Renesas Synergy™ Project

GUIX "Hello World" for SK-S7G2

R12AN0021EU0110
Rev.1.10
Aug 30, 2016

Introduction

This application note describes the process of creating a simple two screen GUI using GUIX Studio for the SK-S7G2. This application demonstrates how easily you can create and configure a new application using the Renesas Synergy Software Package (SSP).

The Synergy Software Package includes Express Logic’s ThreadX real-time operating system (RTOS), the X-Ware suite of stacks (NetX, USBX, GUIX, FileX) and hardware drivers unified under a single robust framework. This powerful suite of tools provides a comprehensive integrated framework for rapid development of complex embedded applications.

The “Hello World” application was developed within e2 Studio using the Synergy Framework.

Target Device

SK-S7G2 board version 2.0

Minimum PC Recommendation

- Microsoft® Windows® 7
- Intel® Core™ family processor running at 2.0 GHz or higher (or equivalent processor)
- 8 GB memory
- 250 GB hard disk or SSD
- USB 2.0
- Connection to the Internet

Installed Software

- Synergy e^2 studio 5.0.0.043 or later
- Synergy Software Platform (SSP) v1.1.0
- GUIX Studio v5.3.0.1

Note: If you do not have one of these software applications you should install it before continuing.

Provided Software Files:

- guiapp_event_handlers.c
- main_thread_entry.c
- lcd_setup.c
- lcd.h

Purpose

This document will guide you through the setup of a GUIX touch screen interface “Hello World” application in e^2 studio. It covers all configuration steps including setting up the hardware functions (LCD, SPI, and I2C interface), setting up threads and message passing, setting up the interrupts, setting up the LCD driver, and setting up the touchscreen. This document covers all steps necessary to create the GUI interface using the GUIX Studio editor. In addition, this application note covers the basic project setup in e^2 studio along with basic debugging operations. When it is running, the application will respond to touchscreen actions, presenting a basic graphical user interface (GUI).

Intended Audience

The intended audience are users who want to design GUI applications.

Contents

1. Overview

This document explains the steps to setup a project and develop a simple GUI based application using GUIX Studio.

2. Import the Project into e^2 studio

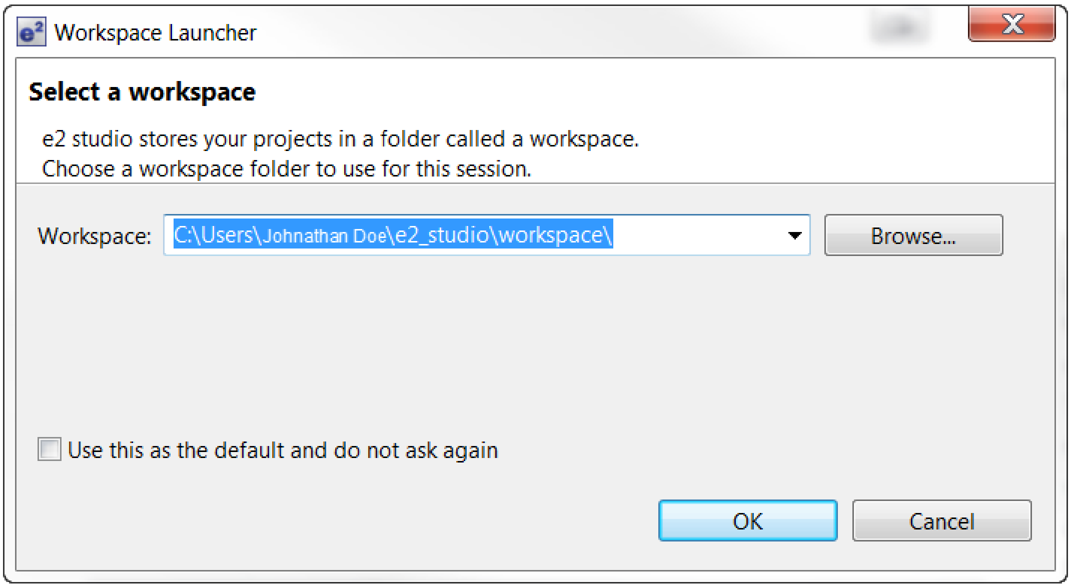
Note: This step is included to give you the ability to skip the development steps and start at the point of verifying a working project on the SK-S7G2. Most people SKIP THIS STEP and proceed to step 3 to create a project in e2Studio. Also, if you do import the project, then skip to section 7. Running the Application.

To skip the development walkthrough in this document and open a completed project in e^2 studio, refer to the "Synergy Project Import Guide.pdf" for instructions on importing the project into e^2 studio and building the project. The included “GUIX_Hello_World_SK-S7G2.zip” file contains the completed project.

3. Creating the Project in e^2 studio ISDE (Integrated Solution Development Environment)

Start by creating a new project in e^2 studio:

1. Open e^2 studio by clicking the “e^2 studio” icon in the Windows Start Menu > “All Programs > Renesas Electronics e2studio” folder.
2. If the workspace launcher dialog box appears, click OK to use the default workspace.



Workspace Launcher

Select a workspace

e2 studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Users\Johnathan Doe\e2_studio\workspace\

□ Use this as the default and do not ask again

Figure 1 Workspace Launcher Dialog

3. Create a new workspace:

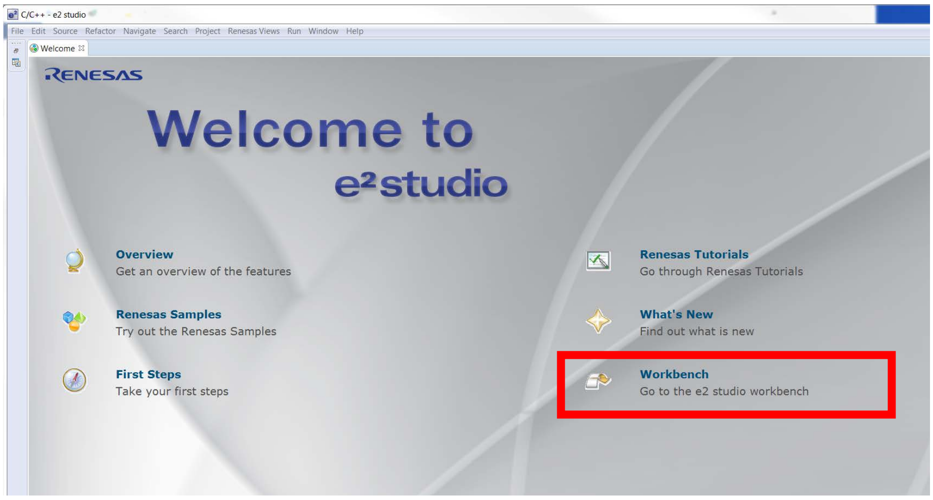
   A. From the “File” drop-down menu select “Switch Workspace > Other…”

4. Append a workspace name:

   A. In the Workspace Launcher window add text to the end of the workspace name to make it unique. We suggest “GUI_APP”. If you installed to the default location, the new workspace name will be “C:\Users\[your name]\e2_studio\workspace\GUI_APP”.

5. Click OK to create the new workspace.

6. Proceed past the Welcome Screen by clicking the Workbench area.



C/C++ - e2 studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Welcome
RENESAS

Welcome to e²studio

Overview
Get an overview of the features

Renesas Samples
Try out the Renesas Samples

First Steps
Take your first steps

Renesas Tutorials
Go through Renesas Tutorials

What's New
Find out what is new

Workbench
Go to the e2 studio workbench

Figure 2 Close the Welcome Window by clicking in the Workbench Area

7. Start a new project by clicking the black triangle, ▼, next to the New icon in the Tool Bar.

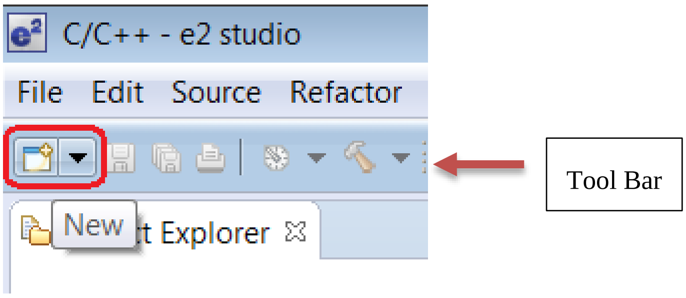


Figure 3 Start a New Project

8. Select "Synergy Project" from the menu.

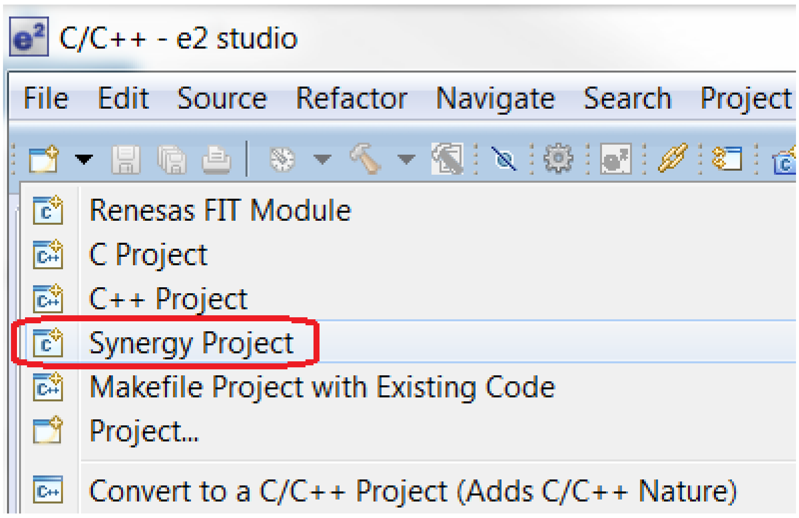
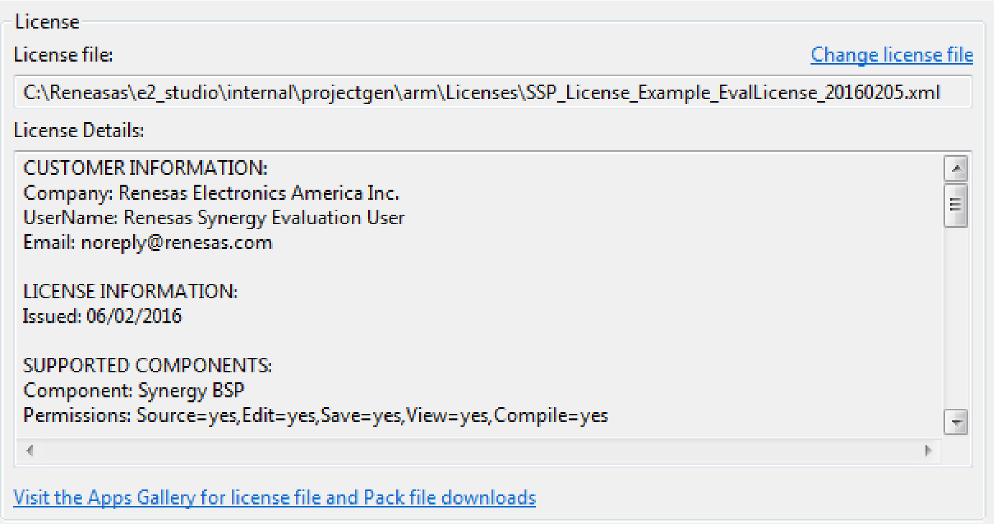

C/C++ - e2 studio
File Edit Source Refactor Navigate Search Project
Renesas FIT Module
C Project
C++ Project
Synergy Project
Makefile Project with Existing Code
Project...
Convert to a C/C++ Project (Adds C/C++ Nature)

Figure 4 Select Synergy Project in the Drop-Down Menu

9. If the License file is configured, the License area of the form will look like **Figure 5**. If it does, skip to step 10. If it is empty, continue with the below steps (A to G).



License

License file: Change license file

C:\Reneasas\e2_studio\internal\projectgen\arm\Licenses\SSP_License_Example_EvalLicense_20160205.xml

License Details:

CUSTOMER INFORMATION:
Company: Renesas Electronics America Inc.
UserName: Renesas Synergy Evaluation User
Email: noreply@renesas.com

LICENSE INFORMATION:
Issued: 06/02/2016

SUPPORTED COMPONENTS:
Component: Synergy BSP
Permissions: Source=yes,Edit=yes,Save=yes,View=yes,Compile=yes

Visit the Apps Gallery for license file and Pack file downloads

Figure 5 Configured License File

A. Click the Change license file button. e2 studio displays the "Preferences" dialog box:

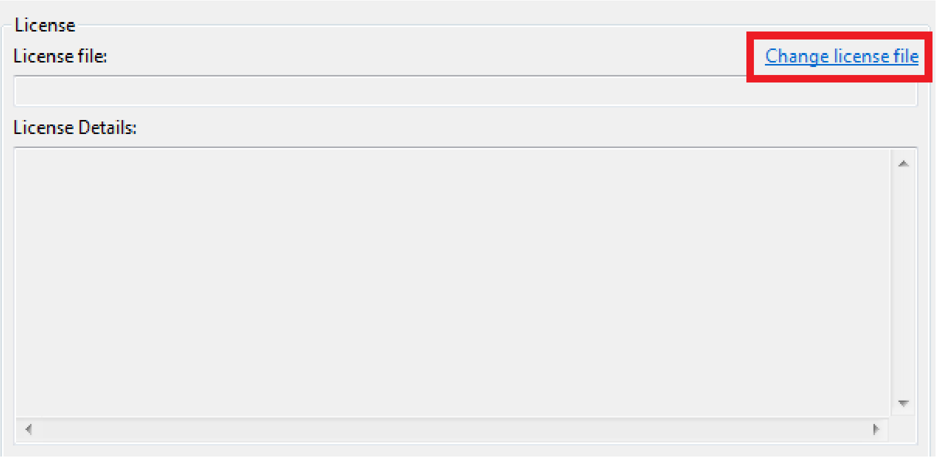


Figure 6 Unconfigured License File

B. Click the Browse … button. e² studio displays the Specify Synergy License Dialog box:

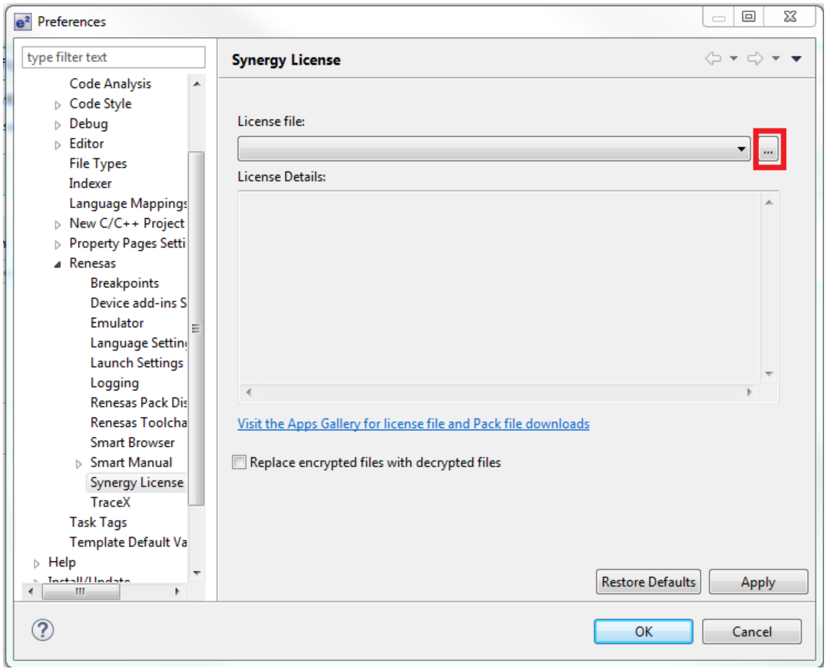


Figure 7 Preferences Dialog Box with Synergy License Configuration

C. Click the Browse … button. e² studio displays the Open Dialog box and should display the Licenses directory:

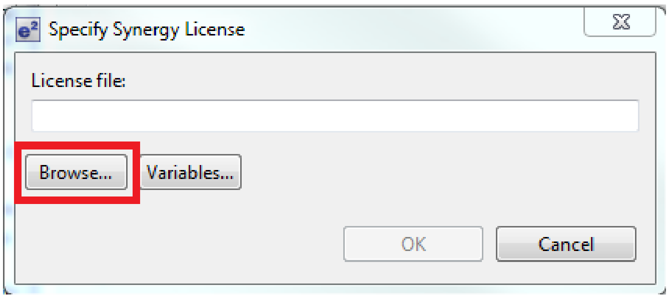


Figure 8 Synergy License Dialog box

Note: If you installed e^2 studio into the default location, the license file is located in the "***C:\Renesas\e2_studio\internal\projectgen\arm\Licenses***" directory.

D. Select the "SSP_License_Example_EvalLicense_*.xml" located in the directory.

E. Click Open to select the License file.

F. Click OK to set the license and close the dialog.

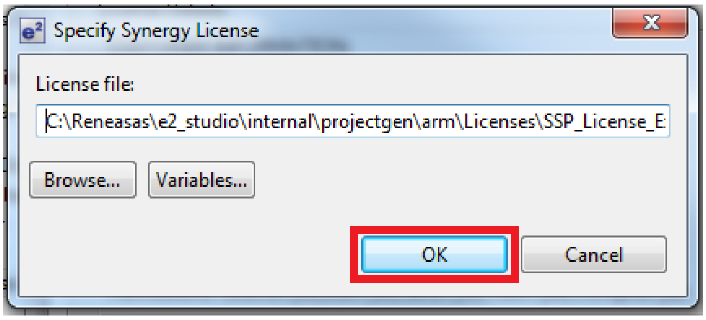
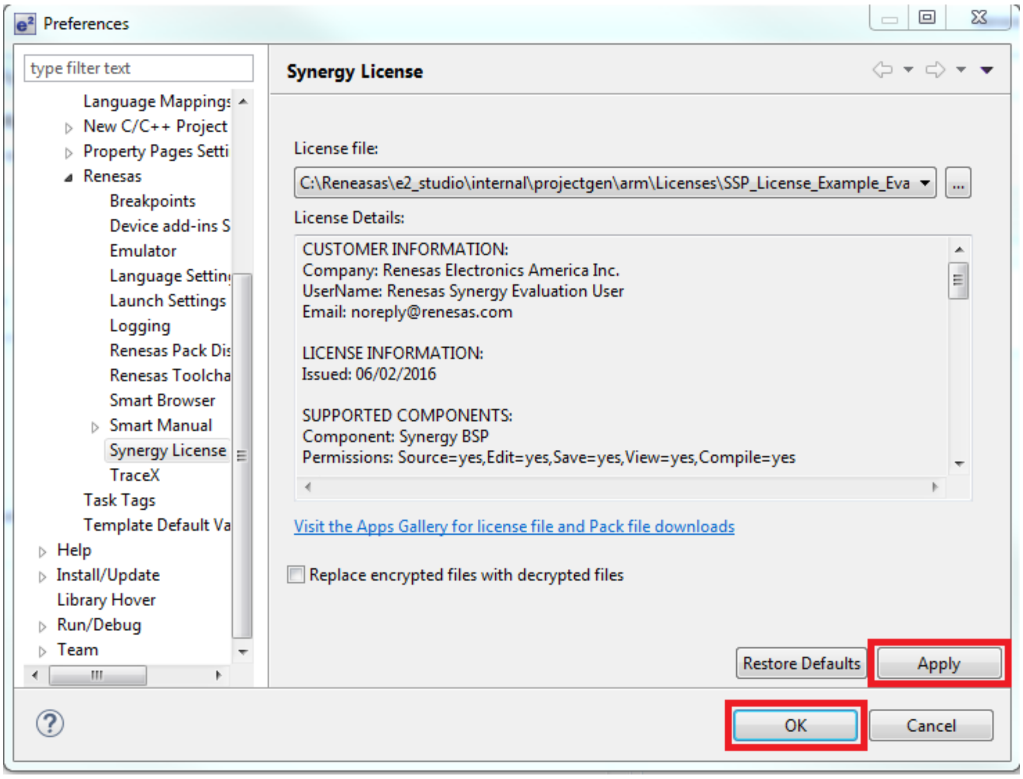

Specify Synergy License
License file:
C:\Reneasas\e2_studio\internal\projectgen\arm\Licenses\SSP_License_E
Browse...
Variables...
OK
Cancel

Figure 9 Confirm License File

G. Click Apply and then OK in the Preferences Dialog box.



CUSTOMER INFORMATION:
Company: Renesas Electronics America Inc.
UserName: Renesas Synergy Evaluation User
Email: noreply@renesas.com

LICENSE INFORMATION:
Issued: 06/02/2016

SUPPORTED COMPONENTS:
Component: Synergy BSP
Permissions: Source=yes,Edit=yes,Save=yes,View=yes,Compile=yes

Visit the Apps Gallery for license file and Pack file downloads
□ Replace encrypted files with decrypted files
Restore Defaults
Apply
OK
Cancel

Figure 10 Apply and Confirm Synergy License File Selection

10. Enter a name for the project in the Project name text field, for example "GUIApp".

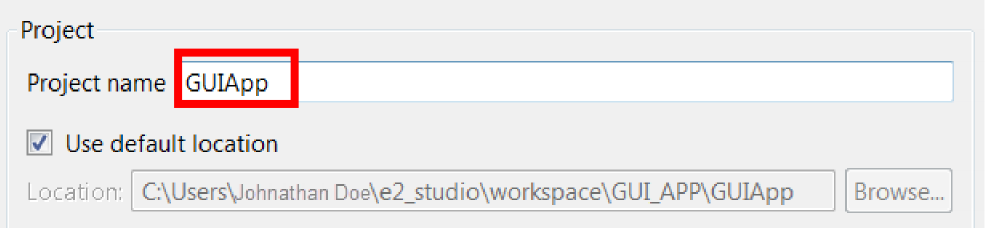
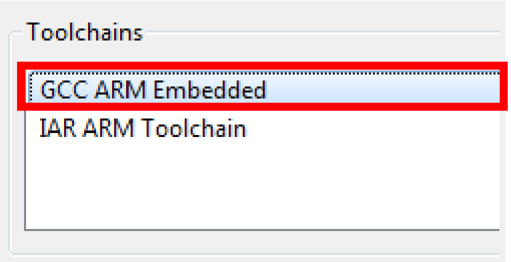

Project
Project name GUIApp
☑ Use default location
Location: C:\Users\Johnathan Doe\e2_studio\workspace\GUI_APP\GUIApp
Browse...

Figure 11 Enter a Project Name

11. On the top right of this page, verify that the Toolchain option is set to “GCC ARM Embedded”.



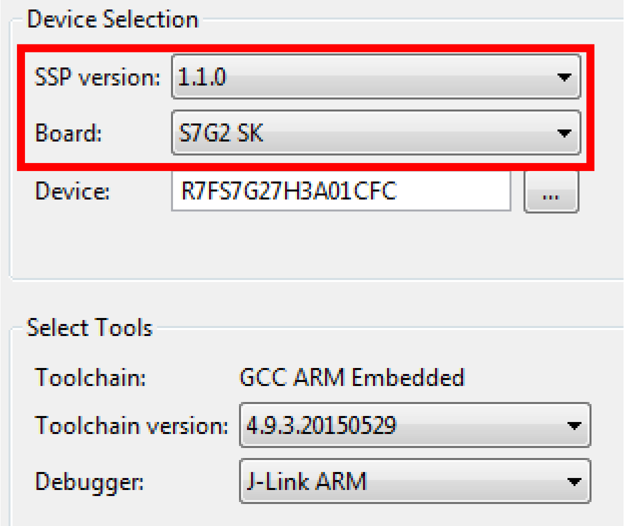
Toolchains

GCC ARM Embedded

IAR ARM Toolchain

Figure 12 Verify GCC ARM Embedded Toolchain

12. Click Next.
13. Under Device Selection (top left), select SSP version 1.1.0 (or later).
14. For Board, select “S7G2 SK”. The Device will be updated automatically.



Device Selection

SSP version: 1.1.0

Board: S7G2 SK

Device: R7FS7G27H3A01CFC ...

Select Tools

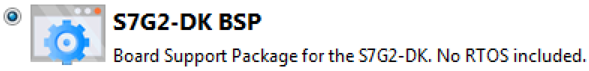
Toolchain: GCC ARM Embedded

Toolchain version: 4.9.3.20150529

Debugger: J-Link ARM

Figure 13 Device Selection

15. Click Next.
16. In the Project Configuration Dialog select the option “S7G2-SK BSP”.

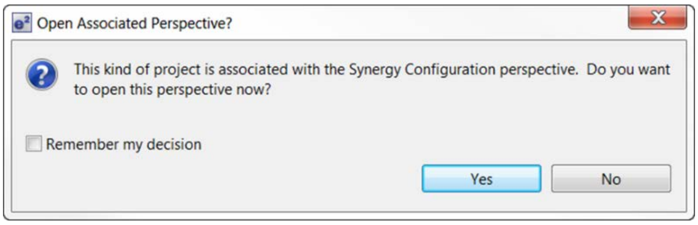


● **S7G2-DK BSP**

Board Support Package for the S7G2-DK. No RTOS included.

Figure 14 Select the S7G2-SK BSP

17. Click Finish.
18. If you have not directed e^2 studio to remember your perspectives, e^2 studio will display the “Open Associated Perspective” dialog box. If opened, click Yes to acknowledge and close.



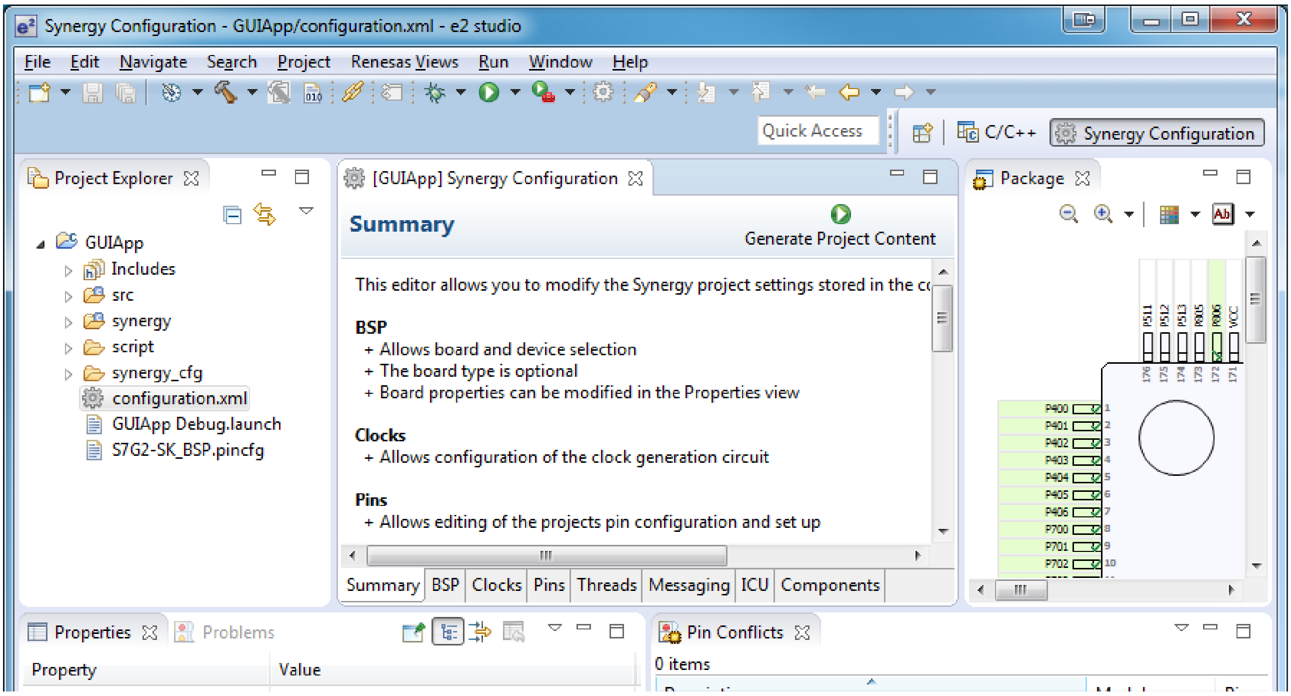
Open Associated Perspective?

This kind of project is associated with the Synergy Configuration perspective. Do you want to open this perspective now?

□ Remember my decision

Figure 15 Open Perspective Dialog Box

When the project finishes being created, you will see the following screen:



Summary

This editor allows you to modify the Synergy project settings stored in the co

BSP
+ Allows board and device selection
+ The board type is optional
+ Board properties can be modified in the Properties view

Clocks
+ Allows configuration of the clock generation circuit

Pins
+ Allows editing of the projects pin configuration and set up

Figure 16 GUIApp Project

4. Configure the Project in the ISDE

Now that the project has been successfully created in section Creating the Project in e^2 studio ISDE (Integrated Solution Development Environment) you can start configuring the project for the GUI application.

1. Open the Synergy Configuration, if not already open, by double-clicking the “configuration.xml” file in the Project Explorer Window.

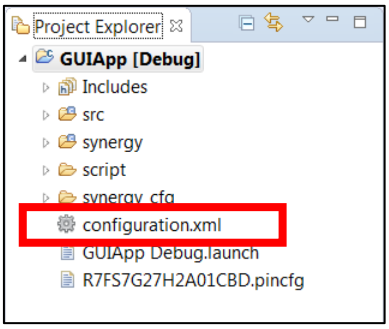


Figure 17 Selecting the configuration.xml file in Project Explorer

2. Click the BSP tab of the Synergy Configuration Window.

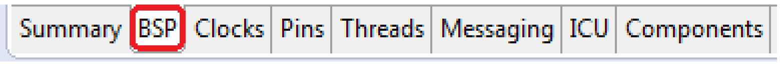
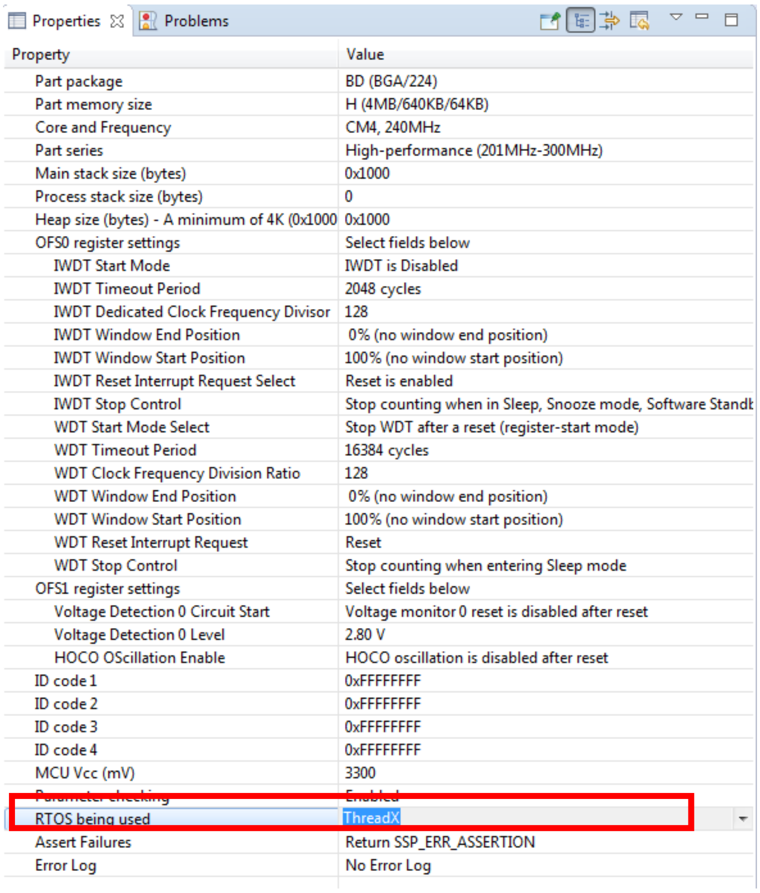


Figure 18 Synergy Configuration BSP Tab

3. Now look in the Properties Window (bottom left window) and set the “RTOS being used” to “ThreadX”. (Helpful hint: After you set ThreadX, click another area of the properties and verify that ThreadX is still set.)



Property	Value
Part package	BD (BGA/224)
Part memory size	H (4MB/640KB/64KB)
Core and Frequency	CM4, 240MHz
Part series	High-performance (201MHz-300MHz)
Main stack size (bytes)	0x1000
Process stack size (bytes)	0
Heap size (bytes) - A minimum of 4K (0x1000	0x1000
OFS0 register settings	Select fields below
IWDT Start Mode	IWDT is Disabled
IWDT Timeout Period	2048 cycles
IWDT Dedicated Clock Frequency Divisor	128
IWDT Window End Position	0% (no window end position)
IWDT Window Start Position	100% (no window start position)
IWDT Reset Interrupt Request Select	Reset is enabled
IWDT Stop Control	Stop counting when in Sleep, Snooze mode, Software Standb
WDT Start Mode Select	Stop WDT after a reset (register-start mode)
WDT Timeout Period	16384 cycles
WDT Clock Frequency Division Ratio	128
WDT Window End Position	0% (no window end position)
WDT Window Start Position	100% (no window start position)
WDT Reset Interrupt Request	Reset
WDT Stop Control	Stop counting when entering Sleep mode
OFS1 register settings	Select fields below
Voltage Detection 0 Circuit Start	Voltage monitor 0 reset is disabled after reset
Voltage Detection 0 Level	2.80 V
HOCO OScillation Enable	HOCO oscillation is disabled after reset
ID code 1	0xFFFFFFFF
ID code 2	0xFFFFFFFF
ID code 3	0xFFFFFFFF
ID code 4	0xFFFFFFFF
MCU Vcc (mV)	3300
Parameter checking	Enabled
RTOS being used	ThreadX
Assert Failures	Return SSP_ERR_ASSERTION
Error Log	No Error Log

Figure 19 Setting the BSP to use the ThreadX RTOS

4. Back in the Synergy Configuration Window, click the Threads tab.

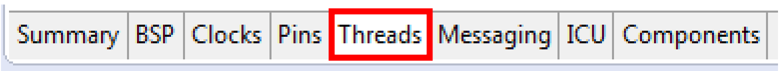
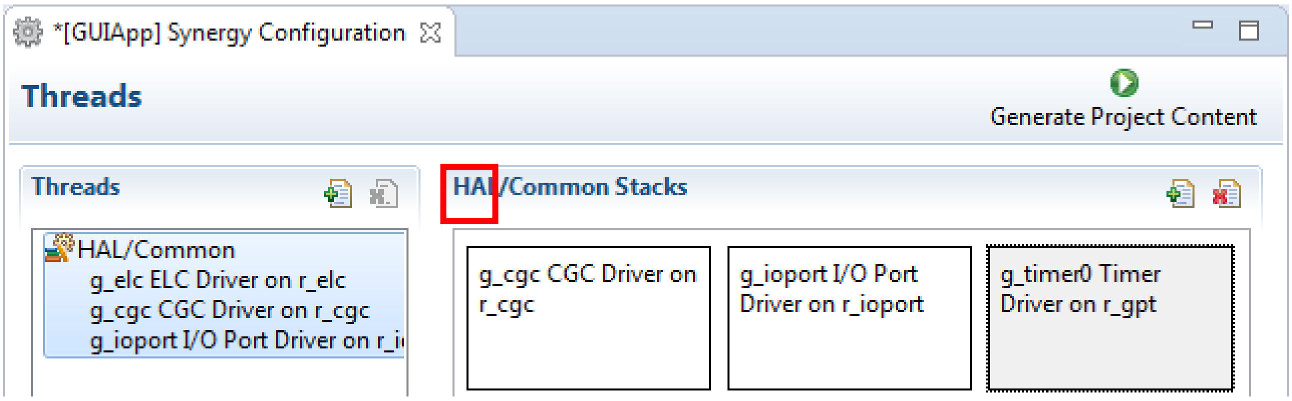


Figure 20 Synergy Configuration Threads Tab

5. Create a new thread by clicking New in the Threads area.



HAL/Common
g_elc ELC Driver on r_elc
g_cgc CGC Driver on r_cgc
g_ioport I/O Port Driver on r_i

g_cgc CGC Driver on r_cgc

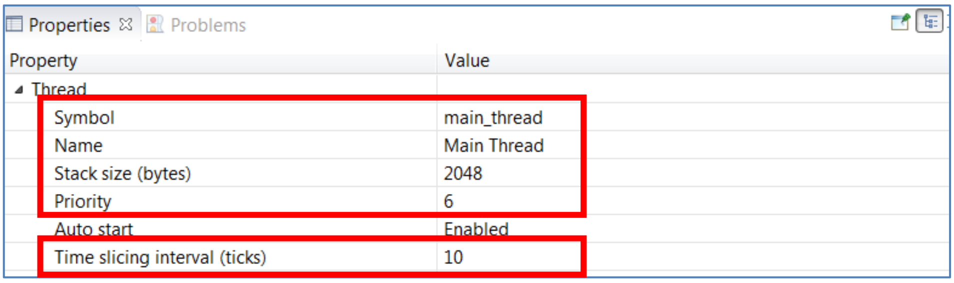
g_ioport I/O Port Driver on r_ioport

g_timer0 Timer Driver on r_gpt

Figure 21 Create a New Thread

6. Click “New Thread” to display the properties.

7. Edit the Properties to match the following:



Property	Value
Thread	
Symbol	main_thread
Name	Main Thread
Stack size (bytes)	2048
Priority	6
Auto start	Enabled
Time slicing interval (ticks)	10

Figure 22 Configure Main Thread Properties

8. Back in the Synergy Configuration Window, Threads tab, Main Thread Stacks area, click New.

Note: Be sure Main Thread is selected before adding new modules.

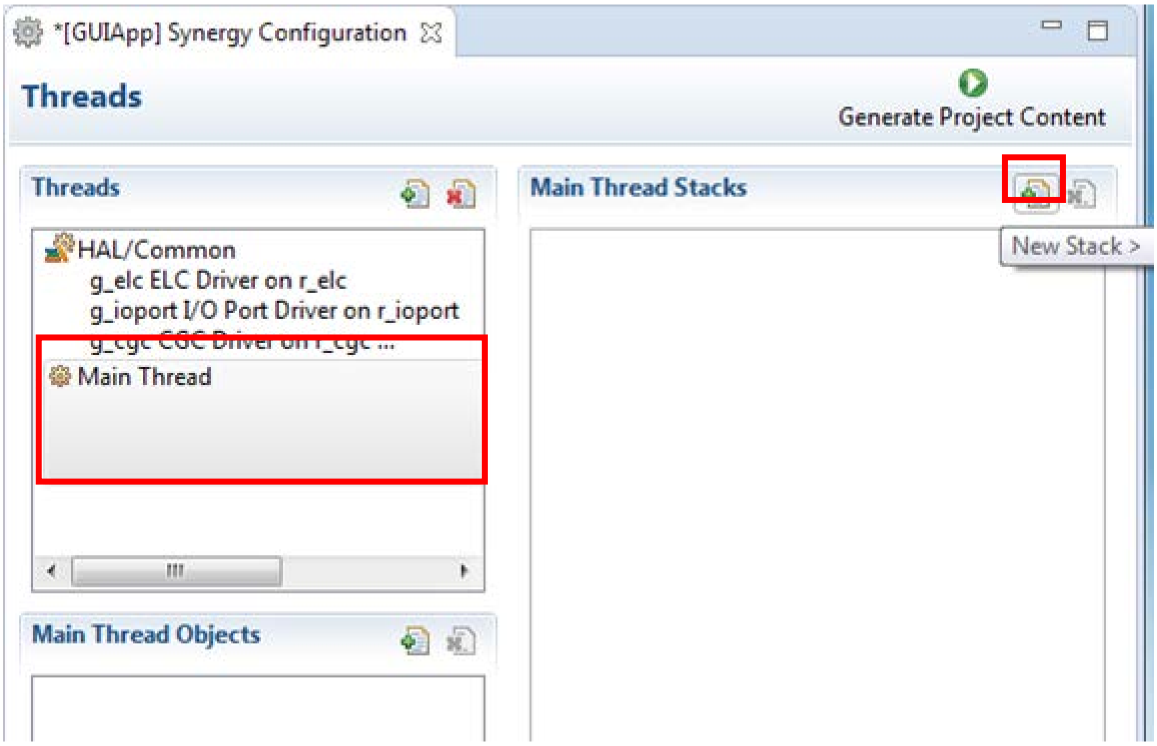

*[GUIApp] Synergy Configuration
Threads
Generate Project Content
Threads
Main Thread Stacks
New Stack >
HAL/Common
g_elc ELC Driver on r_elc
g_ioport I/O Port Driver on r_ioport
g_cgc CGC Driver on r_cgc ...
Main Thread
Main Thread Objects

Figure 23 Main Thread Stacks

9. Add a framework for the touch panel by selecting New “Framework > Input > Touch Panel Framework on sf_touch_panel_i2c”.

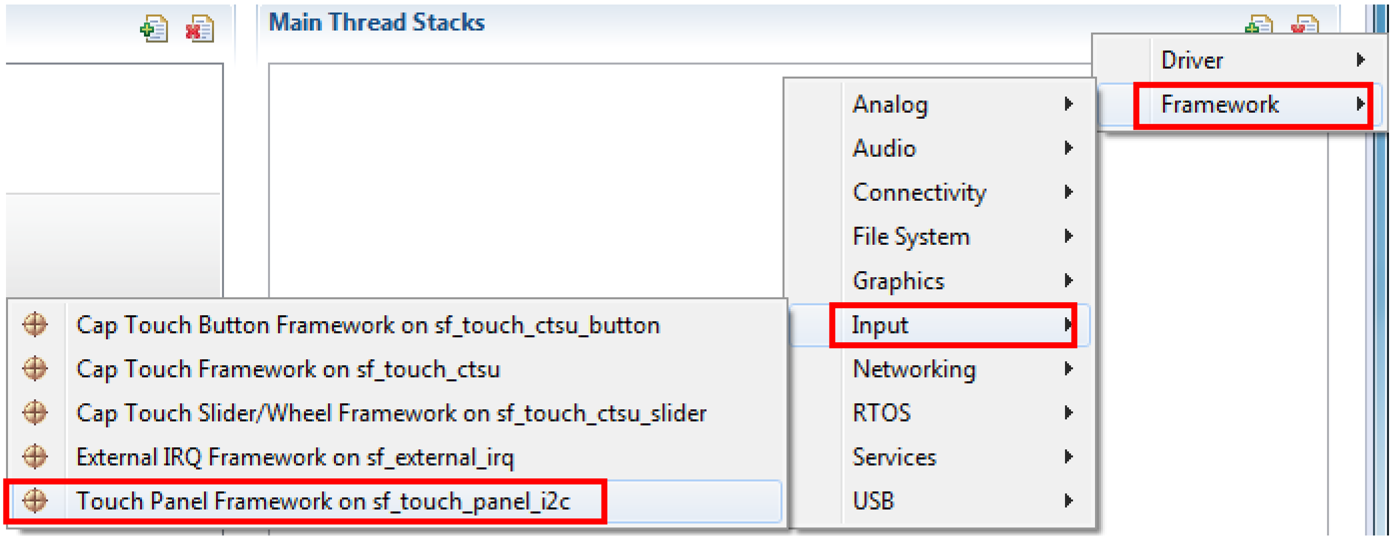
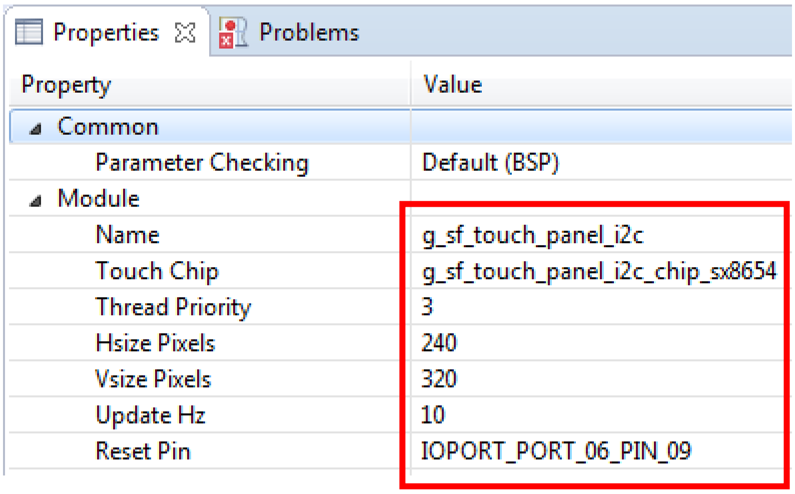

Main Thread Stacks
Driver
Framework
Analog
Audio
Connectivity
File System
Graphics
Input
Networking
RTOS
Services
USB
Cap Touch Button Framework on sf_touch_ctsu_button
Cap Touch Framework on sf_touch_ctsu
Cap Touch Slider/Wheel Framework on sf_touch_ctsu_slider
External IRQ Framework on sf_external_irq
Touch Panel Framework on sf_touch_panel_i2c

Figure 24 Adding Touch Panel Framework

10. Configure the properties as follows.



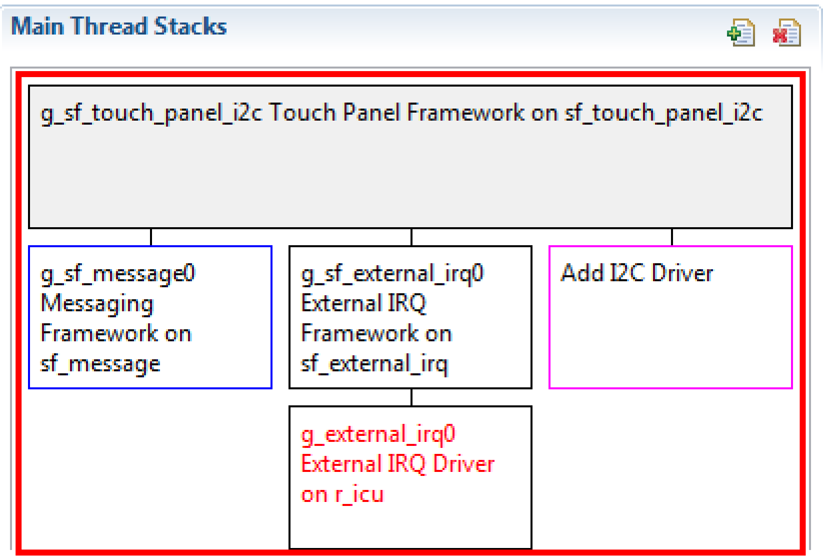
Property	Value
Common	
Parameter Checking	Default (BSP)
Module	
Name	g_sf_touch_panel_i2c
Touch Chip	g_sf_touch_panel_i2c_chip_sx8654
Thread Priority	3
Hsize Pixels	240
Vsize Pixels	320
Update Hz	10
Reset Pin	IOPORT_PORT_06_PIN_09

Figure 25 Configure Touch Panel Properties

11. Notice that the Synergy Configurator has already created the message framework, external IRQ framework, and has a placeholder for the external IRQ and I2C driver stacks as shown below:

The messaging framework is used by other framework layers and tasks to pass messages around the system. This system will be used to pass data from the touch screen driver to the Main Task to handle touch inputs.

The SF External Interrupt is a framework layer used by the touch controller driver as shown below:



g_sf_touch_panel_i2c Touch Panel Framework on sf_touch_panel_i2c

g_sf_message0 Messaging Framework on sf_message

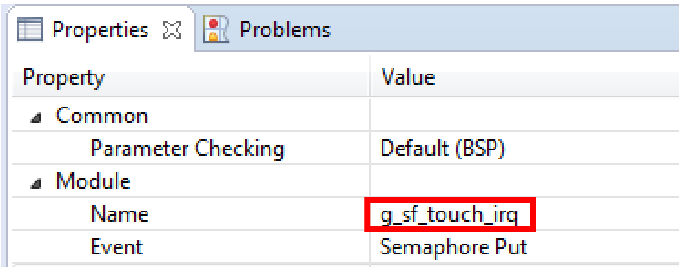
g_sf_external_irq0 External IRQ Framework on sf_external_irq

Add I2C Driver

g_external_irq0 External IRQ Driver on r_icu

Figure 26 Touch Panel Framework Stack

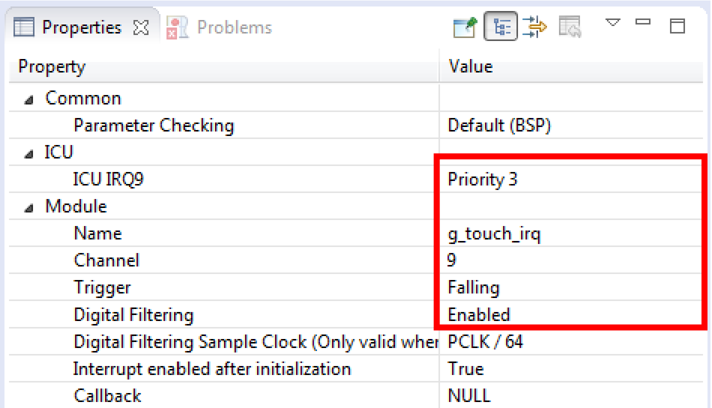
12. Select the “External IRQ Framework on sf_external_irq” and configure the properties as follows:



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

Figure 27 Configuring External Interrupts Properties

13. Select “External IRQ Driver on r_icu”. Configure the properties for the new module as follows. **Hint: Change the “Channel” first.**



Property	Value
◢ Common	
Parameter Checking	Default (BSP)
◢ ICU	
ICU IRQ9	Priority 3
◢ Module	
Name	g_touch_irq
Channel	9
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock (Only valid wher	PCLK / 64
Interrupt enabled after initialization	True
Callback	NULL

Figure 28 Touch screen IRQ properties

14. In the Synergy Configuration Window, Threads tab, Main Thread Stacks area, add a driver for the I2C bus by right-clicking the “Add I2C Driver” then selecting ”New -> I2C Driver on r_iic”.

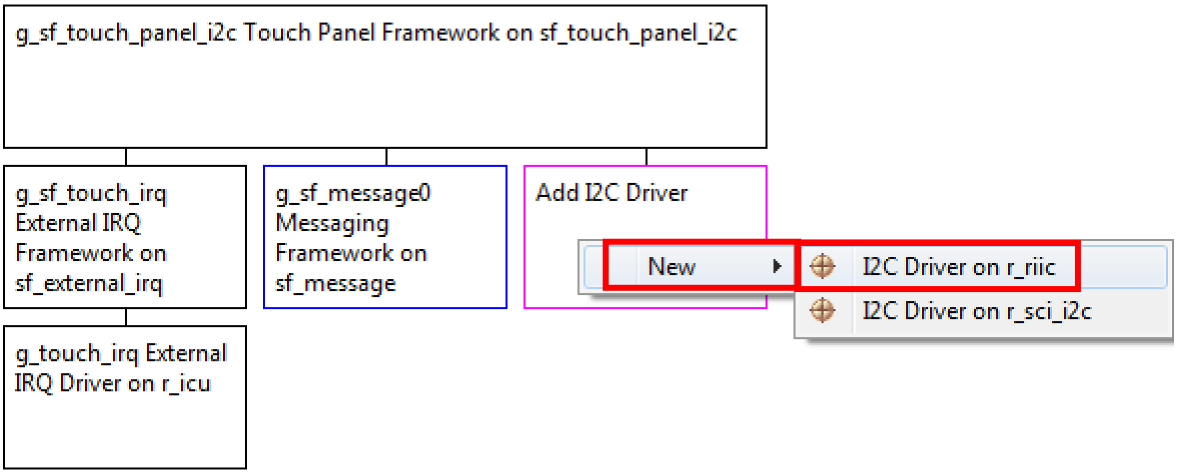
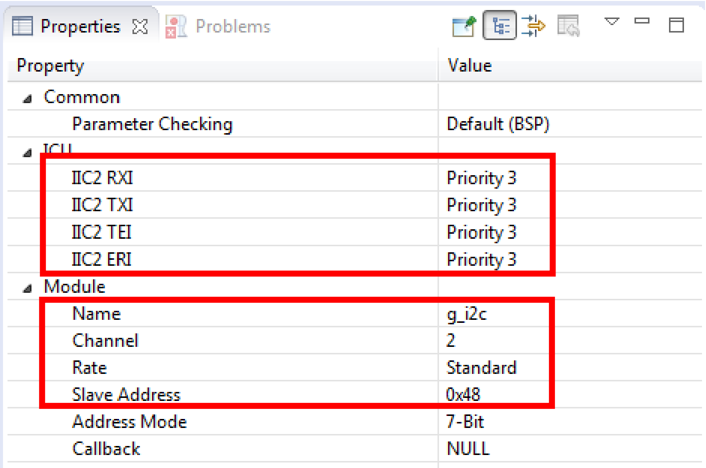

g_sf_touch_panel_i2c Touch Panel Framework on sf_touch_panel_i2c
g_sf_touch_irq External IRQ Framework on sf_external_irq
g_sf_message0 Messaging Framework on sf_message
Add I2C Driver
New
I2C Driver on r_riic
I2C Driver on r_sci_i2c
g_touch_irq External IRQ Driver on r_icu

Figure 29 Adding I2C Driver

15. Configure the properties for the "I2C Driver on RIIC" as follows. **Hint: Change the "Channel" option first.**



Property	Value
◢ Common	
Parameter Checking	Default (BSP)
◢ ICU	
IIC2 RXI	Priority 3
IIC2 TXI	Priority 3
IIC2 TEI	Priority 3
IIC2 ERI	Priority 3
◢ Module	
Name	g_i2c
Channel	2
Rate	Standard
Slave Address	0x48
Address Mode	7-Bit
Callback	NULL

Figure 30 Configuring I2C Driver

16. Under Main Thread Stacks, select New "Framework > Graphics > GUIX on gx".

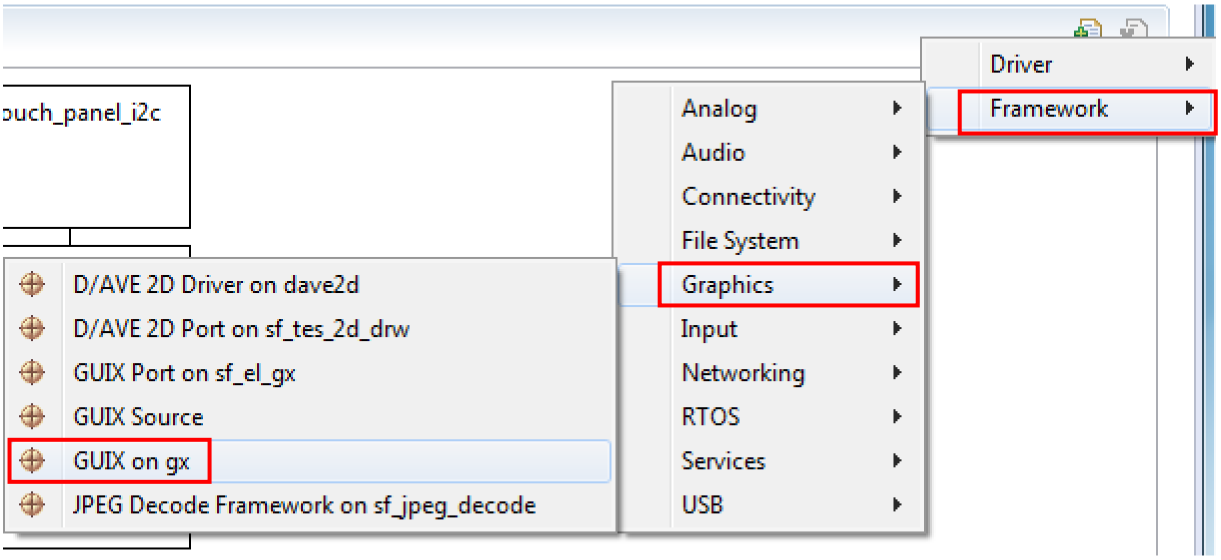


Figure 31 Adding Framework for GUIX on gx

Notice that the Synergy Configurator has already created the “GUIX Port on sf_el_gx” framework, Display Driver, and has a placeholder for the JPEG decode and D/AVE hardware accelerator stacks as shown in the following figure:

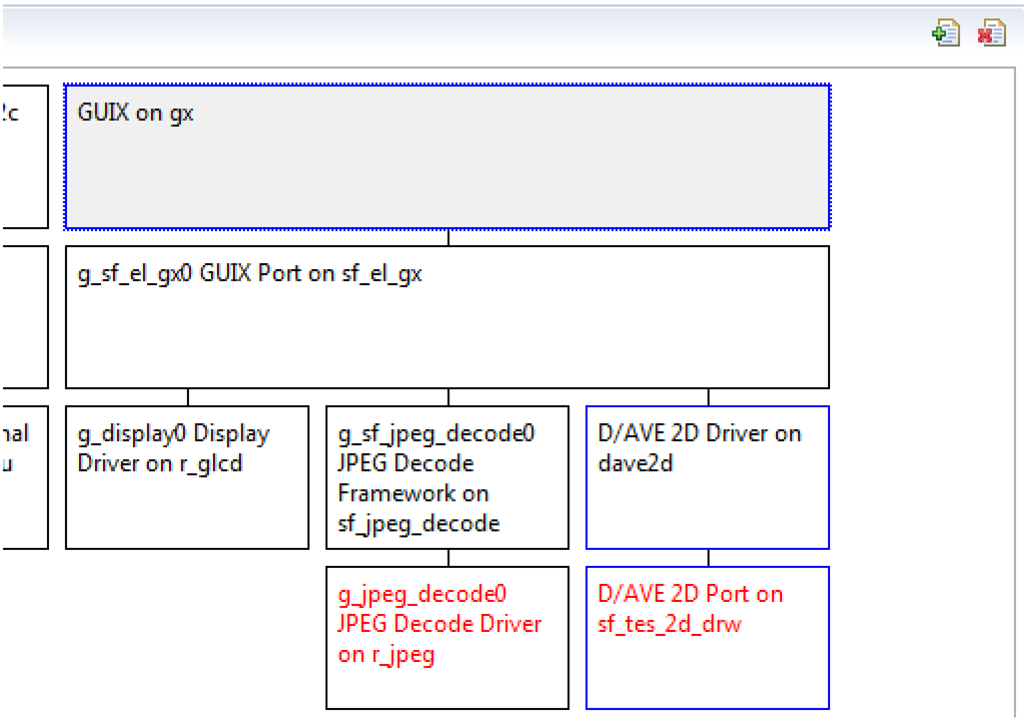
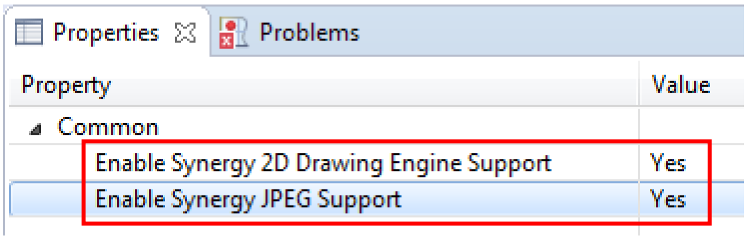

GUIX on gx
g_sf_el_gx0 GUIX Port on sf_el_gx
g_display0 Display Driver on r_glcd
g_sf_jpeg_decode0 JPEG Decode Framework on sf_jpeg_decode
D/AVE 2D Driver on dave2d
g_jpeg_decode0 JPEG Decode Driver on r_jpeg
D/AVE 2D Port on sf_tes_2d_drw

Figure 32 GUIX on gx Stack

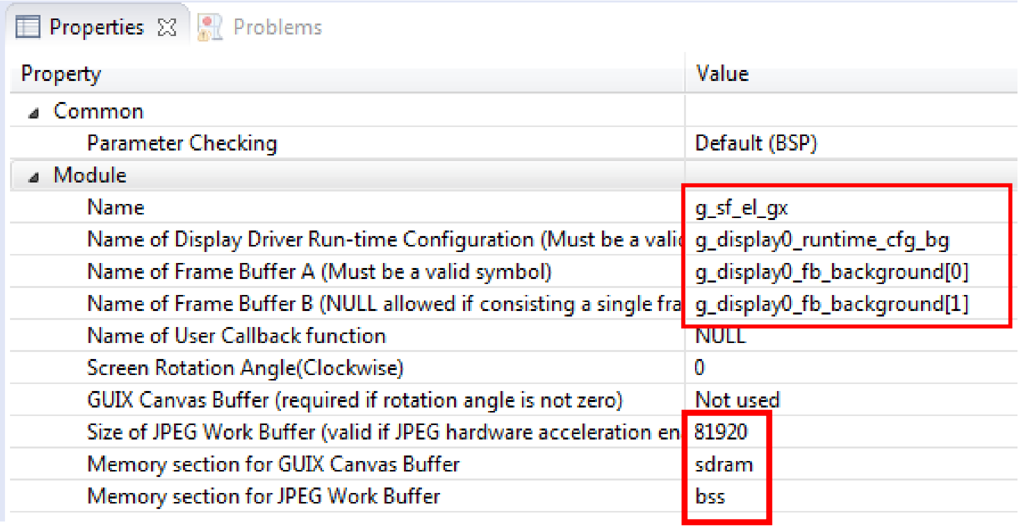
17. Select “GUIX on gx” and configure the Properties as follows:



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

Figure 33 GUIX on gx Properties

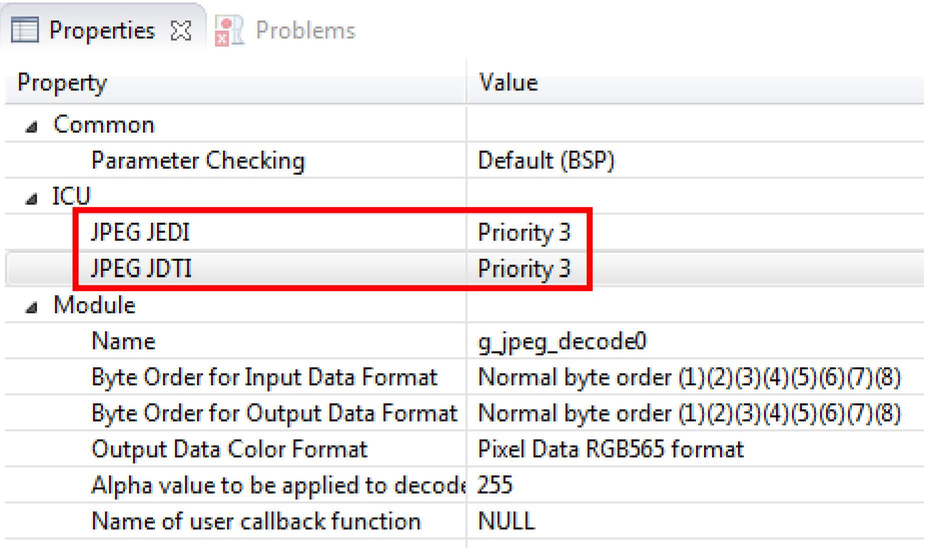
18. Select “GUIX Port on sf_el_gx” and configure the Properties as follows:



Property	Value
Common	
Parameter Checking	Default (BSP)
Module	
Name	g_sf_el_gx
Name of Display Driver Run-time Configuration (Must be a vali	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 fra	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 en	81920
Memory section for GUIX Canvas Buffer	sdram
Memory section for JPEG Work Buffer	bss

Figure 34 Configure GUIX Port property

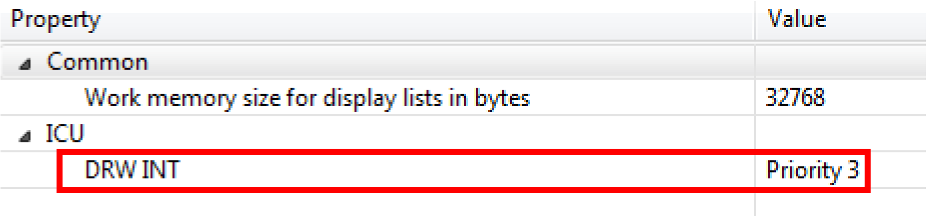
19. Select the “JPEG Decode Driver on r_jpeg” and configure the interrupt properties as follows. Note that Priority 3 is just an arbitrary number:



Property	Value
◢ Common	
Parameter Checking	Default (BSP)
◢ ICU	
JPEG JEDI	Priority 3
JPEG JDTI	Priority 3
◢ Module	
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 decode	255
Name of user callback function	NULL

Figure 35 JPEG Decode Driver on r_jpeg Properties

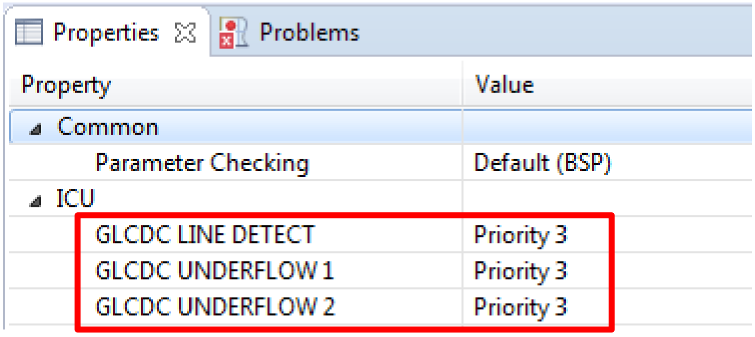
20. Under Main Thread Stacks, select “D/AVE 2D Port on sf_tes_2d_drw” and configure the properties as follows:



Property	Value
◢ Common	
Work memory size for display lists in bytes	32768
◢ ICU	
DRW INT	Priority 3

Figure 36 D/AVE 2D Port Properties

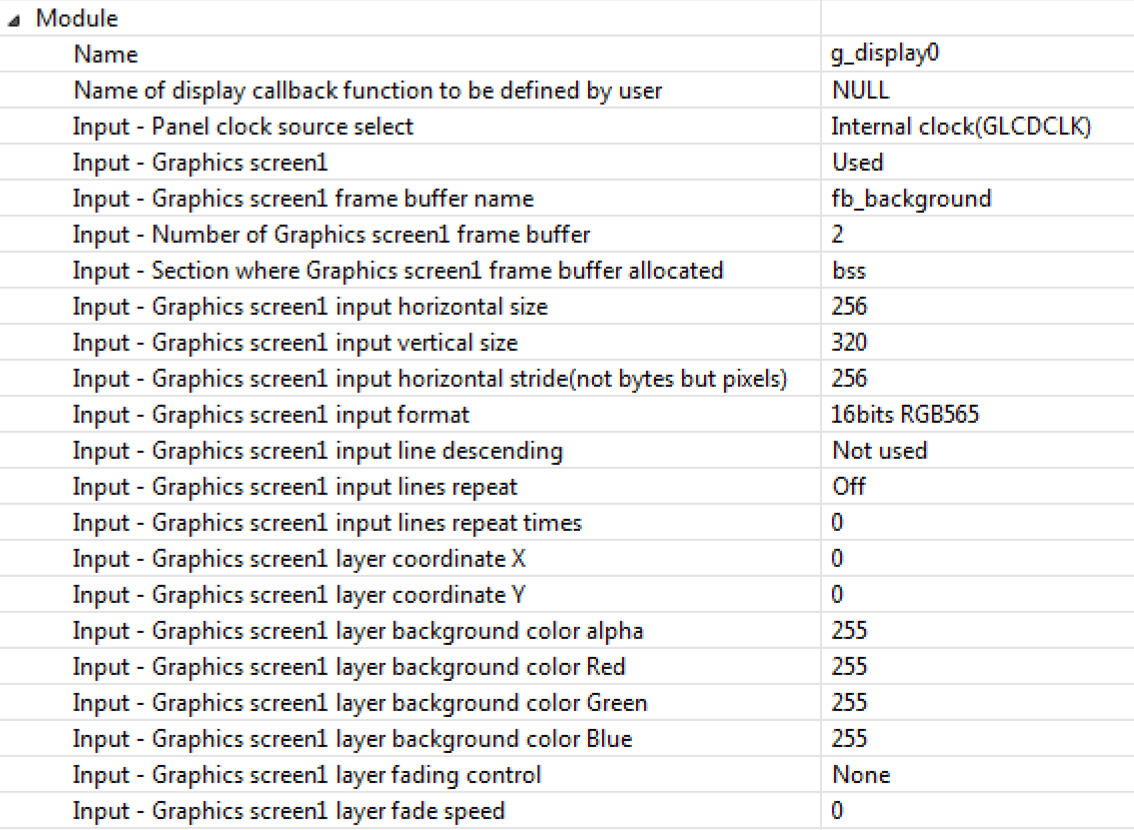
21. Under Main Thread Stacks, Select “Display Driver on r_glcd” and configure the Interrupt Properties as follows:



Property	Value
◢ Common	
Parameter Checking	Default (BSP)
◢ ICU	
GLCDC LINE DETECT	Priority 3
GLCDC UNDERFLOW 1	Priority 3
GLCDC UNDERFLOW 2	Priority 3

Figure 37 Interrupt Properties

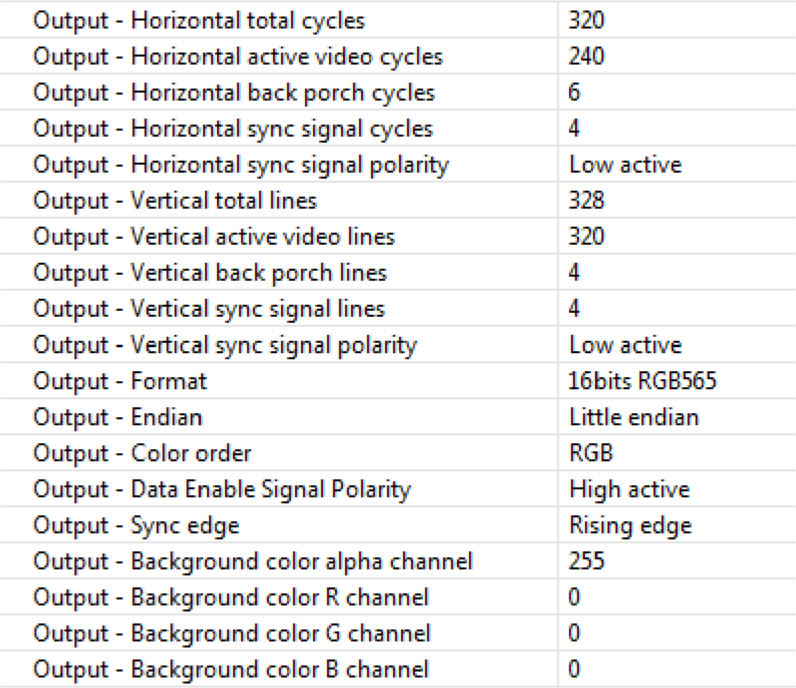
22. Configure the Graphics Screen 1 properties as follows:



◢ 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

Figure 38 Graphics Screen 1 Properties

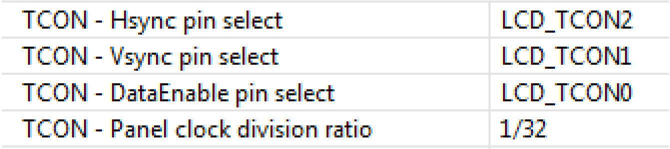
23. Configure the Output properties as follows:



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

Figure 39 Output Screen 2 Properties

24. Configure the TCON pins and clock as follows:



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

Figure 40 TCON Settings

25. Under Main Thread Stacks, select New > Driver > Connectivity > SPI Driver on r_sci_spi.

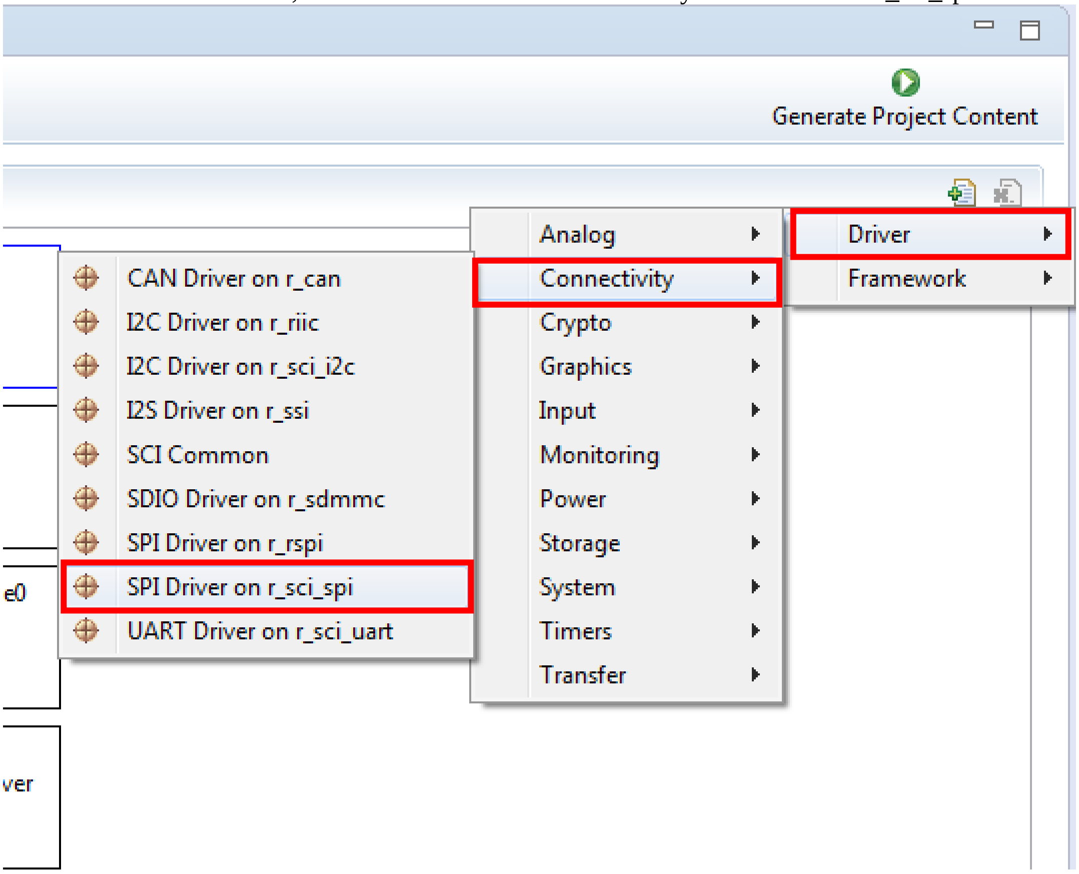
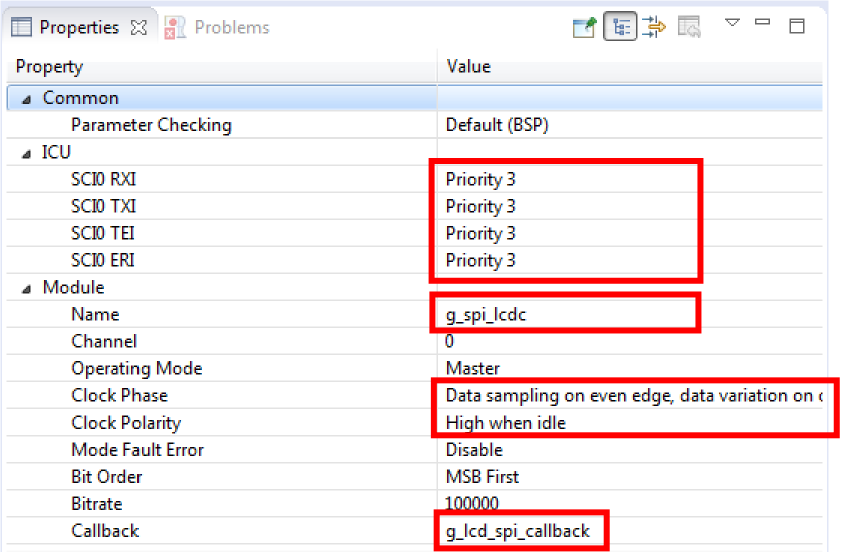


Figure 41 Adding Simple SPI (on SCI) Driver

26. Configure the properties as follows:

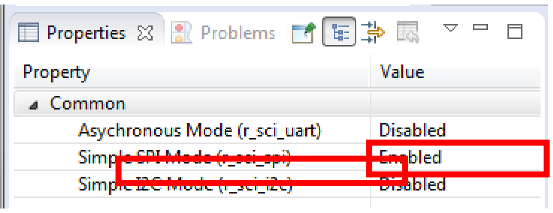


Property	Value
◢ Common	
Parameter Checking	Default (BSP)
◢ ICU	
SCI0 RXI	Priority 3
SCI0 TXI	Priority 3
SCI0 TEI	Priority 3
SCI0 ERI	Priority 3
◢ Module	
Name	g_spi_lcdc
Channel	0
Operating Mode	Master
Clock Phase	Data sampling on even edge, data variation on
Clock Polarity	High when idle
Mode Fault Error	Disable
Bit Order	MSB First
Bitrate	100000
Callback	g_lcd_spi_callback

Figure 42 Configure Simple SPI (on SCI) property

27. Select “SCI Common” on the “r_sci_spi” stack.

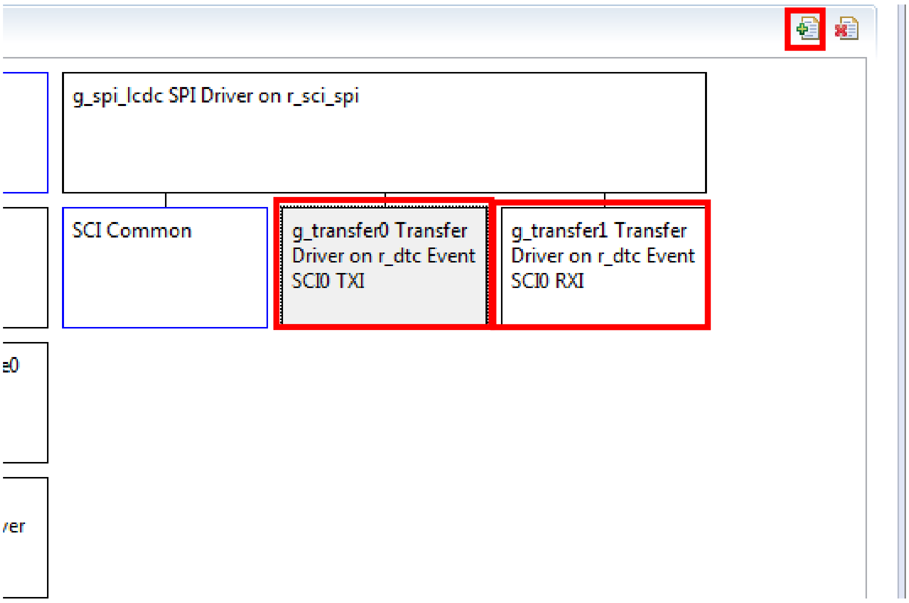
28. Enable “Simple SPI Mode (r_sci_spi)” in the properties as follows:



Property	Value
Common	
Asychronous Mode (r_sci_uart)	Disabled
Simple SPI Mode (r_sci_spi)	Enabled
Simple I2C Mode (r_sci_i2c)	Disabled

Figure 43 Enable Simple SPI in SCI Common property

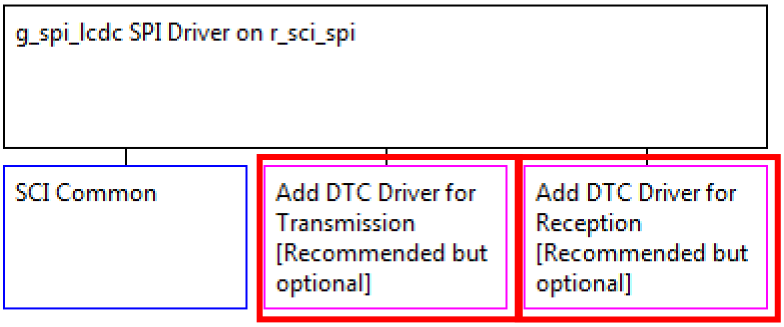
29. Click each “g_transfer” drive and remove it by clicking Remove since it is not needed for the LCD.



g_spi_lcdc SPI Driver on r_sci_spi
SCI Common
g_transfer0 Transfer Driver on r_dtc Event SCI0 TXI
g_transfer1 Transfer Driver on r_dtc Event SCI0 RXI
=0
ver

Figure 44 Remove Transfer Drivers

30. After removing the drivers, the placeholders for adding drivers will remain as shown in Figure 45 below:



g_spi_lcdc SPI Driver on r_sci_spi
SCI Common
Add DTC Driver for Transmission [Recommended but optional]
Add DTC Driver for Reception [Recommended but optional]

Figure 45 Transfer Drivers Placeholders

31. In the Synergy Configuration Window, Threads Tab, make sure the Main Thread is still selected.

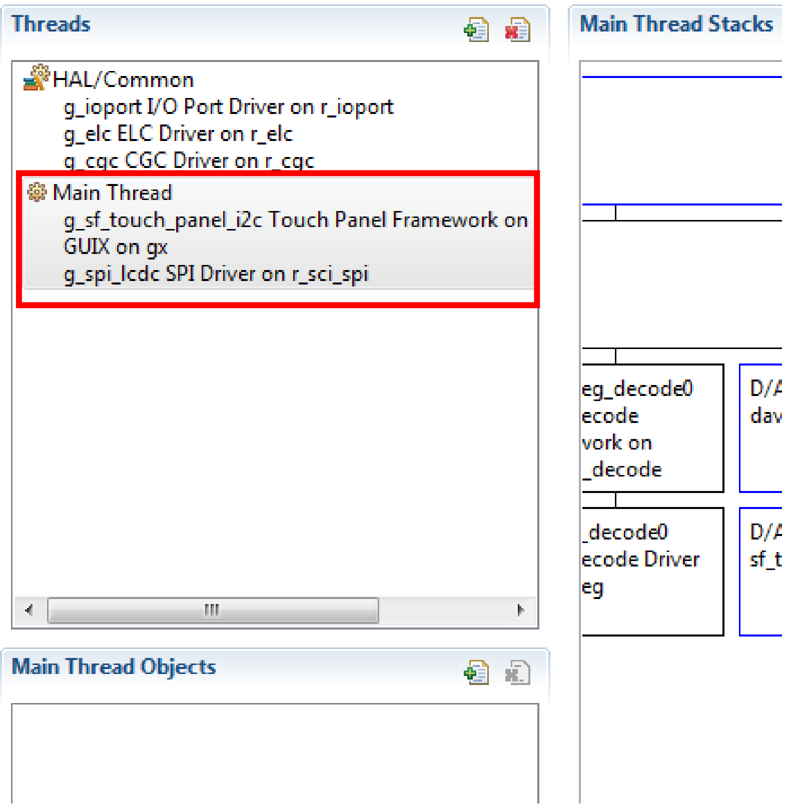

Threads
HAL/Common
g_ioport I/O Port Driver on r_ioport
g_elc ELC Driver on r_elc
g_cgc CGC Driver on r_cgc
Main Thread
g_sf_touch_panel_i2c Touch Panel Framework on
GUIX on gx
g_spi_lcdc SPI Driver on r_sci_spi
Main Thread Objects
Main Thread Stacks
eg_decode0
ecode
vork on
_decode
_decode0
ecode Driver
eg
D/A
dav
D/A
sf_t

Figure 46 Click on “Main Thread”

32. Under the Main Thread Objects, click New > Semaphore.

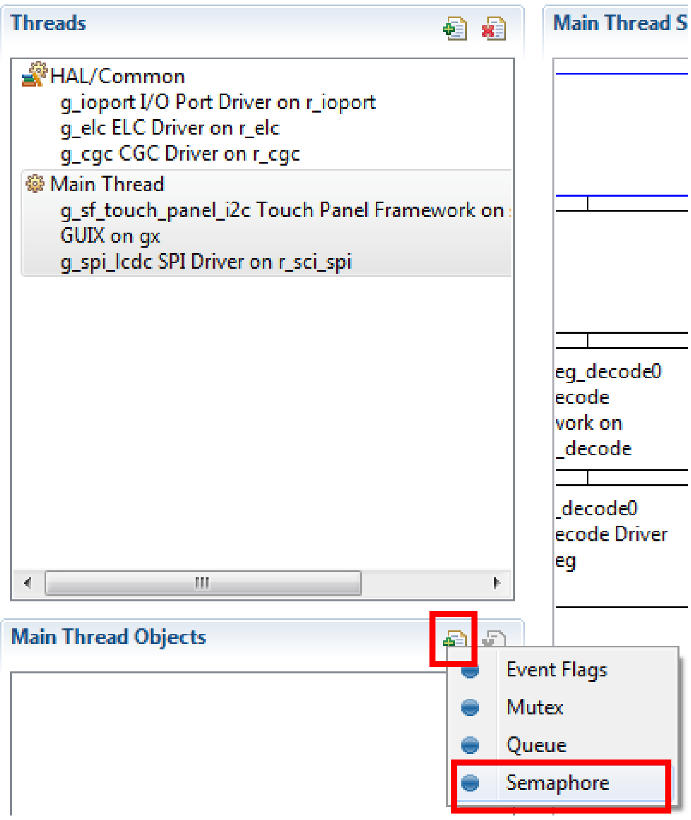
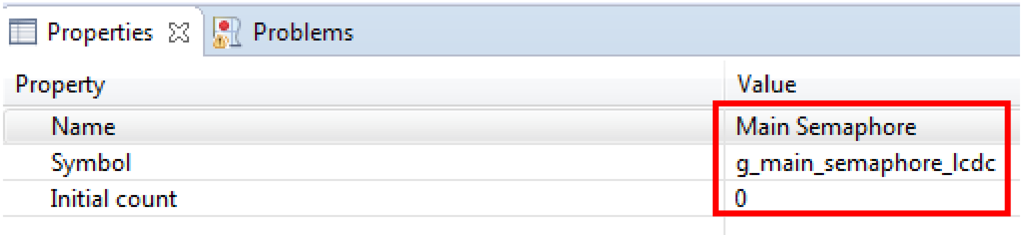

Threads
HAL/Common
g_ioport I/O Port Driver on r_ioport
g_elc ELC Driver on r_elc
g_cgc CGC Driver on r_cgc
Main Thread
g_sf_touch_panel_i2c Touch Panel Framework on
GUIX on gx
g_spi_lcdc SPI Driver on r_sci_spi
Main Thread Objects
Main Thread S
eg_decode0
ecode
vork on
_decode
_decode0
ecode Driver
eg
Event Flags
Mutex
Queue
Semaphore

Figure 47 Add a Semaphore

33. Configure the properties as follows:



Property	Value
Name	Main Semaphore
Symbol	g_main_semaphore_lcdc
Initial count	0

Figure 48 Configure Semaphore

34. In the Synergy Configuration Window, select the Pins tab.

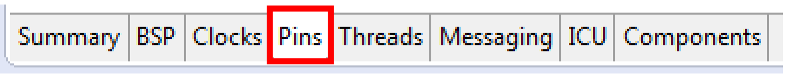
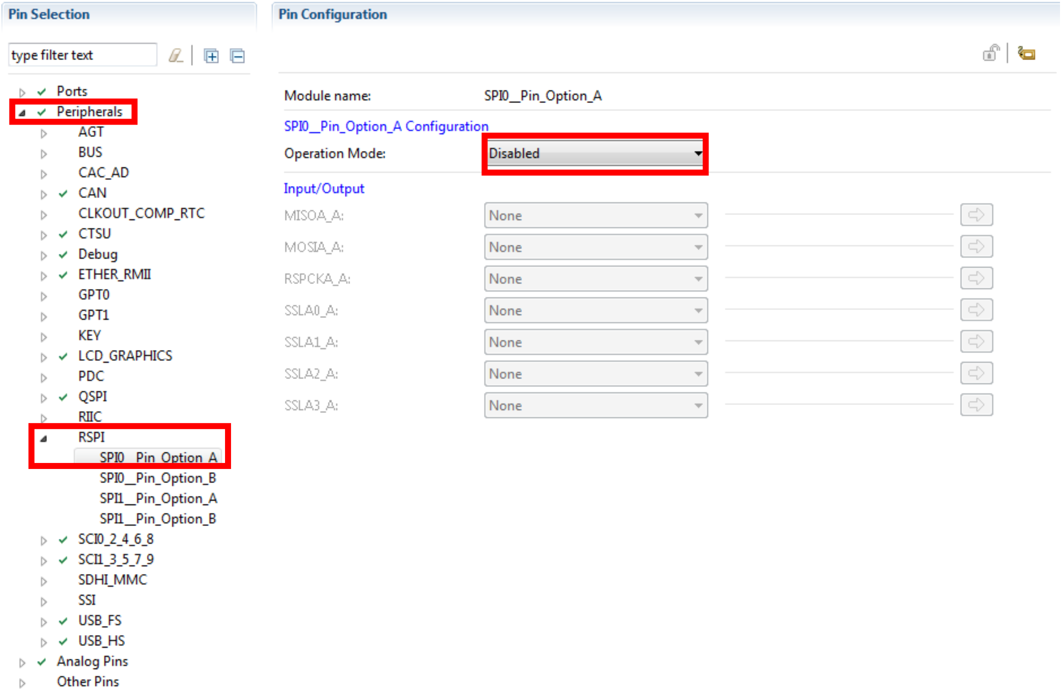


Figure 49 Configuration Pins

35. Select Peripherals > RSPI > SPI0_Pin_Option_A in Pin Selection, and change “Operation Mode” from “SPI Op” to “Disabled” in Pin Configuration of SPI0_Pin_Option_A module. This must be disabled to free the pins it shares with the SCI module.



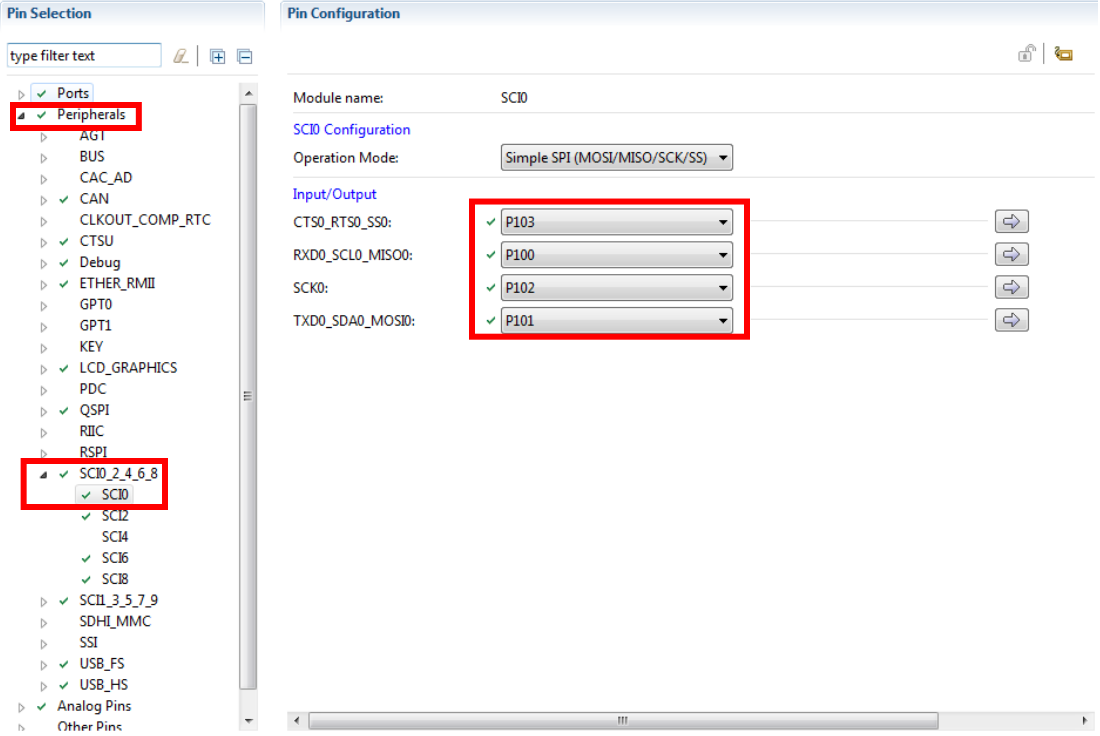
Module name: SPI0_Pin_Option_A

Operation Mode: Disabled

MISOA_A:	None
MOSIA_A:	None
RSPCKA_A:	None
SSLA0_A:	None
SSLA1_A:	None
SSLA2_A:	None
SSLA3_A:	None

Figure 50 Disable SPI0_Pin_Option_A in Pin Configuration

36. Select Peripherals > SCI0_2_4_6_8 > SCI0 in Pin Selection, and configure in Pin Configuration of SCI0 module as follows:



Module name: SCI0

SCI0 Configuration

Operation Mode: Simple SPI (MOSI/MISO/SCK/SS)

Input/Output

CTS0_RTS0_SS0: ✓ P103

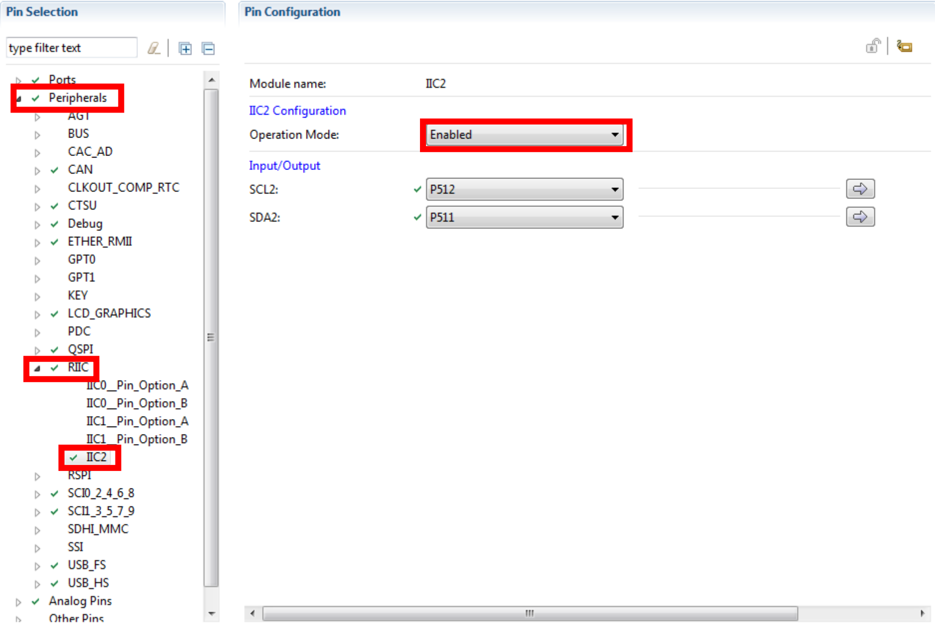
RXD0_SCL0_MISO0: ✓ P100

SCK0: ✓ P102

TXD0_SDA0_MOSI0: ✓ P101

Figure 51 Configure SCI0 Pin Configuration

37. Select Peripherals > RIIC > IIC2 in Pin Selection, and enable IIC2 module in Pin Configuration as follows:



Module name: IIC2

IIC2 Configuration

Operation Mode: Enabled

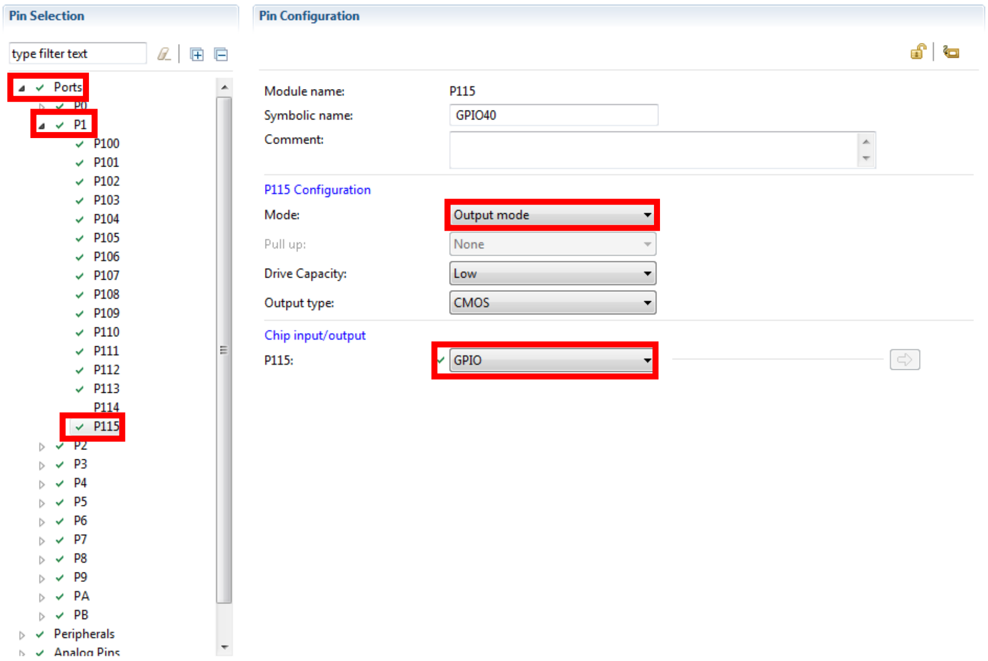
Input/Output

SCL2: ✓ P512

SDA2: ✓ P511

Figure 52 Configure IIC2 Pin Configuration

38. Select Ports > P1 > P115 in Pin Selection, and configure GPIO in Pin Configuration as follows. This pin is connected with the LCD panel on SK-S7G2 boards to control data access timing from LCD_WR signal:



Pin Selection

type filter text

- Ports
  - P0
  - P1
    - P100
    - P101
    - P102
    - P103
    - P104
    - P105
    - P106
    - P107
    - P108
    - P109
    - P110
    - P111
    - P112
    - P113
    - P114
    - P115
  - P2
  - P3
  - P4
  - P5
  - P6
  - P7
  - P8
  - P9
  - PA
  - PB
- Peripherals
- Analog Pins

Pin Configuration

Module name: P115

Symbolic name: GPIO40

Comment:

P115 Configuration

Mode: Output mode

Pull up: None

Drive Capacity: Low

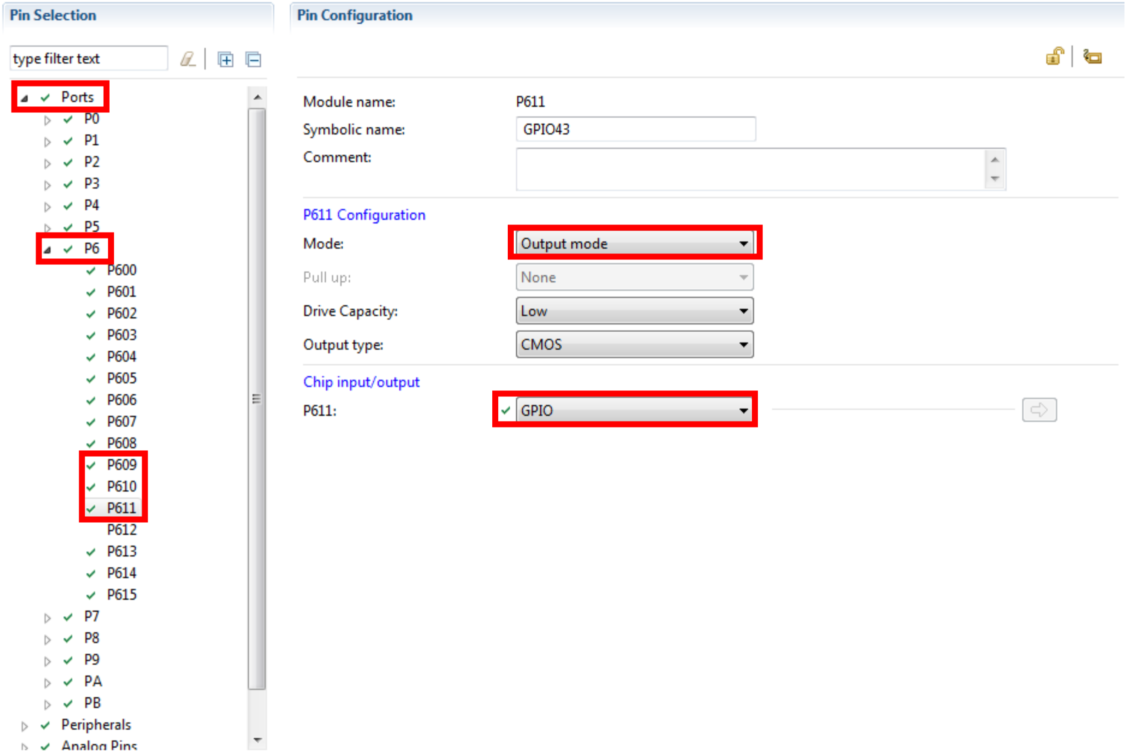
Output type: CMOS

Chip input/output

P115: ✓ GPIO

Figure 53 P115 configuration

39. Select Ports > P6 in Pin Selection, and configure P609 (RESET# for Touch Panel), P610 (LCD_RESET) and P611 (LCD_CS) with output mode of GPIO.



Pin Selection

type filter text

- Ports
  - P0
  - P1
  - P2
  - P3
  - P4
  - P5
  - P6
    - P600
    - P601
    - P602
    - P603
    - P604
    - P605
    - P606
    - P607
    - P608
    - P609
    - P610
    - P611
    - P612
    - P613
    - P614
    - P615
  - P7
  - P8
  - P9
  - PA
  - PB
- Peripherals
- Analog Pins

Pin Configuration

Module name: P611

Symbolic name: GPIO43

Comment:

P611 Configuration

Mode: Output mode

Pull up: None

Drive Capacity: Low

Output type: CMOS

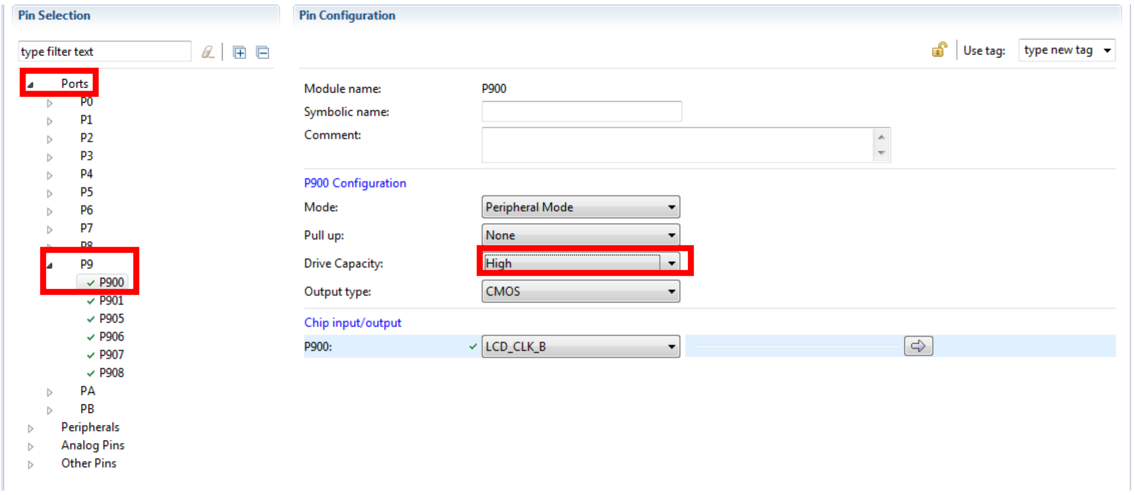
Chip input/output

P611: ✓ GPIO

Figure 54 P609, P610 and P611 configurations

40. Configure “Drive Capacity” into High for all pins related to GLCD_Controller_Pin_Option_B as shown in **Figure 55**. There are two methods for setting the Drive Capacities to High. You may pick either one (A or B):

    A. You can confirm which pins would be used for GLCD_Controller_Pin_Option_B shown in Figure 56 and Figure 57 below.



Module name: P900

Symbolic name:

Comment:

P900 Configuration

Mode: Peripheral Mode

Pull up: None

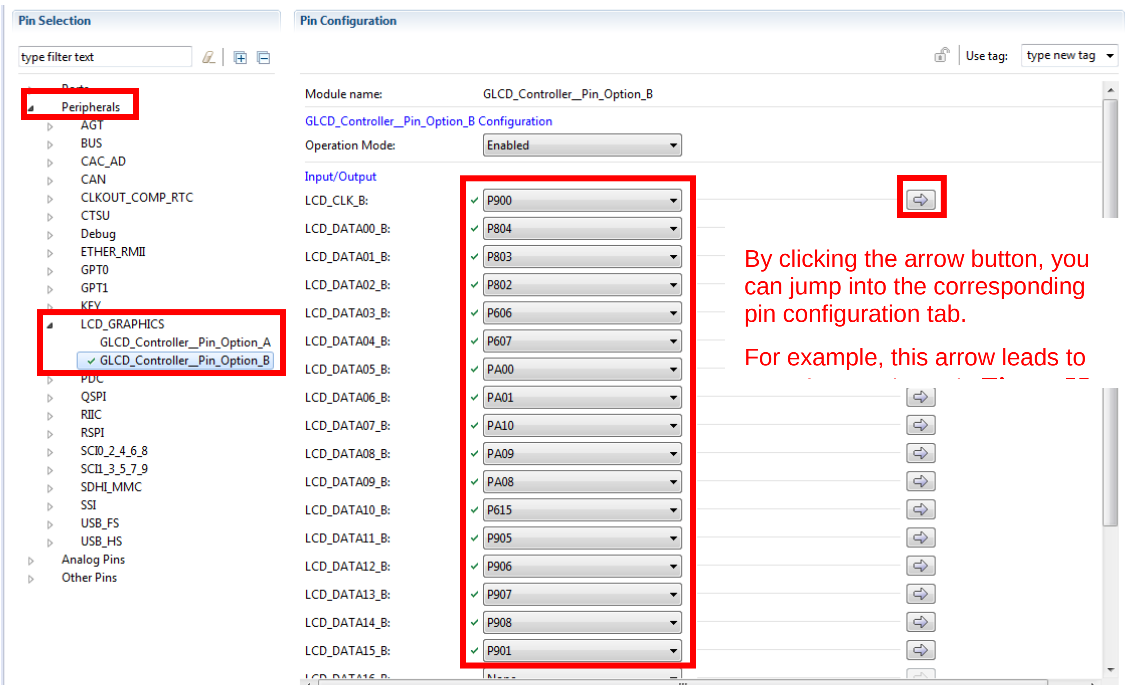
Drive Capacity: High

Output type: CMOS

Chip input/output

P900: LCD_CLK_B

Figure 55 Example of Drive Capability configuration for GLCDC



Module name: GLCD_Controller_Pin_Option_B

GLCD_Controller_Pin_Option_B Configuration

Operation Mode: Enabled

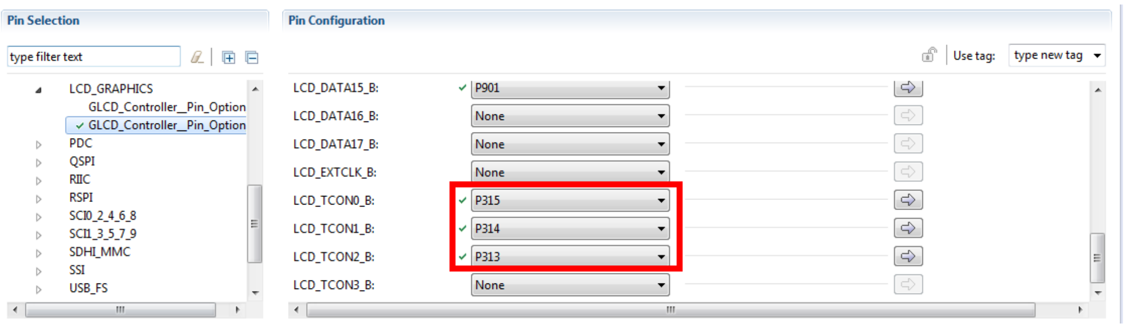
Input/Output

LCD_CLK_B:	P900
LCD_DATA00_B:	P804
LCD_DATA01_B:	P803
LCD_DATA02_B:	P802
LCD_DATA03_B:	P606
LCD_DATA04_B:	P607
LCD_DATA05_B:	PA00
LCD_DATA06_B:	PA01
LCD_DATA07_B:	PA10
LCD_DATA08_B:	PA09
LCD_DATA09_B:	PA08
LCD_DATA10_B:	P615
LCD_DATA11_B:	P905
LCD_DATA12_B:	P906
LCD_DATA13_B:	P907
LCD_DATA14_B:	P908
LCD_DATA15_B:	P901

By clicking the arrow button, you can jump into the corresponding pin configuration tab.

For example, this arrow leads to

Figure 56 Pin assignment for GLCD_Controller_Pin_Option_B



LCD_DATA15_B:	P901
LCD_DATA16_B:	None
LCD_DATA17_B:	None
LCD_EXTCLK_B:	None
LCD_TCON0_B:	P315
LCD_TCON1_B:	P314
LCD_TCON2_B:	P313
LCD_TCON3_B:	None

Figure 57 Pin assignment for GLCD_Controller_Pin_Option_B (continued)

B. You can also set the pins by port. Below is an ordered list of the pins that the “Drive Capacity” needs to be set on “High”. You can access these ports by going to Ports > PX > PXYZ. Where X is the second digit of the port from the list, and PXYZ is the entire port. Once the port is selected, set the “Drive Capacity” to “High” as shown in **Figure 55**:

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

Figure 58 Ordered List of Ports to Configure as High Drive Capacity

41. Select the Messaging tab on the Synergy Configuration Window. The following window displays:

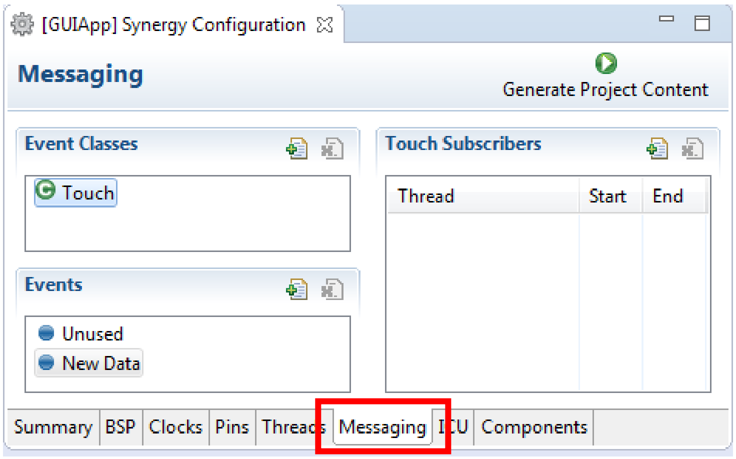


Figure 59 Messaging Tab

Note: This tab configures the event class definitions for the touchscreen events along with the event queue initialization and linking variables. The touch event was automatically generated when “Touch Panel Framework on sf_touch_panel_i2c” was added in the threads menu.

42. Select the Touch Event class.
43. On the touch subscribers menu, click the New button.

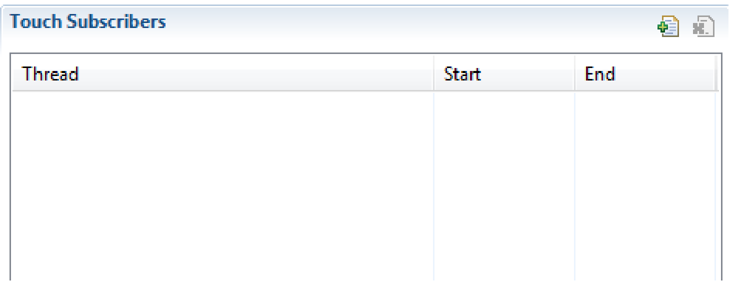
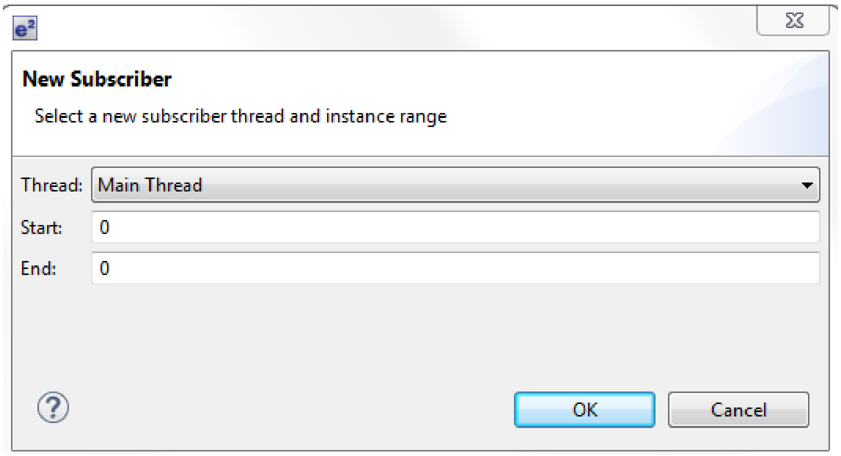


Figure 60 Messaging Tab

44. From the "New Subscriber" dialog, select Main Thread from the Thread drop-down list.



Thread: Main Thread
Start: 0
End: 0

Figure 61 New Subscriber dialog

45. Click OK.
46. Save the project by pressing "Ctrl + s" on the keyboard.
47. Click the Generate Project Content button to update the project files.

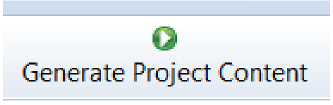


Figure 62 Generate Project Content

48. In the Project Explorer Window, right-click "src" and select New > Folder to bring up the new folder dialog box.

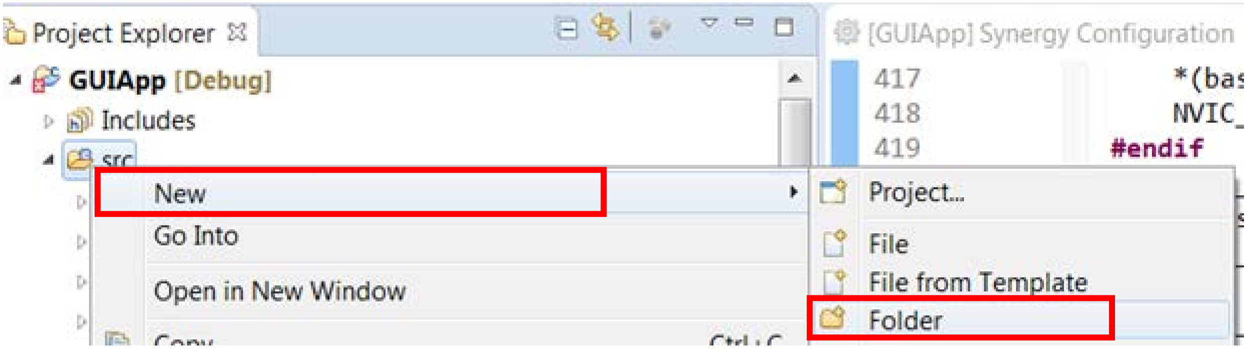
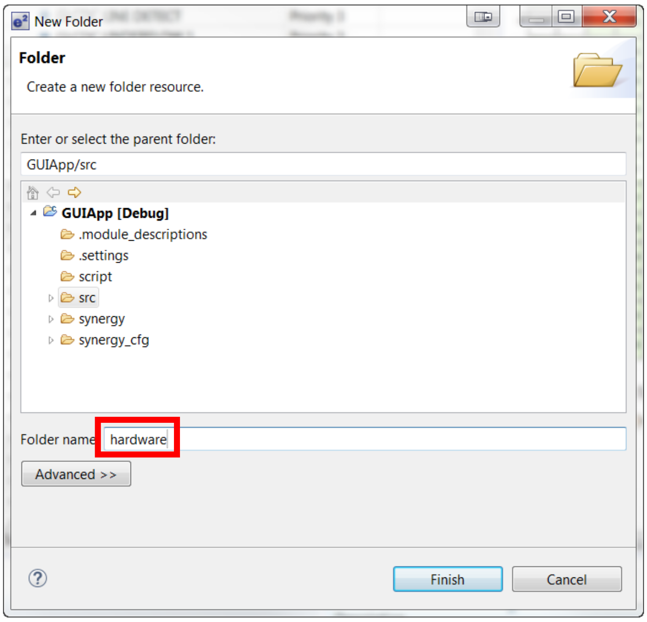


Figure 63 Creating a New Folder

49. Enter the name of the new folder, “hardware”, in the “Folder name:” text box.



New Folder

Folder

Create a new folder resource.

Enter or select the parent folder:

GUIApp/src

Folder name: hardware

Figure 64 New Folder Dialog

50. Click Finish.

51. The folder appears in Project Explorer and is highlighted.

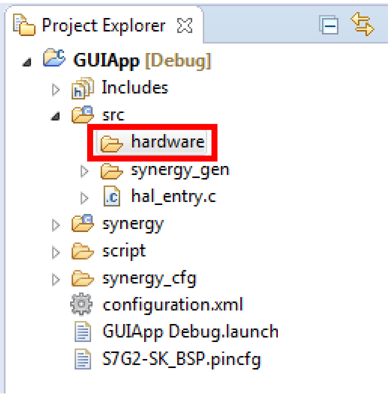
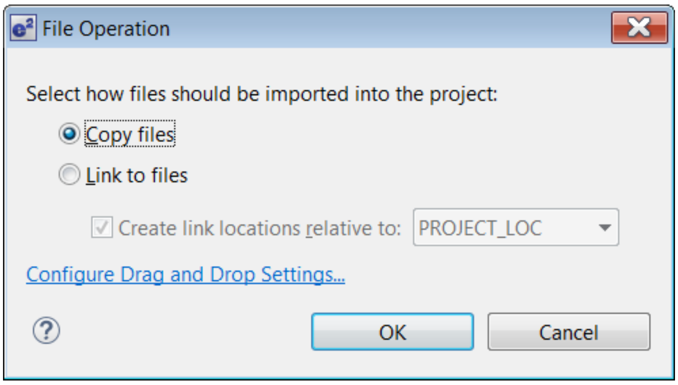


Figure 65 hardware folder

52. Open Windows Explorer and navigate to where you put the files included with this application note. Locate the file “Source Files**lcd.h**”. Now drag the file from the Windows Explorer Window into the new **“hardware”** folder inside the e^2 studio Project Explorer window.

53. When prompted to import the selected files, click OK to copy the files.



File Operation

Select how files should be imported into the project:

● Copy files

○ Link to files

☑ Create link locations relative to: PROJECT_LOC

Figure 66 File Operation Dialog

Note: This file contains the command definitions to control LCD panel.

54. Open Windows Explorer and navigate to where you put the files included with this application note. Locate the file "Source Files\ **lcd_setup.c**". Now drag the file from the Windows Explorer Window into the **"hardware"** folder inside the e^2 studio Project Explorer window.

55. When prompted to import the selected files, click OK to copy the files.

Note: This file contains command protocol via SPI to LCD panel and initialization sequence.

56. Open Windows Explorer and surf to where you put the files included with this application note. Locate the file "Source Files**main_thread_entry.c**". Now drag the file from the Windows Explorer Window into the **"src"** folder inside the e^2 studio Project Explorer window.

57. When prompted to import the selected files, click OK to copy the files.

58. When prompted to overwrite, click Yes.

Note: This file contains the Main Thread event handling code. It reads low level touchscreen events from the queue and transforms them to graphical user interface actions.

59. Locate the folder "synergy->ssp->src->driver->r_riic" inside the e^2 studio Project Explorer window.

60. Right-click the folder and select the Properties menu.

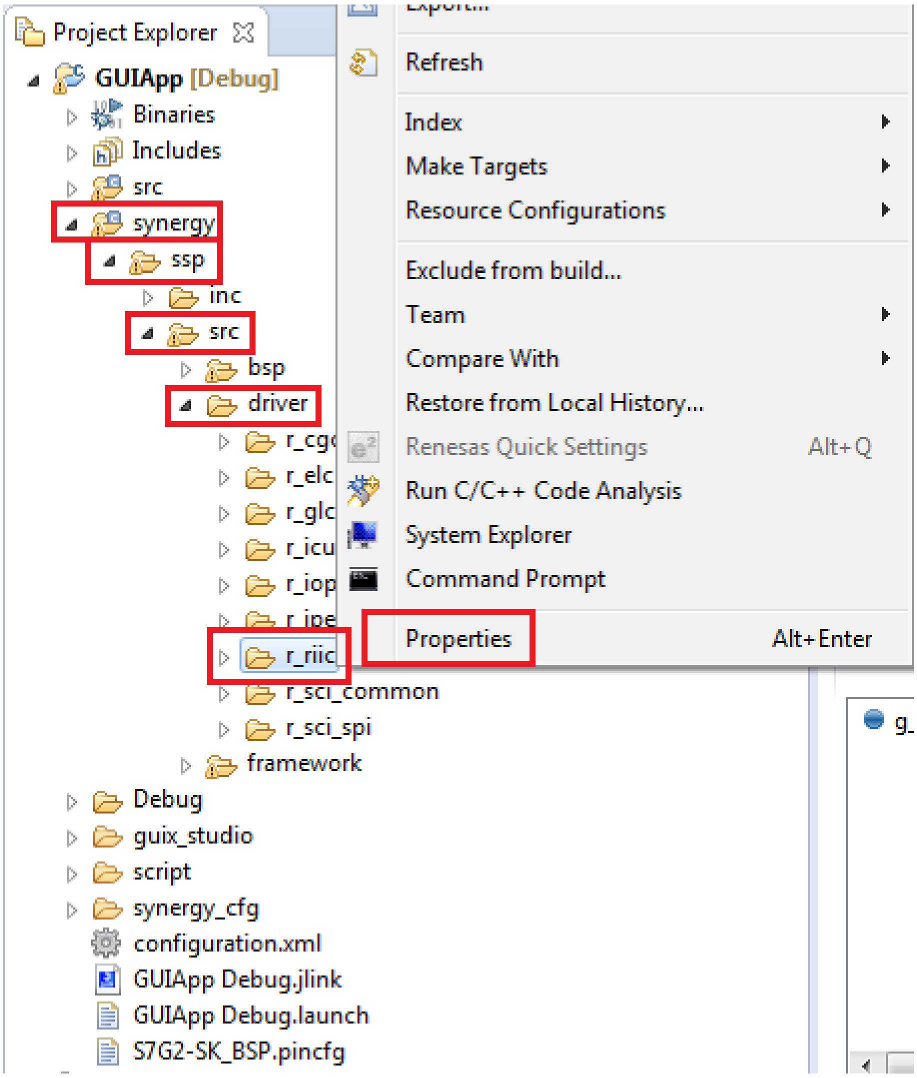
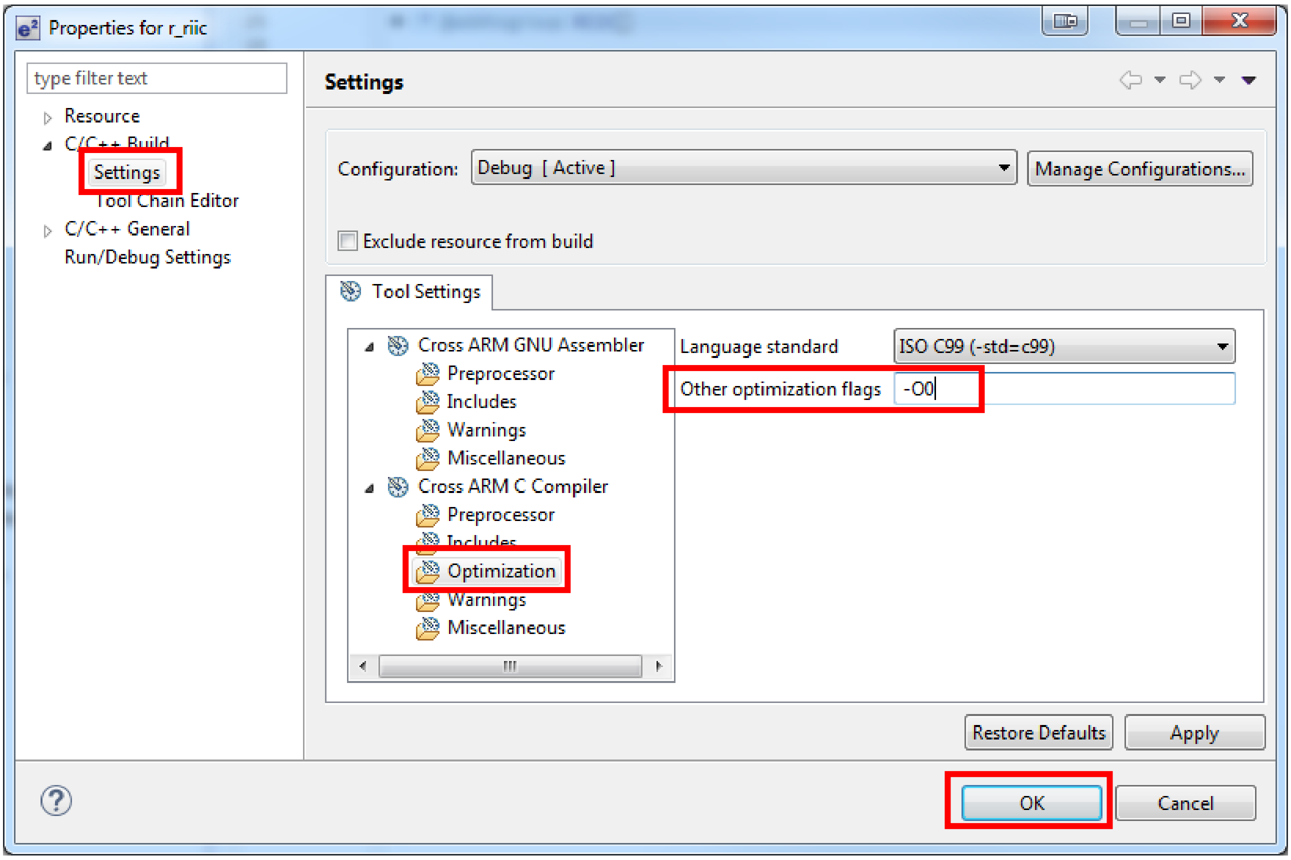


Figure 67 Creating "gui" folder under "src" folder

61. Navigate to: C/C++ Build >Settings >Cross ARM C Compiler > Optimization.

62. Inside the "Other optimization flags" set the following text option: -O0.



Properties for r_riic

Settings

Configuration: Debug [Active]

□ Exclude resource from build

Tool Settings

Language standard: ISO C99 (-std=c99)

Other optimization flags: -O0

Figure 68 Creating “gui” folder under “src” folder

63. Click OK.

Note: This step is included to fix an optimization bug in the RIIC driver.

5. Create the GUIX Interface Using GUIX Studio

Now that the base project is setup you can start adding the GUIX components.

1. Create a new folder named “gui” inside the “src” by right clicking on the “src” folder and selecting New > Folder.

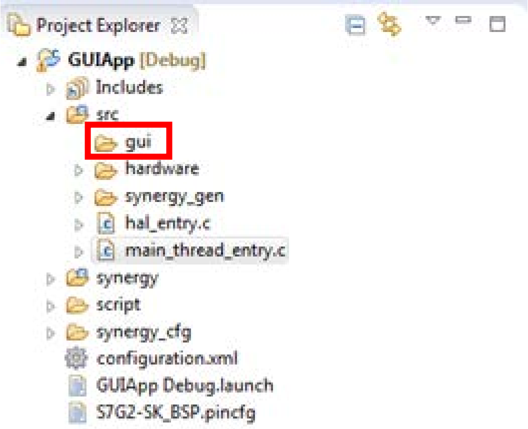


Figure 69 Creating “gui” folder under “src” folder

2. Create another new folder named “guix_studio” in the root folder of the project by right-clicking “GUIApp” and selecting New > Folder. The final folder layout should look like the figure below:

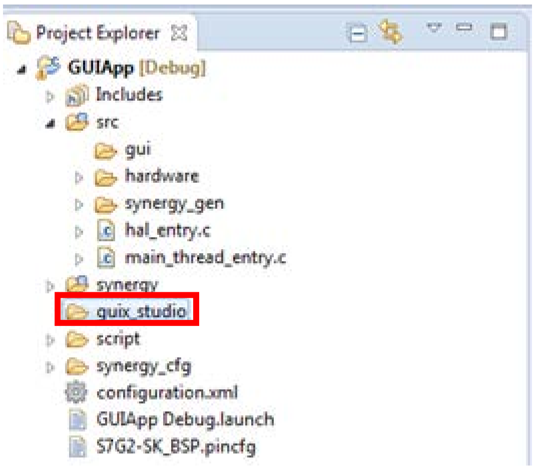


Figure 70 Final Folder List

3. Open GUIX Studio by clicking the desktop icon or by clicking the GUIX icon in the Windows Start Menu, All Programs > Express Logic > GUIX Studio 5.3 folder.

Figure 71 Start GUIX Studio

4. In the Recent Projects dialog click Create New Project…

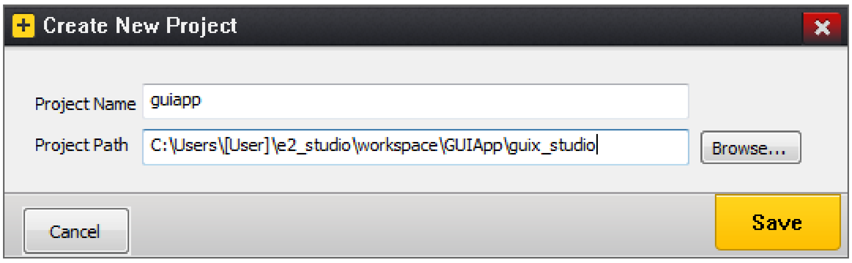
Figure 72 Create New Project

5. Name the project "guiapp".

   IMPORTANT: Filenames are generated by appending names to the project name. Be aware that the project name is case-sensitive. Later, you will add files to the project that you have named "guiapp".

6. For the Project Path, browse to the location of the folder we created earlier called "guix_studio".

Note: If you installed the tools into the default directories, the folder will be located at:
"C:\Users\[User]\e2_studio\workspace\GUIAPP\GUIApp\guix_studio"



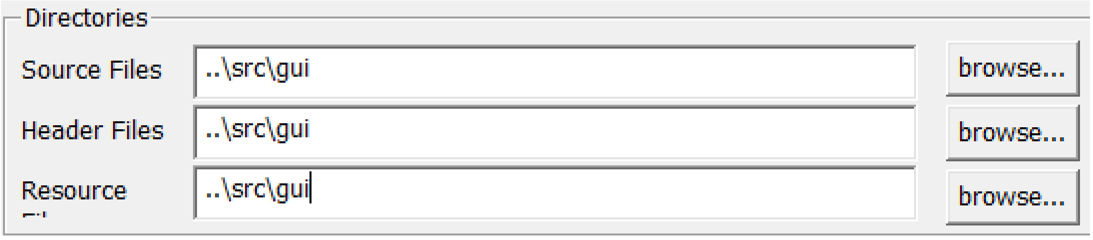
Project Name: guiapp

Project Path: C:\Users\[User]\e2_studio\workspace\GUIApp\guix_studio

Figure 73 Create a New GUIX Project

7. Click Save.

8. Change the Directories for all three options to be ..\src\gui.



Directories

Source Files ..\src\gui

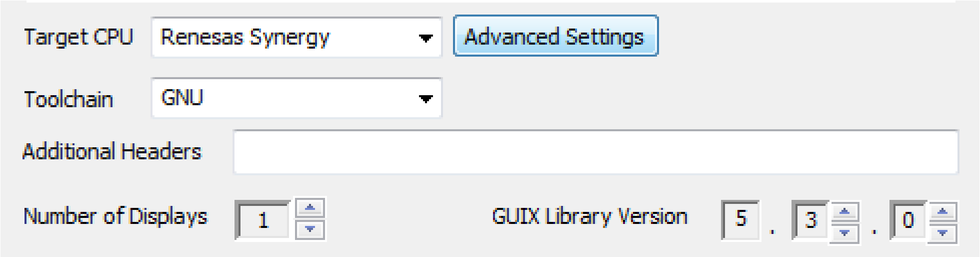
Header Files ..\src\gui

Resource Fi.. ..\src\gui

Figure 74 Correct the file Locations

IMPORTANT: Make sure you put in two dots “..” in the directories above.

9. Change the Target CPU Setting to “Renesas Synergy”.
10. Change the Toolchain setting to “GNU”.



Target CPU: Renesas Synergy

Toolchain: GNU

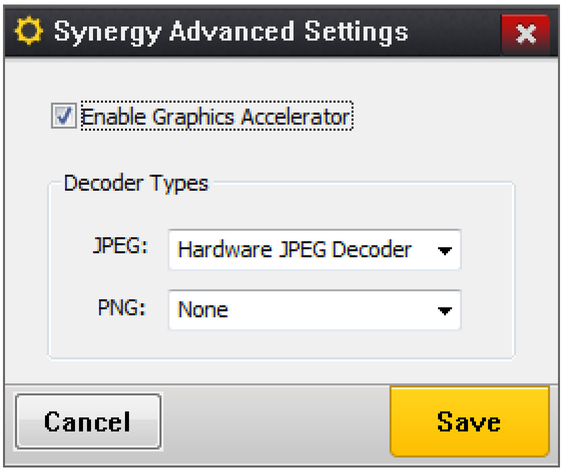
Additional Headers:

Number of Displays: 1

GUIX Library Version: 5 . 3 . 0

Figure 75 Target and GUIX version settings

11. Click Advanced Settings. A dialog will appear.
12. Enable the graphics accelerator and hardware JPEG decoder as shown below:



Synergy Advanced Settings

☑ Enable Graphics Accelerator

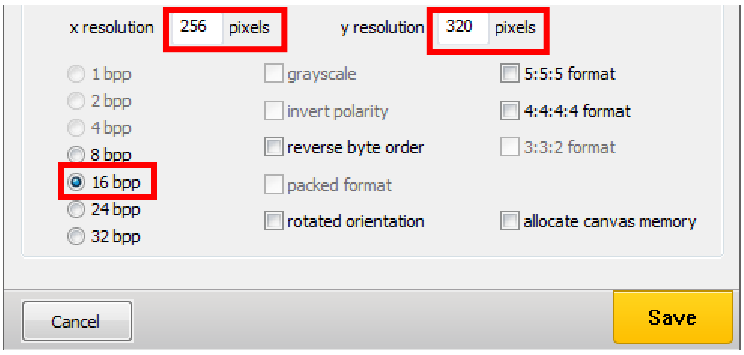
Decoder Types

JPEG: Hardware JPEG Decoder

PNG: None

Figure 76 Synergy Advanced Settings

13. Click Save.
14. Setup the Display Configuration as shown below:



x resolution 256 pixels   y resolution 320 pixels

○ 1 bpp
○ 2 bpp
○ 4 bpp
○ 8 bpp
● 16 bpp
○ 24 bpp
○ 32 bpp

□ grayscale
□ invert polarity
□ reverse byte order
□ packed format
□ rotated orientation

□ 5:5:5 format
□ 4:4:4:4 format
□ 3:3:2 format
□ allocate canvas memory

Figure 77 Configure Project

15. Click Save to generate the project.
16. Right-click display_1 in the project view.
17. Select Insert > Window > Window.

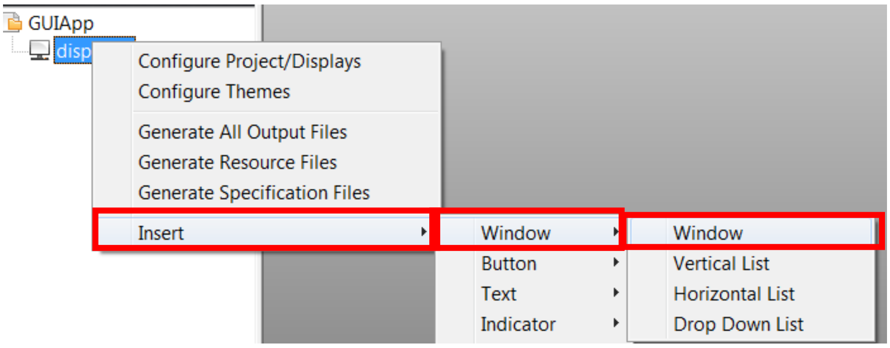
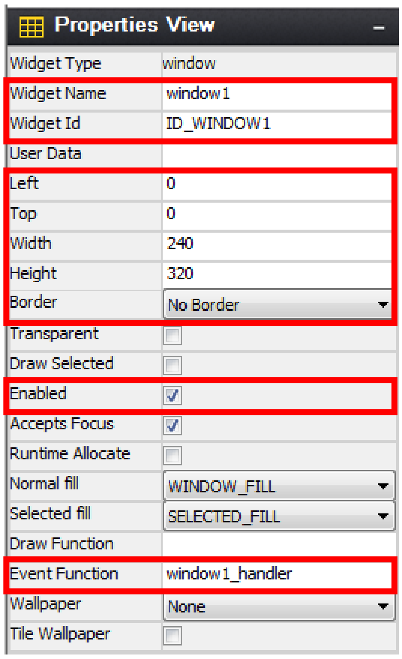


Figure 78 New Window

18. Modify the properties by selecting the new window and editing the Properties View. Update the current settings to match the following. Notice the Event Function field. This is the event that will be initiated when the touch screen is pressed in window1.

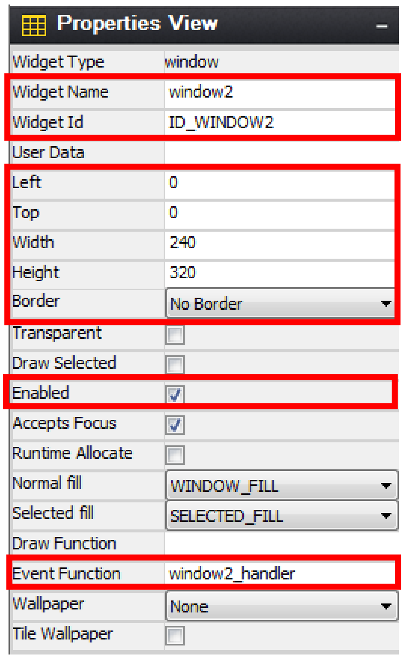


Properties View

Widget Type	window
Widget Name	window1
Widget Id	ID_WINDOW1
User Data	
Left	0
Top	0
Width	240
Height	320
Border	No Border
Transparent	□
Draw Selected	□
Enabled	☑
Accepts Focus	☑
Runtime Allocate	□
Normal fill	WINDOW_FILL
Selected fill	SELECTED_FILL
Draw Function	
Event Function	window1_handler
Wallpaper	None
Tile Wallpaper	□

Figure 79 Configure Window1 Properties

19. At this point, you may notice the window does not occupy the entire display. This is expected when working with GUIX with small screens and will not affect the display once the application is running.
20. In the Project View Window, right-click display_1 and create another window by selecting Insert > Window > Window.
21. Modify the properties to match the following. Notice the Event Function field. This is the event that will be initiated when the touch screen is pressed in window2.



Properties View

Widget Type	window
Widget Name	window2
Widget Id	ID_WINDOW2
User Data	
Left	0
Top	0
Width	240
Height	320
Border	No Border
Transparent	□
Draw Selected	□
Enabled	☑
Accepts Focus	☑
Runtime Allocate	□
Normal fill	WINDOW_FILL
Selected fill	SELECTED_FILL
Draw Function	
Event Function	window2_handler
Wallpaper	None
Tile Wallpaper	□

Figure 80 Configure Window2 Properties

22. In the Project View, right-click window1 and insert a Button (Text Button) by selecting Insert > Button >Text Button.

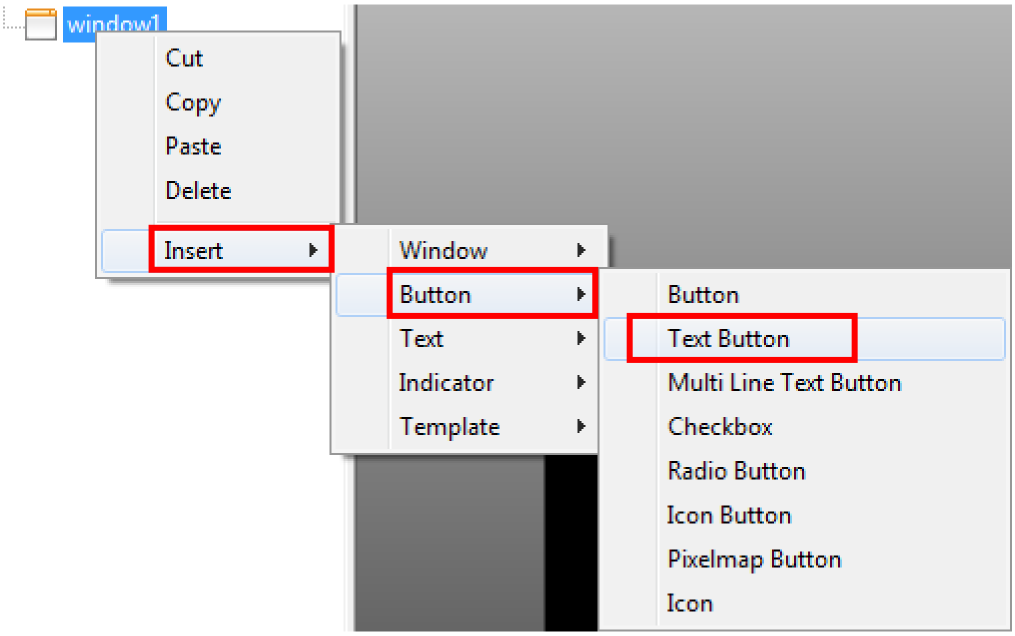


Figure 81 Add a New Text Button

23. In the Project View, right-click window1 and insert a Button Checkbox by selecting Insert > Button > Checkbox.

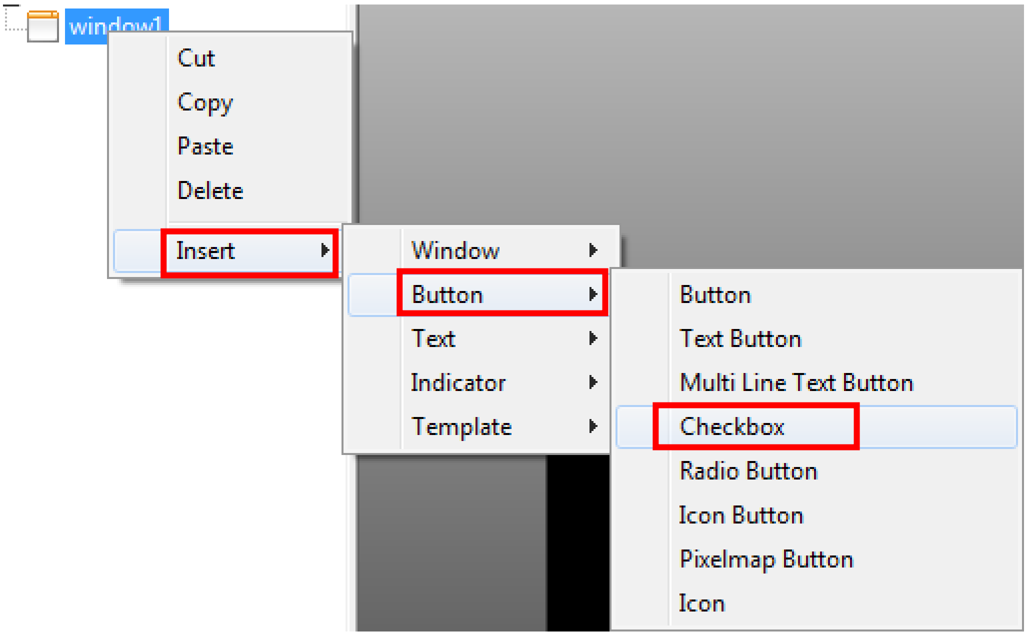


Figure 82 Add a New Checkbox

24. In the Project View, right-click window1 and insert a Text Prompt by selecting Insert > Text > Prompt.

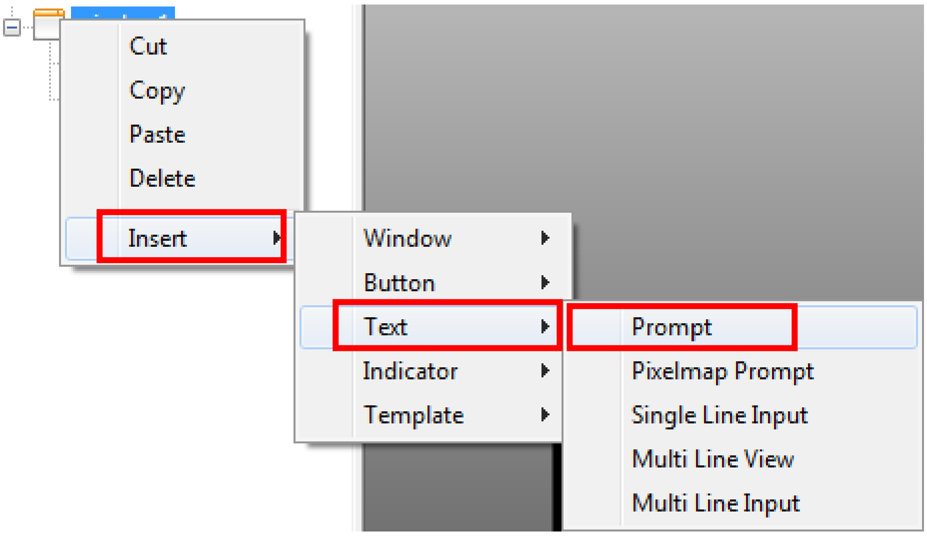


Figure 83 Adding New Prompt

25. In the Project View, right-click window1 and insert another Text Prompt.

26. In the Project View, right-click window2 and insert a Text Prompt.

27. In the Project View, right-click window2 and insert another Text Prompt.

28. If you have followed these directions correctly, your Project View should look like this:

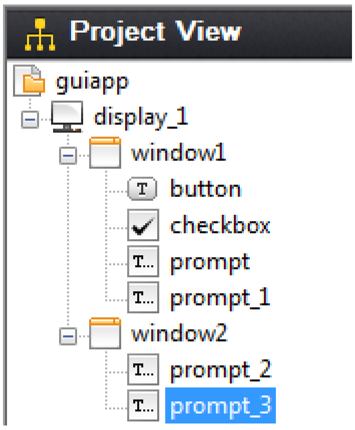


Figure 84 GUIX Project View

29. Expand the Strings menu by clicking +.

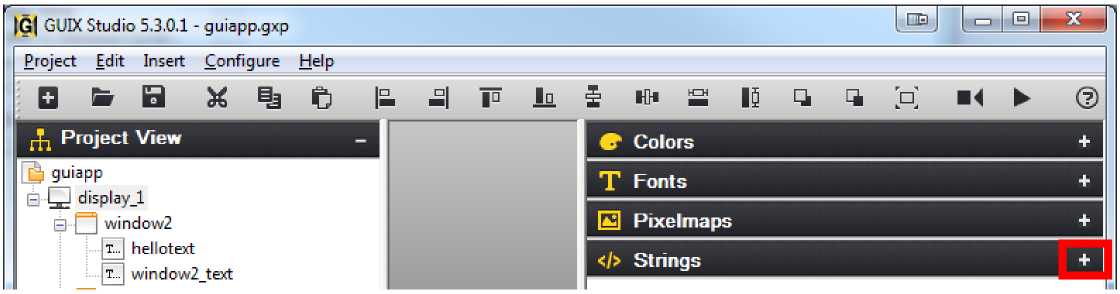
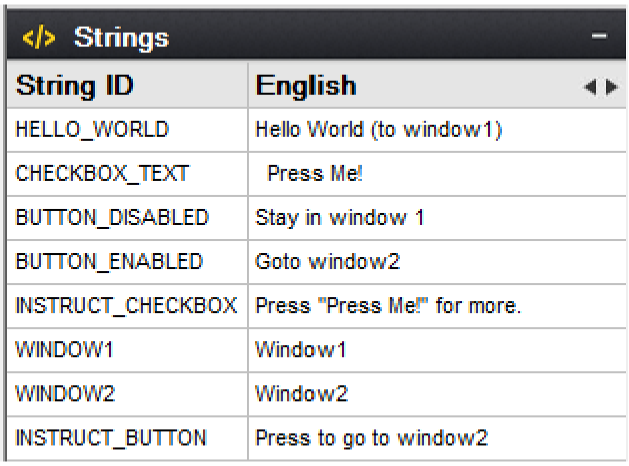


Figure 85 Strings Button

30. Double-click any of the strings to open the String Table Editor.

31. Delete the existing strings by selecting them then clicking the Delete String button in the String Table Editor.

32. Add the following Strings using the Add String button:



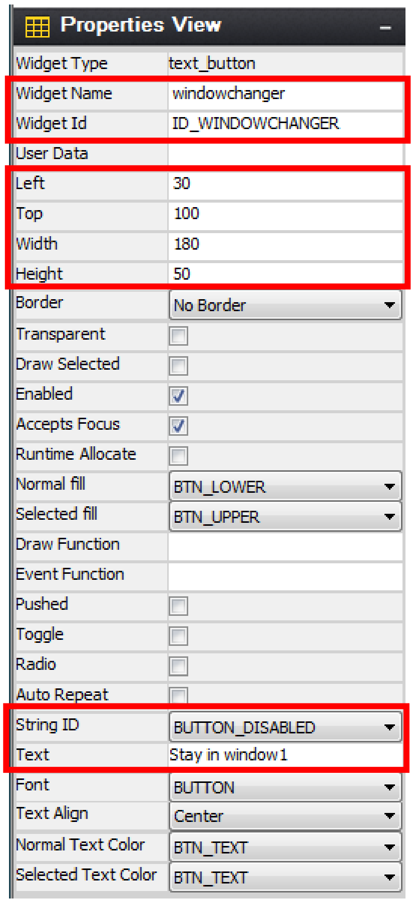
Strings

String ID	English
HELLO_WORLD	Hello World (to window1)
CHECKBOX_TEXT	Press Me!
BUTTON_DISABLED	Stay in window 1
BUTTON_ENABLED	Goto window2
INSTRUCT_CHECKBOX	Press "Press Me!" for more.
WINDOW1	Window1
WINDOW2	Window2
INSTRUCT_BUTTON	Press to go to window2

Figure 86 New Strings

33. When completed, click Save.

34. In the Project View under window1, click on button, then modify the properties (Properties View) to match the following:

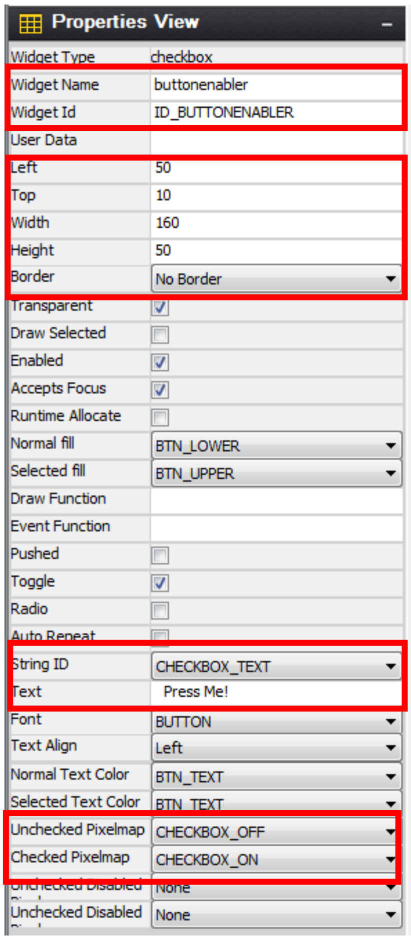


Properties View

Widget Type	text_button
Widget Name	windowchanger
Widget Id	ID_WINDOWCHANGER
User Data	
Left	30
Top	100
Width	180
Height	50
Border	No Border
Transparent	□
Draw Selected	□
Enabled	☑
Accepts Focus	☑
Runtime Allocate	□
Normal fill	BTN_LOWER
Selected fill	BTN_UPPER
Draw Function	
Event Function	
Pushed	□
Toggle	□
Radio	□
Auto Repeat	□
String ID	BUTTON_DISABLED
Text	Stay in window1
Font	BUTTON
Text Align	Center
Normal Text Color	BTN_TEXT
Selected Text Color	BTN_TEXT

Figure 87 Configure windowchanger Button Properties

35. In the Project View under window1, click the checkbox, then modify the properties (Properties View) to match the following:

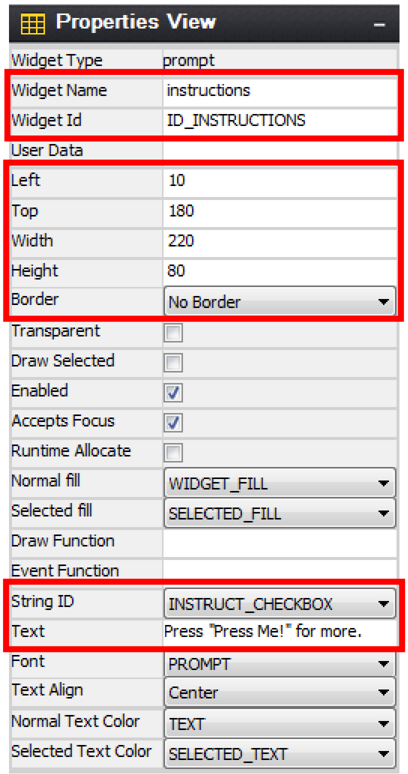


Properties View

Widget Type	checkbox
Widget Name	buttonenabler
Widget Id	ID_BUTTONENABLER
User Data	
Left	50
Top	10
Width	160
Height	50
Border	No Border
Transparent	☑
Draw Selected	□
Enabled	☑
Accepts Focus	☑
Runtime Allocate	□
Normal fill	BTN_LOWER
Selected fill	BTN_UPPER
Draw Function	
Event Function	
Pushed	□
Toggle	☑
Radio	□
Auto Repeat	□
String ID	CHECKBOX_TEXT
Text	Press Me!
Font	BUTTON
Text Align	Left
Normal Text Color	BTN_TEXT
Selected Text Color	BTN_TEXT
Unchecked Pixelmap	CHECKBOX_OFF
Checked Pixelmap	CHECKBOX_ON
Unchecked Disabled	None
Unchecked Disabled	None

Figure 88 Configure Buttonenabler Checkbox Properties

36. In the Project View under window1, click Prompt, then modify the properties to match the following:

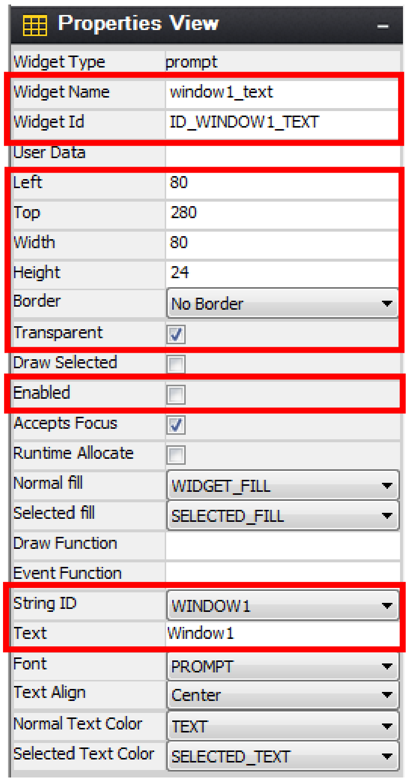


Properties View

Widget Type	prompt
Widget Name	instructions
Widget Id	ID_INSTRUCTIONS
User Data	
Left	10
Top	180
Width	220
Height	80
Border	No Border
Transparent	□
Draw Selected	□
Enabled	☑
Accepts Focus	☑
Runtime Allocate	□
Normal fill	WIDGET_FILL
Selected fill	SELECTED_FILL
Draw Function	
Event Function	
String ID	INSTRUCT_CHECKBOX
Text	Press "Press Me!" for more.
Font	PROMPT
Text Align	Center
Normal Text Color	TEXT
Selected Text Color	SELECTED_TEXT

Figure 89 Configure Prompt Properties

37. In the Project View under window1, click prompt_1, then modify the properties to match the following:

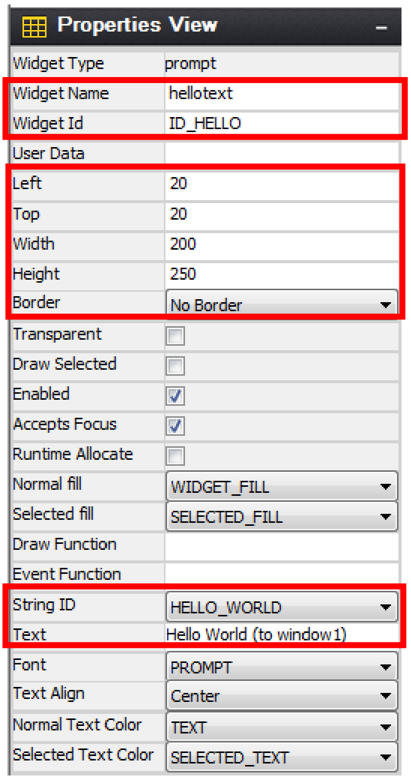


Properties View

Widget Type	prompt
Widget Name	window1_text
Widget Id	ID_WINDOW1_TEXT
User Data	
Left	80
Top	280
Width	80
Height	24
Border	No Border
Transparent	☑
Draw Selected	☐
Enabled	☐
Accepts Focus	☑
Runtime Allocate	☐
Normal fill	WIDGET_FILL
Selected fill	SELECTED_FILL
Draw Function	
Event Function	
String ID	WINDOW1
Text	Window1
Font	PROMPT
Text Align	Center
Normal Text Color	TEXT
Selected Text Color	SELECTED_TEXT

Figure 90 Configure Window Text Properties

38. In the Project View under window2, click prompt_2, then modify the properties to match the following:

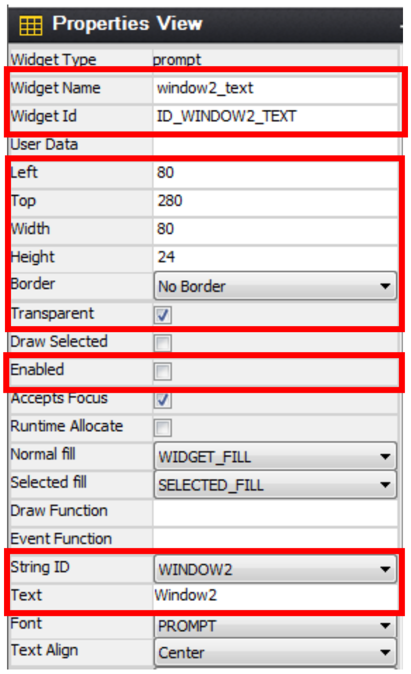


Properties View

Widget Type	prompt
Widget Name	hellotext
Widget Id	ID_HELLO
User Data	
Left	20
Top	20
Width	200
Height	250
Border	No Border
Transparent	☐
Draw Selected	☐
Enabled	☑
Accepts Focus	☑
Runtime Allocate	☐
Normal fill	WIDGET_FILL
Selected fill	SELECTED_FILL
Draw Function	
Event Function	
String ID	HELLO_WORLD
Text	Hello World (to window1)
Font	PROMPT
Text Align	Center
Normal Text Color	TEXT
Selected Text Color	SELECTED_TEXT

Figure 91 Configure Hello Text Prompt Properties

39. In the Project View under window2, click prompt_3, then modify the properties to match the following:



Properties View

Widget Type	prompt
Widget Name	window2_text
Widget Id	ID_WINDOW2_TEXT
User Data	
Left	80
Top	280
Width	80
Height	24
Border	No Border
Transparent	☑
Draw Selected	□
Enabled	□
Accepts Focus	☑
Runtime Allocate	□
Normal fill	WIDGET_FILL
Selected fill	SELECTED_FILL
Draw Function	
Event Function	
String ID	WINDOW2
Text	Window2
Font	PROMPT
Text Align	Center

Figure 92 Configure Window Text Properties

After these configuration steps, the two windows should look similar to the following images:

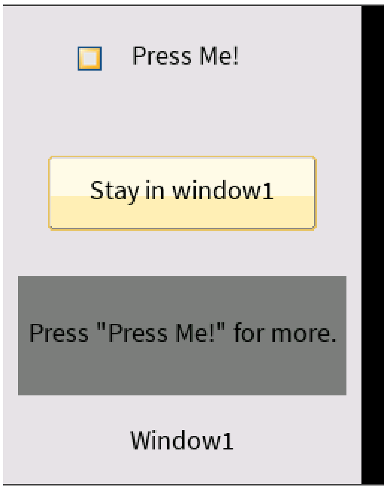


Figure 93 Configured Window1

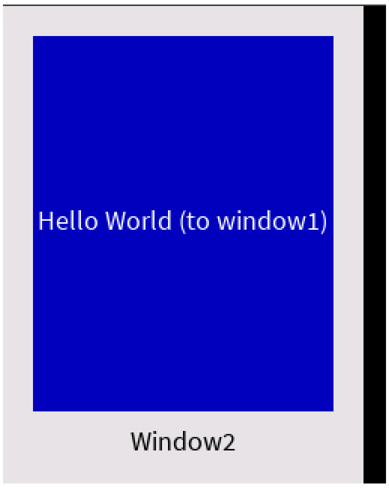


Figure 94 Configured Window2

40. Save the project.

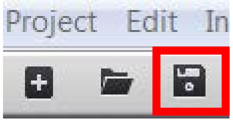


Figure 95 Save Project

41. From Project select Generate all Output Files.

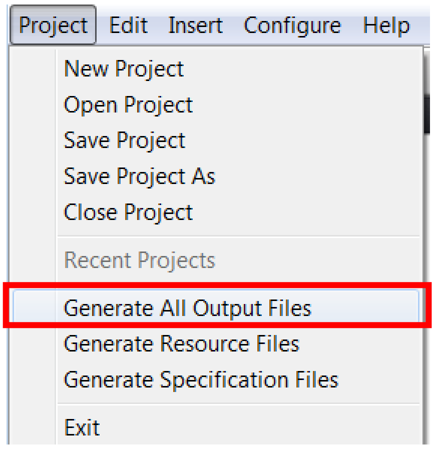


Figure 96 Generate All Output files

42. Return to e^2 studio.

6. Adding Code for Custom Interface Controls and Building project

1. Open Windows Explorer and navigate to where you put the files included with this application note. Locate the file “Source Files\guiapp_event_handlers.c”. Now drag the file from the Windows Explorer Window into the “src” folder inside the e2 studio Project Explorer window.

2. When prompted to import the selected files, click OK to copy the files

Note: This file contains the event management functions for the different graphical elements created in GUIX Studio (window1, window2).

GUIX handles the events that are required at a system level but to handle custom commands like screen transitions and button actions event handler need to be defined. Shown below is the event handler for window1.

```
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;
}
```

Events can be routed based on the ID of the widget and the signal from GUIX. For example the checkbox ID_BUTTONENABLER can have two states; GX_EVENT_TOGGLE_ON and GX_EVENTS_TOGGLE_OFF. When the box is unchecked and then pressed, the event GX_EVENT_TOGGLE_ON is sent to the handler, after which the box will be checked.

3. Turn optimization off:

   A. Right-click GUIApp in the Project Explorer window and select Properties from the context menu.

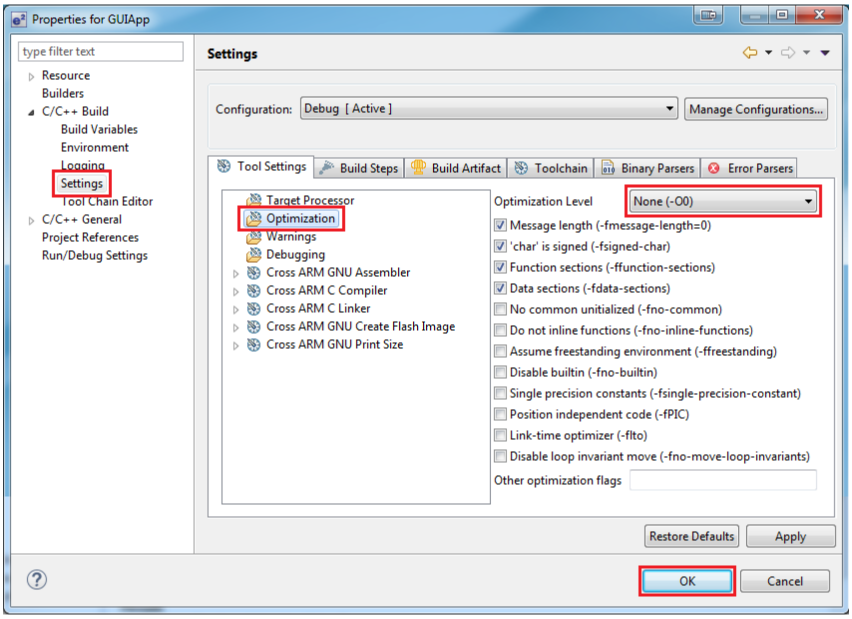
   B. Within the properties window, expand the C/C++ Build tree element.

   C. Select Settings.

   D. In the Tool Settings tab, click Optimization.

   E. Change the Optimization Level to None (-O0).

   F. Click OK to save these changes.



Configuration: Debug [Active]

Optimization Level: None (-O0)

- ☑ Message length (-fmessage-length=0)
- ☑ 'char' is signed (-fsigned-char)
- ☑ Function sections (-ffunction-sections)
- ☑ Data sections (-fdata-sections)
- □ No common unitialized (-fno-common)
- □ Do not inline functions (-fno-inline-functions)
- □ Assume freestanding environment (-ffreestanding)
- □ Disable builtin (-fno-builtin)
- □ Single precision constants (-fsingle-precision-constant)
- □ Position independent code (-fPIC)
- □ Link-time optimizer (-flto)
- □ Disable loop invariant move (-fno-move-loop-invariants)

Other optimization flags

Figure 97 Disabling Compiler Optimizations

4. Build the project by clicking the Hamer icon below the Menu Bar, 🔨. If all steps were followed correctly, there will be no errors reported in the build output.

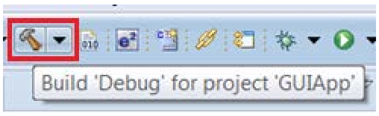
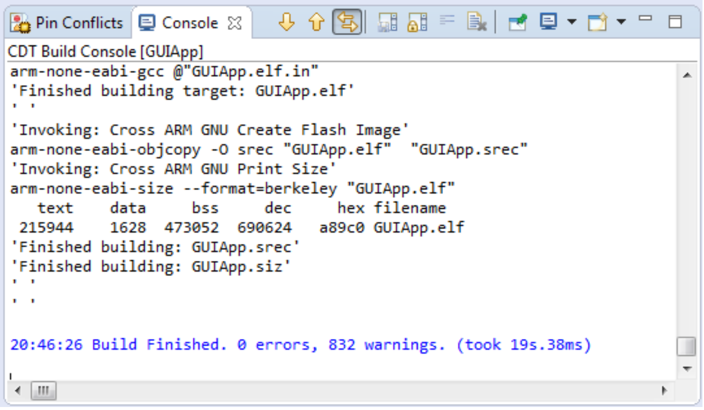


Figure 98 Build the Project



```
CDT Build Console [GUIApp]
arm-none-eabi-gcc @"GUIApp.elf.in"
'Finished building target: GUIApp.elf'
' '
'Invoking: Cross ARM GNU Create Flash Image'
arm-none-eabi-objcopy -O srec "GUIApp.elf"  "GUIApp.srec"
'Invoking: Cross ARM GNU Print Size'
arm-none-eabi-size --format=berkeley "GUIApp.elf"
   text    data     bss     dec     hex filename
 215944    1628  473052  690624   a89c0 GUIApp.elf
'Finished building: GUIApp.srec'
'Finished building: GUIApp.siz'
' '
' '

20:46:26 Build Finished. 0 errors, 832 warnings. (took 19s.38ms)
```

Figure 99 Build finished with 0 errors

7. Running the Application

1. Connect the SK-S7G2 (J19) to the PC with micro USB cable.

Note: The application is not yet ready to be run on the target hardware. The following steps are necessary in order to run it.

2. Click the drop-down menu for the debug icon.

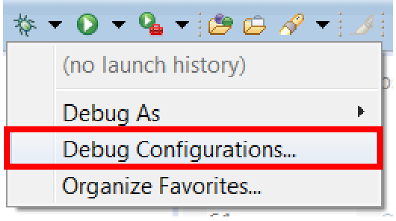
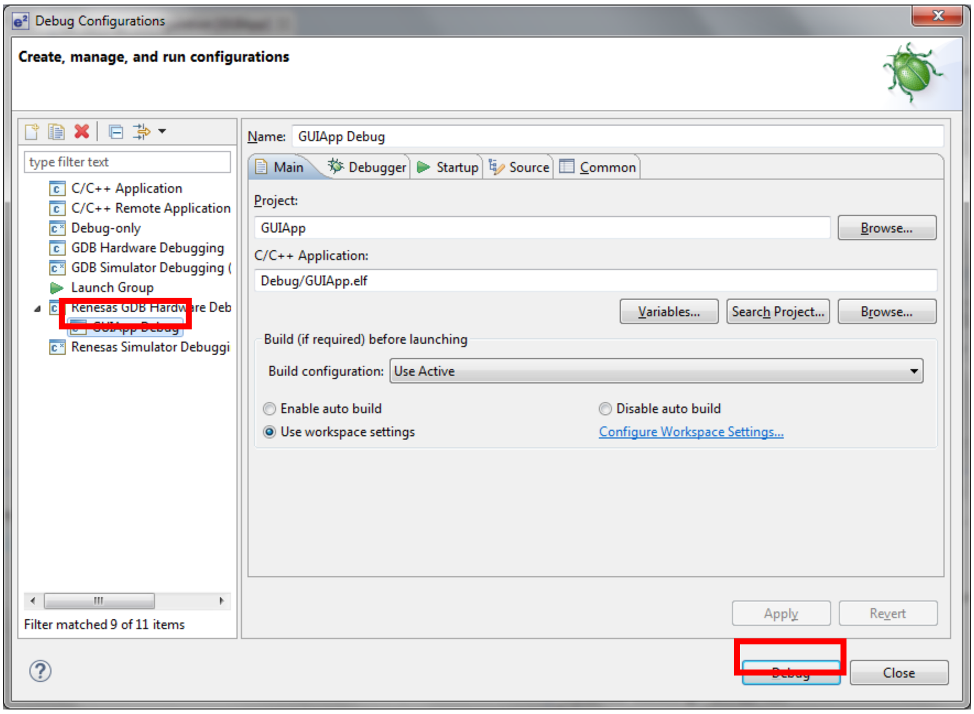


Figure 100 Debug Options

3. Select the Debug Configurations... option
4. Under the Renesas GDB Hardware Debugging section, select GUIApp Debug.
5. Click the Debug button to start debugging.

Note: If the debug button is greyed out then there is likely an issue with the build. Check all steps from the document again for mismatched options.


Debug Configurations

Create, manage, and run configurations

Name: GUIApp Debug

Project: GUIApp

C/C++ Application: Debug/GUIApp.elf

Build (if required) before launching

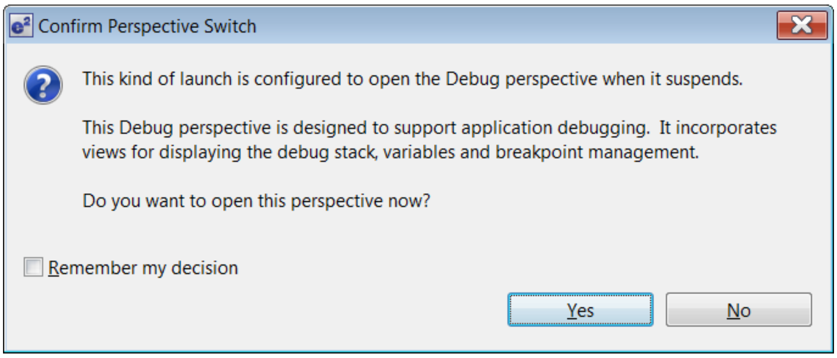
Build configuration: Use Active

○ Enable auto build ○ Disable auto build

● Use workspace settings Configure Workspace Settings...

Figure 101 Debug Configurations

6. If asked to confirm a Perspective Switch, click Yes. (If you previously instructed e^2 studio to remember your decision, this dialog box will not be displayed.)


Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

□ Remember my decision

Figure 102 Perspective Switch Dialog

7. Press F8 or the resume button to start the application. It will stop at main.

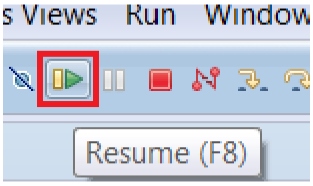


Figure 103 Resume Button

8. Press F8 or the resume button to run the code.

Note: The GUI created earlier should display on the screen.

9. Overview of the Demo:

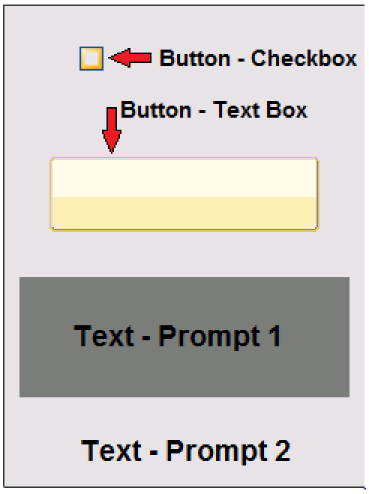


Figure 104 Window1

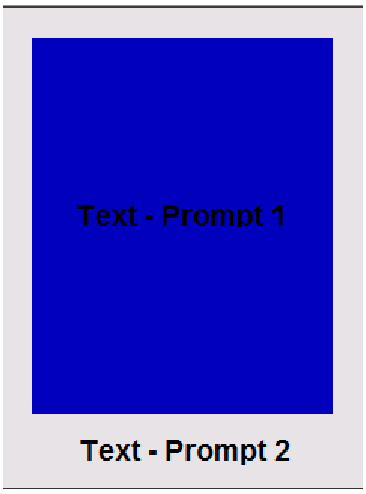
A. Figure 104 shows Window1. In this window are four elements:

   i. **Button – Checkbox**: Use this button to enable navigating to Window2. Text is set to “Press Me!” and it is unchecked. When you click within the Checkbox active area, the event “window1_handler” is activated. This event is picked up inside “guiapp_event_handlers.c” where the code toggles the checkbox then sets the text in “Text –Prompt 1” and “Button – Text Box” to the appropriate message.

   ii. **Button – Text Box**: This box shows which window you will go to if you press outside the “Text – Prompt 1” area. (Refer to “Button – Checkbox” to see how it is changed.) Click in this area to activate the “window1 _handler” event which is picked up by “guiapp_event_handlers.c” where the code changes the window to window2.

   iii. **Text – Prompt 1**: This area instructs you how to control the demo. (Refer to “Button – Checkbox” to see how it is changed.)

   iv. **Text – Prompt 2:** This Prompt is used to show you which window you are in. It never changes (always shows window1).



Text - Prompt 1

Text - Prompt 2

Figure 105 Window2

B. **Figure 105** shows Window2. In this window are two elements:

i. **Text – Prompt 1**: This area presents “Hello World”. Clicking in this area initiates the “window2_handler” event which is picked up guiapp_event_handlers.c and changes the active window to window1.

ii. **Text – Prompt 2:** This Prompt is used to show you which window you are in. It never changes (always shows window2).

10. Press Ctrl + F2 or the stop button to end the debug session.

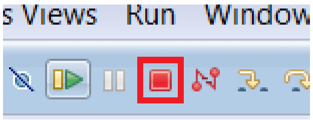


Figure 106 Stop Button

11. This concludes the GUIX “Hello World” for SK-S7G2.

Website and Support

Support: https://synergygallery.renesas.com/support

Technical Contact Details:

- America: https://renesas.zendesk.com/anonymous_requests/new
- Europe: http://www.renesas.eu/support/index.jsp
- Japan: http://japan.renesas.com/contact/index.jsp

All trademarks and registered trademarks are the property of their respective owners.

Revision History

		Description	
Rev.	Date	Page	Summary
1.00	1/22/2016	All	Initial version
1.01	4/12/2016	All	Updated lcd_setup.c to correct semaphore naming issue
1.10	8/30/2016	All	Update to SSP v1.1.0

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

— The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

— The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

— The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

— When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

— The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0