

Renesas Synergy<sup>™</sup> Platform

## GUIX "Hello World" for DK-S7G2

## Introduction

This application note guides you through the process of creating a simple two-screen GUI using GUIX Studio for the DK-S7G2 Synergy MCU kit. Its application project demonstrates how easily you can create and configure a new application using the Renesas Synergy<sup>™</sup> Software Package (SSP).

The Synergy Software Package includes Express Logic's ThreadX<sup>®</sup> real-time operating system (RTOS), the X-Ware suite of stacks (NetX<sup>™</sup>, USBX<sup>™</sup>, GUIX<sup>™</sup>, and FileX<sup>®</sup>), and a set of 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 under e<sup>2</sup> studio using the Synergy Framework.

## **Target Device**

DK-S7G2 board version 3.1 and 4.1

### **Minimum PC Recommendation**

- Microsoft<sup>®</sup> Windows<sup>®</sup> 7/8/10
- Intel<sup>®</sup> Core<sup>™</sup> 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<sup>™</sup> e<sup>2</sup> studio Integrated Solution Development Environment (ISDE) Version 2021 (21.7.0) or later
- Synergy<sup>™</sup> Software Package (SSP) v2.2.0 or later
- GUIX Studio v6.1.8 or later

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
- R7FS7G27H2A01CBD.pincfg

### Purpose

To guide you through the setup of a GUIX touch screen interface for the **Hello World** application in e<sup>2</sup> studio, where you configure the drivers and framework included with the SSP. Project setup in e<sup>2</sup> studio includes setup of basic debugging operations. When you have the configuration ready, you can set up the LCD Controller, touch screen drivers, and messaging framework to communicate with application tasks. You can also create a simple GUI interface using the GUIX Studio editor. Once the application is running, it responds to touchscreen actions using the Touch Panel V2 Framework on sf\_touch\_panel\_v2 Framework, presenting a basic graphical user interface (GUI).

### **Intended Audience**

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



## Contents

1.	Overview	3
2.	Importing the Project into e <sup>2</sup> studio	3
3.	Creating the Project in e <sup>2</sup> studio	3
4.	Configuring the project in e <sup>2</sup> studio	8
5.	Creating the GUIX Interface using GUIX Studio	29
6.	Adding Code for Custom Interface Controls	42
7.	Running the Application	44
8.	Appendix	47
Rev	<i>i</i> ision History	49



## 1. Overview

This application note shows how to set up a project and develop a simple GUI-based application using GUIX Studio.

## 2. Importing the Project into e<sup>2</sup> studio

Note: This step is included to allow you to skip the development steps and go to the point of verifying a working project on the DK-S7G2. Most users SKIP THIS STEP and continue to section 3 to create a project in e<sup>2</sup> studio. If you do choose to import the project, go to section 7, Running the Application.

To skip the development walkthrough steps and open a completed project in  $e^2$  studio, see the *Renesas Synergy Project Import Guide* (REN\_r11an0023eu0121-synergy-ssp-import-guide\_APN\_20181022.pdf) in the package. It contains instructions on importing the project into  $e^2$  studio and building the project. The included GUIX\_Hello\_World\_DK-S7G2.zip file contains the completed project.

## 3. Creating the Project in e<sup>2</sup> studio

Start by creating a new project in e<sup>2</sup> studio.

- 1. Open e<sup>2</sup> studio by clicking on the e<sup>2</sup> studio icon in the **Windows Start Menu > All Programs > Renesas** Electronics e<sup>2</sup> studio folder.
- 2. If the workspace launcher dialog box appears, click **OK** to use the default workspace.

e <sup>2</sup> Workspace Launcher	<b>X</b>
Select a workspace	
e2 studio stores your projects in a folder called a workspace Choose a workspace folder to use for this session.	2.
Workspace: C:\Users\Johnathan Doe\e2_studio\workspace\	Browse
Use this as the default and do not ask again	
	OK Cancel

Figure 1. Workspace Launcher Dialog

- Create a new workspace:
   From the File pulldown menu, select Switch Workspace > Other...
- Append a workspace name: In the Workspace Launcher window, add text to the end of the workspace name to make it unique, such as 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. Click in the Workbench area to proceed past the Welcome Screen.



Figure 2. Close the Welcome Window by Clicking in the Workbench Area

7. Start a new project by clicking the dropdown menu rext to the **New** icon in the Tool Bar.

C/C++ - e2 studio		
File Edit Source Refactor		
<b></b>	←	Tool Bar
<mark>ि New </mark> t Explorer छ		

Figure 3. Start a New Project

8. Select Synergy C/C++ Project from the menu.

	• 🛛 🕞 🔊 • 🍕 •	📑 🗓 🗸 🔻
Ð	Synergy C/C++ Project	
	Project	
	Other	Ctrl+N

Figure 4. Select Synergy C/C++ Project in the Dropdown Menu



### 9. Select Renesas Synergy C Executable project.

Rene	
	sas Synergy C Library Project ibrary Project for Renesas Synergy.
	sas Synergy C++ Executable Project + Executable Project for Renesas Synergy.
	sas Synergy C++ Library Project + Library Project for Renesas Synergy.

Figure 5. Project Type Selection

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

Project name	GUIAnn	
riojectilaria	COMPP	
<b>Use defa</b>	ult location	

### Figure 6. Enter a Project Name

11. On the top right of this page, verify that the **Toolchains** option is set to **GCC ARM Embedded**.

Toolchains
GCC ARM Embedded
IAR ARM Toolchain

Figure 7. Verify GCC ARM Embedded Toolchain



12. Click the **Next** button to continue.

- 13. Under **Device Selection** (top left), select SSP version **2.1.0** (or later).
- 14. For the Board field, select S7G2 DK. The Device field updates automatically.

😨 e2 studio - Project Configuration (Synergy C Executable P	roject) — 🗆 🗙
e2 studio - Project Configuration (Synergy C Executa Select the board support that you require.	ble Project)
Device Selection SSP version: 2.4.0 ~ Board: S7G2 DK ~ Device: R7FS7G27H2A01CBD	Board Details
Select Tools Toolchain: GNU ARM Embedded Toolchain version: 10.3.1.20210824 Debugger: J-Link ARM	<ul> <li>Available Tools</li> <li>GNU ARM Embedded 10.3,1.20210824</li> <li>Debuggers J-Link ARM</li> <li>RTOS Express Logic ThreadX</li> <li>Smart Manual IO Registers Supported Software Manual Supported</li> </ul>
? < Ba	ck Next > Finish Cancel

Figure 8. Device Selection

15. Click the **Next** button to continue.

16. In the **Project Template Selection**, select the option **BSP**.

Project Templa	te Selection
۵	<b>BSP</b> Base Board Support Package for the chosen Synergy family. [Renesas.Synergy.2.4.0.pack]

### Figure 9. Select the BSP



## 17. Click the **Finish** button.

18. If you have not previously directed e<sup>2</sup> studio to remember your perspectives, e<sup>2</sup> studio will display the **Open Associated Perspective?** dialog box. If opened, click **Yes** to acknowledge and close.

e <sup>2</sup> Oper	n Associated Perspective? ×
?	This kind of project is associated with the Synergy Configuration perspective. Do you want to open this perspective now?
<u>R</u> em	ember my decision
	<u>Y</u> es <u>N</u> o

### Figure 10. Open Perspective Dialog Box

When e<sup>2</sup> studio has finished creating the project, it displays the following screen.

le Edit Navigate Search Project	Renesas Views Run Window Help														
🐔 🔅 🔳 🕸 Debug	V 🖸 GUIApp Debug V 🔅	📬 🗕 🗐   🖏	- 🔨	- 🔜	•	💸   🥝	) <del>•</del> *	- 💁	• [ [	ka = \$	€ ⊪	• []]	0 🎼	<b>2</b> d	5 🛛
a • a • ↔ ↔ ↔ • • • •	Ĵ										Q	E	E (	:/C++	戀 :
Project Explorer X	[GUIApp] Synergy Configuration ×	-		न Pac	kage 🗅	<									
		0		1.1		3 4	5 6	7		9 1	0 11	1.00	13		15
GUIApp	Summary	Generate Project Cont	ent	A		P303 VSS	VSS P9		VCCDC		11 P90	1			
> 🗊 Includes				B P109		_									-
> 😕 src	Project Summary	Renesas Synergy™	~	B P109							SS P90:			USBDM P	
> 📇 synergy > 🕞 script	Board: S7G2 DK			C P111	P110	P112 P304	P309 P3			P904 V	SS 9315	5 P205	P207	P409 P	410
> 🦢 synergy_cfg	Device: R7FS7G27H2A01CBD	S7G2		P VCC	VSS	P113 P305	P306 P3	17 F306	P910	¥903 V	CC ¥204	F P413	P412	P411 P	706
configuration.xml	Toolchain: GCC ARM Embedded			E P610	P611	P115 P114	P914 P9	5 P908	P909	P900 P3	13 P414	F P711	P709	P415 1	VSS I
GUIApp Debug.launch	Toolchain Version: 10.3.1.20210824			F P614	P612	P613 P608	P300 P9	16 P907	RES#		10 P712	VSS	heere	USBHSEUS	court I
R7FS7G27H2A01CBD.pincfg S7G2-DK.pincfg	SSP Version: 2.4.0											1US8H	4 <sup>vccus</sup>		
> ⑦ Developer Assistance				G P813	PA15	PA14 P609	PA12 PA	11 PA06			13 PB0		S USBHS	PVSSUSA	VSSUS
	Selected software components			H VCLF	VSS	VCC PA09	PA10 PA	02 PA13	P913	P800 P8	104 PB0	5 VCC	AVCCU	P213 P	212
	SSP Common Code	v2.4.0		J PAG	PA06	PAOS PAO4	PA03 PA	01 PA00	P703	P406 P7	04 ¥80	2 PB05	VSS	хсоит х	
	Clock Generation Circuit: Provides=[CGC]	v2.4.0		K P605	P604	P603 P107	P607 P6	6 F606	¥809	P515 P4	04 270	, Yesos	Pent	VBATT V	1C10 8
	Event Link Controller: Provides=[ELC]	v2.4.0				P500 P106								P706 P	
	Factory MCU Information Module: Provides=			L P602			P811 P8	12 VCC	VSS	P007 P0	03 VSS	vcc			
	I/O Port: Provides=[IO Port]	v2.4.0 v2.4.0		M VSS	VCC	P105 P804	PS05 PS	6 P508	P015	P014 P0	10 P004	P806	P405	¥700 ¥	701
	S7G2_DK Board Support Files Board support package for R7FS7G27H2A0			N P102	P103	P104 P501	P502 P5	17 P510	VREFL	AVSSO PO	11 P008	B FOOZ	P400	P402 P	403
	Board support package for S7G2	v2.4.0		P P101	¥800	P810 P803	¥503 P5	9 VCC	AVCC	VREFL PO	06 P00:	P807	P513	P514 P	401
	Board support package for S7G2	v2.4.0	$\sim$	R P100	P801	P602 P500	¥504 VC	2 VSS		VREFH	09 P005	5 P000	P805	P512 P	~
				1		3 4	5 6		v KEFN	9 1			13		15
	Renesas Synergy						7FS7G275								
	YOU LUDE College			Con	nection st				2		iop vi	<i>cm</i> )			
	Support				Error		Warning	,		ок					
	Summary BSP Clocks Pins Threads Messaging Co	mponents													
Properties 🗙 🛐 Problems 🍓 Sm	art Browser	📑 🕴 🗖 🔀	Pin Co	nflicts	×										
		0 ite							_			_			_
		De	scriptio	n		^				Modu	ıle	P	Pin		L

Figure 11. GUIApp Project



## 4. Configuring the project in e<sup>2</sup> studio

Once successfully created in e<sup>2</sup> studio ISDE, the project can be configured 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 12. Selecting the Configuration.xml File in Project Explorer

2. In the Synergy Configuration Window, click the Threads tab.



Figure 13. Synergy Configuration Threads Tab

### 3. Select the HAL/Common thread.

Threads				
Threads		HAL/Common Stacks		
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	^	g_elc ELC Driver on r_elc	g_ioport I/O Port Driver on r_ioport	g_cgc CGC Driver

### Figure 14. Threads

4. In the HAL/Common Stacks area, click the New Stack button.

Threads 🗿 New	<b>Thread</b> 🔊 Remove	HAL/Common Stacks		New	/ Stack > 📓 Remove
HAL/Common g_ioport I/O Port Driver or g_elc ELC Driver on r_elc g_cgc CGC Driver on r_cg		<ul> <li>g_ioport I/O Port Driver on r_ioport</li> <li>i</li> </ul>	g_elc ELC Driver on r_elc	<pre>   g_cgc CGC Driver on   r_cgc </pre>	<pre>     g_fmi FMI Driver     r_fmi</pre>

Figure 15. Add a Timer Driver Module to the HAL/Common Thread Part 1



### 5. In the New Stack menu, select Driver > Timers > Timer Driver on r\_gpt.

				0	The Aller			
hreads Configuration			Generat	e Project Content	A P302 P303 VSS	5 6 7 8 VSS P905 P911 VCC	9 10 11 VLO VCL1 P90	2 9202
Threads New Thread Remove E	HAL/Common Stacks	🗟 New S	tack > ≜Extend Sta	idc > iii) Remove	= P109 P108 P301 VCC			-
Y 📽 HAL/Common	g_fmi FMI Driver on	g_elc ELC Driver o	Driver	Analog	> 10 P112 P304	P309 P310 P311 P201	1 P904 VSS P31	5 P205 P
g_fmi FMI Driver on r_fmi	r_fmi	g_eic etc briver of	Framework	> Connecti	vity > 55 P113 P305	P306 P307 P308 P910	9903 VCC P20	4 P413 P
g_elc ELC Driver on r_elc g_ioport I/O Port Driver on r_iopo	(D)		X-Ware	> Crypto Graphics	11 P115 P114	P914 P915 P908 P909	P900 P313 P41	4 P711 P
I g_cgc CGC Driver on r_cgc	0	0	Search	Input		P300 P906 P907 RES	P314 P710 P71	2 VSS V
000002299854*1100000000000000000000000000000000000				Monitori	ng > 15 PA14 P609	PA12 PA11 PA08 P615	5 P206 P713 PB0	7 VSS 2USBH
				Power	> SS VCC PAOS	PA10 PA02 PA13 P913	5 PB00 PB04 PB0	6 VCC A
				Storage	> 06 PA05 PA04	PA03 PA01 PA00 P703	5 P405 P704 P80	2 P805
< >>				System	) toput Ca	pture Driver on r_i	ant input can	turo
AL/Common New Object >						pture Driver on r_		
Objects & Remove				Transfer	A PTC Date	er on r_rtc	gpr_mpur_cap	Aure
					and the second second second	iver on r agt		
					H P102 + Timer Dr	iver on r_gpt		
	<			>	P P101 P800 P810 P803	P503 P509 VCC AVCC	VREFL POO6 POO	1 P807 P

Figure 16. Add a Timer Driver Module to the HAL/Common Thread Part 2

6. In the HAL/Common Modules area, select the newly created module g\_timer0 Timer Driver on r\_gpt.

Threads	🗿 New Thread  🔊 Remove	HAL/Common Stacks				🗿 New Stack > 🛛 🙀 Remove
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	* *	g_ioport I/O Port Driver on r_ioport	g_elc ELC Driver on r_elc	g_cgc CGC Driver on r_cgc	g_fmi FMI Driver on r_fmi	<ul> <li>g_pwm_backlight Timer Driver on r_gpt</li> <li>I</li> </ul>

Figure 17. Select the Newly Created Timer Driver Module

7. In the **Properties Window**, change the **Properties** to match those in Figure 18. Hint: Change the channel to 2!

Property	Value
⊿ Common	
Parameter Checking	Default (BSP)
Module g_pwm_backlight Timer Driv	er
Name	g_pwm_backlight
Channel	2
Mode	PWM
Period Value	10
Period Unit	Milliseconds
Duty Cycle Value	100
Duty Cycle Unit	Unit Percent
Auto Start	True
GTIOCA Output Enabled	False
GTIOCA Stop Level	Pin Level Low
GTIOCB Output Enabled	True
GTIOCB Stop Level	Pin Level Low
Callback	NULL
Interrupt Priority	Disabled

### Figure 18. Configure the PWM Module

The next steps add the required software to enable the touch screen and configure the LCD driver.

The touch screen requires several frameworks and drivers to be used. External interrupts are used to know when to read the data. An  $I^2C$  driver handles the reads. A framework translates the register data from the peripheral to touch coordinates the software can use.



8. Create a new thread by clicking the New Thread button in the Threads area.

Threads New Thread	HAL/Common Stacks	🛃 New Stack > 🛛 🙀 Remove
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	<ul> <li>↓ g_ioport I/O Port Driver on r_ioport</li> <li>↓ g_elc ELC Drive</li> <li>↓ g_elc ELC Drive</li> </ul>	r on g_cgc CGC Driver on r_fmi

### Figure 19. Creating a New Thread

- 9. Click on New Thread to pull up the Properties tab.
   10. Edit the Properties to match.
- Properties 23 In \*[GUIApp] Synergy Configuration Generate Project Content New Thread 0 **Threads Configuration** Settings Property Value New Stack > ≜ Extend Stack > ℜ Remove
   🕙 New Thread 🛋 Remove 😑 🛛 Main Thread Stacks Threads ✓ Thread Add Synergy stacks to the selected thread by using the 'New Stack >' toolbar button (above), or by pasting here from the clipboard. A HAL/Common Symbol main\_thread g\_fmi FMI Driver on r\_fmi Name Main Thread ∉ g\_elc ELC Driver on r\_elc Stack size (bytes) 2048 # g\_ioport I/O Port Driver on r\_ioport Priority 6 Enabled @ g\_cgc CGC Driver on r\_cgc Auto start g pwm backlight Timer Driver on r\_gpt Time slicing interval (ticks) 10 Main Thread

### Figure 20. Configuring the Main Thread Properties

11. Back in the Synergy Configuration window > Threads tab > Main Thread Stacks area, click New Stack.

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

Threads	🛃 New Thread	Remove	Main Thread Stacks	Remove
Main Thread		*	Add Synergy stacks to the selected thread by using the 'New Stack >' toolbar button (above), o pasting here from the clipboard.	r by

Figure 21 Main Thread Stacks



12. Add a framework for the touch panel by selecting **New Stack > Framework > Input > Touch Panel V2 Framework on sf\_touch\_panel\_v2**.

hreads Co	onfiguration	Generate Project Content	
Threads	🔄 New Thread 🙀 Remove 🛛	Driver >	в И
✓ ▲ HAL/C # a fi	.ommon mi FMI Driver on r_fmi	Add Synergy stacks to the select Framework Analog	
	oport I/O Port Driver on r_ioport	X-Ware > Audio	
	:gc CGC Driver on r_cgc	Search Connectiv	rity
	elc ELC Driver on r_elc	Crypto	
🙂 g_t 🚳 Main 1	imer0 Timer Driver on r_gpt Thread	File System	n
- 386 IVIGITI	initiad	Graphics	
		Cap Touch Button Framework on sf_touch_ctsu_button      Input	
		Cap Touch Framework on sf_touch_ctsu     LevelX	
		Cap Touch Framework on sf_touch_ctsuv2     Memory	
		Cap Touch Slider/Wheel Framework on sf_touch_ctsu_slider     Networkin	ıg
		External IRQ Framework on sf_external_irq     Services	
		Touch Panel V2 Framework on sf_touch_panel_v2	м
Main Thread (	Objects		N P
			P P
ummary BSP	Clocks Pins Threads Messaging C	mponents	C

Figure 22. Add Touch Panel V2 Framework on sf\_touch\_panel\_v2.

13. In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on g\_sf\_touch\_panel Touch Panel V2 Framework sf\_touch\_panel\_v2. Configure the properties for g\_sf\_touch\_panel Touch Panel V2 Framework sf\_touch\_panel\_v2.

	O Generate Project Content	g_sf_tou	ch_panel_v2_0 Touch Panel V2 Framework on sf_touch_panel_v2	
g_sf_touch_panel_v2_0 Touch Panel V2 Framework on sf_touch_panel_v2 Stacks	New Stack > API Info API Info Property ✓ Common Parameter Common			Value Default (BSP)
ranetoric on si_touch_panel_it statio	Remove		Thread Stack Size	512
A ( ) 1 1 1 1			✓ Module g_sf_touch_panel_v2_0 Touch Panel V2 Framework on sf_touch_panel_v2	
g_sf_touch_panel Touch Panel V2 Framework on			Name	g_sf_touch_panel
sf touch panel v2			Thread Priority	3
			Update Hz	10
•			Touch Coordinate Rotation Angle(Clockwise)	0
S Add Touch Driver			Name of generated initialization function	sf_touch_panel_v2_init
had local piller			Auto Initialization	Enable
			Auto Start	Enable
			Name of touchpanel callback function to be defined by user	touch_panel_v2_callback

Figure 23. Configure the Touch Panel V2 Framework Properties



# 14. In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on Add Touch Driver > New > Touch\_panel\_chip\_sx8654.



Figure 24. Add the Touch\_panel\_chip\_sx8654 Touch driver

15. Configure the Touch\_panel\_chip\_sx8654 properties as shown.

Threads Configuration			Generate Project Cor	ent	g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx8654			
Threads	🐑 New Thread 🔏 Remove 🗈	€ New Stack >		Settings	<ul> <li>Module g touch panel chip sx86</li> </ul>			
✓	Framework on sf_touch_par			Name Horizontal pixel count	g_touch_panel_chip_sx8654 480			
g_elc ELC Driver on r_elc g_loport I/O Port Driver g_fmi FMI Driver on r_fr g_timer() Timer Driver o g_timer() Timer Driver o g_timer()	on r_icport mi	g_sf_touch_panel Touch sf_touch_panel_v2	Panel V2 Framework on		Vertical pixel count Reset Port Reset Pin	272 07 11		
* g_sf_touch_panel Touch Panel V2 Framework on sf_touch_panel_v2		(♥ g_touch_panel_chip_sc6654_0 Touch_panel_chip_sc6654 ©						
		g_sfj2c_device012C     Framework Device on     sfj2c     ©	g_sf_external_ing0     External IRQ     Framework on     sf_external_ing					
¢		g_sf_i2c_bus0 12C Framework Shared Bus on sf_i2c	g_external_inq0 External iRQ Driver on r_icu					
g st touch panel louch Panel on sf_touch_panel_v2 Objects	V2 Framework C New Dojett > 10 Remove	Add I2C Driver	0					

Figure 25. Configure Touch\_panel\_chip\_sx8654 Properties



The Synergy Configurator has created the external IRQ framework and driver and has a placeholder for the I<sup>2</sup>C driver and Touch driver.

The Touch Panel V2 Framework module scans data from a touch controller and invokes the user registered touch panel callback when a touch event occurs. (If the user callback is not registered, the sf\_touch\_panel\_v2\_api\_t::touchDataGet API function can be used to retrieve the data). The **SF External Interrupt** is a framework layer used by the **Touch Controller Driver**.

	Generate Project Content	g_sf_ext	ernal_irq0 External IRQ Framework on sf_external_irq	
sf touch panel Touch Panel V2		Settings	Property	Value
ramework on sf_touch_panel_v2 Stacks	ack > ≜ Extend Stack > 🔊 Remove	API Info	✓ Common	
			Parameter Checking	Enabled
g_sf_touch_panel Touch Panel V2 Framework on sf_to	uch_panel_v2		<ul> <li>Module g_sf_external_irq0 External IRQ Framework on sf_external_irq</li> </ul>	
			Name	g_sf_external_irq0
_			Event	Semaphore Put
0				
T				
g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx865	54			
<b>A</b>				
g_sf_i2c_device0 I2C Framework Device on sf_i2c	⊕ g_sf_external_irg0			
320,202,000,000,000,000,000,000,000,000,	External IRQ			
	Framework on			
0	f_external_irq			
- -	· · · · · · · · · · · · · · · · · · ·			
g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c	g_external_irq0 External			
	IRQ Driver on r_icu			
3	1			
<u>*</u>				

Figure 26. Configure the properties for External IRQ Framework Stack

16. Select **External IRQ Driver on r\_icu**. Configure the properties for the new module as shown.

	Settings	Property	Value
g_sf_touch_panel Touch Panel V2 Tramework on sf touch panel v2 Stacks New Stack > ≜ Extend Stack > № Remove Extend Stack > № Remove Rem	API Info	✓ Common	
ramework on sr_toucn_panel_vz_stacks		Parameter Checking	Enabled
g_sf_touch_panel Touch Panel V2 Framework on sf_touch_panel_v2		<ul> <li>Module g_external_irq0 External IRQ Driver on r_icu</li> </ul>	
		Name	g_external_irq0
		Channel	7
0		Trigger	Falling
<u> </u>		Digital Filtering	Enabled
g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx8654		Digital Filtering Sample Clock (Only valid when Digital Filtering is Enab	PCLK / 64
		Interrupt enabled after initialization	True
		Callback	NULL
•		Pin Interrupt Priority	Priority 3
⊕ g_sf_i2c_device0 I2C Framework Device on sf_i2c     ⊕ g_sf_external_irq0     External IRQ     Framework on     ③ sf_external_irq			
<sup>®</sup> g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c <sup>®</sup> g_external_irq0 External IRQ Driver on r_icu <sup>®</sup> 0 <sup>®</sup> 0			

Figure 27. Configure the Properties for IRQ Driver on r\_icu



### In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on g\_sf\_i2c\_device0 I2C Framework Device on sf\_i2c. Configure the properties for g\_sf\_i2c\_device0 I2C Framework Device on sf\_i2c.

	Generate Project Content	g_sf_i2c	_device0 I2C Framework Device on sf_i2c	
		Settings	Property	Value
g_sf_touch_panel Touch Panel V2 € New Sta Framework on sf_touch_panel_v2 Stacks	ack > ≜ Extend Stack > ▲ Remove	API Info	✓ Common	
Hallework on si_touch_panel_vz_stacks			Parameter Checking	Enabled
g_sf_touch_panel Touch Panel V2 Framework on sf_touch	uch_panel_v2		<ul> <li>Module g_sf_i2c_device0 I2C Framework Device on sf_i2c</li> </ul>	
			Name	g_sf_i2c_device0
			Slave Address	0x48
0			Address Mode	7-Bit
<ul> <li>         ⊕ g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx865         _0         </li></ul>				
g_sf_i2c_device0 I2C Framework Device on sf_i2c	g_sf_external_irq0 External IRQ Framework on			
0	<pre>① sf_external_irq</pre>			
	⊕ g_external_irq0 External IRQ Driver on r_icu			
Ø	0			
<u>+</u>				

Figure 28. Configure the Properties for g\_sf\_i2c\_device0 I2C Framework Device on sf\_i2c.

 In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click g\_sf\_i2c\_bus0 I2C Framework Shared Bus on sf\_i2c. Configure the properties for g\_sf\_i2c\_bus0 I2C Framework Shared Bus on sf\_i2c

Generate Project Content	g_sf_i2c_	bus0 I2C Framework Shared Bus on sf_i2c	
sf_touch_panel Touch Panel V2	Settings API Info	Property V Module g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c Name	Value g_sf_i2c_bus0
g_sf_touch_panel Touch Panel V2 Framework on sf_touch_panel_v2			
g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx8654			
g_sf_i2c_device0 I2C Framework Device on sf_i2c			
s sexternal_irq			
g_sf_i2c_bus0 12C Framework Shared Bus on sf_i2c   g_external_irq0 External IRQ Driver on r_icu			
•			

Figure 29. Configure g\_sf\_i2c\_bus0 I2C Framework Shared Bus on sf\_i2c Properties



#### 19. In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on Add I2C Driver > New > I2C Master Driver on r\_sci\_i2c



Figure 30. Add I2C Master Driver on r\_sci\_i2c



# 20. In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on I2C Master Driver on r\_sci\_i2c. Configure the properties for I2C Master Driver on r\_sci\_i2c

of south and Touch Percentary and the second south of the second southow of the second south of the second south of the second	Settings	Property	Value
sf_touch_panel Touch Panel V2 amework on sf touch panel v2 Stacks amework on sf touch panel v2 Stacks	Remove API Info	✓ Common	
anework on si_touch_paner_vz_stacks		Parameter Checking	Enabled
g_sf_touch_panel Touch Panel V2 Framework on sf_touch_panel_v2		<ul> <li>Module g_i2c I2C Master Driver on r_sci_i2c</li> </ul>	
		Name	g_i2c
		Channel	7
0		Rate	Fast-mode
A	1	Slave Address	۵ 0
g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx8654		Address Mode	la 7-Bit
		SDA Output Delay (nano seconds)	300
D		Bit Rate Modulation Enable	Enable
		Callback	NULL
<sup>₱</sup> g_sf_i2c_device0 I2C Framework Device on sf_i2c <sup>●</sup> g_sf_external_irq0	1	Receive Interrupt Priority	Priority 3
g_si_ize_deviceo ize rialitettorit Device ori si_ize		Transmit Interrupt Priority	Priority 3
Framework on		Transmit End Interrupt Priority	Priority 3
() sf_external_irq			
↑ ↑			
g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c			
IRQ Driver on r_icu			
0 0			
A	·		
g_i2c I2C Master Driver on r_sci_i2c			
D			
×			
⊕ g_transfer0 Transfer ⊕ g_transfer1 Transfer			
Driver on r_dtc SCI7 TXI Driver on r_dtc SCI7			
RXI			
0			

Figure 31. Configure the Properties of Master Driver on r\_sci\_i2c

21. In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on g\_transfer0 Transfer Driver on r\_dtc SCI7 TXI and configure the properties for g\_transfer0 Transfer Driver on r\_dtc SCI7 TXI

	fer0 Transfer Driver on r_dtc SCI7 TXI	
Settings API Info	Property Common Brameter Checking	Value
		Enabled Disabled .ssp_dtc_vector_table g_transfer0 @ Normal @ 1 Byte @ Fixed @ Incremented @ Source @ After all transfers have completed @ NULL @ NULL @ NULL @ 0 @ 6 @ C17 TX1 @ False @ NULL Disabled
		API Info         Common           Parameter Checking         Software Start           Linker section to keep DTC vector table         Module g_transfer0 Transfer Driver on r_dtc SCI7 TXI           Name         Mode           Transfer Size         Destination Address Mode           Source Address Mode         Source Address Mode           Aubort Finasfers         Number of Transfers           Number of Transfers         Number of Blocks (Valid only in Block Mode)           Activation Source (Must enable IRQ)         Auto Enable           Callback (Only valid with Software start)         Callback (Only valid with Software start)

Figure 32. Configure the Properties of g\_transfer0 Transfer Driver on r\_dtc SCI7 TXI



22. In the Synergy Configuration window > Threads tab > Main Thread Stacks area, click on g\_transfer1 Transfer Driver on r\_dtc SCI7 RXI and configure the properties for g\_transfer1 Transfer Driver on r\_dtc SCI7 RXI

	Generate Project Content		Property	Value
sf_touch_panel Touch Panel V2	tack > ≗ Extend Stack > 🔊 Remove	Settings	roperty ✓ Common	value
amework on sf_touch_panel_v2 Stacks		API Info	Parameter Checking	Enabled
g_sf_touch_panel Touch Panel V2 Framework on sf_tc	with papel v2		Software Start	Disabled
g_si_couci_panel loadi ranel vz rianiework of si_co	aci_panei_vz		Linker section to keep DTC vector table	.ssp_dtc_vector_table
			✓ Module g_transfer1 Transfer Driver on r_dtc SCI7 RXI	
0			Name	g_transfer1
<u>+</u>			Mode	Normal
g_touch_panel_chip_sx8654_0 Touch_panel_chip_sx86	54		Transfer Size	🙆 1 Byte
			Destination Address Mode	Incremented
			Source Address Mode	Fixed
•			Repeat Area (Unused in Normal Mode)	Destination
⊕ g_sf_i2c_device0 I2C Framework Device on sf_i2c	<sup>⊕</sup> g_sf_external_irq0		Interrupt Frequency	After all transfers have complete
<ul> <li>g_st_lzc_device0 tzc_Framework Device on st_lzc</li> </ul>	External IRQ		Destination Pointer	NULL
	Framework on		Source Pointer	NULL
0	sf_external_irg		Number of Transfers	۵ 0
1			Number of Blocks (Valid only in Block Mode)	۵ 0
g_sf_i2c_bus0 I2C Framework Shared Bus on sf_i2c	g_external_irq0 External		Activation Source (Must enable IRQ)	SCI7 RXI
	IRQ Driver on r_icu		Auto Enable	False
			Callback (Only valid with Software start)	NULL
0	0		ELC Software Event Interrupt Priority	Disabled
A	1			
g_i2c I2C Master Driver on r_sci_i2c				
•				
×	4			
⊕ g_transfer0 Transfer     ⊕ g_transfer1 Transfer				
Driver on r_dtc SCI7 TXI Driver on r_dtc SCI7				
RXI				
0 0				

Figure 33. Configure the Properties of g\_transfer1 Transfer Driver on r\_dtc SCI7 RXI

### 23. Under Main Thread Stacks, select New Stack, and then X-Ware > GUIX > GUIX on gx.



Figure 34. GUIX on gx



Notice that the Synergy Configurator has now already created the **GUIX Port on sf\_el\_gx framework**, **Display Driver**, **JPEG decoder**, and **D/AVE hardware accelerator drivers** as shown in the following figure.



Figure 35. GUIX on gx

24. Select **GUIX on gx** and configure the **Properties** as the following figure shows.

🔲 Proj	perties 🔀 🛐 Problems	
Proper	ty	Value
⊿ Cor	nmon	
	Enable Synergy 2D Drawing Engine Support	Yes
	Enable Synergy JPEG Support	Yes

Figure 36. GUIX on gx Properties



### 25. Add JPEG common to the Decode Driver on r\_jpeg\_decode.



Figure 37. JPEG Common Module

### 26. Select **GUIX Port on sf\_el\_gx** and configure the properties as shown.

Property	Value
Common	
Parameter Checking	Default (BSP)
Module g_sf_el_gx0 GUIX Port on sf_el_gx	
Name	g_sf_el_gx
Display Driver Configuration Inheritance	Inherit Graphics Screen 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 enable	ed) 1000
Memory section for GUIX Canvas Buffer	sdram
Memory section for JPEG Work Buffer	sdram

Figure 38. GUIX Port on sf\_el\_gx Properties



## 27. Select the **JPEG Decode Driver on r\_jpeg** and configure the interrupt properties as shown. Note that Priority 3 is just an arbitrary number.

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	B 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)

### Figure 39. JPEG Decode Driver on r\_jpeg Properties

## 28. Under Main Thread Stacks, select D/AVE 2D Port on sf\_tes\_2d\_drw and configure the properties as shown.

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

### Figure 40. D/AVE 2D Port Properties

29. Under **Main Thread Stacks**, select **Display Driver on r\_glcd** and configure the Interrupt Properties as shown.

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	Disabled

#### Figure 41. Interrupt Properties

### 30. Configure the **Graphics Screen 1** properties as shown.

Property	Value
a Common	
Parameter Checking	Default (BSP)
Module g_display0 Display Driver on r_glcd	
Name	g_display
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	sdram
Input - Graphics screen1 input horizontal size	480
Input - Graphics screen1 input vertical size	272
Input - Graphics screen1 input horizontal stride(not bytes but pixels	s) 480
Input - Graphics screen1 input format	16bits RGB565

Figure 42. Graphics Screen 1 Properties



### 31. Configure the **Output** properties as shown.

Output - Horizontal total cycles	582
Output - Horizontal active video cycles	480
Output - Horizontal back porch cycles	43
Output - Horizontal sync signal cycles	41
Output - Horizontal sync signal polarity	Low active
Output - Vertical total lines	286
Output - Vertical active video lines	272
Output - Vertical back porch lines	12
Output - Vertical sync signal lines	10
Output - Vertical sync signal polarity	Low active
Output - Format	16bits RGB565

### Figure 43. Output Screen 2 Properties

32. Configure the TCON pins and clock as shown.

TCON - Hsync pin select	LCD_TCON1
TCON - Vsync pin select	LCD_TCON2
TCON - DataEnable pin select	LCD_TCON0
TCON - Panel clock division ratio	1/16

### Figure 44. TCON Settings

- 33. Save the project by pressing Ctrl + s on the keyboard.
- 34. Click the Generate Project Content button to update the project files.

# Generate Project Content

### Figure 45. Generate Project Content

- 35. Close the Synergy Configuration window.
- 36. Open Windows Explorer and go to the directory where you put the files included with this application note. Locate the file <code>Source Files\ R7FS7G27H2A01CBD.pincfg</code>. Now drag the file from the Windows Explorer Window into the <code>GUIApp</code> root directory inside the **e<sup>2</sup> studio Project Explorer** window.
  - A. When asked how to import the selected files, click **OK** to copy the files.
  - B. When asked if you want to overwrite, click Yes.
  - Note: This file contains the pin configuration for the DK-S7G2 Synergy MCU.
- 37. In the Synergy Configuration window, under the Pins tab Select the Import the pin configuration, as shown.

Figure 46. Synergy Pin Configuration



38. Click **File System** to access the pin configuration given in the source file as shown.

e <sup>2</sup>	Import Pin	Configurati	ions from File	>	
File:					
?	ОК	Cancel	Workspace	File System	11

### Figure 47. Importing Pin Configuration

39. Select the pin configuration in the source file and press **Open** as shown. Click **OK** to import the pin configurations.

Name	Date modified	Туре
R7FS7G27H2A01CBD.pincfg	18-05-2017 12:45	PINCFG File
۲		
< File <u>n</u> ame: R7FS7G27H2A01CBD.pincfg	✓ *.pincfg	
	✓ *.pincfg	v Cancel

### Figure 48. Synergy Configuration

40. If you get the following error message when importing the pin configuration file, click **OK**.



Figure 49. Conflict Found Error Box



## 41. Once you select the reference pin configuration, you will get a pin error message, **Pin Dangling**. This means two functions are using the same pin. As shown, the error is in the **Peripherals**.

	LCD_DATA20:	None
E ··· Other Pins	LCD_DATA21:	None
	LCD_DATA22:	None
	LCD_DATA23:	None
	LCD_TCON0:	✓ P315
	LCD_TCON1:	✓ P314
	LCD_TCON2:	≅ *P313

### Figure 50. Selecting Pin Configuration

### 42. Click Peripherals, then Storage:SDHI. Select SDHI0 and go to DAT7 to change it to None as shown.

Graphics:PDC ⊕… ✓ Storage:QSPI	DAT7:	None	
⊡ ✓ Storage:SDHI	CD:	✓ P903	$\Rightarrow$
SDHI1	WP:	✓ P414	

Figure 51. Making P313 Usable

The next steps show how to configure pins of the S7G2 MCU to control the LCD panel and touch screen on DK-S7G2 MCU. Proceed to Step 43 to skip this optional informational section.

The **Timer Driver on r\_gpt** is used to configure the peripheral as a PWM to control the backlight level using a hardware pin on the S7G2 Synergy MCU. For the DK-S7G2 MCU, the pin that controls the backlight for the LCD is located on P7\_12, as shown from the following snippet showing the DK-S7G2 MCU breakout board schematic.

	33
P7 10 LCD ON	35
PA 3/SCL7	37
P7 12 LCD BLEN	39

Figure 52. LCD Backlight Pin

Since an existing pin configuration is being used, it is not necessary to set this pin up using the pin configurator. If you are interested, follow the steps below to see how it was configured.

#### 43. Select the **Pins** tab in the **Synergy Configuration Window**.



Figure 53. Configuration Pins



### 44. Expand Ports and P7 to show the port 7 pins.

⊿ ✓ Ports
⊳ P0
▶ ✓ P1
⊳ ✓ P2
⊳ ✓ P3
⊳ ✓ P4
⊳ ✓ P5
N 2 P6
a 🗸 P7
✓ P700
✓ P701
✓ P702
✓ P703
✓ P704
✓ P705
✓ P706
✓ P707
P708
P709
✓ P710
✓ P711
✓ P712
✓ P713

Figure 54. Port 7 Pins

45. Select **P712** to show the options for this pin.

n Selection	Pin Configur	ition	
/pe filter text	Ē		ii   🔁
✓ P700	<ul> <li>Module nan</li> </ul>	e: P712	
✓ P701 ✓ P702	Symbolic na	me: GPT02_PIN_OPTION_B_GTIOC2B_B	
✓ P702	Comment:		*
✓ P704			*
✓ P705	P712 Config	uration	
✓ P706 ✓ P707	Mode:	Peripheral Mode 🗸	
P708	Pull up:	None	
P709 ✓ P710	Drive Capac	tv:	
0711			
✓ P712	Output type	CINOS	
✓ P713	Chip input/	utput	
⊳ ✓ P8 ⊳ ✓ P9	P712;	✓ GTIOC2B_B	

Figure 55. Pin Configuration for P7\_12

- Module name: Selected GPT2.
- Pin Group Selection: Selected **Mixed**
- Operation Mode: Selected channel GTIOCA or GTIOCBoutput type: Only selected channel GTIOCB for the pin P712. This option changes based on the mode setting. In this case, GTIOC2B\_B is selected to use as the Timers:GPT 2 B output.
- (Hint: Disable pin P712 first before select and setting **Timers:GPT ; GPT2**.

Pins can also be configured using the peripheral as a starting point.

This view shows the pins that are available for different functions.



Pin Selection		Pin Configuration			
type filter text 🖉					
GPT0 GPT1	^	Module name:		GPT2	
GPT2		Pin Group Selection:		Mixed	~
GP13		Operation Mode:		GTIOCA or GTIOCB	~
GPT4		operation model		onock of onoco	
GPT5		Input/Output			
GPT6		GTIOCA:		None	~
GPT7		diloca:	_	None	*
GPT8		GTIOCB:	~	P712	~
GPT9					
GPT10					
GPT11					
GPT12					
GPT13					
Timer:OPS					

### Figure 56. GPT02 Pin Configuration

In most cases, after you enable and select a pin, it is automatically configured. This pin can be configured by pressing in, which shows the following screen.

The interrupt for the touch controller is located on pin **P0\_1** as seen in the breakout board schematic.

38	TOUCH IRQ#	P0 1
40	LCD RESET#	P7 13

### Figure 57. Touch IRQ

The touch panel pin is configured as an IRQ in the **Pin Configuration** window.

Pin Selection		Pin Configuration	
type filter text			
⊿ ✓ Ports		Module name:	P001
⊿ ✓ P0 ✓ P000		Symbolic name:	GPIO9
✓ P001		Comment:	
✓ P002			
P003 P004	=	Note:	INPUT ONLY pin. DO NOT set to Output Mode! (Only Input Mode or Analog Mode are allowed.)
P005			(only input wode of Analog wode are allowed.)
P006		P001 Configuration	
P007		Mode:	Input mode 🔹
P008		IRQ:	IRQ7_DS 🗸
P009			
P010 P011		Chip input/output	
		P001:	GPIO 👻
P014			
P015			

### Figure 58. P0\_01 Pin Configuration

For the LCD board schematic, see the touch controller, which is the SX8656.





Figure 59. Touch Controller

The following figure shows the touch controller's reset pin is located on P07\_11. To make use of this function, set the pin up as a GPIO output.

Pin Selection		Pin Configuration	
type filter text			
⊳ ✓ P3	^	Module name:	P711
⊳ ✓ P4		Symbolic Name:	GPIO8
⊳ ✓ P5 ⊳ ✓ P6		Comment:	
⊿ ✓ P7			
✓ P700		Port Capabilities:	CTSU0: TS15
✓ P701 ✓ P702			ETHERCO: TX_CLK SCI1: CTS_RTS_SS
✓ P702 ✓ P703		P711 Configuration	
✓ P704		Mode:	Output mode (Initial Low)
✓ P705			
✓ P706 ✓ P707		Pull up:	None 🗸
P707		Drive Capacity:	Low
P709		Output type:	CMOS v
✓ P710			
✓ P711		Chip input/output	
✓ P712 ✓ P713		P711:	✓ GPIO 🗸

Figure 60. P7\_11 Pin Configuration

The SCI driver can be configured for different serial communication protocols for the DK-S7G2 pin PA\_3 and PA\_2 are used to handle the  $I^2C$  functionality.

P7 10	LCD ON	T	35		36	I		PA 2/SDA7
PA 3/SCL7			37		38		TOUCH IRQ#	P0 1
P7 12	LCD BLEN		39		40		LCD RESET#	P7 13
		-						

### Figure 61. I<sup>2</sup>C Pins

The pins are configured in the Pin Configurator under the peripheral section.



Pin Selection		Pin Configuration			
type filter text					
<ul> <li>▷ Analog:DAC12</li> <li>▷ Connectivity:CAN</li> <li>▷ ✓ Connectivity:ETHERC</li> <li>▷ Connectivity:IIC</li> </ul>	^	Module name: Usage:	open drain.	e I2C mode, ensure port pins output type etween I2C and other modes, first disable	
✓ Connectivity:SCI SCI0 SCI1 SCI2		Pin Group Selection: Operation Mode:	Mixed Custom	~	
SCI3 SCI4 SCI5 SCI6	- 1	Input/Output TXD_MOSI: RXD_MISO:	PA02	~	
✓ SCI7 ✓ SCI8 SCI9		SCK: CTS_RTS_SS:	None	~ ~	
<ul> <li>Connectivity:SPI</li> <li>Connectivity:SSI</li> </ul>		SDA:	None	• •	
▷ ✓ Connectivity:USBFS		SCL:	None	¥	

Figure 62. SCI7 Configuration



The LCD pin configuration is based on Option B for the  $g\_lcd$  controller as seen in the pin configurator below.

<sup>P</sup> in Sel	ection	Pin Configuration			
type fi	lter text 🧟 🗄 🖽				
▷ ✓	Ports			L	
4 -	Peripherals	Input/Output			
Þ	Monitoring:CAC	LCD_CLK:	~	P900	
⊳	✓ Analog:ADC	LCD_DATA00:	1	P804	
	Analog:CMP Analog:DAC12		*		
		LCD_DATA01:	~	P803	0
Þ	<ul> <li>Connectivity:ETHERC</li> </ul>	LCD_DATA02:	-	P802	
$\triangleright$	Connectivity:IIC	LCD_DATA03:	~	P606	
⊳	<ul> <li>Connectivity:SCI</li> </ul>	Construction and a construction of the	*		
$\triangleright$	Connectivity:SPI	LCD_DATA04:	~	P607	
⊳	Connectivity:SSI Connectivity:USBFS	LCD_DATA05:	~	PA00	0
	<ul> <li>Connectivity:03BF3</li> <li>Connectivity:03BHS</li> </ul>	LCD_DATA06:		PA01	
Þ	Input:CTSU		*	PAUI	
⊳	Input:IRQ	LCD_DATA07:	*	PA10	
$\triangleright$	Input:KINT	LCD_DATA08:	~	PA09	
4	<ul> <li>Graphics:GLCDC</li> </ul>	LCD_DATA09:	-	PA08	0
	GLCDC0 Graphics:PDC		*		
	<ul> <li>Storage:QSPI</li> </ul>	LCD_DATA10:	~	P615	
Þ	Storage:SDHI	LCD_DATA11:	~	P905	0
Þ	<ul> <li>System:BUS</li> </ul>	LCD_DATA12:		P906	
⊳	System:CGC	LCD_DATATZ:	*	P900	
$\triangleright$	<ul> <li>System:DEBUG</li> </ul>	LCD_DATA13:	*	P907	0
⊳	System:TRACE Timer:AGT	LCD_DATA14:	1	P908	
	✓ Timer:GPT			P901	
Þ	Timer:OPS	LCD_DATA15:	~	P901	
Þ	Timer:POEG	LCD_DATA16:		None	
Þ	Timer:RTC	LCD_DATA17:		None	0
$\triangleright$	Other Pins	LCD_DATA18:		None	
		LCD_DATATO:			
		LCD_DATA19:		None	
		LCD_DATA20:		None	
		LCD_DATA21:		None	
		LCD_DATA22:		None	
		LCD_DATA23:		None	
		LCD_TCON0:	~	P315	
		LCD_TCON1:	~	P314	
		LCD_TCON2:	*	P313	
		LCD_TCON3:		None	.)
		LCD_EXTCLK:		None	
		-			

Figure 63. GLCD Pin Option B



The breakout board schematic shows the full list of pins. The DK-D7G2 MCU uses a 16-bit LCD interface.



Figure 64. LCD Connector Pin Out

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



Figure 65. Generate Project Content

- 48. Open **Windows Explorer** and find 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**<sup>2</sup> **studio Project Explorer** window.
  - A. When asked how to import the selected files, click **OK** to copy the files.
  - B. When asked if you want 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.

## 5. Creating the GUIX Interface using GUIX Studio

Now that the base project has been set up, GUIX components can be added.



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



Figure 66. Creating a New Folder

2. Create another new folder named **guix\_studio** in the root folder of the project by right clicking on **GUIApp** and selecting **New > Folder**. The final folder layout should look like the following figure.



Figure 67. Final Folder List

3. Open GUIX Studio by clicking the desktop icon. The version of GUIX Studio must be 6.1.8.0 or later.



Figure 68. Start GUIX Studio

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

Create New Project...

Figure 69. Create New Project



### 5. Name the project **guiapp**.

**WARNING:** Filenames will be generated by appending names to the project name. You must be careful to make names case sensitive when you define your project name. Later, when files are added to the project, it's assumed that you have called this GUIX project **guiapp**.

 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\GUI APP\GUIApp\guix studio.

Create N		
Project Name	guiapp	
Project Path	C:\Users\[User]\e2_studio\workspace\GUIApp\guix_studio	Browse

Figure 70. 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	browse
Header Files	\src\gui	browse
Resource Files	\src\gui	browse

Figure 71. Correct the file locations

Caution: 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 and GUIX Library version 6.1.8 to .4.1.

Target CPU Renesa	as Synergy	✓ Advanced Settings	
Toolchain GNU		✓ big endian	
Additional Headers			Insert Before
Number of Displays	1	GUIX Library Version	5 • . 1 • . 8 •
		Ma	ajor Minor Patch

Figure 72. Target and GUIX version settings



- 11. Click the Advanced Settings button. A dialog window appears.
- 12. Set **Enable 2D Drawing Engine** to enable the graphics accelerator and **Hardware JPEG Decoder** as shown.

🗘 Synergy	Advanced Setting	s 🗙	
Enable 2	D Drawing Engine		
Runtime Ir	nage Decoder		
JPEG:	Hardware JPEG Decod	ler v	
PNG:	None	~	
Cancel		Save	

Figure 73. Synergy Advanced Settings

### 13. Click Save.

14. Set up the **Display Configuration** as shown.

Display Configuration Display Number 1	Name display_1	
× resolution 480	pixels y resolution 272	pixels
() 1 bpp	grayscale	5:5:5 format
2 bpp 0 4 bpp	invert polarity	4:4:4:4 format
0 4 bpp 0 8 bpp	reverse byte order	3:3:2 format
16 bpp	packed format	
○ 24 bpp ○ 32 bpp	rotated orientation	allocate canvas memory

### Figure 74. Configure the Display

- 15. Click **Save** to generate the project.
- 16. Right-click on **display\_1** in the project view.
- 17. Select Insert > Window > Window.

🔓 GUIApp					
disp	Configure Proje Configure Them				
	Generate All Ou Generate Resou Generate Specit	rce Files			
	Insert	F	Window	•	Window
			Button	۱.	Vertical List
			Text	•	Horizontal List
			Indicator	•	Drop Down List

Figure 75. New Window



18. Modify the properties by selecting the new window and editing the Properties View. Update the current settings to match those shown.

Propertie	es View –
Widget Type	window
Widget Name	window 1
Widget Id	ID_WINDOW1
User Data	
Left	0
Тор	0
Width	480
Height	272
Border	No Border 🗸
Transparent	
Draw Selected	
Enabled	
Accepts Focus	<b>V</b>
Runtime Allocate	
Normal fill	WINDOW_FILL -
Selected fill	SELECTED_FILL -
Draw Function	
Event Function	window1_handler
Wallpaper	None
Tile Wallpaper	

### Figure 76. Configure Window1 Properties

- 19. In the **Project View Window**, right click on **display\_1** and create another window by selecting **Insert > Window > Window**.
- 20. Modify the properties to match the following figure.

Properties	- View
Widget Type	window
Widget Name	window2
Widget Id	ID_WINDOW2
User Data	
Left	0
Тор	0
Width	480
Height	272
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 77. Configure Window2 Properties



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



Figure 78. Add a New Text Button

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



Figure 79. Add a New Checkbox



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



Figure 80. Adding New Prompt

- 24. In the Project View, right-click on window1 and insert another Text Prompt.
- 25. In the Project View, right-click on window2 and insert a Text Prompt.
- 26. In the **Project View**, right-click on **window2** and insert another T**ext Prompt**. After you have followed these directions, your Project View should look like the following screen.

🕂 Project View –
guiapp
display_1
T prompt_2
□ prompt_3
T button
checkbox
T prompt

### Figure 81. GUIX Project View

27. Press the '+' character on right of the </>
Strings to expand the Strings menu.

G Colors	+
T Fonts	+
Pixelmaps	•
Strings	

### Figure 82. Strings Button

- 28. Double-click on any of the strings to open the String Table Editor.
- 29. Delete the existing strings by selecting them, then click the **Delete String** button in the **String Table Editor**.
- 30. Add the strings using the  $\boldsymbol{Add}$   $\boldsymbol{String}$  button as shown.



StringId 🔻	English 🔻 📢					
HELLO_WORLD	Hello World -> Press anywhere to go to window 1					
CHECKBOX_TEXT	Press Me!					
BUTTON_DISABLED	Stay in window1					
BUTTON_ENABLED	Goto window2					
INSTRUCT_CHECKBOX	Press to activate (blue), press "Press me" for more.					
WINDOW1	Window 1					
WINDOW2	Window 2					
INSTRUCT BUTTON	Press the Goto window2 button to show the next screen.					

### Figure 83. New Strings

- 31. When correct, click the **Save** button.
- 32. In the **Project View** under window1, click on the button, then modify the properties in the **Properties** View to match the following figure.

<b>Properties</b>	View	-
Widget Type	text_button	
Widget Name	windowchanger	
Widget Id	ID_WINDOWCHANGER	
User Data		
Left	30	
Тор	30	
Width	180	
Height	50	
Border	No Border	•
Transparent		
Draw Selected		
Enabled	V	
Accepts Focus	V	
Runtime Allocate		
Normal fill	BTN_LOWER	•
Selected fil	BTN_UPPER	-
Draw Function		
Event Function		
Pushed		
Toggle		
Radio	[7]	
Auto Repeat		
String ID	BUTTON_DISABLED	
Text	Stay in window 1	
Font	BUTTON	•
Text Align	Center	•
Normal Text Color	BTN_TEXT	
Selected Text Color	BTN_TEXT	-

Figure 84. Configure Windowchanger Button Properties


33. In the **Project View** under **window1**, click the checkbox, then modify the properties in the **Properties View** to match the following figure.

<b>Properties</b>	View	i esta
Widget Type	checkbox	
	buttonenabler	
Widget Id	ID_BUTTONENABLER	
User Data		
Left	350	
Тор	30	
Width	115	
Height	50	
Border	No Border	-
Transparent		
	<b>v</b>	_
	<ul> <li>Image: A start of the start of</li></ul>	
	[7]	
Normal fill	BTN_LOWER	-
Selected fill	BTN_UPPER	•
Draw Function	_	
Event Function		
Pushed	[m]	
Toggle	<b>V</b>	
Radio	[]]	
Auto Repeat	and a	
String ID	CHECKBOX_TEXT	-
Text	Press Me!	
Font	BUTTON	•
Text Align	Left	-
Normal Text Color	BTN TEXT	•
Selected Text Color	BTN_TEXT	
Unchecked Pixelmap		
Checked Pixelmap	CHECKBOX_ON	
Unchecked Disabled	None	•
Unchecked Disabled	None	-

Figure 85. Configure Buttonenabler Checkbox Properties



34. In the **Project View** under **window1**, click the prompt, then modify the properties to match the window below.

Propertie	es View –
Widget Type	prompt
	instructions
Widget Name Widget Id	
	ID INSTRUCTIONS
User Data	
Left	15
Тор	120
Width	450
Height	80
Border	No Border 🔹
Transparent	
Draw Selected	
Enabled	<b>V</b>
Accepts Focus	
Runtime Allocate	
Normal fill	WIDGET_FILL -
Selected fill	SELECTED_FILL
Draw Function	
Event Function	
String ID	INSTRUCT_CHECKBOX -
Text	Press to activate (blue), press "Press me" fo
Font	PROMPT -
Text Align	Center 👻
Normal Text Color	12.11
Selected Text Cold	r SELECTED_TEXT -

Figure 86. Configure Prompt Properties

35. In the **Project View** under **window1**, click **prompt\_1**, then modify the properties to match the following figure.

I Propertie	s View –
Widget Type	prompt
Widget Name	window1_text
Widget Id	ID_WINDOW1_TEXT
User Data	
Left	216
Тор	250
Width	80
Height	20
Border	No Border 🗸 🗸
Transparent	<b>V</b>
Draw Selected	
Enabled	
Accepts Focus	<b>V</b>
Runtime Allocate	
Normal fill	WIDGET_FILL -
Selected fill	SELECTED_FILL -
Draw Function	
Event Function	
String ID	WINDOW1 -
Text	Window 1
Font	PROMPT -
Text Align	Center -
Normal Text Color	TEXT
Selected Text Cold	

Figure 87. Configure Window Text Properties



36. In the **Project View** under **window2**, click **prompt\_2**, then modify the properties to match the following figure.

E Propertie	es View –
Widget Type	prompt
Widget Name	hellotext
Widget Id	ID_HELLO
User Data	
Left	25
Тор	25
Width	430
Height	222
Border	No Border 🗸 🗸
Transparent	
Draw Selected	
Enabled	<b>V</b>
Accepts Focus	<b>V</b>
Runtime Allocate	
Normal fill	WIDGET_FILL -
Selected fill	SELECTED_FILL
Draw Function	
Event Eurotion	
String ID	HELLO_WORLD
Text	Hello World -> Press anywhere to go to win
Font	PROMPT -
Text Align	Center 👻
Normal Text Color	TEXT
Selected Text Cold	r SELECTED_TEXT -

Figure 88. Configure Hello Text Prompt Properties

37. In the **Project View** under **window2**, click **prompt\_3**, then modify the properties to the following figure.

Deservition 1	V	
Properties V	View	-
Widget Type p	prompt	
Widget Name v	window2_text	
Widget Id I	ID_WINDOW2_TEXT	
User Data		_
Left 2	216	_
Тор	250	_
	80	
	20	_
-	No Border	
		_
	V	
	WIDGET_FILL	-
Selected fill	SELECTED_FILL	
Draw Function		
Event Function		
String ID	WINDOW2	-
Text V	Window 2	
Font F	PROMPT	-
	Center	-
	TEXT	-
	SELECTED_TEXT	-
	OCCCOTED_TEXT	•

Figure 89. Configure Window Text Properties



After these configuration steps, the two windows should look like Figure 90 and Figure 91.

Stay in window1 Press Me!	
Press to activate (blue), press "Press Me" for more.	
Window1	

Figure 90. Configured Window1

Hello World -> Press anywhere to go to window 1	
Window 2	

Figure 91. Configured Window2

38. Expand the **Pixelmaps** section on the right by clicking the **+** symbol.

#### 39. Click System.

Pixelmaps –				
View	Dims	Name	Size	
<mark>=</mark> S	ystem			
	16 x 16	CHECKBOX_OFF	1KB	
<b>V</b>	16 x 16	CHECKBOX_ON	1KB	
0	16 x 16	RADIO_OFF	1KB	
•	16 x 16	RADIO_ON	1KB	
i C	ustom			

Figure 92.	Configuration of Pixelmaps
------------	----------------------------

- 40. Double-click **CHECKBOX\_OFF** to edit the **Pixelmap**.
- 41. Deselect Compress Output and click Save.
- 42. Double-click **CHECKBOX\_ON** to edit the **PixeImap**.
- 43. Deselect **Compress Output** and click **Save**.
- 44. **Save** the project.





#### Figure 93. Save Project

## 45. From the pull-down menu, select **Project > Generate all Output Files**.

Pr	oject Edit Insert Configure Help
	New Project Open Project Save Project
	Save Project As Close Project
Г	Recent Projects Generate All Output Files
	Generate Resource Files Generate Specification Files
	Exit



46. Click on Generate.



✓ Select Export Resources	
Displays            display_1               color table               pixelmap table               English	
Specify Resource File Name	
☐ binary mode	
Binary Mode File Format                 S-Record Binary Memory Offset:	
Cancel	

Figure 95. Select Export Resources

47. Return to  $e^2$  studio.

## 6. Adding Code for Custom Interface Controls

- 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. Drag the file from the Windows Explorer window into the src folder inside the e<sup>2</sup> studio Project Explorer window.
- When asked how 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 (button, checkbox, prompt).

Build the project by clicking the **Hammer icon**, below the **Menu Bar**. There should be no errors reported in the build output.



#### Figure 96 Build Button

Handlers can be found in the source file given with the application description. To add the handlers to your code, right click on **src**, then **System Explorer** as shown in the following figure.





Figure 97. Adding handlers

After you are done importing the file from the file source, you should have a structure similar to the following figure.

Ė∾ <mark>29 src</mark>
🕀 🔁 gui
🗄 🗁 svnerav gen
uiapp_event_handlers.c
ter lo hal_entry.c
🗄 main_thread_entry.c

Figure 98. Final src structure

GUIX handles events at a system level. To handle custom commands like screen transitions and button actions event handler need to be defined. The following event handler for window1 provides an example.



```
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_button_text_id(widget>gx_widget_parent, ID_WINDOWCHANGER,
GX_STRING_ID_BUTTON_ENABLED);
       update_promt_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_button_text_id(widget>gx_widget_parent, ID_WINDOWCHANGER,
GX_STRING_ID_BUTTON_DISABLED);
        update_promt_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 and the box will be checked.

## 7. Running the Application

- 1. On the DK-S7G2 Synergy MCU, perform the following steps:
  - A. Set DIPSW S5 DRAM switch to on.
  - B. Set DIPSW S5 EXP to off.
  - C. Connect 5VDC Power to J1 (power plug).
  - D. Connect the JLink-OB on J17 of the DK-S7G2 MCU main board to the PC using a micro USB cable.
  - Note: The application is not yet ready to be run on the target hardware. The following steps are necessary to run it.



- 2. On the PC, click the dropdown menu for the debug icon.
- 3. Select the **Debug Configurations...** option.

**	- 🖸 - 💁 - 🙋 🖉 - 🏒
	(no launch history)
	Debug As 🔸
	Debug Configurations
	Organize Favorites
	C.4

#### Figure 99. Debug Options

- 4. Under the **Renesas GDB Hardware Debugging** section, select **GUIApp Debug**.
- Click on 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.

P Debug Configurations	
reate, manage, and run configurations	TO-
Image: Second Secon	Name:       GUIApp Debugg         Main       Image: Debugger         Project:       GUIApp         GUIApp       Browse         C/C++ Application:       Debug/GUIApp.elf         Variables       Search Project         Build (if required) before launching       Build configuration:         Use Active           ©       Enable auto build         Image: Use workspace settings       Configure Workspace Settings
Filter matched 10 of 13 items	Apply Revert
0	Debug

Figure 100. Debug Configurations



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

e² Con	firm 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?
<u>R</u> e	member my decision
	Yes No

Figure 101. Perspective Switch Dialog

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

s Views	Run Window
× 🕩 1	I 🛛 🕅 3. 3
Re	sume (F8)

Figure 109. Resume Button

- Press F8 or the resume button to run the code.
   Note: The GUI created earlier should now be on the screen.
- 9. An overview of the Demo is as follows.

Figure 102. Window1

- A. Figure 102 shows **window1**. In this window are four elements:
  - Button Text Box: This box simply shows what the window does if you press outside the Text

     Prompt 1 area. (Refer to Button Checkbox to see how it is changed.) Press in this area to activate the window1 \_handler event that is picked up by guiapp\_event\_handlers.c
     where the code changes the window to window2.
  - b. Button Checkbox: This button is used to enable going to window2. Text is set to Press Me! and it is unchecked. When you press within the Checkbox active area you activate the event window1\_event\_handler. 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.
  - c. **Text Prompt 1**: This area instructs you how to control the demo. (Refer to **Button – Checkbox** to see how it is changed.)
  - d. **Text Prompt 2:** This Prompt is used to show you the window being displayed. It never changes (always shows **window1**).





Figure 111. Window2

- B. The above figure shows window2. In this window are two elements:
  - a. **Text Prompt 1**: This area presents **Hello World** and instructs you how to return to **window1**. Pressing in this area initiates the window2\_handler event that is picked up by guiapp\_event\_handlers.c and changes the active window to **window1**.
  - b. **Text Prompt 2:** This prompt is used to show you the window being displayed. It never changes (always shows **window2**).
- 10. Press **Ctrl + F2** or the **Stop** button to end the debug session.

× 🕨 II 🔲 M 😎 🕫	s Vie	ws Run	Window	Λ	
	ال 🖉	>	N 2 C	9	

#### Figure 112. Stop Button

11. This concludes the **GUIX Hello World** for the DK-S7G2 Synergy MCU.

### 8. Appendix

The GUIX image resources files are default stored in the internal code flash. The resource files can also be stored in the external flash such as QSPI. Refer the Knowledgebase link (<u>https://en-</u>support.renesas.com/knowledgeBase/18054800) about using QSPI for storing the image resource files.

Note: Users are required to make the QSPI pins drive capacity to High instead of Low when QSPI is used for external storage (On DK-S7G2 Board).



## Website and Support

Visit the following URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Platform MCUs Synergy Software Package Software add-ons SSP Components MCU Components Kits

Synergy Solutions Gallery Partner projects Application projects

Self-service support resources:

Knowledgebase Forums Training Videos Chat and web ticket www.renesas.com/renesas-synergy-platform-mcus www.renesas.com/synergy/ssp www.renesas.com/synergy/addons www.renesas.com/synergy/sspcomponents www.renesas.com/synergy/components-synergy-mcus www.renesas.com/synergy/kits

www.renesas.com/synergy/solutionsgallery www.renesas.com/synergy/partnerprojects www.renesas.com/synergy/applicationprojects

www.renesas.com/synergy/knowledgebase www.renesas.com/synergy/forum www.renesas.com/synergy/training www.renesas.com/synergy/videos www.renesas.com/synergy/resourcelibrary



# **Revision History**

		Description		
Rev.	Date	Page	Summary	
1.00	Jan.22.15	All	Created Initial Document	
1.01	Jul.06.16	All	Updated for SSP v1.1.0	
1.02	May.15.17	All	Updated for SSP v1.2.1	
1.03	Aug.11.17	All	Updated for SSP v1.3.0	
1.04	Nov.13.17	All	Updated for SSP v1.3.1	
1.05	Feb.27.18	All	Updated for SSP v1.4.0	
1.06	Jun.18.18	—	Sample codes updated.	
1.07	Oct.10.18	—	Updated for SSP v1.5.0	
1.08	Mar.08.19	—	Updated for SSP v1.6.0	
1.09	Oct.15.21	—	Updated for SSP v1.6.0 "Touch Panel V2 Framework"	
1.10	Nov.10.21	—	Updated for SSP v1.6.0 to SSP v2.1.0	
1.11	Apr.21.23	—	Deleted SSP licensing section and messaging framework references	



### 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. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

#### 2. Processing at power-on

The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. 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 the 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.

#### 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.

#### 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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

- 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
   Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended 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; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems: safety control equipment: etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

- 7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
- 8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 11. Renease Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
   Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas
- Electronics products.
- (Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## **Corporate Headquarters**

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

#### www.renesas.com

#### Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## **Contact information**

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: <a href="http://www.renesas.com/contact/">www.renesas.com/contact/</a>.