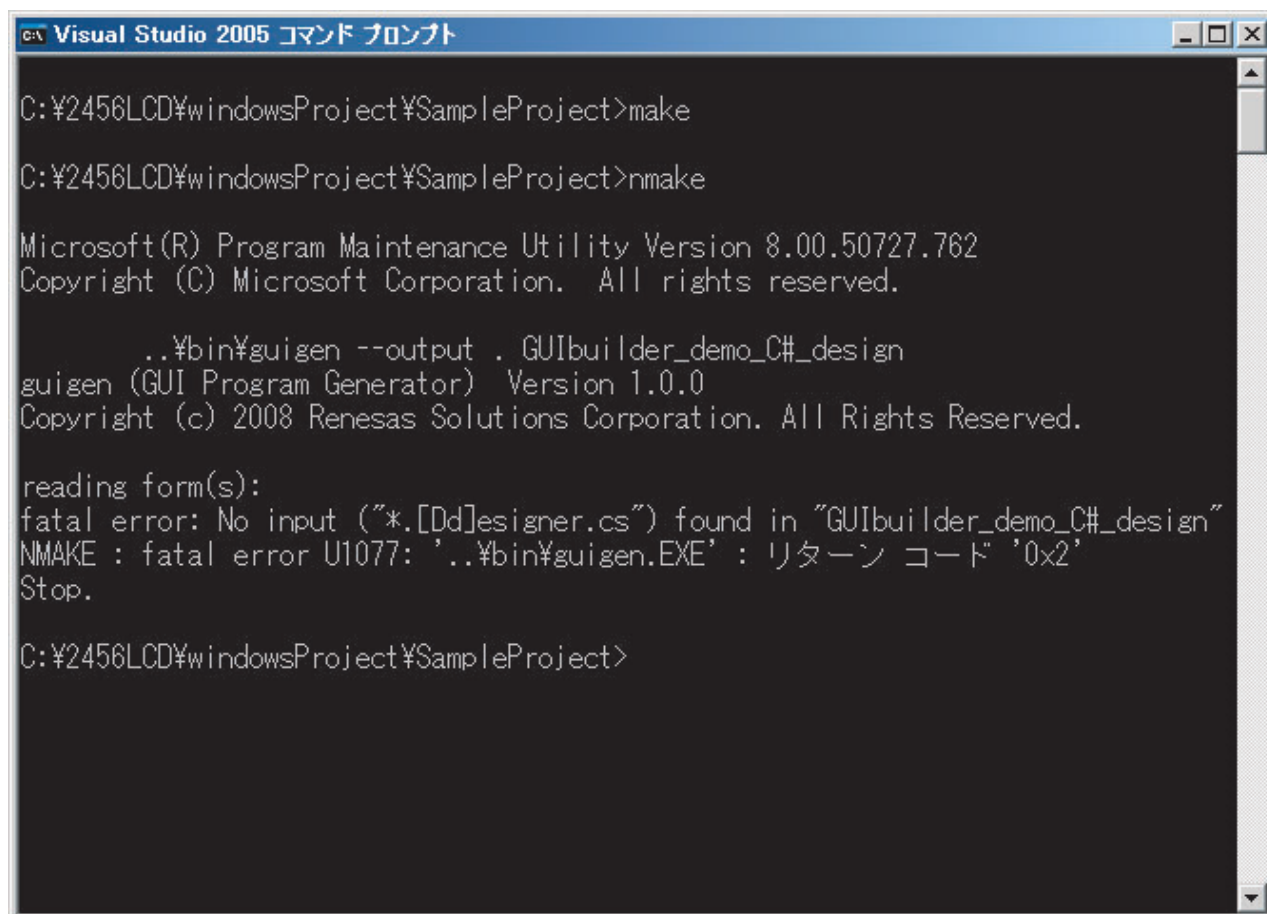
guigen:
    $(GUIGEN) --output . GUIbuilder_demo_C#_design

clean:
    $(RM) *.obj *.manifest vc80.idb

distclean: clean
    $(RM) $(TARGET) *.dll gui.out.*
-----

```

パスが間違っており、C#のソースが見つからない場合はビルド時に下記のメッセージが表示されますのでパスを修正してください。



```

Visual Studio 2005 コマンド プロンプト

C:\¥2456LCD¥windowsProject¥SampleProject>make

C:\¥2456LCD¥windowsProject¥SampleProject>nmake

Microsoft(R) Program Maintenance Utility Version 8.00.50727.762
Copyright (C) Microsoft Corporation. All rights reserved.

    ..¥bin¥guigen --output . GUIbuilder_demo_C#_design
guigen (GUI Program Generator) Version 1.0.0
Copyright (c) 2008 Renesas Solutions Corporation. All Rights Reserved.

reading form(s):
fatal error: No input ("*.Designer.cs") found in "GUIbuilder_demo_C#_design"
NMAKE : fatal error U1077: '..¥bin¥guigen.EXE' : リターン コード '0x2'
Stop.

C:\¥2456LCD¥windowsProject¥SampleProject>
  
```


4.2 userevent.c ソースの修正

userevent.c はユーザのイベント処理を追加するためのファイルです。このファイルにはデフォルトでは mgt_hook () 関数しかありません。この関数には C# のデザインで追加したボタン等のイベント処理を追加する必要があります。ただし、OnClickGoto プロパティのフォーム (画面) 切り替えのイベント処理は gui.out.c 内に自動生成されますので、mgt_hook () 関数には処理の追加が不要です。

以下に mgt_hook () 関数の記述例を示します。

```
void mgt_hook(int type, int id)
{
    switch (type) {
        case MGT_HOOK_ENTER:
            フォーム (画面) 開始時の処理を記述します。
            break;

        case MGT_HOOK_LEAVE:
            フォーム (画面) の終了時の処理を記述します。
            break;

        case MGT_HOOK_POLLING:
            周期処理を記述します。
            break;

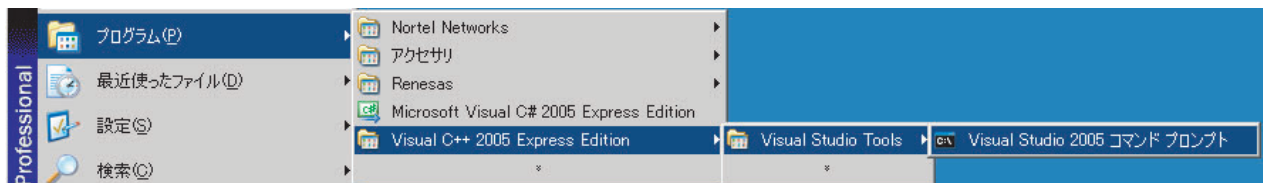
        case MGT_HOOK_ONFOCUS:
            フォーカス取得時のイベントを記述します。
            break;

        case MGT_HOOK_ONCLICK:
            ボタンのクリックイベントを記述します。
            break;
    }
}
```

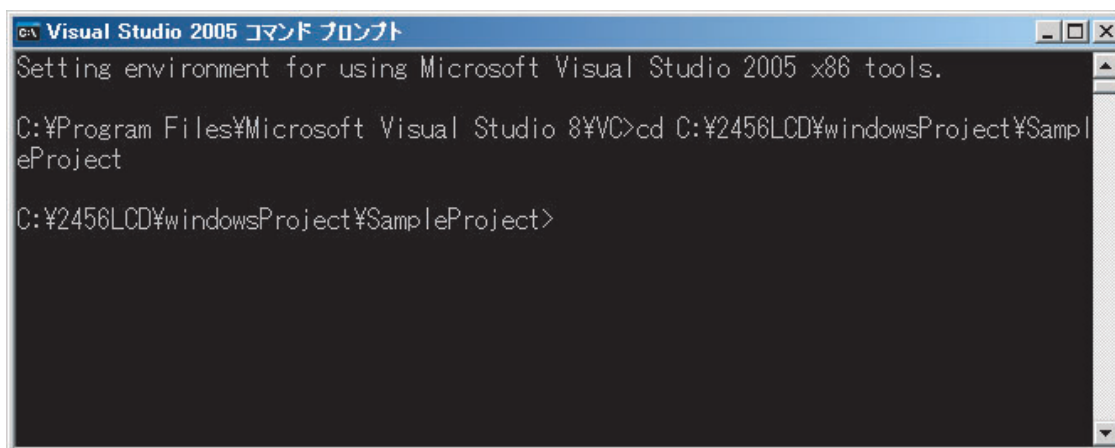
この他にも mgt_hook() 関数のイベント分岐から呼ばれるユーザ作成の関数を追加してください。より具体的な参考例は SampleProject¥ userevent.c の記述例を参考としてください。

4.3 Windows 上のビルド

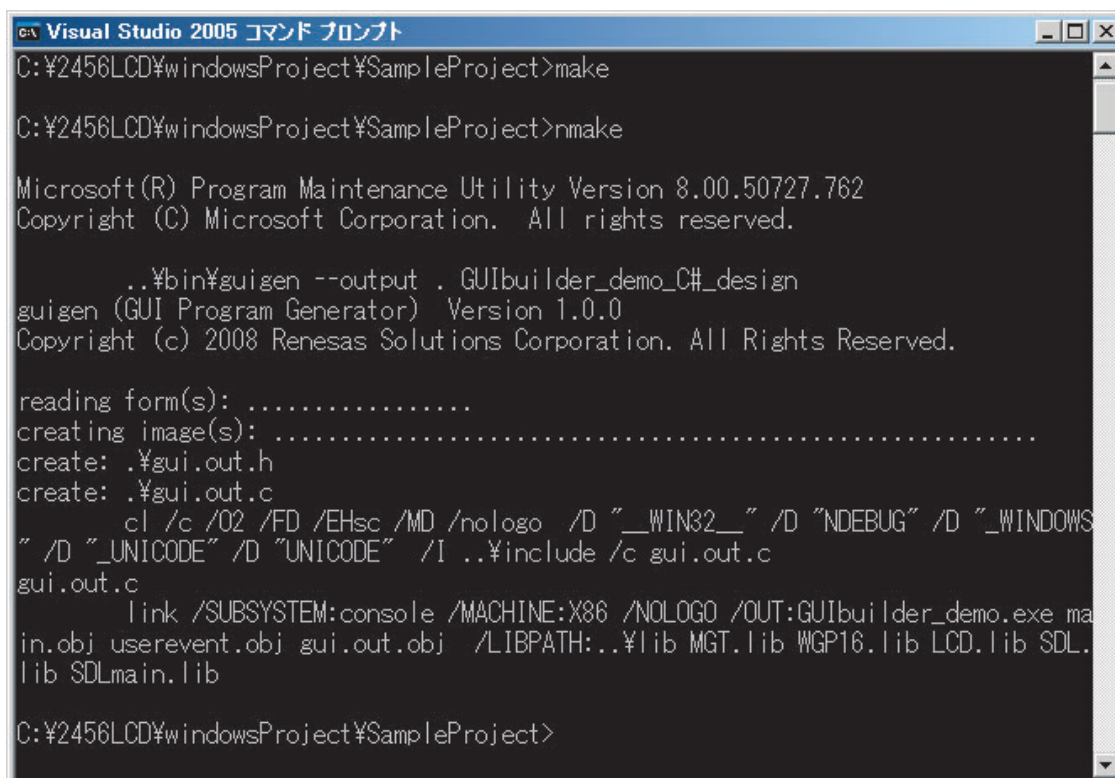
VisualC++のコマンドプロンプットを起動します。



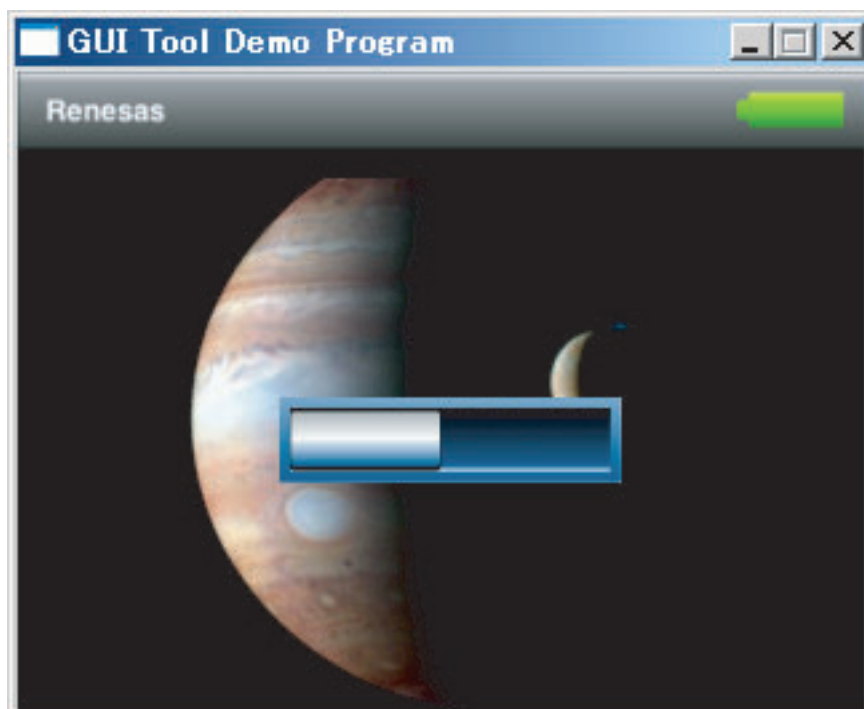
ディレクトリを 4 章の表 1 で示す SampleProject フォルダの位置にセットします。



コマンドライン上で make と打ち込み，NMAKE を起動します。



SampleProject フォルダの下に GUIbuilder_demo.exe が生成されます。同じフォルダ内に LCD.dll, SDL.dll, WGP16.dll がコピーされるので、GUIbuilder_demo.exe をクリックすればデモが動作します。



GUI プログラム生成ツールがエラーメッセージを出力して終了した場合は GUI 開発ツールユーザーズマニュアルの [9.5. エラーメッセージ] を参照して修正ください。

5. H8S/2456 実機環境での動作

図 4 に H8S/2456 のシステム構成を、表 2 に構成品の一覧を示します。開発を行うためには表 3 の内容が別途必要となります。



図 4 H8S/2456 のシステム構成

表 2 H8S/2456 のシステム構成一覧

型番	内容	備考
R0K402456D000BR (ルネサス テクノロジ)	H8S/2456CPU 基板	有償
R0K402472D010BR (ルネサス テクノロジ)	• LCD I/F ボード • バックライト接続ケーブル • 住友電工製のフラットケーブル (SML2CD-33×200-AD×10-P0.5-S4-F UL2896) • 評価ボード接続ケーブル	有償
TCG057QV1AD-G10 (京セラ製)	LCD パネル	有償
CXA-L0605A-VJL (TDK ラムダ製)	バックライト用ボード	有償
SPU30-102 (SINPRO 製)	5V- AC アダプタ	有償

表 3 開発ツール一覧

名称	内容	備考
High-performance Embedded Workshop (ルネサス テクノロジ)	H8S 用コンパイラ含む	有償
E10A エミュレータ (ルネサス テクノロジ)	—	有償
H8S 用実行ライブラリ	GUI 実行ライブラリ	有償
H8S/2456 用 High-performance Embedded Workshop 参考プロジェクト	GUI 実行環境	無償 (H8S 用実行ライブラリは除く)
外部 Flash 書き込みファイル	E10A 経由で R0K402456D000BR 上の外部 Flash を書き込むためのファイル	無償

5.1 H8S/2456 用 High-performance Embedded Workshop 参考プロジェクト

H8S/2456 用 High-performance Embedded Workshop 参考プロジェクト (2456lcd_demo) のファイル一覧を表 4 に示します。京セラ製タッチパネル付き LCD パネル用の GUI 実行環境が含まれておりますので 4.3 章で生成したファイルを一部手直ししてこのプロジェクトにコピー後にビルドするだけで H8S/2456 上で動作する GUI アプリケーションを作成することができます。詳細は 5.2 章を参照ください。

参考プロジェクトのフォルダには図 4 および表 2 の構成向けに変更済みのファイルが含まれておりますので、そのまま H8S/2456 に書き込みを行い、動作確認を行うことができます。

なお、表 4 中の(*1)のファイルは別途購入が必要となります。ルネサス テクノロジまたは特約店にご確認ください。High-performance Embedded Workshop でビルドする前に購入した製品の該当ファイルを表 4 で示すディレクトリにコピーしてください。

表 4 H8S/2456 用 High-performance Embedded Workshop 参考 Work Space (2456lcd_demo) のファイル一覧

ディレクトリ名	ファイル名	内容	備考
2456lcd_demo	2456lcd_demo.Hbp	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo	2456lcd_demo.hws	High-performance Embedded Workshop 管理ファイル	クリックすると High-performance Embedded Workshop が起動する
2456lcd_demo	2456lcd_demo.tws	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	2456lcd_demo.hwp	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	2456lcd_demo.nav	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	2456lcd_demo.pgs	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	2456lcd_demo.tps	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	commondef.h	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	dbscct.c	High-performance Embedded Workshop 生成の C ソース	—
2456lcd_demo¥2456lcd_demo	DefaultSession.hsf	High-performance Embedded Workshop 管理ファイル	—
2456lcd_demo¥2456lcd_demo	iodef_h8s2456.h	H8S/2456 レジスタ定義	—
2456lcd_demo¥2456lcd_demo	resetprg.c	リセット後に実行する	必要に応じて修正してください
2456lcd_demo¥2456lcd_demo	sbrk.c	High-performance Embedded Workshop 生成の C ソース	—

ディレクトリ名	ファイル名	内容	備考
2456lcd_demo¥2456lcd_demo	sbrk.h	High-performance Embedded Workshop 生成の C ソース	—
2456lcd_demo¥2456lcd_demo	stacksct.h	スタックサイズの定義	必要に応じて修正してください
2456lcd_demo¥2456lcd_demo	typedefine.h	High-performance Embedded Workshop 生成の C ソース	—
2456lcd_demo¥2456lcd_demo ¥Debug	2456lcd_demo.abs	デバッグ情報付ファイル	High-performance Embedded Workshop から書き込む場合に指定する
2456lcd_demo¥2456lcd_demo ¥Debug	2456lcd_demo.mot	モトローラ形式ファイル	—
2456lcd_demo¥2456lcd_demo ¥gui_usrapp	gui.out.h	guigen.exe が生成したヘッダファイル	guigen.exe が生成したファイルをそのままコピーする
2456lcd_demo¥2456lcd_demo ¥gui_usrapp	gui.out.img****	guigen.exe が生成した画像ファイル	guigen.exe が生成したファイルをそのままコピーする
2456lcd_demo¥2456lcd_demo ¥gui_usrapp	guiout.c	guigen.exe が生成した C ファイル	このファイルは修正が必要です
2456lcd_demo¥2456lcd_demo ¥gui_usrapp	main.c	メイン処理	Windows 側の main.c と共有できません。このフォルダに含まれる main.c をお使いください。
2456lcd_demo¥2456lcd_demo ¥gui_usrapp	userevent.c	イベント処理	必要に応じてユーザがイベント処理を追加してください
2456lcd_demo¥2456lcd_demo ¥H8S_LIB	libmgtH8S.lib	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥H8S_LIB	libwgp16r.lib	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥include	lcd.h	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥include	mgt.h	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥include	wgp.h	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥include	wgp_chr.h	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥include	wgp_depth.h	H8S 用実行ライブラリ	(*1)別途購入が必要
2456lcd_demo¥2456lcd_demo ¥LCD	DirectLCD_CNF_Kyocera.h	LCD 関連の設定値	京セラ LCD 用
2456lcd_demo¥2456lcd_demo ¥LCD	gui_drv.c	GUI 用 low level ドライバ	—
2456lcd_demo¥2456lcd_demo ¥LCD	hwsetup.c	H8S/2456 レジスタ設定	—
2456lcd_demo¥2456lcd_demo ¥LCD	lcd_common.h	LCD 関連の設定値	—

ディレクトリ名	ファイル名	内容	備考
2456lcd_demo¥2456lcd_demo¥LCD	LCD_driver.h	LCD 表示ドライバ	京セラ LCD 用
¥LCD	LCD_Kyocera.c	LCD 表示ドライバ	京セラ LCD 用
2456lcd_demo¥2456lcd_demo¥ TouchScreen	Calib.c	タッチパネルドライバ	京セラ LCD 用
2456lcd_demo¥2456lcd_demo¥ TouchScreen	Calib.h	タッチパネルドライバ	京セラ LCD 用
2456lcd_demo¥2456lcd_demo¥ TouchScreen	TouchScreen2.c	タッチパネルドライバ	京セラ LCD 用
2456lcd_demo¥2456lcd_demo¥ TouchScreen	TouchScreen2.h	タッチパネルドライバ	京セラ LCD 用
2456lcd_demo¥2456lcd_demo¥ TouchScreen	TouchScreen.c	タッチパネルドライバ	京セラ LCD 用
2456lcd_demo¥2456lcd_demo¥ TouchScreen	TouchScreen.h	タッチパネルドライバ	京セラ LCD 用

5.2 ファイルの修正

High-performance Embedded Workshop 上でのビルド前に表 5 に示す 4.3 章で生成した SampleProject フォルダ下のファイルを表 4 で示す H8 側の 2456lcd_demo¥2456lcd_demo¥gui_usrapp フォルダの下にコピーします。これ以外のファイルは基本的に修正が不要です。必要があれば修正ください。

gui.out.h と gui.out.img****はそのままフォルダにコピーします。

gui.out.c と userevent.c は手直しが必要となりますので、それぞれ 5.2.1 章および 5.2.2 章を参照ください。

なお、Windows 用の main.c を H8S/2456 側で利用するといろいろと不都合が生じるため、main.c は SampleProject¥main.c のものを使ってください。

表 5 コピー対象の SampleProject フォルダ下のファイル

ファイル名	内容	処理
gui.out.h	C#から生成したヘッダファイル	Windows 用をそのままコピーする
gui.out.img****	画像ファイル	Windows 用をそのままコピーする
gui.out.c	C#から生成した C ソース	(1) "."が 2 つ含まれると High-performance Embedded Workshop 上でコンパイルエラーとなるので gui.out.c の名称を guiout.c に変更する。 (2) 画像ファイルを外部 Flash に割り当てる。
userevent.c	ユーザ作成のイベント処理	Windows 用をそのままコピーする mgt_hook () 内のデバッグ出力用の printf()は H8S 側がサポートしていないので、コメントアウトする。

5.2.1 gui.out.c ファイルの修正

gui.out.c は guigen.exe ツールが C# から自動生成した C ソースです。H8S/2456 の High-performance Embedded Workshop 環境でビルドするためには下記の 2 点の修正が必要となります。

- (1) High-performance Embedded Workshop が "." を 2 つ以上含むファイル名を処理できないので、gui.out.c のファイル名を guiout.c に変更する必要があります。H8S/2456 用 High-performance Embedded Workshop 参考プロジェクト (2456lcd_demo) には guiout.c で登録してあります。
- (2) gui.out.c にインクルードされる画像ファイル (gui.out.img****) を外部 Flash エリアにマッピングしてください。H8S/2456 は Flash 容量が 128KB または 256KB の品種があります。画像データを多く使用すると内蔵 Flash に収めることができませんので、この場合は外部 Flash メモリに割り付けるようにしてください。このアプリケーションノートで使用している R0K402456D000BR ボードには 0x00400000 番地以降に Spansion 製の 4Mbytes NOR Flash (S29JL032H70TFI410) が搭載されております。
High-performance Embedded Workshop 上で外部 Flash メモリ空間にセクション名 ([EXT_FLASH](#)) が設定されておりますので、gui.out.c にインクルードされる画像ファイル (gui.out.img****) の部分を [#pragma section EXT_FLASH](#) と [#pragma section](#) // [EXT_FLASH](#) で挟んでください。
なお、VisualC# 上のデザインで画像データを指定しなかった場合には画像ファイル (gui.out.img****) は作成されません。

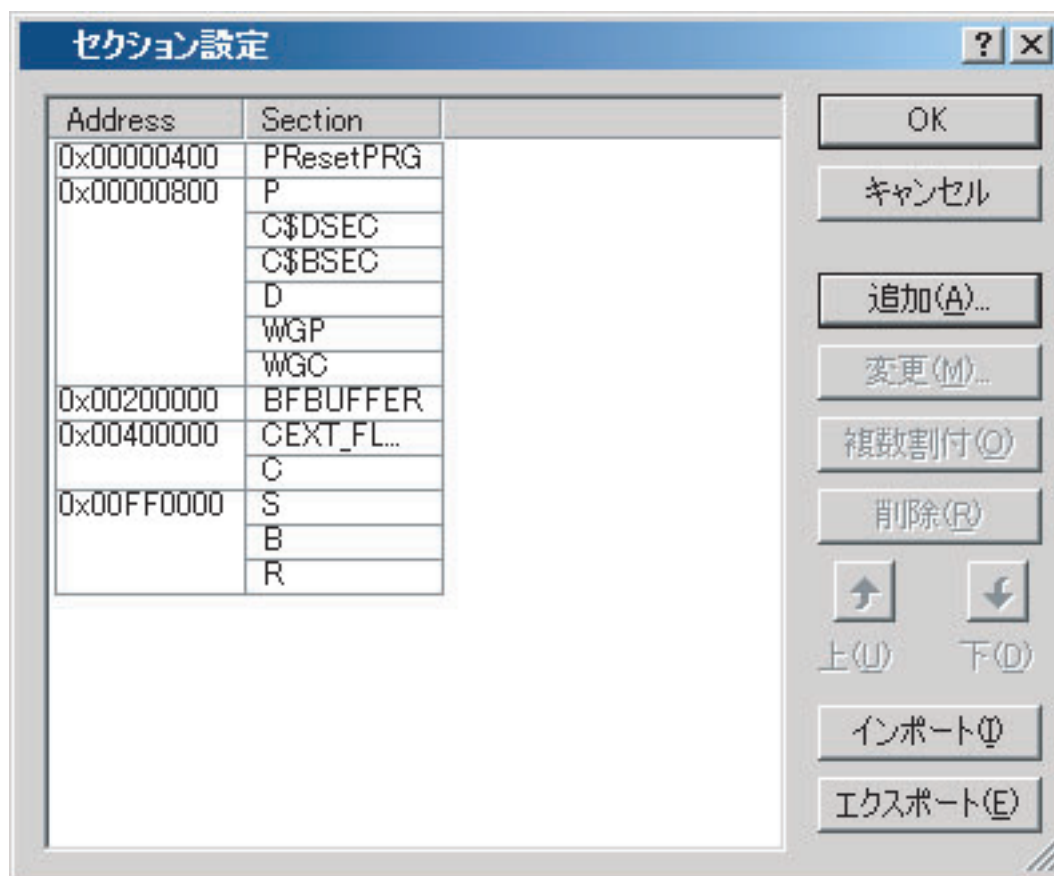


図 5 外部 Flash メモリ空間にセクション名

【gui.out.c の変更前】

```
/* -*- encoding: sjis -*- */
/* generated by guigen (GUI Program Generator) Version 1.0.0 */

#include <stddef.h>
#include "mgt.h"
#include "gui.out.h"

static const unsigned short mgt_img0001[] = {
#include "gui.out.img0001"
};
```

上記と同等の記述が画像ファイルの数だけ存在します。

```
static const unsigned short mgt_img0057[] = {
#include "gui.out.img0057"
};
```

【gui.out.c の変更後】

```
/* -*- encoding: sjis -*- */
/* generated by guigen (GUI Program Generator) Version 1.0.0 */

#include <stddef.h>
#include "mgt.h"
#include "gui.out.h"
```

#pragma section EXT_FLASH

```
static const unsigned short mgt_img0001[] = {
#include "gui.out.img0001"
};
```

上記と同等の記述が画像ファイルの数だけ存在します。

```
static const unsigned short mgt_img0057[] = {
#include "gui.out.img0057"
};
```

#pragma section // EXT_FLASH

5.2.2 userevent.c ファイルの修正

- (1) printf()関数は H8S 側がサポートしていないので、mgt_hook () 関数内で Windows 側のデバッグ出力用に利用している printf()関数をコメントアウトしてください。

printf()関数を使っていると High-performance Embedded Workshop 上で下記のエラーメッセージが表示されます。

```
Phase OptLinker starting
L2310 (E) Undefined external symbol "_write" referenced in "_flshbuf"
Optimizing Linkage Editor Abort
Phase OptLinker finished
```

- (2) Windows の C++はデフォルト 32 ビット定数ですが、H8 はデフォルト 16 ビット定数のため、定数値の積算結果が 16 ビット範囲を超えると想定した動作となりません。このような場合は数値に"L"をつけて意図的に long (32 ビット) の指定を行う必要があります。

5.3 main.c ファイルについて

Windows 用の main.c を H8S/2456 側で利用するといろいろと不都合が生じるため、main.c は SampleProject¥main.c のものを使ってください。Windows 側の main.c を変更した場合は H8S/2456 側も同等の修正を行ってください。

(1) フレームバッファ 2 面は外部 RAM に割り当てる

別紙の GUI 開発ツールユーザーズマニュアルの「付録 B. PC 用メインプログラムのサンプル」にはフレームバッファが 1 面の場合の main.c の参考例が記載されておりますが、R0K402456D000BR ボードでフレームバッファを 1 面で構成するとイベントごとの画面の書き換えが見えてしまい、画面がちらついて見えます。このため、ちらつき防止のため、フレームバッファは 2 面構成を推奨します。

RGB565 フォーマット (16 ビット/ドット) で 320 × 240 のフレームバッファを 2 面使っていますので、RAM 容量は 0x4B000 必要となります。これは 0x0020000 番地以降に存在する外部 RAM 上にアサインします。セクションで FBUFFER を定義してありますので、main.c 上で下記のように指定しています。

#pragma section FBUFFER

```
static unsigned short fb1[WIDTH * HEIGHT];
```

```
static unsigned short fb2[WIDTH * HEIGHT];
```

#pragma section //FBUFFER

(2) Windows の C++ はデフォルト 32 ビット定数ですが、H8 はデフォルト 16 ビット定数のため、定数値の積算結果が 16 ビット範囲を超えると想定した動作となりません。このような場合は数値に "L" をつけて意図的に long (32 ビット) の指定を行う必要があります。

フレームバッファサイズは 16 ビット範囲を超えるため、WIDTH と HEIGHT の定数値には "L" をつけています。

(3) Windows 別紙の GUI 開発ツールユーザーズマニュアルでは Main() 関数内で mgt_fontset16 フォントをつかっておりますが、ビルド後の ROM 容量が非常に大きいため、ROM 容量が少ない H8S/2456 では mgt_fontset8 フォントを使用することを推奨いたします。

mgt_fontset16 フォントを使った場合は mgt_fontset8 フォントを使った場合に比べて 247KB 程度余分に ROM 容量が必要となります。外部に大容量の Flash を搭載する場合はこの限りではありません。

参考の 2456lcd_demo フォルダでは mgt_fontset8 を使っております。

(4) Windows 別紙の GUI 開発ツールユーザーズマニュアルではで main() 関数の while (1) ループの中でタイミング調整用に _lcd_delay() 関数を呼び出し、タイミングを調整する例が書かれておりますが、_lcd_delay() 関数は Windows 側の main.c のみで使うことを推奨します。R0K402456D000BR ボードは H8S/2456 の EXDMAC のサイクルスチールモードで LCD 表示を実現しているため、低価格で QVGA LCD 表示を実現できるメリットがある反面、外部バスアクセスはその分遅くなります。H8S/2456 側で _lcd_delay() 関数を実行すると main 関数の処理が重くなり、タッチパネルの反応が悪くなります。

5.4 High-performance Embedded Workshop セクション定義

H8S/2456 内蔵メモリの他に 0x00200000 番地以降に存在する外部 RAM と 0x00400000 番地以降の 4Mbytes NOR Flash にセクションを割り当てています。

- (1) gui.out.c にインクルードされる画像ファイル (gui.out.img****) は外部 Flash エリアに割り当てる (5.2.1 章を参照)
- (2) 定数 (セクション C) は外部 Flash エリアに割り当てています。
- (3) フレームバッファ 2 面は外部 RAM に割り当てる (5.3 章を参照)

上記(1) (2)の ROM 容量が内蔵 ROM エリアに収まる場合はセクション C とセクション EXT_FLASH の領域は内蔵 ROM にアサインしてください。

EXDMAC を使って外部 RAM と LCD 間で画像データの転送を行うことと、フレームバッファサイズが大きいため、(3)の外部 RAM の割り当ては必須です。

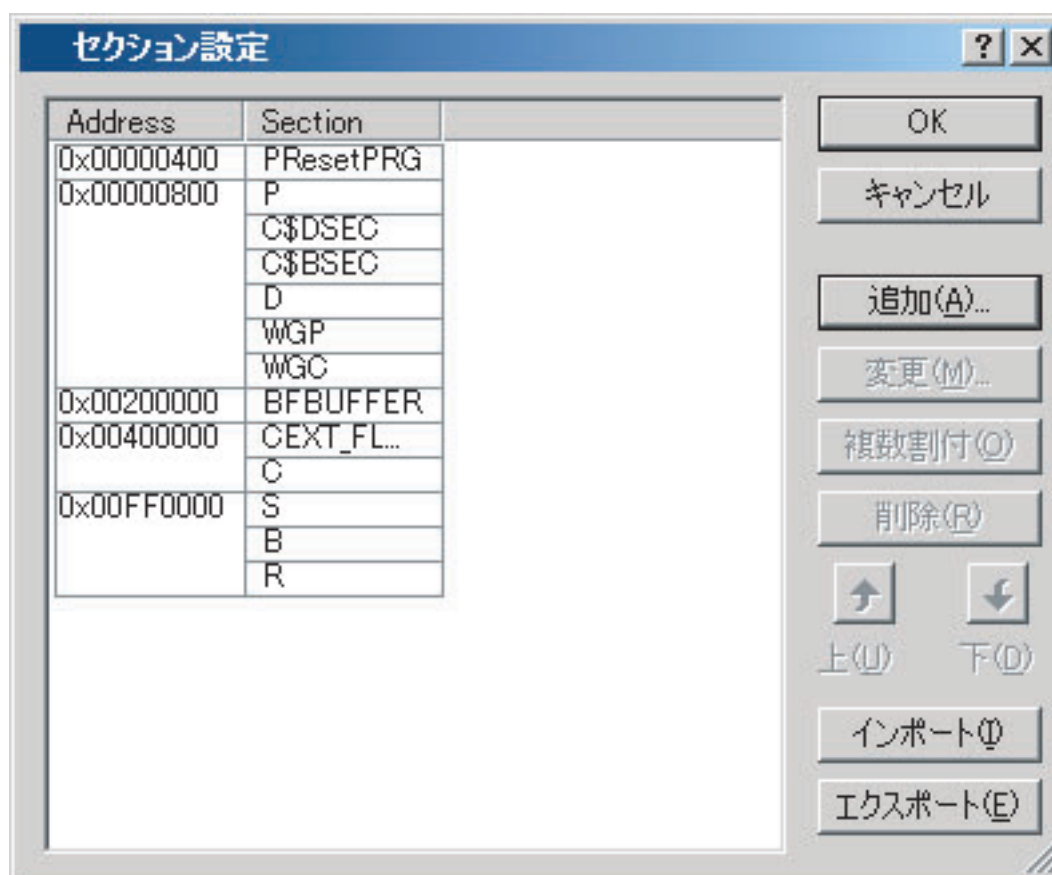


図 6 High-performance Embedded Workshop セクション定義

5.5 High-performance Embedded Workshop **上のビルド**

Build メニューの Build All または Build を選択して全ファイルをビルドしてください。High-performance Embedded Workshop の使い方については High-performance Embedded Workshop のマニュアルを参照ください。

5.6 外部 Flash 書き込み用ファイルのコピー

High-performance Embedded Workshop 上から R0K402456D000BR ボードに搭載されている Spansion 製の 4Mbytes NOR Flash (S29JL032H70TFI410) に消去と書き込みを行うためのドライバと設定ファイルを Windows 上にあらかじめコピーしておきます。コピー先はどこでもかまいません。

FMTOOL.mot のコピー先に合わせて 1MB.EFF および 3MB.EFF ファイルのパスを修正してください。

[FLASH_MODE]

USE_FLASH=1

ERASE_FLASH=1

[FLASH_MODULE]

FLIB_NAME=C:\Program Files\Renesas\ExtFlashDriver\FMTOOL.mot

FLIBOF_FLAG=0

FLIB_OFFSET=0

FLIB_TOP=ff2000

表 6 外部 Flash 書き込み用ファイル一覧

ファイル名	機能	備考
FMTOOL.mot	R0K402456D000BR ボード用 S29JL032H70TFI410 書き込みおよび消去 ドライバ	1MB.EFF または 3MB.EFF 設定ファイル から間接的に指定される
1MB.EFF	0x400000～0x4ffff までの外部 Flash 領域 の消去と書き込みに対応する設定ファイル	External Flash setting file 1MB.EFF または 3MB.EFF のいずれかを High-performance Embedded Workshop から指定する
3MB.EFF	0x400000～0x6ffff までの外部 Flash 領域 の消去と書き込みに対応する設定ファイル	同上

5.7 R0K402456D000BR ボードと E10A の接続

R0K402456D000BR ボードと E10A を図 7 のコネクタに接続します。

H8S/2456 の動作モードを内蔵 ROM 有効拡張かつエミュレータ使用モードに設定するため、E10A コネクタの隣にある DIPSW (DSW1) の 1 番と 2 番を ON に 3 番と 4 番を OFF に設定します。(図 8 参照)

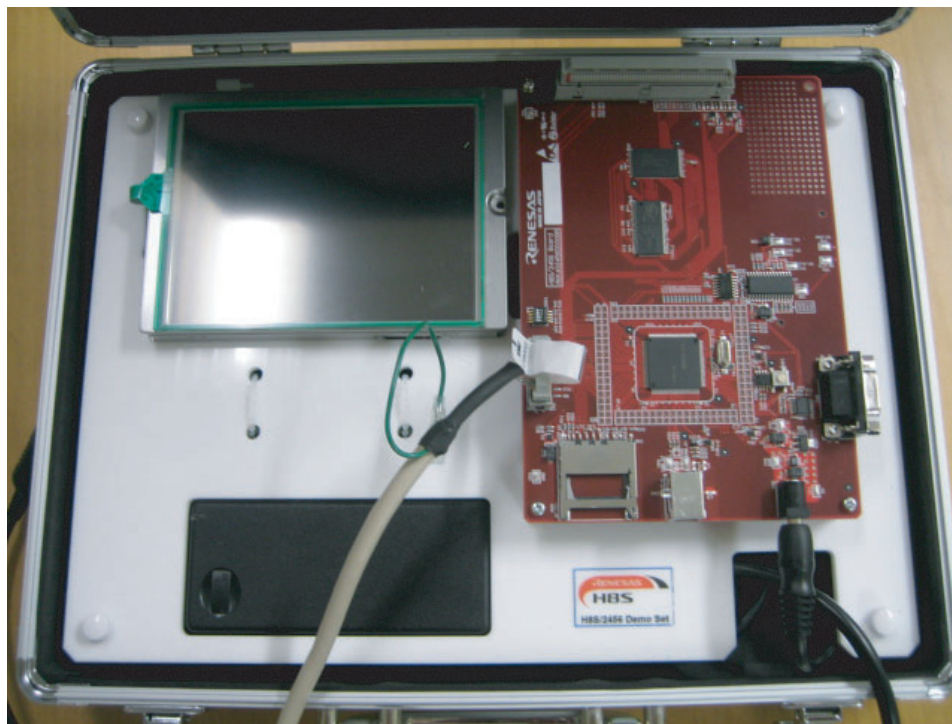


図 7 R0K402456D000BR ボードと E10A の接続

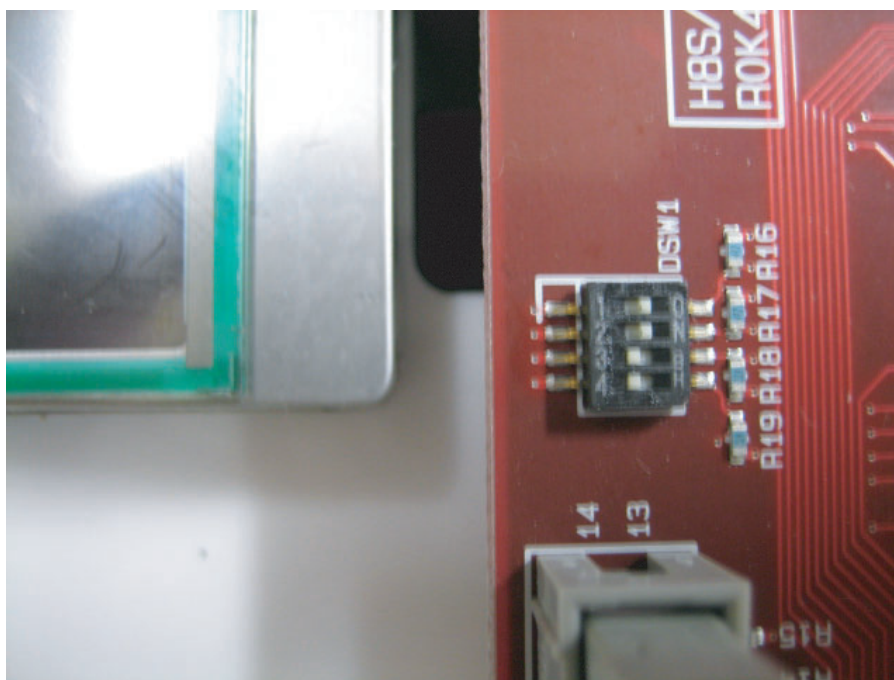


図 8 E10A 動作時の SW 設定

5.8 外部 Flash を含めたプログラムの書き込み

あらかじめ R0K402456D000BR ボードに電源を供給した状態で E10A の USB コネクタを Windows に接続しておきます。R0K402456D000BR ボードに電源が供給されていない状態で E10A の USB コネクタを Windows に接続しても E10A が認識できませんのでご注意ください。

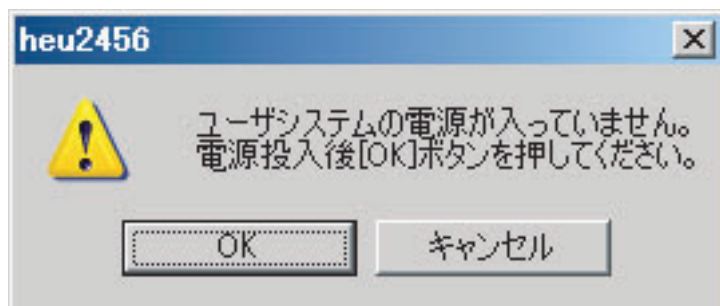


図 9 電源が供給されていない状態で E10A の USB コネクタを Windows に接続した場合

2456lcd_demo¥2456lcd_demo.hws をクリックして High-performance Embedded Workshop を起動します。図 10 で

- (1) Device は H8S/24568 を選択
- (2) Mode は E10A-USB Emulator を選択
- (3) 外部 Flash を書き込むため、Use External Flash memory setting にチェックを入れます

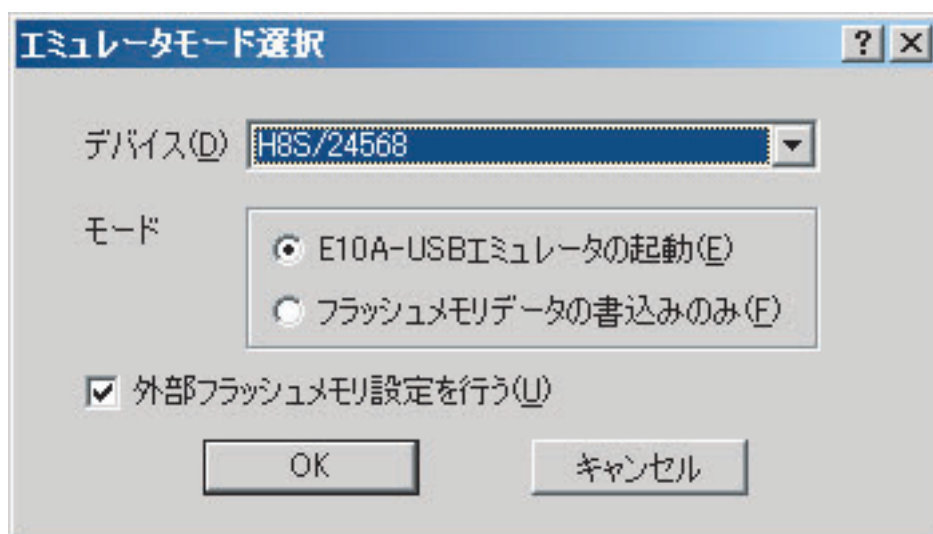


図 10 Emulator mode の選択

5.6 章であらかじめ Windows 上にコピーしておいた External Flash setting file (3MB.EFF) を指定します。このファイルは外部 Flash を書き込むための設定と外部 Flash の消去と書き込みドライバのパスが書き込まれております。

1MB までの消去と書き込みに対応した External Flash setting file (1MB.EFF) もありますが、本アプリケーションの例では外部 Flash を約 2.5Mbytes 使用しておりますので、3MB.EFF のファイルを指定してください。

図 11 の Browse ボタンを押して Select External Flash setting file で保存先を指定します。

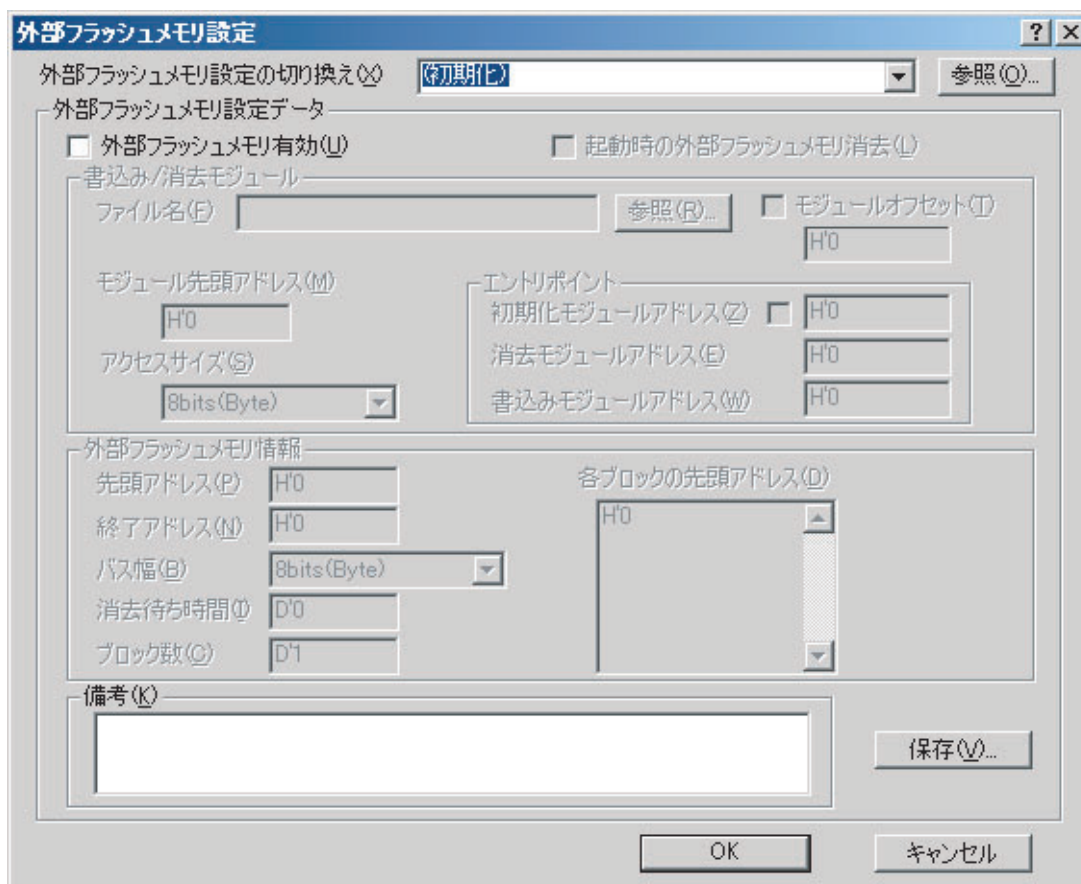


図 11 External Flash setting file の指定

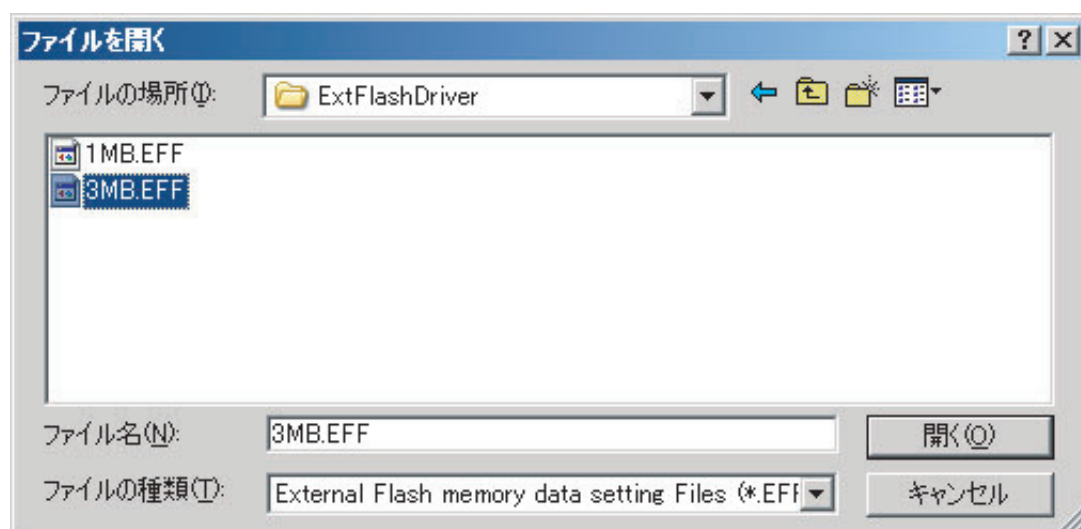


図 12 External Flash setting file の指定その 2

図 13 External Flash setting file の読み込みが完了したら OK ボタンを押します。

図 13 External Flash setting file の読み込み後

EXTAL の周波数の値 (16MHz) を設定します。

図 14 EXTAL の周波数の指定

ID Code はデフォルトのままで OK ボタンを押します。

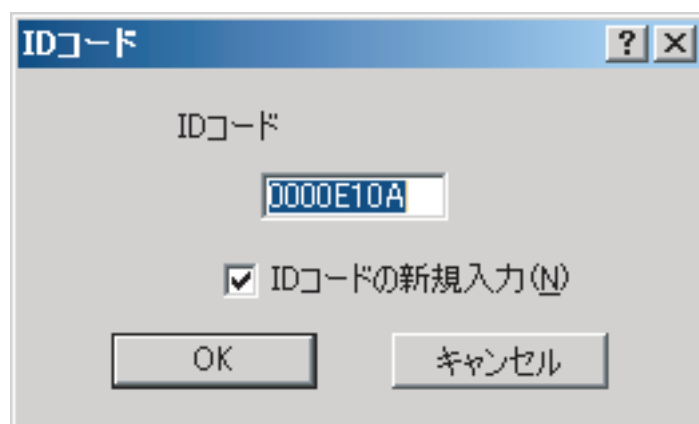
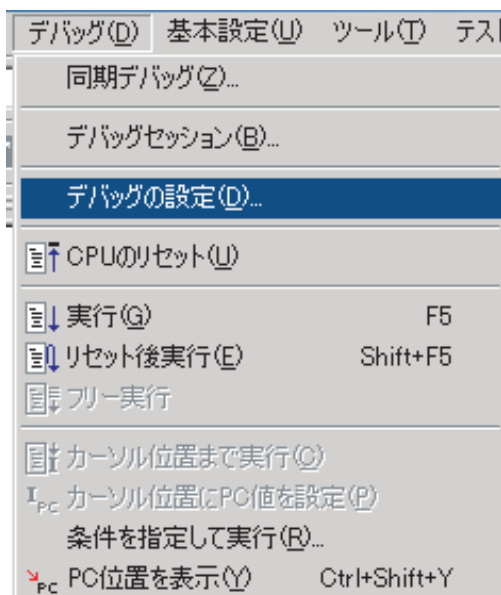


図 15 ID Code の入力



図 16 Connecting 画面

High-performance Embedded Workshop が起動したら「デバッグ(D)」メニューから「デバッグの設定(D)」を選択します。



ターゲットは H8S/2456 E10A-USB SYSTEM(CPU2600)を選択します。

デバッグフォーマットは Elf/Dwarf2 を選択します。

ダウンロードモジュールの Filename は \$(CONFIGDIR)\\$(PROJECTNAME).abs とします。

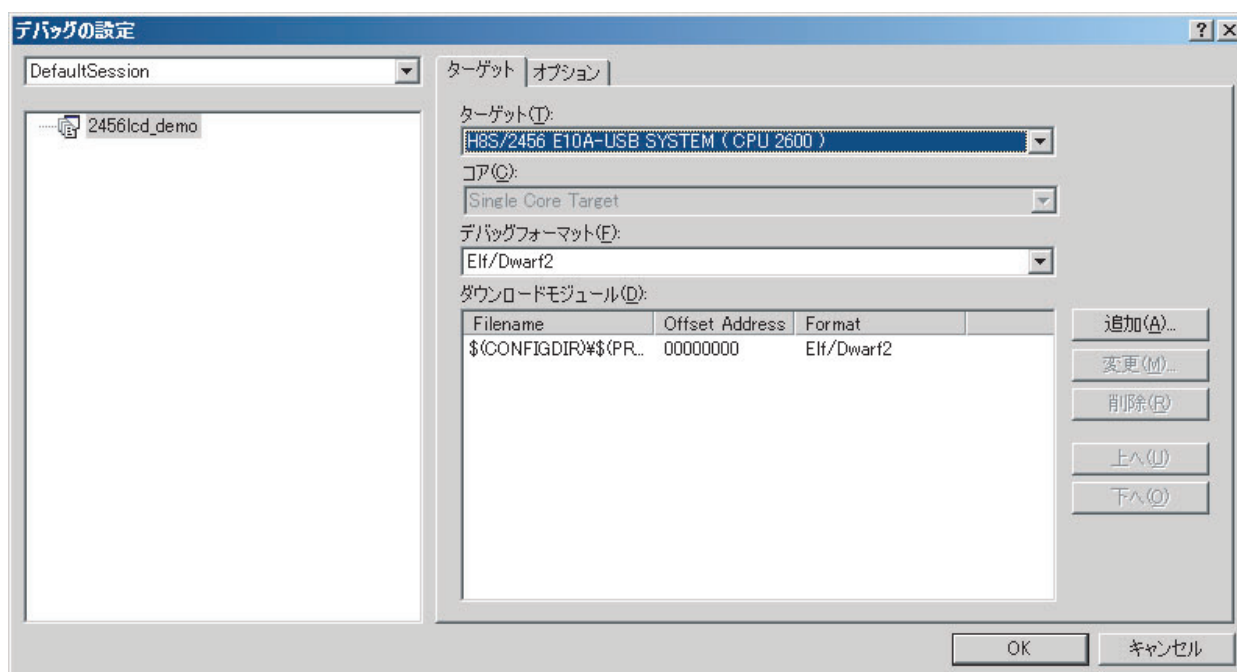


図 17 デバッグの設定画面

Debug メニューの Download modules で、5.5 章でビルド済みの 2456lcd_demo.abs を指定します。

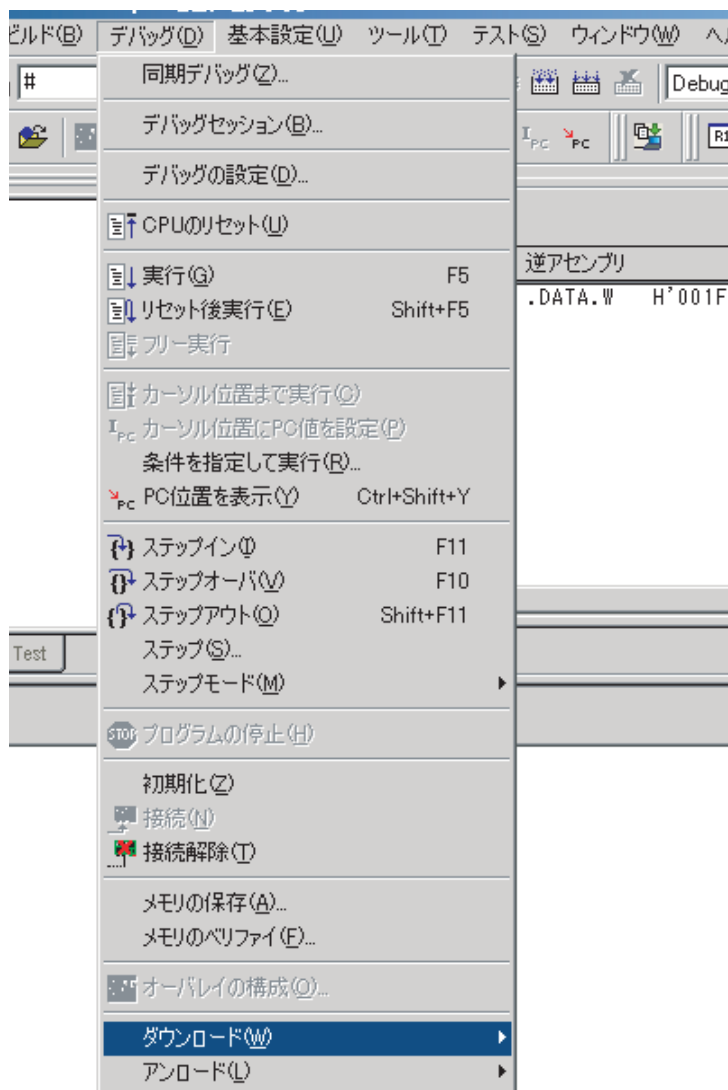


図 18 2456lcd_demo.abs ファイルの指定

reset ボタンと GO ボタンを押して内蔵 Flash と外部 Flash にプログラムを書き込みます。

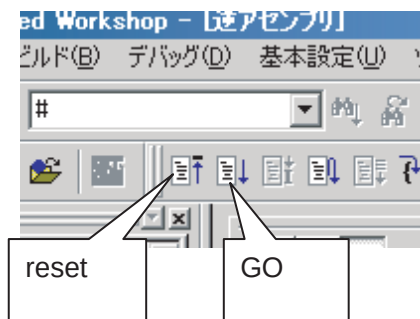


図 19 reset と GO

下記のメッセージが表示されれば書き込みが完了しています。

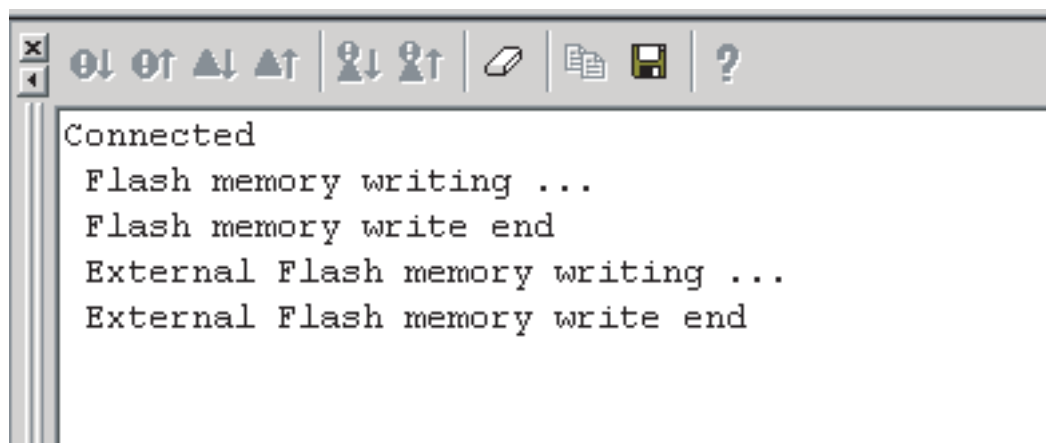


図 20 外部 Flash 書き込み完了

5.9 プログラムの実行

High-performance Embedded Workshop 上で GO ボタンを押すとプログラムの書き込み完了後に H8S 側のプログラムが実行されます。

プログラム実行後に最初にタッチパネルをキャリブレーションする必要があります。先の細いもので左上の十マークをクリックしてください。続いて右下に表示される十マークを同様にクリックしてください。

クリック位置がずれるとボタンの検出位置がずれますので、正確に行ってください。

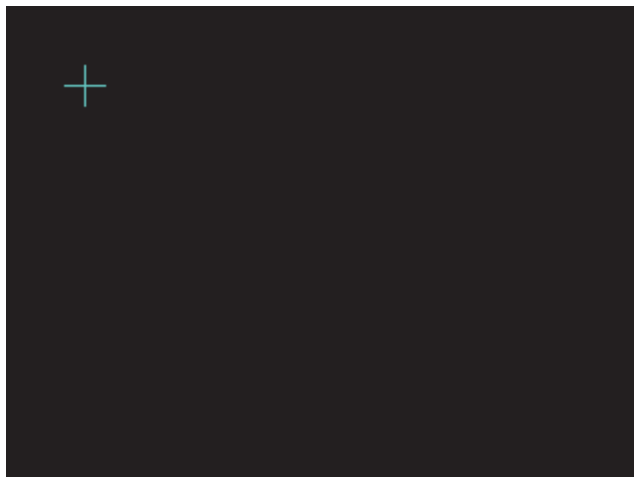


図 21 タッチパネルキャリブレーションその 1



図 22 タッチパネルキャリブレーションその 2

キャリブレーション完了後に main 画面が起動します。また、Home 画面から main ボタンを押すとこの画面に遷移します。

中央のボタンを押すと VisualC#の OnClickGoto イベントで Home 画面に遷移しますので, userevent.c ファイルには対応するイベントコードがありません。

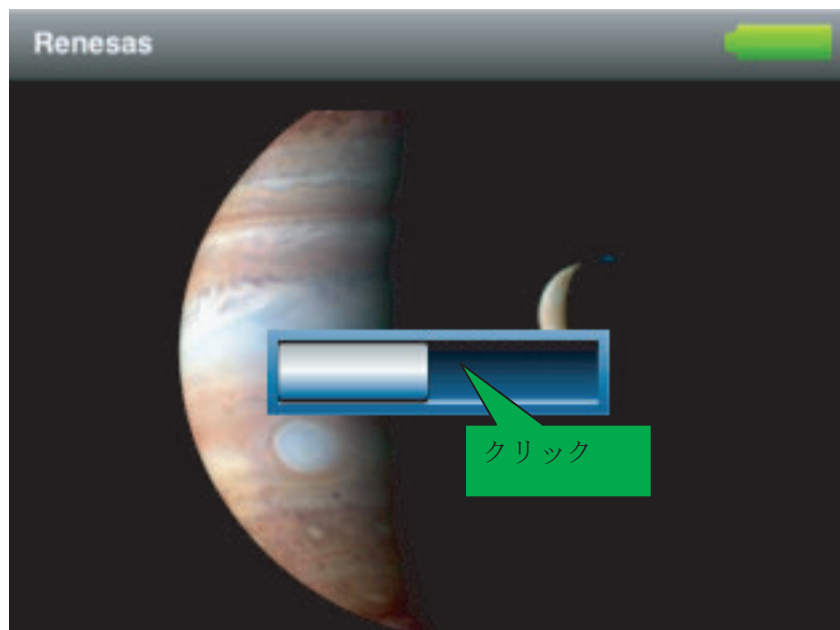


図 23 main 画面

中央のボタンをクリックすると、Home 画面が起動します。Home 画面にはボタンが5つあり、押されたボタンに対応する画面に切り替わります。このイベント処理は VisualC#の OnClickGoto イベントで実装しますので, userevent.c ファイルには対応するイベントコードがありません。



図 24 Home 画面

0～9, *, # ボタンには VisualC# の OnClickGoto イベントも useevent.c ファイルに対応するイベントコードも記述していませんので、ボタンを押しても何も起こりません。

Offhook ボタンを押すと MGTPictureBox 上にアニメーションが表示されます。この状態で onhook ボタンを押すと MGTPictureBox 上のアニメーションが停止します。アニメーションが停止している状態で onhook ボタンを押すと Home 画面に遷移します。これは useevent.c ファイル中の mgt_hook() 関数中の MGT_HOOK_POLLING (周期処理) と MGT_HOOK_ONCLICK (ボタンクリックイベント) で実現しています。



図 25 Phone 画面

Photos 画面には 6 つのボタンがあります。ボタンには小さな画像が貼り付けてあり、ボタンを押すと対応する画面 (大きな画像) が表示されます。X ボタンを押すと Home 画面に戻ります。

これらのイベント処理は VisualC# の OnClickGoto イベントで実装していますので、useevent.c ファイルには対応するイベントコードがありません。



図 26 Photos 画面

図 26 と同じ仕組みで実装されています。左方向のボタンで前の画面に戻ります。また右方向のボタンで次の画面に進みます。その他の Photos 画面は同じ仕組みで実装されていますので、説明を省略いたします。

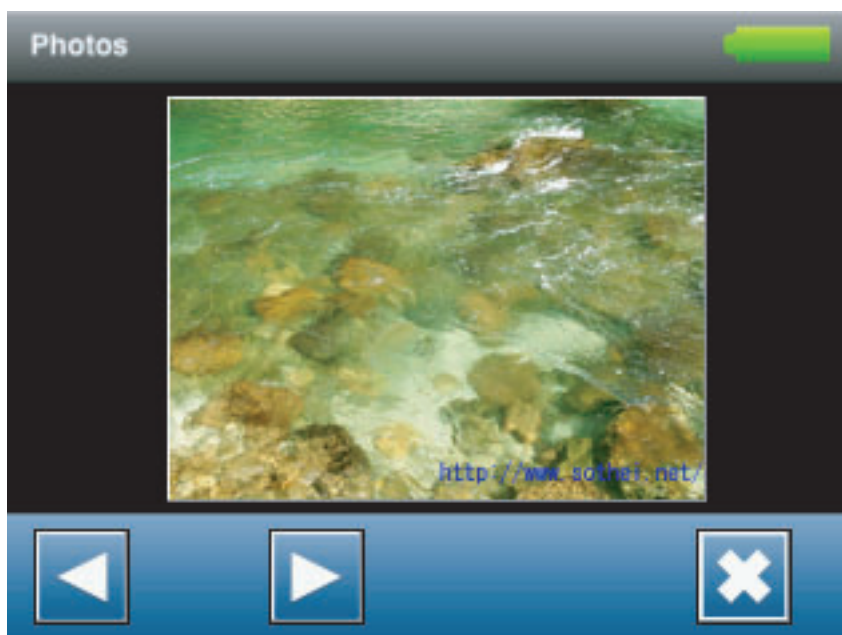


図 27 Photos 画面その 2

WWW 画面は 2～3 つのボタンと BackgroundImage 画像で構成されています。VisualC#上でフォーム（画面）の BackgroundImage プロパティに WWW サイトの画像を貼り付けています (3.5 章参照)。WWW は 3 画面ありますが、どれも図 26 と同じ仕組みで実装しています。

左方向のボタンで前の画面に戻ります。また右方向のボタンで次の画面に進みます。また、X ボタンを押すと Home 画面に戻ります。



図 28 WWW 画面

Music 選択画面は MGTPictureBox と MGTListBox とボタン 4 つで構成されています。

左方向のボタンと右方向のボタンで MGTPictureBox の画像と MGTListBox のテキストを切り替えます。再生ボタンで Music 再生画面に切り替わります。また、Music 再生画面で X ボタンを押すと Music 選択画面に戻ります。Music 選択画面で X ボタンを押すと Home 画面に戻ります。

これらは VisualC# の OnClickGoto イベントと userevent.c ファイル中の mgt_hook() 関数中の MGT_HOOK_ENTER (画面の初期化) , MGT_HOOK_POLLING (周期処理) と MGT_HOOK_ONCLICK (ボタンクリックイベント) で実現しています。



図 29 MUSIC 選択画面

ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
0.90	2009.03.16	—	初版発行
1.00	2009.09.28	27-28	表 4 の H8S/2456 用 High-performance Embedded Workshop 参考 Work Space (2456lcd_demo) のファイル一覧の構成ファイルの変更

すべての商標および登録商標は、それぞれの所有者に帰属します。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認いただきますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したものです。万一本資料の記述の誤りに起因する損害がお客様に生じた場合においても、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場合の故障および事故につきましては、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害などを生じさせないように、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなくよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444