Renesas RA Family

Getting Started with GUIX Thermostat Application

Introduction

This application, which is a Thermostat application, provides a reference for developing complex multithreaded applications with a touch screen graphical Human Machine Interface (HMI) by using Renesas FSP and Azure RTOS GUIX. It describes steps to create a basic GUIX for FSP, integrates touch driver, handles multiple hardware accesses, system updates, and event handling.

This application is developed using the Renesas RA Flexible Software Package (FSP), which provides a quick and versatile way to build secure connected Internet of Things (IoT) devices using the Renesas RA family of Arm microcontrollers (MCUs). RA FSP provides production ready peripheral drivers to take advantage of the RA FSP ecosystem along with Azure RTOS GUIX library and Azure RTOS. "In addition, FSP also provides Ethernet, USB, File System and other middleware stacks as well. This powerful suite of tools provides a comprehensive, integrated framework for rapid development of complex embedded applications.

This application note assumes that you are familiar with the concepts associated with writing multi-threaded applications under a Real Time Operating System (RTOS) environment, such as Azure RTOS. This application note makes use of RTOS features such as threads and semaphores. Prior experience in using Azure RTOS would be helpful for easy understanding of the provided application project. For more detailed information on Azure RTOS features, refer to the Azure RTOS User Manual.

The Graphics application is developed using the Renesas e² studio Integrated Solution Development Environment (IDE). e² studio is integrate with the FSP platform installer, which can be downloaded from Renesas website. The intuitive configurators and code generators in e² studio and FSP will help the application developers in creating such complex multi-threaded graphics applications very quickly. This application note walks you through all the necessary steps in creating, building and running a complex graphics project, including the following:

- Board setup.
- Install tools.
- Build and run application.
- Azure RTOS GUIX Studio project integration.
- Setup Azure RTOS GUIX Studio project.
- Add Touch Driver.
- Create FSP GUIX project.
- Hardware Setup.
- Using the General Purpose Timer to drive a PWM backlight control signal.

Required Resources

Development tools and software

- e² studio IDE 2024-01
- Renesas Flexible Software Package (FSP) v5.2.0
- Azure RTOS GUIX Studio V6.4.0.0

Hardware

Renesas EK-RA6M3G kit (RA6M3 MCU Group)
 (https://www.renesas.com/us/en/products/software-tools/boards-and-kits/eval-kits/ek-ra6m3g.html)

Reference Manuals

- RA Flexible Software Package Documentation Release v5.2.0
- Azure RTOS GUIX and GUIX Studio v6.4.0.0
- Renesas RA6M3 Group User's Manual Rev.1.20
- EK-RA6M3G-v1.0 Schematics

Contents

1.	Installing Tools	3
1.1	Overview	3
1.2	Procedural Steps	3
2.	Create Application Note Project	5
2.1	Overview	5
2.2	Procedural Steps	5
3.	Using GUIX Widget Timer to Trigger a Screen Transition	19
3.1	Overview	19
3.2	Procedural Steps	19
4.	Add Touch Driver to Thermostat_GUIX_EK_RA6M3G Project	21
4.1	Overview	21
4.2	Procedural Steps	21
5.	Control LCD Backlight	
5. 5.1	Control LCD Backlight	29
	-	29 29
5.1	Overview	29 29 29
5.1 5.2	Overview Procedural Steps	29 29 29 33
5.1 5.2 6.	Overview Procedural Steps Update Date/Time and Temperature	29 29 29 33 33
5.1 5.2 6. 6.1	Overview Procedural Steps Update Date/Time and Temperature Overview	29 29 33 33 33
 5.1 5.2 6. 6.1 6.2 	Overview Procedural Steps Update Date/Time and Temperature Overview Procedural Steps	29 29 33 33 33 33
 5.1 5.2 6. 6.1 6.2 7. 	Overview Procedural Steps Update Date/Time and Temperature Overview Procedural Steps Setting Date/Time in A Full Function Project	29 29 33 33 33 36 36
 5.1 5.2 6. 6.1 6.2 7. 7.1 	OverviewProcedural Steps Update Date/Time and Temperature Overview Procedural Steps Setting Date/Time in A Full Function Project Overview	29 29 33 33 36 36 37

1. Installing Tools

1.1 Overview

In this section you will copy the application note (AN) materials to your PC and install e^2 studio v2024-01/FSP v5.2.0 and Azure RTOS GUIX Studio v6.4.0.0.

1.2 Procedural Steps

- 1. If you already have e² studio v2024-01 with FSP v5.2.0 or later installed, you can skip this step. Otherwise, you can download it from this <u>link</u>.
- 2. You can get Azure RTOS GUIX Studio v6.4.0.0 or greater from this <u>link</u>. If it goes well, you will see the window in the next step on the web browser.

Note: It needs Microsoft Store working on your PC to install Azure RTOS GUIX Studio.

3. Click Get to start installing Azure RTOS GUIX Studio.



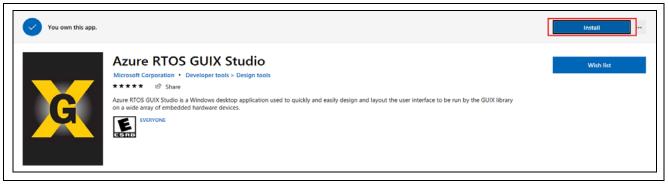
Figure 1. Clicking Get to Start Installing Azure RTOS GUIX

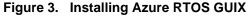
4. Click Open Microsoft Store to continue installing Azure RTOS GUIX Studio.

Open Microsoft Store?	
https://www.microsoft.com wants to open this application.	
Open Microsoft Store Cancel	

Figure 2. Clicking Open Microsoft Store-

5. Click Install to continue. A window shows up to ask for a Microsoft account, which is seen in the next step.





6. Ignore it by clicking "X" on the top-right to close this pop-up window and continue Azure RTOS GUIX Studio installation.

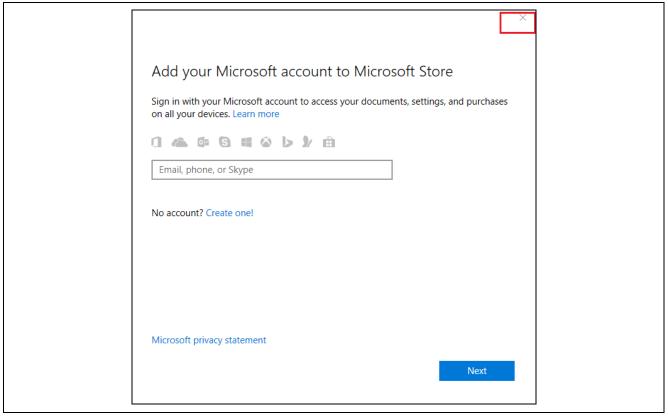


Figure 4. Closing Pop-up Window to Continue Installing Azure RTOS GUIX

7. Downloading and installation of Azure RTOS GUIX Studio starts.

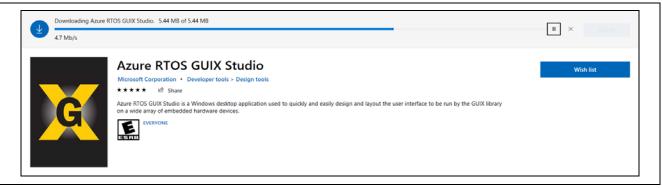


Figure 5. Starting of Downloading and Installation

8. Click Launch to launch Azure RTOS GUIX Studio.

This product is installe	-d.	Launch
G	Azure RTOS GUIX Studio Microsoft Corporation • Developer tools > Design tools ***** □ □ Share Azure RTOS GUIX Studio is a Windows desktop application used to quickly and easily design and layout the user interface to be run by the GUIX library on a wide array of embedded hardware devices. For a wide array of embedded hardware devices.	Wish list



9. Azure RTOS GUIX Studio launched.

G Azure												_		×
Project E					~	ka	đ	-		-		~		
and the second s		8	ж	13	0	P	-	<u>I</u>	<u>.</u>	2	101	2	ΙŽ	9
👬 Pro	ject Vi	iew	1.000											
🚺 No Proj	ject		_											
i i														
1														
III Pro	perties	s View												
1	1972	1.												

Figure 7. Azure RTOS GUIX Studio Launched

10. Close Azure RTOS GUIX Studio, for now, you will open it again later.

2. Create Application Note Project

2.1 Overview

In this section, you will create a project to which you will add pre-written source code and integrate it with a pre-created Azure RTOS GUIX Studio project.

2.2 Procedural Steps

1. Create a new **RA C/C++ project**. Name it as Thermostat_GUIX_EK_RA6M3G.

🛐 New C/C++ Project	- 🗆 X
Templates for Renesas RA Project	
All Renesas RA C/C++ Project C/C++ Create an executable or static library C/C++ project for	or Renesas RA.
	>
(?) < Back Next > Finish	Cancel
•	
Renesas RA C/C++ Project	– 🗆 X
Renesas RA C/C++ Project Project Name and Location	
Project name Thermostat_GUIX_EK_RA6M3G Use default location Cotation Cotation Cotation Cotation You can download more Renease packs here	5M3G Browse
? < Back Next > Finish	Cancel

Figure 8. Creating New RA C/C++ Project

2. Set board to EK-RA6M3G.

Renesas RA C/C++ Project			
Device and Tools Selection			
Device Selection			
FSP Version: 5.2.0	~	Board Description	
Board: EK-RA6M3G			Graphics for RA6M3 MCU Group
			renesas.com/ra/ek-ra6m3g to get kit user's
Device: R7FA6M3AH3CF	C		rt guide, errata, design package, example projects,
Core: CM4		etc.	
Language:		Device Details	
		TrustZone	No
		Pins	176
		Processor	Cortex-M4
Toolchains		Debugger	
GNU ARM Embedded		J-Link ARM	~
LLVM Embedded Toolchain for	Arm		
ARM Compiler 6.21			
13.2.1.arm-13-7	~		

Figure 9. Setting Board to EK-RA6M3G

3. Select Azure RTOS ThreadX (v6.2.1+FSP.5.2.0).

Renesas RA C/C++ Project		- 🗆 X
enesas RA C/C++ Project Build Artifact and RTOS Selection		Ď
Build Artifact Selection Executable Project builds to an executable file Static Library Project builds to a static library file Executable Using an RA Static Library Project builds to an executable file Project builds to an executable file Project uses an existing RA static library project 	RTOS Selection Azure RTOS ThreadX (v6.2.1+fsp.5.2.0)	~
?	< Back Next > Finish	Cancel

Figure 10. Selecting Azure RTOS ThreadX

4. Use Azure RTOS ThreadX-Minimal template.

Renesas RA C/C++ Projec	t			
Project Template Selection				
Project Template Selection				
Azure RTOS	TOS ThreadX - Blinky ThreadX project that includes BSF X will also be initialized and a sing v.5.2.0.pack]			ICU using the
	TOS ThreadX - Minimal adX FSP project with no threads. 1 .5.2.0.pack]	alize the MCU using	the BSP.	
Code Generation Settings				
✓ Use Renesas Code Forma	itter			

Figure 11. Selecting Azure RTOS ThreadX – Minimal Template

5. Open project configuration, go to **BSP** tab, change Heap size to 0x2000.

at_GUIX_EK_RA6M3G		_GUIX_EK_RA6M3G] FSP Configuration \times	
at_GUIX_EK_RA6M3G	Board Supp		
uration.xml ostat_GUIX_EK_RA6M3G D¢ oper Assistance	Device Select FSP version: Board: Device: Core: RTOS:	5.2.0 EK-RA6M3G ~ 2	Visit <u>https://ww</u>
>	Summary BSP	Clocks Pins Interrupts Event Links Stacks Com	ponents
< 🔝 Problems 🏼 🏶 Smart	Browser 📃 Cor	nsole 🛷 Search 🥫 Memory Usage	
i			
pperty Main stack size (bytes) Heap size (bytes) MCU Vcc (mV) Parameter checking Assert Failures		3300 Disabled	
	perty Main stack size (bytes) Heap size (bytes) MCU Vcc (mV) Parameter checking	FSP version: Board: Device: Core: RTOS: Summary BSP Problems Smart Browser Core Perty Main stack size (bytes) Heap size (bytes) MCU Vcc (mV) Parameter checking Assert Failures	FSP version: 5.2.0 Board: EK-RA6M3G I Board: EK-RA6M3G I Device: R7FA6M3AH3CFC I Device: CM4 RTOS: Azure RTOS ThreadX Summary BSP Clocks Pins Interrupts Event Links Stacks Come Summary BSP Clocks Pins Interrupts Event Links Stacks Come Summary BSP Clocks Pins Interrupts Event Links Stacks Come Memory Usage Value 0x400 Heap size (bytes) 0x2000 MCU Vcc (mV) 3300 Parameter checking Disablee Assert Failures Return F

Figure 12. Changing Heap size (bytes) in Project Configuration

6. You can remove Azure RTOS ThreadX Port if you want.

💱 *[Thermostat_GUIX_EK_RA6M3G] FSP Configuration $~ imes~$		
Stacks Configuration		Generate Project Content
Threads New Thread Remove	Azure RTOS ThreadX Port (rm_threadx_port) Stacks	🛃 New Stack > 🚢 Extend Stack > 🔊 Remove
 g ioport I/O Port (r ioport) Azure RTOS ThreadX Port (rm_threadx_port) 	Azure RTOS ThreadX Port (rm_threadx_port)	
	1	
Objects New Object > 🔊 Remove		
ummary BSP Clocks Pins Interrupts Event Links Stacks	Components	

Figure 13. Remove Azure RTOS ThreadX Port

7. Add a **New Thread** and name it as **System Thread** with the following settings.

Threads	👰 New Thread 🙀 Remov	ve 📄 Syster	m Thread Stacks	🛃 New Stack 🤅
4	AL/Common g_ioport I/O Port Driver on r_ioport ystem Thread	i	Add stacks to the selected th clipboard.	read by using the 'New Stack >' toolbar button (above),
Objects	🔊 New Object > 🔬 Re	emove		
Pin Cor	nflicts 🔲 Properties 🔀		1	
System				
	Thread Property ✔ Common		Value	
System	Thread Property		Value	
System	Thread Property Common Seneral		Value	
System	Thread Property Common General Timer Trace Performance		Value	
System	Thread Property Common General Timer Trace Performance RA		Value	
System	Thread Property Common General Timer Trace Performance RA Interrupts		Value	
System	Thread Property Common General Timer Trace Performance RA Interrupts Thread			
System	Thread Property Common General Timer Trace Performance RA Interrupts		system_thread	
System	Thread Property Common General Timer Trace RA Interrupts Thread Symbol Name			
System	Thread Property Common General Timer Trace Performance RA Interrupts Thread Symbol		system_thread System Thread	
System	Thread Property Common General Timer Trace Proformance RA Interrupts Thread Symbol Name Stack size (bytes)		system_thread System Thread 1024	

Figure 14. Adding a New Thread and Naming it System Thread

Г

8. Add Azure RTOS GUIX to System Thread.

🐺 *[Thermostat_GUIX_EK_RA6M3G] FS	P Configuration \times			% FSP Visualization $ imes$
Stacks Configuration		Generate P	Operation of the second sec	E A P P
Threads New Thread	System Thread Stacks Add stacks to the selected thread by using from the clipboard.	New Stack Al the 'New Stack' toolbar but Analog Audio Bootloader CapTouch Connectivity DSP	> emove > here > > >	€ RENESAS
Objects		Graphics Input Monitoring Motor Networking Power Security	> ⊕ D/AV > ⊕ D/AV > ⊕ Graph > ⊕ JPEG	RTOS GUIX E 2D (r_drw) E 2D Port Interface (r_drw) hics LCD (r_glcdc) Codec (r_jpeg) el Data Capture (r_pdc)



9. Property settings for the g_display0 Graphics LCD on r_glcdc.

	🖹 Problems 📮 Console 🦇 Smart Browser 🔑 Smart Manua	al 🛷 Search 🔋 Memory
	raphics LCD (r_glcdc)	
Settings Prop	perty	Value
Settings V M	10dule g_display0 Graphics LCD (r_glcdc)	
A PL Info	General	
	Interrupts	
	/ Input	
	 ✓ Graphics Layer 1 	
	✓ General	
	Enabled	Yes
	Horizontal size	480
	Vertical size	272
	Horizontal position	0
	Vertical position	0
	Color format	RGB565 (16-bit) Disabled
	Line descending mode	Disabled
	> Background Color	
	✓ Framebuffer	<u> </u>
	Framebuffer name	fb_background
	Number of framebuffers	2
	Section for framebuffer allocation	.bss
Prope	> Line Repeat rties 🗙 😰 Problems 📮 Console 👒 Smart Browser 斗 Smart Man	ual 🔗 Search 🚺 Memory
g_displa	rties × R Problems Console Smart Browser Smart Man ay0 Graphics LCD (r_glcdc)	
g_displa Settings	rties × 😰 Problems 🗳 Console 🚸 Smart Browser 🖓 Smart Man ay0 Graphics LCD (r_glcdc) Property	ual 🔗 Search 🚺 Memory Value
g_displa	rties × Problems © Console 🏶 Smart Browser 🖓 Smart Man ay0 Graphics LCD (r_glcdc) Property > General	
g_displa Settings	rties × 😰 Problems 🗳 Console 🚸 Smart Browser 🖓 Smart Man ay0 Graphics LCD (r_glcdc) Property	
g_displa Settings	rties × Problems © Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts	
g_displa Settings	rties × Problems Console Smart Browser Smart Man ay0 Graphics LCD (r_glcdc) Property > General > Interrupts > Input	
g_displa Settings	rties × Problems © Console & Smart Browser % Smart Man ay0 Graphics LCD (r_glcdc) Property > General > Interrupts > Input < Output < Timing Horizontal total cycles	Value 525
g_displa Settings	rties × Problems © Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input * Output * Timing Horizontal total cycles Horizontal active video cycles	Value 525 480
g_displa Settings	rties × Interrupts > Input • Output • Output • Timing Horizontal total cycles Horizontal back porch cycles	Value 525
g_displa Settings	rties × Repoblems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input > Output > Output > Output Voutput Horizontal total cycles Horizontal active video cycles Horizontal back porch cycles Horizontal sync signal cycles	Value 525 480 40 1
g_displa Settings	rties × Problems Console Smart Browser Smart Man ay0 Graphics LCD (r_glcdc) Property > General > Interrupts > Input V Output V Timing Horizontal total cycles Horizontal sync signal cycles Horizontal sync signal polarity	Value 525 480 40 1 1 Low active
g_displa Settings	rties × Problems Console Smart Browser Smart Man ay0 Graphics LCD (r_glcdc) Property > General > Interrupts > Input V Output V Output V Timing Horizontal total cycles Horizontal back porch cycles Horizontal sync signal cycles Horizontal sync signal polarity Vertical total lines	Value 525 480 40 1 1 Low active 316
g_displa Settings	tties × Problems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input • Output • Timing Horizontal total cycles Horizontal total cycles Horizontal back porch cycles Horizontal sync signal cycles Horizontal sync signal polarity Vertical total lines Vertical active video lines	Value 525 480 40 1 Low active 316 272
g_displa Settings	rties × Problems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input • Output • Timing Horizontal total cycles Horizontal back porch cycles Horizontal back porch cycles Horizontal sync signal polarity Vertical total lines Vertical active video lines Vertical active video lines	Value 525 480 40 1 1 Low active 316
g_displa Settings	rties × Problems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input > Output Votput Votput Votput Horizontal total cycles Horizontal back porch cycles Horizontal sync signal cycles Horizontal sync signal cycles Horizontal sync signal cycles Vertical total lines Vertical back porch lines Vertical back porch lines Vertical back porch lines Vertical back porch lines	Value 525 480 40 1 Low active 316 272
g_displa Settings	rties × Problems Console Smart Browser Smart Man ay0 Graphics LCD (r_glcdc) Property > General > Interrupts > Input Voltput Voltput Viriang Horizontal total cycles Horizontal active video cycles Horizontal spac signal cycles Horizontal sync signal polarity Vertical active video lines Vertical active video lines Vertical active video lines Vertical active video lines Vertical active signal polarity Vertical active video lines Vertical active signal polarity Vertical active signal polarity Vertical active video lines Vertical sync signal lines Vertical sync signal polarity	Value 525 480 40 1 Low active 316 272 8 8 1
g_displa Settings	rties × Problems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input > Output Votput Votput Votput Horizontal total cycles Horizontal back porch cycles Horizontal sync signal cycles Horizontal sync signal cycles Horizontal sync signal cycles Vertical total lines Vertical back porch lines Vertical back porch lines Vertical back porch lines Vertical back porch lines	Value 525 480 40 1 Low active 316 272 8 1 1 Low active
g_displa Settings	rties × Problems Console Smart Browser Smart Man ay0 Graphics LCD (r_glcdc) Property > General > Interrupts > Input < Output < Output < Output < Timing Horizontal total cycles Horizontal active video cycles Horizontal sync signal polarity Vertical active video lines Vertical active video lines Vertical sync signal polarity Vertical sync signal polarity Vertical sync signal polarity Vertical sync signal polarity Data Enable Signal Polarity Data Enable Signal Polarity	Value 525 480 40 1 1 Low active 316 272 8 1 1 Low active High active
g_displa Settings	rties × Problems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input > Output V Output V Timing Horizontal total cycles Horizontal back porch cycles Horizontal sync signal cycles Horizontal sync signal cycles Horizontal sync signal cycles Horizontal sync signal polarity Vertical total lines Vertical sync signal polarity Vertical sync signal polarity Data Enable Signal Polarity Data Enable Signal Polarity Sync edge	Value 525 480 40 1 1 Low active 316 272 8 1 1 Low active High active
g_displa Settings	tties × Problems Console Smart Browser Smart Man ayO Graphics LCD (r_glcdc) Property > General > Interrupts > Input • Output • Timing Horizontal total cycles Horizontal total cycles Horizontal back porch cycles Horizontal sync signal polarity Vertical total lines Vertical active video lines Vertical sync signal polarity Vertical sync signal polarity Data Enable Signal Polarity Data Enable Signal Polarity Sync edge • Format	Value 525 480 40 1 Low active 316 272 8 1 Low active High active Falling edge

Figure 16. Graphics LCD Display Driver Configuration

10. Configure the Hsync, Vsync, Data Enable pins and clock as shown.

-	es 🗙 🖹 Problems 📮 Console 🁒 Smart Browser 🤑 Smart Manual 🔗 Search 🚦	Memory
g_display	/0 Graphics LCD (r_glcdc)	
Settings	Property VTCON	Value
API Info	Hsync pin select	LCD_TCON0
	Vsync pin select	LCD_TCON1
	Data enable (DE) pin select	LCD_TCON2
	Panel clock source	Internal clock (GLCDCLK)
	Panel clock division ratio	1/24
	> Color Correction	
	> Dithering	

Figure 17. TCON Properties

11. In Pin Configuration, change P603's mode to Output mode (Initial high) to enable LCD panel backlight.

Select Pin Configuration				🔛 Export to CSV file 🗧	Configure Pin Driver Warnings
RA6M3G-EK.pincfg	✓ Mar	nage configurations		🗹 Generate data:	g_bsp_pin_cfg
Pin Selection	⊞ ⊫ ↓ <mark>a</mark>	Pin Configuration			
Type filter text		Name		Value	Link
✓ ✓ P6	^	Symbolic Name			
✓ P600		Comment			
✓ P601		Mode		Output mode (Initial High)	
✓ P602		Pull up		None	
✓ P603		Drive Capacity		Low	
P604		Output type		CMOS	
P605		✓ Input/Output			
P606		P603		V GPIO	
P607					
✓ P608					
✓ P609					
✓ P610		Module name:	P603		
✓ P611	~	Port Capabilities:	BUS0: D13_DQ13 GPT7: GTIOCA		
Pin Function Pin Number					

Figure 18. Changing Pin Configuration Mode to Enable LCD Panel Backlight

O

12. In RA Configurator, click Generate Project Content to generate project content. Make sure project is active,

click to build the project. It may take a long period of time to finish building an Azure RTOS/GUIX project on your PC.

13. Copy Azure RTOS GUIX Studio project to e² studio project (Themostat_GUIX_EK_RA6M3G) by copying "guix_studio" folder in the application note (**AN**) folder (FSP_GUIX_Thermostat) and pasting it in the Themostat_GUIX_EK_RA6M3G project.

Name	Date modified	Туре
2.23	7/1/2021 11:10 AM	File folder
2.28	7/1/2021 11:10 AM	File folder
3.1	7/1/2021 11:10 AM	File folder
4.11	7/1/2021 11:10 AM	File folder
5.6	7/1/2021 11:10 AM	File folder
6.6	7/1/2021 11:10 AM	File folder
completed_project	7/1/2021 11:11 AM	File folder
guix_studio	7/1/2021 11:10 AM	File folder
touch_ft5x06	7/1/2021 11:10 AM	File folder
le Edit Navigate Search Project Re	New	
📄 🛞 🕶 🎸 🔶 🐐 🏟 🗄	Go Into	
Project Explorer X	Open in New Window	N
	Show In	Alt+Shift+W
🗁 Thermostat GUIX EK RA6M3G 🖳	Сору	Ctrl+C
Thermostat_GUIX_EK_RA6M3G US > Binaries	Copy Paste	Ctrl+C Ctrl+V
> 🐇 Binaries		
> 🐇 Binaries	Paste	Ctrl+V
> 🐇 Binaries 🛅 > 🔊 Includes 🕺	Paste Delete	Ctrl+V
> 🐇 Binaries 🛅 > 🔊 Includes X > 😂 ra	Paste Delete Source	Ctrl+V
> Similaries	Paste Delete Source Move	Ctrl+V Delete
> Similaries > Similaries > Similaries > Similaries > Similaries > Similaries	Paste Delete Source Move Rename	Ctrl+V Delete
> Similaries > Similaries > Similaries > Similaries > Similaries > Debug	Paste Delete Source Move Rename Import	Ctrl+V Delete
 Similaries Similaries Includes Pagen <li< th=""><td>Paste Delete Source Move Rename Import Export</td><td>Ctrl+V Delete</td></li<>	Paste Delete Source Move Rename Import Export	Ctrl+V Delete
 Includes ra ra_gen ra_gen rsc Debug Debud build ra_cfg script configuration.xml 	Paste Delete Source Move Rename Import Export Renesas FSP Export	Ctrl+V Delete
 Similaries Similaries Includes Pagen <li< th=""><td>Paste Delete Source Move Rename Import Export Renesas FSP Export Build Project</td><td>Ctrl+V Delete</td></li<>	Paste Delete Source Move Rename Import Export Renesas FSP Export Build Project	Ctrl+V Delete

Figure 19. Copying the Azure RTOS GUIX Studio Project to e² studio

14. GUIX Studio project is now in Thermostat_GUIX_EK_RA6M3G project. In e² studio, right-click the "guix_studio" folder and exclude it from the build since it contains the Azure GUIX Studio project, which will not be built by FSP.

🍋 Project Explorer 🗦		New Go Into	>] FSP Cor	nfiguration $ imes$
∨ 📂 Thermostat_GI		Show In	Alt+Shift+W >		
> 🐝 Binaries	ľ	Сору	Ctrl+C		
> 🔊 Includes	Ē	Paste	Ctrl+V		
> 🐸 ra	×	Delete	Delete		✓ Manage configura
> 😕 ra_gen		Move			
> 🐸 src		Rename	F2	□ ↓ ^a	Pin Configuration
> 🗁 Debug	പ്പ	Import			Name
> 🗁 build	4	Export			Symbolic Name
> 🗁 guix_studio		Build Project	Ctrl+B	^	Comment
> 🗁 ra_cfg	æ	Refresh	F5		Mode
> 🧀 script		Source	>		Dullun
configuration					<
📄 ra_cfg.txt 🖹 Thermostat	_	Build Targets	>	~	Module name: P603
> ⑦ Developer A		Resource Configurations	>		ude from Build
<	0	Run As	>		et to Default ent Links Stacks Compoi
	夺	Debug As	>	· .	•
		Exclude from build			
		Exclude object(s) from build in	the following configurations		
		Release			
		Se	lect All Deselect All		

Figure 20. Excluding guix_studio Folder from the Build

15. Get to Thermostat_GUIX_EK_RA6M3G project folder by right clicking the e² studio project and select "System Explorer" as shown below.

-Ma		
12	Debug As	>
	Team	>
	Compare With	>
	Restore from Local History	
	MISRA-C	>
1	C/C++ Project Settings	Ctrl+Alt+P
	Save build settings report	
	Change Device	
*	Run C/C++ Code Analysis	
1	System Explorer	
65.	Command Prompt	
\checkmark	Validate	
	Configure	>
	Source	>
	Properties	Alt+Enter

Figure 21. Selecting System Explorer

16. Open thermostat.gpx project file in "guix_studio>GNU" sub-folder in your Thermostat_GUIX_EK_RA6M3G folder. If you have several GUIX Studio versions in your system, make sure you choose the right one, which is v6.4.0.0 or later.

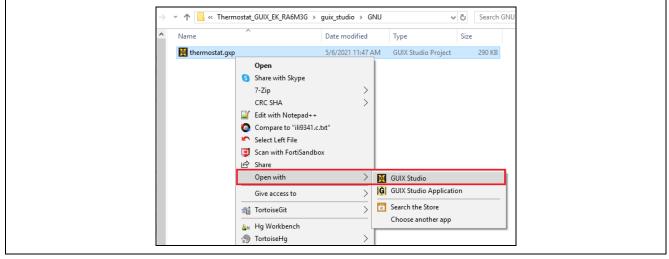


Figure 22. Opening the Project File

17. This GUIX Studio project has a complete design of this Thermostat application. The next several steps describe the process to generate resources, application code and integrate them with an e² studio project.



Figure 23. GUIX Studio Thermostat Application View

18. The Azure RTOS GUIX Studio project consists of 5 screens, including Splash, Main Page, Settings, Thermostat and Help from top to bottom:

	Ver : 1.0.0	
	SAS	
_{May} 11, 2021 Tuesday	09:30:00 AM	
23.8 ºC May 11, 2	2021 * 09:30:00 AM	
Indoor: 23.8°C Set To: 33.0°C Fan: System: AUTO OFF	O Settings	
23.8 °C May 11, 2 Year Month 2021 MAY V	2021 * 09:30:00 AM Thermostat Day Set Time 11 Set Date Sound	
Fan: Syst	8°C	
Fan: AUTO When the fan is set to AUTO it wi air conditioning unit tells it to. When the fan is set to on it will of whether or not the heater and air System: OFF If the system is OFF then all heati to ON then it will not shut off wh COOL When the system is set to COOL oneed to meet the "Set to" temp	stat - Help ill only turn on when either the heater or the continue to circulate the air regardless of r conditioning units are on. ing and cooling units are off. If your fan is set en the system is off. the air conditioning unit will turn on when	

Figure 24. Azure RTOS GUIX Studio Project Screens

19.	Click "Configure->Project/Display	" and confirm the following settings.
-----	-----------------------------------	---------------------------------------

Directories Source Files//src/guix_	gen	browse
Header Files//src/guix_	gen	browse
Resource Files//src/guix_	gen	browse
Target CPU Renesas RA	 ✓ Advanced Setti 	ngs
Toolchain GNU	✓ big endian	
Additional Headers		Insert Before
Number of Displays	GUIX Library Ve	rsion 6 • . 4 • . 0 • Major Minor Patch
Display Configuration		Hajor Hinor Fater
Display Number 1	Name display	
x resolution 480 pix	els y resolution 272	pixels
O 1 bpp	grayscale	1:5:5:5 format
2 bpp	invert polarity	4:4:4:4 format
○ 4 bpp ○ 8 bpp	reverse byte order	3:3:2 format
16 bpp	packed format	
○ 24 bpp○ 32 bpp	allocate canvas memory	Rotation: None ~
Number of Palette Mode Anti-al	ased Text Colors: 8	~

Figure 25. Configure Project Settings

20. Go back e² studio project (Thermostat_GUIX_EK_RA6M3G), right click "**src**", then select "**New->Folder**" and create a folder named "**guix_gen**".

			🕲 New Folder – 🗆 🗙
			Folder Create a new folder resource.
			Enter or select the parent folder:
File Edit Navig	ate Search Project Renesas Views Run Renesas	Al Window Help	Thermostat_GUIX_EK_RA6M3G/src
: <u></u>	New Solution State	Project C/C++ Project	≥ script ≥ src
Project E: ✓ i Project E:	Open in New Window Show In Alt+Shift+W	File File File from Template	
> 🐇 Bin 📄 > 🔊 Incl 🕥	Copy Ctrl+C Paste Ctrl+V	G Class	Folder name: guix_gen
> 😂 ra 🗙 > 😂 ra κ		Header File	Advanced >>
> 😂 src > 🗁 Del	Move Rename F2	Source Folder	
> 🗁 bui 🚵 > 🗁 gui 🍌	Import Export	Ctrl+N	? Finish Cancel

Figure 26. Creating a "guix_gen" in e²studio Project

21. Confirm "guix_gen" is created before moving to next step.

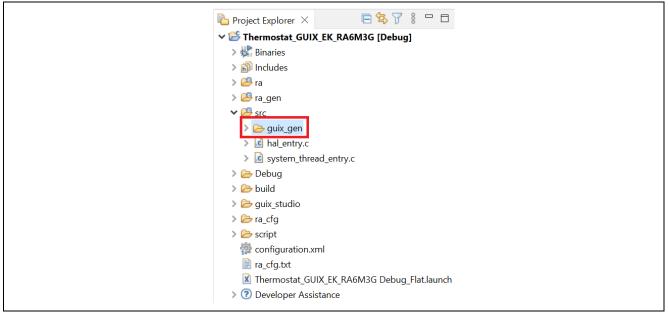


Figure 27. Confirming Creation of "guix_gen"

22. In Azure RTOS GUIX Studio, click **Project->Generate All Output Files** to generate resource files, header files and source files of this GUIX design.

Azure RTOS GUIX Studio 6.4.0 Project Edit Insert Configure F		
New Project Open Project Save Project Save Project As Close Project Import Project	Ctrl+N Ctrl+O Ctrl+S Shift+Ctrl+S	i display color table i font table i fon
Recent Projects	>	Specify Resource File Name
Generate All Output Files		☐ binary mode
Generate Resource Files Generate Specification Files		Binary Mode File Format S-Record OBinary Memory Offset: 0000 0000
Exit	Alt+F4	Cancel Generate

Figure 28. Clicking Generate All Output Files

Click Generate to generate all output files. If succeeded, you will see below notification.

A Notification	×
All Output files have been updated	
	ОК



23. All output files are now in "guix_gen" folder.

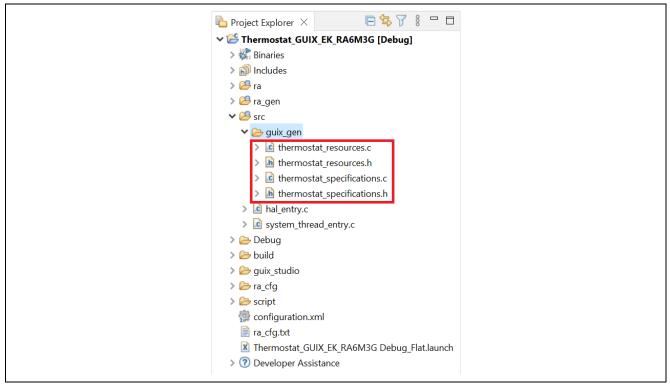


Figure 30. Location of Output Files

24. In the Azure RTOS GUIX Studio Project, click "Splash" and pick up "Widget Name" and "Event Function" definitions. These definitions are used to create a screen and handle it in the e2studio/FSP project. The other windows have similar definitions.

B Splash			Rene	Ver : 1.0.0
Properties V	liew			
Widget Type	window			
Widget Type Widget Name	window Splash	^		
		^	M 44 0004	
Widget Name		_	May 11, 2021	
Widget Name Widget Id		^		09:30:00 AM
Widget Name Widget Id User Data Left Top	Splash 0 0	^ 	May 11, 2021 Tuesday	09:30:00 AM
Widget Name Widget Id User Data Left Top Width	Splash 0 0 480	^ 		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height	Splash 0 0	^ 		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border	Splash 0 0 480			09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent	Splash 0 0 480 272	~		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected	Splash 0 0 480 272	×		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled	Splash 0 0 480 272	~		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled Accepts Focus	Splash 0 0 480 272			09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled Accepts Focus Runtime Allocate	Splash 0 0 480 272 No Border	×		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled Accepts Focus Runtime Allocate Normal fill	Splash 0 0 480 272 No Border WINDOW_FILL	>		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Accepts Focus Runtime Allocate Normal fill Selected fill	Splash 0 0 0 222 No Border WINDOW_FILL WINDOW_FILL	> Y		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled Accepts Focus Runtime Allocate Normal fill Disabled fill	Splash 0 0 480 272 No Border WINDOW_FILL	× □ □ Ŋ □ × ×		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled Accepts Focus Runtime Allocate Normal fill Selected fill Disabled fill Template	Splash 0 0 0 222 No Border WINDOW_FILL WINDOW_FILL	>		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Oraw Selected Accepts Focus Runtime Allocate Normal fill Selected fill Disabled fill Template Visible At Startup	Splash 0 0 0 222 No Border WINDOW_FILL WINDOW_FILL	× □ □ Ŋ □ × ×		09:30:00 AM
Widget Name Widget Id User Data Left Top Width Height Border Transparent Draw Selected Enabled Accepts Focus Runtime Allocate Normal fill Selected fill Disabled fill Template	Splash 0 0 0 222 No Border WINDOW_FILL WINDOW_FILL	>		09:30:00 AM

Figure 31. Definitions in the Azure RTOS GUIX Studio Project

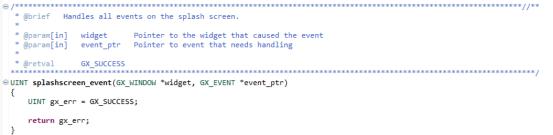
- 25. Copy and replace the files in "src" folder in e² studio project with the files in "2.23" folder in the AN folder:
 - hmi_event_handler.c
 - system_thread_entry.c

Build Thermostat_GUIX_EK_RA6M3G project you will see several warnings, but we will address them in later steps.

26. **Code highlight:** The following example creates a screen based on Widget Name in GUIX project and attached it to the root window. In this case, it is the "Splash" screen. Refer to system_thread_entry.c for more details.

```
/* Create the widget and attached to root window.*/
gx_err = gx_studio_named_widget_create("Splash", (GX_WIDGET *) p_root, (GX_WIDGET **) &p_splash_screen);
if(GX_SUCCESS != gx_err)
{
     APP_ERR_TRAP(FSP_ERR_ASSERTION);
}
```

27. **Code highlight:** An event function associated with a screen needs to be defined to handle events on that screen. Refer to hmi_event_handler.c for more details. All event functions are empty at this point.



28. Get your EK-RA6M3G ready to run the project. Connect LCD board to **Graphics Expansion** connector on EK-RA6M3 as shown below.



Figure 32. Connecting LCD Board to Graphics Expansion Connector of EK-RA6M3

29. Connect EK-RA6M3G kit to your PC using **J10**. **Download and Run** Thermostat_GUIX_EK_RA6M3G project, you will see a black screen.

30. Add the following code to **splashscreen_event** function in **hmi_event_handler.c** to show Splash screen. **Build** the e² studio project.

```
switch (event_ptr->gx_event_type)
{
    case GX_EVENT_SHOW:
        gx_err = gx_window_event_process(widget, event_ptr);
        if(GX_SUCCESS != gx_err) {
            while(1);
        }
            break;
    default:
        gx_err = gx_window_event_process(widget, event_ptr);
        if(GX_SUCCESS != gx_err) {
            while(1);
        }
        break;
}
```

Please refer to splashscreen_event function in hmi_event_handler.c in "**2.28**" folder in the AN folder. 31. **Download and Run** the project, you will see the Splash screen on LCD panel.



Figure 33. Splash Screen View on LCD

3. Using GUIX Widget Timer to Trigger a Screen Transition

3.1 Overview

In this section, you will implement a simple use of GUIX Widget timer, which is to trigger a screen transition.

3.2 Procedural Steps

 Copy and replace the files in "src" folder in e² studio project with the files in "3.1" folder in the AN folder: hmi_event_handler.c

system_thread_entry.c

2. **Code highlight**: The following code in splashscreen_event function starts a GUIX Widget timer and trigger a screen transition that hides Splash screen and shows Main Page screen.

```
switch (event_ptr->gx_event_type)
    {
        case GX EVENT TIMER:
            gx_system_timer_stop(widget, 10);
            toggle_screen(p_mainpage_screen,p_splash_screen);
            break;
        case GX EVENT SHOW:
            gx_system_timer_start(widget, 10 , SPLASH_TIMEOUT,
SPLASH TIMEOUT);
            qx err = qx window event process(widget, event ptr);
            if(GX_SUCCESS != gx_err) {
                while(1);
            }
            break;
        default:
            gx_err = gx_window_event_process(widget, event_ptr);
            if(GX_SUCCESS != gx_err) {
                while(1);
            }
            break;
    }
```

3. **Build, Download, and Run** the project, you will see the transition from Splash screen to Main Page screen in about 3 seconds.

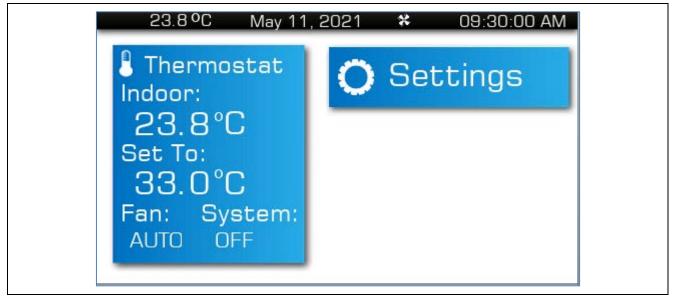


Figure 34. Main Page Screen

4. Add Touch Driver to Thermostat_GUIX_EK_RA6M3G Project

4.1 Overview

In this section, you will add the ft5x06 touch driver to the project to handle touch events on LCD panel.

4.2 Procedural Steps

1. In Thermostat_GUIX_EK_RA6M3G project, create a folder by right-clicking "src", then select "New->Folder".

📴 New Folde	r					\times
Folder Create a new f	older resource.					
Enter or select	the parent folder	:				
Thermostat_	GUIX_EK_RA6M30	6/src				
 Image: Image: Im	oug x_studio cfg gen ipt	RA6M3G [Deb	ıg]			
Folder name:	touch_ft5x06					
Advanced >						
?				Finish	Can	cel

Figure 35. Creating New Folder in Thermostat_GUIX_EK_RA6M3G Project Click Finish to create "touch_ft5x06" folder. 2. Copy touch_ft5x06.c and touch_ft5x06.h from "touch_ft5x06" folder in the Lab folder to the one in e² studio project.

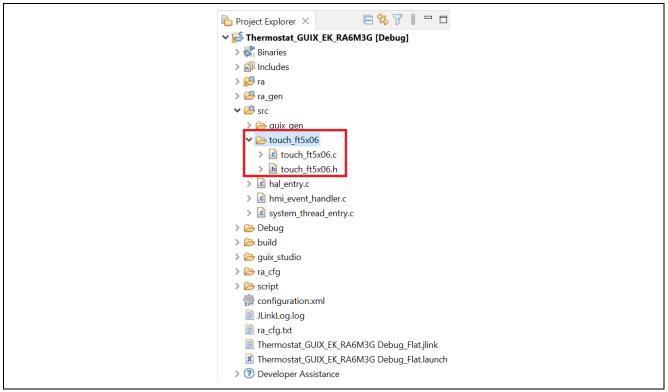


Figure 36. Copying files to the e² studio Project

3. Open project configuration and create **Touch Thread** with the settings below.

Threads	🛃 New Thread 🔊 Remove	E Touch	Thread Stacks
♥ ♥ @ Sy ♥	AL/Common g_ioport I/O Port (r_ioport) stem Thread <u>Azure RTOS</u> GUIX uch Thread	0	Add stacks to the selecte from the clipboard.
Objects	New Object > 🔟 Remo	/e	
	ies 🗙 🖹 Problems 📮 Console 🏶 Smart Browser 🤑 Smart	Manual 🤗 🤅	Search 🚺 Memory
Touch Touch Touch Touch Touch	Property	Manual 🥠 S	Search 🚺 Memory Value
Touch T	Property V Common > General	Manual 🛷 S	
Touch T	Property Common Seneral Timer	Manual 🛷 S	
Touch T	Property ✓ Common > General > Timer > Trace	Manual 🔗 S	
Touch T	Property Common General Timer Trace Performance	Manual 🔗 S	
Touch T	Property Common General Timer Trace Performance RA	Manual 🔗 S	
Touch T	Property Common General Timer Trace Performance RA Interrupts	Manual 🔗 S	
Touch T	Property Common General General Timer Frace Performance RA Interrupts Thread	Manual 🔗 S	Value
Touch T	Property Common General Timer Tirace Performance RA Interrupts Thread Symbol	Manual 🔗 S	Value touch_thread
Touch T	Property Common General General Timer Trace Performance RA Interrupts Thread Symbol Name	Manual 🔗 S	Value touch_thread Touch Thread
Touch T	Property Common General Timer Trace Performance RA Interrupts Thread Symbol Name Stack size (bytes)	Manual 🔗 S	Value touch_thread Touch_Thread 1024
Touch T	Property Common General General Timer Trace Performance RA Interrupts Thread Symbol Name	Manual 🔗 S	Value touch_thread Touch Thread

Figure 37. Creating Touch Thread

- 4. The pins marked in red below are used for touch panel controller on the LCD board:
 - IRQ0 interrupt (P206) is used to trigger touch events.
 - I2C channel 2 (P512, P511) is used to read and write data to the touch controller. P304 is used to reset the touch controller.

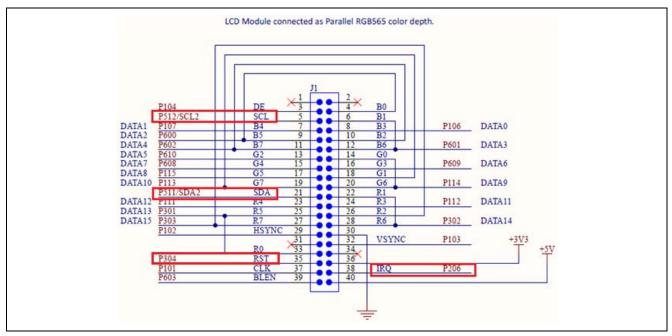


Figure 38. Pins Used in Touch Panel Controller (marked in red)

5. Since the IRQ0 (P206) needs a pull-up to function properly with the LCD board, do not change the setting that was done by FSP, as shown below.

Select Pin Configuration				📑 Export to CSV fil	e 🔚 Configure Pin Driv	/er Warnings
RA6M3G-EK.pincfg	✓ <u>Ma</u>	inage configurations		🗹 Generate d	ata: g_bsp_pin_cfg	
Pin Selection	⊞ ⊑ ↓ <mark>a</mark> z	Pin Configuration				
Type filter text		Name		Value	Link	
✓ ✓ P2		Symbolic Name				
P200		Comment				
P201		Mode		Input mode		
✓ P202		Pull up		input pull-up		
✓ P203		IRQ		IRQ0-DS		
✓ P204		Drive Capacity		Low		
✓ P205		Output type		CMOS		
✓ P206		Input/Output				
P207		P206		GPIO	\rightarrow	
✓ P208						
✓ P209						
✓ P210						
✓ P211		Module name:	P206			
✓ P212		Port Capabilities:	BUS0: WAIT			
✓ P213	×		CTSU0: TS01			
in Function Pin Number						

Figure 39. FSP Setting

6. In e² studio project configuration, add **External IRQ Driver on r_icu** to **Touch Thread** with the following settings.

Threads		된 New Thread 💼 Remove 📒	Touch T	hread Stacks
		A New Miedo a Nemove	ioucii i	meau stacks
	AL/Common		🕀 a	touch_irq External IRQ
	g_ioport I/O Port (r_ioport)			icu)
	ystem Thread			
	Azure RTOS GUIX		(i)	
	ouch Thread			
4	g_touch_irq External IRQ (r_icu)			
		D		
Objects		된 New Object > 💼 Remove		
Summary	BSP Clocks Pins Interrupts Event	t Links Stacks Components		
Proper	rties 🛛 🖹 Problems 📮 Console	Smart Browser 🛄 Smart N	1anual 🛷 Se	arch
g_exter	nal_irq0 External IRQ (r_icu)			
Settings	Property	V	alue	
-	Property ✓ Common	v	alue	
Settings API Info			alue efault (BSP)	
-	✓ Common	D		
-	✓ Common Parameter Checking	D D		
-	Common Parameter Checking Module g_external_irq0 External	D D	efault (BSP) _touch_irq	
-	Common Parameter Checking Module g_external_irq0 Extern Name	bal IRQ (r_icu) g 0	efault (BSP) _touch_irq	
-	 Common Parameter Checking Module g_external_irq0 External Name Channel Trigger Digital Filtering 	D al IRQ (r_icu) 9 0 Fr Ei	efault (BSP) _touch_irq alling nabled	
-	 Common Parameter Checking Module g_external_irq0 External Name Channel Trigger Digital Filtering Digital Filtering Sample Clo 	al IRQ (r_icu) 9 9 0 Fr Er bock (Only valid when Digital Filte Pr	efault (BSP) _touch_irq alling nabled CLK / 64	
-	 Common Parameter Checking Module g_external_irq0 External Name Channel Trigger Digital Filtering Digital Filtering Sample Cloo Callback 	al IRQ (r_icu) 9 9 6 Fi Er bock (Only valid when Digital Filte P to	efault (BSP) touch_irq alling nabled CLK / 64 puch_irq_cb	
-	 Common Parameter Checking Module g_external_irq0 Extern Name Channel Trigger Digital Filtering Digital Filtering Sample Clo Callback Pin Interrupt Priority 	al IRQ (r_icu) 9 9 6 Fi Er bock (Only valid when Digital Filte P to	efault (BSP) _touch_irq alling nabled CLK / 64	
-	 Common Parameter Checking Module g_external_irq0 External Name Channel Trigger Digital Filtering Digital Filtering Sample Cloo Callback 	al IRQ (r_icu) g 0 Fr Er bock (Only valid when Digital Filte P to Pi	efault (BSP) touch_irq alling nabled CLK / 64 puch_irq_cb	

Figure 40. Adding External IRQ Driver on r_icu to Touch Thread

7. In project configuration, add I2C Master Driver on r_iic_master to Touch Thread with below settings.

Stacks (Configuration				Generate Project Content
Threads		횐 New Thread 🔋 Remove 📃	Touch Thread Stacks	🔊 New Stack	< > 🚢 Extend Stack > 🔊 Remove
	L/Common g_joport I/O Port (r_joport) stem Thread Azure RTOS GUIX uch Thread g_ stouch_irg External IRQ (r_ g_j2c_touch I2C Master (r_jii		g_touch_irq External IRQ (r_icu)	g_j2c_touch 12C Master (r_iic g_j2c_touch 12C Master (r_iic g_transfer0 Transfer (r_dtc) IIC2 TXI (Transmit data empty) g_	g_transfer1 Transfer (r_dtc) IIC2 RXI (Receive data full)
Propert		s Event Links Stacks Components Console 🏶 Smart Browser 🖽 Sm master)		у	
Settings API Info	Common Parameter Checking DTC on Transmissio 10-bit slave address Module g_i2c_touch I2 Name Channel Rate Rise Time (ns) Fall Time (ns)	n and Reception sing	Default (BSP) Enabled Disabled g_j2c_touch 2 Fast-mode 120 120		
	Duty Cycle (%) Slave Address Address Mode Timeout Mode Timeout during SCL Callback Interrupt Priority Lee ✓ Pins SDA SCL		50 0x38 7-8it Short Mode Enabled touch_12c_callback Priority 6 P511 P512		

Figure 41. Adding I2C Master Driver on r_iic_master to Touch Thread

8. In project configuration, add **Touch Semaphore** as shown below. We use this semaphore to signal the Touch thread when a touch event occurs. The Touch thread then sends the touch event to GUIX.

Stacks Configuration	Generate Project Conte
Threads New Thread Remove # HAL/Common # g_ioport I/O Port (r_ioport) # System Thread # Azure RTOS GUIX Touch Thread # g_touch_irq External IRQ (r_icu) # g_i2c_touch I2C Master (r_iic_master > > Objects Particular Semaphore g_touch_semaphore Semaphore	Touch Thread Stacks Image: Remove Image: g_touch_irq External IRQ (r_icu) Image: g_i2c_touch I2C Master (r_icu) Image: g_touch_irq External IRQ (r_icu) Image: g_i2c_touch I2C Master (r_icu) Image: g_touch_irq External IRQ (r_icu) Image: g_i2c_touch I2C Master (r_icu) Image: g_touch_irq External IRQ (r_icu) Image: g_i2c_touch I2C Master (r_icu) Image: g_transfer (r_dtc) IIC2 TXI (Transmit data empty) Image: g_i2c_touch I2C Master (r_icu)
Summary BSP Clocks Pins Interrupts Event Link	
g_new_semaphore0 Semaphore	Smart Browser 🔑 Smart Manual 🔗 Search 🏮 Mernory
Settings Property Name Symbol Initial count	Value Touch Semaphore g_touch_semaphore 0

Figure 42. Adding Touch Semaphore

9. In project configuration, add **I2C Semaphore** as shown below. This semaphore is used in the ft5x06 driver to trigger data reading when a touch-panel interrupt occurs.

Stacks Co	onfiguration	Generate Project Content
Threads		Touch Thread Stacks
 System A Touce G G G Objects 	_ioport I/O Port (r_ioport) em Thread zure RTOS GUIX	Image: Stacks Remove Image: g_touch_irq External IRQ (r_icu) Image: g_touch_irq External IRQ (r_icu) Image: g_transfer (r_icu) Image: g
Summary BS	SP Clocks Pins Interrupts Event Links	s Stacks Components
	5 × 🖹 Problems 📮 Console 🌸 naphore0 Semaphore	Smart Browser 🔑 Smart Manual 🔗 Search 🔋 Memory
	Property Name Symbol Initial count	Value I2C Semaphore g_i2c_semaphore 0

Figure 43. Adding I2C Semaphore

= touch_data.point[0].y;

- 10. In RA Configurator, click Generate Project Content to generate project content.
- 11. Copy and replace the files in "**src**" folder in the e² studio project with the files in "**4.11**" folder in the AN folder:
 - hmi_event_handler.c
 - system_thread_entry.c
 - touch_thread_entry.c

12. Code highlight: Below code in touch_thread_entry.c get touch data and send touch event to GUIX.

```
/* Get touch data from the FT5X06 */
```

```
ft5x06_payload_get(&touch_data);
```

```
/* Send touch data*/
```

```
if(1 == touch_data.num_points)
```

```
{
```

```
gxe.gx_event_payload.gx_event_pointdata.gx_point_x = touch_data.point[0].x;
```

```
gxe.gx_event_payload.gx_event_pointdata.gx_point_y
gxe.gx_event_type = GX_EVENT_PEN_DOWN;
```

```
gx_system_event_send(&gxe);
```

```
}
```

```
else if (GX_EVENT_PEN_DOWN == gxe.gx_event_type)
```

```
{
```

```
gxe.gx_event_type = GX_EVENT_PEN_UP;
```

gx_system_event_send(&gxe);

}

13. All the screens designed in the Azure RTOS GUIX Studio project are now created in system_thread_entry.c

```
/* Create a screen and attached it to root window.*/
gx_err = gx_studio_named_widget_create("Splash", (GX_WIDGET *) p_root, (GX_WIDGET **) &p_splash_screen);
if(GX_SUCCESS != gx_err)
{
    APP_ERR_TRAP(FSP_ERR_ASSERTION);
}
gx_err = gx_studio_named_widget_create ("Settings", GX_NULL, (GX_WIDGET **) &p_settings_screen);
if(GX_SUCCESS != gx_err)
{
    APP ERR TRAP(FSP ERR ASSERTION);
gx_err = gx_studio_named_widget_create ("MainPage", GX_NULL, (GX_WIDGET **) &p_mainpage_screen);
if(GX_SUCCESS != gx_err)
{
   APP_ERR_TRAP(FSP_ERR_ASSERTION);
}
gx_err = gx_studio_named_widget_create ("Thermostat", GX_NULL, (GX_WIDGET **) &p_thermostat_screen);
if(GX_SUCCESS != gx_err)
{
    APP_ERR_TRAP(FSP_ERR_ASSERTION);
3
gx_err = gx_studio_named_widget_create ("Help", GX_NULL, (GX_WIDGET **) &p_help_screen);
if(GX_SUCCESS != gx_err)
ł
    APP_ERR_TRAP(FSP_ERR_ASSERTION);
}
```

The code marked in red in hmi_event_handler.c handle touch event when Thermostat button and Settings button are clicked. Refer to hmi_event_handler.c for more details.



14. Build, Download, and Run the e² studio project. Then, you will be able to go back and forth from the Main Page screen to Thermostat screen and Settings screen using Thermostat and Settings buttons on Main Page screen and "Back" button on the other two screens.

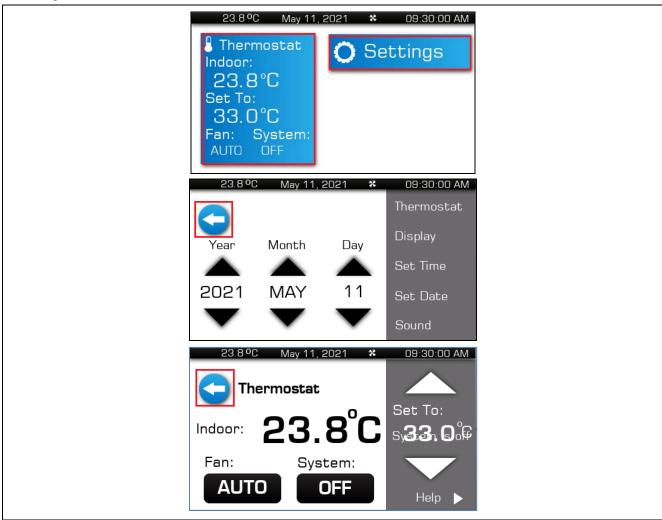


Figure 44. Navigating between Main Page Screen and Thermostat Screen

5. Control LCD Backlight

5.1 Overview

In this section, you will use a PWM output pin of a GPT timer to control the intensity (brightness) of LCD backlight.

5.2 Procedural Steps

1. In LCD board schematics below, the LCD_BLEN signal, which is connected to the P603 on the RA6M3 MCU, is configured in PWM mode to control the intensity of LCD backlight.

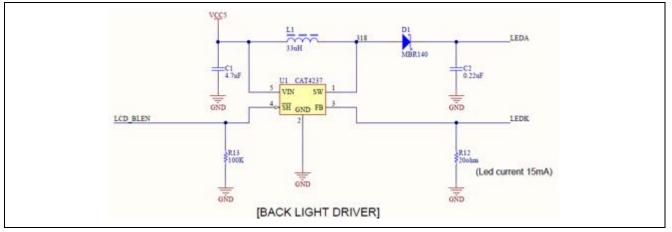


Figure 45. LCD Board Schematic

2. To configure P603 in PWM output mode, we disable it in Pin Configuration at first. **Save this change before moving to the next step.**

Select Pin Configuration			🔛 Export to CSV f	ile 🔚 Configure Pin Driver	Warnings
RA6M3G-EK.pincfg	✓ Ma	nage configurations	🗹 Generate	data: g_bsp_pin_cfg	
Pin Selection	⊞ ⊟ ↓ <mark>a</mark>	Pin Configuration			
Type filter text	^	Name Symbolic Name Comment	Value	Link	
> 🗸 P4		Mode	Disabled		
> V P5		Pull up	None		
✓ ✓ P6		Drive Capacity	Low		
✓ P600 ✓ P601		Output type	CMOS		
✓ P601		✓ Input/Output			
		P603	None	\Rightarrow	
P604					
P605					
P606					
P607					
✓ P608					
✓ P609		Module name: P603			
✓ P610		Port Capabilities: BUS0: D13 GPT7: GTI			

Figure 46. Disabling P603 in Pin Configuration

3. In Pin Configuration, set P603 as GPT7 GTIOCA output.

Select Pin Configuration			Export to CSV file	Configur	e Pin Driver	Warnings
RA6M3G-EK.pincfg	✓ Mat	nage configurations	🗹 Generate data:	g_bsp_pi	n_cfg	
Pin Selection	₽ ₽ ↓ <mark>a</mark>	Pin Configuration				
Type filter text		Name	Value	Lock	Link	
✓ ✓ Timer:GPT		Pin Group Selection	Mixed			
✓ ✓ Timer.OPT ✓ GPT0		Operation Mode	GTIOCA or GTIOCB			
GPT0 GPT1		✓ Input/Output				
GPT2		GTIOCA	✓ P603	e e e e e e e e e e e e e e e e e e e	\Rightarrow	
GPT2 GPT3		GTIOCB	None			
GPT3 GPT4						
GPT5						
GPT6						
GPT6 ✓ GPT7						
GPT8						
GPT9						
GPT10						
GPT11		Module name: GPT7				
GPT12 GPT13						

Figure 47. Setting P603 as GPT7 GTIOCA Output in Pin Configuration

4. In project configuration, add **Timer Driver on r_gpt** to **System Thread** with below settings.

Threads		횐 New Thread 😰 Remove	🖻 g_tin	ner0 Timer, General PWN	/l (r_gpt) Stacks
 ✓ Ø Sy Ø Objects Summary 	AL/Common g_joport I/O Port (r_joport) stem Thread Azure RTOS GUIX g_timer_PWM Timer, General P BSP_Clocks_Pins_Interrupts_E ties XProblemsCon	New Ohiert > Remo	ts		
	0 Timer, General PWM (r				
Settings	Property Common		Value		
API Info	Parameter Checking		Defaul	t (BSP)	
	Pin Output Support		Enable	d	
	Write Protect Enable		Disabl	ed	
	✓ Module g_timer_PWM Tim	ner, General PWM (r_gpt)			
	✓ General				
	Name		g_time	er_PWM	
	Channel		7		
	Mode		Saw-w	ave PWM	
	Period		50000		
	Period Unit		Nanos	econds	
	✓ Output				
	> Custom Waveform				
		only applicable in PWM mode)	10		
	GTIOCA Output Ena	bled	True		
	GTIOCA Stop Level		Pin Lev	/el Low	
	GTIOCB Output Ena	bled	False		
	GTIOCB Stop Level		Pin Lev	and the second sec	

Figure 48. Adding Timer Driver on r_gpt to System Thread

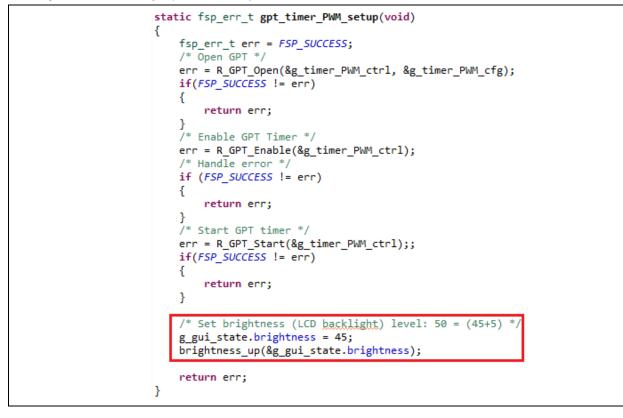
Even though the duty cycle of PWM output is purposely set to **10%** here, it will be changed to **50%** later in the code.

0

- 5. In RA Configurator, click Generate Project Content to generate project content.
- 6. Copy and replace the files in "**src**" folder in e² studio project with the files in "**5.6**" folder in the AN folder:
 - hmi_event_handler.c
 - system_thread_entry.c
 - brightness.c
 - brightness.h
 - system_api.h
 - system_cfg.h
- 7. brightness_up and brightness_down functions in brightness.c are used to set the PWM duty cycle, as shown below:

```
/* Get the current period setting. */
R_GPT_InfoGet(&g_timer_PWM_ctrl, &info);
/* Calculate the desired duty cycle based on the current period. */
duty_cycle_count = (uint32_t) ((info.period_counts *
brightness)/GPT_PWM_MAX_PERCENT);
err = R_GPT_DutyCycleSet(&g_timer_PWM_ctrl, duty_cycle_count,
GPT_IO_PIN_GTIOCA);
```

8. Looking at gpt_timer_PWM_Setup function in system_thread_entry.c, you will see brightness (duty cycle of PWM output) is set to 50 percent.



9. **Build, Download, and Run** the e² studio project. By clicking the **Settings** button on **Main Page** screen, you can access **Settings** screen.

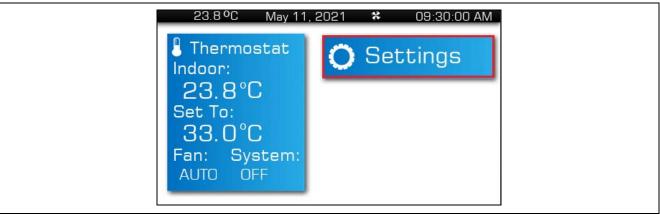


Figure 49. Settings Button on Main Page Screen

10. PWM output measured on pin P603 with brightness is set to 50%.

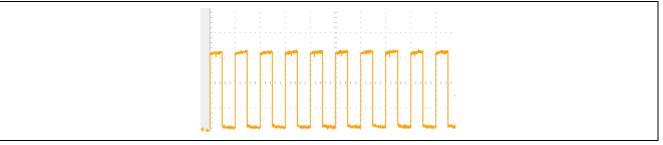


Figure 50. PWM Output on P603 at 50% Brightness

11. Click "**Display**" menu on **Settings** screen, you can use "**Up**" and "**Down**" buttons to change the brightness of LCD backlight.



Figure 51. Display on Settings Screen

12. PWM output measured on pin P603 after changing brightness to **65%**.

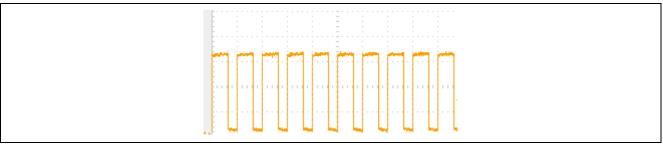


Figure 52. PWM Output on P603 at 65% Brightness

6. Update Date/Time and Temperature

6.1 Overview

In this section, you will enable RTC controller as a timekeeper and one ADC channel to read the MCU die's temperature sensor and use it as Thermostat temperature data.

6.2 Procedural Steps

1. In project configuration, create **Temperature Time Thread**.

Threads	🔕 New Thread 🔊 Remove	E Temper	ature Time Thread Sta	acks
🗸 🚀 НА	L/Common		Add stacks to the sele	lected
49	g_ioport I/O Port (r_ioport)		clipboard.	
🗸 🍈 Sys	stem Thread			
49	Azure RTOS GUIX			
+	g_timer_PWM Timer, General PWM (r_gp	t)		
🗸 🍈 Tor	uch Thread			
49	g_touch_irg External IRQ (r_icu)			
	g i2c touch I2C Master (r iic master)			
Ter	nperature Time Thread			
Objects	🗐 New Object > 驞 Remo	NO		
-	_ , _	ve		
	ch_semaphore Semaphore			
😑 g_i2c_	semaphore Semaphore			
Summary	BSP Clocks Pins Interrupts Event Links	Stacks Comp	onents	
🔲 Propert	ies 🔀 Problems 📮 Console 🁒	Smart Browser	🤑 Smart Manual 🛷	Sear
Tempera	ture Time Thread			
	Property	Value		
Settings	Property	Value		
Settings	> Common	Value		
Settings	> Common		ture time thread	
Settings	> Common ✓ Thread Symbol	tempera	ture_time_thread ture_Time_Thread	
Settings	> Common	tempera	ture_time_thread ture Time Thread	
Settings	> Common Thread Symbol Name	tempera Tempera		
Settings	 Common Thread Symbol Name Stack size (bytes) 	tempera Tempera 1024		

Figure 53. Create Temperature Time Thread

2. In project configuration, add **RTC Driver on g_rtc** to **Temperature Time Thread**.

🗸 🏈 ни	AL/Common		
	g_ioport I/O Port (r_ioport)	g_rtc Realtime Cl	lock (r_rtc)
	stem Thread		
	Azure RTOS GUIX	(i)	
	g_timer_PWM Timer, General PWM (r_gpt)		
	uch Thread		
	g_touch_irq External IRQ (r_icu)		
	g_i2c_touch I2C Master (r_iic_master)		
🗙 🍪 Te	mperature Time Thread		
	g_rtc Realtime Clock (r_rtc)		
Objects	🕢 New Object > 🔬 Remove		
	ich_semaphore Semaphore		
■ g_i2c	_semaphore Semaphore		
Summary	BSP Clocks Pins Interrupts Event Links Stac	cks Components	
			uud 🥔 Saarah
Propert	ties 🗙 🖹 Problems 📮 Console 🏟 Smart		nual 🛷 Search
Propert			nual 🛷 Search
Propert	ies × 💽 Problems 📮 Console 🐳 Smart ealtime Clock (r_rtc)		nual 🤗 Search
Propert g_rtc0 R Settings	ties × Problems Console Smart Clock (r_rtc) Property	Browser 🤑 Smart Mar	nual 🤗 Search
Propert	ies × 💽 Problems 📮 Console 🐳 Smart ealtime Clock (r_rtc)	Browser 🤑 Smart Mar	nual 🔗 Search
Propert g_rtc0 R Settings	ties × Problems © Console ealtime Clock (r_rtc) Property > Common V Module g_rtc0 Realtime Clock (r_rtc)	Browser 🔑 Smart Mar Value	nual 🔗 Search
Propert g_rtc0 R Settings	ties × Problems © Console ealtime Clock (r_rtc) Property > Common V Module g_rtc0 Realtime Clock (r_rtc)	Browser 🤑 Smart Mar	nual 🔗 Search
Propert g_rtc0 R Settings	ties × Problems Console Smart ealtime Clock (r_rtc) Property > Common Module g_rtc0 Realtime Clock (r_rtc) Name	Browser Smart Mar Value g_rtc LOCO	nual 🔗 Search
Propert g_rtc0 R Settings	ites X Problems ☐ Console Smart ealtime Clock (r_rtc) Property > Common Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source	Browser Smart Mar Value g_rtc LOCO	nual 🔗 Search
Propert g_rtc0 R Settings	ites X Problems Console Froperty Common Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source Frequency Comparision Value (LOCO)	Browser Smart Mar Value <u>g_rtc</u> LOCO 255	nual 🔗 Search
Propert g_rtc0 R Settings	ealtime Clock (r_rtc) Property Common Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source Frequency Comparision Value (LOCO) Automatic Adjustment Mode	Browser Smart Mar Value <u>g_rtc</u> LOCO 255 Enabled	nual 🔗 Search
Propert g_rtc0 R Settings	ealtime Clock (r_rtc) Property > Common ✓ Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source Frequency Comparision Value (LOCO) Automatic Adjustment Mode Automatic Adjustment Period	Browser Smart Mar Value UCCO 255 Enabled 10 Seconds	nual 🔗 Search
Propert	 Problems Console Smart Property Common Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source Frequency Comparision Value (LOCO) Automatic Adjustment Mode Automatic Adjustment Period Adjustment Type (Plus-Minus) 	Browser Smart Mar Value 9.rtc LOCO 255 Enabled 10 Seconds NONE	nual 🔗 Search
Propert g_rtc0 R Settings	 Problems Console Smart Property Common Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source Frequency Comparision Value (LOCO) Automatic Adjustment Mode Automatic Adjustment Period Adjustment Type (Plus-Minus) Error Adjustment Value 	Browser Smart Mar Value g_rtc LOCO 255 Enabled 10 Seconds NONE 0	nual 🔗 Search
Propert	 Problems Console Start Property Common Module g_rtc0 Realtime Clock (r_rtc) Name Clock Source Frequency Comparision Value (LOCO) Automatic Adjustment Mode Automatic Adjustment Period Adjustment Type (Plus-Minus) Error Adjustment Value Callback 	Browser Smart Mar Value g_rtc LOCO 255 Enabled 10 Seconds NONE 0 time_update_callback	nual 🔗 Search

Figure 54. Adding RTC Driver on g_rtc to Temperature Time Thread

3. In project configuration, add **ADC Driver on r_adc** to **System Thread**.

Threads New Thread Remove g_adc ADC (r_adc) Stacks > @ Touch Thread > @ Temperature Time Thread @ g_adc ADC (r_adc) < > > Summary BSP Clocks Pins Interrupts Event Links Stacks Components					
🖹 Problems 📮 Console 🔲 Properties 🗙 🌸 Smart Browser 🐺 Smart Manual 🔋 Memory					
g_adc ADC (r_adc)					
Settings	Property	Value			
API Info	✓ Common				
Arrino	Parameter Checking	Default (BSP)			
	 Module g_adc ADC (r_adc) 				
	✓ General				
	Name g_adc				
	Unit	0			
	Resolution	12-Bit			
	Alignment	Right			
	Clear after read	On			
	Mode	Continuous Scan			
	Double-trigger	Disabled			
	> Input				
	> Interrupts				
	> Extra				

Figure 55. Adding ADC Driver on r_adc to System Thread

4. Select **Temperature Sensor** as input source for g_adc module.

g_adc A	DC (r_adc)	
Settings	Property	Value
API Info	✓ Common	
7471110	Parameter Checking	Default (BSP)
	 Module g_adc ADC (r_adc) 	
	> General	
	✓ Input	
	 Channel Scan Mask (channel availability varies by MCU) 	
	Channel 0	
	Channel 1	
	Channel 2	
	Channel 3	
	Channel 4	
	Channel 5	
	Channel 6	
	Channel 7	
	Channel 8	
	Channel 9	
	Channel 10	
	Channel 11	
	Channel 12	
	Channel 13	
	Channel 14	
	Channel 15	
	Channel 16	
	Channel 17	
	Channel 18	
	Channel 19	
	Channel 20	
	Channel 21	
	Channel 22	
	Channel 23	
	Channel 24	
	Channel 25	
	Channel 26	
	Channel 27	
	Channel 28	
	Temperature Sensor	
	Internal Reference Voltage	
	> Group B Scan Mask (channel availability varies by MCU)	
	> Addition/Averaging Mask (channel availability varies by	V

Figure 56. Selecting Temperature Sensor as Input Source for g_adc

5. Create **g_timer_semaphore** with the following settings. We use this semaphore to trigger the date and time update every second.

Objects New Object > Remove • g_i2c_semaphore Semaphore • g_touch semaphore Semaphore • g_timer_semaphore Semaphore Semaphore • g_timer_semaphore Semaphore Semaph					
Summary BSP Clocks Pins Interrupts Event Links Stacks Components					
👔 Problems 📮 Console 🔲 Properties 🔀 🁒 Smart Browser 🐺 Smart Manual 🔅 Debug					
g_timer_semaphore Semaphore					
Settings Property Value					
Name Timer Semaph					
Symbol g_timer_sema	phore				
Initial count 0					

Figure 57. Creating g_timer_semaphore

O

- 6. In RA Configurator, click ^{Generate Project Content} to generate project content.
- 7. Copy and replace the files in "**src**" folder in e² studio project with the files in "**6.6**" folder in the Lab folder:
 - hmi_event_handler.c
 - system_thread_entry.c
 - system_time.c
 - system_time.h
 - system_api.h
- 8. In System Thread, date/time data and temperature data get updated every second. It then sends out events to trigger GUIX updates.
- 9. The following is an example of handling temperature and time update events in the Main Page screen event handler.
- 10. **Build, Download, and Run** the e² studio project. You will see time and temperature get updated every second.

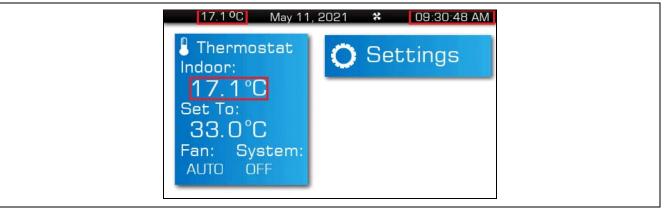


Figure 58. Time, Temperature on Main Page Screen

7. Setting Date/Time in A Full Function Project

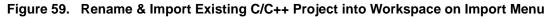
7.1 Overview

In this section, you will import and run the complete Thermostat project that enables the settings of date and time. Upon user press date and time buttons on the settings screen, a message will be sent to the system thread to update the date and time, then the system thread will send a GUIX event to trigger time display update on screens.

7.2 Procedural Steps

 You can try the completed project in "completed_project" folder that has a full function Thermostat application. Use "Rename & Import Existing C/C++ Project into Workspace" feature of Import menu in e² studio to do so since you already had a project with the same in the workspace.

			Import -	
			Select	V
			Rename and Import and Existing C/C++ Project into the workspace	23
File	Edit Navigate Search Project	Renesas Views R	Select an import wizard:	
	New	Alt+Shift+N >	type filter text	
	Open File		← 🦢 General	
	Open Projects from File System		CMSIS Pack	
	Recent Files	>	Existing Projects into Workspace File System	
	Close Editor	Ctrl+W	Preferences	
	Close All Editors	Ctrl+Shift+W	Projects from Folder or Archive	
	Save	Ctrl+S	Renesas CS+ Project for CA78K0R/CA78K0	
	Save As		Renesas CS+ Project for CC-RX and CC-RL > > C/C++	
R	Save All	Ctrl+Shift+S	> 🧀 Install	
	Revert		> > > Oomph > > Run/Debug	
	Move		> 🗁 Team	
	Rename	F2	> > > Tracing	
88	Refresh	F5		
	Convert Line Delimiters To	>		
۵	Print	Ctrl+P		
\geq_1	Import Import			
4	Export			
	Properties	Alt+Enter		
	Switch Workspace	>		
	Restart		? < Back Next > Finish	
			(2) < Back Next > Finish	Cancel



8. Website and Support

Visit the following URLs to learn about key elements of the RA family, download components and related documentation, and get support:

RA Product Information RA Product Support Forum RA Flexible Software Package Renesas Support renesas.com/ra renesas.com/ra/forum renesas.com/FSP renesas.com/support

Revision History

		Descript	Description	
Rev.	Date	Page	Summary	
1.00	May.13.24	—	Initial release	
1.01	Jul.17.23	—	Updated for FSP v4.4.0.	
1.02	Jun.03.24	—	Updated for FSP v5.2.0.	

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. 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 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.
- 2. 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.
- 5. 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 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, 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. Renesas 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: www.renesas.com/contact/.