

Renesas RA Family

"GUIX Thermostat" for EK-RA8D1 Parallel GLCD Display

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 integrated 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 Version: 2024-04 (24.4.0) or greater
- Renesas Flexible Software Package (FSP) v5.4.0.
- Azure RTOS GUIX Studio V6.4.0.

Hardware

- Renesas EK-RA8D1 kit (RA8D1 MCU Group)
 - ER-TFT043-3 with Capacitive Touch Panel 40 pins connection
 - Recommended user should use RA6M3G kit's LCD.
- Renesas EK-RA8D1's "SW1" switches setting.
- Renesas-app-lcd-conv_v1_b_mfg order from the link: <u>https://oshpark.com/shared_projects/pzfp0mCD</u>
 - User needs to click to "Actions" button to order LCD converter board. Refer to section "2". Step 2.2.28.



Reference Manuals

- RA Flexible Software Package Documentation Release v5.4.0
- Azure RTOS GUIX and GUIX Studio v6.4.0.0
- Renesas RA8D1 Group User's Manual Rev.1.1 .0
- EK-RA8D1-v1.0 Schematics

Purpose

This document will guide you through the setup of an Azure RTOS GUIX touch screen interface Thermostat application in e² studio. This document will show how to configure the drivers and library included with the FSP. These will allow you to set up the parallel LCD Display Controller, the touch screen driver, and semaphores, queue, and Mutex to communicate with application tasks. It also shows the steps necessary to create a simple GUI interface using the Azure RTOS GUIX Studio editor. In addition, this app note will also cover project setup along with basic debugging operations. When it is running, the application will respond to touchscreen actions, presenting a basic graphical user interface (GUI).

Intended Audience

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



Contents

1 1.1 1.2	Download and Installing Tools
2. 2.1 2.2	Create the Application Project and Enable Backlight
3.	Using GUIX Widget Timer to Trigger a Screen Transition
3.1	Overview
4. 4.1 4.2	Add Touch Driver to Thermostat_GUIX_EK_RA8D1_GLCD Project
5.	Control LCD Backlight
5.1	Overview
5.2	Procedural Steps
6.	Update Date/Time and Temperature
6.1	Overview
6.2	Procedural Steps
7.	Testing and debugging in A Full Function Project45
7.1	Overview
7.2	Procedural Steps45
8.	Website and Support47
Rev	ision History48



1. Download and Installing Tools

1.1 Overview

In this section, you will copy the application note (AN) materials to your PC and install e^2 studio v2024-04/FSP v5.4.0 and Azure RTOS GUIX Studio v6.4.0.0.

1.2 Procedural Steps

- 1. If you already have e² studio with FSP v5.4.0 or later installed, you can skip this step. Otherwise, you can download it from this <u>link</u>.
- 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 Download to local PC and start installing Azure RTOS GUIX Studio.

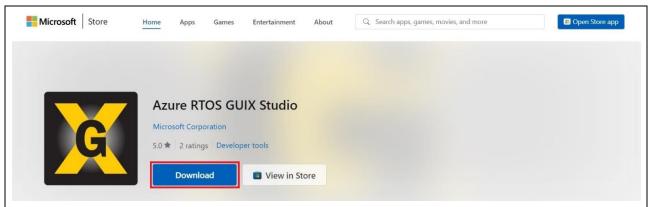


Figure 1. Get Azure RTOS GUIX Studio

4. Click downloaded file "Azure RTOS GUIX Studio installer.exe" and continue install Azure RTOS GUIX studio.





		GUIX Studi		- (0)/- 000	- Mi						1000		×
Proje	ect Edit In	isert Confi	gure He	Û	1	의	To	<u>Io</u>	a	₽©4	2	10	G
	Project	View	-										
🔝 No	o Project												
	Propert	ies View	-										
												111111	and the second s



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

2. Create the Application Project and Enable Backlight

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. This section also will show the user how to enable and use SDRAM. Setting Azure RTOS GUIX store in SDRAM.

2.2 Procedural Steps

1. Create a new RA C/C++ project. Named it Themostat_GUIX_EK_RA8D1_GLCD.

EK-RABD1 - e ¹ studio File Edit Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Run Renexas Al Window Help Source Refactor Navigate Search Project Renexas Views Source Refactor Navigate Search Project Renexas Views Source Refactor Ref	- □ × Q : ::::::::::::::::::::::::::::::::::
Create a new C/C++ project Create a new e ² studio C/C++ project Create a new e ² studio C/C++ project Create a new c/C++ Project Templates for New C/C++ Project All CMake Renesas RA C/C++ Project Cnake Renesas RA C/C++ Project Create an executable or static library C/C++ project for Renesas RA.	roject tion
Image: Control of the second secon	e Reneisas packs here Next > Emish Cancel start up

Figure 4. Create New Project

2. Select and set board to EK-RA8D1.



Renesas RA C	/C++ Project		
Device and Too	ols Selection		
Device Selection	n		
FSP Version:	5.4.0	 Board Description 	
Board:	EK-RA8D1 V		RA8D1 MCU Group
Device:	R7FA8D1BHECBD	manual, quick sta	v.renesas.com/ra/ek-ra8d1 to get kit user's art guide, errata, design package, example
Core:	CM85	projects, etc.	
Language:		Device Details	
		TrustZone Pins Processor	Yes 224 Cortex-M85
Toolchains		Debugger	
GNU ARM Em	bedded	J-Link ARM	
LLVM Embedo	ded Toolchain for Arm 7 <u>Manage Toolchain</u>		

Figure 5. Select and Set board to EK-RA8D1

3. Select Project Type, Build Artifact and Azure RTOS ThreadX.

Renesas RA C/C++ Projec Renesas RA C/C++ Projec				
Project Type Selection			1	
Project Type Selection Flat (Non-TrustZone) Renesas RA device p separation	Project oroject without TrustZone			
 All code, data and a configured in this p 	Renesas RA C/C++ Project Renesas RA C/C++ Project Build Artifact and RTOS Selection		□ ×	
 TrustZone Secure Prc Renesas RA device execution All code, data and ş will be initialized as Secure project setti linker maps and a li passed to a selecte After initialization, handler will be mas TrustZone Non-secur Renesas RA device execution All code, data and ş will be initialized as Must be associated bundle 	Build Artifact Selection Executable Project builds to an executable file	RIOS Selection	sp.5.4.0)] ~	
Non-secure startup secure code initializ		lext > Einish	Cancel	
•	~ Dark	Linear Fundar	cancer	

Figure 6. Select Azure RTOS ThreadX

4. Use Azure RTOS ThreadX - Minimal template.



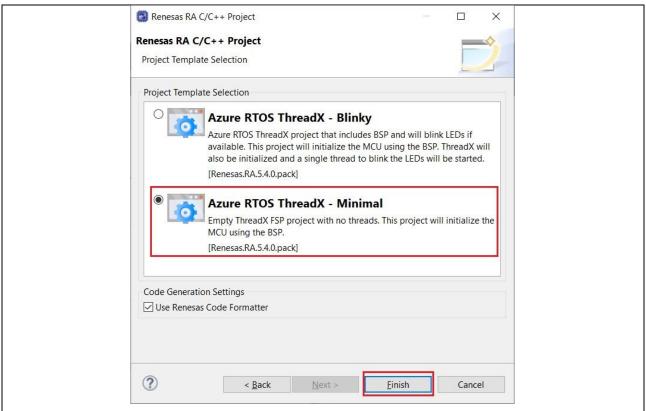


Figure 7. Selecting Azure RTOS ThreadX – Minimal Template

5. Open the project configuration and go to the **BSP** tab. Settings **SDRAM** and **Heap size (bytes)** to **0x2000**.

oard Supp	port Package Configuration				Generate Project Content	EK-RA8	01	
					Restore Defa	Cattings	Property V RA8D1	Value
							series	8
Device Select	tion						✓ RA8D1 Family	
	5.4.0	~	Board	d Details			✓ SDRAM	
FSP version:	5.4.0	~	Evalua	ation kit for RA8D1	I MCU Group		✓ Timings	
Board:	EK-RA8D1 ~	. 220			as.com/ra/ek-ra8d1 to get kit		tRAS (cycles)	6 cycles
Davian	R7FA8D1BHECBD				t guide, errata, design		tRCD (cycles)	3 cycles
Device:	R/FA8D1BHECBD			age, example proje			tRP (cycles)	3 cycles
Core:	CM85	~~					tWR (cycles)	2 cycles
RTOS:	Azure RTOS ThreadX	100					tCL (cycles)	3 cycles
NIUS.	Azure KIOS IIIreaux						tRFC (cycles)	937
							tREFW (cycles)	8 cycles
							✓ Initialization	
							Auto-Refresh Interval (ARFI)	10 cycles
							Auto-Refresh Count (ARFC)	8 times
							Precharge Cycle Count (PRC)	3 cycles
							SDRAM Support	Enabled
							Address Multiplex Shift	10-bit shift
							Endian Mode	Little Endian
							Continuous Access Mode	Enabled
							Bus Width	16-bit
							> Security	
							> OFS0 register settings	
							OFS1_SEL register settings OFS1 register settings	
							> OFS1 register settings	
							Solution Settings (BPS)	
							 Permanent Block Protection Settings (BPS) 	(DE
							 First Stage Bootloader (FSBL) 	fe r
							> Clocks	
							> Cache settings	
							Enable inline BSP IRQ functions	Enabled
							Dual Bank Mode	Disabled
							Main Oscillator Wait Time	8163 cycles
							✓ RA Common	
							Main stack size (bytes)	0x400
_						>	Heap size (bytes)	0x2000
and the second se	Clocks Pins Interrupts Event Links Stacks						MCU Vcc (mV)	3300

Figure 8. Setting SDRAM Properties



GUI Storage in MCU memory or SDRAM:

This Project is using Azure RTOS GUIX with multiple images output settings: 1 image = 480 x 272 pixels x 24bpp = approximately 3.2 MB which is larger than 2 MB of code flash memory. If it is stored in the MCU code flash memory, it does not fit. Because MCU code flash memory is limited to 2 MB. The alternative is to use the SDRAM. The SDRAM memory is available built in on board (SDRAM is 512Mbit, which is 64Mbyte). Following section 2 and Figure 12. Step # 2.2.9 Settings properties for **Graphics LCD**: "Input > Framebuffer > Section for framebuffer allocation > . sdram (it points to SDRAM and use SDRAM). After step # 2.2.29: Generate Project Content and Build Project". You will see the SDRAM initialize generated in hal_entry.c file ("R_BSP_SdramInit(true)").

#if BSP_CFG_SDRAM_ENABLED
 /* Setup SDRAM and initialize it. Must configure pins first. */
 R_BSP_SdramInit(true);
#endif

Figure 9: SDRAM Initialization

6. Click tab "Clocks" and set "Clocks" for the LCDCLK

Clocks Configurat	tion					Generate Project Conten
						Restore Defau
XTAL 20MHz	h			Clock Src: PLL1P	✓ → CPUCLK Div /1	✓ → CPUCLK 480MHz
	-	_				
	> PLL Src: XTAL	~			S ICLK Div /2	✓ → ICLK 240MHz
HOCO 48MHz ~	PLL Div /2	✓ → PLL1P Div /2	✓ → PLL1P 480MHz		> PCLKA Div /4	✓ → PCLKA 120MHz
	4					
LOCO 32768Hz	PLL Mul x96.00	> > PLL1Q Div /2	✓ → PLL1Q 480MHz		> PCLKB Div /8	✓ → PCLKB 60MHz
MOCO 8MHz	PLL 960MHz	PLL1R Div /2	✓ → PLL1R 480MHz		> PCLKC Div /8	→ PCLKC 60MHz
moco dinne						
SUBCLK 32768Hz	PLL2 Disabled	~			PCLKD Div /4	✓ → PCLKD 120MHz
	PLL2 Div /2	✓ → PLL2P Div /2	✓ → PLL2P 0Hz		> PCLKE Div /2	✓ → PCLKE 240MHz
	FLLE DIV/2	STLLET DIV/2	PLLZP UNZ		PEEKE DIV/2	PCLKE 240WIN2
	PLL2 Mul x96.00	> > PLL2Q Div /2	✓ → PLL2Q 0Hz		SDCLK Enabled	$\sim \longrightarrow$ SDCLK 120MHz
		DU 20 Div (2	01120 011			0014 (2014)
	PLL2 0Hz	\rightarrow PLL2R Div /2	✓ → PLL2R 0Hz		→ BCLK Div /4	→ BCLK 120MHz
					EBCLK Div /2	✓ → EBCLK 60MHz
					FCLK Div /8	 ✓ → FCLK 60MHz
	<u> </u>			CLKOUT Disabled		✓ → CLKOUT 0Hz
				SCICLK Disabled	✓ → SCICLK Div /4	✓ → SCICLK 0Hz
				SPICLK Disabled	✓ → SPICLK Div /4	✓ → SPICLK 0Hz
	-			CANFDCLK Disabled	✓ → CANFDCLK Div /8	✓ → CANFDCLK 0Hz
				LCDCLK Src: PLL1Q	✓ → LCDCLK Div /5	 → LCDCLK 96MHz
				I3CCLK Disabled	✓ → I3CCLK Div /3	 → I3CCLK 0Hz
	<u> </u>			UCK Disabled	✓ → UCK Div /5	→ UCK 0Hz
				U60CK Disabled	✓ → U60CK Div /5	 → U60CK 0Hz
					~ -> OCTASPICLK Div /4	

Figure 10. Clocks Setting For LCDCLK

7. Add "New Thread" and name System Thread and Setting Properties.



Renesas RA Family

"GUIX Thermostat" for **EK-RA8D1** Parallel GLCD Display

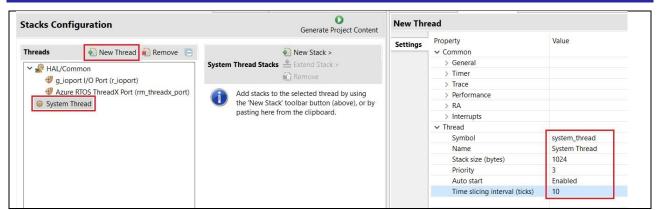


Figure 11. Add New System Thread and Setting Properties Add "New Stack" Azure RTOS GUIX to System Thread.

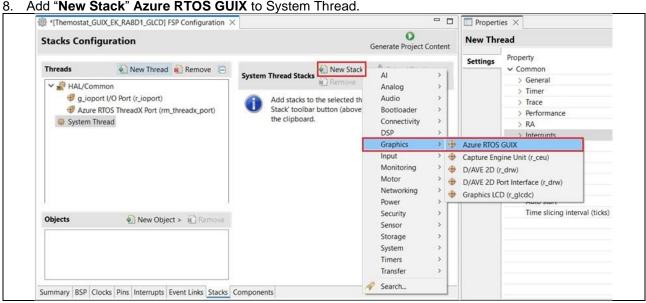


Figure 12. Add Azure RTOS GUIX

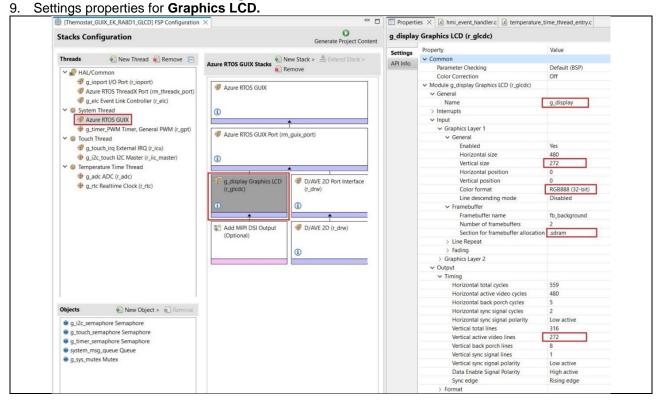




Figure 13. Setting Properties for GLCDC DISPLAY

10. Pin Configuration, change P404's mode to Output mode (Initial high) to enable LCD panel backlight.

Pin Configuration					Generate Projec	t Conten
Select Pin Configuration		Export to CSV	file 🚺 Config	ure Pin Driver Warning	gs	
RA8D1 EK	✓ Manage configurations	Generate	data: g_bsp_p	oin_cfg]	
Pin Selection	Pin Configuration				Cycle Pi	n Group
Type filter text	Name Symbolic Name Comment	Value DISP_BLEN	Link			
 P400 P401 P402 P403 P404 P405 P406 P407 P408 P409 P410 P411 P412 	Mode Pull up/down IRQ Output Type Drive Capacity V Input/Output P404	Output mode (Initial High None CMOS L & GPIO				
P412 P413 P414 P415)3 : ETO_EXOUT II: FTO_FXOLIT				^ ~

Figure 14. Settings PWM Timer properties

- 11. 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.
- 12. Copy Azure RTOS GUIX Studio project to e² studio project (Themostat_GUIX_EK_RA8D1_GLCD) by copying "**guix_studio**" folder in the application note (**AN**) folder (FSP_GUIX_Thermostat) and pasting it in the Themostat_GUIX_EK_RA8D1_GLCD project.



Thermostat_GUIX_EK_RA8D1_GLCD_Fi > Source	✓ ひ Search Source	Q,
<u>^</u>		
2.2.30	8/16/2024 2:48 PM	File folder
3.2.1	8/16/2024 2:48 PM	File folder
4.2.12	8/16/2024 2:48 PM	File folder
5.2.6	9/10/2024 1:00 PM	File folder
6.2.8	8/21/2024 12:31 PM	File folder
completed_project	9/5/2024 4:21 PM	File folder
guix_studio	9/5/2024 4:22 PM	File folder
touch_ft5x06	8/21/2024 12:30 PM	File folder
B EK-RA8D1 - Themostat_GUIX_EK_RA8D1_GLCD/o File Edit Source Refactor Navigate Search B No ← ≪ ← 🖉 🏷 🐼 11 🔍 🎋 ← 9	Project Renesas Views Run Re	
File Edit Source Refactor Navigate Search	Project Renesas Views Run Re	nesas Al Window Help
File Edit Source Refactor Navigate Search	Project Renesas Views Run Re	
File Edit Source Refactor Navigate Search Image: Search Image: Search Iman	Project Renesas Views Run Re	🗖 🗖 🕼 [Themostat
File Edit Source Refactor Navigate Search	Project Renesas Views Run Re	🗖 🗖 🕼 [Themostat
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Project Explorer × File Project Explorer × File Edit Source Refactor Navigate Search File Edit Source Refact	Project Renesas Views Run Re	🗖 🗖 🕼 [Themostat
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Project Explorer × File Edit Source Refactor Navigate Search Project Explorer × File Edit Source Refactor Navigate Search File Edit Source Refactor Na	Project Renesas Views Run Re View Go Into Open in New Window Show In	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Project Explorer × File Project Explorer × File Edit Source Refactor Navigate Search File Edit Source Refact	Project Renesas Views Run Re New Go Into Open in New Window Show In Copy	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Froject Explorer × File Project Explorer × File Constat_GUIX_EK_RA8D1_GLCD [Debug Source Refactor Navigate Search File Constat_GUIX_EK_RA8D1_GLCD [Debug Source Refactor Navigate Search File Constat_GUIX_EK_RA8D1_GLCD [Debug Source Refactor Navigate Search File Constat_GUIX_EK_RA8D1_GLCD [Debug File Constat_GUIX_EK_RA8D1_GUIX_EK_RA8D1_GLCD [Debug Fi	Project Renesas Views Run Re View Go Into Open in New Window Show In	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Froject Explorer × File Edit Source Refactor Navigate Search File	Project Renesas Views Run Re View Go Into Open in New Window Show In Copy Paste	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Froject Explorer × Froject Explorer × File Edit Source Refactor Navigate Search File Edit Source Refactor Na	Project Renesas Views Run Re View Go Into Open in New Window Show In Copy Paste View Delete	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Froject Explorer × Froject Explorer × File Themostat_GUIX_EK_RA8D1_GLCD [Debug Source Ragen Source Ragen So	Project Renesas Views Run Re View Go Into Open in New Window Show In Copy Paste View Source	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Project Explorer × File Edit Source Refactor Navigate Search Project Explorer × File Edit Source Refactor Navigate Search File Edit Source Refactor Na	Project Renesas Views Run Re New Go Into Open in New Window Show In Copy Paste Delete Source Move Rename	
File Edit Source Refactor Navigate Search File Edit Source Refactor Navigate Search Froject Explorer × File Edit Source Refactor Navigate Search File	Project Renesas Views Run Re View Go Into Open in New Window Show In Copy Paste Source Move Rename Import	

Figure 15. Copying the Azure RTOS GUIX Studio Project to e² studio

13. GUIX Studio project is now in Themostat_GUIX_EK_RA8D1_GLCD 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.

The guix_studio folder holds the GUIX thermostat project, the source of the graphics, and the fonts. The graphics and the fonts will be used by the GUIX thermostat project when it is compiled by the GUIX Studio application. The content in this folder will be used in a later step to generate the GUIX .c and .h source files using the GUIX Studio Application. This folder will not be compiled by e² studio IDE.



Project Explorer 🗙		6 \$ 7	000	🗆 🎡 [Themostat_	GUIX_EK_RA8D1_	GLCD] FSP Co	nfiguration
	X_EK_RA8D1_GLCD [Debug]			Exclude from build	1	- 0	×
> 🐇 Binaries				Exclude nom band			~
> D Includes				Exclude object(s) from	n build in the follo	wing configu	rations
> 🐸 ra				Debug			
> 😕 ra_gen > 😕 src				Release			
> 🔁 Debug							
> > build							
> 🕞 guix_studio							
> 🧁 ra_cfg	New	>					
> 🗁 script	Go Into						
💮 configurat	Show In	Alt+Shift+W>					
ra_cfg.txt	Сору	Ctrl+C					
Themostal	Paste	Ctrl+V					
> 🕜 Developer 🎽	Delete	Delete					
	Move						
	Rename	F2					
<u>a</u>	Import				Select All	Desele	ect All
R.S.	Export						
	Build Project	Ctrl+B					
E	Refresh	F5		?	OK	Can	cel
	Source	>					
	Build Targets	>					
	Resource Configurations	>	Exclud	de from Build			
0	Run As	>	Reset	to Default	New New	Object > 👔	Remove

Figure 16. guix_studio folder containing

14. Get to Themostat_GUIX_EK_RA8D1_GLCD project folder by right clicking the e² studio project and select "**System Explorer**" as shown below.

谷	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	
	Validate	
	Configure	>
	Source	>
	Properties	Alt+Enter

Figure 17. Selecting System Explorer

15. Open thermostat.gpx project file in "guix_studio > GNU" sub-folder in your Themostat_GUIX_EK_RA8D1_GLCD 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.



Themostat_GUIX_EK_RA8D	1_GLCD > guix_studio 3	GNU
Name	Date modified	Туре
thermostat.gxp	6/27/2024 3:50 PM	GUIX Studio
Open ☑ Edit with Notepad ☑ CrowdStrike Falcor ☑ Classify and protect ☑ Move to OneDrive ७ Scan with FortiSand ☑ Scan with FortiSand ☑ Share Open with Keep using this approximation	n malware scan (in progre ct dbox	ess)
GUIX Stu	dio	
Other options More apps ↓ Always use to	his app to open .gxp OK	files

Figure 18. Opening the Project File

16. 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.

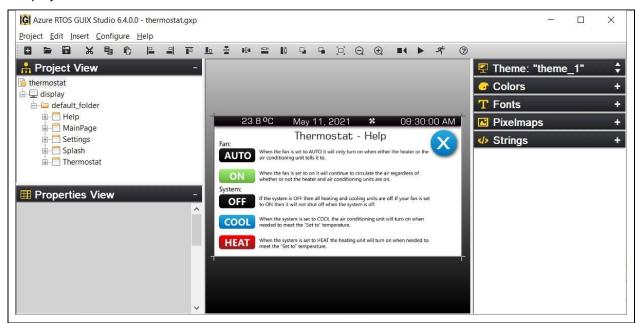


Figure 19. GUIX Studio Thermostat Application View

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



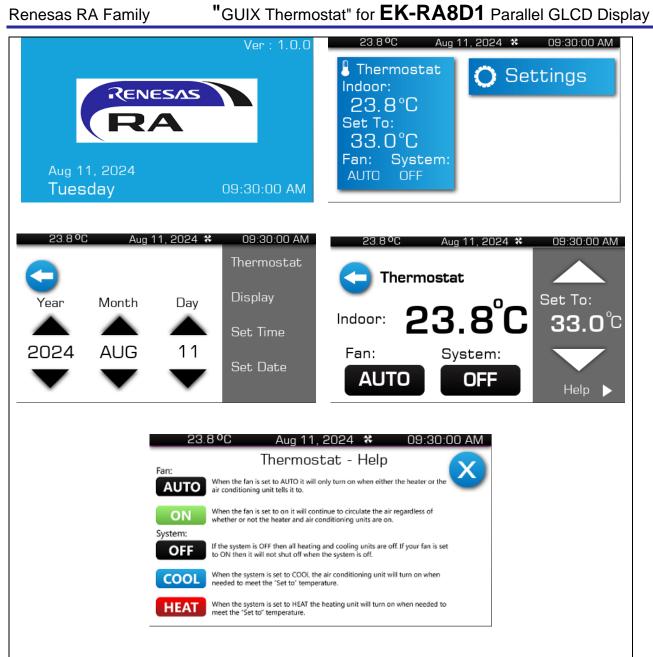


Figure 20. Azure RTOS GUIX Studio Project Screens

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



Directories
Source Files//src/guix_gen browse
Header Files//src/guix_gen browse
Resource Files browse
Target CPU Renesas RA V Advanced Settings
Toolchain GNU \checkmark Dig endian
Additional Headers
Number of Displays 1 GUIX Library Version 6 . 4 . 0 .
Display Configuration
Display Number 1 🜩 Name display
x resolution 480 pixels y resolution 272 pixels
1 topp grayscale 1:5:5:5 forma
2 bpp invert polarity 4:4:4:4 format
0 16 bpp packed format
ⓐ 24 bpp ○ 32 bpp Rotation: None ✓
Number of Palette Mode Anti-aliased Text 8

Figure 21. Configure Project Settings

19. Go back e² studio project (Themostat_GUIX_EK_RA8D1_GLCD), right click "**src**", then select "**New->Folder**" and create a folder named "**guix_gen**".

Project Explo	orer ×		E	🕏 🍸 🕴 🗖 🗖	🔅 [Themost	t 📴 New Folder 🛛 🗆 🗙 🗌	
> 🔊 Includ > 😕 ra					2 6 23 24	Folder Create a new folder resource.	
 > Construction <	New Go Into Open in New Window Show In Copy Paste Delete Source Move Rename	Alt+Shift+W > Ctrl+C Ctrl+V Delete > F2		Project C/C++ Project File File from Template Folder Class Header File Source File Source Folder	25	Inter or select the parent folder: Themostat_GUIX_EK_RA8D1_GLCD/src Image: Settings Image: Setin	
> ⑦ De 📷	Import Export Build Project Refresh Index Build Targets Resource Configurations Source Team Compare With	Ctrl+B F5 > > > > > > > > > > > > > > > > > >		Example	Ctrl+N 43 44 45 46 47 48 49 50 51 52 53	>> ap Folder name: guix.gen Advanced >> p_ 11 ch ? Einish Cancel N)	

Figure 22. Creating a "guix_gen" in e² studio Project

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



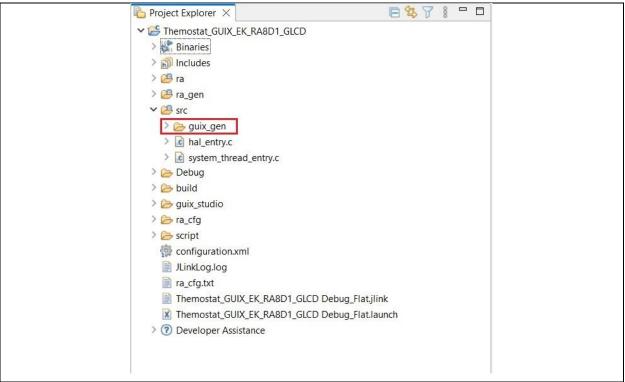


Figure 23. Confirming Creation of "guix_gen"

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

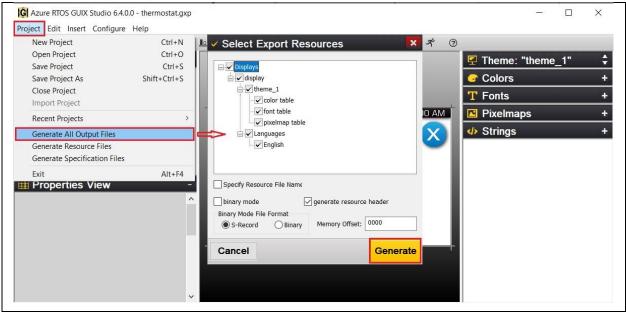


Figure 24. Clicking Generate All Output Files

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



A Notification	×
All Output files have been updated.	
	ок

Figure 25. All Output Files Updated Notification

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

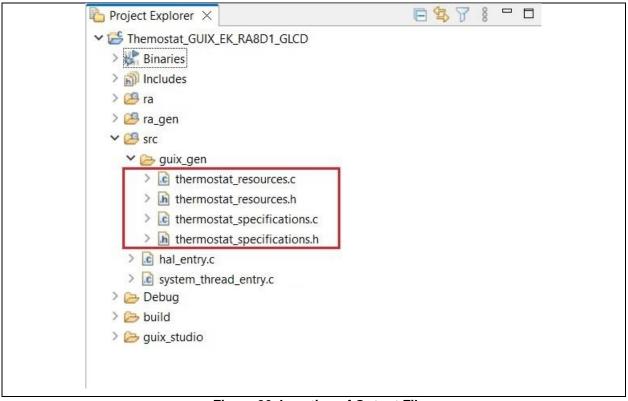


Figure 26. Location of Output Files

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.



Properties View Wdget Type window Wdget Name Spissh Vdget Name Spissh Jser Data 0 Jser Data 0 Jser Data 0 Vote 0 Spected 0 Inaibid Accepts Focus Vintum Allocate UmOOW_J Spected fil WINDOW_J Spasbed fil WINDOW_J			Ver : 1.0.0
Vidget B 0 Jser Data 0 Jser Data 0 Vopp 0	low A	R	
Vidget B 0 Jser Data 0 Jser Data 0 Vopp 0			
eff 0 top 0 Vidth 480 teight 272 Border No Border transparent Jorder broketedd Imabed tacepts Focus Juntime Allocate tormal fill WINDOW_J bisected fill WINDOW_J			
top 0 Vikith 480 teight 272 sorder No Border transparent		May 11, 2021	
Vidth 480 reight 272 Sorder No Border ransparent Traw Selected Inabled Accepts Focus Runtime Allocate formal fill WINDOW_1 Disabled fill WINDOW_2		Tuesday	09:30:00 AM
teight 272 broder No Border ransparent Vraw Selected nabled Accepts Focus funtime Alcoate Unitime Alcoate Unitime Alcoate Unitime Alcoate UNINDOW_1 Disabled fill WINDOW_2			
No Border No Border Transparent Transparent Transpled Accepts Focus Tuntime Alccate Lormal fill WINDOW_1 Disabled fill WINDOW_2			
Transparent Trans Seetced Inabled Accepts Focus Nuntime Allocate Formal fill WINDOW_1 Disabled fill WINDOW_1			
Straw Selected accepts Focus Accepts Focus Autime Allocate formal fill WINDOW_1 Selected fill WINDOW_1 Stabled fill WINDOW_1			
Enabled Accepts Focus Untime Allocate Jormal fill WINDOW_F Becketed fill WINDOW_F Disabled fill WINDOW_F			
Accepts Focus Runtime Allocate fill VINDOW_f Selected fill VINDOW_f Visabled fill VINDOW_f			
Runtime Allocate Iormal fil WINDOW_5 Selected fil WINDOW_6 Disabled fil WINDOW_6			
Normal fill WINDOW_F Selected fill WINDOW_F Disabled fill WINDOW_F			
Selected fill WINDOW_F			
Disabled fill WINDOW_F			
Template			
emplate			
/isible At Startup			
Draw Function			
vent Function splashscre	shscreen_event		

Figure 27. 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 **"2.2.30"** folder in the AN folder:
 - hmi_event_handler.c
 - system_thread_entry.c

Build Thermostat_GUIX_EK_RA8D1_GLCD 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);
}
```

Figure 28: Splash screen

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.



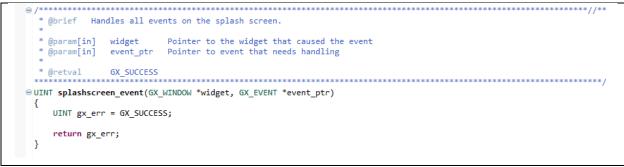


Figure 29: Splashscreen event

28. Using Renesas-app-lcd-conv_v1_b_mfg board to connect 1 side to RA6M3G's LCD and the other side connect to J57. (40 pins connector for LCD) follow description on convert board and the image from Figure 32.

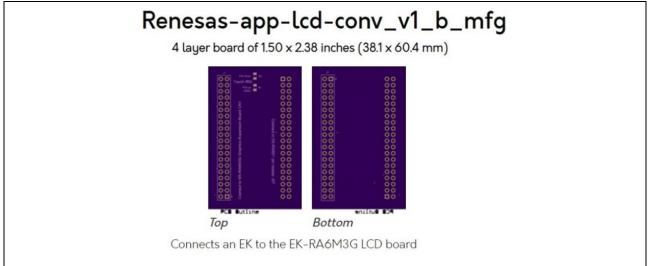
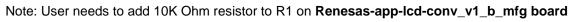


Figure 30: 4 layer mfg board



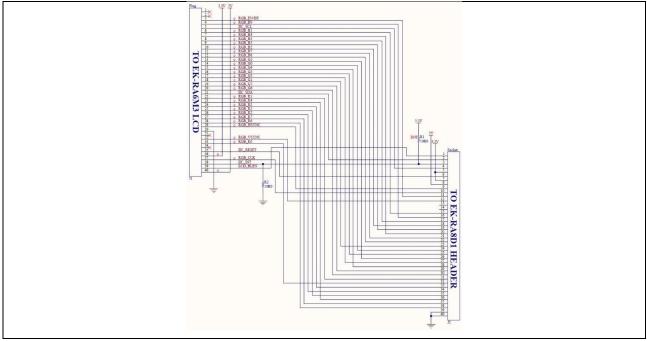


Figure 31: 40 pins connection from KA-RA6M3G LCD connector to J57 of EK-RA8D1 on board



29. Get your EK-RA8D1 ready to run the project. Connect LCD board to **Graphics Expansion** connector on EK-RA6M3 as shown below.

- Switch SW1-6 for GLCDC set "ON" and switch SW1-7 for SDRAM set "ON". (https://www.renesas.com/us/en/products/microcontrollers-microprocessors/ra-cortex-m-mcus/ekra8d1-evaluation-kit-ra8d1-mcu-group)

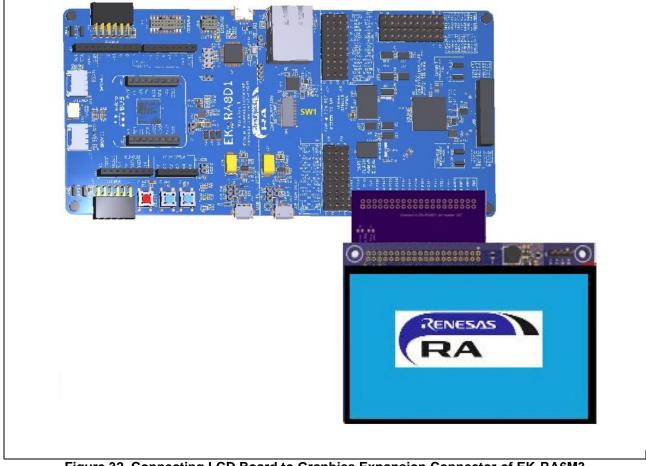


Figure 32. Connecting LCD Board to Graphics Expansion Connector of EK-RA6M3

30. Connect USB Debug using J10 on EK-RA8D1 board to your PC. Build Project and start Debug

Thermostat_GUIX_EK_RA8D1_GLCD project, you will see a black screen.

31. Add the following code to **splashscreen_event** function in **hmi_event_handler.c** to show The 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;
        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 "2.2.30" folder in the AN folder.



32. **Download and Run** the project, you will see the Splash screen on LCD panel.

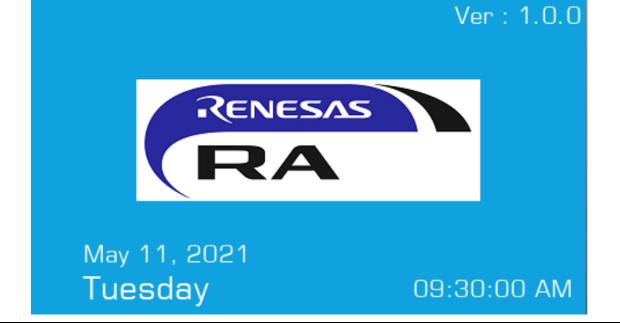


Figure 33. Splash Screen View on LCD



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

- 1. Copy and replace these files in "**src**" folder in e² studio project with the files in "**3.2.1**" 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:
            qx err = qx window event process (widget, event ptr);
            if (GX SUCCESS != gx err) {
                while(1);
            }
            break;
    }
```

Figure 34: Splashscreen event function code

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



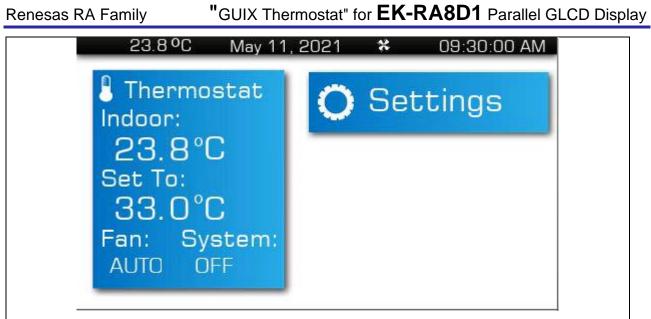


Figure 35. Main Page Screen



4. Add Touch Driver to Thermostat_GUIX_EK_RA8D1_GLCD 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_RA8D1_GLCD project, create a folder by right-clicking "src", then select "New->Folder".

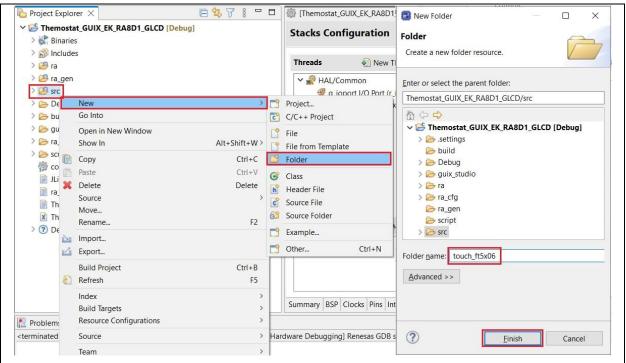


Figure 36. Creating New Folder in Thermostat_GUIX_EK_RA8D1_GLCD Project

2. Copy touch_ft5x06.c and touch_ft5x06.h from "touch_ft5x06" folder in the Source file to the one in e² studio project.



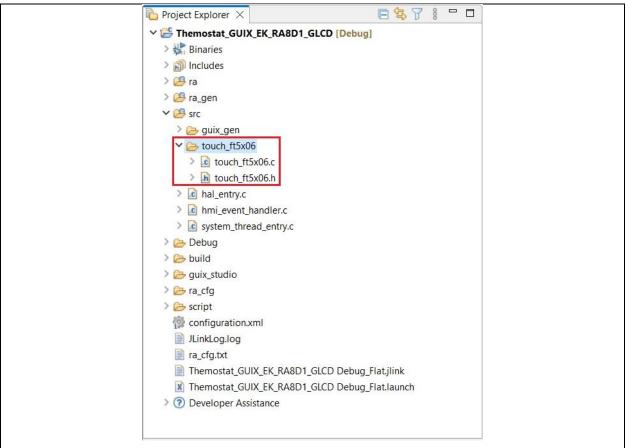


Figure 37. Copying files to the e² studio Project

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

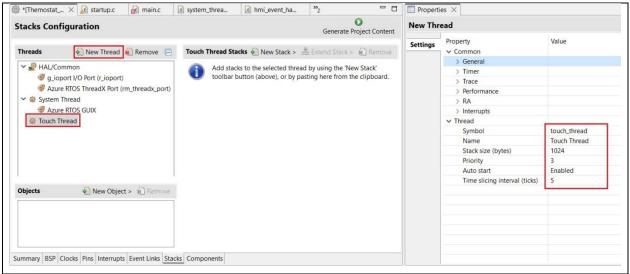


Figure 38. Creating Touch Thread

- 4. The pins marked in rectangles below are used for touch panel controller on the LCD board:
 - IRQ3 interrupt (P510) is used to trigger touch events.
 - I2C channel 1 (P512, P511) is used to read and write data to the touch controller. PA01 is used to reset the touch controller.



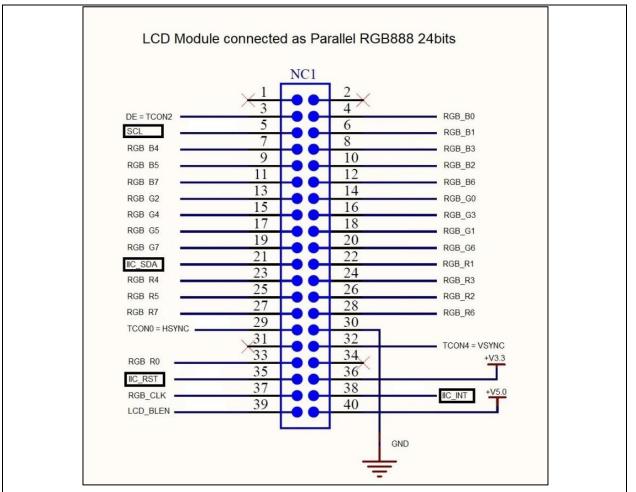


Figure 39. RA6M3G LCD Pin address

5. Setting DISP_RST IRQ (PA01) Pin Configuration



				Generate Project Conter
elect Pin Configuration		🔛 Ex	port to CSV file	Configure Pin Driver Warnings
RA8D1 EK	✓ Manage configuration	<u>IS</u> E	☑ Generate data:	g_bsp_pin_cfg
Pin Selection $\models \oplus \models \downarrow_2^a$	Pin Configuration			😲 Cycle Pin Group
Type filter text	Name	Value	Link	
✓ ✓ PA ^	Symbolic Name	DISP_RST		
PA00	Comment	-		
✓ PA01	Mode	Input mode		
✓ PA02	Pull up/down	None		
PA03	Output Type	CMOS		
✓ PA04	Drive Capacity	L		
✓ PA05	✓ Input/Output PA01	V GPIO	4	
✓ PA06	PAUL	V GPIO		
✓ PA07				
PA08				
✓ PA09				
✓ PA10				
🛩 PA11				
PA12	<			>
✓ PA13	Module name: PA01			
✓ PA14	Port Capabilities: BUS: RD			
ΦΔ15 < >	GLCD: LC	D_TCON0		

Figure 40. Setting DISP_RST IRQ Pin Configuration

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

Settings Property Settings Property Common Settings Property Image: Settings Extend Stacks Extend Stacks Analog Analog Analog Analog Analog Analog Analog Image: Settings Extend Stacks Remove Analog Analog <th< th=""><th>*[Themostat_GUIX_EK_RA8D1_GLCD] FSP Configuration Stacks Configuration</th><th></th><th>Generate Project Content</th><th>Touch Th</th><th></th></th<>	*[Themostat_GUIX_EK_RA8D1_GLCD] FSP Configuration Stacks Configuration		Generate Project Content	Touch Th	
Input > ◆ External IRQ (r_icu) Monitoring Name Motor > Motor > Priority Priority Power > Auto start Security > Storage > System > Timers >	✓	Touch Thread Stacks New Stack >	Fxtend Stack > Al Analog Audio Bootloader Connectivity	Settings	Common General Timer Trace Performance RA
Objects Security Security Sensor > Sorrage > System > Timers >	🚳 Touch Thread		Input Monitoring Motor Networking Power	+ External	IRQ (r_icu) Name Stack size (bytes) Priority Auto start
	Objects New Object > 🐑 Remove		Sensor > Storage > System > Timers >		

Figure 41. Adding External IRQ Driver on r_icu to Touch Thread

7. Adding and setting the External IRQ Properties.

Note: Click to " <unavailable>

" from the Pins row to set **P510** for **IRQ3**.



Renesas RA Family

cks Configuration	0	g_extern	al_irq0 External IRQ (r_ic	u)	
	Generate Project Content		Property		Value
eads 🕢 New Thread 🔬 Remove 📃 g	_external_irq0 External 🕢 New Stack >	Settings	✓ Common		value
	RQ (r_icu) Stacks	API Info	Parameter Checking		Default (BSP)
MAL/Common			✓ Module g_touch_irq Extern	ial IRQ (r_icu)	
g_ioport I/O Port (r_ioport) Azure RTOS ThreadX Port (rm_threadx_port)	g_touch_irq External IRQ		Name		g_touch_irq
g_elc Event Link Controller (r_elc)	(r_icu)		Channel		3
System Thread	0		Trigger Digital Filtering		Falling Enabled
Azure RTOS GUIX			Digital Filtering Sample	Clock (Only valid when	PCLK / 64
g_timer_PWM Timer, General PWM (r_gpt)			Callback		touch_irq_cb
Touch Thread			Pin Interrupt Priority		Priority 12
<pre> g_touch_irq External IRQ (r_icu) </pre>			✓ Pins IRO3		<unavailable></unavailable>
ects 🔊 New Object > 🖗 Remove					
hary BSP Clocks Pins Interrupts EventLinks Stacks C *[Themostat_GUIX_EK_RA8D1_C					Diect Conten
RA8D1 EK	Manage configura		o CSV file 🚺 Con	ngure Pin Driv	ver warnings
Select Pin Configuration RA8D1 EK Generate data: g_bsp_pin_	Manage configura		o Csv file 💽 Con	ngure Pin Driv	ver warnings
RA8D1 EK	Manage configura		o Csv file ⊪ _ Con		le Pin Group
RA8D1 EK	✓ Manage configura				
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection : ① ① □ ↓ ^a z Type filter text	Manage configura cfg Pin Configuration Name	itions	Je	Cyc	le Pin Group
RA8D1 EK Generate data: g_bsp_pin_ Pin Selection : The sel	Manage configura cfg Pin Configuration Name Pin Group Selection	valu Valu	Je ed	Cyc	le Pin Group
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode	itions Valu	Je ed	Cyc	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection	valu Valu	Je ed	Cyc	le Pin Group
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode Input/Output	tions Valu Mixu Cust	ie ed tom	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode	tions Valu Mixe Cust	ue ed tom v510	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode Input/Output	Valu Mixe Cust	ue ed tom 2510 None	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode Input/Output	Valu Mixe Cust	ue ed tom v510	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode Input/Output	Valu Mixe Cust	ue ed tom 2510 None	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode Input/Output	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configuration Pin Configuration Name Pin Group Selection Operation Mode VInput/Output IRQ3	Valu Mixe Cust	ue ed tom 2510 None P208	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode Input/Output	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Constraint of the selection Type filter text > ✓ HMI:GLCD > ✓ Interrupt:IRQ IRQ IRQ IRQ1 IRQ2 ✓ IRQ3 IRQ4	Manage configuration Pin Configuration Name Pin Group Selection Operation Mode VInput/Output IRQ3	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Constraint of the selection Type filter text ✓ HMI:GLCD ✓ HMI:GLCD ✓ HMI:GLCD Interrupt:IRQ IRQ0 IRQ1 IRQ2 ✓ IRQ3 IRQ4 IRQ5	Manage configuration Pin Configuration Name Pin Group Selection Operation Mode VInput/Output IRQ3	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode V Input/Output IRQ3	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Constraint of the selection ✓ Interrupt: Image: Constraint of the selection ✓ HMI: GLD ✓ HMI: Image: Constraint of the selection ✓ Interrupt: Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Constraint of the selection Image: Conset of the se	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode V Input/Output IRQ3	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link
RA8D1 EK ✓ Generate data: g_bsp_pin_ Pin Selection Image: Ima	Manage configura cfg Pin Configuration Name Pin Group Selection Operation Mode V Input/Output IRQ3	Valu Mixe Cust	ue ed tom 2510 None P208 2212	Cyc Lock	le Pin Group Link

Figure 42. Settings External IRQ Properties

8. Adding and setting Properties I2C Master Driver on r_iic_master to Touch Thread



Renesas RA Family

"GUIX Thermostat" for **EK-RA8D1** Parallel GLCD Display

		Generate F	Project Cont	tent Property
hreads New Thread Rem * # HAL/Common # g_ioport I/O Port (r_ioport) # Azure RTOS ThreadX Port (rm_thread # g_elc Event Link Controller (r_elc) # Azure RTOS GUIX # Azure RTOS GUIX g_timer_PWM Timer, General PWM I # G_touch_Thread # g_touch_irq External IRQ (r_icu)	ix_port)	tacks Remove Al Analo Audic Booti Conn DSP Grapi Input Moni Moto	o loader ectivity hics toring r orking r r orking r ity or ge m rs fer	 Common General Timer Trace Performance Azure RTOS USBX DFU Azure RTOS USBX HLDC Azure RTOS USBX HHID Azure RTOS USBX HHID Azure RTOS USBX HUVC Azure RTOS USBX NUVC Azure RTOS USBX OTG CDC Azure RTOS USBX PAUD Azure RTOS USBX PAUD Azure RTOS USBX PAUD Azure RTOS USBX PHID Azure RTOS USBX PHID Azure RTOS USBX PHID Azure RTOS USBX PMN CAN FD Lite (r_canfdlite) I2C Communication Device (rm_comms_i2c)
mmary BSP Clocks Pins Interrupts Event L *(Themostat_GUIX_EK_RA8D1_GLCD) FSP Configuratic tacks Configuration	1	Generate Project Content	Propertie g_i2c_mas	
		Generate Project Content	Settings	Property Value
hreads 🕢 New Thread 🔬 Remove 📄	g_i2c_master0 I2C Master	New Stack >	API Info	Common Module g_i2c_touch I2C Master (r_iic_master)

Figure 43. Adding I2C Master Driver on r_iic_master to Touch Thread

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.



tacks Configuration		Generate Project Content		g_new_semaphore0 Semaphore		
hreads 🔄 New Thread 🔬 Remove 📄	g_rtc0 Realtime Clock (r rtc) Stacks	New Stack >	Settings	Property Name Symbol	Value I2C Semaphore g_i2c_semaphore	
 ^A HAL/Common ^A g_ioport I/O Port (r_ioport) ^A Azure RTOS ThreadX Port (rm_threadx_port) ^A System Thread ^A Azure RTOS GUIX ^A G_ucuch Thread ^A g_touch_irq External IRQ (r_icu) ^A g_i2c_touch I2C Master (r_iic_master) ^A g_i2c_semaphore Semaphore ^A Mutex ^A Queue ^A Semaphore Semaphore ^A Semaphore ^A Semaphore ^A System Thread ^A G_ueue ^A ^A		Extend Stack > Remove		Initial count		

Figure 44. Adding Touch Semaphore

10. 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.

n X		- 0	Properti	es ×	
	O Generate Proje	ect Content	g_touch_	semaphore Sema	phore
g_rtc0 Realtime Clock			Settings	Property Name	Value Touch Semaphore
(r_rtc) Stacks		Remove		Symbol Initial count	g_touch_semaphore 0
	g_rtc0 Realtime Clock (r_rtc) Stacks g_touch_irq Exte (r_icu)	Generate Proje g_rtc0 Realtime Clock (r_rtc) Stacks Extend Stack > g_touch_irq External IRQ (r_icu)	Generate Project Content	Generate Project Content g_touch_ g_touch_	g_touch_semaphore Seman g_rtc0 Realtime Clock Image: New Stack > image: Settings (r_rtc) Stacks Image: Extend Stack > image: Remove g_touch_ing External IRQ (r_icu) Image: Image: Settings

Figure 45. Adding I2C Semaphore

- 11. In RA Configurator, click ^{Generate Project Content} to generate project content.
- 12. Copy and replace these files in "**src**" folder in the e² studio project with the files in "**4.2.12**" folder in the AN folder:
 - hmi_event_handler.c
 - system_api.h
 - system_thread_entry.c



touch_thread_entry.c

13. 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 = touch_data.point[0].y;
    gxe.gx_event_type = GX_EVENT_PEN_DOWN;
    gx_system_event_send (&gxe);
}
else if (GX_EVENT_PEN_DOWN == gxe.gx_event_type) // @suppress("10.2b If statement")
{
    gxe.gx_event_type = GX_EVENT_PEN_UP;
    gx_system_event_send (&gxe);
}
```

Figure 46: Sending touch event to GUIX

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



Figure 47: Screens created in systems thread entry

15. 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.



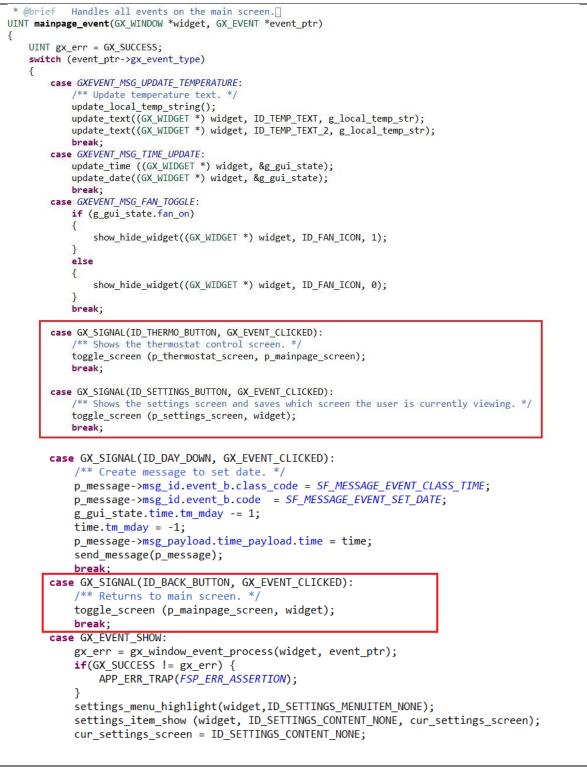


Figure 48: Thermostat button and Settings button clicked

16. 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.



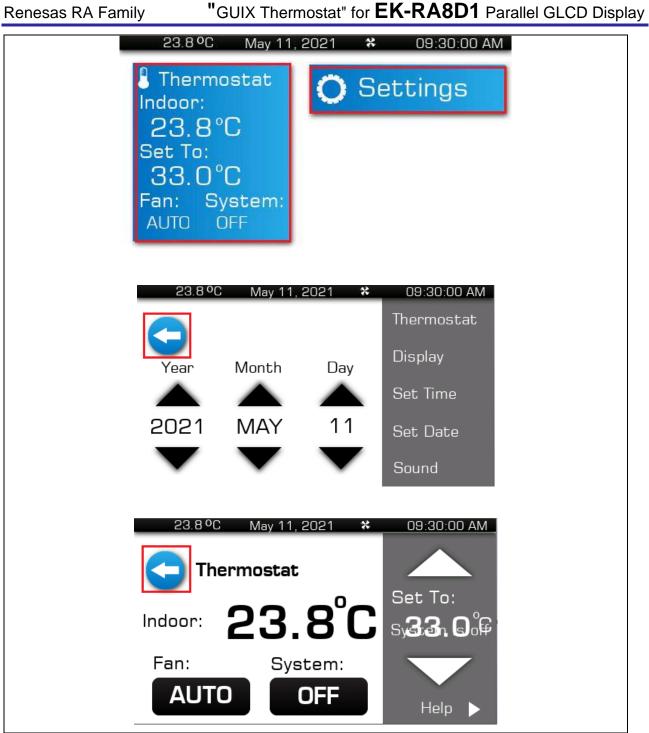


Figure 49. Navigating between Main Page Screen and Thermostat Screen



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 P404 on the RA8D1 MCU, is configured in PWM mode to control the intensity of LCD backlight.

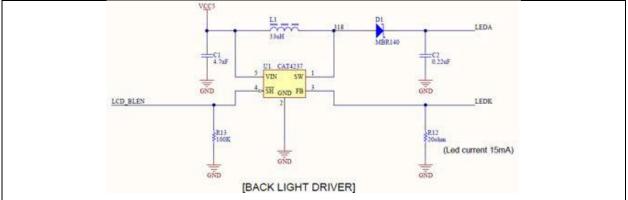


Figure 50. LCD Board Schematic

2. To configure P404 in PWM output mode, we disable it in Pin Configuration at first. **Save this change before moving to the next step.**

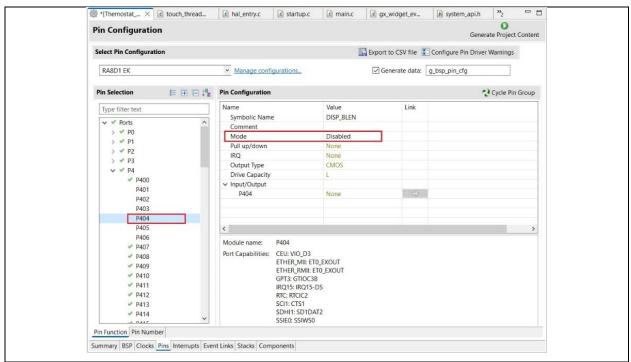


Figure 51. Disabling P404 in Pin Configuration

3. In Pin Configuration, set P404 as GPT3 - GTIOCB output.



Select Pin Configuration		III Consta	- COVEL-	Configure	e Pin Driver War	
Select Pin Configuration		Export t	o CSV file	- Configure	e Pin Driver war	nings
RA8D1 EK	✓ Manage configurations	Ger Ger	nerate data:	g_bsp_pin	_cfg	
Pin Selection $\blacksquare \boxdot \Box \downarrow_{\mathbb{Z}}^{a}$	Pin Configuration				Cy	cle Pin Group
Type filter text	Name	Value	Lock	Link		
	Pin Group Selection	Mixed				
> TRG:CAC ^ > Timers:AGT	Operation Mode	GTIOCA or GTIOCB				
✓ ✓ Timers:GPT	✓ Input/Output					
GPT0	GTIOC3A	None	D			
GPT1	GTIOC3B	✓ P404	ď			
GPT2						
GPT3						
GPT4						
GPT5						
GPT6						
GPT7						
GPT8	<					>
GPT9	Module name: GPT3					
GPT10	Module Hume. Grip					
GPT11						
GPT12						
GPT13						
> Timers:GPT_OPS						
> Timers:GPT_POEG						
> Timers:RTC						
> Timers:ULPT						

Figure 52. Setting P404 as GPT3 - GTIOCB Output in Pin Configuration

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

Stacks Configuration		Generate Project Content		PWM Timer, General PWM (r	🛷 🖩 🖓 🖳 🛷 着 🏹 🗔 🛷 gpt)
Threads New Thread Remove E Image: Second Seco	g_timer_PWM Timer, General PWM (r_gpt) Stacks general PWM (r_gpt) (1)	New Stack >	Settings API Info	Property Property Common Parameter Checking Pin Output Support Write Protect Enable	g_timer_PWM 3 Saw-wave PWM SOOOO Nanoseconds

Figure 53. 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.

O

- 5. In RA Configurator, click Generate Project Content to generate project content.
- 6. Copy and replace these files in "**src**" folder in e² studio project with the files in "**5.2.6**" folder in the AN folder:
 - hmi_event_handler.c
 - system_thread_entry.c



- brightness.c
- brightness.h
- system_api.h
- system_cfg.h
- system_timer.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. Note that if the period could be larger than
 * UINT32_MAX / 100, this calculation could overflow. */
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_GTIOCB);
if (FSP_SUCCESS == err)
{
 *p_brightness = (uint8_t)brightness;
}
```

Figure 54: Setting the PWM duty cycle

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.

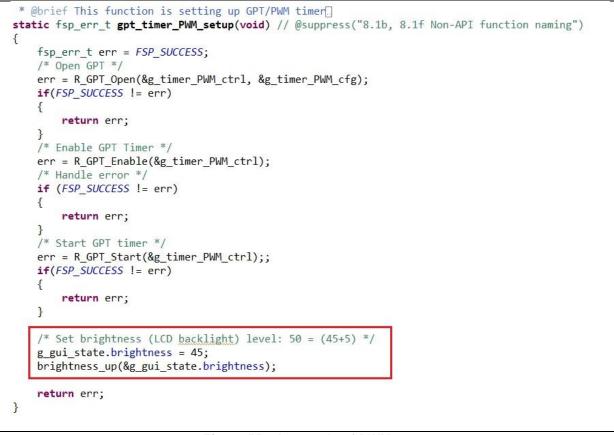


Figure 55. duty cycle of PWM output

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



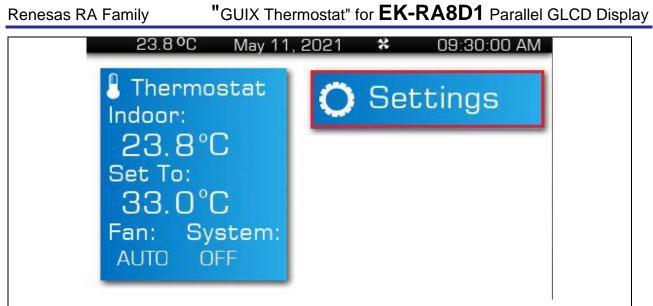


Figure 56. Settings Button on Main Page Screen

10. PWM output measured on pin P404 with brightness is set to 50%.

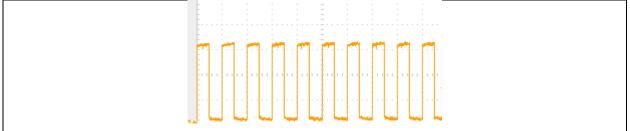


Figure 57. PWM Output on P603 at 50% Brightness

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

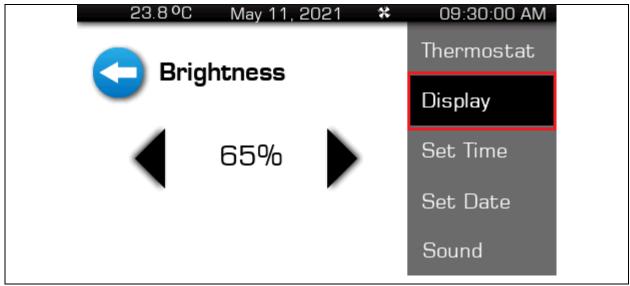


Figure 58. Display on Settings Screen

12. PWM output measured on pin P404 after changing brightness to 65%.



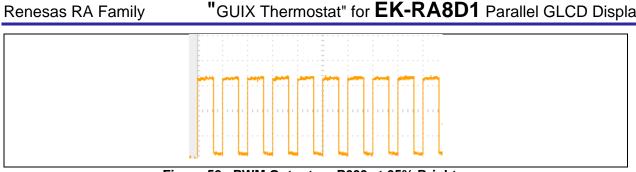


Figure 59. PWM Output on P603 at 65% Brightness



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**.

tacks Configuration	Generate Project Co	ntent	New Thre	ead	
Intreads New Thread Remove Remove	Generate Project Co	ve	New Thro	ead Property Common General Timer Frace Performance RA Performance RA Thread Symbol Stack size (bytes) Priority Auto start Time slicing interval (ticks)	Value temperature_time_thread Temperature Time Thread 1024 5 Enabled 10
S S					

Figure 60. Create Temperature Time Thread

2. Adding **RTC Driver on g_rtc** to **Temperature Time Thread** and **Setting the Properties**.

tacks Configuration	Generat	e Project Content	New Thr	ead	
Image: System Thread Remove Image: HAL/Common Image: Graph of the system of the system of the system thread the system Thread Image: Graph of the system of t	Add stacks to the selected threa New Stack' toolbar button (abo here from the clipboard.	Al Analog Audio Bootloader Connectivity DSP Graphics Input Monitoring Motor Networking Power Security Sensor Storage System	Settings >	Property ✓ Common > General > Timer > Trace > Performance > RA > Interrupts ✓ Thread Symbol Name Stack size (bytes) Priority Auto start Time slicing interval (ticks)	
Objects 🔄 New Object > 👔 Remove		Timers Transfer		Output Enable for GPT (r_poeg)	_
 g_i2c_semaphore Semaphore g_touch_semaphore Semaphore 		Search		Itime Clock (r_rtc) ee-Phase PWM (r_gpt_three_phase) er, General PWM (r_gpt) er, Low-Power (r_agt) er, Ultra-Low-Power (r_ulpt)	



Renesas RA Family

"GUIX Thermostat" for **EK-RA8D1** Parallel GLCD Display

acks Configuration		Generate	Project Content	g_rtc0 Realtime Clock (r_rtc)			
hreads 🔄 New Thread 👔 Remove 📄	g rtc0 Realtime Clock	New Stack >		Settings	Property V Common	Value	
	(r_rtc) Stacks	Extend Stack >	Remove	API Info	Parameter Checking	Default (BSP)	
A with the second secon					Set Source Clock in Open	Disabled	
g_ioport I/O Port (r_ioport)	🕀 g_rtc Realtime Clo	ock I			✓ Module g_rtc0 Realtime Clock (r_rtc)		
Azure RTOS ThreadX Port (rm_threadx_port)	(r_rtc)	JCK I			Name	g_rtc	
System Thread	02,007				Clock Source	LOCO	
Azure RTOS GUIX	(i)				Frequency Comparision Value (LO	255	
g_timer_PWM Timer, General PWM (r_gpt)					Automatic Adjustment Mode	Enabled	
Touch Thread					Automatic Adjustment Period	10 Seconds	
g_touch_irg External IRQ (r_icu)					Adjustment Type (Plus-Minus)	NONE	
g_i2c_touch I2C Master (r_iic_master)					Error Adjustment Value	0	
Temperature Time Thread					Callback	time_update_callback	
					Alarm Interrupt Priority	Disabled	
g_rtc Reartime Clock (i_rtc)					Period Interrupt Priority	Priority 9	
					Carry Interrupt Priority	Priority 12	
					✓ Pins		
>					RTCIC0	<unavailable></unavailable>	
					RTCIC1	<unavailable></unavailable>	
ijects 🔄 New Object > 🎣 Remove					RTCIC2	<unavailable></unavailable>	
q_i2c_semaphore Semaphore					RTCOUT	<unavailable></unavailable>	
g_touch_semaphore Semaphore							
mary BSP Clocks Pins Interrupts Event Links Stac	ks Components						

Figure 61. Adding RTC Driver on g_rtc to Temperature Time Thread and Setting Properties

3. In project configuration, add ADC Driver on r_adc to System Thread and Setting Properties.

Threads	🕢 New Thread 👔 Remove 📄		rature Time	New Stack >	>			etting	✓ Common
 Azure F System Th Azure F g_timer Touch Three g_touch g_i2c_touch g_i2c_semaph 	tt I/O Port (r_joport) RTOS ThreadX Port (rm_threadx_port read RTOS GUIX _PWM Timer, General PWM (r_gpt) ead _jrq External IRQ (r_jcu) puch I2C Master (r_jic_master) re Time Thread ealtime Clock (r_rtc) New Object > Remove	Thread	Add stacks to	Extend Stacl		Al Analog Audio Bootloader Connectivity DSP Graphics Input Monitoring Motor Networking Power Security Security Sensor Storage System Timers Transfer Search		⊕ / ⊕ (Seneral Seneral ADC (r_adc) ADC-DMAC Integration (r_adc) Comparator, High-Speed (r_acmphs) DAC (r_dac) Thread Symbol Name Stack size (bytes) Priority Auto start Time slicing interval (ticks)



hreads 🔹 New Thread 🕷 Remove 📄	Generate Project Content	Settings	Property	Value
	Extend Stack >	API Info	Parameter Checking	Default (BSP)
* MAL/Common	Remove		 Module g_adc0 ADC (r_adc) 	
g_loport I/O Port (r_loport)	a. Reniove		✓ General	
Azure RTOS ThreadX Port (rm_threadx_port) g_adc0 ADC (r_adc)			Name	g_adc
Y ∰ System Thread			Unit	0
# Azure RTOS GUIX			Resolution	12-Bit
g_timer_PWM Timer, General PWM (r_gpt)			Alignment	Right
V @ Touch Thread	- 1 1		Clear after read	On
<pre># g_touch_irg External IRQ (r_icu)</pre>			Mode	Continuous Scan
g_i2c_touch I2C Master (r_iic_master)			Double-trigger	Disabled
			✓ Input	
			 Channel Scan Mask (channel availability variable) 	es by MCU)
g_rtc Realtime Clock (r_rtc)			Channel 0	
g_adc0 ADC (r_adc)			Channel 1	8
			Channel 2	₹
< >>			Channel 3	th (
			Channel 4	A
Objects 🔄 New Object > 🅡 Remove			Channel 5	Č.
g_i2c_semaphore Semaphore			Channel 25	
g_touch semaphore Semaphore			Channel 26	
g_touch_semaphore semaphore			Channel 27	
			Channel 28	
			Temperature Sensor	
			Internal Reference Voltage	
			> Group B Scan Mask (channel availability var	ies by MCU)
mmary BSP Clocks Pins Interrupts Event Links Stacks Components			> Addition/Averaging Mask (channel availabil	ity varies by
mmary BSP Clocks Pins Interrupts Event Links Stacks Components			> Sample and Hold	

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

Stacks Configuration			Genera	te Project Content	g_timer_s	emaphore Sem	aphore
Threads New Thread Remo	g_adc0 dx_port) (r_gpt) ()	ADC (r_adc) Stacks	New Stack		g_timer_s Settings	emaphore Sem Property Name Symbol Initial count	aphore Value Timer Semaphore g_timer_semaphore 0
Objects New Object > R R g_i2c_semaphore Semaphore g_touch_semaphore Semaphore g_timer_semaphore Semaphore	> emove						

Figure 63. Creating g_timer_semaphore

5. Create **system_msg_queue Queue** with the following settings.



"GUIX Thermostat" for **EK-RA8D1** Parallel GLCD Display

Stacks Con	figuration				Generate	Project Content
Threads	🙆 New T	'hread 👔 Ren			New Stack >	
 	port I/O Port (r e RTOS Thread Thread e RTOS GUIX ner_PWM Time 'hread uch_irq Externa	IX Port (rm_thre er, General PWI al IRQ (r_icu) aster (r_iic_mas read ck (r_rtc)	eadx_port)	ADC (r_adc) Stack	S 🐣 Extend Stack	>
g_touch_se g_timer_se	aphore Semap emaphore Sem maphore Sem Clocks Pins Ir	hore aphore aphore	Remove Event Flags Mutex Queue Semaphore t Links Stacks Compo			
*[Themostat × 💽		hmi_event_ha	startup.c C main.c "1	Proper	ties ×	
		. 8	Generate Pro	Settings	Property	Value

Figure 64. Creating system_msg_queue Queue

6. Create **g_sys_mutex Mutex** with the following settings.



"GUIX Thermostat" for **EK-RA8D1** Parallel GLCD Display

Stacks Cor	ofiguration						Generate	Project	Content
								nojeci	
Threads	🕢 New Thread	Remov	e 🖻			1	New Stack >		
V 🎇 HAL/C	ommon			g_adc0 A	DC (r_adc) S	itacks 🚔	Extend Stack	>	
	oport I/O Port (r_ioport	+)				10	Remove		
				1					
	ure RTOS ThreadX Port	(rm_thread)	x_port)	🕀 q a	dc0 ADC (r_	adc)			
💙 🎡 System									
	are RTOS GUIX								
🕀 g_ti	imer_PWM Timer, Gene	eral PWM (r	_gpt)	í					
🗸 🍪 Touch	Thread								
₽ g_t	ouch_irq External IRQ ((r_icu)							
	2c_touch I2C Master (r_								
	erature Time Thread								
		-)							
	tc Realtime Clock (r_rtc	-)							
🗇 g_a	dc0 ADC (r_adc)								
<			>						
-15 50		100							
Objects	된 New Objec	t > 💼 Re	move						
	anhoro Comanhoro	😑 Ev	ent Flags	. 1					
g_izc_sen	naphore Semaphore			11 A.					
		🔵 M	utex						
	emaphore Semaphore								
g_timer_s	emaphore Semaphore	e Qi	ueue						
g_timer_s		e Qi							
g_timer_s	emaphore Semaphore	e Qi	ueue						
g_timer_s	emaphore Semaphore	e Qi	ueue						
 g_timer_s system_m 	emaphore Semaphore sg_queue Queue	QuSe	ueue emaphore		onto				
 g_timer_s system_m 	emaphore Semaphore	QuSe	ueue emaphore		ents				
 g_timer_s system_m Summary BSP 	emaphore Semaphore sg_queue Queue Clocks Pins Interrup	ts Event Li	ueue emaphore	s Compon	ents	Propert	ies ×		
● g_timer_s ● system_m Summary BSP	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev	ts Event Li	ueue emaphore	s Compon	»1 - 0	Propert g sys mi			
 g_timer_s system_m Summary BSP 	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev	ts Event Li	ueue emaphore	s Compon		g_sys_m	utex Mutex		/alue
• g_timer_s • system_m Summary BSP *(Themostat ×] tacks Configura	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev	ts Event Li	ueue emaphore	s Compon	»1			_	/alue
g_timer_s g_timer_s system_m Summary BSP *(Themostat ×) tacks Configura Threads	emaphore Semaphore sg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev tion	ts Event Li	ueue emaphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_system_m Summary BSP IThemostat × tacks Configura Threads	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev tion New Thread @ Remove E Port (r_ioport)	ts Event Li	ueue emaphore nks <u>Stack</u> startup.c	cs Compon	**1 Project Content	g_sys_m	Property Name	2	System Mutex
g_timer_s g_timer_s system_m Summary BSP *(Themostat × tacks Configura Threads €) 1 ✓	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev tion New Thread @ Remove E	e Qu Se ts Event Liu rent_ha C	ueue emaphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_system_m Summary BSP IThemostat × tacks Configura Threads	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup System_threa Christian New Thread Remove Port (r_ioport) ThreadX Port (rm_threadx_port)	e Qu Se ts Event Liu rent_ha C	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_system_m Summary BSP *(Themostat × [tacks Configura threads g_ioport I/O g_loport I/O g_Joport I/O g_Azure RTOS Azure RTOS	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup System_threa Christian New Thread Remove Port (r_ioport) ThreadX Port (rm_threadx_port)	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_system_m Summary BSP Intreads Guidentify Azure RTOS T Azure RTOS T Azure RTOS T Guidentify System Thread Chread	emaphore Semaphore isg_queue Queue Clocks Pins Interrup @ system_threa @ hmi_ev tion New Thread @ Remove Port (r_joport) ThreadX Port (m_threadx_port) SUIX M Timer, General PWM (r_gpt)	e Qu Se ts Event Liu rent_ha C	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP IThemostat × [tacks Configura fhreads	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup System_threa Christian New Thread Remove Port (r_joport) ThreadX Port (rm_threadx_port) SUIX M Timer, General PWM (r_gpt) External IRQ (r_jcu)	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP IThemostat × [tacks Configura fhreads	emaphore Semaphore Isg_queue Queue Clocks Pins Interrup system_threa @ hmi_ev tion New Thread @ Remove Port (r_joport) ThreadX Port (rm_threadx_port) SUIX M Timer, General PWM (r_gpt) External IRQ (r_jcu) I2C Master (r_jic_master)	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_system_m system_m Summary BSP *(Themostat × [tacks Configura Threads @] 1 @ HAL/Common # g.joport I/O # Azure RTOS 1 @ g.timer_PWN @ Touch Thread # g.jicuch.jirg 1	emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread ® Remove © Port (r_joport) ThreadX Port (rm_threadx_port) SUIX d Timer, General PWM (r_gpt) External IRQ (r_jcu) I2C Master (r_jicumaster) me Thread the Clock (r_ntc)	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP *(Themostat × [tacks Configura threads	emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread ® Remove © Port (r_joport) ThreadX Port (rm_threadx_port) SUIX d Timer, General PWM (r_gpt) External IRQ (r_jcu) I2C Master (r_jicumaster) me Thread the Clock (r_ntc)	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP IThemostat × [tacks Configura threads	emaphore Semaphore isg_queue Queue Clocks Pins Interrup system_threa system_threa system_threa system_threa emaphore file system_thread file file system_thread file file file file file file file file	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP *[Themostat × [tacks Configura threads	emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread © Remove © Port (r_ioport) ThreadX Port (m_threadx_port) SUIX M Timer, General PWM (r_gpt) External IRQ (r_icu) I2C Master (r_jic_master) me Thread the Clock (r_rtc) (r_adc)	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s system_m Summary BSP *(Themostat × [tacks Configura threads % Ature RTOS % System Thread for Azure RTOS g_touch Thread g_touch,ing t g_jl2c, touch i g_g_tact, Realtim g_g_d add ADC Dbjects</td <td>emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread @ Remove © Port (r_ioport) ThreadX Port (rm_threadx_port) SUIX M Timer, General PWIM (r_gpt) External IRQ (r_icu) 2C Master (r_iic_master) me Thread to Clock (r_rtc) (r_adc) New Object > @ Remove</td> <td><pre>g_adc0 ADC (0 </pre></td> <td>emaphore maphore nks <u>Stack</u> startup.c</td> <td>s Compon</td> <td>**1 Project Content</td> <td>g_sys_m</td> <td>Property Name Symbol</td> <td>2</td> <td>System Mutex g_sys_mutex</td>	emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread @ Remove © Port (r_ioport) ThreadX Port (rm_threadx_port) SUIX M Timer, General PWIM (r_gpt) External IRQ (r_icu) 2C Master (r_iic_master) me Thread to Clock (r_rtc) (r_adc) New Object > @ Remove	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP IThemostat × [tacks Configura Threads flat_common # g_ioport //0 # Azure RTOS 1 # g_timer_PWN % Touch Thread # g_timer_PWN % Touch Thread # g_tick_PUN % Touch Thread	emaphore Semaphore isg_queue Queue Clocks Pins Interrup system_threa tion New Thread Remove Port (r_joport) ThreadX Port (rm_threadx_port) SUIX M Timer, General PWM (r_gpt) External IRQ (r_jcu) 2C Master (r_jic_master) me Thread to E Clock (r_rtc) (r_adc) New Object > Remove iemaphore	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s system_m Summary BSP *(Themostat × [tacks Configura threads % Ature RTOS % System Thread for Azure RTOS g_touch Thread g_touch,ing t g_jl2c, touch i g_g_tact, Realtim g_g_d add ADC Dbjects</td <td>emaphore Semaphore isg_queue Queue Clocks Pins Interrup system_threa system_threa system</td> <td><pre>g_adc0 ADC (0 </pre></td> <td>emaphore maphore nks <u>Stack</u> startup.c</td> <td>s Compon</td> <td>**1 Project Content</td> <td>g_sys_m</td> <td>Property Name Symbol</td> <td>2</td> <td>System Mutex g_sys_mutex</td>	emaphore Semaphore isg_queue Queue Clocks Pins Interrup system_threa system_threa system	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s system_m Summary BSP (Themostat × [tacks Configura threads	emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread @ Remove © Port (r_ioport) ThreadX Port (m_threadx_port) SUIX M Timer, General PWM (r_gpt) External IRQ (r_icu) 2C Master (r_iic_master) me Thread te Clock (r_rtc) (r_adc) New Object > @ Remove e Semaphore e Semaphore e Semaphore e Semaphore e Semaphore e Semaphore e Semaphore	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex
g_timer_s g_timer_s system_m Summary BSP (Themostat × [tacks Configura tacks Configura threads fineads g_joprt1/0 g_touch,ira [g_itc.cellin g_itc.semaphore S g_timer_semaphor g_timer_semaphore }	emaphore Semaphore isg_queue Queue Clocks Pins Interrup © system_threa © hmi_ev tion New Thread @ Remove © Port (r_ioport) ThreadX Port (m_threadx_port) SUIX M Timer, General PWM (r_gpt) External IRQ (r_icu) 2C Master (r_iic_master) me Thread te Clock (r_rtc) (r_adc) New Object > @ Remove e Semaphore e Semaphore e Semaphore e Semaphore e Semaphore e Semaphore e Semaphore	<pre>g_adc0 ADC (0 </pre>	emaphore maphore nks <u>Stack</u> startup.c	s Compon	**1 Project Content	g_sys_m	Property Name Symbol	2	System Mutex g_sys_mutex

Figure 65. Creating g_sys_mutex Queue

- 7. In RA Configurator, click ^{Generate Project Content} to generate project content.
- 8. Copy and replace these files in "src" folder in e² studio project with the files in "6.2.8" folder in the Lab folder:
 - hmi_event_handler.c -

- system_thread_entry.c
- system_time.c
- system_time.h
- system_api.h
- system_cfg.h
- brightness.c
- brightness.h
- touch_thread_entry.c
- temperature_time_thread_entry.c
- 9. In System Thread, date/time data and temperature data get updated every second. It then sends out events to trigger GUIX updates.
- 10. The following is an example of handling temperature and time update events in the Main Page screen event handler.

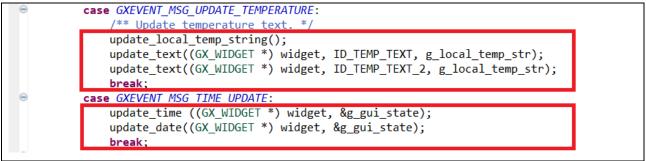


Figure 66: Main page screen event handler

11. **Build, Download, and Run** the e² studio project. You will see time and temperature get updated every second.



Figure 67. Time, Temperature on Main Page Screen



7. Testing and debugging 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 functional 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.

File	Edit Navigate Sear	ch Project	Renesas Views	Ri
	New		Alt+Shift+N >	Simport – 🗆 X
	Open File			Select
ĉ,	Open Projects from File	e System		Rename and Import and Existing C/C++ Project into the workspace
	Recent Files		>	
	Close Editor		Ctrl+W	Select an import wizard: type filter text
	Close All Editors		Ctrl+Shift+W	✓ ➢ General
	Save		Ctrl+S	D Archive File CMSIS Pack
	Save As			🥵 Existing Projects into Workspace 🤤 File System
C	Save All		Ctrl+Shift+S	Preferences
	Revert			Rename & Import Existing C/C++ Project into Workspace
	Move			🚘 Renesas CS+ Project for CC-RX and CC-RL
	Rename		F2	> > > C/C++ > > > Install
8	Refresh		F5	> 😕 Oomph > 📂 Run/Debug
	Convert Line Delimiters	; То	>	> 😕 Team > 🔁 Tracing
	Print		Ctrl+P	> 🦻 XML
		Import	carri	1
	Import	Import		1
G	Export			
	Properties		Alt+Enter	
	Switch Workspace		>	
	Restart			
	Exit			? < Back Next > Finish Cancel

Figure 68. Rename & Import Existing C/C++ Project into Workspace on Import Menu



2. Once the project is imported, open the configuration.xml Stack, click Generate Project Content, compile the project without errors and proceed with the evaluation.

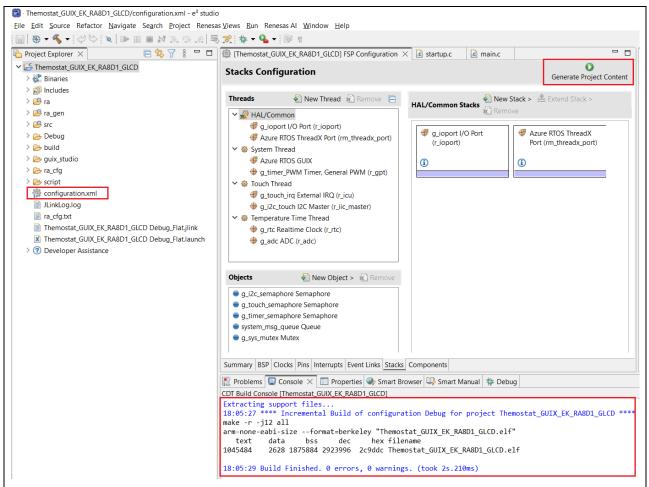


Figure 69. Project imported, Generate Project Content and Built Project.



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 EK-RA8D1 - Evaluation Kit for RA8D1 MCU Group | Renesas renesas.com/ra/forum renesas.com/FSP renesas.com/support



Revision History

		Description		
Rev.	Date	Page	Summary	
1.00	Oct.10.2024	—	Initial release	



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) "Renease 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/.