

本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

## はじめに

本アプリケーションノートでは、SK-S7G2用の簡易2画面グラフィカル・ユーザ・インタフェース（GUI）を GUIX Studio を使用して作成するプロセスについて説明します。また、Renesas Synergy™ Software Package（SSP）を使用して簡単に新しいアプリケーションを作成・設計できることを紹介します。

Synergy Software Package には、Express Logic 社の ThreadX®リアルタイムオペレーティングシステム（RTOS）、X-Ware（NetX™、USBX™、GUIX™、FileX®）、および単体フレームワークに統合されたハードウェアドライバが含まれます。組み込まれたアプリケーションが複雑な場合でも短時間で開発できる包括的統合フレームワークがこのツールセットにより提供されます。

「Hello World」は Synergy フレームワークを使い、e<sup>2</sup>studio 内で開発されたアプリケーションです。

## 対象デバイス

SK-S7G2 ボード version 3.1

PK-S5D9 ボード version 1.0.

## PC 最小推奨構成

- Microsoft® Windows® 7
- Intel® Core™ ファミリプロセッサ 2.0 GHz 以上 (または同等品)
- メインメモリ：8GB 以上
- HDD もしくは SSD：2GB 以上の空き容量
- USB 2.0 インタフェース
- インターネット接続環境

## インストール済みソフトウェア

- Synergy e<sup>2</sup>studio 5.3.1.002 以降
- Synergy Software Platform (SSP) v1.2.0
- Synergy™ Standalone Configurator 5.3.1.002
- GUIX Studio v5.3.2.2
- IAR for Synergy 7.71.1.11989

注： 上記のうちいずれかのソフトウェアアプリケーションが無い場合は、あらかじめインストールをしてください。

## 提供するソフトウェアファイル

- guiapp\_event\_handlers.c
- main\_thread\_entry.c
- lcd\_setup.c
- lcd.h.

## 本ドキュメントの目的

本書では e<sup>2</sup>studio による、GUIX タッチスクリーンインターフェース “ Hello World ” アプリケーションの設定の仕方を説明します。ハードウェア機能 (LCD、SPI、I2C interface)、スレッドおよびメッセージバス、割り込み、LCD ドライバおよびタッチスクリーンの各種設定だけでなく、GUIX Studio エディタを使った GUI インターフェースを作成する手順のほかデバッグ動作による e<sup>2</sup>studio の基本プロジェクトセットアップ方法も説明します。e<sup>2</sup>studio 動作中、アプリケーションはタッチスクリーンの動作に反応して、GUI として機能します。

## 目次

1. 概要 .....	2
2. e <sup>2</sup> studio へのプロジェクトのインポート .....	2
3. e <sup>2</sup> studio の 統合ソリューション開発環境、ISDE (Integrated Solution Development Environment)内の新規プロジェクトの作成 .....	<b>エラー! ブックマークが定義されていません。</b>
4. ISDE 内のプロジェクト設定 .....	8
5. GUIX Studio による GUIX インターフェースの作成 .....	30
6. カスタムインターフェースコントロールのコード追加およびプロジェクトのビルド .....	43
7. アプリケーションの作動 .....	46

## 1. 概要

GUIX Studio を使用して、プロジェクトを設定し、簡易 GUI ベースのアプリケーションを開発する時の手順を説明します。

## 2. e<sup>2</sup>studio へのプロジェクトのインポート

注： この手順により開発手順を省き、SK-S7G2 もしくは PK-S5D9 で動作しているプロジェクトを検証するところから開始することが可能になります。この手順をスキップして、e<sup>2</sup>studio でプロジェクトを作成するステップ 3 に進むことができます。プロジェクトをインポートする場合はこの手順をスキップして、7 章 “ アプリケーションの作動 ” に進んでください。

本書内の開発ウォークスルーを省略し、e<sup>2</sup>studio 内で完成したプロジェクトを開く場合、[r11an0023eu0116-synergy-ssp-import-guide.pdf] に記載の e<sup>2</sup>studio へプロジェクトをインポートして、プロジェクトをビルドする手順を参照してください。添付の GUIX\_Hello\_World\_SK-S7G2.zip と GUIX\_Hello\_World\_PK-S5D9.zip ファイルには、完成プロジェクトが含まれています。

### 3. e<sup>2</sup>studio の 統合ソリューション開発環境、ISDE (Integrated Solution Development Environment)内の新規プロジェクトの作成

まず、e<sup>2</sup>studio 内に新規のプロジェクトを作成します。

1. 下記メニュー（Windows スタートメニュー 全てのプログラム ルネサスエレクトロニクス e<sup>2</sup>studio フォルダ）から e<sup>2</sup>studio アイコンをクリックして e<sup>2</sup>studio を開きます。
2. [Workspace Launcher] ダイアログボックスが表示されたら、[OK] をクリックし、デフォルトワークスペースを使用します。

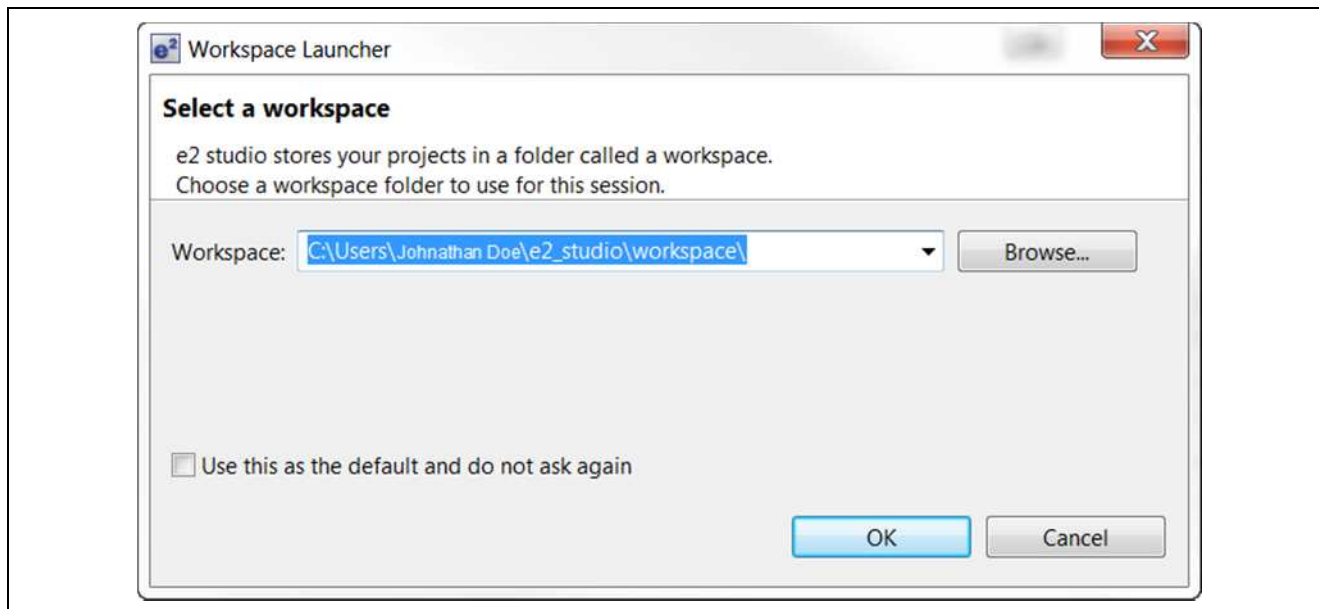


図 3.1 Workspace Launcher ダイアログボックス

3. 新規のワークスペースを作成します。  
[File] メニューのドロップダウンメニュー[Switch Workspace]>[Other...] を選択します。
4. ワークスペース名を追記します。  
[Workspace Launcher] ウィンドウで、[Workspace]名の後ろにテキスト加えユニークな名にしてください。ここでは「GUI\_APP」と入力します。デフォルトの場所にインストールする場合、新しいワークスペース名は以下の通りになります。  
[C:\Users\[your name]\e2\_studio\workspace\GUI\_APP]
5. [OK] をクリックし、新しいワークスペースを作成します。
6. Workbench 付近をクリックして Welcome 画面を終了し次に進みます。



図 3.2 Workbench 領域をクリックして Welcome 画面を閉じる

## Renesas Synergy™ プラットフォーム SK-S7G2 用 GUIX “ Hello World ”

7. 図 3.3 に示すように、ツールバーの [New] アイコンのとなりにある黒い三角形をクリックして、新しいプロジェクトをスタートさせます。

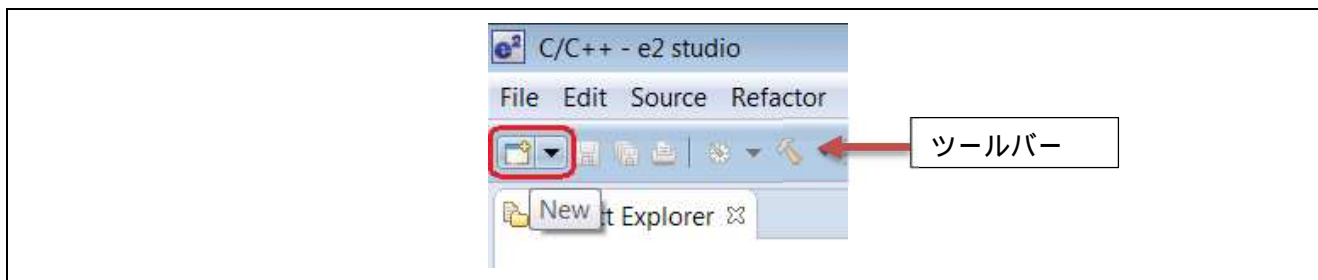


図 3.3 Start a New Project

8. [Synergy C Project] をメニューから選択する。

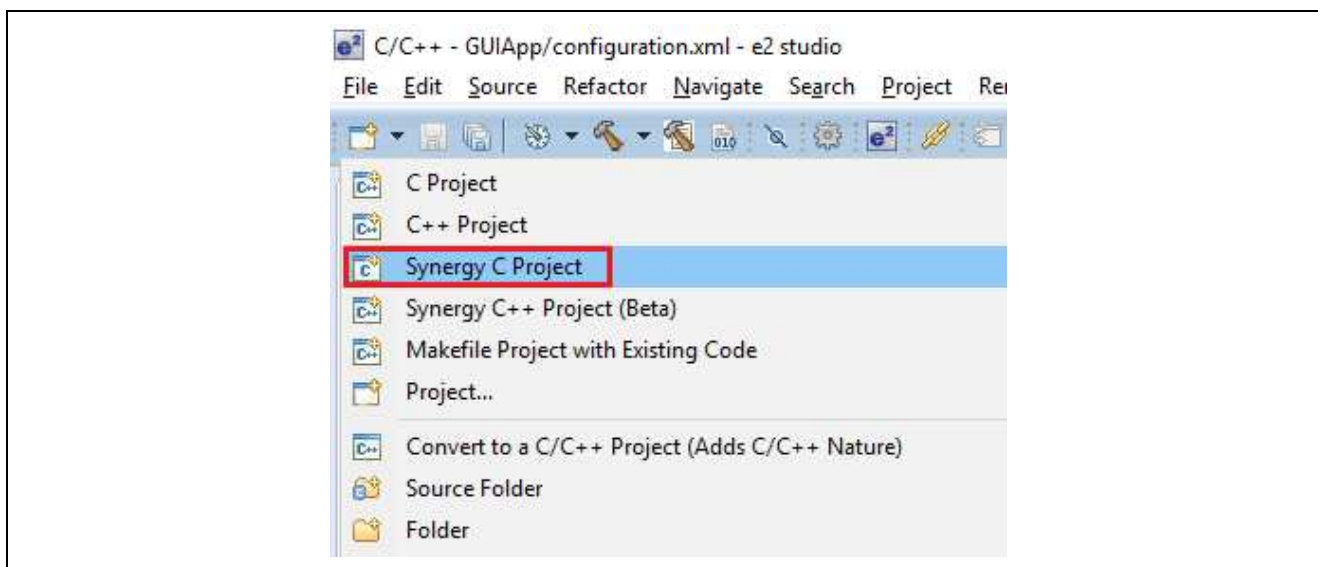


図 3.4 プルダウンメニューの Synergy プロジェクト選択

9. [License file]が設定されている場合、フォーム内のライセンスは図 3.5 のように表示されます。図 3.5 のように表示された場合、手順 10 を省きます。フォーム内が空欄である場合、下記手順 (A~G) を続けて行ってください。

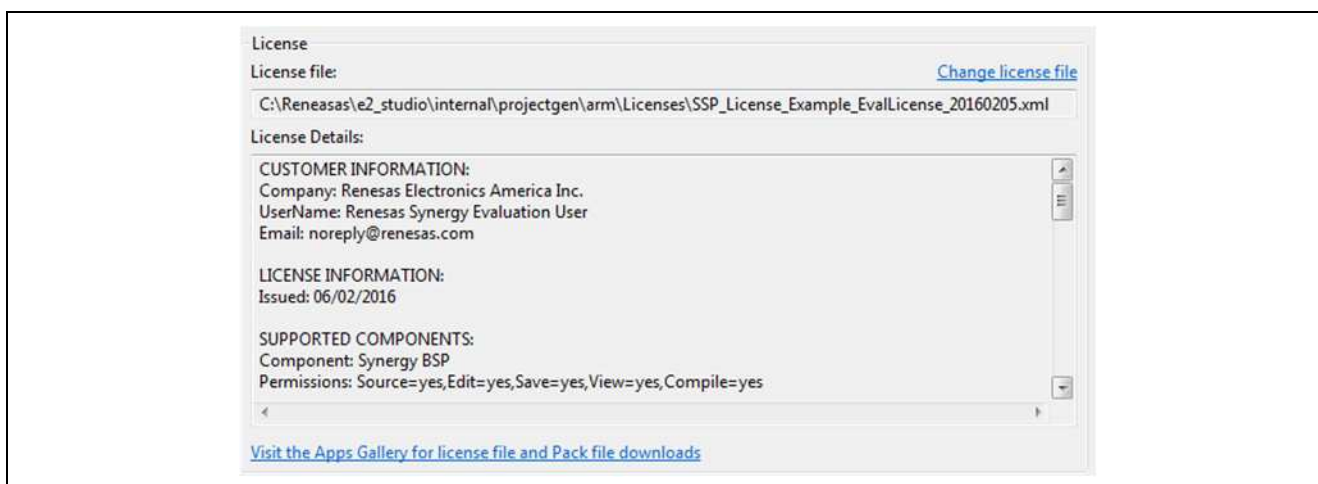


図 3.5 License file が設定されている場合

- A. [Change license file] ボタンをクリックしてください。e<sup>2</sup>studio が [Preferences] ダイアログボックスを表示します。

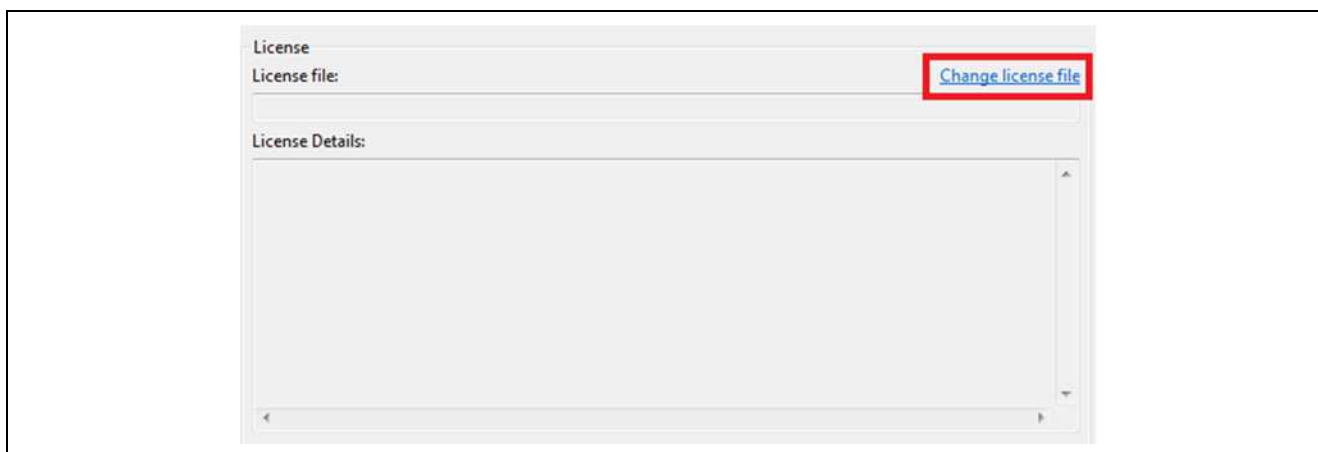


図 3.6 ライセンスファイルが設定されていない場合

- B. ブラウズ [...] ボタンをクリックし、[Specify Synergy License] ダイアログボックスを開いてください。

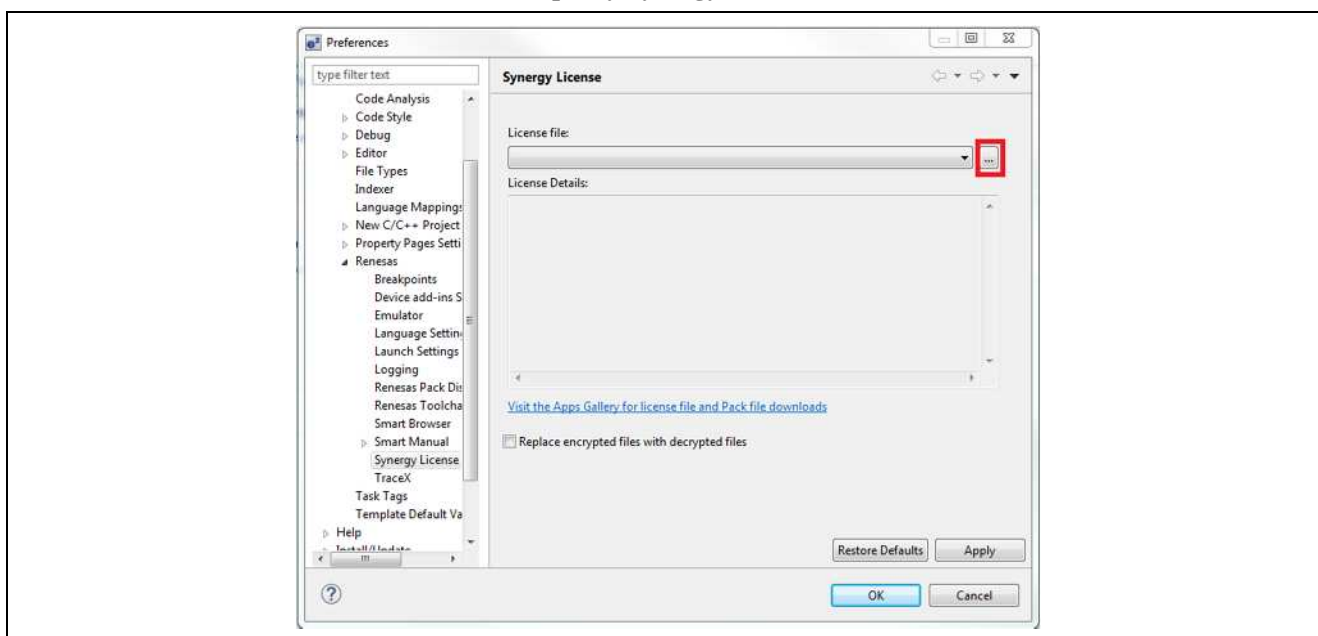


図 3.7 Synergy ライセンス設定をする Preferences ダイアログボックス

- C. [Browse...] ボタンをクリックしてください。e<sup>2</sup>studio が、[Open] ダイアログボックスを表示し、ライセンスのディレクトリを表示します。

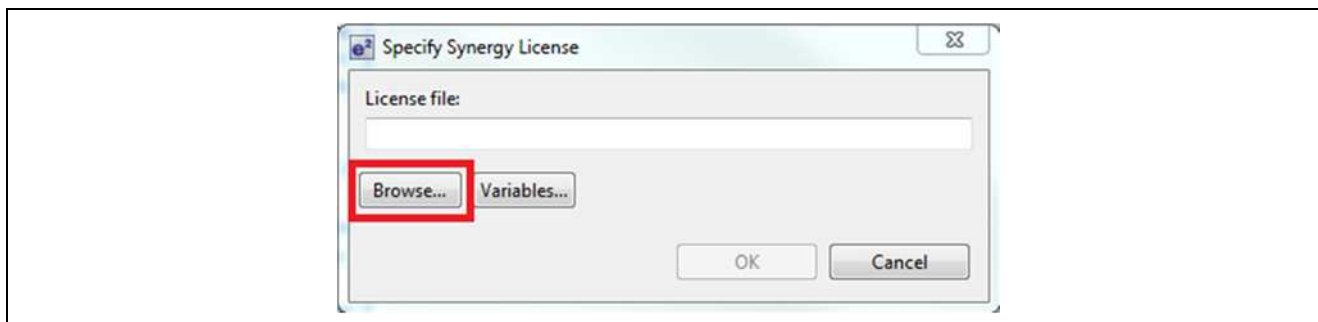


図 3.8 Synergy ライセンスダイアログボックス

## Renesas Synergy™ プラットフォーム SK-S7G2 用 GUIX “ Hello World ”

注： e<sup>2</sup>studio をデフォルト位置にインストールした場合は、ライセンスファイルは以下のディレクトリになります。

[C:\Renesas\e2\_studio\internal\projectgen\arm\Licenses]

- D. ディレクトリ内の [SSP\_License\_Example\_EvalLicense\_\*.xml] を選択してください。
- E. [Open] をクリックし、ライセンスファイルを選択してください。
- F. [OK] をクリックし、ライセンスを設定したら、ダイアログボックスを閉じてください。

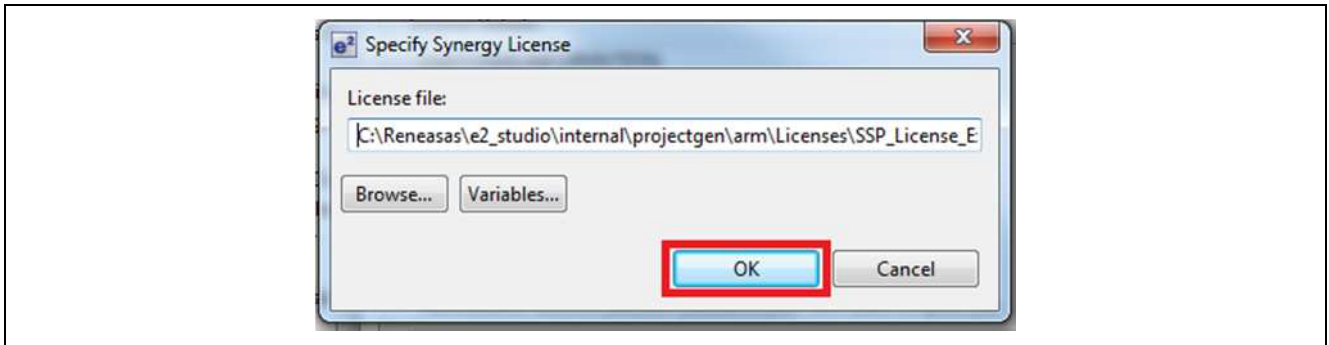


図 3.9 ライセンスファイルの確認

- G. [Preferences]ダイアログボックスの [Apply] をクリックし、[OK] をクリックしてください。

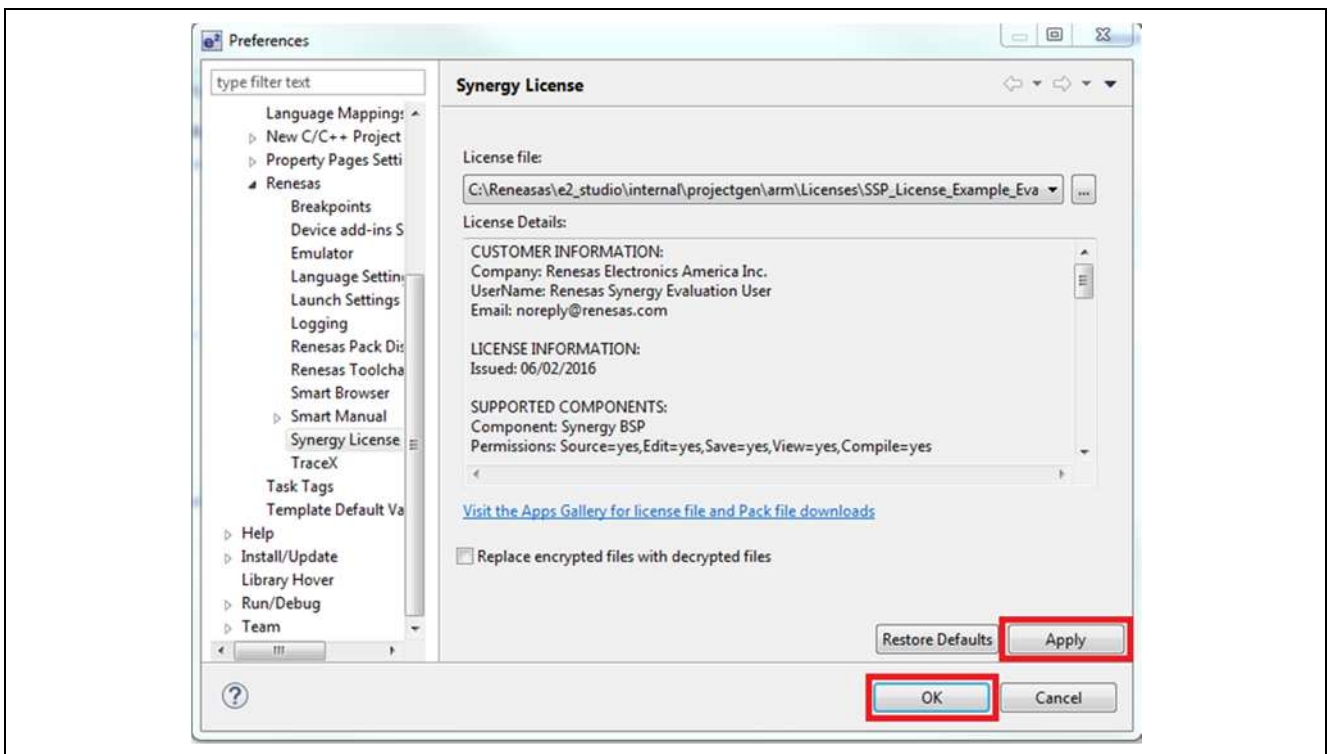


図 3.10 Synergy ライセンスファイル選択の適用と確認

- 10. [Project name] テキストフィールドにプロジェクト名を入力してください。例：「 GUIApp 」

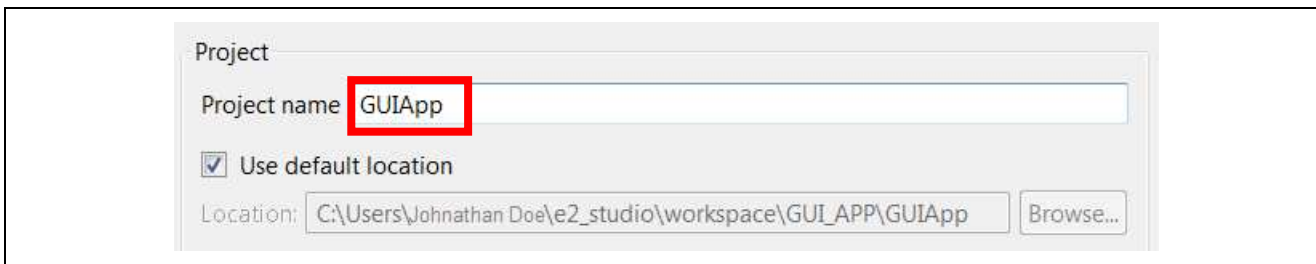


図 3.11 プロジェクト名の入力

11. このページの右上部で、[Toolchain] オプションが[GCC ARM Embedded] に設定されていることを確認してください。

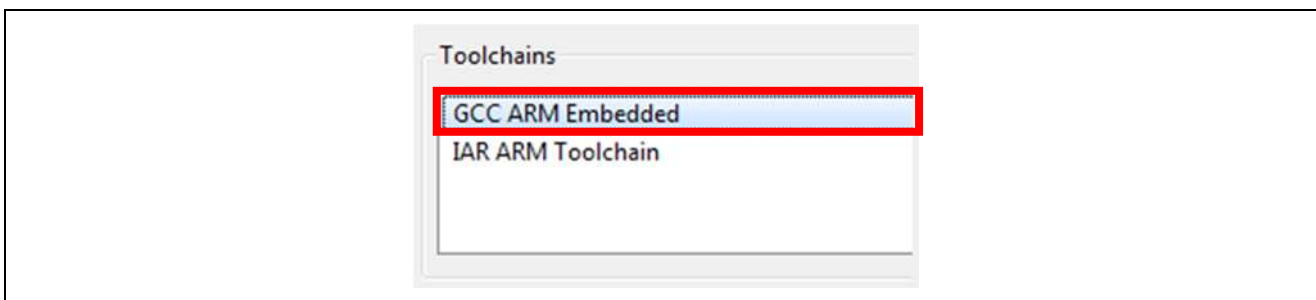


図 3.12 GCC ARM Embedded ツールチェーンの確認

12. [Next] をクリックしてください。  
— [Device Selection] (左上) の下、[SSP version] 欄は[1.2.0] (またはそれ以降) を選択してください。
13. [Board] 欄に関しては、[S7G2 SK] を選択してください。[Device] 欄は自動的にアップデートされます。

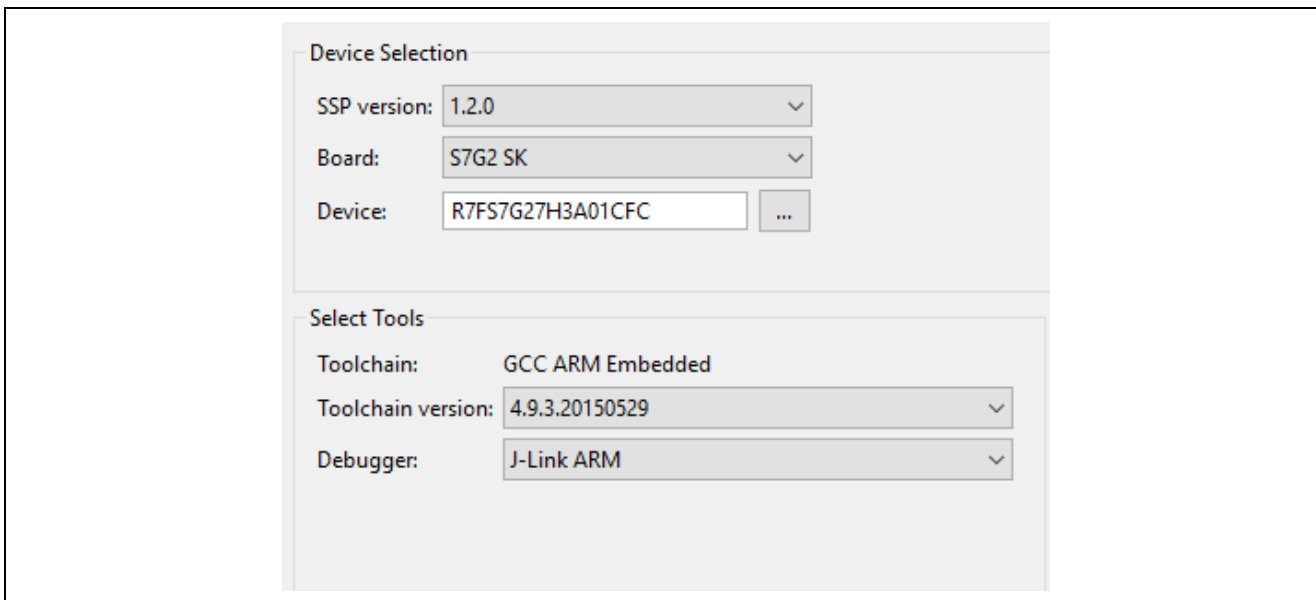


図 3.13 デバイス選択

14. [Next] をクリックしてください。
15. [Project Configuration] ダイアログで、オプション [BSP] を選択してください。

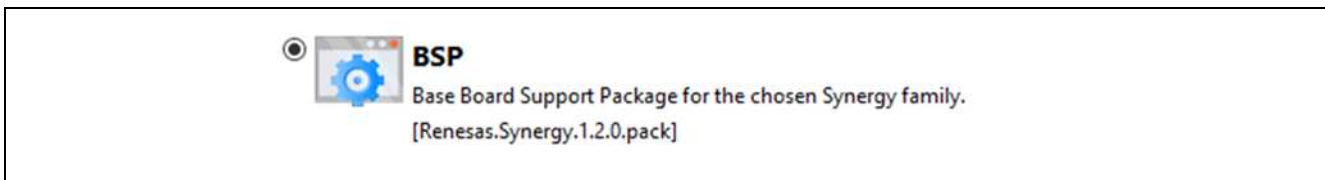


図 3.14 [BSP] を選択

16. [Finish] をクリックしてください。
17. 設定した[Perspectives]を e<sup>2</sup>studio に認識させていない場合、[Open Associated Perspective?] ダイアログボックスが表示されます。開いたら、認識させるために[Yes]をクリックして承認し、ボックスを閉じます。

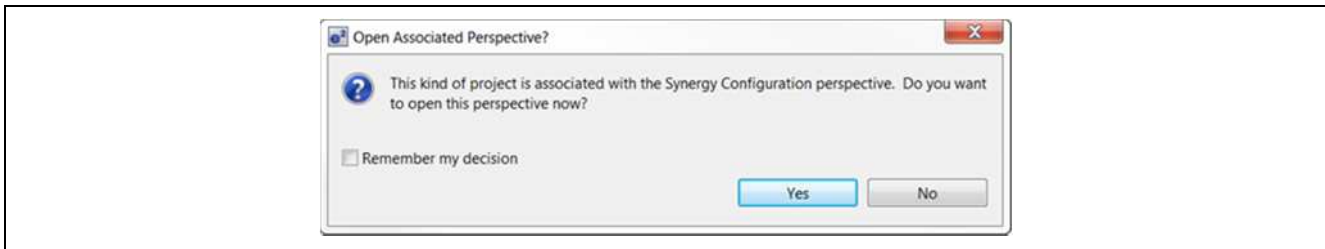


図 3.15 [Open Associated Perspective] ダイアログボックス

プロジェクトを作成されたら、以下の画面が表示されます。

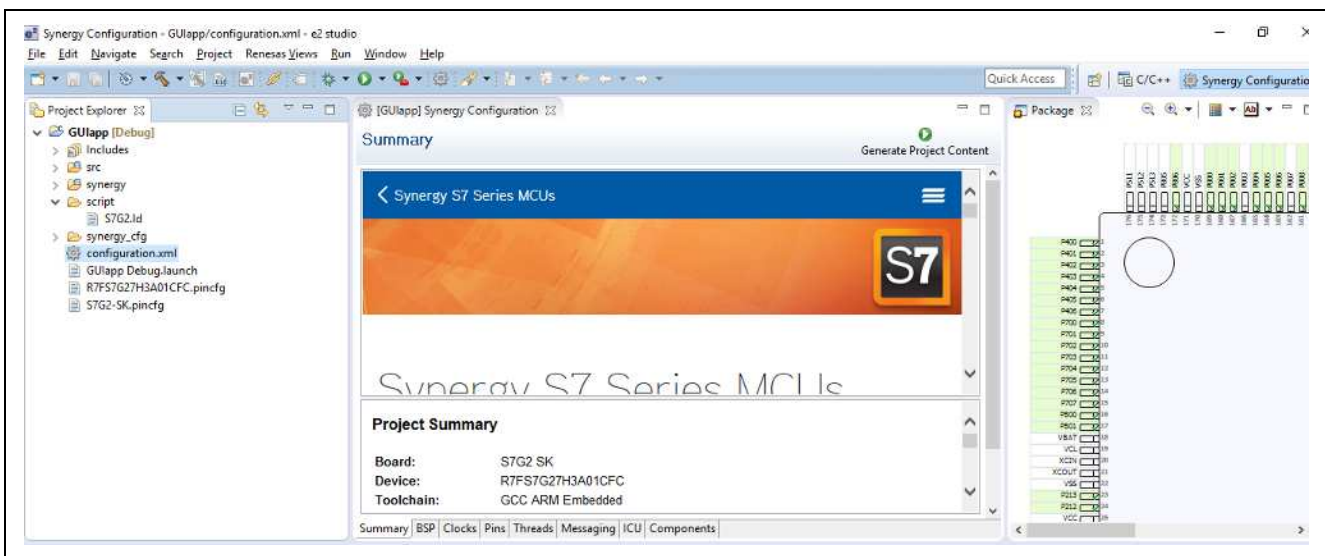


図 3.16 GUIApp プロジェクト

#### 4. ISDE 内のプロジェクト設定

ここで、プロジェクトは e<sup>2</sup>studio ISDE (Integrated Solution Development Environment) で作成され、GUI アプリケーション用にプロジェクトを設定可能になります。



## Renesas Synergy™ プラットフォーム SK-S7G2 用 GUIX “ Hello World ”

1. Synergy Configuration を開いてください。もし開いていない場合は、[Project Explorer]ウィンドウの [configuration.xml] ファイルをダブルクリックして開いてください。

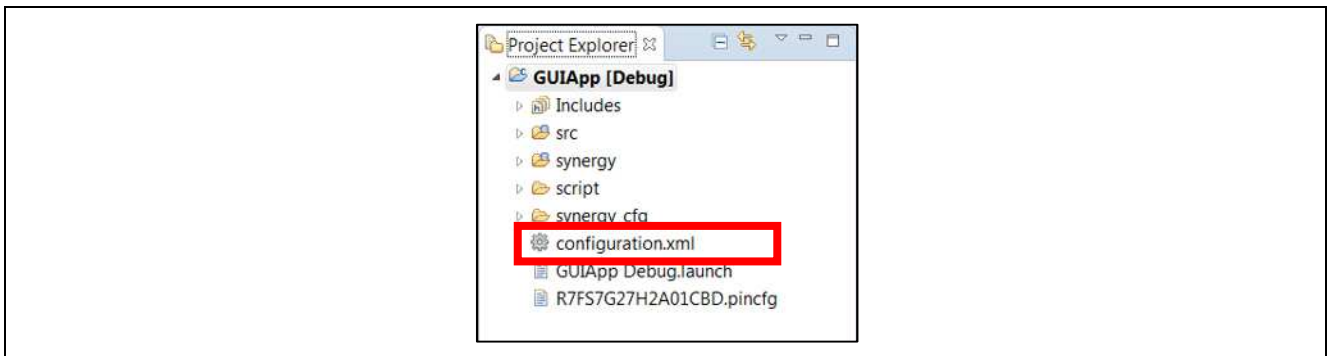


図 4.1 プロジェクトエクスプローラで [configuration.xml] ファイルを選択

2. [Synergy Configuration] ウィンドウで、[Threads] タブをクリックします。



図 4.2 [Synergy Configuration Threads] タブ

3. [Threads] 領域にある [New] をクリックして、新しいスレッドを作成してください。

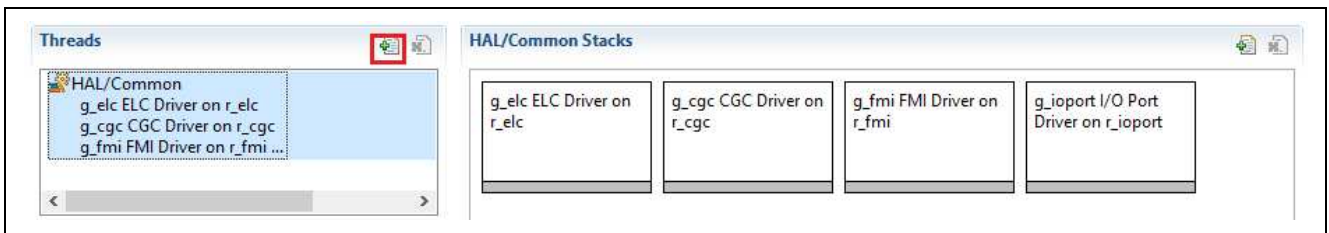


図 4.3 新しいスレッドの生成

4. [New Thread] をクリックし、プロパティを開いてください。
5. [Properties] を、以下に示すように変更してください。

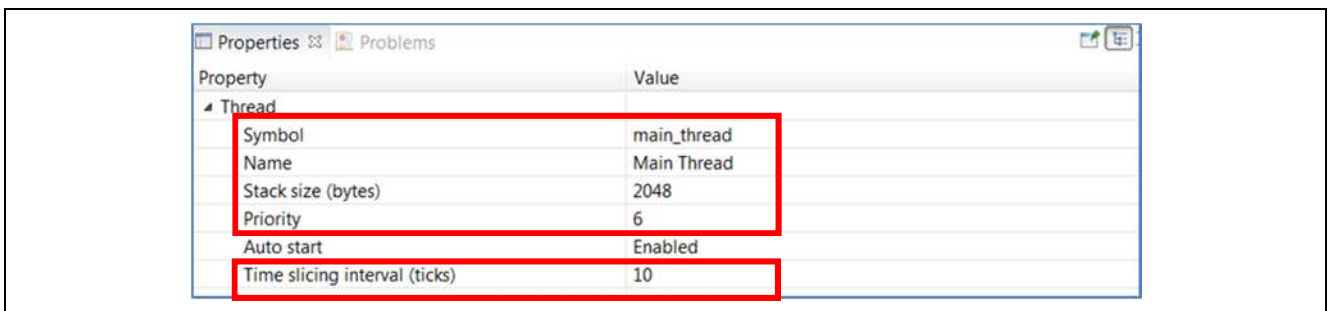


図 4.4 [Main Thread] プロパティの設定

6. [Synergy Configuration] ウィンドウ、[Threads] タブ、[Main Thread Stacks] 領域に戻り、[New] をクリックしてください。

注： [Main Thread] は、必ず新しいモジュールを追加する前に選択してください。

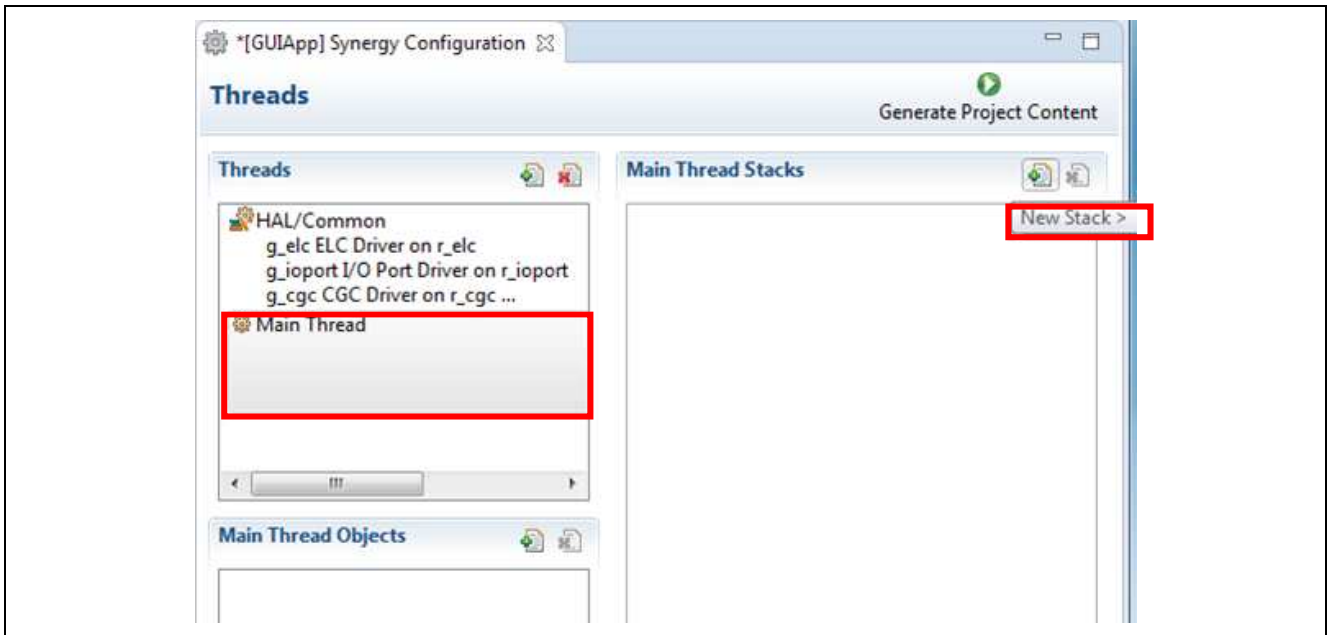


図 4.5 [Main Thread] スタック

7. [New] -> [Framework] -> [Input] -> [Touch Panel Framework on sf\_touch\_panel\_i2c] を選択して、タッチパネルのフレームワークを追加してください。

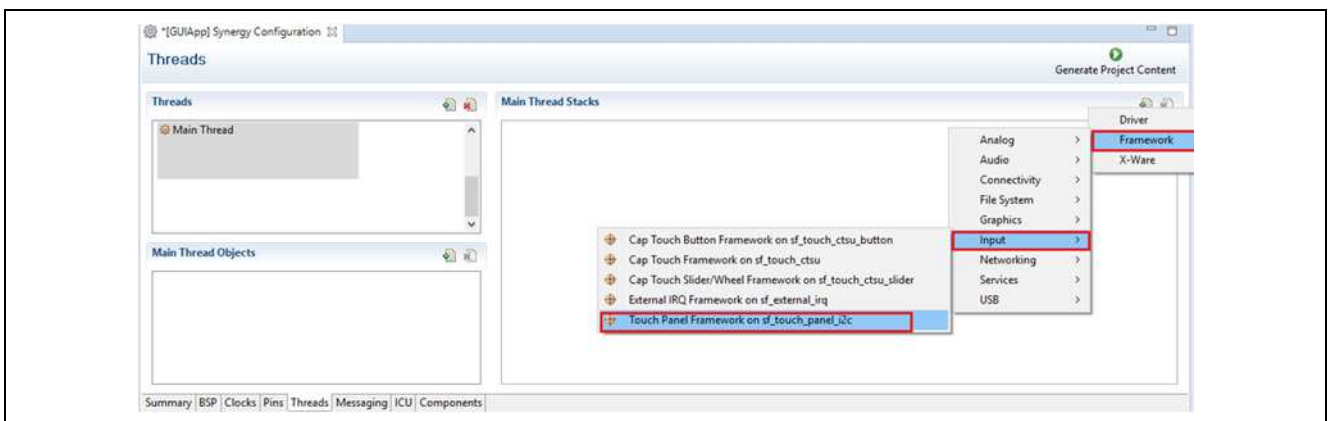


図 4.6 [Touch Panel Framework] の追加

8. プロパティを、以下に示すように設定してください。

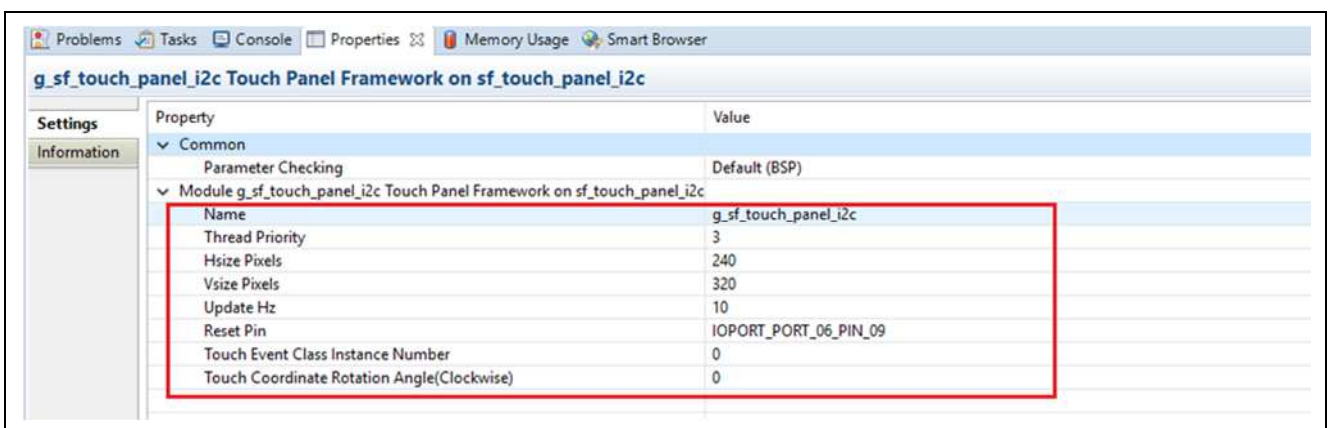


図 4.7 [Touch Panel] プロパティの設定

9. Synergy コンフィグレータは、メッセージフレームワークと外部 IRQ フレームワークを作成し、下記のように外部 IRQ と I2C ドライバスタック用のプレースホルダを持っています。  
 メッセージングフレームワークは、システム周辺にメッセージを渡す他のフレームワークレイヤとタスクに使われます。このシステムは、タッチ入力を処理するために、タッチスクリーンドライバからメインタスクにデータを渡すために使われます。

SF 外部割り込みは、以下に示すようにタッチコントローラドライバに使われるフレームワークレイヤです。

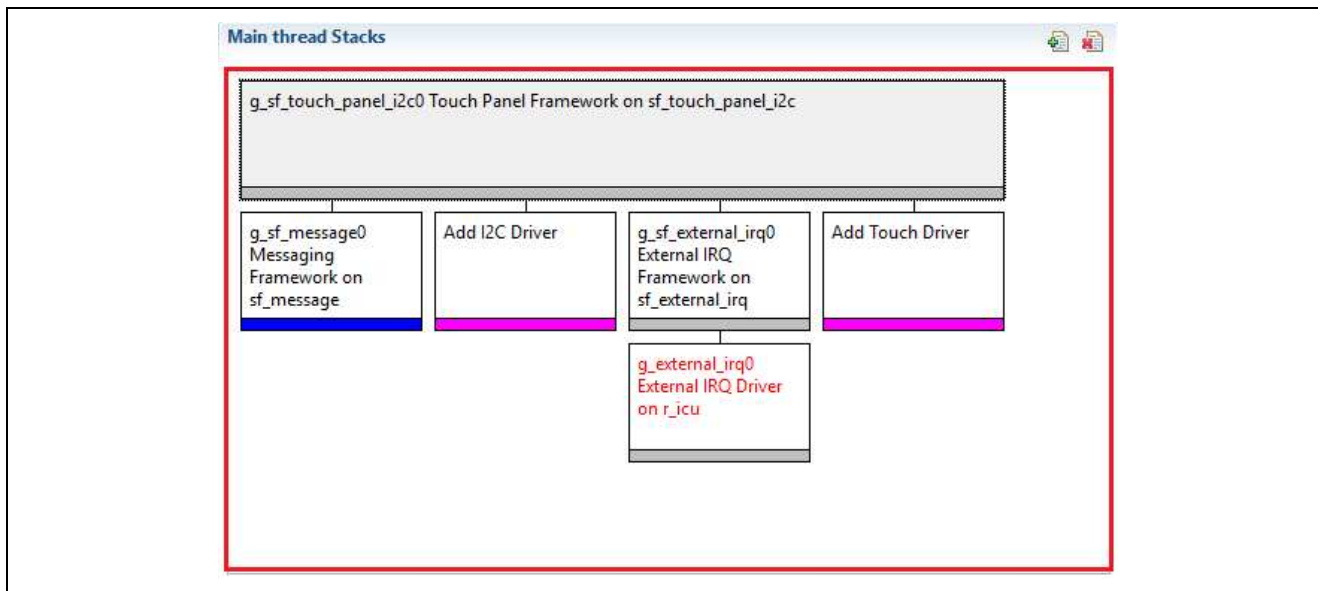


図 4.8 [Touch Panel Framework Stack]

10. [External IRQ Framework on sf\_external\_irq] を選択して、以下に示すようにプロパティを設定してください。

Property	Value
Common	
Parameter Checking	Default (BSP)
Module	
Name	g_sf_touch_irq
Event	Semaphore Put

図 4.9 [External Interrupts] プロパティの設定

11. [External IRQ Driver on r\_icu] を選択してください。新しいモジュールのプロパティを、以下の通りに設定してください。[Channel] は最初に変更してください。

Property	Value
▼ Common	
Parameter Checking	Default (BSP)
▼ Module g_touch_irq External IRQ Driver on r_icu	
Name	g_touch_irq
Channel	9
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock (Only valid when Digital Filtr	PCLK / 64
Interrupt enabled after initialization	True
Callback	NULL
Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 4.10 [Touch Screen IRQ]プロパティ

12. [Synergy Configuration] 画面、[Threads] タブ、[Main Thread Stacks] 領域で、[Add I2C Driver] 上で右クリックし、それから [New] -> [I2C Driver on r\_sci\_i2c] を選択して、I<sup>2</sup>C バス用のドライバを追加してください。

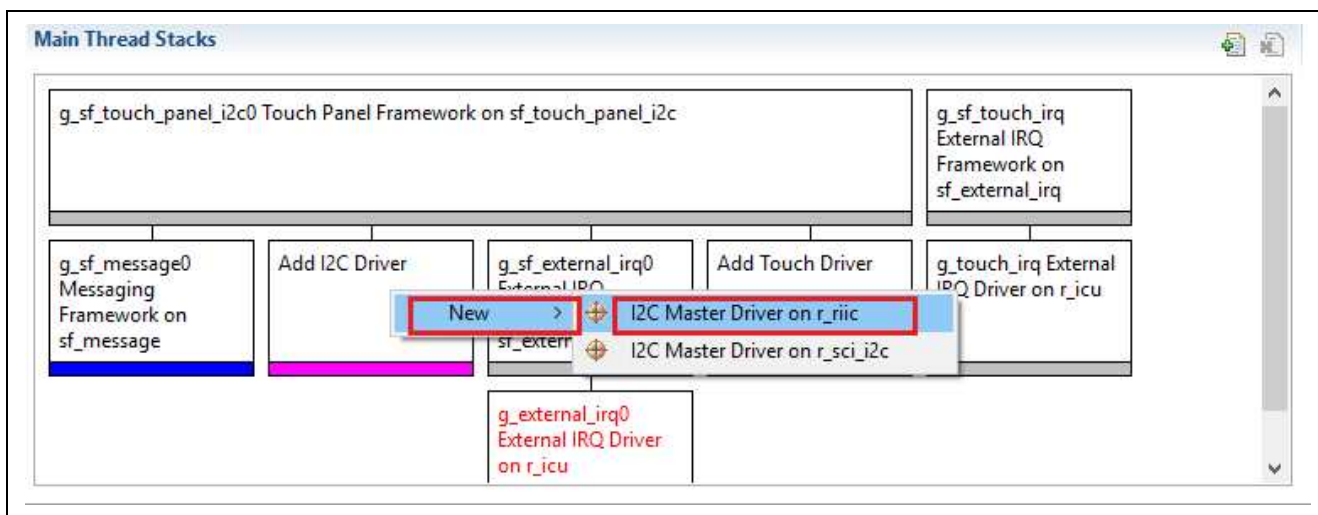


図 4.11 I<sup>2</sup>C バスを追加

13. [I2C Master Driver on r\_riic] のプロパティを、以下に示すように設定してください。 [Channel] オプションは最初に変更してください。

Property	Value
▼ Common	
Parameter Checking	Default (BSP)
▼ Module g_i2c I2C Master Driver on r_riic	
Name	g_i2c
Channel	2
Rate	Standard
Slave Address	0x48
Address Mode	7-Bit
Callback	NULL
Receive Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Transmit Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Transmit End Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Error Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 4.12 I2C ドライバの設定

14. [Touch Driver] のプロパティを、以下のように設定してください。

The screenshot shows the 'Main thread Stacks' window with a tree view of components. A 'New' button is highlighted next to 'Touch Panel Driver on touch\_panel\_sx8654'. Below the window, a property table is shown with the 'Name' property set to 'g\_sf\_touch\_panel\_i2c\_chip\_sx8654'.

Property	Value
▼ Module Touch Panel Driver on touch_panel_sx8654	
Name	g_sf_touch_panel_i2c_chip_sx8654

図 4.13 [Touch Driver] の設定

15. [Main Thread Stacks] の元で、New [X-Ware] -> [GUIX] -> [GUIX on gx] を選択してください。

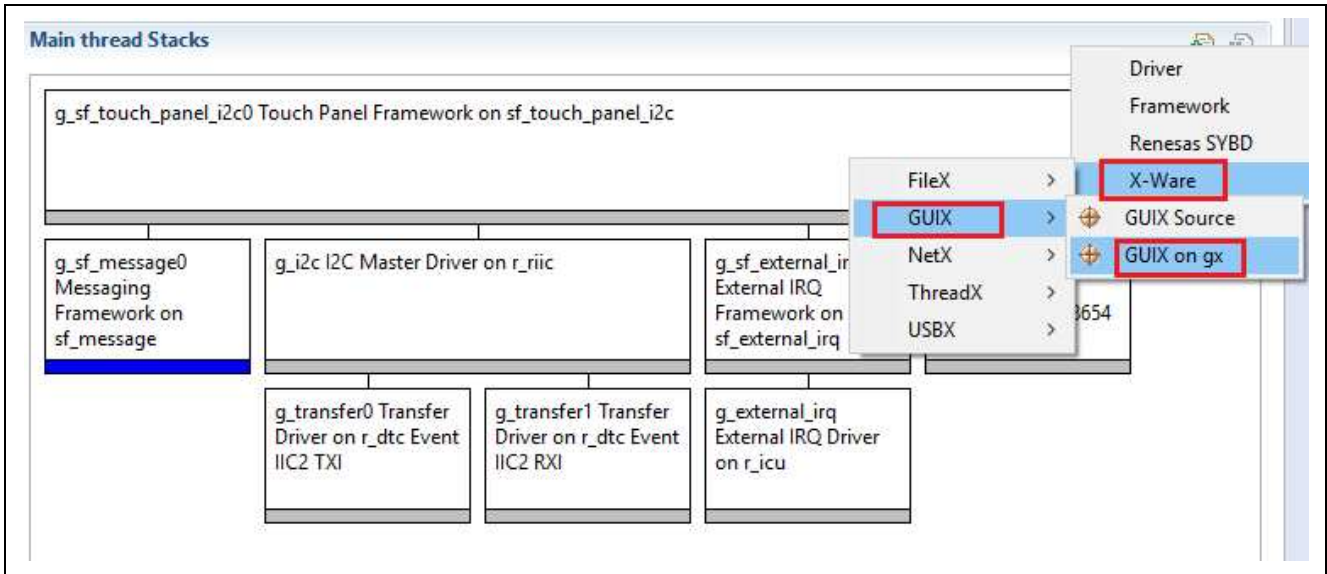


図 4.14 [GUIX on gx] のフレームワーク追加

図 4.15 に示すように、Synergy コンフィグレータは [GUIX Port on sf\_el\_gx] フレームワーク、ディスプレイドライバを作成し、下記のように JPEG デコードと D/AVE ハードウェアアクセラレータスタック用のプレースホルダを持っています。

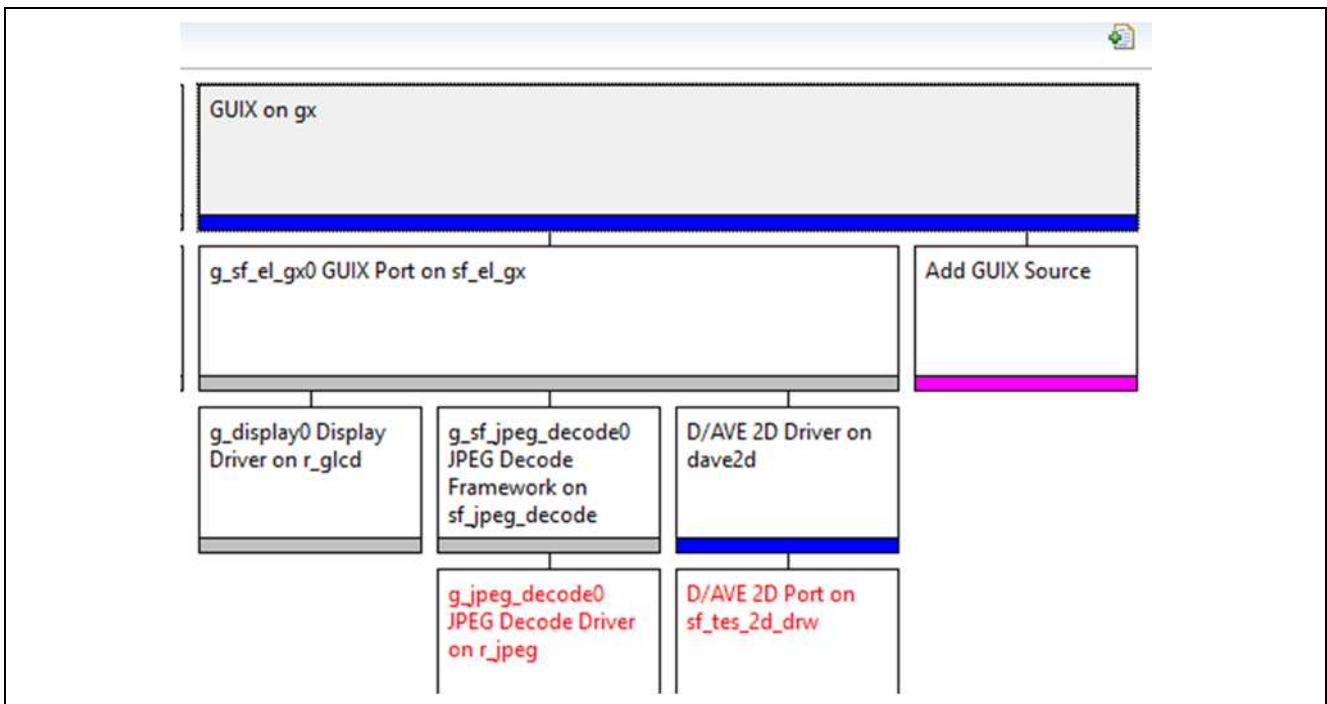


図 4.15 GUIX on gx stack

16. [GUIX on gx] を選択し、プロパティを以下に示すように設定してください。

Property	Value
Common	
Enable Synergy 2D Drawing Engine Support	Yes
Enable Synergy JPEG Support	Yes

図 4.16 [GUIX on gx] プロパティ

17. [GUIX Port on sf\_el\_gx] を選択し、以下に示すようにプロパティを設定してください。

Property	Value
Common	
Parameter Checking	Default (BSP)
Module	
Name	g_sf_el_gx
Name of Display Driver Run-time Configuration (Must be a valid symbol)	g_display0_runtime_cfg_bg
Name of Frame Buffer A (Must be a valid symbol)	g_display0_fb_background[0]
Name of Frame Buffer B (NULL allowed if consisting a single frame)	g_display0_fb_background[1]
Name of User Callback function	NULL
Screen Rotation Angle(Clockwise)	0
GUIX Canvas Buffer (required if rotation angle is not zero)	Not used
Size of JPEG Work Buffer (valid if JPEG hardware acceleration enabled)	81920
Memory section for GUIX Canvas Buffer	sdram
Memory section for JPEG Work Buffer	bss

図 4.17 [GUIX Port] プロパティ設定

18. [JPEG Decode Driver on r\_jpeg] を選択し、以下に示すように割り込みプロパティを設定してください。  
Priority 3 は単に任意の数値である点に注意してください。

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_jpeg_decode0 JPEG Decode Driver on r_jpeg	
Name	g_jpeg_decode0
Byte Order for Input Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8)
Byte Order for Output Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8)
Output Data Color Format	Pixel Data RGB565 format
Alpha value to be applied to decoded pixel data(only valid for ARGB8888 format)	255
Name of user callback function	NULL
Decompression Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Data Transfer Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 4.18 [JPEG Decode Driver on r\_jpeg] プロパティ

19. [Main Thread Stacks] の元で、[D/AVE 2D Port on sf\_tes\_2d\_drw] を選択し、以下に示すようにプロパティを設定してください。

Property	Value
Common	
Work memory size for display lists in bytes	32768
DRW Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 4.19 [D/AVE 2D Port] プロパティ

20. [Main Thread Stacks] の元で、[Display Driver on r\_glcd] を選択し、以下に示すように [Interrupt Properties] を設定してください。

Line Detect Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Underflow 1 Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Underflow 2 Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 4.20 [Interrupt Properties]

21. 以下に示すように [Graphics Screen 1] プロパティを設定してください。

Module	
Name	g_display0
Name of display callback function to be defined by user	NULL
Input - Panel clock source select	Internal clock(GLCDCLK)
Input - Graphics screen1	Used
Input - Graphics screen1 frame buffer name	fb_background
Input - Number of Graphics screen1 frame buffer	2
Input - Section where Graphics screen1 frame buffer allocated	bss
Input - Graphics screen1 input horizontal size	256
Input - Graphics screen1 input vertical size	320
Input - Graphics screen1 input horizontal stride(not bytes but pixels)	256
Input - Graphics screen1 input format	16bits RGB565
Input - Graphics screen1 input line descending	Not used
Input - Graphics screen1 input lines repeat	Off
Input - Graphics screen1 input lines repeat times	0
Input - Graphics screen1 layer coordinate X	0
Input - Graphics screen1 layer coordinate Y	0
Input - Graphics screen1 layer background color alpha	255
Input - Graphics screen1 layer background color Red	255
Input - Graphics screen1 layer background color Green	255
Input - Graphics screen1 layer background color Blue	255
Input - Graphics screen1 layer fading control	None
Input - Graphics screen1 layer fade speed	0

図 4.21 [Graphics Screen 1] プロパティ

22. 以下に示すように [Output] プロパティを設定してください。



Output - Horizontal total cycles	320
Output - Horizontal active video cycles	240
Output - Horizontal back porch cycles	6
Output - Horizontal sync signal cycles	4
Output - Horizontal sync signal polarity	Low active
Output - Vertical total lines	328
Output - Vertical active video lines	320
Output - Vertical back porch lines	4
Output - Vertical sync signal lines	4
Output - Vertical sync signal polarity	Low active
Output - Format	16bits RGB565
Output - Endian	Little endian
Output - Color order	RGB
Output - Data Enable Signal Polarity	High active
Output - Sync edge	Rising edge
Output - Background color alpha channel	255
Output - Background color R channel	0
Output - Background color G channel	0
Output - Background color B channel	0

図 4.22 [Output Screen 2] プロパティ

23. 以下に示すように [TCON] 端子とクロックを設定してください。

TCON - Hsync pin select	LCD_TCON2
TCON - Vsync pin select	LCD_TCON1
TCON - DataEnable pin select	LCD_TCON0
TCON - Panel clock division ratio	1/32

図 4.23 [TCON] の設定

24. [Main Thread Stacks] の元で、[New] -> [Driver] -> [Connectivity] -> [SPI Driver on r\_sci\_spi]を選択してください。

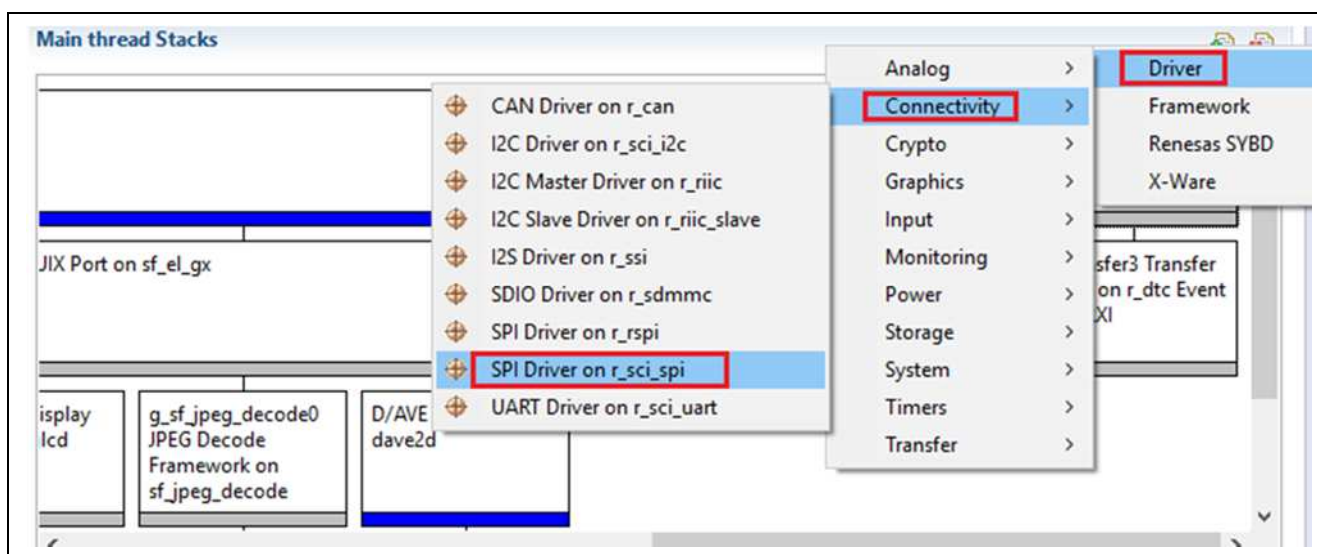


図 4.24 [Simple SPI (on SCI)] ドライバの追加

25. プロパティを、以下に示すように設定してください。

Property	Value
▼ Common	
Parameter Checking	Default (BSP)
▼ Module g_spi_lcdc SPI Driver on r_sci_spi	
Name	g_spi_lcdc
Channel	0
Operating Mode	Master
Clock Phase	Data sampling on even edge, data variation on odd edge
Clock Polarity	High when idle
Mode Fault Error	Disable
Bit Order	MSB First
Bitrate	100000
Bit Rate Modulation Enable	Enable
Callback	g_lcd_spi_callback
Receive Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Transmit Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Transmit End Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)
Error Interrupt Priority	Priority 3 (CM4: valid, CM0+: lowest - not valid if using ThreadX)

図 4.25 [Simple SPI (on SCI)] プロパティ設定

26. LCD では不要なため、各 [g\_transfer] ドライブをクリックしてから [Remove] をクリックしてそれを削除します。

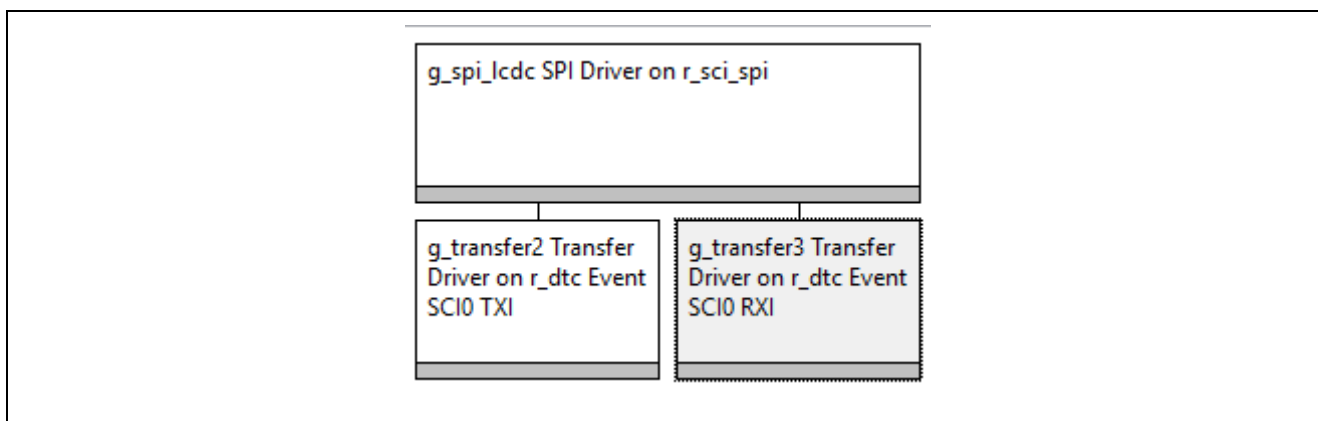


図 4.26 転送ドライバの削除

27. ドライバの削除の後も、図 4.27 に示すようにドライバ追加用のプレースホルダが残ります。

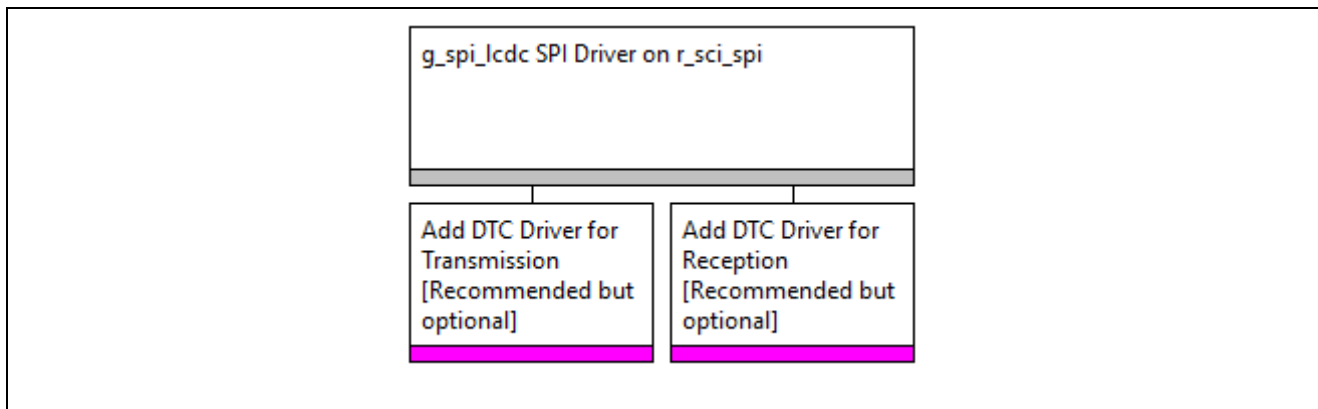


図 4.27 Transfer Drivers Placeholders

28. [Synergy Configuration] ウィンドウ、[Threads] タブにおいて、[Main thread] がまだ選択されていることを確認してください。

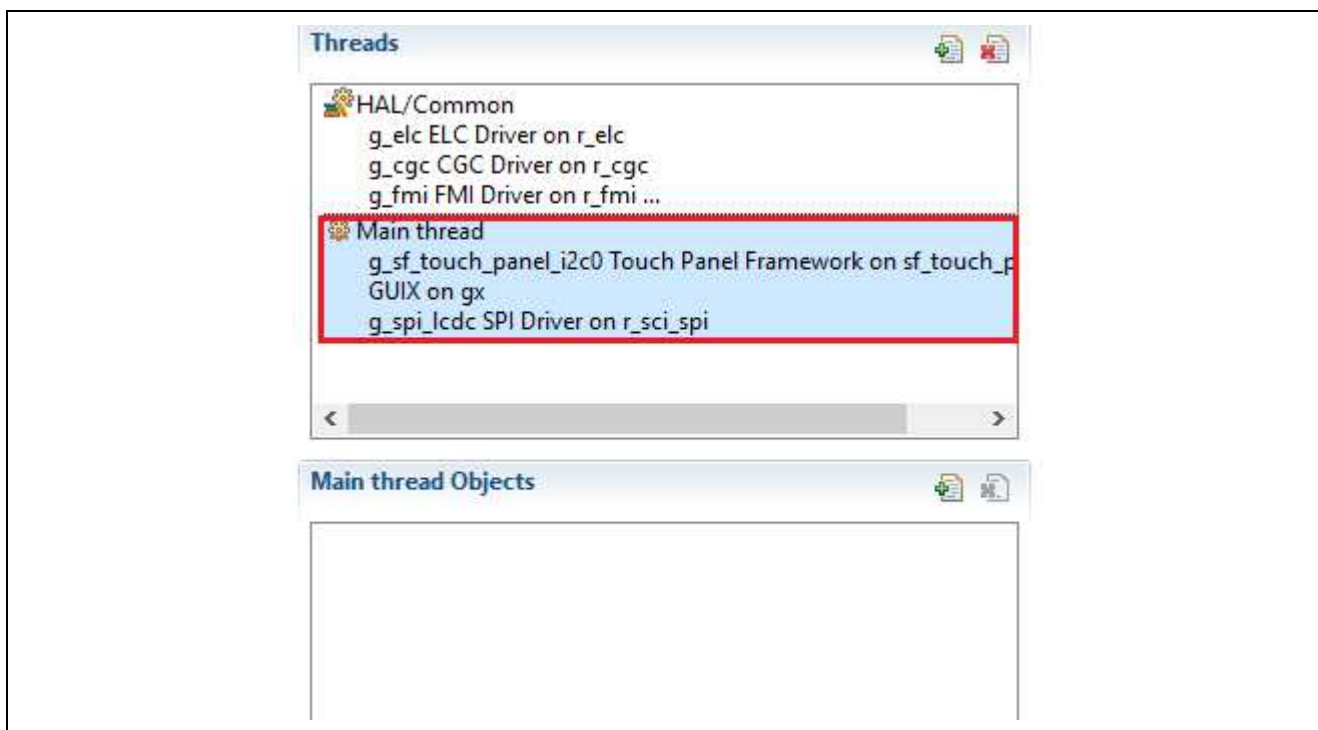


図 4.28 [Main thread] をクリックする

29. [Main thread Objects] の元で、[New] -> [Semaphore] をクリックしてください。

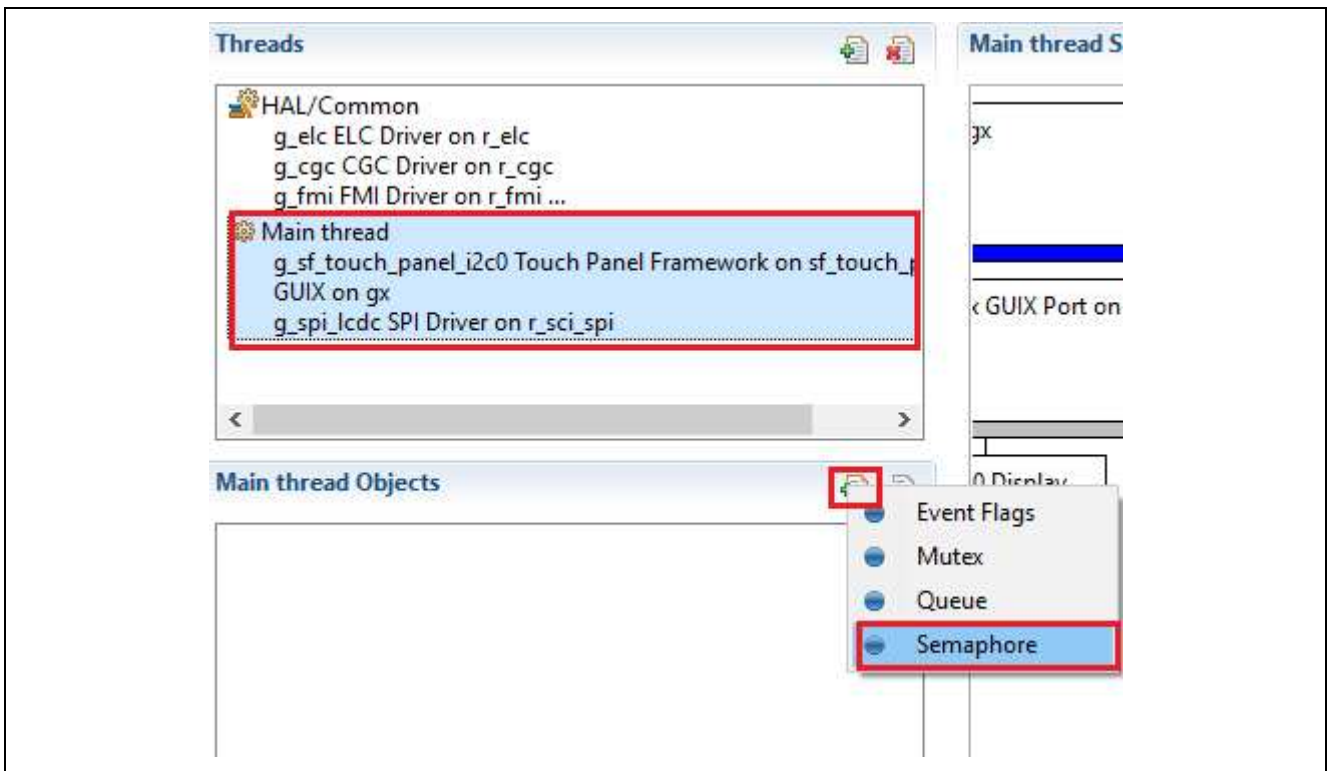


図 4.29 [Semaphore] の追加

30. プロパティを、以下に示すように設定してください。



図 4.30 [Semaphore] の設定

31. [Synergy Configuration] ウィンドウで、[Pins] タブを選択してください。

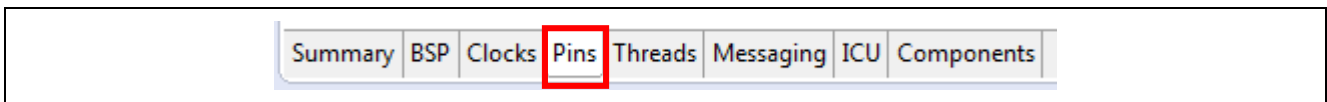


図 4.31 端子の設定

32. SPI0 の端子設定の [Pin Selection] で、[Peripherals] -> [Connectivity:SPI] -> [SPI0] を選択し、[Operation Mode] を [Disabled] に変更してください。SCI モジュールと共有する端子を開放するため、無効にする必要があります。

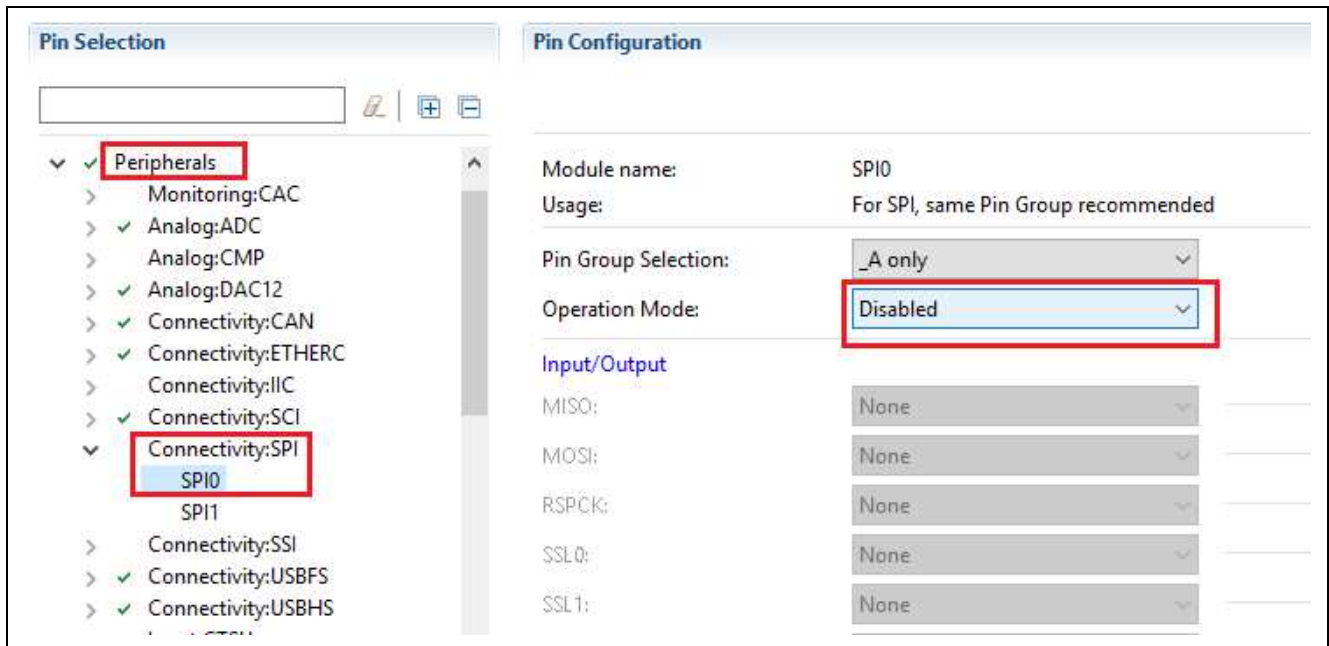


図 4.32 端子設定で [SPI0\_Pin\_Option\_A] を無効化

33. 以下に示すように、[Pin Selection] で[Peripherals] -> [Connectivity:SCI] -> [SCI0] を選択し、SCI0 モジュールの端子を設定してください。

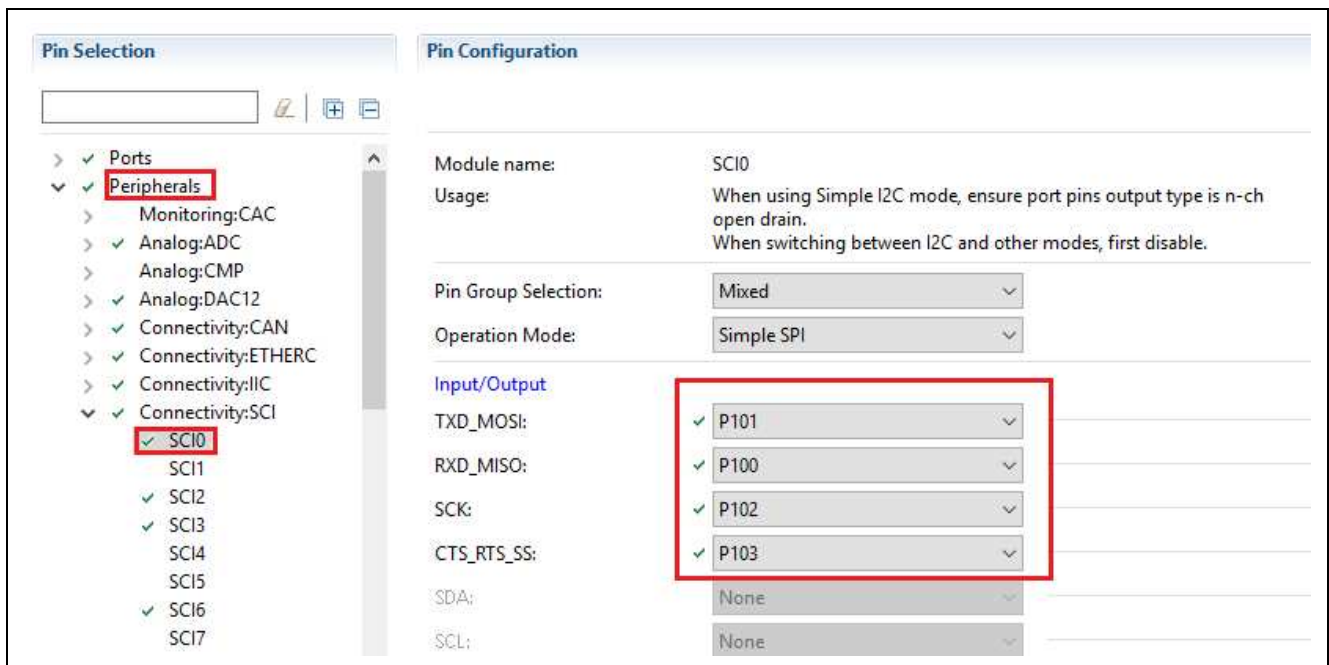


図 4.33 [SCI0] 端子設定

34. 以下に示すように、[Pin Selection] で、[Peripherals] -> [Connectivity: IIC] -> [IIC2]を選択し、[Pin Configuration] の [IIC2] モジュールを有効にしてください。

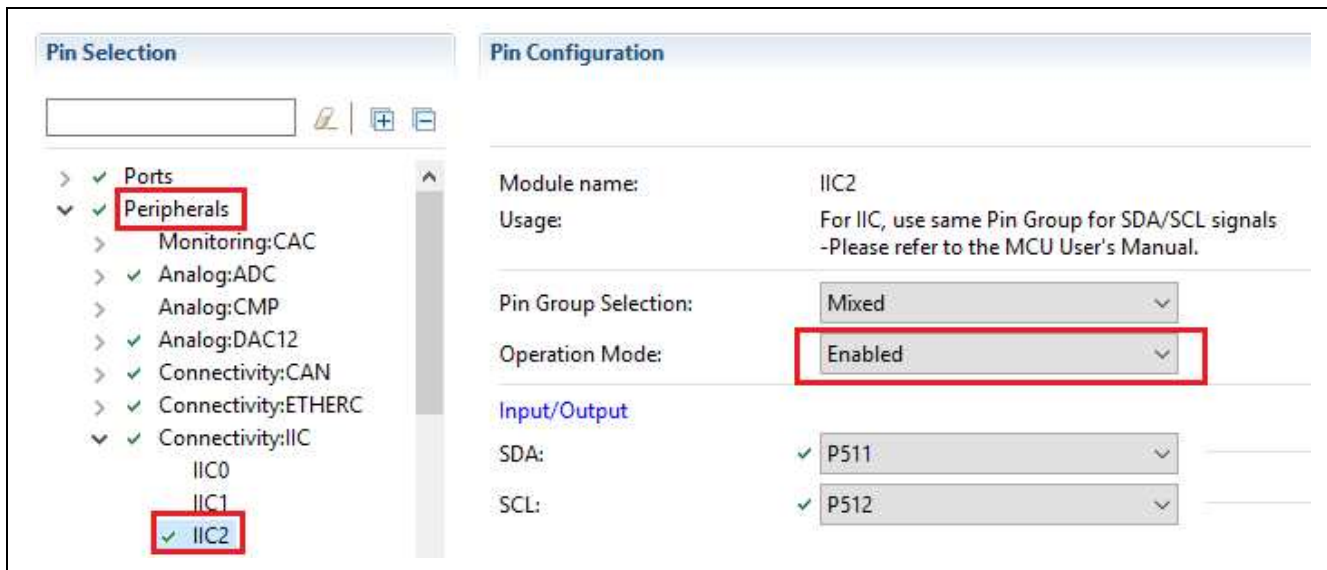


図 4.34 [IIC2] 端子設定

35. 以下に示すように、[Pin Selection] で、[Ports] -> [P1] -> [P115] を選択し、[Pin Configuration] で、[GPIO] を設定してください。この端子は、[LCD\_WR] 信号からのデータアクセスタイミングを制御するため、SK-S7G2 ボード上の LCD パネルと接続されています。

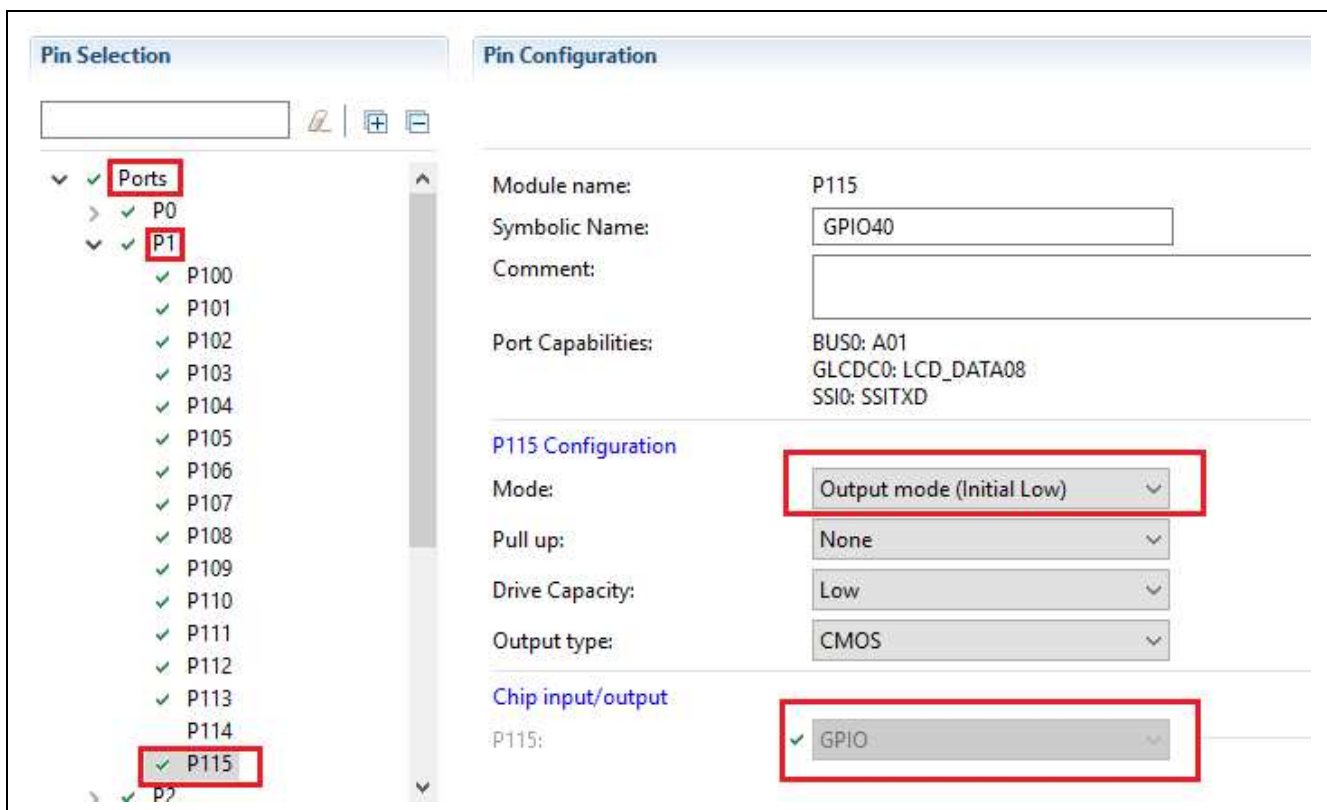


図 4.35 [P115] の設定

36. [Pin Selection] で [Ports] -> [P6] を選択し、P609 (タッチパネル用 RESET#)、P610 (LCD\_RESET)と P611 (LCD\_CS) を GPIO の output モードで設定してください。

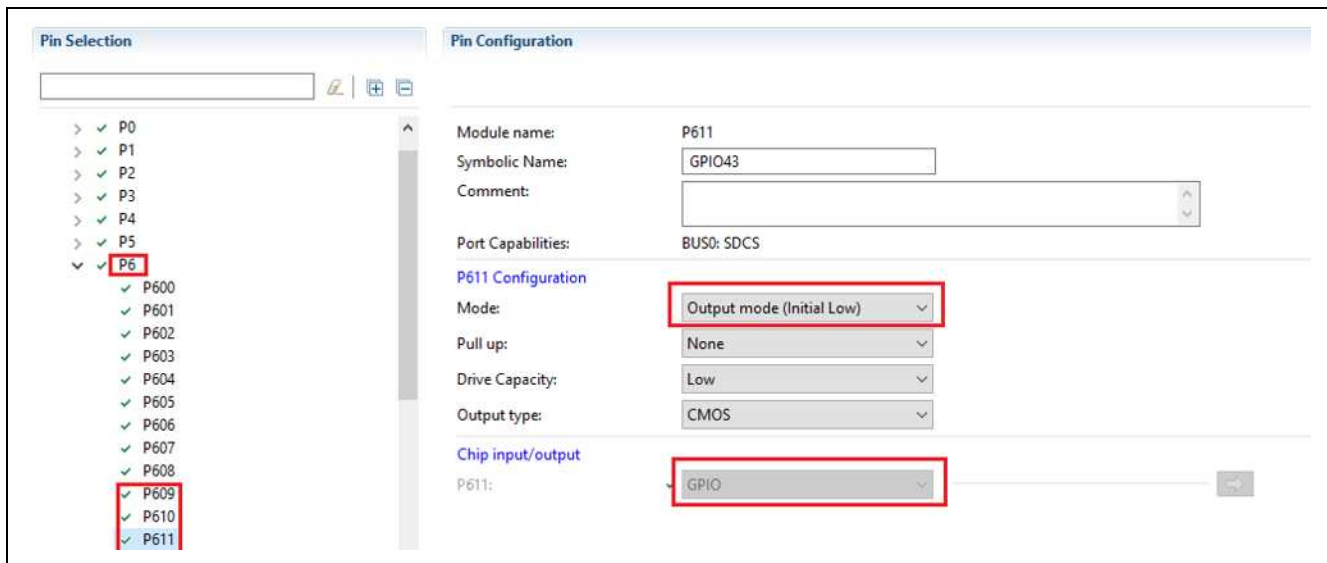


図 4.36 [P609]、[P610]、[P611]の設定

37. 図 4.37 に示すように、[GLCD\_Controller\_Pin\_Option\_B] に関係するすべての端子に対し、[Drive Capacity] を [High] に設定してください。[Drive Capacity] を High に設定するには、2つの方法があります。A および B のうちどちらも選択可能です。

A. 以下の図 4.37 と 図 4.38 に示すように、どの端子に [GLCD\_Controller\_Pin\_Option\_B] が使用されているかを確認できます。

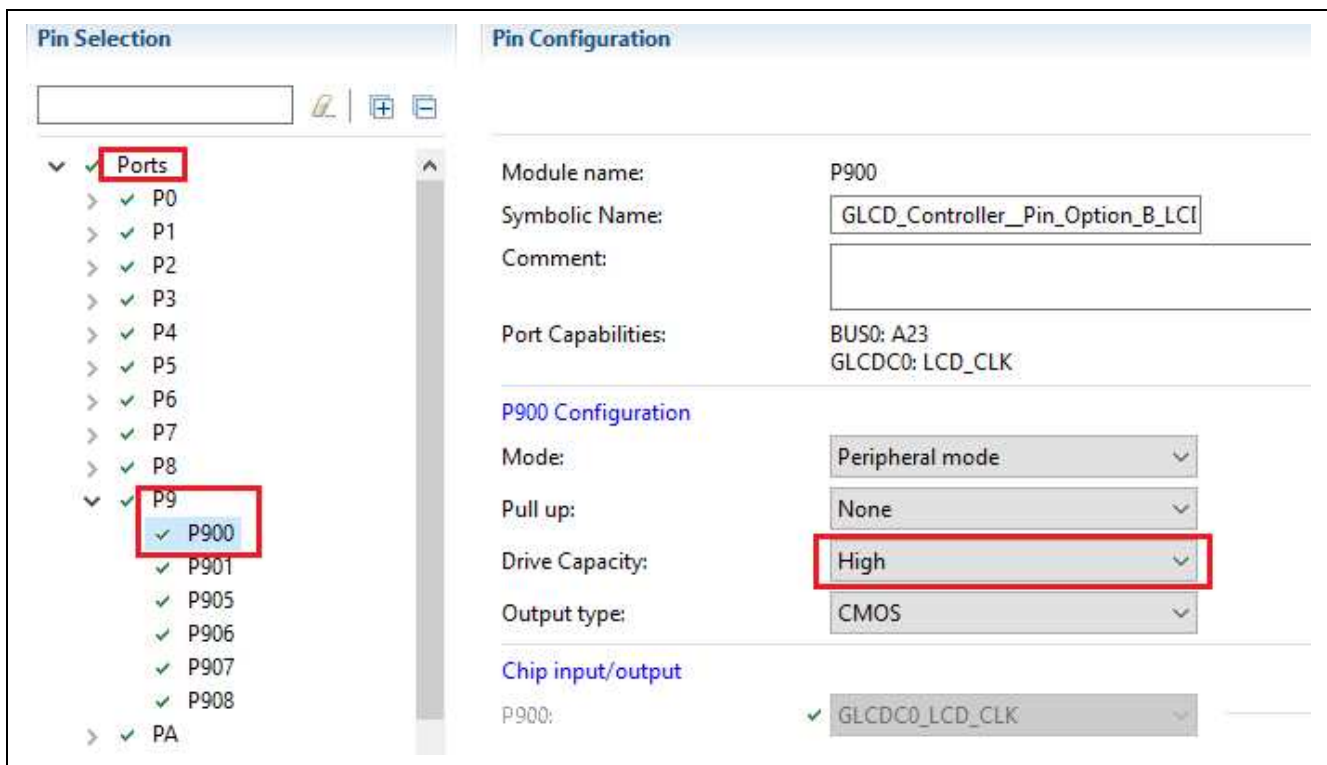


図 4.37 GLCDC の Drive Capability 設定の例

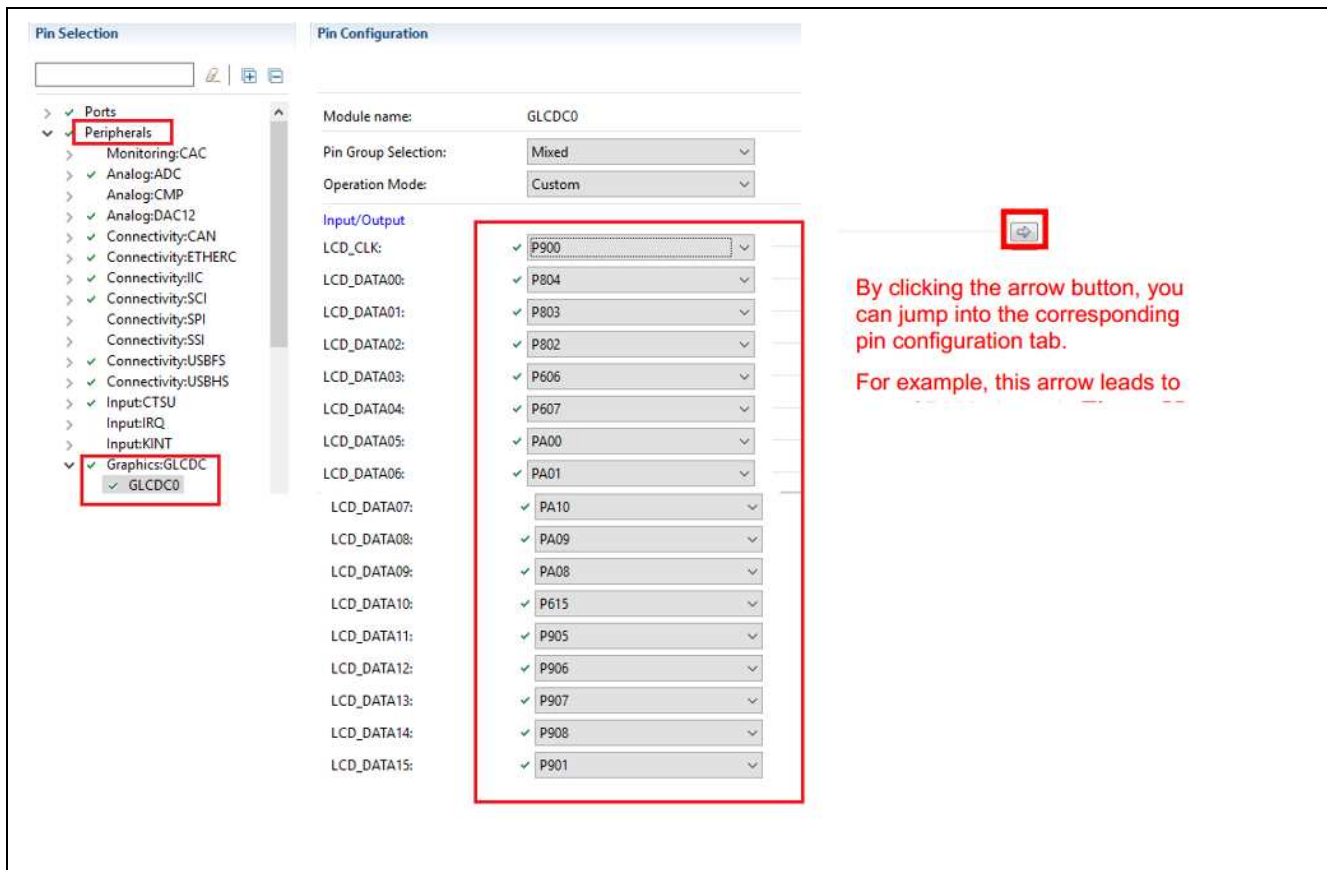


図 4.38 GLCD\_Controller\_Pin\_Option\_B の端子配置

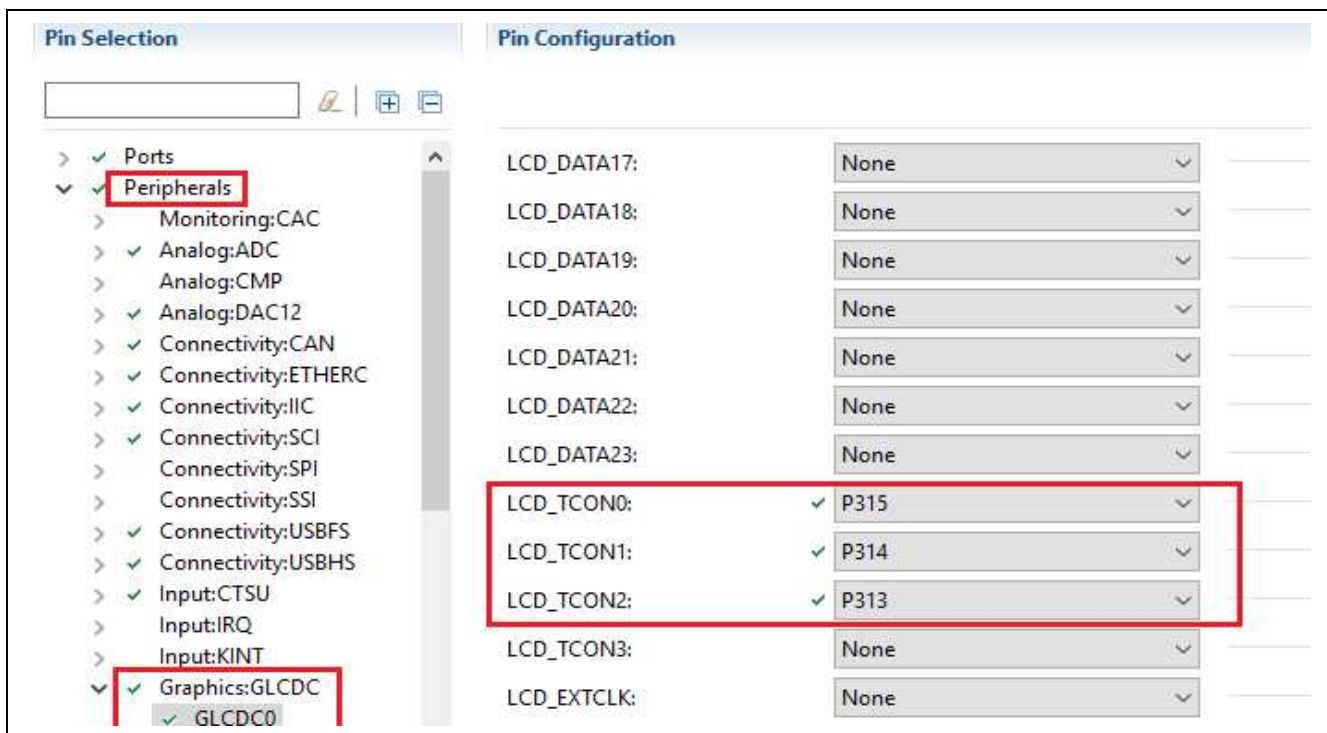


図 4.39 [GLCD\_Controller\_Pin\_Option\_B] の端子配置（続き）

B. ポートでの設定も可能です。以下に、[Drive Capacity] を High に設定する必要がある端子の番号順リストを示します。これらのポートへは、[Ports] -> [PX] -> [PXYZ]. の順に選択することでアクセス



することができます。ここで、X はリストでポートの 2 桁目を表し、PXYZ でポート全体を示します。一旦ポートを選択したら、図 4.37 に示すように [Drive Capacity] を **High** に設定してください。

S7G2 端子
P313
P314
P315
P606
P607
P615
P802
P803
P804
P900
P901
P905
P906
P907
P908
PA00
PA01
PA08
PA09
PA10

図 4.40 高駆動能力に設定するポートの番号順リスト

38. 以下の図に示すように、[Synergy Configuration] ウィンドウで [Messaging] タブを選択してください。

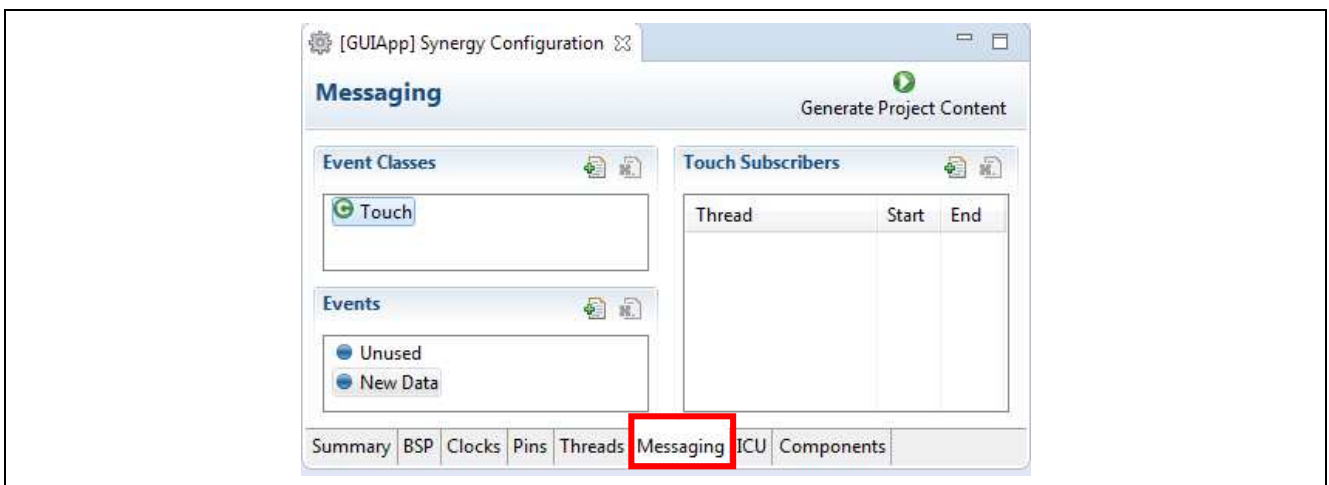


図 4.41 [Messaging] タブ

注：このタブは、イベントキュー初期化とリンク変数とともに、タッチスクリーンイベント用のイベントクラス定義の設定を行います。[Touch Panel Framework on sf\_touch\_panel\_i2c] がスレッドメニューに追加されると、タッチイベントは自動的に作成されます。

39. タッチイベントクラスを選択してください。

40. タッチサブスクライバメニューで [New] ボタンをクリックしてください。



図 4.42 [Messaging] タブ

41. [New Subscriber] ダイアログで、[Thread] プルダウンリストの [Main Thread] を選択してください。

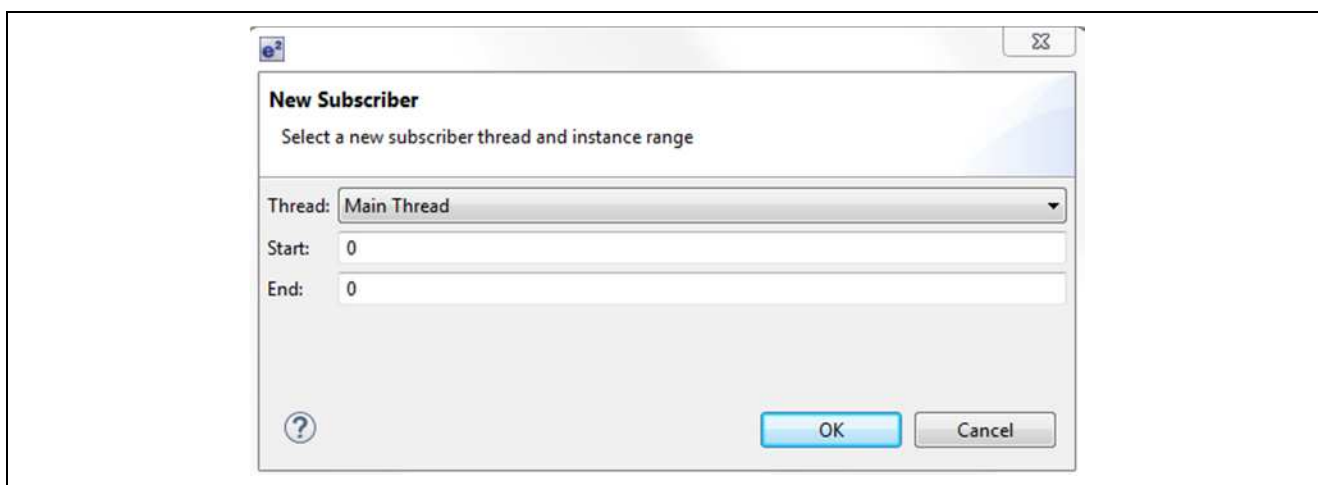


図 4.43 [New Subscriber] ダイアログ

42. OK をクリックしてください。

43. キーボードで [Ctrl + s] を押し、プロジェクトを保存してください。

44. [Generate Project Content] ボタンをクリックし、プロジェクトファイルを更新してください。

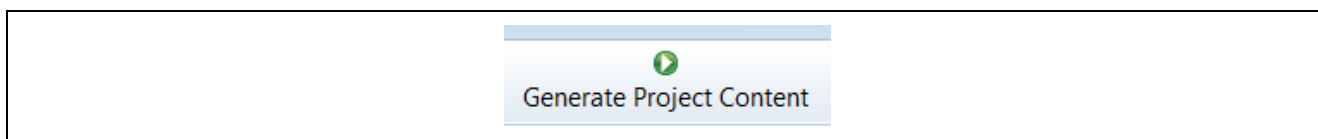


図 4.44 プロジェクト内容の作成

45. プロジェクトエクスプローラ ウィンドウで、[src] を右クリックし、[New] > [Folder] と進み [New Folder] ダイアログボックスを開きます。

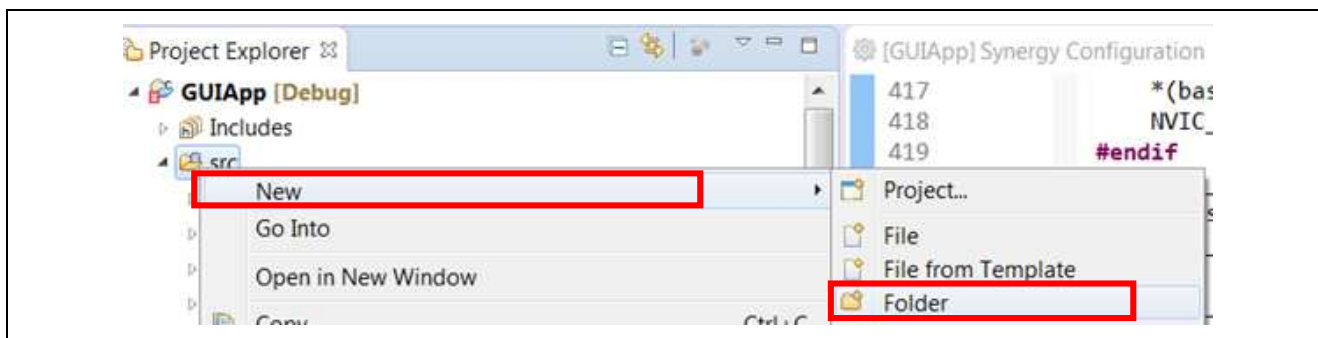


図 4.45 新規フォルダの作成

46. [Folder name:]テキストボックスに、新規フォルダの名前 hardware を入力してください。

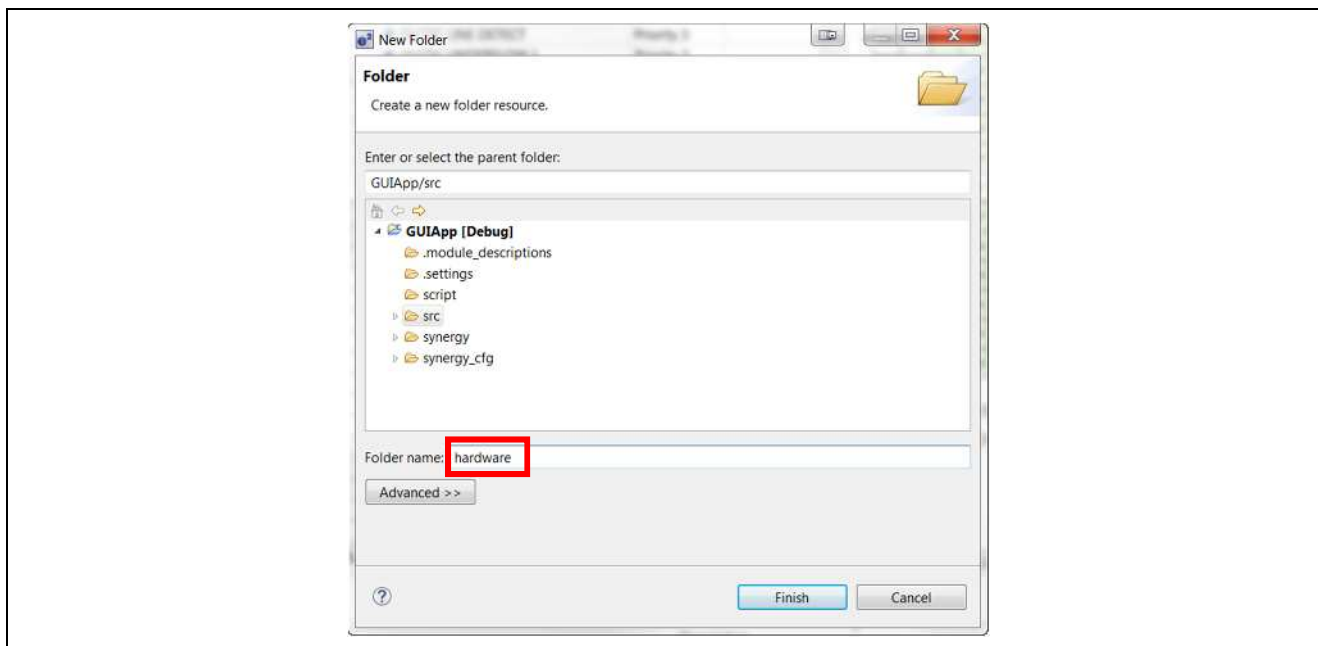


図 4.46 新規フォルダダイアログ

47. [Finish] をクリックしてください。

48. そのフォルダがプロジェクトエクスプローラに現れ、ハイライトされます。

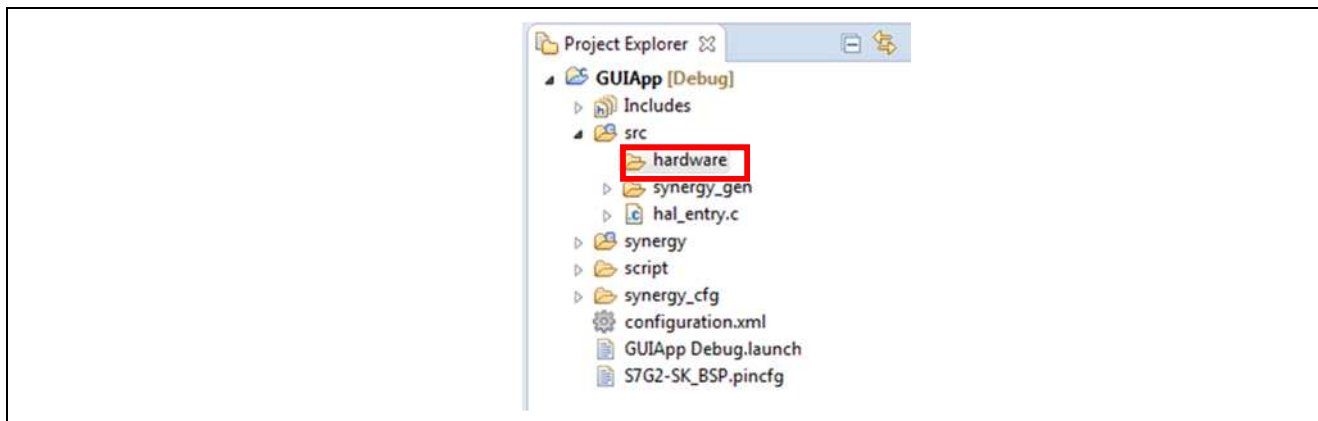


図 4.47 ハードウェアフォルダ

49. Windows エクスプローラを開き、このアプリケーションノートが含まれていたファイルを格納したフォルダに行き、[Source Files\lcd.h] ファイルを見つけてください。ここで、そのファイルを Windows エクスプローラ画面から、e<sup>2</sup>studio プロジェクトエクスプローラ画面の中にある新しい [hardware] フォルダに直接ドラッグしてください。
50. 選択したファイルをインポートするプロンプトが表示されたら、[OK] をクリックしてファイルをコピーしてください。

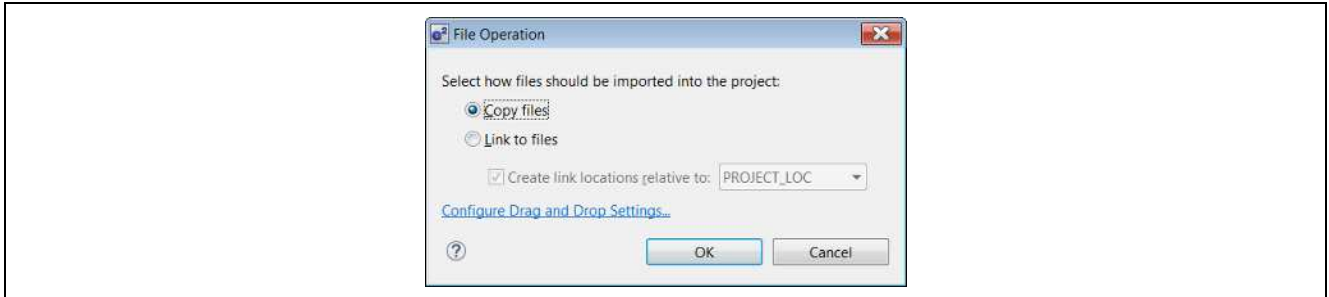


図 4.48 ファイル操作ダイアログ

注：このファイルは、LCD パネルを制御するコマンド定義を含んでいます。

51. Windows エクスプローラを開き、このアプリケーションノートが含まれていたファイルを格納したフォルダに行き、[Source Files\ lcd\_setup.c] ファイルを見つけてください。ここで、そのファイルを Windows エクスプローラ画面から、e<sup>2</sup>studio プロジェクトエクスプローラ画面の中にある [hardware] フォルダに直接ドラッグしてください。
52. 選択したファイルをインポートするプロンプトが表示されたら、[OK] をクリックしてファイルをコピーしてください。  
注：このファイルは、SPI を経由して LCD パネルに送るコマンドプロトコルと、初期化シーケンスを含んでいます。
53. Windows エクスプローラを開き、このアプリケーションノートが含まれていたファイルを格納したフォルダに行き、[Source Files\main\_thread\_entry.c] ファイルを見つけてください。ここで、そのファイルを Windows エクスプローラ画面から、e<sup>2</sup>studio プロジェクトエクスプローラ画面の中にある [src] フォルダに直接ドラッグしてください。
54. 選択したファイルをインポートするプロンプトが表示されたら、[OK] をクリックしてファイルをコピーしてください。
55. 上書きするプロンプトが表示されたら、[Yes] をクリックしてください。  
注：このファイルは、メインスレッドイベントハンドリングコードを含んでいます。これは、ロウレベルタッチスクリーンイベントをキューから読み込み、それをグラフィックユーザーインターフェースアクションに変換します。
56. e<sup>2</sup>studio プロジェクトエクスプローラのウィンドウの中の、[synergy] -> [ssp] -> [src] -> [driver] -> [r\_riic] フォルダを見つけてください。
57. フォルダを右クリックし、[Properties] メニューを選択してください。

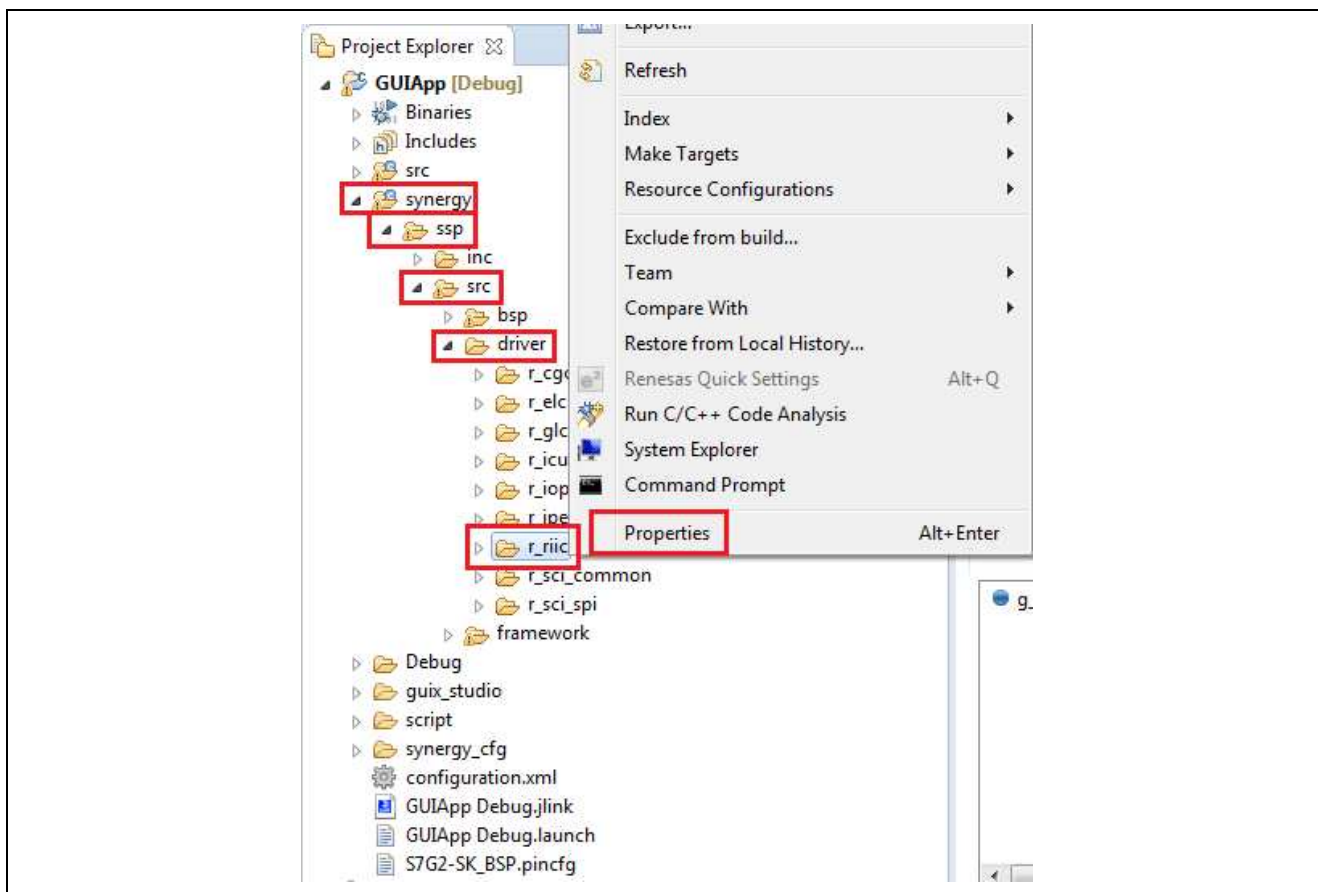


図 4.49 [src] フォルダの下に [gui] フォルダを作成する

58. [C/C++ Build] > [Settings] > [Cross ARM C Compiler] > [Optimization] の順に進んでください。

59. [Other optimization flags] 内で、テキストオプション[-O0]を設定してください。

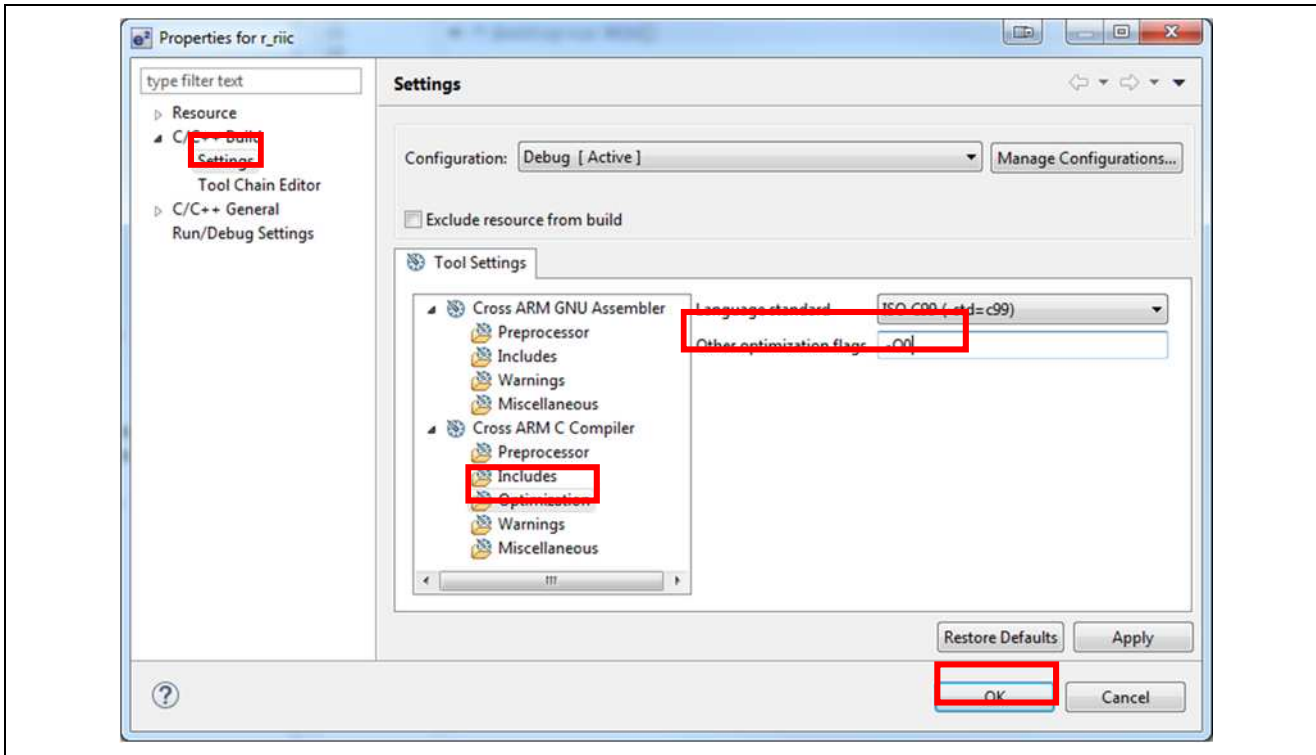


図 4.50 [src] フォルダの下に [gui] フォルダを作成する

60. [OK] をクリックしてください。

注：このステップは、RIIC ドライバの最適化バグの対策として行なう必要があります。

## 5. GUIX Studio による GUIX インターフェースの作成

ここで、ベースプロジェクトのセットアップができたので、GUIX コンポーネントの追加を開始できます。

1. [src] フォルダで右クリックし [New] -> [Folder] を選択することで、[src] の中に [gui] という名前の新規フォルダを作成してください。

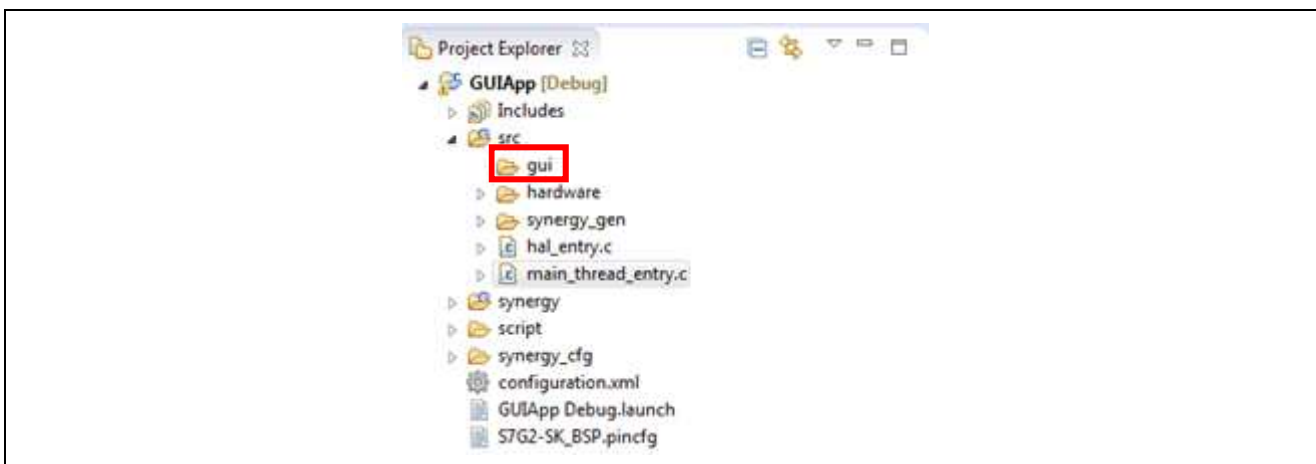


図 5.1 [src] フォルダの下に [gui] フォルダを作成する

2. [GUIApp] フォルダを右クリックし [New] -> [Folder] を選択することで、プロジェクトのルートフォルダの中に [guix\_studio] という名前のもう一つの新規フォルダを作成してください。最終的なフォルダ配置は、以下の図のようになります。

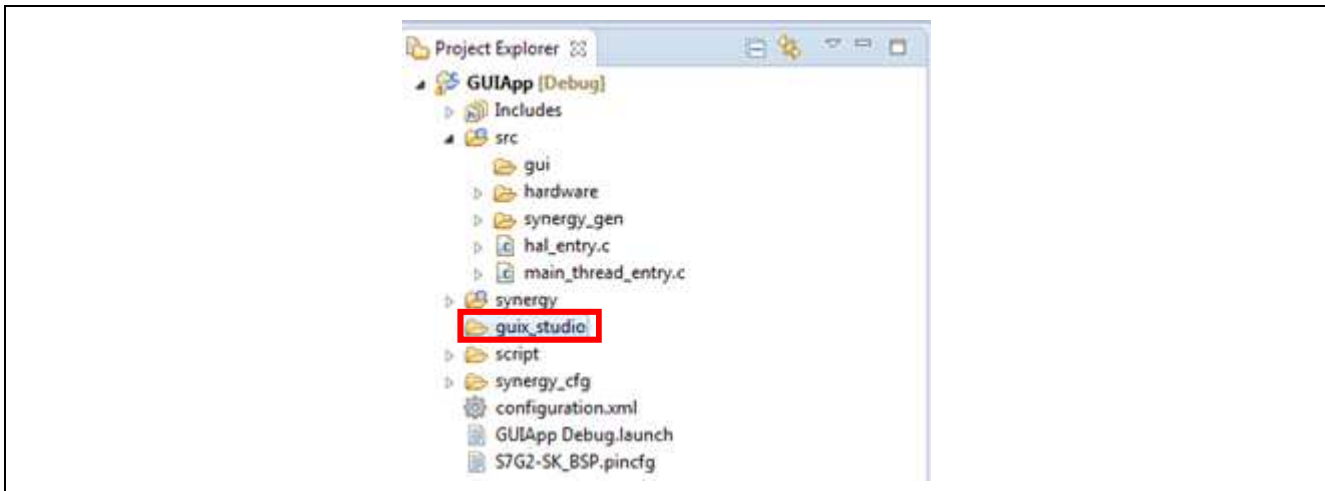


図 5.2 最終的なフォルダ配置

3. デスクトップのアイコンをクリックするか、Windows スタートメニューの**すべてのプログラム** > [Express Logic] -> [GUIX Studio 5.3] フォルダで GUIX アイコンをクリックするかにより、GUIX Studio を起動してください。



図 5.3 GUIX Studio の起動

4. Recent Projects ダイアログで、[Create New Project...] をクリックしてください。



図 5.4 [Create New Project]

5. 新規プロジェクトに [guiapp] という名前を付けます。

**重要：** ファイル名は、プロジェクト名に名前を加えたものになります。プロジェクト名は大文字小文字を区別するので、注意してください。後に、[guiapp] と命名されたプロジェクトにファイルを追加してください。

6. プロジェクトのパスについては、以前に[guix\_studio] と命名して作成したフォルダの場所を参照してください。

**注：** ツールをデフォルトのディレクトリにインストールした場合、そのフォルダの場所は以下になります。

[C:\Users\[User]\e2\_studio\workspace\GUIAPP\GUIApp\guix\_studio]



図 5.5 新しい GUIX プロジェクトの作成

7. [Save] をクリックしてください。
8. ディレクトリの 3 つのオプションを、[..\src\gui] に変更してください。

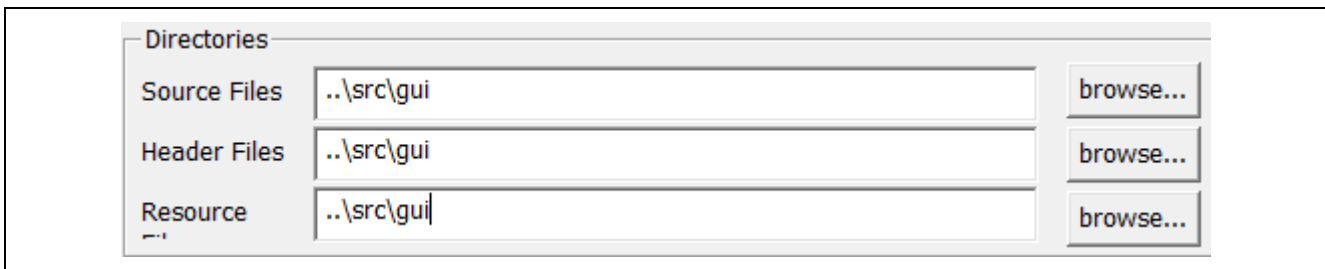


図 5.6 ファイルの場所の修正

重要：上記のディレクトリ名の前に、2 つのドット[..]を入れてあることを確認してください。

9. [Target CPU] の設定を、[Renesas Synergy] に変更してください。
10. [Toolchain] の設定を、[GNU] に変更してください。

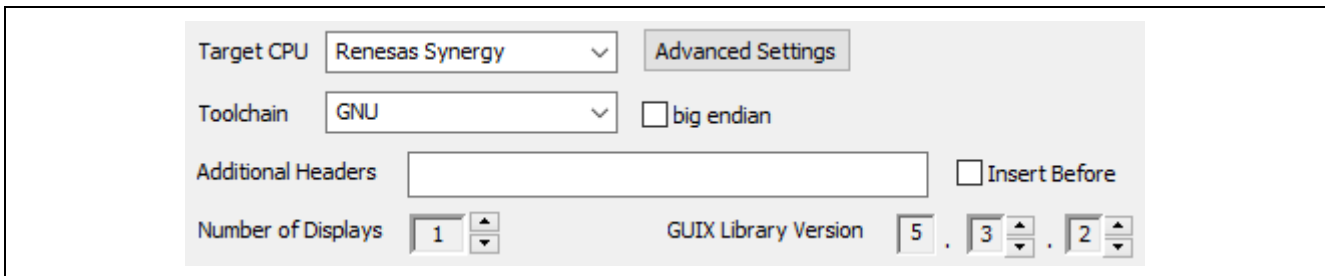


図 5.7 対象 CPU と GUIX バージョンの設定

11. [Advanced Settings] をクリックしてください。ダイアログが現れます。
12. 以下のようにして、グラフィックアクセラレータとハードウェア JPEG デコーダを有効にしてください。





図 5.8 [Synergy Advanced Settings]

13. [Save] をクリックしてください。
14. [Display Configuration] を以下に示すように設定してください。

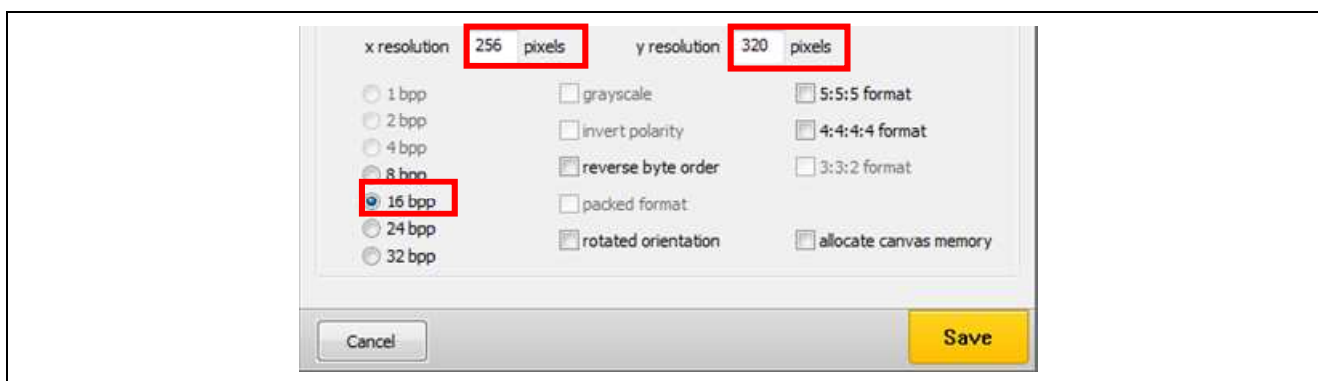


図 5.9 プロジェクトの設定

15. [Save] をクリックして、プロジェクトを作成してください。
16. プロジェクトビューで、[display\_1] を右クリックしてください。
17. [Insert] -> [Window] -> [Window] を選択してください。

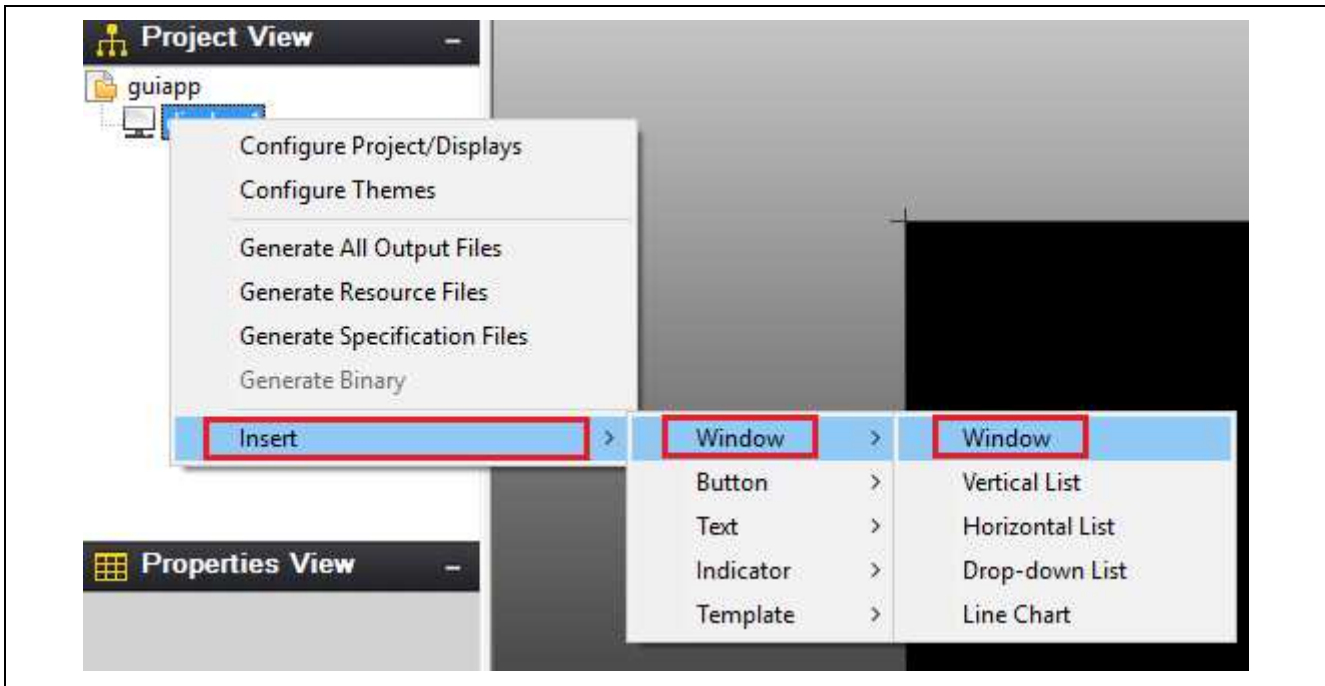


図 5.10 新規ウィンドウ

18. 新規ウィンドウを選択し、[Properties View] を編集してプロパティを変更します。現状の設定を以下に示すように変更してください。[Event Function] フィールドに注意してください。これは、[window1] でタッチスクリーンが押された時に発生するイベントです。

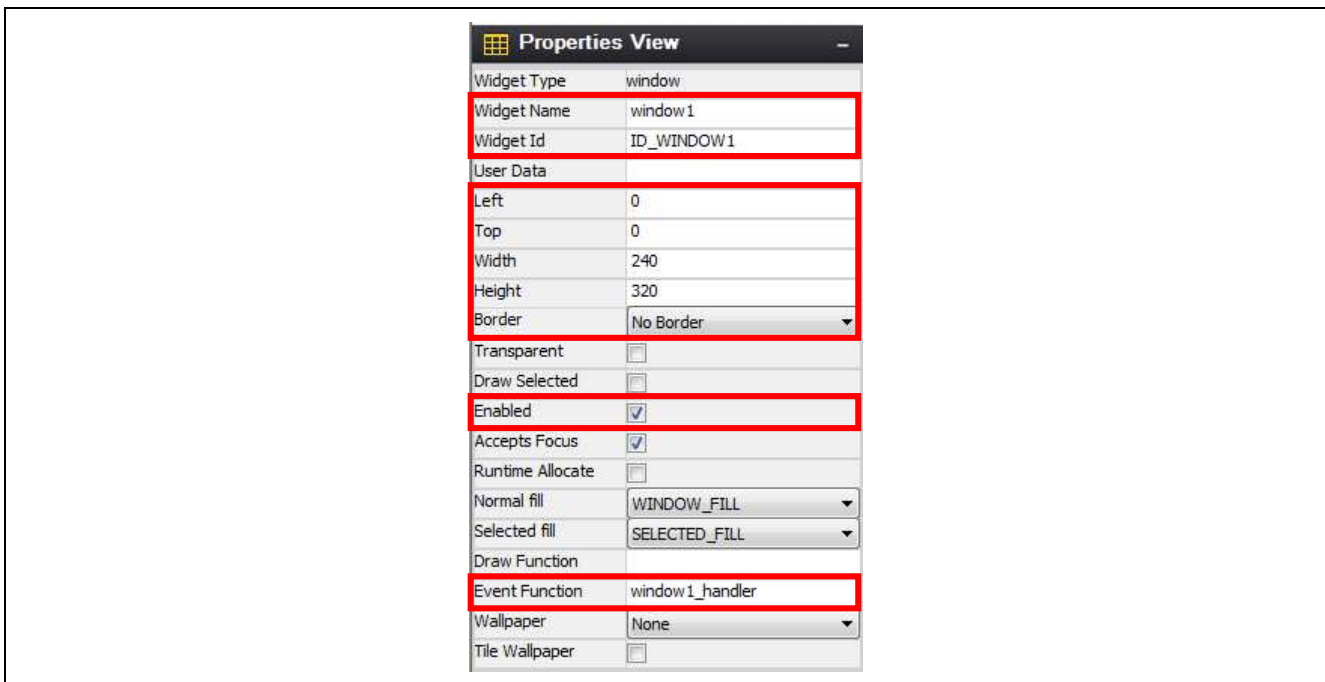


図 5.11 [Window1] プロパティの設定

19. この時点で、ウィンドウがディスプレイ全体を占めなくなります。これは、GUIX を小さな画面で動作させると起こります。一旦アプリケーションが動くとディスプレイに影響します。
20. プロジェクトビューウィンドウで、[display\_1] 上を右クリックし、[Insert] -> [Window] -> [Window] を選択してもう一つのウィンドウを生成してください。
21. プロパティを、以下に示すように変更してください。[Event Function] フィールドに注意してください。これは、[window2] でタッチスクリーンが押された時に発生するイベントです。

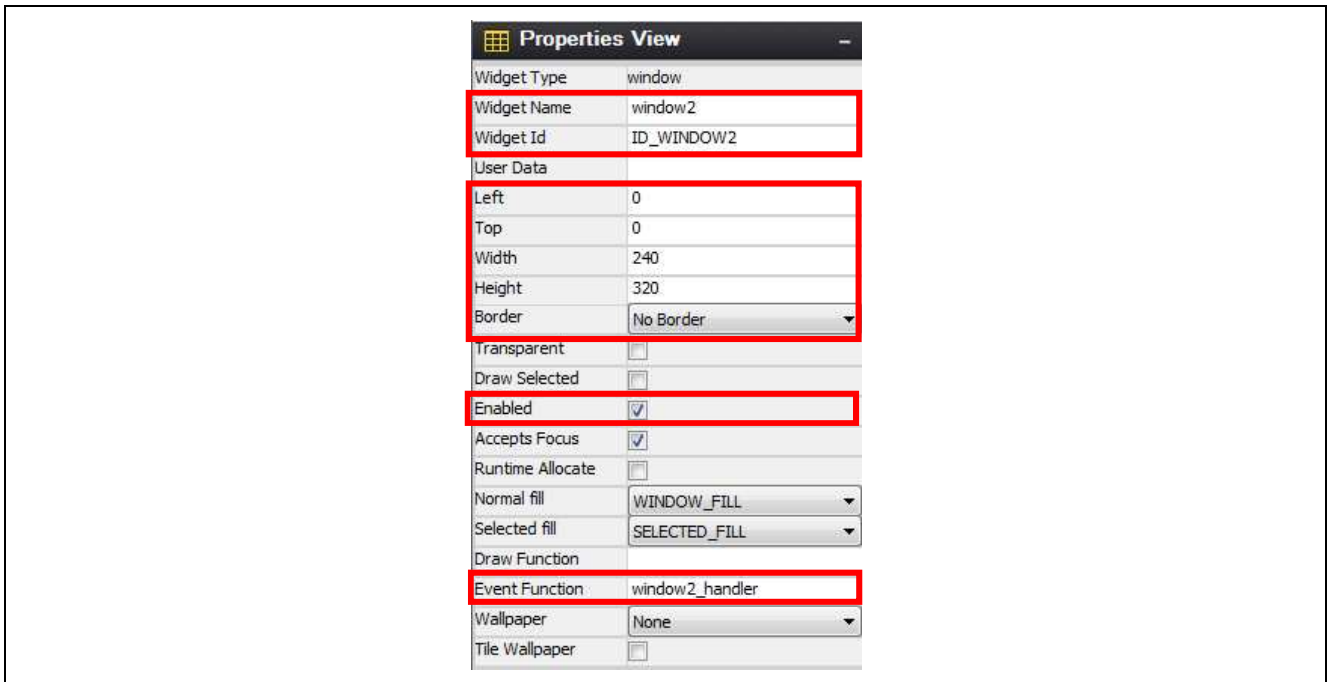


図 5.12 [Window2] プロパティの設定

22. プロジェクトビューで [window1] を右クリックし、[Insert] -> [Button] -> [Text Button] を選択してボタン [Text Button] を挿入します。

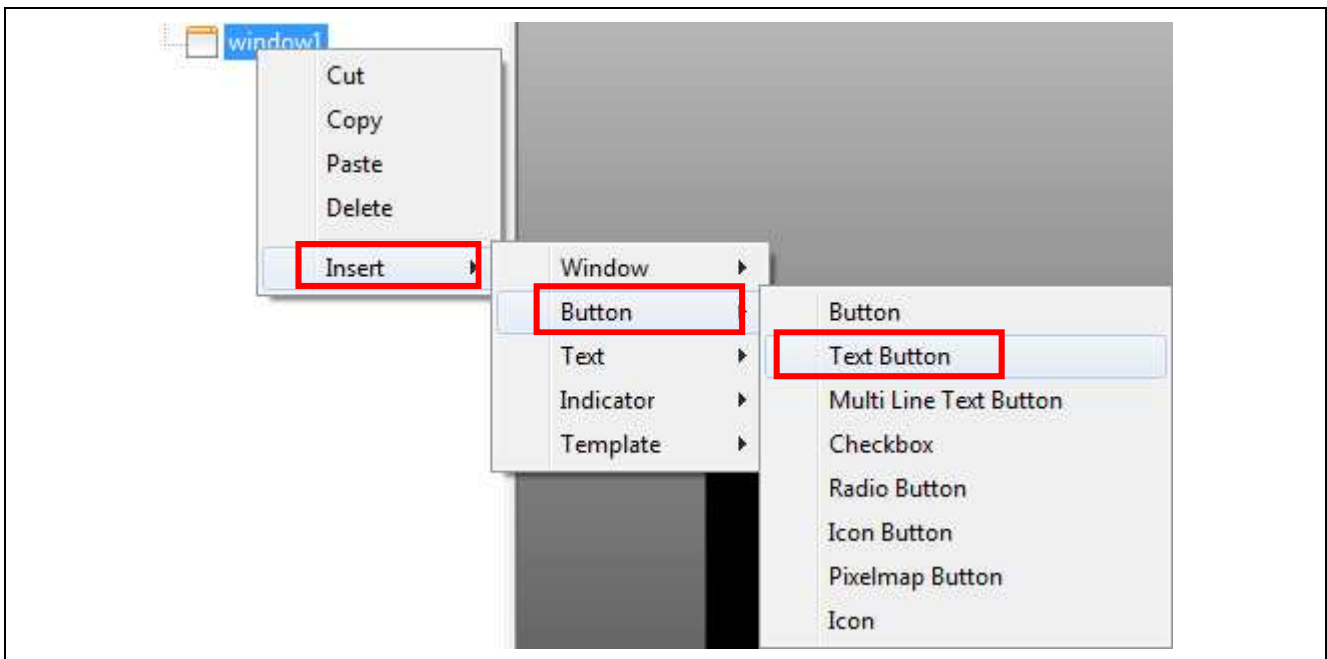


図 5.13 [Text Button] の新規追加

23. プロジェクトビューで [window1] を右クリックし、[Insert] -> [Button] -> [Checkbox] を選択してボタン [Checkbox] を挿入します。

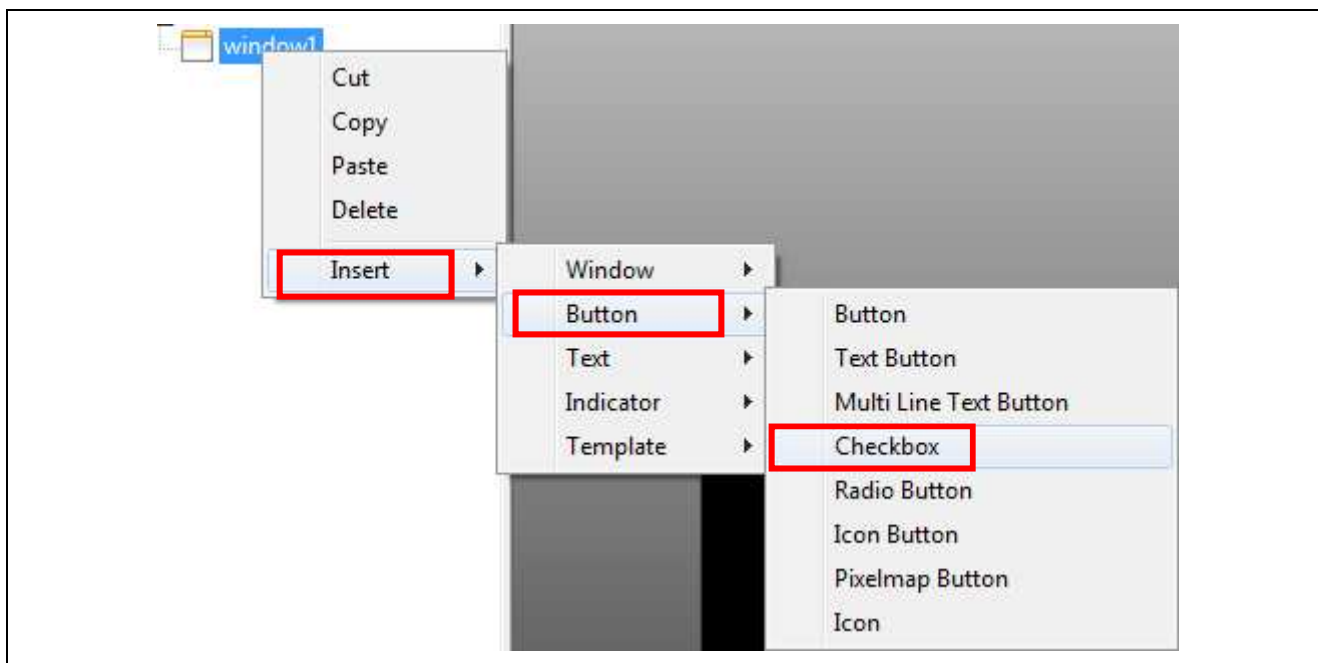


図 5.14 [Checkbox] の新規追加

24. プロジェクトビューで window1 を右クリックし、[Insert] -> [Text] -> [Prompt] を選択して [Text Prompt] を挿入します。

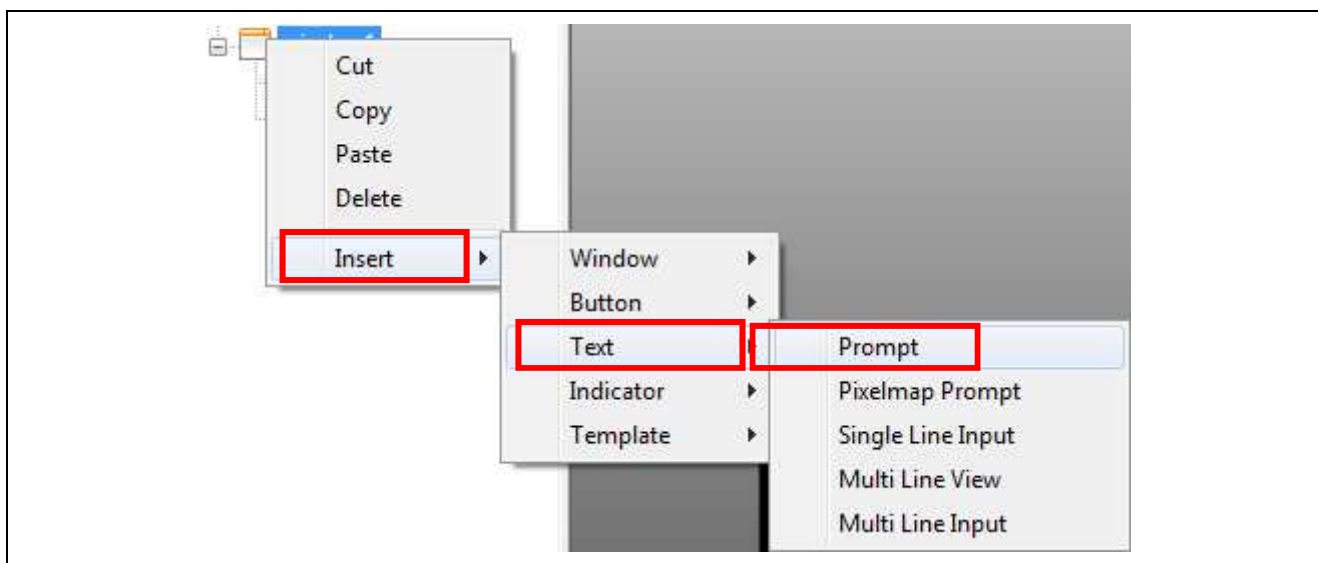


図 5.15 [Prompt] の新規追加

25. プロジェクトビューで [window1] を右クリックし、もう一つの [Text Prompt] を挿入します。  
 26. プロジェクトビューで [window2] を右クリックし、[Text Prompt] を挿入します。  
 27. プロジェクトビューで [window2] を右クリックし、もう一つの [Text Prompt] を挿入します。  
 28. この手順を正しく実行すると、プロジェクトビューは以下ようになります。

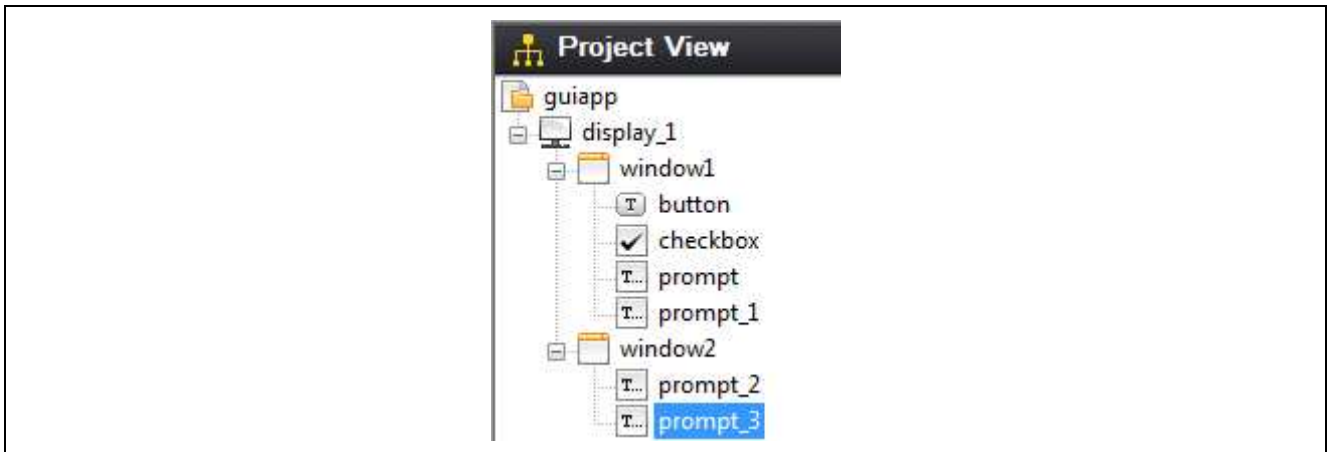


図 5.16 GUIX プロジェクトビュー

29. Strings の [+] をクリックして、[Strings] メニューを拡げてください。

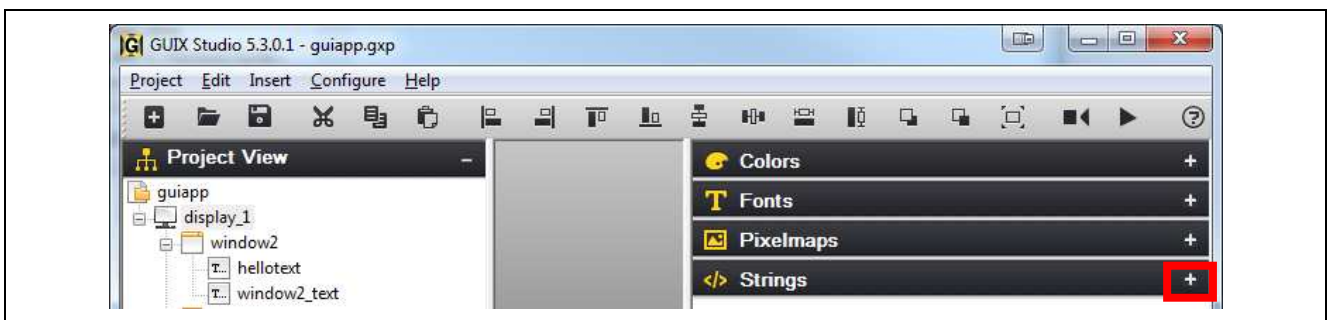


図 5.17 [Strings] ボタン

30. いずれかの [String] をダブルクリックし、[String Table Editor] を開いてください。

31. [String Table Editor] で既存の [String] を選択し、[Delete String] ボタンをクリックして削除してください。

32. [Add String] ボタンを使って以下の [String] を追加してください。



図 5.18 新規 [String]

33. 終了したら、[Save] をクリックしてください。

34. [window1] のプロジェクトビューでボタンをクリックし、プロパティ [Properties View] を以下に示すように変更してください。

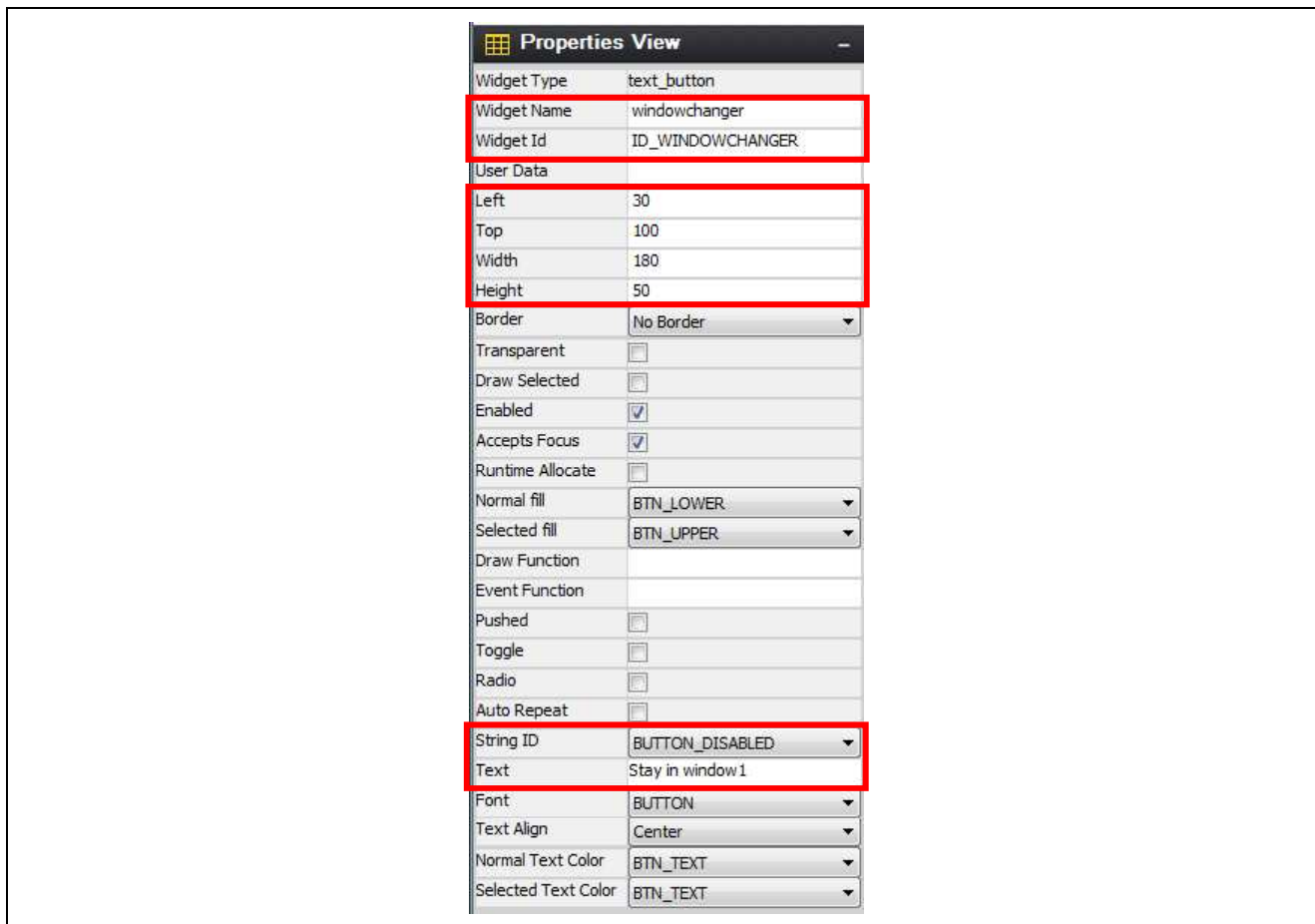


図 5.19 [Windowchanger] ボタンプロパティの設定

35. [window1] のプロジェクトビューで、チェックボックスをクリックし、プロパティ [Properties View] を以下に示すように変更してください。

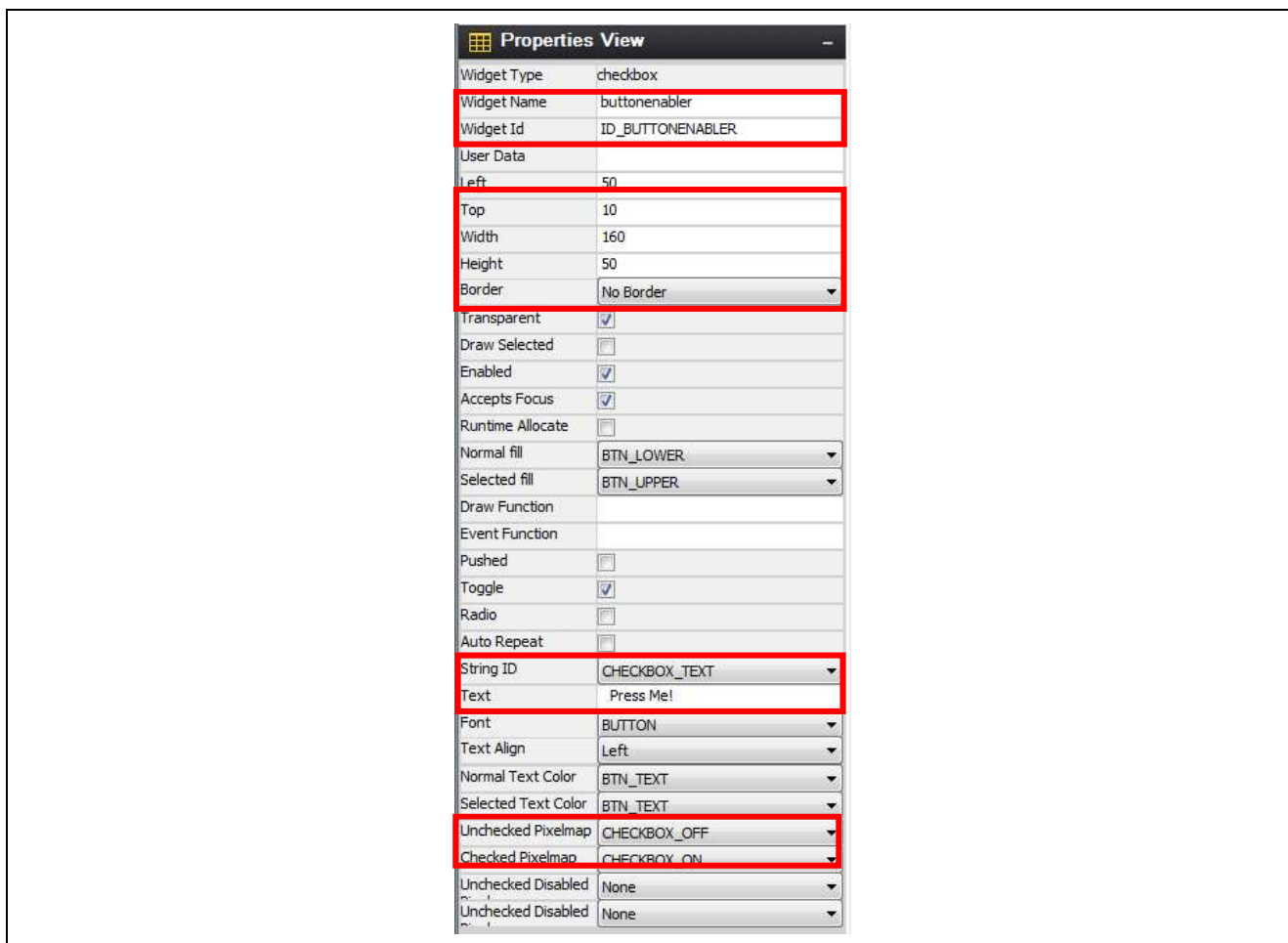


図 5.20 [Buttonenabler] チェックボックスプロパティの設定

36. [window1] のプロジェクトビューで、プロンプトをクリックし、プロパティを以下に示すように変更してください。

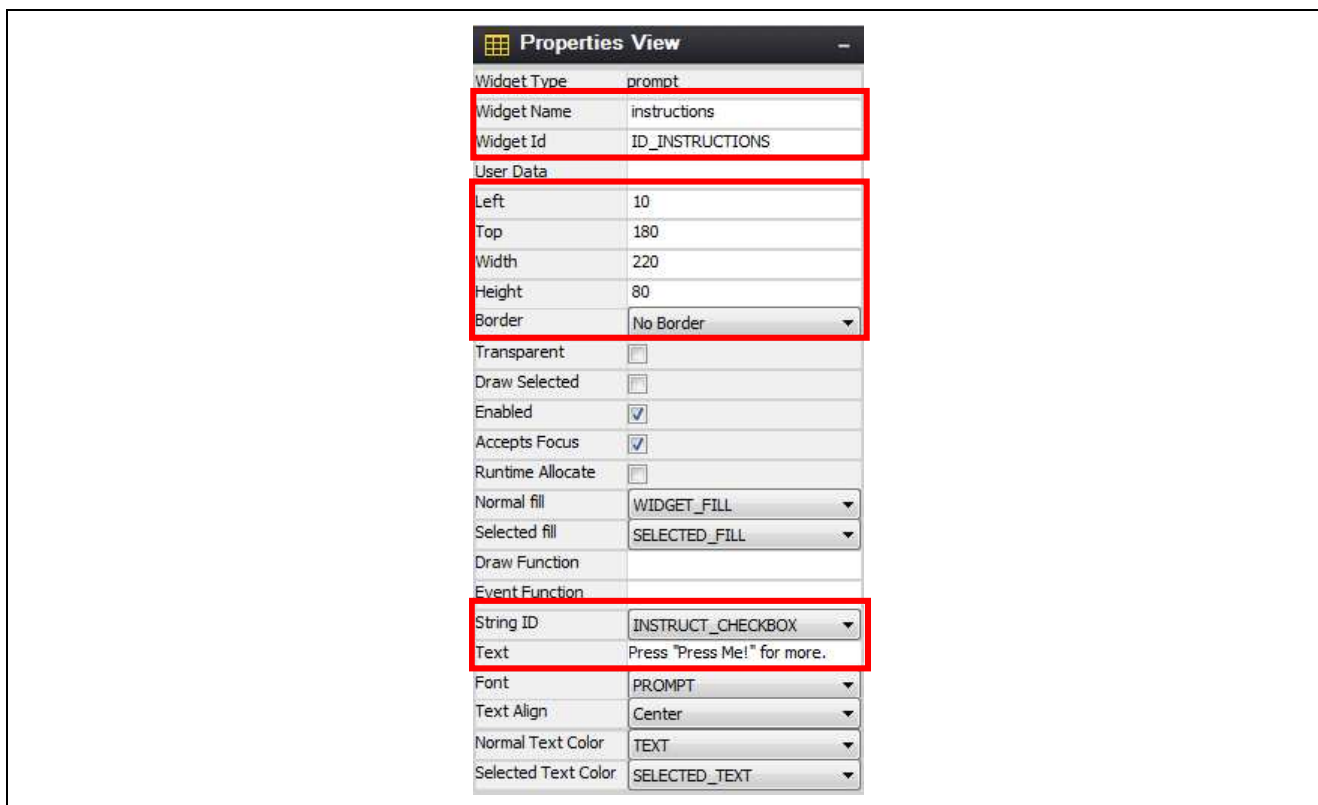


図 5.21 プロンプトプロパティの設定

37. [window1] のプロジェクトビューで、[Prompt] をクリックし、プロパティを以下に示すように変更してください。



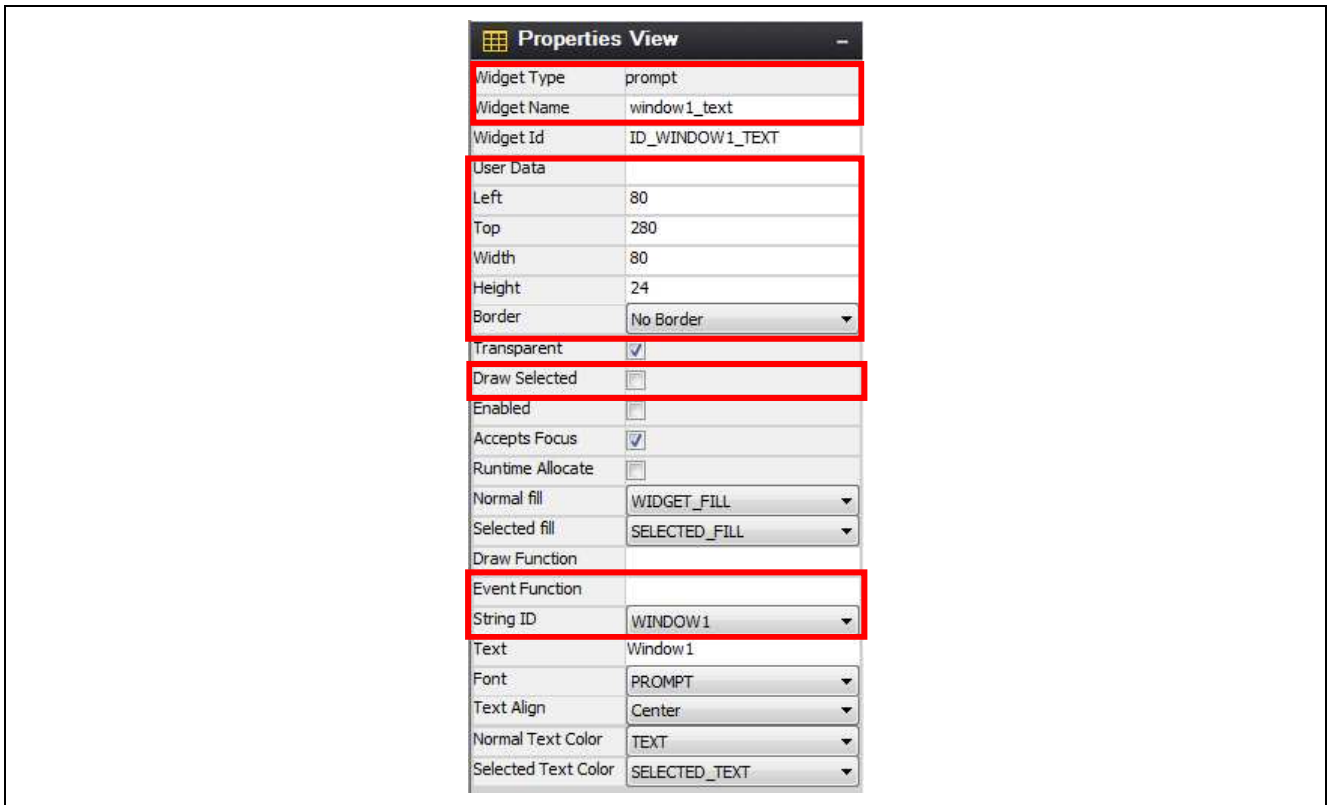


図 5.22 [Window Text] プロパティの設定

38. [window2] のプロジェクトビューで、[prompt\_2] をクリックし、プロパティを以下に示すように変更してください。

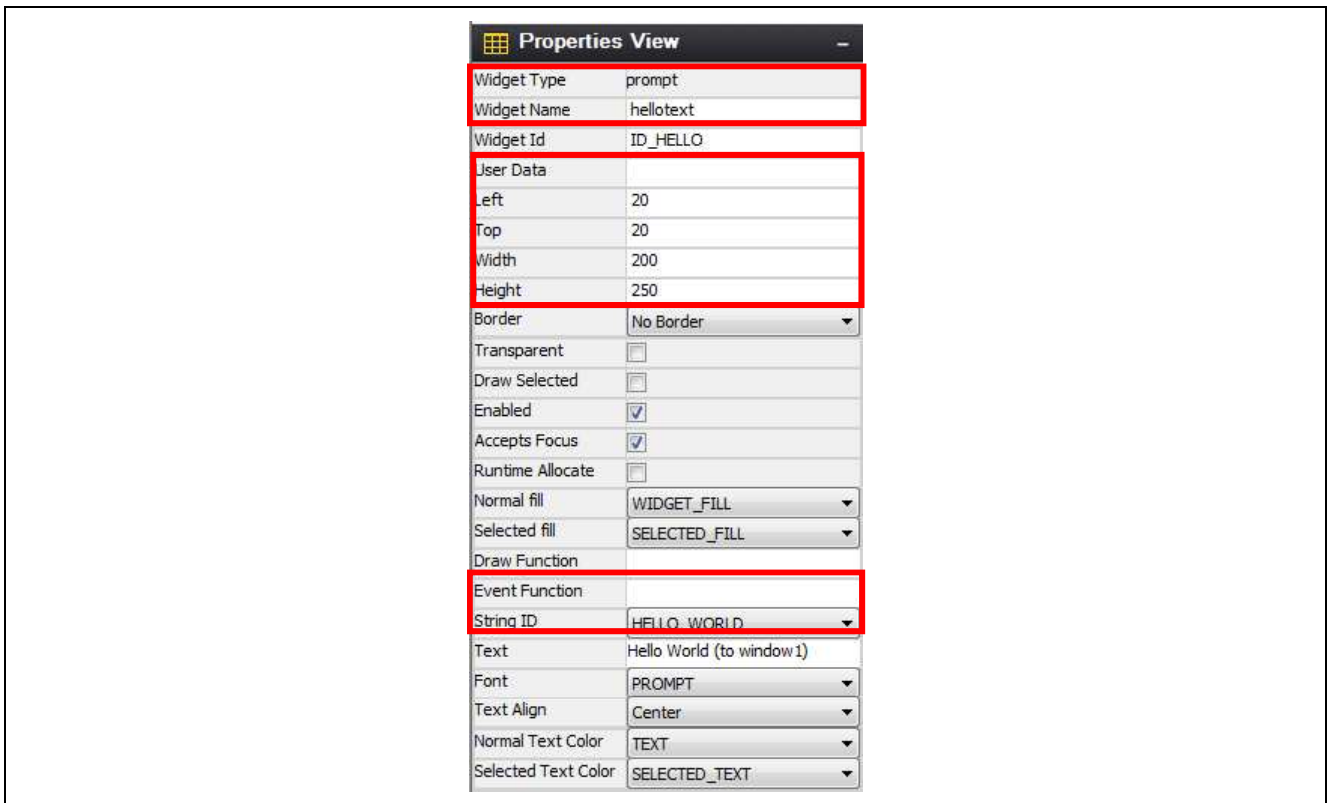


図 5.23 [Hello Text Prompt] プロパティの設定

39.[window2] のプロジェクトビューで、[prompt\_3] をクリックし、プロパティを以下に示すように変更してください。

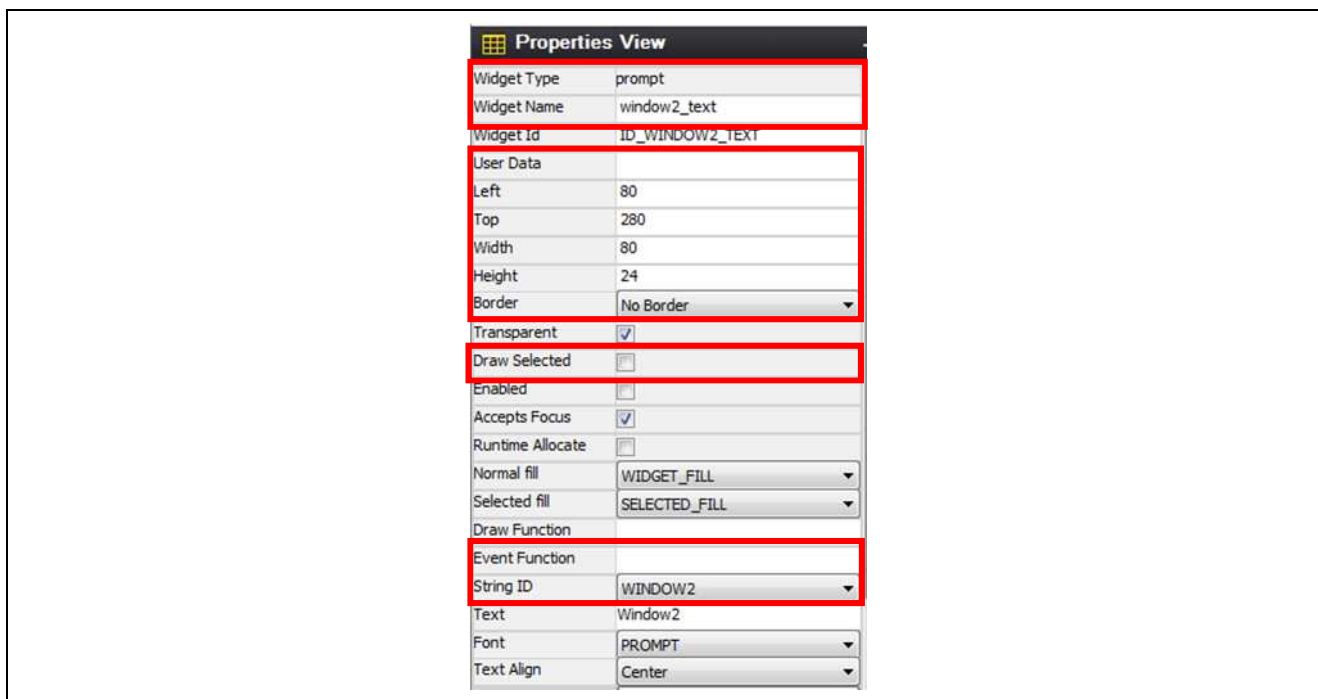


図 5.24 [Window Text] プロパティの設定

これらの設定ステップの後、ふたつのウィンドウは、以下の画像のようになります。



図 5.25 設定後の [window1]

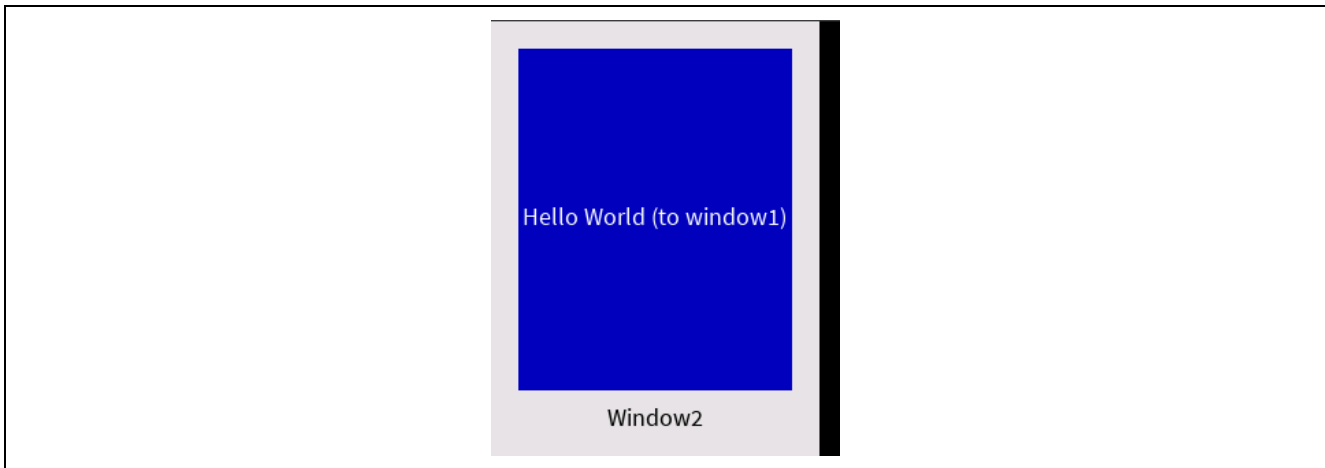


図 5.26 設定後の [window2]

40. プロジェクトを保存してください。

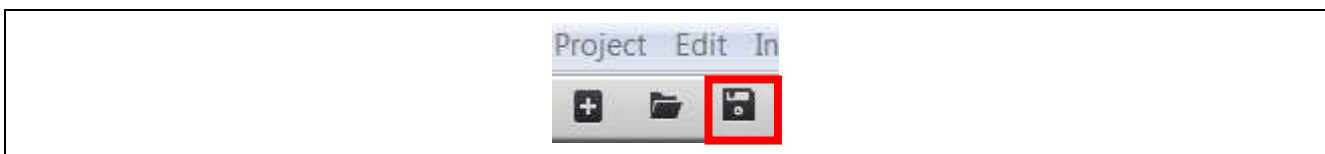


図 5.27 プロジェクト保存

41. Project メニューから、[Generate All Output Files] を選択してください。

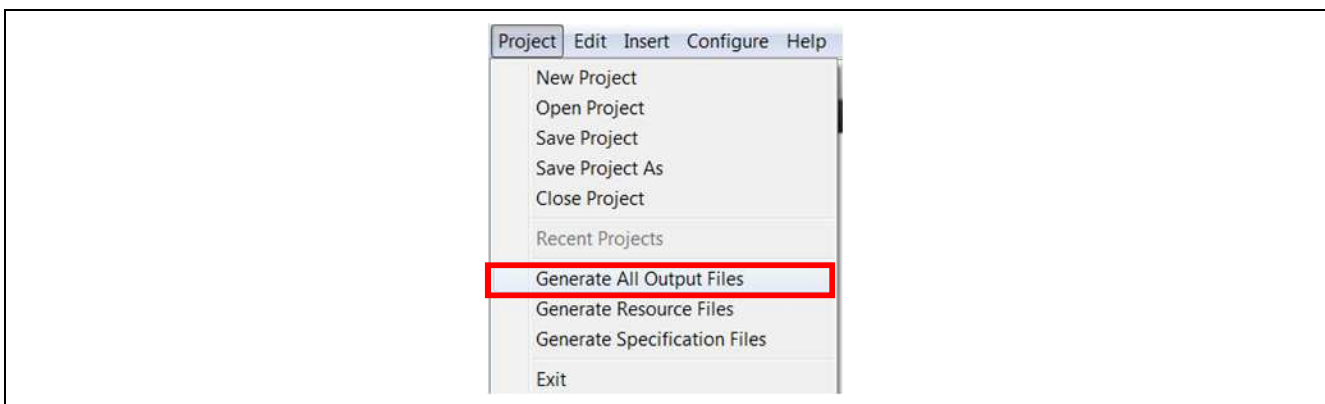


図 5.28 全ての出力ファイルの生成

42. e<sup>2</sup>studio に戻ってください。

## 6. カスタムインターフェースコントロールのコード追加およびプロジェクトのビルド

1. Windows エクスプローラで、このアプリケーションノートが含まれていたファイルを格納したフォルダに行き、[Source Files\guiapp\_event\_handlers.c] ファイルを見つけてください。ここで、そのファイルを Windows エクスプローラ画面から、e<sup>2</sup>studio プロジェクトエクスプローラ画面の中にある [src] フォルダに直接ドラッグしてください。
2. 選択したファイルをインポートするプロンプトが表示されたら、[OK] をクリックしてファイルをコピーしてください。

注：このファイルは、GUIX Studio ([window1]、[window2])で生成した異なるグラフィックエレメントのためのイベントマネジメント機能を含んでいます。

GUIX はシステムレベルで要求されたイベントを取り扱いますが、スクリーン遷移やボタン動作などのカスタムコマンドを扱うためには、イベントハンドラ定義が必要です。[window1] のイベントハンドラを以下に示します。

```
UINT window1_handler(GX_WINDOW *widget, GX_EVENT *event_ptr)
{
    UINT result = gx_window_event_process(widget, event_ptr);

    switch (event_ptr->gx_event_type)
    {
        case GX_SIGNAL(ID_BUTTONENABLER, GX_EVENT_TOGGLE_ON):
            button_enabled = true;
            update_text_id(widget->gx_widget_parent, ID_WINDOWCHANGER,
GX_STRING_ID_BUTTON_ENABLED);
            update_text_id(widget->gx_widget_parent, ID_INSTRUCTIONS,
GX_STRING_ID_INSTRUCT_BUTTON);
            break;
        case GX_SIGNAL(ID_BUTTONENABLER, GX_EVENT_TOGGLE_OFF):
            button_enabled = false;
            update_text_id(widget->gx_widget_parent, ID_WINDOWCHANGER,
GX_STRING_ID_BUTTON_DISABLED);
            update_text_id(widget->gx_widget_parent, ID_INSTRUCTIONS,
GX_STRING_ID_INSTRUCT_CHECKBOX);
            break;
        case GX_SIGNAL(ID_WINDOWCHANGER, GX_EVENT_CLICKED):
            if(button_enabled){
                show_window((GX_WINDOW*)&window2, (GX_WIDGET*)widget, true);
            }
            break;
        default:
            gx_window_event_process(widget, event_ptr);
            break;
    }

    return result;
}
```

イベントは、ウィジェットの ID と GUIX からの信号に基づいて処理されます。例えば、チェックボックス [ID\_BUTTONENABLER] には、2 つの状態：[GX\_EVENT\_TOGGLE\_ON] と [GX\_EVENTS\_TOGGLE\_OFF] をとらせることができます。ボックスがチェックされずに押された場合、イベント [GX\_EVENT\_TOGGLE\_ON] がハンドラに送られ、それからボックスがチェックされます。

3. 最適化をオフにする：
  - A. プロジェクト・エクスプローラーウィンドウの [GUIApp] を右クリックして、コンテキスト・メニューから [Properties] を開きます。
  - B. プロパティウィンドウで、[C/C++ Build] ツリーを開きます。
  - C. [Settings] を選んでください。
  - D. [Tool Settings] タブで、[Optimization] をクリックしてください。
  - E. [Optimization Level] を[-O0]に変更してください。
  - F. [OK] をクリックし、変更を保存してください。

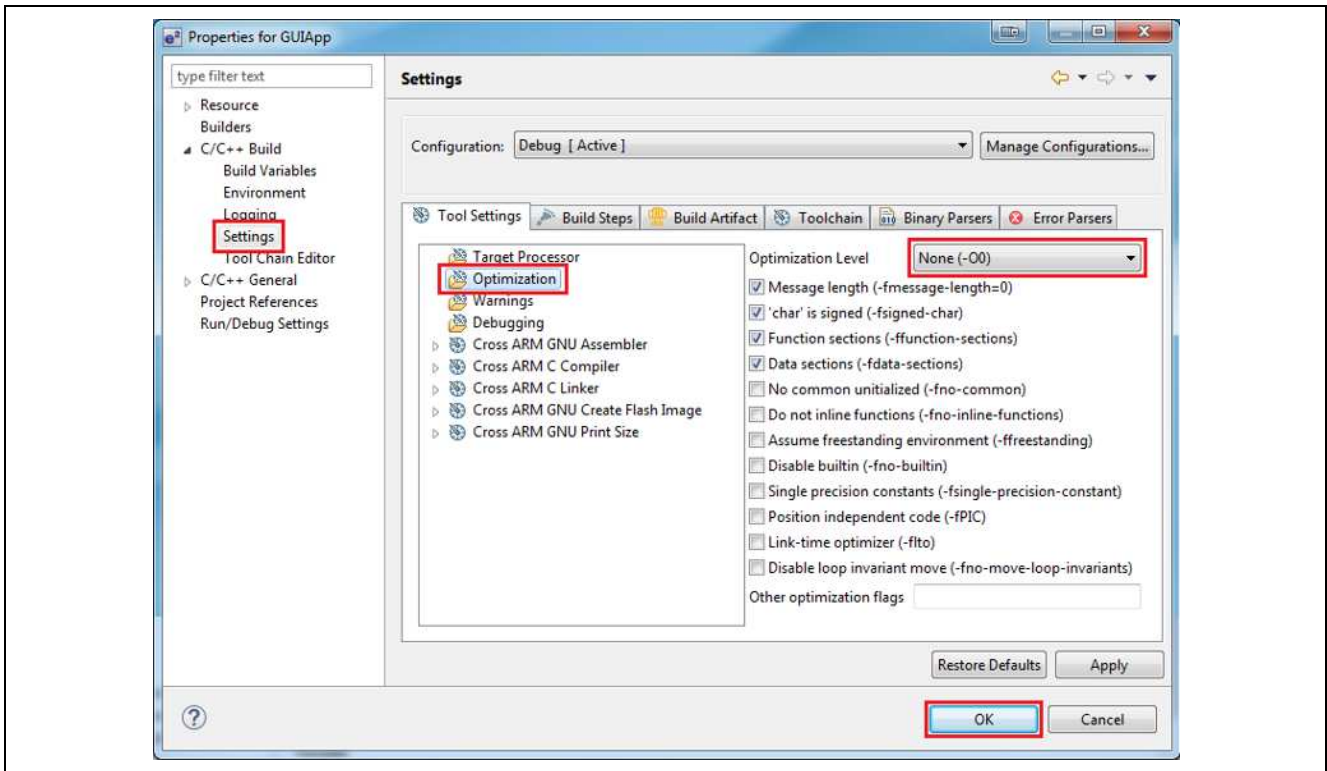


図 6.1 コンパイラ最適化の無効化

4. メニューバーの下のハンマー型のアイコンをクリックして、プロジェクトをビルドしてください。全てのステップを正しく実施すると、ビルドの出力時にエラーレポートは出力されません。

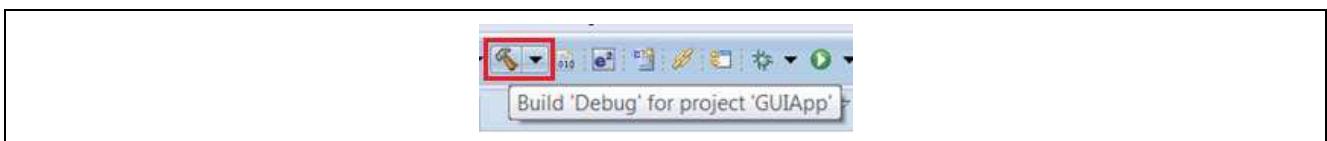


図 6.2 プロジェクトのビルド

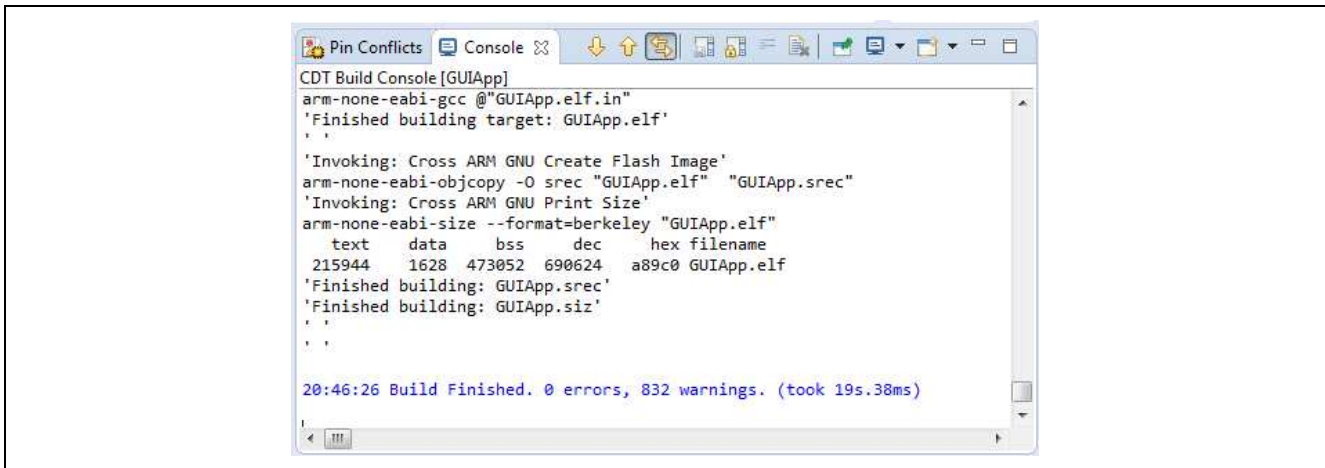


図 6.3 エラーなしでビルドが完了した場合

## 7. アプリケーションプログラムの作動

1. マイクロ USB ケーブルで、SK-S7G2 (J19) を PC に接続してください。  
注：アプリケーションを対象ハードウェアで実行する準備は、まだできていません。実行するには、以下の手順が必要です。
2. [debug] アイコンのプルダウンメニューをクリックしてください。

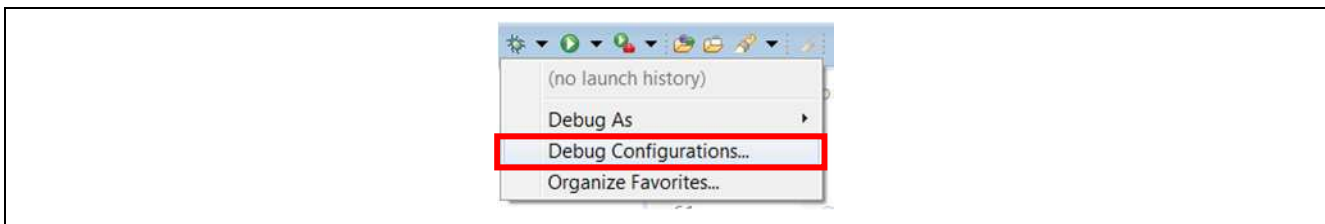


図 7.1 デバッグオプション

3. [Debug Configurations...] オプションを選択してください。
4. [Renesas GDB Hardware Debugging] にある、[GUIApp Debug] を選択してください。
5. デバッグを開始するため、[Debug] ボタンをクリックしてください。  
注：[Debug] ボタンがグレーアウトしている場合は、おそらくビルドで問題がおきています。オプションのミスマッチについて手順全体をドキュメントから再確認してください。

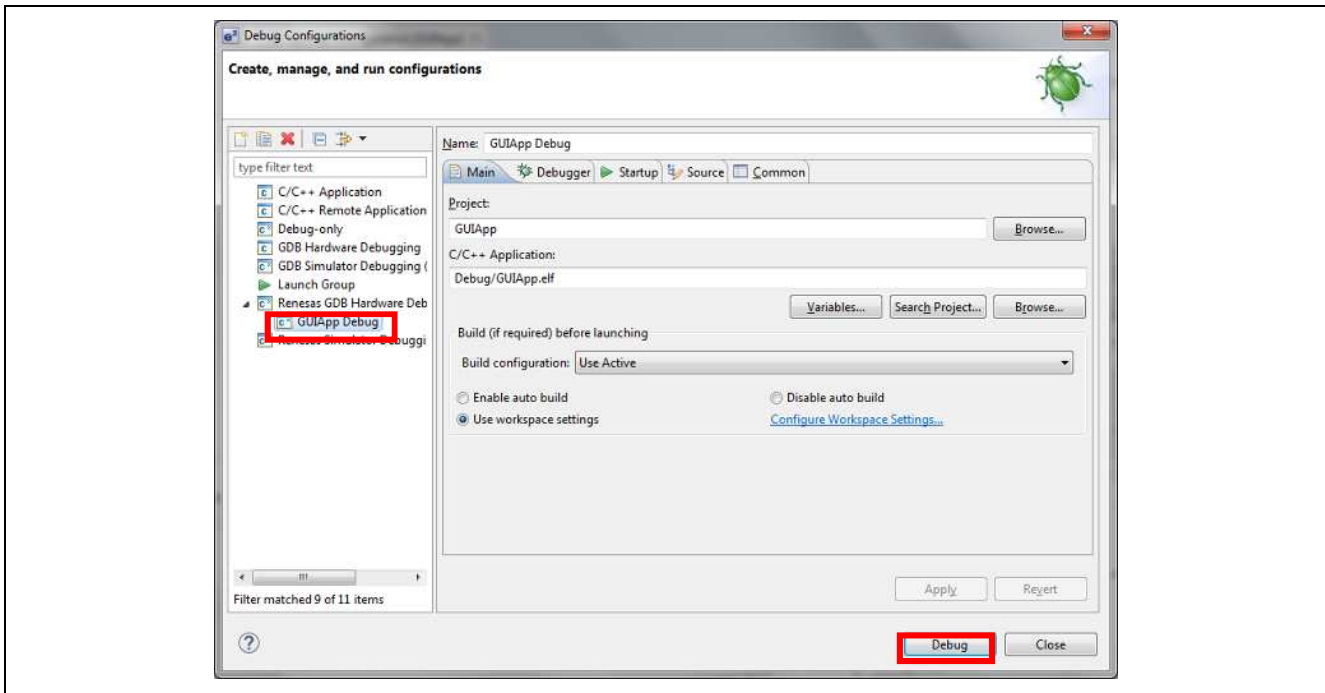


図 7.2 デバグの設定

6. パースペクティブスイッチを確認するかを聞かれたら、[Yes] をクリックしてください。（e<sup>2</sup>studio で既にこの指定をしてある場合は、このダイアログボックスが表示されません。）

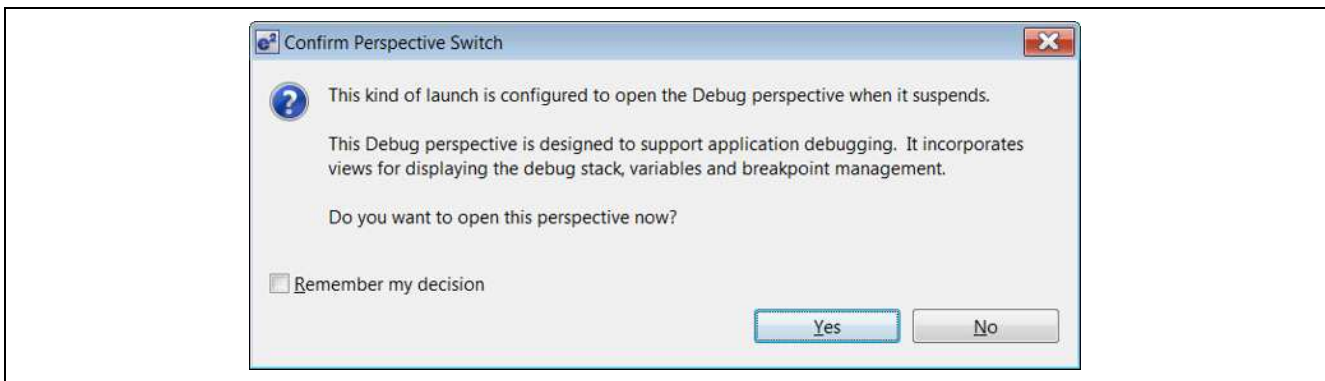


図 7.3 [Perspective Switch] ダイアログボックス

7. ファンクションキー-F8 を押すか [Resume] ボタンを押して、アプリケーションを起動してください。アプリケーションは[main] で停止します。



図 7.4 [Resume] ボタン

8. ファンクションキー-F8 を押すか [Resume] ボタンを押して、コードを実行してください。  
注：ここで、以前作成した GUI がスクリーンに表示されます。

9. デモンストレーションの概要

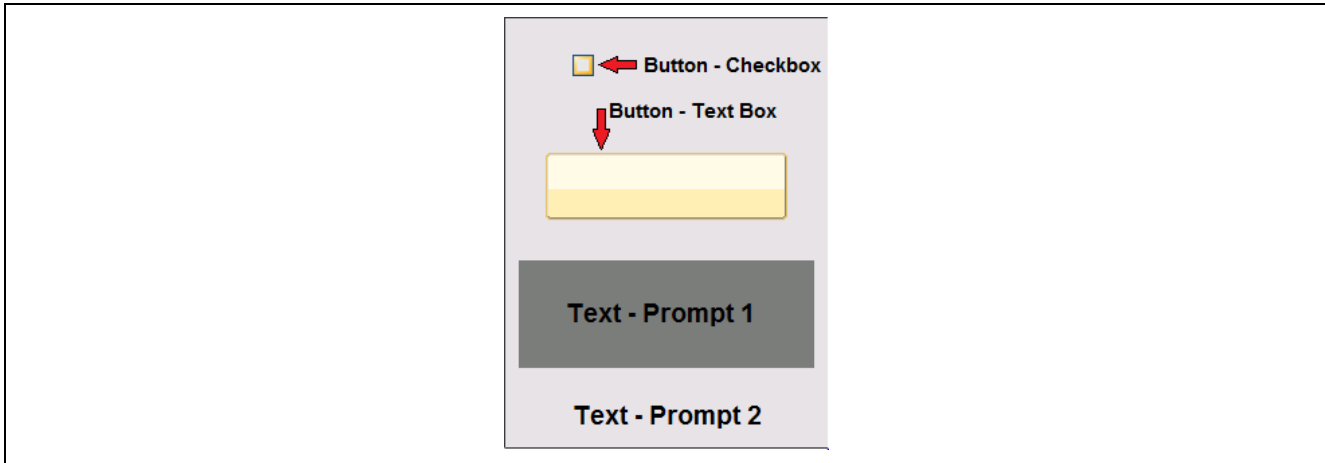


図 7.5 Window1

- A. 図 7.5 に、[Window1] を示します。このウィンドウには 4 つの要素があります。
- ボタン - [Checkbox]** : [Window2] への移動を有効にするため、このボタンを使います。デフォルト設定はテキストが [Press Me!] となり、チェックがつかない状態です。チェックボックス内のアクティブ領域をクリックすると、[window1\_handler] イベントをアクティブにします。  
[guiapp\_event\_handlers.c] の内部で、このイベントは起動します。  
[guiapp\_event\_handlers.c] はコードでチェックボックスを切り替えることで、[Text - Prompt 1] と [Button - Text Box] のテキストを適切なメッセージに設定します。
  - [Button - Text Box]** : このボックスは、[Text - Prompt 1] 領域の外をクリックした場合に、どのウィンドウに移動するかを示すものです。（どのように変わるかは、[Button - Checkbox] を参照してください。） [window1\_handler] イベントを起動するため、この領域をクリックしてください。  
[window1\_handler] イベントは、コードを [window2] に切り替える、  
[guiapp\_event\_handlers.c] によって起動されます。
  - [Text - Prompt 1]** : この領域は、デモをどのように制御するかを指示します。（どのように変わるかは、[Button - Checkbox] を参照してください。）
  - [Text - Prompt 2]** : このプロンプトは、ユーザに現在どのウィンドウにいるのかを示します。常に [window1] を表示し、変わることはありません。

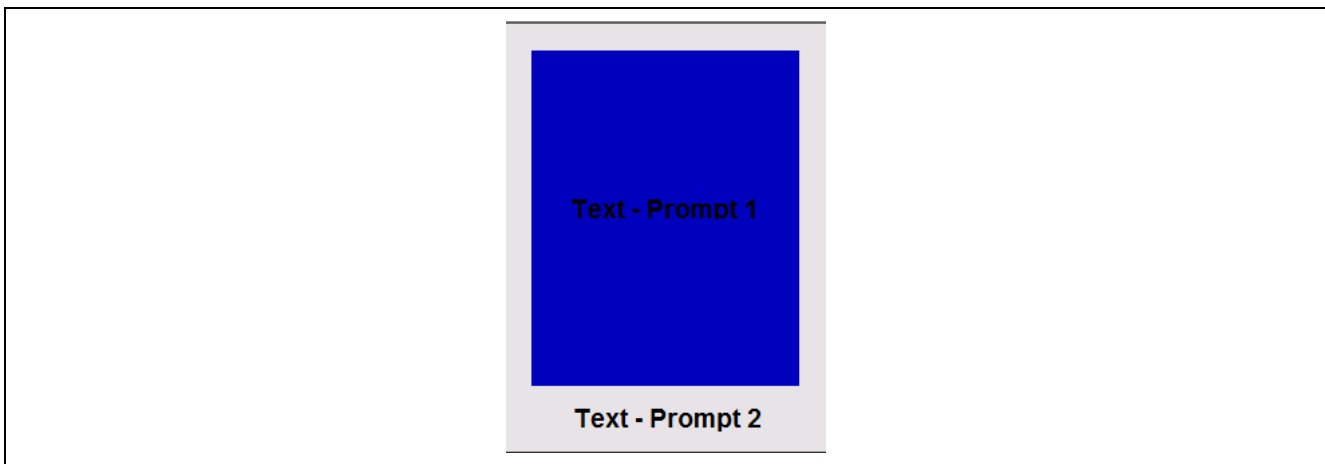


図 7.6 Window2

- B. 図 7.6 に、[Window2] を示します。このウィンドウには 2 つの要素があります。
- [Text - Prompt 1]** : この領域は、[Hello World] と表示します。この領域をクリックすると、[window2\_handler] イベントを起動します。 [window2\_handler] イベントは、  
guiapp\_event\_handlers.c を起動し、アクティブウィンドウを [window1] に切り替えます。



- b. [Text - Prompt 2] : このプロンプトは、ユーザが現在どのウィンドウにいるのかを示します。常に [window2] を表示し、変わることはありません。
10. [Ctrl + F2] か [stop] ボタンを押して、デバッグセッションを終了してください。

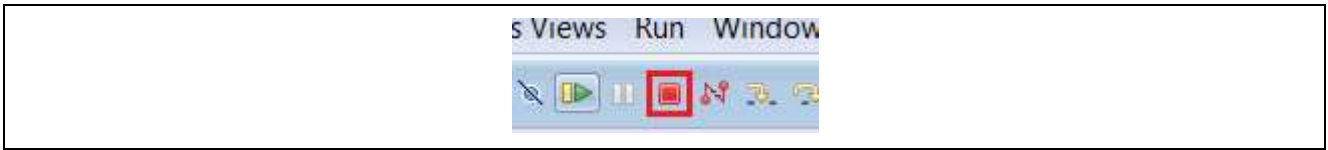


図 7.7 [Stop] ボタン

11. 以上で、SK-S7G2 用 GUIX “ Hello World ” を終わります。

## ホームページとサポート窓口

サポート <https://synergygallery.renesas.com/support>

技術的な質問の窓口の詳細

- 米国: [https://renesas.zendesk.com/anonymous\\_requests/new](https://renesas.zendesk.com/anonymous_requests/new)
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

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。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2016.01.22	-	初版
1.01	2016.01.12	-	セマフォ命名の問題を修正するため、lcd_setup.c を修正。
1.10	2016.08.30	-	SSP 1.1.0 バージョンに更新
1.11	2016.11.18	-	書式の小有変更。
1.12	2016.01.06	-	SSP v1.2.0.b.1 バージョンに更新
1.13	2017.02.28	-	SSP v1.2.0 バージョンに更新

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 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・システムの設計において、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て生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 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。
  4. 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製等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。
  5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
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、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しており、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
  9. 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・システムに使用することはできません。また、当社製品および技術は、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。  
当社製品および技術を輸出、販売または移転等する場合は、「外国為替及外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
  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禁じます。
  12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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。

(Rev.3.0-1 2016.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>