

Renesas RA Family

# Firmware Update from Host MCU

# Introduction

This document describes a sample application that updates the RYZ012 firmware from the host MCU.

The application example works in a configuration that uses the EK-RA4M2 board with RA4M2 as the host MCU and connects the PMOD<sup>™</sup> Expansion Board for RYZ012 Bluetooth LE Module to the PMOD connector. The steps in this document show the user how to transfer an RYZ012 firmware file to the MCU host, and program the firmware into the RYZ012.

# Target Devices

- RA4M2
- RYZ012

## Related Documents

- Renesas Flexible Software Package (FSP) User's Manual (<u>R11UM0155</u>)
- e<sup>2</sup> studio 2022-10 User's Manual: Quick Start Guide MCU RA Family (<u>R20UT4989</u>)
- EK-RA4M2 Quick Start Guide (R20QS0018)
- RYZ012A1 PMOD Expansion Board Quick Start Guide (R21QS0002)
- RYZ012 Datasheet (R12DS0002)
- RYZ012 Bluetooth LE Sample Application (<u>R01AN6116EJ</u>)
- QE for BLE [RA,RE,RX] Release Note (R20UT5145EJ)

#### **Required Resources**

To build and run the RYZ012 firmware update application example, the following resources are needed.

#### Development tools and software

- e<sup>2</sup> studio IDE v2022-10 (<u>e2studio</u>)
- Flexible Software Package (FSP) v4.2.0 (Flexible Software Package FSP)
- QE for BLE Tool [RA, RE, RX] V1.5.0 for e<sup>2</sup> studio IDE (<u>QE for BLE: Tool for Bluetooth<sup>®</sup> Low Energy</u>)
- SEGGER J-Link RTT Viewer V7.60f (SEGGER RTT Viewer)
- Tera Term V4.106 (<u>Tera Term Open-Source Project</u>)
- GATTBrowser v1.0.3 or Android or v1.1.4 iOS (see Google Play or iOS App Store)

#### Hardware

- EK-RA4M2 kit (RTK7EKA4M2S00001BE) EK-RA4M2
- PMOD Expansion Board for RYZ012x1 (RTKYZ012A1B00000BE) <u>PMOD Expansion Board</u> Must be programmed with v5.4 Firmware or later to support MCU based firmware updates
- PC running Windows<sup>®</sup> 10
- 2 x Micro USB cables

PMOD<sup>™</sup> is registered to Digilent Inc.



# Contents

1.	Overview	4
1.1	Operating Environment	4
1.1.1	Hardware	4
1.1.2	Software	4
1.1.3	How to Assemble RYZ012 Module and EK-RA4M2	5
1.1.4	EK-RA4M2 Jumper Setting	5
1.2	RYZ012 Firmware Update Process	6
2.	Firmware Update Application	7
2.1	Importing the Application Project	7
2.1.1	Specify e <sup>2</sup> studio Workspace	8
2.1.2	Import Project	8
2.1.3	Select Existing Project	9
2.1.4	Select Project	9
2.2	Project Build and Download	. 10
2.3	Application Project Operation	. 10
2.3.1	Launch J-Link RTT Viewer	. 10
2.3.2	Start the Application Execution	. 11
2.3.3	Launch Tera Term	. 12
2.4	RYZ012 Firmware Update Considerations	. 17
2.4.1	RYZ012 Firmware Image Invalid / Update Failed	. 17
2.4.2	RYZ012 Image Not Found	. 18
2.4.3	Tera Term XMODEM Transfer Halt	. 19
2.4.4	Initialize BLE Driver Failed	. 19
3.	Application Software Architecture	.20
3.1	Microcontroller Peripheral Functions	. 22
3.2	FSP Modules	. 22
4.	Application Project Implementation	.23
4.1	Entry Point	. 23
4.2	Main Loop	. 24
4.3	BLE Initialization Process	. 26
4.4	Register BLE Callback Functions	. 27
4.5	Firmware Image Helper Functions	. 28
4.6	Get Firmware Version Function	. 29
4.7	Firmware Update Function	. 30
5.	How to Make and Configure a New Project	.30
5.1	Create a new project	. 31
5.2	BSP Heap and Stack configuration	. 31



# RYZ012 and RA MCU

5.3	Add the I/O Port Stack	31
5.4	Add and Configure the SPP BLE Module	32
5.4.1	SPP BLE Abstraction Driver Configuration	32
5.4.2	Configure Peripherals for SPP BLE Abstraction Driver	34
5.5	Add USB PCDC for Tera Term Menu and XMODEM File Transfer	38
5.6	Add MCU Flash Driver for XMODEM File Storage	38
5.7	Add General PWM Timers for LED Indications	40
5.8	Create BLE Profile with QE for BLE	40
5.9	Build and Modify Application Skeleton Code	41
6.	Next Steps	41
Revis	sion History	45



# 1. Overview

The RYZ012 BLE module is a highly integrated wireless communication module that provides a pre-certified solution for Bluetooth<sup>®</sup> 5.0 Low Energy (LE). The module is available in two configurations (RYZ012A1 and RYZ012B1) with or without a mounted antenna. Supported by the RA MCU family's Flexible Software Package (FSP) and the QE for BLE tool, customers can focus on application development without dealing with the details of Bluetooth LE.

Once an RYZ012 module is designed into an end customer application, designers will need to be able to update the firmware to adapt to changing conditions or requirements for the end systems. This update could be done via the host MCU or through a BLE radio update. This application note covers an example of updating the RYZ012 module via the host MCU including details of handling issues encountered when updating the BLE module.

The application project uses an EK-RA4M2 board connected with an RYZ012 module. Communication between the EK-RA4M2 and RYZ012 module is based on a command system called Serial Port Profile (here in after referred to as SPP).

Please check the *SPP Bluetooth Low Energy Abstraction with RYZ012 (rm\_ble\_abs\_spp)* part of the *RA Flexible Software Package User's Manual* (<u>R11UM0155</u>) for more information on the APIs and callback event limitations.

# 1.1 Operating Environment

#### 1.1.1 Hardware

The hardware requirements used in the sample application are shown in the following table.

#### Table 1. Hardware Requirements

Hardware	Description
EK-RA4M2	RTK7EKA4M2S00001BE
PMOD Expansion Board for RYZ012x1	RTKYZ012A1B00000BE
	Programmed with SPP SDK v5.4 Firmware or later to support MCU based firmware updates
Windows <sup>®</sup> 10 PC	
2 x Micro USB Cables	for EK-RA4M2 USB Debug J10 connecter (micro-B)
	and USB FS J11 conn to PC Tera Term (micro-B)

#### 1.1.2 Software

The software requirements used in the sample application are shown in the following table.

#### Table 2. Software Requirements

Software	Version
e <sup>2</sup> studio IDE	2022-10
GCC Compiler	10.3.1
Renesas FSP	4.2.0
QE for BLE [RA,RE,RX]	V1.5.0 for e <sup>2</sup> studio IDE
SEGGER J-Link RTT Viewer	V7.60f
Tera Term	V4.106
GATT Browser	v1.0.3 Android, or v1.1.4 iOS



# 1.1.3 How to Assemble RYZ012 Module and EK-RA4M2

This section describes how to assemble RYZ012 PMOD module and EK-RA4M2. The RYZ012 PMOD module and EK-RA4M2 are connected by one of 2 x 6 PMOD connectors. In this application note, the PMOD connector must be mounted as:

#### RYZ012 PMOD : CN1 > EK-RA4M2 : J26 PMOD1 (SPI / UART)

Connect CN1 of RYZ012 PMOD and J26 PMOD 1 Connector of EK-RA4M2.



Figure 1. EK-RA4M2 and RYZ012 PMOD Assembly

#### 1.1.4 EK-RA4M2 Jumper Setting

The USB Full Speed interface of EK-RA4M4 is used to communicate with Tera Term for XMODEM file transfer. Set the EK-RA4M2 J11 USB FS option jumpers as follows :



- J12: Jumper installed on pins 2-3
- J15: Jumper installed on pins 1-2

Connect the two micro-B USB cables to USB Full Speed (J11) and DEBUG1 (J10) with other cable ends to PC ports.

# 1.2 RYZ012 Firmware Update Process

In this application note, the RYZ012 module firmware is upgraded by a host MCU. The RYZ012 PMOD is connected to the EK-RA4M2 board. Only the RYZ012 firmware is upgraded and the host MCU is not upgraded.

The RYZ012 firmware image files to use are supplied in the accompanying application project. See the root directory of the project that is imported into e<sup>2</sup> studio. Instructions on how to import, build, and run the application project are provided in section 2, Firmware Update Application.

The firmware image file that is used to upgrade the RYZ012 PMOD is opened and transferred to the MCU with Tera Term XMODEM file transfer. The entire RYZ012 firmware image is stored in MCU Flash prior to upgrading the RYZ012 module. Tera Term is then used to trigger the upgrade process where the MCU checks for presence of the firmware image in MCU Flash memory and then updates the RYZ012 with the new image. The status of the upgrade process is shown on the RTT Viewer.

After the RYZ012 is updated, it restarts and runs the new image. Then the MCU reads the RYZ012 firmware version from the module and displays it in RTT Viewer.





Figure 2. MCU User Application Flow Diagram

Figure 2 shows the MCU user application (**project/qe\_gen/ble/app\_main.c**) flow diagram for the firmware update process. The FSP provided helper function **R\_BLE\_VS\_UpdateModuleFirmware(...)** simplifies the work required by calling the SPP command sequence required to perform the upgrade.

See Figure 3 for details of the command sequence that is executed.





Figure 3. SPP Command Sequence for RYZ012 Firmware Update

- Note that the RYZ012 must receive the **SPP\_CMD\_REBOOT\_BLE** command, else the firmware upgrade will not fully complete to run the new version firmware image. It will run the roll-back image, which is the version that ran prior to starting the upgrade.
- All firmware image frames must be sent with SPP\_CMD\_SEND\_FW\_DATA commands for the transferred firmware image to pass the integrity check. If any of the data frames are missing or corrupted during the transfer process, the new image will fail integrity check and the RYZ012 will run the roll-back image instead after SPP\_CMD\_REBOOT\_BLE is issued.
- If the RYZ012 is reset during the SPP command sequence prior to completing all the update steps, then the RYZ012 will run the roll back image.
- If the RYZ012 loses power during the SPP command sequence prior to completing all the update steps, then the RYZ012 will run the roll back image when power is restored.

To start using the application project immediately, see section 2, Firmware Update Application.

The design details of the application software architecture are covered in section 3, Application Software Architecture.

The implementation details of the application software are covered in section 4, Application Project Implementation.

# 2. Firmware Update Application

# 2.1 Importing the Application Project

The steps to import the application project into e<sup>2</sup> studio are shown in the following sections.



# 2.1.1 Specify e<sup>2</sup> studio Workspace

Launch e<sup>2</sup> studio, specify the workspace directory, and click the **Launch** button.

🔋 e² studio Launcher	>
elect a directory as workspace	
e² studio uses the workspace directory to s	store its preferences and development artifacts.
-	
Norkspace: <sup>9</sup> C:¥MyWorkspace	✓ Browse
Use this as the default and do not ask ag	gain
Recent Workspaces	
	Launch Cancel
	Launch Cancel

Figure 4. e<sup>2</sup> studio Workspace

# 2.1.2 Import Project

Select File > Import from the menu bar.

0	MyWo	rkspace -	e² stu	dio				
File	Edit	Navigat	e Se	arch	Project	R	lenesas Views	R
	New						Alt+Shift+N	>
	Open	File						
È,	Open	Projects	from F	ile Sys	tem			
	Recen	nt Files						>
	Close	Editor					Ctrl+W	
	Close	All Editor	s				Ctrl+Shift+W	
	Save						Ctrl+S	
	Save /	As						
ß	Save /	All					Ctrl+Shift+S	
	Rever	t						
	Move							
	Renar	me					F2	
\$	Refree	sh					F5	
	Conve	ert Line D	elimite	ers To				>
۵	Print.						Ctrl+P	
2	Impor	rt	N					
4	Expor	rt	h	ŝ				
	Prope	erties					Alt+Enter	
	Switc	h Worksp	ace					>
	Resta	rt						
	Exit							

Figure 5. Importing Project



#### 2.1.3 Select Existing Project

Select Existing Projects into Workspace and click Next.

Import —           Select           Create new projects from an archive file or directory.		×
Select an import wizard: type filter text    General    CMSIS Pack   CMSIS Pack   CMSIS Pack   File System  File System  Preferences  Rename & Import Existing C/C++ Project into Workspace   Rename & Import Existing C/C++ Project into Workspace		~
(?) < <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel	

Figure 6. Selecting Existing Projects

#### 2.1.4 Select Project

Choose **Select root directory**, click **Browse** and select the directory for the project to import. Check the box in **Projects**: window and click **Finish** to import the project. If importing from the zip file, then choose **Select archive file** instead and navigate to zip file to import.

Import	
Import Projects	
Select a directory to search for existing Eclipse projects.	
C:\kenesas\e2_studio\workspace\mcu_rwuj <	Browse
O Select archive file:	Browse
Projects:	
mcu_fwupdate_ryz012_ra4m2_baremetal (C:\Renesas\e2_studio\wd	Select All
	Decelect All
	Deserver Air
	Refresh
x	
Ontions	
Search for nested projects	
Copy projects into workspace	
Close newly imported projects upon completion	
Hide projects that already exist in the workspace	
- Working sets	
Add project to working sets	New
Working sets:	Select
in the second seco	Jereetin .
(?) < Back Next > Finish	Cancel
L	

Figure 7. Selecting Existing Projects



## 2.2 Project Build and Download

- 1. Select **Project > Build Project** from the menu bar or click the Build icon store to build the project.
- 2. Make sure that the hardware is connected according to section 1.1, Operating Environment. Click the debug icon 🔅 to launch the project. When the project starts, the application will be downloaded to the EK-RA4M2.

# 2.3 Application Project Operation

#### 2.3.1 Launch J-Link RTT Viewer

Launch J-Link RTT Viewer, set as follows, and click the **OK** button.

- Connection to J-Link : USB
- Specify Target Device : R7FA4M2AD
- Target Interface & Speed : SWD, 4000 kHz
- RTT Control Block : Address, 0x20000f14

J-Link RTT	Viewer V7.60f   Configuration	×
Connection to	)-Link	
O Existing Ses	sion	
Specify Target	Device	
R7FA4M2AD		×
-Script file (opti	nal)	
- Target Interfac	e & Speed	
SWD	•	4000 kHz 🔻
RTT Control Blo	ck	
<ul> <li>Auto Detect</li> </ul>	ion 🖲 Address 🛛 🔾 S	earch Range
Enter the addre Example: 0x200	ss of the RTT Control block. 00000	
0x20000f14		
	ОК	Cancel

Figure 8. J-Link RTT Viewer Configuration

Note: The RTT Control Block Address can be found in the .bss.\_SEGGER\_RTT section address of the map file generated in the Debug directory. If the application source code is modified with your custom changes, this address will change.

🍋 Project Explorer 🗙 📄 😫 🍸 🖇	🗖 🗖 📄 mcu_fwupd	ate_ryz012_ra4m2_barem	etal.map $ imes$		
∨ 👺 mcu_fwupdate_ryz012_ra4m2_baremetal [Debug]	3918		0x80030000		ospi device 1 region start ad
> 💒 Binaries	3919		0x90030000		ospi device 1 region end add
> 🔊 Includes	3920		0x80030000		tz_OSPI_DEVICE_1_N =ospi_
> 🖂 ge gen	3921				
	3922	.noinit	0x20000ed8	0x20	
	3923		0x20000ed8		. = ALIGN (0x4)
> 🛃 ra_gen	3924		0x20000ed8		noinit_start = .
> 🐸 src	3925	*(.noinit*)			
🗸 🗁 Debug	3926	.noinit	0x20000ed8	0x1c	./ra/fsp/src/bsp/mcu/all/bsp_clock
> 🕞 ge gen	3927		0x20000ef8		. = ALIGN (0x8)
) (a. ra	3928	*fill*	0x20000ef4	0x4	
	3929	*(.heap.*)			
> 🔁 ra_gen	3930		0x20000ef8		noinit_end = .
> 🗁 src	3931				
> 🕸 mcu_fwupdate_ryz012_ra4m2_baremetal.elf - [arm/le]	3932	.bss	0x20000ef8	0x1cfc	
ham makefile	3933		0x20000ef8		. = ALIGN (0x4)
makefile init	3934		0x20000ef8		bss_start = .
modelineand	3935	*(.bss*)			
mcu_rwupdate_ryzv12_ra4m2_baremetai.eir.in	3936	.bss	0x20000ef8	0x1c	c:/users/a5134013/appdata/local/pr
mcu_fwupdate_ryz012_ra4m2_baremetal.hex	3937	.bssSEGGER_	RTT		
mcu_fwupdate_ryz012_ra4m2_baremetal.map	3938		0x20000f14	0xa8	./src/SEGGER_RTT/SEGGER_RTT.o
mcu fwupdate ryz012 ra4m2 baremetal.rpd	3939		0x20000f14		_SEGGER_RTT

Figure 9. RTT Control Block Address



#### 2.3.2 Start the Application Execution

Click the resume icon  $\square$  in the e<sup>2</sup> studio Debug Perspective to run the application.

RTT Viewer logging will show that the application is running, and 3 board LEDs (blue, green, red) will blink on/off to indicate that the program is running. In Figure 10, observe the instructions to start Tera Term.

If the LEDs fail to blink and the white LED 4 is not lighted, then verify a USB cable is plugged into USB Full Speed J11 (with other end connected to PC) and the jumpers are set correctly for J12, J15. See section 1.1.4, EK-RA4M2 Jumper Setting for more details.

	-	
File Terminals Input Logging Help		
All Terminals Terminal 0		
00>		
00> ** Start RTT Viewer for Logging ** 00>		
00> ** Start Tera Term for XMODEM File Transfer of RYZ012A FW Image to MCU **		
00> ** Tera Term : New connection->Serial->COMx : USB Serial Device (COMx) **		
00> 00> ** Hit Enter Key (CR) in Tera Term Window **		
	Enter	Clear
	Enter	Clear
LOG: [0][0]: E007E000 LOG: [0][0]: E0044000 CID B105900D PID 00588906 DEVARCH 00000000 DEVTYPE 14 CTI	Enter	Clear
LOG: RUMIDILOJ U E00F2000 LOG: [0][0]: E0044000 CID B105900D PID 0058B906 DEVARCH 00000000 DEVTYPE 14 CTI LOG: [0][1]: E0047000 CID B105900D PID 003B8908 DEVARCH 00000000 DEVTYPE 12 CSTF LOG: [0][2]: E0047000 CID B105900D PID 003B8961 DEVARCH 00000000 DEVTYPE 12 CSTF	Enter	Clear
LOG: ROHIDI[0] @ COOFCOOD LOG: [0][0]: E0044000 CID B105900D PID 0058B906 DEVARCH 00000000 DEVTYPE 14 CTI LOG: [0][1]: E0047000 CID B105900D PID 0038B908 DEVARCH 00000000 DEVTYPE 12 CSTF LOG: [0][2]: E0048000 CID B105900D PID 001B8061 DEVARCH 00000000 DEVTYPE 21 TMC LOG: [0][3]: E0049000 CID B105900D PID 001B8101 TSG LOG: [0][3]: E0049000 CID B105900D PID 001B8101 TSG	Enter	Clear
LOG: RUMIDI[0] @ E00F2000 LOG: [0][0]: E0044000 CID B105900D PID 0058B906 DEVARCH 00000000 DEVTYPE 14 CTI LOG: [0][1]: E0047000 CID B105900D PID 003B8908 DEVARCH 00000000 DEVTYPE 12 CSTF LOG: [0][2]: E0048000 CID B105900D PID 001B8061 DEVARCH 00000000 DEVTYPE 21 TMC LOG: [0][3]: E0049000 CID B105900D PID 001B8101 TSG LOG: [0][4]: E00440000 CID B105900D PID 001B8101 TSG LOG: [0][5]: E00FF000 CID B105900D PID 0008B84C9 ROM Table	Enter	Clear
LOG:       RUMIDI[0] @ E00FE000         LOG:       [0][0]:       E0044000       CLD B105300D PID 00588306 DEVARCH 0000000 DEVTYPE 14 CTI         LOG:       [0][1]:       E0047000       CLD B105300D PID 00388308 DEVARCH 0000000 DEVTYPE 12 CSTF         LOG:       [0][2]:       E0044000       CLD B105500D PID 00188301 DEVARCH 0000000 DEVTYPE 21 TWC         LOG:       [0][3]:       E00440000       CLD B105500D PID 00188101 TSG         LOG:       [0][5]:       E00440000       CLD B105500D PID 0008B201 DEVARCH 00000000 DEVTYPE 11 Cortex-M33         LOG:       [0][5]:       E00FF000       CLD B105500D PID 0008B420 ROM Table         LOG:       [0][5]:       E00FF000       CLD B105100D PID 0008B21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33	Enter	Clear
LOG: R0H1D1[0] @ E00FE000	Enter	Clear
LOG: ROHIDI[0] @ E00FE000 LOG: [0][0]: E0044000 CID B105900D PID 005BB906 DEVARCH 00000000 DEVTYPE 14 CTI LOG: [0][1]: E0047000 CID B105900D PID 003BB908 DEVARCH 00000000 DEVTYPE 12 CSTF LOG: [0][2]: E0043000 CID B105900D PID 001BB361 DEVARCH 00000000 DEVTYPE 21 TMC LOG: [0][3]: E0049000 CID B105900D PID 001BB301 T56 LOG: [0][4]: E0040000 CID B105900D PID 001BB301 T56 LOG: [0][4]: E0040000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33 LOG: [0][5]: E004F7000 CID B105900D PID 000BBD21 DEVARCH 00000000 DEVTYPE 11 Cortex-M33 LOG: [0][5]: E004F000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33 LOG: [1][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 PMT LOG: [1][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 FPB LOG: [1][3]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB LOG: [1][3]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB	Enter	Clear
LUG:       ROMIDI[0] @ 20072000         LUG:       [0][0]:       E0044000 CID 8105900D PID 00588906 DEVARCH 00000000 DEVTYPE 14 CTI         LOG:       [0][1]:       E0044000 CID 8105900D PID 00388908 DEVARCH 00000000 DEVTYPE 12 CSTF         LOG:       [0][2]:       E0043000 CID 8105900D PID 00188961 DEVARCH 00000000 DEVTYPE 12 CSTF         LOG:       [0][3]:       E0043000 CID 8105700D PID 00188101 TS6         LOG:       [0][4]:       E0047000 CID 8105100D PID 000881021 DEVARCH 00000000 DEVTYPE 11 Cortex-M33         LOG:       [0][5]:       E00FF000         LOG:       [1][6]:       E00FF000         LOG:       [1][6]:       E00FF000         LOG:       [1][2]:       E000F000         LOG:       [1][3]:       E00FF000         LOG:       [1][2]:       E000F000         LOG:       [1][3]:       E00FF000         LOG:       [1][3]:       E00FF000         LOG:       [1][3]:       E00FF000       E00HARCH 47701A03       DEVTYPE 00	Enter	Clear
LOG:       ROWIDI[0] @ C007C000	Enter	Clear
ION: ROWIDI[0] @ COOPERSON	Enter	Clear
LOG: ROWHDI[0] @ E0072000	Enter	Clear

Figure 10. RTT Viewer – Instructions to Start Tera Term



# 2.3.3 Launch Tera Term

Launch Tera Term, set as follows, and click the **OK** button. The Serial COM port may be different for your PC. Choose the **Port > COMx: USB Serial Device (COMx)**, where X is the number assigned by your PC. See Figure 11 for details.

Tera Term: New co	nnection X	
⊖ TCP/IP	Host: myhost.example.com <	
	✓ History Service: ○ Telnet TCP port#: 22	
	$\odot$ SSH SSH version: SSH2 $\sim$	
	$\odot$ Other IP version: AUTO $\sim$	
● Serial	Port: COM4: USB Serial Device (COM4)	
	OK Cancel Help	

#### Figure 11. Tera Term Connect Configuration

Select and click inside the Tera Term window to receive keyboard input and press the **Enter** key to get the Option Menu shown below in Figure 12.



#### Figure 12. Tera Term Menu

Press 1 to select the RYZ012 firmware image file to transfer to the MCU Flash. See Figure 13.





Figure 13. Tera Term – Press 1 File Transfer to MCU

Navigate to the **root directory** of the Application Project imported in the e<sup>2</sup> studio workspace and select one of the RYZ012 firmware files such as **8258\_moduleV5\_5.bin** and Click **Open**. See Figure 14.

File Edit Setup Control Window He	p			<u>^</u>	
Press 1 RYZ012 File Transfer to	o MCU, 2 Write Ima	ige to RYZ012			
Blank Checking the RY2012 firm	vare image slot				
RIZUIZ FIPMWARE IMAGE SIOT DIA	IK				
Start Xmodem transfer					
Xmodem send : File->Transfer->	MODEM->Send				
Choose RYZ012 Binary Image File					
System will automatically reset	t after successful	l download			
Tera Term: XMODEM Send					$\times$
Look in: 📙 mcu_fwupdate_ryz012_ra4m2_barem 🗸 🞯 🎓 📂	·				
Name	Date modified	Туре	Size		^
.secure_azone	7/13/2022 12:33 PM	SECURE_AZONE File	4 KB		
.secure_xml	6/28/2022 3:24 PM	SECURE_XML File	7 KB		
8258_moduleV5_4.bin	2/4/2022 10:31 AM	BIN File	93 KB		
8258_moduleV5_5.bin	3/30/2022 10:35 AM	BIN File	101 KB		
configuration.xml	6/28/2022 3:24 PM	XML Document	65 KB		
mcu_fwupdate_ryz012_ra4m2_baremetal Debug_Flat.jiink	7/13/2022 12:33 PIVI	JLINK FILE	1 KB		
R7FA4M2AD3CFP.pincfq	6/25/2022 10:35 AM	PINCEG File	2 KB		
a_cfg.txt	6/28/2022 3:24 PM	Text Document	44 KB		~
File name: 8258_moduleV5_5.bin				Open	
Files of type: All(*.*)				<ul> <li>✓ Cancel</li> </ul>	
				Help	
Option					

#### Figure 14. Tera Term – Select File XMODEM to Send

The file will start transferring to the EK-RA4M2 MCU Flash. If the file transfer does not start after the file is selected, then either **Press 1** option was not performed prior to choosing the file or too much time elapsed after **Press 1** option to the time the file was chosen and you may see the error message shown in Figure 16.

In this case, hit the enter key in Tera Term to start over and press 1 again, navigate to and select the file to transfer.

When the file transfer starts, the transfer progress dialog will be shown as in Figure 15.

# RYZ012 and RA MCU







Figure 16. Tera Term – XMODEM Timeout Waiting on File



#### Figure 17. Tera Term – XMODEM Image Transferred to MCU

After the file transfer has completed, "Image Transferred to MCU" will appear in Tera Term and you will be prompted to "Hit Enter Key in Tera Term" and then "Choose Option 2 to Write the Image" to the RYZ012. See Figure 17.





Figure 18. Tera Term – Choose Option 2 Write Image to RYZ012

Hit **Enter** key in Tera Term window to get the Option menu, then "**Press 2 Write Image to RYZ012**". See Figure 18.

Next you will see Tera Term logging update as shown in Figure 19.



Figure 19. Tera Term – Option 2 Writing Image to RYZ012 In Progress

At this point, you are done using Tera Term so move your attention to RTT Viewer to see the firmware upgrade status there. You should observe logging similar to Figure 20.

RTT Viewer will show the results of the RYZ012 firmware upgrade process.







When the RYZ012 firmware upgrade completes with success, RTT Viewer will show logging as shown in Figure 21.



Figure 21. RTT Viewer – RYZ012 Firmware Image Update Success

Notice that the RYZ012 firmware version was originally v5.4 before the upgrade, and then after upgrade, is running at v5.5.



To verify that the RYZ012 is functioning use the GATT Browser app on a mobile device and scan for BLE devices. The RYZ012 module will advertise as "RYZ012" when the firmware upgrade has completed.

In the event that the firmware upgrade fails with error messages shown in RTT Viewer, the next section 2.4, RYZ012 Firmware Update Considerations will help you to troubleshoot and resolve the issues.

# 2.4 RYZ012 Firmware Update Considerations

In the event that the upgrade process fails, see the following information to help troubleshoot your issue.

- First verify that the EK-RA4M2 and RYZ012 PMOD have been setup and connected correctly. Refer to section 1.1, Operating Environment.
- Next, verify the RYZ012 firmware image transferred is the one of two available images included in the app project.

The firmware images are located in the e<sup>2</sup> studio project workspace at the project directory

#### mcu\_fwupdate\_ryz012\_ra4m2\_baremetal/8258\_moduleV5\_4.bin or

#### mcu\_fwupdate\_ryz012\_ra4m2\_baremetal/8258\_moduleV5\_5.bin:

- The RYZ012 has built-in verification steps to recover from an incorrect or failing firmware update.
- The RYZ012 attempts to revert to the previous image if the update fails or halts.
- Once a firmware upgrade succeeds, the user must re-run the entire firmware update process to downgrade to a previous firmware version.
- Review the troubleshooting section below for solutions to the most common issues. For all other issues that cannot be resolved in this section, please check the FAQs, search the Knowledge Base, or submit a support ticket at <u>Renesas Support</u>

#### 2.4.1 RYZ012 Firmware Image Invalid / Update Failed



Figure 22. RTT Viewer – RYZ012 Firmware Image Invalid / Update Failed

In Figure 22, notice that the RYZ012 firmware version is v5.4 before the upgrade starts and is at v5.4 after the upgrade ends. This indicates that the firmware image was either corrupted, had an invalid signature, or not all packets were received during the transfer to the RYZ012. In this case, the integrity check failed on the RYZ012 before programming the memory in the RYZ012. Since the transferred image failed integrity check, the old image v5.4 (roll back image) was retained and selected by the RYZ012 to execute.



Make sure that the EK-RA4M2 board remains powered during the file transfer and write to RYZ012. If the EK-RA4M2 board is reset during the firmware upgrade then the file integrity check will fail on the transfer and the RYZ012 will run the last known good image version (roll back image).

#### 2.4.2 RYZ012 Image Not Found

For the case that a RYZ012 firmware image is not found, see Figure 23. This can be due to the XMODEM firmware transfer to MCU Flash failing or the image file did not fit into the MCU Flash space allocated.

First reset the EK-RA4M2 with Red Reset button (S3), then select the Tera Term window, and Press 1 to Select the RYZ012 FW image file to transfer to the MCU Flash. See Figure 13 to confirm all the steps of the XMODEM file transfer.

If the issue continues, then verify that the MCU Flash size can hold the entire firmware image and check the **#defines** match your chosen MCU and the required firmware image size in section 6, Next Steps, step 4.



Figure 23. RTT Viewer – RYZ012 Image Not Found



#### 2.4.3 Tera Term XMODEM Transfer Halt

Figure 24. Tera Term XMODEM Transfer Halt : Image Size Exceeds MCU Flash Allocated

Figure 24 shows the result when the RYZ012 firmware image XMODEM transfer halts unexpectedly. Notice that the transfer progress % (green) stops, and the entire file is not transferred to the MCU. This occurs because the MCU Flash does not have enough space allocated to write the entire RYZ012 firmware image.

Verify that the MCU Flash size can hold the entire firmware image and check that the **#defines** match your chosen MCU and required firmware image size in section 6, Next Steps, step 4.

#### 2.4.4 Initialize BLE Driver Failed

	_	
🔜 J-Link RTT Viewer V7.60f		×
File Terminals Input Logging Help		
All Terminals Terminal 0		
00> 00> RYZ012 Module Firmware Update : Success		^
00>		
00> BLE Driver Close : Success		
00> Init BLE Driver and Profiles		
00>		
00> BLE Driver and Profile Init : Success		
00> RYZ012 FW Version v5.4		
00>		
00> Running RYZ012 FW 00>		
00> ** Start RTT Viewer for Logging **		
00> ** Start lera lerm for XMODEM File Transfer of RY2012 FW Image to MCU **		
00> ** Tera Term : New connection->Serial->COMx : USB Serial Device (COMx) **		
00> 00> ** Hit Enter Kov (CD) in Tona Tona Window **		
00>		
00> ** Start RTT Viewer for Logging **		
00> 00> ** Start Tera Term for XMODEM File Transfer of RV7012 EW Image to MCU **		
00>		
00> ** Tera Term : New connection->Serial->COMx : USB Serial Device (COMx) **		
00> 00> ** Hit Enter Kev (CR) in Tera Term Window **		
00>		
00> ** Start MCU FW Update of RYZ012 **		
00> Initialize BLE Driver		
00> [ERR] In Function: app_main(),		
00> ** Initialize BLE Driver : failed ** 00> [EPR] In Eurotion: ann main()		
00> ** Check that RYZ012 is connected to Eval Kit **		
00> [ERR] In Function: app_main(),		
00> ** Check that RYZ012 is inserted with correct pin orientation **		
00> Returned Error Code: 0x5		~

Figure 25. Init BLE Driver Failed



If you receive the **Initialize BLE Driver: failed** message then check that the RYZ012 PMOD is installed correctly in PMOD1 connector (J26) on the EK-RA4M2. Make sure the PMOD Mode Select GPIO pin is set for the correct initial logic state for the SPI or UART mode that you are using. See section 6, Next Steps, step 1 to review the SPP BLE Module and PMODx configuration settings. Make certain the RYZ012 PMOD has a factory image programmed before attempting module upgrade.

# 3. Application Software Architecture

The application software architecture for the MCU based RYZ012 firmware upgrade project is covered in this section.



Figure 26. Application Software Architecture (Bare Metal)

Figure 26 shows the software architecture of a Bluetooth LE application in a BareMetal environment. The BLE Application performs initialization and BLE related processing. The QE for BLE tool generates C source code for the Bluetooth LE base skeleton program for the MCU application (located in project **qe\_gen/ble/app\_main.c**) and the BLE Profile. The base program is extended further for the functionality required in the user application. In this case, we are using the FSP BLE Abstraction APIs with SPP APIs to update the firmware of the RYZ012 Module.

# For more information on FSP APIs see **Renesas Flexible Software Package (FSP) User's Manual** (<u>R11UM0155</u>) and section on SPP Bluetooth Low Energy Abstraction with RYZ012 (rm\_ble\_abs\_spp).

The FSP SPP APIs are used by the MCU to communicate with the RYZ012 module. **Figure 3-2** shows the software components of the application running on the RA4M2 MCU with a comms interface to the RYZ012. The user application and BLE Profile are customized to add features unique to your application.





Figure 27. MCU Host and Bluetooth LE Serial Port Profile Interface

Figure 27 provides a more detailed look at the interface of the SPP and BLE APIs provided by FSP. The user application calls the BLE ABS APIs to interact with the RYZ012 BLE Module via the SPP Command Generation (SPP Driver) and SPP Command Interpreter residing on the RYZ012. The MCU communications interface to the RYZ012 PMOD can be UART or SPI. The accompanying software project uses the SPI interface.

The details of the update process are covered in section 1.2, RYZ012 Firmware Update Process.

The details of the application project software implementation are covered in section 4, Application Project Implementation.



# 3.1 Microcontroller Peripheral Functions

The microcontroller peripheral functions used in the application project are shown below.

Table 3. Microcor	troller peripher	al functions
-------------------	------------------	--------------

Module	Pin	Description
Serial	SCI0(SPI0)	SPI communication with RYZ012 on PMOD1 (J26)
Communication		MOSI : P207
Interface		MISO : P206
		SCK : P400 SCLK
		SS : P401 Chip Select
		IRQ12 : P008 Interrupt Request
		RESET : P403
		MODE SELECT : P402 SPI Mode (High)
General-Purpose	GPT0	LED Blue PWM
Timer	GPT1	LED Red PWM
	GPT2	LED blink timer
	GPT3	LED Green PWM
USB 2.0 Full-	USBFS	Communication with Tera Term for XMODEM transfer and User
Speed Module		Input
I/O Port	P008	Interrupt Request RYZ012 PMOD1
	P403	Reset pin control of RYZ012 PMOD1
	P402	Mode Select (HIGH = SPI) of RYZ012 PMOD1
	P103	Blinker Timer PWM
	P415	User LED1 (Blue) PWM
	P404	User LED2 (Green) PWM
	P405	User LED3 (Red) PWM
Flash Memory	FLASH0	MCU storage for RYZ012 firmware image
External Interrupt	IRQ12	RYZ012 PMOD1 Interrupt Request
Request		

# 3.2 FSP Modules

The FSP modules used in the application project are shown below. See Figure 28, FSP Module Summary for a view of all the modules in the project.

Module Type	Module Name		Usage
System	I/O Port	r_ioport	GPIOs and LED indicators
BLE API	SPP BLE	r_ble_abs_spp	RYZ012 BLE
	Abstraction		
Connectivity	USB PCDC	r_usb_pcdc	Tera Term XMODEM File
		r_usb_basic	Transfer & FW Upgrade
Storage	Flash High	r_flash_hp	MCU Storage for RYZ012 FW
	Performance		
Input	External IRQ	r_icu	RYZ012 PMOD Interrupt Req
Connectivity	SPI	r_sci_spi	RYZ012 PMOD comms
Timers	Timer	r_gpt	Blue, Red, Green Blink LED PWM

Note: This application program is a bare metal version.



Threads 📀 New Thread 💼 Remo	ve 📄 HAL	/Common Stacks						
<ul> <li>Alu/Common</li> <li>gi jopol U Ø Þrtí (r.jopot)</li> <li>g "ocdo U SØ ÞrCi (r.job poti)</li> <li>g "ocdo U SØ ÞrCi (r.job poti)</li> <li>g "opt, aret Timer, General PWM (r.</li> <li>g "opt, aret Timer, General PWM,</li> <li>g "opt, blue Timer, General PWM,</li> <li>g "fashö Fissh (r.flash-hp)</li> <li>SPP BLE Abstraction (m., ble, abstraction (m., b</li></ul>	gpt) _gpt) (r_gpt) (r_gpt) _spp)	g.joport I/O Port (r_joport)	g_pcdc0 USB PCDC (r_     g)     g_basic0 USB (r_usb_bi     g)     f     Add DMAC Driver for     Transmission     [Optional]	ssb_pcdc)		<ul> <li></li></ul>	g_gpt_green Timer, General PWM (r_gpt)     G	g_gpt_blue Timer, General PWM (r_gpt)     G
[mcu_fwupdate_ryz012_ra4m2_baremetal] F	SP Configuration	×						
Stacks Configuration								
-								Generate Project Content
New Thread Remove	HAL/Common	Stacks					🔊 New Stack > 🗢	Generate Project Content
Threads	HAL/Common	Stacks	A such blue Terre	A - firsto link		an bla she san)	🗞 New Stack >  🚊	Generate Project Content
Threads	HAL/Common S ,red Timer, al PWM (r_gpt)	Stacks	t) g_gpt_blue Timer, General PWM (r_gp	) g_flash0 Flash (r_flash_hp)	SPP BLE Abstraction (r	m_ble_abs_spp)	ᡚ New Stack > 🚊	Generate Project Content
Thread () Remove	HAL/Common S ,red Timer, al PWM (r_gpt)	Stacks	t) g_gpt_blue Timer, General PWM (r_gp	) g_flash0 Flash (r_flash_hp) (1)	<ul> <li>SPP BLE Abstraction (r</li> <li>Image: SPP BLE Abstraction (r</li> </ul>	m_ble_abs_spp)	∳] New Stack > ≗	Generate Project Content Extend Stack > Remove
New Thread Remove     New Thread Remove     Sector 100 Post (r_joport)     g_joport 100 Post (r_joport)     g_jopingt Timer, General PWM (r     g_ggt, red Timer, General PWM (r     ggt, red	HAL/Common S red Timer, al PWM (r_gpt)	Stacks g.gpt.green Timer, General PWM (r_gp 1	t) g.gpt, blue Timer, General PWM (r_gp	) (r_flash) Flash (r_flash_hp) (1)	SPP BLE Abstraction (r	m_ble_abs_spp)	New Stack >	Generate Project Content
Mew Thread         Remove           International State         International State           International State         International State <td< td=""><td>HAL/Common 1</td><td>Stacks</td><td>t) g_gpt_blue Timer, General PWM (r_gp</td><td>) ) (r_flash0Flash (r_flash_hp) ()</td><td>SPP BLE Abstraction (r</td><td>m_ble_abs_spp)</td><td>New Stack &gt;  New Stack &gt;  Pl (m_ble_abs_spp_transport)</td><td>CGenerate Project Content Extend Stack &gt; 🔊 Remove</td></td<>	HAL/Common 1	Stacks	t) g_gpt_blue Timer, General PWM (r_gp	) ) (r_flash0Flash (r_flash_hp) ()	SPP BLE Abstraction (r	m_ble_abs_spp)	New Stack >  New Stack >  Pl (m_ble_abs_spp_transport)	CGenerate Project Content Extend Stack > 🔊 Remove
Move Thread         Pernove           Image: Second Seco	HAL/Common S red Timer, al PWM (r_gpt)	Stacks	t) ( g.gpt, blue Timer, General PWM (r.gp	) (_fissh_tp) ①	SPP BLE Abstraction (r      D      FV2012 SPP Driver Layer	m_ble_abs_spp)	New Stack>	Generate Project Content Estend Sack > 1/2 Remove
New Thread         Pernove           Internet         Internet           Interne         Interne </td <td>HAL/Common 1 ,red Timer, al PWM (r_gpt)</td> <td>Stacks</td> <td>t) ( g.ggt, blue Timer, General PWM (r.gg</td> <td>) (_fissh_tp)</td> <td>SPP BLE Abstraction (r      D      FYZ012 SPP Driver Layer</td> <td>m_ble_abs_spp)  BLE SPP Transport on SI</td> <td>New Stack&gt;</td> <td>Generate Project Content Estand Stack &gt; 1() Bernove</td>	HAL/Common 1 ,red Timer, al PWM (r_gpt)	Stacks	t) ( g.ggt, blue Timer, General PWM (r.gg	) (_fissh_tp)	SPP BLE Abstraction (r      D      FYZ012 SPP Driver Layer	m_ble_abs_spp)  BLE SPP Transport on SI	New Stack>	Generate Project Content Estand Stack > 1() Bernove
Move Thread         Remove           Image: State of the state o	HAL/Common S red Timer, al PWM (r_gpt)	Stacks	R)	) (_flash_tp) (_flash_tp)	SPP BLE Abstraction (r	m_ble_abs_spp)  BLE SPP Transport on SI  g g_g_spl0 SPI (r_sci_spi)  ()	New Stack>	Generate Project Content Estand Sack > 1() Bernove
Thread New Thread () Bernove Wew Thread () Bernove # HAL/Common @ g.jopot 1/0 Port () joport) @ g.g.blinker Timer, General PWM () @ g.g.pt.green Timer, General PWM () @ g.g.pt.green Timer, General PWM () @ g.g.pt.glub Timer, G	HAL/Common 3	Stacks	t) g_gpt_blue Timer, General PWM (r_gp	) (r]ssh (r) (r]ssh (r) (r)	SPP BLE Abstraction (n	m_ble_abc_spp)	New Stack >	Generate Project Content Estend Sack > (i) Bernove

Figure 28. FSP Module Summary

# 4. Application Project Implementation

This section describes the application project implementation.

The firmware update is implemented in app\_main.c. The *app\_main()* includes BLE and system peripheral initialization and the implementation of the main loop.

When using QE for BLE tool, a minimal skeleton code of app\_main.c is automatically generated which is customized with FSP APIs and helper functions to add the desired Bluetooth module update functionality.

# 4.1 Entry Point

In hal\_entry.c the function hal\_entry() initializes the GPT module for indicator LEDs, starts the XMODEM console used by Tera Term, and calls *app\_main()* to perform the BLE module update as follows:

```
* The RA Configuration tool generates main() and uses it to generate threads if an RTOS is used. This function is
 * called by main() when no RTOS is used.
void hal_entry(void)
 {
     fsp_err_t err = FSP_SUCCESS;
     /* Initialize GPT module */
     err = common_init();
if (FSP_SUCCESS != err)
     {
          /* Turn ON RED LED to indicate fatal error */
         TURN RED ON
         APP_ERR_PRINT("\r\n ** Initialize GPT module : failed ** \r\n");
         APP_ERR_TRAP(err);
     }
     /* Start XMODEM Menu on TeraTerm */
    /* Start XMODEM Menu on TeraTerm */
APP_PRINT("\r\n ** Start RTT Viewer for Logging ** \r\n");
APP_PRINT("\r\n ** Start Iera Term for XMODEM File Transfer of RYZ012 FW Image to MCU ** \r\n");
APP_PRINT("\r\n ** Iera Term : New connection->Serial->COMx : USB Serial Device (COMx) ** \r\n");
APP_PRINT("\r\n ** Hit Enter Key (CR) in Iera Term Window ** \r\n");
     xmodem_console();
        Start the MCU FW update of RYZ012 : Use RTT Viewer for Status Logging */
     APP_PRINT("\r\n ** Start MCU FW Update of RYZ012 ** \r\n");
     app_main();
 }
```



# 4.2 Main Loop

The *app\_main()* includes the BLE and profile initialization, BLE module firmware update, and the main loop that processes the BLE events. See the source code below for *app\_main()* with the implementation details. The steps of execution are:

- 1. Initialize the BLE driver to start the communications with the RYZ012.
- 2. Get the current firmware version running on the RYZ012.
- 3. Check for a RYZ012 firmware image stored in EK-RA4M2 flash and display firmware version.
- 4. If a firmware image is found, determine the length of the file and update the RYZ012 with **R\_BLE\_VS\_UpdateModuleFirmware**(...).
- 5. Log the status of the firmware update to RTT Viewer
- 6. Close the BLE driver and call ble\_init() to initialize the BLE module and BLE Profile, then run the new RYZ012 firmware.
- 7. Get the RYZ012 firmware version and display to confirm updated image is running.
- 8. Enter the main while loop to operate the RYZ012 and process BLE events.

```
> void app_main(void)
 {
      fsp_err_t err;
      ble_status_t status;
      APP_PRINT("\r\nInitialize BLE Driver\r\n");
       /* Initialize BLE Driver */
      err = RM_BLE_ABS_Open(&g_ble_abs0_ctrl, &g_ble_abs0_cfg);
      if (FSP SUCCESS != err)
      {
           APP_ERR_PRINT("\r\n ** Initialize BLE Driver : failed ** \r\n");
APP_ERR_PRINT("\r\n ** Check that RYZ012 is connected to Eval Kit ** \r\n");
APP_ERR_PRINT("\r\n ** Check that RYZ012 is inserted with correct pin orientation ** \r\n");
           APP ERR TRAP(err);
      APP PRINT("\r\nBLE Driver Init : Success\r\n");
       /* Get RYZ012 FW Version that is running */
      get fw version();
      APP PRINT("\r\nCheck for RYZ012 FW Image in MCU Memory\r\n");
      bool b_image = false;
      b_image = is_flash_image_found( (uint8_t *) RYZ012_FWIMAGE_START_ADDRESS, (uint8_t *) RYZ012_FWIMAGE_END_ADDRESS );
      APP_PRINT("\r\nRYZ012 FW Image %s\r\n", (b_image ? "Found": "Not Found") );
APP_PRINT("\r\nStarted RYZ012 Module Firmware Update\r\n");
       /* Call Vendor Specific API : Run the RYZ012 Module Firmware Update : MCU based procedure. */
      if ( b_image )
Ð
      {
            /* A FW Image is present
             * Determine the FW Image Length as programmed by XMODEM.
             * NOTE: If not programmed by XMODEM, the get_fw_image_length(...) function
             * will need to be modified to determine the Image end of file.
             **/
           uint32_t image_len;
           image_len = get_fw_image_length( (uint8_t *) RYZ012_FWIMAGE_START_ADDRESS, (uint8_t *) RYZ012_FWIMAGE_END_ADDRESS );
APP_PRINT("\r\nRYZ012 FW Image size = %u bytes\r\n", image_len );
if ( ( 0 < image_len ) && (( RYZ012_FWIMAGE_END_ADDRESS - RYZ012_FWIMAGE_START_ADDRESS ) > image_len ))
            {
                status = R_BLE_VS_UpdateModuleFirmware( (uint8_t const *) RYZ012_FWIMAGE_START_ADDRESS, image_len);
           }
           else
           {
                 // Image size not in range
                status = BLE ERR INVALID DATA;
           }
      }
      else
      {
           status = BLE_ERR_NOT_FOUND; // RYZ012 flash image not found @ MCU RYZ012_FWIMAGE_START_ADDRESS
      }
```



```
if (BLE_SUCCESS != status)
  {
      APP_ERR_PRINT("\r\n ** RYZ012 Module Firmware Update : Failed ** \r\n");
      switch ( status )
      {
          case BLE ERR INVALID DATA :
              APP_ERR_PRINT("\r\n ** FW Image Size Invalid** \r\n");
              break;
          case BLE ERR NOT FOUND :
              APP_ERR_PRINT("\r\n ** No FW Image Found in Memory @ RYZ012_FWIMAGE_START_ADDRESS** \r\n");
              break;
          default :
              break;
      APP_ERR_TRAP(status);
}
  APP_PRINT("\r\nRYZ012 Module Firmware Update : Success\r\n");
  /* Close the BLE driver so that it can be re-initialized and run updated firmware. */
  err = RM_BLE_ABS_Close(&g_ble_abs0_ctrl);
  if (FSP_SUCCESS != err)
  {
      APP_ERR_PRINT("\r\n ** Close BLE Driver : Failed ** \r\n");
      APP_ERR_TRAP(err);
  APP_PRINT("\r\nBLE Driver Close : Success\r\n");
  APP_PRINT("\r\nInit BLE Driver and Profiles\r\n");
  /* Initialize BLE and profiles */
  status = ble init();
  if (FSP_SUCCESS != status)
  {
      APP_ERR_PRINT("\r\n ** BLE Driver and Profiles Init : Failed ** \r\n");
      APP_ERR_TRAP(status);
  APP_PRINT("\r\nBLE Driver and Profile Init : Success\r\n");
  /* Get RYZ012 FW Version that is running */
  get_fw_version();
  APP_PRINT("\r\nRunning RYZ012 FW\r\n");
  /* main loop */
  while (1)
  {
      /* Process BLE Event */
      R_BLE_Execute();
  3
   /* Terminate BLE */
  RM_BLE_ABS_Close(&g_ble_abs0_ctrl);
```

}



# 4.3 BLE Initialization Process

QE for BLE tool was used to create a basic BLE Profile with "RYZ012" set as the advertise name. From the tool, the source code of the *ble\_init()* function is automatically generated and placed at the top of the app\_main() functions. See below where the placement of **ble\_init()** call must be relocated in the sequence after the firmware update process steps are completed.

```
APP PRINT("\r\nRYZ012 Module Firmware Update : Success\r\n");
      /* Close the BLE driver so that it can be re-initialized and run updated firmware. */
      err = RM_BLE_ABS_Close(&g_ble_abs0_ctrl);
      if (FSP_SUCCESS != err)
Θ
      {
          APP ERR PRINT("\r\n ** Close BLE Driver : Failed ** \r\n");
          APP_ERR_TRAP(err);
     APP_PRINT("\r\nBLE Driver Close : Success\r\n");
APP_PRINT("\r\nInit BLE Driver and Profiles\r\n");
      /* Initialize BLE and profiles */
      status = ble init();
      if (FSP_SUCCESS != status)
      Ł
          APP_ERR_PRINT("\r\n ** BLE Driver and Profiles Init : Failed ** \r\n");
          APP ERR TRAP(status);
      3
      APP_PRINT("\r\nBLE Driver and Profile Init : Success\r\n");
      /* Get RYZ012 FW Version that is running */
      get_fw_version();
      APP_PRINT("\r\nRunning RYZ012 FW\r\n");
      /* main loop */
     while (1)
Θ
      {
          /* Process BLE Event */
          R_BLE_Execute();
      }
      /* Terminate BLE */
      RM_BLE_ABS_Close(&g_ble_abs0_ctrl);
 }
```

See the details of the ble\_init() function implementation below that initializes the BLE module, registers the callback functions for BLE, registers the GATT database, and any other additional services added with the QE for BLE tool during BLE Profile generation.



```
* Function Name: ble_init.

Ble_status_t ble_init(void)
Ble_status_t ble_status_t ble_status_t ble_status_t
Ble_status_t ble_status_t ble_status_t
Ble_status_t ble_status_t
Ble_status_t ble_status_t
Ble_status_t ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble_status_t
Ble
     {
                   ble_status_t status;
                   fsp_err_t err;
                    /* Initialize BLE */
                   err = RM_BLE_ABS_Open(&g_ble_abs0_ctrl, &g_ble_abs0_cfg);
                   if (FSP_SUCCESS != err)
 Θ
                    {
                                 return err;
                   }
                   /* Initialize GATT Database */
                   status = R_BLE_GATTS_SetDbInst(&g_gatt_db_table);
                   if (BLE_SUCCESS != status)
                    {
                                 return BLE_ERR_INVALID_OPERATION;
                   }
                   /* Initialize GATT server */
                   status = R BLE SERVS Init();
                   if (BLE_SUCCESS != status)
 {
                                return BLE ERR INVALID OPERATION;
                   }
                    /*Initialize GATT client */
                   status = R_BLE_SERVC_Init();
                   if (BLE_SUCCESS != status)
 {
                                 return BLE_ERR_INVALID_OPERATION;
                   }
                    /* Set Prepare Write Queue */
                   R_BLE_GATTS_SetPrepareQueue(gs_queue, BLE_GATTS_QUEUE_NUM);
                    /* Initialize GATT Service server API */
                   status = R_BLE_GATS_Init(gats_cb);
                   if (BLE_SUCCESS != status)
                   {
                                 return BLE ERR INVALID OPERATION;
                   }
                    return status;
     }
```

# 4.4 Register BLE Callback Functions

Registration of the callback functions are performed in **ble\_init()** which are required to execute processing according to events from each layer such as GAP, GATT Server, GATT Client, and Profile Server API in the Bluetooth LE protocol stack. See **Bluetooth Low Energy Sample Application (R01AN6116EJ)** for more details of the callback registration process for RYZ012 and RA MCU and refer to **RA Flexible Software Package User's Manual (R11UM0155)** for more information on the type of callback events to handle.



# 4.5 Firmware Image Helper Functions

The following source code provides helper functions **is\_flash\_image\_found(...)** and **get\_fw\_image\_length(...)** which determine whether there is an RYZ012 firmware image in MCU flash and the length of the firmware image file in bytes.

```
/* Look for an RYZ012 FW Image in MCU Code Flash */

bool is_flash_image_found( uint8_t * p_FlashAddressStart, uint8_t * p_FlashAddressEnd )

 {
     bool image found;
     uint8_t * p_src;
     image_found = false;
     p_src = p_FlashAddressStart;
Θ
     do
      {
          if ( 0xFF != *p_src )
{
              image found = true;
          }
          p src++;
     } while ( (false == image found) && ((uint8 t *)p FlashAddressEnd > p src) );
     return image found;
 }
 /* Determine the RYZ012 Firmware Image Length in MCU Code Flash */
wint32_t get_fw_image_length( uint8_t * p_FlashAddressStart, uint8_t * p_FlashAddressEnd )
 {
     bool end = false;
     uint32_t * p_src;
     uint32 t len = 0;
     p_src = (uint32_t *) p_FlashAddressStart;
Θ
     do
     {
Θ
          if ( 0x1A1A1A1A == *p src )
          {
              /* four consecutive 0x1A bytes found, as written by XMODEM transfer
* this is the end of the image.
              * NOTE: If FW image is transferred using a method
               * other than XMODEM, end of file determination will be different
              * and require modification here.
               * XMODEM transfer writes "0x1A1A1A1A" after the end of the file in Code flash
               * and is not part of the FW binary file itself.
              */
              end = true;
              if (len >= 12) len -= 12; /* compensate for the 12 bytes counted beyond end of file */
          }
          else
\Theta
          {
              p_src += 4; /* next address to check, 0x60000, 0x60010, 0x60020 */
                           /* count the 16 bytes found at the address */
              len += 16;
      } while ( ( false == end ) && ( (uint32_t *)p_FlashAddressEnd > p_src ));
      return len;
  }
```



# 4.6 Get Firmware Version Function

See the source code below to for the details of utility function **get\_fw\_version(void)** which is used to read the firmware version that is currently running on the RYZ012 module. The function calls the FSP provided function **R\_BLE\_VS\_GetFirmwareVersion()** and uses SPI to get the version response back from the module immediately (synchronously) and parses the result for display on RTT Viewer.

```
/* Get the Firmware Version running on RYZ012 Module */

    bool get_fw_version(void)

 {
      static r_ble_spp_payload_t payload_data;
r_ble_spp_cmd_rsp_t ret_val = R_BLE_SPP_SUCCESS;
      ble_status_t status;
      #define MIN_FW_VERSION_BYTES 3 /* Payload must have 3 or more bytes */
      status = R_BLE_VS_GetFirmwareVersion();
      if (BLE_SUCCESS != status)
Θ
      {
          APP ERR PRINT("\r\n ** Get RYZ012 FW Version : Failed ** \r\n");
          return false;
      }
      /* To get the FW version data immediately use R_BLE_SPP_SPI_Read(&payload_data)
Θ
        * otherwise, use Process BLE Event by calling R_BLE_Execute() and waiting
      * for callback event BLE_VS_EVENT_GET_FW_VERSION_COMP to be received in vs_cb(...)
       * vs_cb(uint16_t type, ble_status_t result, st_ble_vs_evt_data_t *p_data)
      R_BSP_SoftwareDelay(10, BSP_DELAY_UNITS_MILLISECONDS);
      ret_val = R_BLE_SPP_SPI_Read(&payload_data);
      if ( R_BLE_SPP_SUCCESS == ret_val )
Θ
          if ( ( BLE_VS_EVENT_GET_FW_VERSION_COMP == payload_data.event_id ) && ( MIN_FW_VERSION_BYTES <= payload_data.out_len ) )
          {
                 pavload contains fw version data */
              g_ble_version_data.major = payload_data.out_data[1];
              g_ble_version_data.minor = payload_data.out_data[2];
              APP_PRINT("\r\nRYZ012 FW Version v%d.%d \r\n", g_ble_version_data.major, g_ble_version_data.minor);
              return true;
          }
          APP ERR PRINT("\r\n ** Get RYZ012 FW Version : Failed ** \r\n");
          return false:
     }
      APP_PRINT("RYZ012 FW Version data read Error: %x\r\n", ret_val);
      return false;
  }
```



# 4.7 Firmware Update Function

The FSP provided function **R\_BLE\_VS\_UpdateModuleFirmware(...)** implements the execution steps of the firmware update to the RYZ012. The steps are covered in section 1.2, RYZ012 Firmware Update Process.

```
⊖ble status t R BLE VS UpdateModuleFirmware (uint8 t const * const p firmware image, uint32 t firmware image_size)
 {
      /* Start the firmware update procedure and wait for a response. */
     BLE_SPP_COMMAND(R_BLE_VS_StartFirmwareUpdate(),
Θ
                      BLE VS EVENT START FW UPDATE COMP
                      BLE_MODULE_START_FW_UPLOAD_TIMEOUT);
      /* Calculate the total number of frames to send. */
     uint16_t total_frames =
          (uint16_t) ((firmware_image_size + BLE_MODULE_SEND_FIRMWARE_DATA_SIZE - 1) /
                      BLE MODULE_SEND_FIRMWARE_DATA_SIZE);
     uint16_t last_index = total_frames - 1;
     for (uint16_t i = 0; i < last_index; i++)</pre>
     Ł
          /* Send the next 'Send Firmware Data' command and wait for a response. */
Θ
          BLE_SPP_COMMAND(R_BLE_VS_SendFirmwareData(i, BLE_MODULE_SEND_FIRMWARE_DATA_SIZE,
                                                    p_firmware_image + (i * BLE_MODULE_SEND_FIRMWARE_DATA_SIZE)),
                          BLE VS EVENT SEND FW DATA COMP,
                          BLE_MODULE_RESET_TIMEOUT);
     }
     /* Calculate the size of the last data frame. */
     uint16_t last_frame_size = firmware_image_size % BLE_MODULE_SEND_FIRMWARE_DATA_SIZE;
     if (0 == last_frame_size)
     {
          last frame size = BLE MODULE SEND FIRMWARE DATA SIZE;
     }
     /* Send the last 'Send Firmware Data' command and wait for a response. */
     BLE_SPP_COMMAND(R_BLE_VS_SendFirmwareData(last_index, last_frame_size,
                                                p_firmware_image + (last_index * BLE_MODULE_SEND_FIRMWARE_DATA_SIZE)),
                      BLE_VS_EVENT_SEND_FW_DATA_COMP,
                      BLE MODULE RESET TIMEOUT);
      /* Send the 'End Firmware Update' command and wait for a response. */
     BLE_SPP_COMMAND(R_BLE_VS_EndFirmwareUpdate(last_index),
                      BLE_VS_EVENT_END_FW_UPDATE_COMP
                     BLE_MODULE_END_FW_UPLOAD_TIMEOUT);
     /* Reboot the module. */
     return R BLE VS RestartModule();
 }
```

The steps of execution are:

- 1. Send the SPP command to Start Firmware Update with R\_BLE\_VS\_StartFirmwareUpdate() and wait for a response.
- 2. Calculate the total number of frames to transfer to RYZ012 based on image file size.
- 3. Send the data frames of image to RYZ012 with SPP command R\_BLE\_VS\_SendFirmwareData() and wait for a response. Repeat sending frames until last data frame remains.
- 4. Send the last data frame of image with R BLE VS SendFirmwareData() and wait for a response.
- 5. Send the End Firmware Update to RYZ012 with SPP command R\_BLE\_VS\_EndFirmwareUpdate() and wait for a response.
- 6. Reboot the RYZ012 module with R\_BLE\_VS\_RestartModule().

# 5. How to Make and Configure a New Project

This section describes the configuration to create an RA MCU based firmware update application for the RYZ012. The application that is generated will include the system peripheral initialization, BLE Profile, and BLE API framework that will then need to be customized to include the RYZ012 module firmware update sequence. The update sequence is implemented by calling the FSP provided update and utility functions detailed in section 4, Application Project Implementation.



# 5.1 Create a New Project

Please see the **Renesas Flexible Software Package (FSP) User's Manual (<u>R11UM0155EU</u>) for instructions on how to create a new project in e<sup>2</sup> studio for a BareMetal environment, choose <b>BareMetal – Minimal**. Follow the instructions to select the FSP Version, Board, Device, and Toolchains. Continue through all prompts until the project is created. Next the BSP Heap and Stack must be configured.

#### 5.2 BSP Heap and Stack configuration

Set heap and stack configuration as follows on the FSP configuration **BSP** tab. If the properties tab is not visible, choose **Window > Show View > Properties** on the  $e^2$  studio menu bar.

- [RA Common] > [Main stack size (bytes)] : 0x4000
- [RA Common] > [Heap size (bytes)] : 0x1000

Summary F	BSP Clocks Pins Interrupts Event Links Stacks Components	
R Problem	ns 😑 Console 🔲 Properties 🗙 ௸ Smart Browser 🏾 🙀 Stack Analysis 🕮 Smart Manual 🔗 Searc	:h 🔋 Memory 🔫 Progress
EK-RA4N	12	
Settings	Property	Value
	✓ RA Common	
	Main stack size (bytes)	0x4000
	Heap size (bytes)	0x1000
	MCU Vcc (mV)	3300

Figure 29. BSP Configuration

# 5.3 Add the I/O Port Stack

Stacks	Configuration		
Threads	🐑 New Thread 🙀 Remove 📄	g_ioport I/O Port (r_ioport) Stacks	
	IAU/Common P g_ioport//O Port (r_ioport) g_pcdc0 USB PCDC (r_usb_pcdc) g_blinker Timer, General PWM (r_gpt) g_gpt_gream Timer, General PWM (r_gpt) g_gpt_blue Timer, General PWM (r_gpt) g_gftsh0 Flach (flash, hp) \$ SPP BLE Abstraction (rm_ble_abs_spp)	<pre># g ioport I/O Port   (r.ioport)   </pre>	
Objects	🙆 New Object > 📓 Remove		
Summany	PSD Clocks Ding Interrupts Event Links	taske Componente	
Darkie	no. El Cassala El Desastion X 🚇 Su	naters components	lanual 🖉 Canada 🖪 Marajana 📼 Danamara
Topie 1	ms 📮 Console 📋 Properties 🗶 🛶 Sm	fart Browser 📲 Stack Analysis 🤬 Smart N	ianual 🔗 Search 📗 Memory 🛶 Progress
g_iopor	t I/O Port (r_ioport)		
g_ioport	t I/O Port (r_ioport) Property		Value
g_ioport	t I/O Port (r_ioport) Property Vodule g_ioport I/O Port (r_ioport)		Value
g_ioport Settings API Info	t I/O Port (r_ioport) Property v Module g_ioport I/O Port (r_ioport) Name		Value g_ioport
g_ioport Settings API Info	t I/O Port (r_ioport) Property Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source		Value g_ioport Disabled
g_ioport Settings API Info	t I/O Port (r_ioport)  Property  Module g_ioport I/O Port (r_ioport)  Name  Port 1 ELC Trigger Source  Port 2 ELC Trigger Source		Value g_ioport Disabled Disabled
g_ioport	t L/O Port (r_ioport) Property ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source		Value g_ioport Disabled Disabled
g_ioport Settings API Info	t I/O Port (r_ioport) Property ➤ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 3 ELC Trigger Source		Value g_ioport Disabled Disabled Disabled
g_ioport Settings API Info	t L/O Port (r_joport) Property  Mame Port 1ELC Trigger Source Port 2ELC Trigger Source Port 3ELC Trigger Source Port 3ELC Trigger Source Port 4ELC Trigger Source Port 4ELC Trigger Source Port 4ELC Trigger Source Port 8ELC Trigger Source Port 8EL		Value g_ioport Disabled Disabled Disabled Disabled
g_ioport Settings API Info	t I/O Port (r_ioport) Property ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 8 ELC Trigger Source Port 8 ELC Trigger Source Port 8 ELC Trigger Source		Value g_ioport Disabled Disabled Disabled Disabled Disabled
g_ioport Settings API Info	t I/O Port (r_joport)  Property  Module g_joport I/O Port (r_joport)  Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 8 ELC Trigger Source Port 8 ELC Trigger Source Port 6 ELC Trigger Source Port C ELC Trigger Source Port B ELC Trigger Source Port B ELC Trigger Source Port Source Po		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled
g_ioport Settings API Info	t L/O Port (r_joport)  Property  ✓ Module g_ioport I/O Port (r_joport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 8 ELC 7 FIGHT 8 ELC		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled
g_ioport Settings API Info	t I/O Port (r_joport)  Property  ✓ Module g_joport I/O Port (r_joport)  Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 8 ELC Trigger Source Port 8 ELC Trigger Source Port C ELC Trigger Source Port C ELC Trigger Source Port C ELC Trigger Source Port E ELC Trigger Source Port C ELC Trigger		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled d Disabled d Disabled
g_ioport Settings API Info	t L/O Port (r_ioport)  Property  ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port 6 ELC Trigger Source Port C ELC Trigger Source Port 6 ELC Trigger		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled g_bsp_pin_cfg
g_ioport Settings API Info	t I/O Port (r_ioport)  Property  ✓ Module g_ioport I/O Port (r_ioport)  Name Port 1 EL Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 6 ELC Trigger Source Port 5 ELC Trigger Source Port 6 ELC Trigger Source Pin C Configuration Name ▼ Pins TK		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled g_sbaled g_sbaled g_sbaled g_sbaled g_sbaled
g_ioport Settings API Info	t L/O Port (r_ioport)  Property  ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port C ELC Trigger Source Port T ELC Trigger Source The Configuration Name ✓ Pins TCK TDI		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Pisabled Pisabled Pisabled Pisabled Pisabled Pisabled Disabled
g_ioport Settings API Info	t L/O Port (r_joport)  Property  ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port 6 ELC Trigger Source Tot C Tot Tot Tot Tot Tot		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled P300 P110 P100
g_ioport Settings API Info	t I/O Port (r_joport)  Property  ✓ Module g_joport I/O Port (r_joport)  Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 3 ELC Trigger Source Port 3 ELC Trigger Source Port 8 ELC Trigger Source Port 8 ELC Trigger Source Port D ELC Trigger Source Port D ELC Trigger Source Port SUCC Trigger Source Pin SUCC Trigger Source		Value g_ioport Disabled
g_ioport Settings API Info	t L/O Port (r_ioport)  Property  ✓ Module g_ioport //O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port 6 ELC Trigger Source Port 7 ELC Trigger Source Port 6 ELC Trigger Source The Source The Source The Source The Source The Source Port 7 ELC Trigger Source Port 6 ELC Trigger Source Port 7 ELC Trigg		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled g_bsp_pin_cfg P300 P110 P109 P109 P109 P108 c consciliables
g_ioport Settings API Info	t L/O Port (r_ioport)  Property  Module g.joport I/O Port (r_ioport)  Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 6 ELC Trigger Source Port 5 ELC Trigger Source Port 5 ELC Trigger Source Port 5 ELC Trigger Source Port 6 ELC Trigger Source Port 6 ELC Trigger Source Port 6 ELC Trigger Source Pins TCK TDI TDO TNS SWCLK SSWIPO		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Sisabled Disabled Sisabled
g_ioport Settings API Info	LI/O Port (r_joport)      Property      Module g_ioport I/O Port (r_joport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port 5 ELC Trigger Source Trigger Source Port 5 ELC Trigger S		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled P300 P110 P109 P108 < unavailable> < unavailable>
g_ioport Settings API Info	t L/O Port (r_joport)  Property  ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 8 ELC Trigger Source Tot 8 ELC Trigger Source Tot 8 ELC Trigger Source Tot 8 ELC Trigger Source Port 8 ELC Trigger Source Tot 8 ELC Trigger Source Tot 8 ELC Trigger Source Port 8 ELC Trigger Source Tot 8 ELC Trigger Source Port 8 ELC 8		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled g_bsp_pin_cfg P300 P100 P109 P108 < unavailable> < unavailable> < unavailable>
g_ioport	t L/O Port (r_ioport)  Property  ✓ Module g_ioport I/O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port 5 ELC Trigger 5 ELC Trigge		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Q_bsp_pin_cfg P300 P110 P109 P108 < unavailable> < unavailable> P212 P222
g_ioport Settings API Info	t L/O Port (r_ioport)  Property  ✓ Module g_ioport //O Port (r_ioport) Name Port 1 ELC Trigger Source Port 2 ELC Trigger Source Port 2 ELC Trigger Source Port 4 ELC Trigger Source Port 4 ELC Trigger Source Port 6 ELC Trigger Source Port 6 ELC Trigger Source Port 6 ELC Trigger Source Port 7 ELC Trigger Source Port 7 ELC Trigger Source Port 6 ELC Trigger Source Port 7 ELC Trigger Source Fort 7 ELC Trigger Source Port 7 ELC Trigger 7 ELC Trigger Source Port 7 ELC Trigger 7		Value g_ioport Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled Disabled P100 P100 P109 P122 P213 P213

Figure 30. Adding I/O Stack



# 5.4 Add and Configure the SPP BLE Module

This section describes how to add and configure SPP BLE Abstraction Driver for communicating with the RYZ012 module in the application. Open **configuration.xml** in the project and add / configure **SPP BLE Abstraction Driver** on FSP configuration **Stacks** tab.

The procedure to add the SPP BLE Abstraction Driver is different for BareMetal and FreeRTOS environments. This document describes the procedure for BareMetal environment.

#### Add RYZ012 module in BareMetal environment

Click **New Stack** and add **Networking->SPP BLE Abstraction (rm\_ble\_abs\_spp)** to **HAL/Common**. If the stack cannot be found use the **Search** box in e<sup>2</sup> studio since stack categories and organization may change in future FSP releases.

, , , , , , , , , , , , , , , , , , ,			
hreads 🔹 🐑 New Thread 😰 Remove	SPP BLE Abstraction (rm_b	ole_abs_spp) Stacks	
<ul> <li>✓ ALL/Common</li> <li> <i>gioport I/O Port (r_ioport)</i> </li> <li> <i>gpcdc0 USB PCDC (r_usb_pcdc)</i> </li> <li> <i>gblinker Timer, General PWM (r_gpt)</i> </li> <li> <i>ggpt_red Timer, General PWM (r_gpt)</i> </li> </ul>	<ul> <li>SPP BLE Abstraction</li> <li>Image: SPP BLE Abstraction</li> </ul>	(rm_ble_abs_spp)	
<ul> <li>g_gpt_green Timer, General PWM (r_g</li> <li>g_gpt_blue Timer, General PWM (r_gp</li> <li>g_flash0 Flash (r_flash_hp)</li> <li>SPP BLE Abstraction (rm_ble_abs_spp)</li> </ul>	t) PRYZ012 SPP Driver Layer	BLE SPP Transport on SPI (rm_ble_abs_spp	_transport)
		g_spi0 SPI (r_sci_spi)	g_external_irq0 External IRQ (r_icu)
Objects 🔬 New Object > 🔬 Remo	ve	(1)	(1)
		<ul> <li>         ⊕ g_transfer0 Transfer         (r_dtc) SCI4 TXI         (Transmit data         (1) empty)         </li> <li>         ⊕ g_transfer1         (r_dtc) SCI4         (Received d         (1)         </li> </ul>	Transfer RXI ata full)
	ta Barta Canada		
Immary BSP Clocks Pins Interrupts Event Lin Problems	hks Stacks Components	is 👊 Smart Manual 🛷 Search 🔋 Memory 🗉	Progress
ummary BSP Clocks Pins Interrupts Event Lir Problems Console Properties X PP BLE Abstraction (rm_ble_abs_spp) ettings Property	hks Stacks Components	is 🔑 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value
ummary BSP Clocks Pins Interrupts Event Lir Problems Console Properties X ( PP BLE Abstraction (rm_ble_abs_spp) Property Reset Port	hks Stacks Components	is 🔑 Smart Manual 🛷 Search 🔋 Memory 🖬	Progress Value 04
ummary BSP Clocks Pins Interrupts Event Lir Problems Console Properties X PP BLE Abstraction (rm_ble_abs_spp) Property Reset Port Reset Pin	hks Stacks Components	is 🔑 Smart Manual 🛷 Search 🔋 Memory 🗉	Progress Value 04 03
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X       X         PP BLE Abstraction (rm_ble_abs_spp)         ettings       Property         Reset Port         Reset Pin       UART/SPI Select Port (PB5)	hks Stacks Components	is 👊 Smart Manual 🛷 Search 🔋 Memory 🗉	Progress Value 04 03 04
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       Image: Console       Properties       Image: Console         PP BLE Abstraction (rm_ble_abs_spp)       Property       Reset Port         Reset Port       Reset Pin       UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)       UART/SPI Select Pin (PB5)	hks Stacks Components	is 😳 Smart Manual 🔗 Search 🔋 Memory 🗉	Value 04 03 04 02
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X         PP BLE Abstraction (rm_ble_abs_spp)         Settings       Property         Reset Port         Reset Pin         UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)         SPI Software SSL Port	hks Stacks Components	is 😳 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value 04 03 04 02 04
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X         PP BLE Abstraction (rm_ble_abs_spp)         Settings       Property         Reset Port         Reset Pin       UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)         SPI Software SSL Pin	nks Stacks Components	is 😳 Smart Manual 🔗 Search 📋 Memory 🖬	Value 04 03 04 02 04 01
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X         Settings       Property       X       X         Property       Reset Port       X       X         UART/SPI Select Port (PB5)       UART/SPI Select Port (PB5)       X       X         SPI Software SSL Port       SPI Software SSL Pin       Transmit Power Level (in dBm)	hks Stacks Components	is 🔑 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value 04 03 04 02 04 01 4.57
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X       X         SPP BLE Abstraction (rm_ble_abs_spp)         Property       Reset Port         Reset Pin       UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)       SPI Software SSL Port         SPI Software SSL Pin       Transmit Power Level (in dBm)         V Module SPP BLE Abstraction (rm_	hks Stacks Components	is 😳 Smart Manual 🔗 Search 🔋 Memory 🗉	Value 04 03 04 02 04 01 4.57
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X       X         SPP BLE Abstraction (rm_ble_abs_spp)         Property       Reset Port         Reset Pin       UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)       SPI Software SSL Port         SPI Software SSL Pin       Transmit Power Level (in dBm)         V Module SPP BLE Abstraction (rm_       V General	hks Stacks Components Smart Browser Stack Analys ble_abs_spp)	is 😳 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value 04 03 04 02 04 01 4.57
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X       X         SPP BLE Abstraction (rm_ble_abs_spp)         Property       Reset Port         Reset Pin       UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)       SPI Software SSL Port         SPI Software SSL Pin       Transmit Power Level (in dBm)         V Module SPP BLE Abstraction (rm_       Y General         Name       Name	hks Stacks Components	is 🤑 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value 04 03 04 02 04 01 4.57 g_ble_abs0
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X       X         SPP BLE Abstraction (rm_ble_abs_spp)         Property       Reset Port         Reset Pin       UART/SPI Select Port (PB5)         UART/SPI Select Pin (PB5)       SPI Software SSL Port         SPI Software SSL Pin       Transmit Power Level (in dBm)         V Module SPP BLE Abstraction (rm_       Y General         Name       GAP callback	hks Stacks Components	is 🔑 Smart Manual 🔗 Search 🔋 Memory 🖬	Value 04 03 04 02 04 02 04 01 4.57 g_ble_abs0 gap_cb
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       X         PP BLE Abstraction (rm_ble_abs_spp)         Property       Reset Port         Reset Port       Reset Port         QUART/SPI Select Port (PB5)       UART/SPI Select Pin (PB5)         SPI Software SSL Port       SPI Software SSL Port         SPI Software SSL Pin       Transmit Power Level (in dBm)         V Module SPP BLE Abstraction (rm_       Y General         Name       GAP callback         Vendor specific callback       Yendor specific callback	hks Stacks Components	is 🔑 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value 04 03 04 02 04 01 4.57 g_ble_abs0 gap_cb vs_cb
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       Improverties       Improverties         Settings       Property       Reset Port       Improverties       Improverties         VART/SPI Select Port (PB5)       UART/SPI Select Port (PB5)       UART/SPI Select Pin (PB5)       SPI Software SSL Pin         Transmit Power Level (in dBm) <ul> <li>Module SPP BLE Abstraction (rm_</li> <li>General</li> <li>Name</li> <li>GAP callback</li> <li>Vendor specific callback</li> <li>GATT server callback parameter</li> </ul>	hks Stacks Components	is 😳 Smart Manual 🔗 Search 🔋 Memory 🖬	Value 04 03 04 02 04 01 4.57 g_ble_abs0 gap_cb vs_cb gs_abs_gatts_cb_param
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       Improventies       Improventies         Settings       Property       Reset Port       Improventies       Improventies         Settings       Property       Reset Port       Improventies       Improve	hks Stacks Components Smart Browser E Stack Analys ble_abs_spp) eter ar	is 😱 Smart Manual 🔗 Search 🔋 Memory 🖬	Progress Value 04 03 04 02 04 01 4.57 g_ble_abs0 gap_cb vs_cb gs_abs_gatts_cb_param 2
ummary       BSP       Clocks       Pins       Interrupts       Event Lir         Problems       Console       Properties       Improperties       Improperties       Improperties         Settings       Property       Reset Port       Improperties       Improperties       Improperties         Settings       Property       Reset Port       Improperties       Improperties       Improperties         VART/SPI Select Port       Reset Port       Reset Port       Improperties       Improperties         VART/SPI Select Port       Reset Port       SPI Software SSL Port       SPI Software SSL Port       SPI Software SSL Pin         Transmit Power Level (in dBm)       Module SPP BLE Abstraction (rm_       General       Name         GAP callback       Vendor specific callback       GATT server callback paramu         GATT server callback numbe       GATT client callback paramu	eter eter	is 😳 Smart Manual 🔗 Search 🔋 Memory 🗉	Progress           Value           04           03           04           02           04           01           4.57           g_ble_abs0           gap_cb           vs_cb           gs_abs_gatts_cb_param           2           gs_abs_gattc_cb_param

#### Figure 31. SPP BLE ABS Module

#### 5.4.1 SPP BLE Abstraction Driver Configuration

This section describes the SPP BLE Abstraction Driver configuration options and related modules. SPP BLE Abstraction Driver includes the following configuration for the **EK-RA4M2** and **SPI comms** interface to



**PMOD1 with RYZ012**. Users can modify these configurations according to their own specific hardware requirements. See Figure 31, SPP BLE ABS Module for more details.

Configuration options	Comment
Reset port	Specify port number of Reset Pin
EK-RA4M2 case: 04	
Reset Pin	Specify pin number of Reset Pin
EK-RA4M2 case: 03	
UART / SPI Select Port (PB5)	Specify port number of PMOD1 Mode Select Pin.
EK-RA4M2 case: 04	
UART / SPI Select Pin (PB5)	Specify pin number of PMOD1 Mode Select Pin.
EK-RA4M2 case: 02	Configure mode as SPI
SPI Software SSL Port	Specify port number of SPI Slave Select (SS)
EK-RA4M2 case: 04	
SPI Software SSL Pin	Specify pin number of SPI Slave Select (SS)
EK-RA4M2 case: 01	
Transmit Power Level (in dBm)	Specify required transmit power level.
EK-RA4M2 case: 4.57dBm	
Gap callback	Do NOT change.
Default: gap_cb	
Vendor specific callback	Do NOT change.
Default: vs_cb	
GATT server callback parameter	Do NOT change.
Default: gs_abs_gatts_cb_param	
GATT server callback number	Do NOT change.
Default: 2	
GATT client callback parameter	Do NOT change.
Default: gs_abs_gattc_cb_param	
GATT client callback number	Do NOT change.
Default: 2	



# 5.4.2 Configure Peripherals for SPP BLE Abstraction Driver

The SPP BLE Abstraction Driver uses the SPI Driver on r\_sci\_spi to communicate with RYZ012 module attached to the RA4M2 with PMOD1 connector (J26). This section describes how to configure the SPI driver. See additional information in section 6, Next Steps for using the SPP BLE Driver with other communication interfaces such as UART.

1. Click g\_spi0 SPI Driver on r\_sci\_spi.



Figure 32. SPI Module Properties

See Figure 32, SPI Module Properties. For the SPI configuration, it is necessary to specify which SCI channel to use. Match the SCI channel used and configure it as the SPI interface for PMOD1 in the **Pins Tab** see Figure 34, Pin Configuration > SCI to configure SPI Mode and Pins. Here SCI Channel 4 is being used on the RA4M2 in SPI Mode with the Pins assigned to PMOD1 (conn J26). See the Schematic for EK-RA4M2 for more details on the PMOD1 signals and pin assignments.



# RYZ012 and RA MCU

				▲	
			<ul> <li>g_spi0 SPI (r_sci_spi)</li> <li>1</li> </ul>		<ul> <li>g_external_irq0</li> <li>External IRQ (r_icu)</li> <li>i</li> </ul>
Objects	🐑 New Object > 🕷 Remove		<ul> <li>g_transfer0 Transfer (r_dtc) SCI4 TXI (Transmit data</li> <li>empty)</li> </ul>	g_transfer1 Transfer     (r_dtc) SCI4 RXI     (Received data full)     (1)	
mmary BS	SP Clocks Pins Interrupts Event Links Sta	cks Components			
Problems	Console Properties X 🛞 Smar	rt Browser 🖪 Stack Analysis	🕮 Smart Manual 🛷 Searc	h 👖 Memory 💷 Progress	
		a state in all size	- y onder mandar y ocar		
spio sri	(1_301_341)				
ettings	Property			Value	
DUL C	Clash Dhasa				
Pl into	Clock Phase			🔒 Data sampling on odd edg	e, data variation on even edge
	Clock Polarity			Data sampling on odd edg Low when idle	e, data variation on even edge
	Clock Phase Clock Polarity Mode Fault Error			<ul> <li>Data sampling on odd edg</li> <li>Low when idle</li> <li>Disable</li> </ul>	e, data variation on even edge
	Clock Phase Clock Polarity Mode Fault Error Bit Order			<ul> <li>Data sampling on odd edg</li> <li>Low when idle</li> <li>Disable</li> <li>MSB First</li> </ul>	e, data variation on even edge
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback			Data sampling on odd edg     Low when idle Disable     MSB First     rm_ble_spp_host_spi_callb	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority			Data sampling on odd edg     Low when idle     Disable     MSB First     m_ble_spp_host_spi_callb     Priority 12	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority			Data sampling on odd edg     Low when idle     Disable     MSB First     m_ble_spp_host_spi_callb     Priority 12     Priority 12	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Transmit End Interrupt Priority			Data sampling on odd edg     Low when idle     Disable     MSB First     m_ble_spp_host_spi_callb     Priority 12     Priority 12     Priority 12	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Transmit End Interrupt Priority Error Interrupt Priority			Data sampling on odd edg     Low when idle Disable     MSB First     m_ble_spp_host_spi_callb Priority 12 Priority 12 Priority 12 Priority 12 Priority 12	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Transmit End Interrupt Priority Error Interrupt Priority Bitrate			Data sampling on odd edg     Low when idle Disable     MSB First     m_ble_spp_host_spi_callb Priority 12 Priority 12 Priority 12 Priority 12 8000000	e, data variation on even edge ack
P  INTO	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Transmit End Interrupt Priority Error Interrupt Priority Bitrate Bitrate Modulation			Data sampling on odd edg     Low when idle     Disable     mr_ble_spp_host_spi_callb     Priority 12     Priority 12     Priority 12     Priority 12     S000000     Disabled	e, data variation on even edge ack
	Clock Phase Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Transmit End Interrupt Priority Error Interrupt Priority Bitrate Bitrate Modulation			Data sampling on odd edg     Low when idle     Disable     mr_ble_spp_host_spi_callb     Priority 12     Priority 12     Priority 12     Priority 12     S000000     Disabled	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Error Interrupt Priority Bitrate Bitrate Bitrate Modulation			Data sampling on odd edg     Low when idle     Disable     MSB First     m_ble_spp_host_spi_callb     Priority 12     Priority 12     Priority 12     Priority 12     Priority 12     Piosabled     P207	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Error Interrupt Priority Bitrate Bitrate Modulation V Pins TXD4 RXD4			Data sampling on odd edg     Low when idle     Disable     m_ble_spp_host_spi_callb     Priority 12     P	e, data variation on even edge ack
	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Error Interrupt Priority Bitrate Bitrate Modulation V Pins TXD4 RXD4 SCK4			Data sampling on odd edg     Low when idle     Disable     MSB First     m.ble_spp_host_spi_callb     Priority 12     Pri	e, data variation on even edge ack
0711119  - - - - - - - - - - - - - - - - -	Clock Phase Clock Polarity Mode Fault Error Bit Order Callback Receive Interrupt Priority Transmit Interrupt Priority Error Interrupt Priority Bitrate Bitrate Modulation V Pins TXD4 RXD4 SCK4 CTS4			Data sampling on odd edg     Low when idle     Disable     MSB First     m_ble_spp_host_spi_callb     Priority 12     Pri	e, data variation on even edge ack

# Figure 33. SPI Module Properties – Pin Assignments

2. Change the following SPI Configuration Properties in the properties window as in Figure 32 and Figure 33.

#### Table 6. SPI configuration

Property	Changed Value	Default Value
Common $\rightarrow$ DTC Support	Enabled	Disabled
Module g_spi0 SPI $\rightarrow$ Channel	4 for EK-RA4M2 case	0
Pins→TXD4	P207	None
Pins→RXD4	P206	None
Pins→SCK4	P400	None
Pins→CTS4	None	None
Pins→CTSRTS4	P401	None



- 3. Add the DTC driver (r\_dtc) to both transmit and receive. See Figure 32, SPI Module Properties,
  - g\_transfer0 SCI4 TXI **and** g\_transfer1 SCI4 RXI for more details.

Select Pin Configuration		📑 Export t	o CSV file 🛛 🔚 Configure Pi	n Driver Warning	gs
RA4M2 EK	✓ Manage configurations.		nerate data: g_bsp_pin_cf	g	]
Pin Selection	E ⊕ □ ↓ªz	Pin Configuration			
Type filter text         ✓       Peripherals         >       Analog:ADC         >       Analog:ANALOG         >       Analog:ANALOG         >       Connectivity:CAN         >       Connectivity:IIC         ✓       Connectivity:SCI         ✓       SCI0         SCI1       ✓         ✓       SCI2	^	Name Pin Group Selection Operation Mode Input/Output TXD4 RXD4 SCK4 CTS4 SDA4 SCL4 CTSRTS4	Value Mixed Simple SPI P207 P206 P206 P400 None None None None P401	Lock	Link C C C C C C C C C C C C C
SCI3	~	Module name: SCI4 Usage: When using When switch	Simple I2C mode, ensure po ing between I2C and other i	ort pins output ty modes. first disa	/pe is n-ch open dra ble.

Figure 34. Pin Configuration > SCI

For the TXD\_MOSI and RXD\_MISO pin assignments, it is necessary to specify which SCI channel to use on the **Pins** tab. For the EK-RA4M2 with RYZ012 PMOD connected to PMOD1 (conn J26), SPI is assigned to SCI Channel 4 (SCI4) and the **Operation Mode** set to **Simple SPI**. The Input/Output Pins are assigned as specified in Figure 34, Pin Configuration > SCI. If your application uses a different RA MCU or PMOD connector then refer to the RA MCU Schematic to determine the Pin assignments for the PMOD connector.

In the **Pin Configuration**>**Ports**, confirm that the **Ports** > **P4 Pin assignments** match the SPI Configuration in Table 5, SPP Module Configuration. These Pin assignments are for the EK-RA4M2 and RYZ012 connected to PMOD1 connector (J26). You will need to modify these assignments when using a different PMOD connector or evaluation kit / board type.



Select Pin Configuration		Export	to CSV file 🔚 Configure Pin Driver	Warnings
RA4M2 EK	✓ Manage configurations		ienerate data: g_bsp_pin_cfg	
Pin Selection		Pin Configuration		
Tuno filter tout		Name	Value	Link
lypeflitertext		Symbolic Name	PMOD1 SS	LINK
V Ports	^	Comment	1111001_00	
> 🗸 P0		Mode	Peripheral mode	
> V P1		Pull up	None	
> 2 P2		IRQ	None	
× × P4		Drive Capacity	Low	
✓ P400		Output type	CMOS	
✓ P401		✓ Input/Output		D
✓ P402		P401	SCI4_CISRIS	$\Rightarrow$
✓ P403		Markel Bull		
✓ P404		Nodule name: P401	TV0	
✓ P405	~	Port Capabilities: CANU: C		
Pin Function Pin Number				
· · · · ·				
🔅 [mcu_fwupdate_ryz012_ra4m2_bare	metal] FSP Configuration $ imes$			
Die Confirment				
Pin Configuration				
Select Pin Configuration		📑 Ехрог	t to CSV file 🛛 🗄 Configure Pin Drive	er Warnings
RA4M2 EK	Manage configuration	s	Generate data: a bsp pin cfa	
			<u> </u>	
Pin Selection	E ⊕ ₽↓ <mark>a</mark> z	Pin Configuration		
Type filter text		Name	Value	Link
Mark Borts		Symbolic Name	PMOD1_RST	
V POILS	<u>^</u>	Comment		
P1		Mode	Output mode (Initial Low)	
> V P2		Pull up	None	
> 🗸 P3		IRQ	None	
✓ ✓ P4		Drive Capacity	Low	
✓ P400		Output type	CMUS	
✓ P401		V Input/Output	- GRIO	
✓ P402		F405		~
✓ P403		Module name: P403		
✓ P404		Port Canabilities: AGTO: A	GTIO0	
• • • • • • • • • • • • • • • • • • • •	•		CTI01	
Pin Function Pin Number		1		
🛱 (				
[mcu_twupdate_ryz012_ra4m2_barer				
Pin Configuration			to CSV file 🔚 Configure Pin Driver	Warnings
Pin Configuration Select Pin Configuration		Export		
Pin Configuration Select Pin Configuration RA4M2 EK	Manage configurations	🖸 C	Generate data: g_bsp_pin_cfg	
Pin Configuration Select Pin Configuration RA4M2 EK	Manage configurations	🖸 C	Generate data: g_bsp_pin_cfg	
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection	✓ Manage configurations I □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	In Configuration	enerate data: g_bsp_pin_cfg	
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text	✓ Manage configurations Image: Image configurations Image: Image configurations	In Configuration Name	ienerate data: g_bsp_pin_cfg Value	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports	✓ <u>Manage configurations</u>	Image: Pin Configuration  Name Symbolic Name	enerate data: g_bsp_pin_cfg Value PMOD1_IO1	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports > V P0	✓ Manage configurations I → I → I → Z		Value PMOD1_IO1	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports > V P0 > V P1	✓ Manage configurations I → I → I	Pin Configuration  Name Symbolic Name Comment Mode Put up	Value PMOD1_IO1 Output mode (Initial Low)	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports V V Ports V V Pot V V P1 V V P2	✓ Manage configurations I → □ ↓ <sup>a</sup> <sub>Z</sub>	Pin Configuration  Pin Configuration  Name Symbolic Name Comment Mode Pull up IBO	value PMOD1_IO1 Output mode (Initial Low) None None	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports V V Ports V V P1 V V P1 V V P2 V P3	Manage configurations I ⊕ I <sup>a</sup> Z		enerate data: g_bsp_pin_cfg Value PMOD1_IO1 Output mode (Initial Low) None None Low	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports V V Ports V V Pot V P Pot V V P1 V V P2 V P3 V V P4 V P4 V P4	Manage configurations I ⊕ I <sup>a</sup> Z		Value PMOD1_IO1 Output mode (Initial Low) None Low CMOS	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text V V Ports V V Ports V V Pot V P 1 V P 2 V P 3 V V P 4 V P 400 V P 400 V P 400 V P 400 V P 401	✓ Manage configurations E ⊞ ⊟ J <sup>a</sup> z		Value PMOD1_IO1 Output mode (Initial Low) None Low CMOS	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text  V V Ports > V P0 > V P1 > V P2 > V P3 V P4 V P400 V P401 V P401 V P402	▼ Manage configurations	Pin Configuration  Name Symbolic Name Comment Mode Pull up IRQ Drive Capacity Output type v Input/Output P402	Generate data: g_bsp_pin_cfg Value PMOD1_IO1 Output mode (Initial Low) None Low CMOS ✓ GPIO	Link
Pin Configuration Select Pin Configuration RA4M2 EK Pin Selection Type filter text  V V Ports V V Ports V V P0 V P1 V V P2 V P3 V P4 V P400 V P400 V P401 V P402 V P403	Manage configurations I → I → I Z	Fin Configuration  Pin Configuration  Name Symbolic Name Comment Mode Pull up IRQ Drive Capacity Output type ∨ Input/Output P402	Value PMOD1_IO1 Output mode (Initial Low) None Low CMOS ✓ GPIO	Link
Pin Configuration           Select Pin Configuration           RA4M2 EK           Pin Selection           Type filter text           ▼ ♥ Ports           > ♥ P0           > ♥ P1           > ♥ P2           > ♥ P4           ♥ ♥ P400           ♥ P401           ♥ P403	Manage configurations E + □ J <sup>a</sup> <sub>Z</sub>	Export     Configuration     Name     Symbolic Name     Comment     Mode     Pull up     IRQ     Drive Capacity     Output type     VInput/Output     P402     Module name: P402	Value PMOD1_IO1 Output mode (Initial Low) None Low CMOS ♥ GPIO	Link

Figure 35. Pin Configuration > Ports





Figure 36. Stack Configuration > g\_pcdc0 USB PCDC (r\_usb\_pcdc)

# 5.5 Add USB PCDC for Tera Term Menu and XMODEM File Transfer

Add the USB PCDC stack for the USB Full Speed Port (J11) on the EK-RA4M2 and Tera Term communications.

# 5.6 Add MCU Flash Driver for XMODEM File Storage



Figure 37. Stack Configuration > g\_flash0 Flash (r\_flash\_hp)

Common > Code Flash Programming Enable must be set to Enabled.



The MCU Flash size must be large enough for the RYZ012A firmware binary image file. For the EK-RA4M2, in this sample project (see file src/common\_init.h), the code flash used for the RYZ012 firmware image starts at

#define RYZ012\_FWIMAGE\_START\_ADDRESS 0x00060000 which is above the user application space. Make sure you adjust the RYZ012\_FWIMAGE\_START\_ADDRESS and RYZ012\_FWIMAGE\_END\_ADDRESS address for your custom application.

The largest RYZ012A firmware image used in this sample application is **102436 bytes** for **8258\_moduleV5\_5.bin**.

The RYZ012A firmware image size can increase in future RYZ012 firmware releases as new features are added. Be sure to increase the number of Flash Blocks required to store the firmware image file as the RYZ012A image size increases.

See file src/common\_init.h #define RYZ012\_FWIMAGE\_END\_ADDRESS 0x00080000 and #define
RYZ012\_FWIMAGE\_NUM\_BLOCKS.

Stacks	Configuration			Stacks	Configuration		
Stacks	comgaration			Stacks	configuration		
Threads	🐑 New Thread 🔹 Remove 😑	g_blinker Timer, General PWM (r_gpt) Stacks		Threads	🐑 New Thread 🔒 Remove 😑	g_gpt_green Timer, General PWM (r_gpt) Stacks	
<ul> <li>✓</li></ul>	MU/Common g_ippert U/O Fort (r, ipport) g_ippert of USB PCC (r, usb, pcde) g_ipplr ed Timer, General PVM (r, gpt) g_ipplr garen Timer, General PVM (r, gpt) g_ippl tube Timer, General PVM (r, gpt)	g_blinker Timer, General PWM (r_gpt)			AL/Common g.pcdt U/D pot (f_ioport) g.pcdt U/D sP (Cf_ioport) g.gct and U/D sP (Cf_iopt) g.gt/cera Timer, General PVM (r.gpt) g.gt/cera Timer, General PVM (r.gpt) g.gt/but Timer, General PVM (r.gpt) g.fabr/Ebak (fish-fp) SPP BLE Abstraction (m_ble_abs_spp)	÷ a_get_green Timer, General PVM (r_gpt) 0	
				Objects	ᡚ New Object > 🔬 Remove		
Summary	BSP Clocks Pins Interrupts Event Links	Stacks Components					
Probler	ms 📮 Console 🔲 Properties 🗙 🐝 S	nart Browser 👔 Stack Analysis 🎼 Smart Manual 🗸	🔗 Search 🚦 Memory	Summary	BSP Clocks Pins Interrupts Event Links	Stacks Components	
g_blinke	er Timer, General PWM (r_gpt)			Probler	ns 📮 Console 🔲 Properties 🗙 🏟 Sm	nart Browser 👔 Stack Analysis 斗 Smart Manual 🛷 S	earch 🔋 Memo
Settings	Property		Value	g_gpt_g	een Timer, General PWM (r_gpt)		
API Info	✓ Common			Settings	Property		Value
	Parameter Checking		Default (BSP)	API Info	Common     Parameter Checking		Default (DSD
	Write Protect Enable		Disabled		Pin Output Support		Enabled
	Clock Source		PCLKD		Write Protect Enable		Disabled
	Module g_blinker Timer, General PWI     General	/ (r_gpt)			<ul> <li>Module g_gpt_green Timer, General PV</li> </ul>	WM (r_gpt)	T COND
	> Output				> General		
	> Input				> Input		
	> Interrupts > Extra Features				> Interrupts > Evtra Features		
	✓ Pins				✓ Pins		
	✓ Pins GTIOC2A GTIOC2B		P103 None		✓ Pins GTIOC3A GTIOC3B		None P404
×	V Pins GTIOC2A GTIOC2B		P103 None	🖉 (mcu	Pins     GTIOC3A     GTIOC3A     GTIOC3B  fwupdate.rvz012 ra4m2 baremetall FSP Co	nfiguration ×	None P404
<b>Ö</b> (mcu_fw	Pins     GTIOC2A     GTIOC2B wupdate_rys012_rs4m2_baremetal] FSP Config	uration X	P103 None	₩ (mcu Stacks	Pins     GTIOC3A     GTIOC3B     Fwupdate_ryz012_rs4m2_baremetal] FSP Co     Configuration	nfiguration ×	None P404
‡ (mcu_fw Stacks C	Pins     GTIOC2A     GTIOC2B  wupdate_ryz012_rs4m2_baremetal] FSP Config configuration	uratice ×	P103 None	Č (mcu Stacks	V Pins GTIOC3A GTIOC38 (wupdate_ryz012_ra4m2_barenetal) FSP Co	rfguration X	None P404
[mcu_fw Stacks C Threads	Pins GTIOC2A GTIOC2A Store (Configuration)     Pins New Thread    Remove	uration × g.gpt_blue Timer, General PWM (r.gpt) Stacks	P103 None	Stacks Thread	V Pins GTICCIA GTICCIA GTICCIA (mupdate_ryc012_ra4m2_baremetal) FSP.Col Configuration Configuration Configuration	nfguration ×	None P404
(mcu_fw Stacks C Threads HA	V Pins     GTOC2A     GTIOC2A     GTIOC2A     GTIOC2A     GTIOC2A     Configuration     New Thread    Remove       (appl deproved to the second deproved deprov	uration ★ g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) ①	P103 None	Stacks Thread	<ul> <li>✓ Piss GTOCSA GTOCSA GTOCSB</li> <li>Joupdate,yg012,sc6n2,bacenetal [SP Coll Configuration]</li> <li>← New Text (Lypert)</li> <li>← New Text (Lypert)</li> <li>← Mark Timer, General PWM (Lypt)</li> <li>← Application Timer, General PWM (Lipt)</li> <li>← Applica</li></ul>	nfgunton X g.gpt_red Timer, General PWM (r.gpt) Stacks	None P404
Constants	V Pins GTOC2A GTIOC2A GTIOC2A Configuration New Thread	uration × g.gpt_blue Timer, General PWM (r.gpt) Stacks	P103 None	Stacks	<ul> <li>✓ Piss GTOC3A GTOC3B</li> <li>Jourdate syd12 cade 2 barenetid [SP Coll Configuration</li> <li>→ New New Constant (SP Coll Paulation)</li> <li>→ New New Constant (SP Coll Paulation)</li> <li>→ Second PM (Lapped)</li> <li>→ Agendet (SP Coll Cusb pack)</li> <li>→ Agendet (SP Coll Cusb pack)</li></ul>	nfguration X g.gpt_red Timer, General PVM (r.gpt) Stacks from g.gpt_red Timer, General PVM (r.gpt) 0	None P404
	V Pins     GOC2A     GTIOC2B      Configuration     Pins Thread    Remove      Aucommon     gupdat gray color (gupdate)     grad Obs PCC (gubd_ped)     gupdat Obs PCC (gubd_ped)     gupdat Obs PCC (gubd_ped)     gupdat Diss (General PWM (gupd)     gupt grean Times, General PWM	uration ★ g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) () () () () () () () () () (	P103 None	C (mcc) Stacks Thread ✓ 20 Objects	<ul> <li>✓ Piss GTOCSA GTOCSA GTOCSB</li> <li>Joupdate,ydf2,cs6n2,barenetal [SP Coll Configuration</li> <li>→ New Theorem (Pisson Coll Barbon (Cush pack)</li> <li>→ Second Pisson (Cush pack)</li> <li>→ gackdu SP Coll (Cush pack)</li> <li>→ SP BLE Akstaction (m_ble, abs, pp)</li> </ul>	nfgunton X g.gpt_red Timer, General PVM (r.gpt) Stacks	None P404
Constraints of the second seco	V Pins     GIOC2A     GIIOC2A     GIIOCA	uration × g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) che Sept_blue Timer, General PWM (r.gpt) che Sept_blu	P103 None	Contraction of the second seco	V Piss GTOC3A GTOC3A GTOC3B /mupdate.ycp?l2.ycfn2	nfguration X g.gpt_red Timer, General PVM (r.gpt) Stacks (g.gpt_red Timer, General PVM (r.gpt) 0 Stacks   Components	None P404
Stacks C Stacks C Threads T	Pins GTOC2A GTOC2A GTOC2A Configuration Configuratio Configuration Configuration Configuration Configuration Co	untion X g.gpt.blue Timer, General PWM (r.gpt) Stacks g.gpt.blue Timer, General PWM (r.gpt) G Components throwser Stack Analysis S Smart Manual S Sec	P103 None	Co Summary Co Summary Problem	V Pris GT0C3A GT0C3B      (mupdate.ydd12.eddn2.barentedij FSP Co Configuration      (********************************	nfguration ×  g.gpt_rod Timer, General PVMM (r.gpt) Stacks  fig.g.gpt_rod Timer, General PVMM (r.gpt)  Sacks Components Stacks Components Stacks Components Stacks Components Stack Analysis CQ Smart Manual #	None P404
<ul> <li>(mcu_fw</li> <li>Stacks C</li> <li>Threads</li> <li>Threads</li></ul>	Pins     GTOC2A     GTIOC2A     GTIOC2     GTIOC	uration X g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) g. cts Components t Browser L Stack Analysis Q Smart Manual & Se	P103 None	Co Summary Summary Summary Proble Summary	V Pris GTOC3A GTOC3A GTOC3B Insuptite cycl2 cycle _ beventeld [SP Col Configuration B	nfguation X g.gt.ted Timer, General PVM (r.gpt) Stack: g.ggt.ted Timer, General PVM (r.gpt) Stacks (Components) Mark Browser Stack Analysis Stant Manual X	earch [] Memo
<ul> <li>Imeau fw</li> <li>Stacks C</li> <li>Threads</li> <li>♥</li> <li>♥&lt;</li></ul>	V Pins     GOOC2A     GTIOC2B      Configuration      New Thread    Remove      New Thread    Remove      Question of the second of the	uration × g. gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) g. cbs_components t Browser Stack Analysis Stack Smart Manual & Sec	rch () Memory =;; Pr Value Default (SSP)	C (mea Stacks Thread ♥ ₽ Object ♥ ♥ Pobl S.eptJ Setting	V Pins GTOC3A GTOC3A GTOC3B Jourdate.ycdl2.scfa_2.bcemetd1 FSP Co Configuration Configura	nfguration X g.g.pt_red Timer, General PVM (r.g.pt) Stack:	earch () Memo
© [mcu_fn Stacks C Threads	Pins GTOC2A GTIOC2A GTIOC2A Configuration Configuration Configuration Configuration Configuration Configuration Configuration Configuration Sectors Sectors Sectors Sectors Sectors Concerned PWM (cgPl) Sectors Sectors Sectors Sectors Concerned PWM (cgPl) Sectors Sectors Sectors Sectors Concerned PWM (cgPl) Sectors Secto	uration × g.gpt_blue Timer. General PWM (r.gpt) Stacks g.gpt_blue Timer. General PWM (r.gpt) Components tBrowser Stack Analysis CS Smart Manual Sec.	arch () Memory =;; Pi Value Default (BSP) Enabled	Co Summary APlinto	V Pris GT0C3A GT0C3B (nupdate, yot12, zelon, 2, barometal) ESP Co- Configuration (a) Second	nfguration X g.gpt_ref Timer, General PWM (r.gpt) Stacks	earch () Memo
Stacks C Threads Th	V Pris     GTOC2A     GTIOC2A     GTIOC2     GTI	uration X g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) Components tBrowser B Stack Analysis G Smart Manual & Se	P103 None	Contractions C	V Pris GTOC3A GTOC3A GTOC3B (mupdate, yod12, zeka, 2, bavenetal) ESP Co Configuration HAU/Common g _upedvis Proc(usb, pod) g _upedvis Proc(usb, pod) g _upedvis Procent PMN (c, gpt) g _upedvis Procent PMN (c, gpt) P (Cocks Proc Interrupts Tevent Links mm _ C cocks Procent PMN (c, gpt) Property V Common	nfguation X  g.g.pt.ref Times General PVMM (r.g.pt) Stacks  g.g.g.pt.ref Times  g.g.g.pt.ref Times  stacks [Components]  matBrowser S Stack Analysis C Smart Manual #	earch () Memo Value Default (BSP Enabled
Stacks C Threads Th	V Pins     GOC2A     GTIOC2A     GTI	uration ★ g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) () () cks [Components] t Browser [b] Stack Analysis [ii] Smart Manual \$\angle\$ See (r.gpt)	P103 None arch () Memory III P Value Default (BSP) Enabled PCLKD	Contractions C	V Piss GTOCSA GTOCSA GTOCSA GTOCSB //wpdate.ydfl2.sefm2.beremetal [SP Col Configuration #	nfguration X g.gpt, red Timer, General PVM (r.gpt) Stack: g.gpt, red Timer, g.gpt, red	earch () Memo Value Defaux (SSP Enabled Pict KD
Stacks C Threads Th	Y Pins     GTOC2A     GTOC2	uration × g.gpt_blue Timer, General PWM (r.gpt) Stacks g.gpt_blue Timer, General PWM (r.gpt) cts Components t Browser  Stack Analysis Stack Manual  Stack	P103 None	Co Summary API info	V Pris GT0C3A GT0C3B (Nupdate, ydd1, zelon, 2, barenetal) EPP Col- Configuration (Nupdate, ydd1, zelon, 2, barenetal) EPP Col- Configuration (Nupdate, ydd1, zelon, 2, barenetal) (Nupdate, ydd1, ydd1,	nfgurøtion X g.gpt_ref Timer, General PVM (r.gpt) Stacks fig g.gpt_ref Timer, General PVM (r.gpt) 0 Stacks (Components) mart Browser 🚡 Stack Analysis 💭 Smart Manual 🖋 5 M (r.gpt)	earch () Memc
Stacks C Threads V # H4 + H4	V Pris     GTOC2A     GTOC2	suntion X g. gpt, blue Timer, General PWM (r.gpt) Stacks g. gpt, blue Timer, General PWM (r.gpt) g. eks: Components t Browser S Stack Analysis S Smart Manual & See (r.gpt)	P103 None	Conception Conceptication Conception Conception Conception Conception Concep	V Pris GTOC3A GTOC3A GTOC3B      (mupdate, yod12, ex6m, 2, barenetal) ESP Co Configuration     (Configuration     (Configuration     (Configuration     (Configuration     (Configuration     (Constant)     (C	nfguration X  g.g.pt.rof Times. General PVMM (r.g.pt) Stacks  (g.g.pt_rof Times. General PVMM (r.g.pt)  (g.g.pt_rof Times.  Stacks [ Components ] mant Browser Stack Analysis [] Smart Manual # 4  M(r.g.pt)	earch [] Memo Value Default (BSP Enabled Disabled PCLKD
Stacks C Threads Th	V Pins     GOOC2A     GOOC2	uration X  g.gpt_blue Timer, General PWM (r.gpt) Stacks  g.gpt_blue Timer, General PWM (r.gpt)  ( cts [ Components]  t Browser ] Stack Analysis [2] Smart Manual  Sec  (r.gpt)	P103 None	Contractions of the second se	V Piss GTOC3A GTOC3A GTOC3B Jourdate cycl2 cycl2 cycla 2 berneticit JSP Co Configuration HAU/Common Pi gioped VI Pert (Lipert) Pi pert VI Pert (Lipert) Pert VI Pert VI Pert (Lipert) Pert VI Pert VI Pert VI Pert VI Pert VI Pert VI Pert VI Pert VI Pert VI Pert VI VI Pert VI Pert VI Pert VI VI Pert VI Pert VI Pert VI VI Pert VI VI Pert VI VI Pert VI Pert VI Pert VI Pert VI Pert VI Pert VI VI Pert VI Per	nfguration X  g.gpt.red Timer, General PWM (r.gpt) Stacks  (g.gpt_red Timer, (g.gpt)  Stacks Components  Stacks Components  M(r.gpt)	earch () Memo Value Default 68 Disabled PCLKD
Stacks C Threads Stacks C Threads Stacks C Threads Poly Objects Summary    S Problem S get bli API Info	V Pris     GTOC2A     GTOC2	g gpt_blue Timer, General PWM (r.gpt) Stacks  g.gpt_blue Timer, General PWM (r.gpt)  g cks Components t browser  Stack Analysis  Smart Manual  Sec (r.gpt)	P103 None	Contraction of the second seco	V Pris GT0C3A GT0C3A GT0C3B     (mupdate,yot12,yot6,2,barenetal) ESP Col Configuration     (mupdate,yot12,yot6,2,barenetal) ESP Col (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,yot12,yot6,2,barenetal) (mupdate,	nfgurston X  g.g.pt_ref Time: General PVM (r,gpt) Stacks  fig.g.g.pt_ref Time: General PVM (r,gpt)  fig.g.g.t  stacks Components  stacks Components  M(r,gpt)  M(r,gpt)	earch () Memo
Stacks C Threads C C C C C C C C C C C C C	V Pris     GTOC2A	g. gpt, blue Timer, General PWM (r. gpt) Stacks  g. gpt, blue Timer, General PWM (r. gpt)  ches Components  t Browser S Stack Analysis S Smart Manual	P103 None arch () Memory = p Value Default (SSP) Enabled Diabled PCLKD P415	C Incod Stacks Tread Stacks Summary Su	V Pins GTOC3A GTOC3A GTOC3B  (mupdate, yold2, y	nfguration X  g.g.pt_rod Times, General PVMM (r.g.pt) Stacks  figuration graph and Times, General PVMM (r.g.pt)  Stacks Components mat Browser S Stack Analysis (L) Smart Manual X  (r.g.pt)	earch () Memo Value Value Default (BSP) Enabled Deabled PCLKD

Figure 38. Stack Configuration Settings



# 5.7 Add General PWM Timers for LED Indications

This step is optional and is not required for Firmware Update functionality. It provides an indication on the board LEDs that the application is running. Add the General PWM Timers ( $r_gpt$ ) for the LED blinker Timer and board LEDs.

# 5.8 Create BLE Profile with QE for BLE

QE for BLE can generate a custom BLE profile and the Bluetooth LE application skeleton code. You must modify this source code according to your application BLE functional requirements. See Renesas QE for BLE Tool information <u>QE BLE Tool Dev Assistance Documents</u> about the usage of QE for BLE.

The BLE Profile created in this application project is as follows. From the e<sup>2</sup> studio Menu go to **Renesas Views > Renesas QE > R\_BLE Custom Profile RA,RE,RX (QE).** Once the BLE Custom Profile opens, then select the **Project:** <YourProject> and **Module:** RYZ012 from the drop-down boxes.

Profile Perinheral Central			Module:	YZ012 ∨ Projec	t: mcu_fwupdate_ryz012_ra4m2_baremeta	1
✓ (P) Profile	Name:	GAP Service				1
Iserver, Client] GAP Service Device Name	UUID:	0x1800			16 bits 🗸 🗸	i l
C Appearance	Abbreviation:	gan				1
C Peripheral Preferred Connection Parameters C Central Address Resolution	Description	The conoria acces	- contine containe a	anaris information	about the device	
C Resolvable Private Address Only	Description:	The generic_acces	s service contains g	eneric information	about the device.	1
Server] GATT Service     Service Changed	Aux Properties:	Authorization				
> 🦕 Service Changed	Security Level:	Level 1: No Sec     Level 2: Unauth     Level 3: Auther     Level 4: Auther	urity (No Authentic nenticated pairing w nticated pairing with nticated LE Secure C	ation and no Encry with Encryption n Encryption Connections with pa	ption) iring with Encryption	
	Callback:	Enable Write C	allback allback			
	Included:	GATT Service				
		÷ X				1
		Name	Code			
	Error Codes:					

Figure 39. Creating BLE Profile with QE for BLE

The minimum BLE Profile is shown above and nothing specific is needed for the MCU based Firmware Update of the RYZ012A. The only optional modification required is to select the **Peripheral** tab to change **Local Name** to use for BLE Advertising in the RYZ012. Set **Complete local name** to "**RYZ012**".



🕂 🛪 🗶 Export Import 🔂 Gen	rrate Code Module: RYZ012 v Project: mcu_fwupdate_ryz012_ra4m2_baremetal
Profile Peripheral Central	
<ul> <li>✓ Advertising Data 11/31</li> <li>✓ Flags</li> <li>☐ Service Class UUIDs</li> </ul>	Local Name     Short local name     Short local name
✓ Local Name Tx Power Level Slave Connection Interval Range	Complete local name      RVZ012
<ul> <li>Service Solicitation UUIDs</li> <li>Service Data</li> <li>Public Target Address</li> </ul>	
Random Target Address     Appearance     Advertiging Interval	
Manufacturer Specific Data      Scan Response Data 0/31	
<ul> <li>Service Class UUIDs</li> <li>Local Name</li> <li>Tx Power Level</li> </ul>	
Slave Connection Interval Range Service Solicitation UUIDs Service Data Public Target Address Random Target Address	Sets the name of the device. Specify the local name so that it is the same as the device name for the characteristics of the GAP Service. The type of name can be selected as either of the following two. • Short Local name: Shows an abbreviated name of the device. • Complete local name: Shows the complete name of the device.
<ul> <li>☐ Appearance</li> <li>☐ Advertising Interval</li> <li>☐ Manufacturer Specific Data</li> </ul>	

Figure 40. Setting Complete Local Name

The GATT Browser app on a mobile device can be used to verify that the RYZ012 is running (advertising) after a firmware update.

After the BLE Profile is created and customized for your application, press **Generate Code** to create the BLE application skeleton code and BLE Profile. The tool generated source code will appear under /qe\_gen/ble/app\_main.c and qe\_ble\_profile.c.

# 5.9 Build and Modify Application Skeleton Code

- Select **Project > Build Project** from the menu bar or click the Build icon 10 to build the project.
- Click the debug icon 🔅 to launch the project. When the project starts, the sample application will be downloaded to EK-RA4M2.

# 6. Next Steps

Now that you have a basic understanding of the process to update the RYZ012 firmware with the EK-RA4M2 MCU you may want to customize the application. Please consider this information as you make those changes:

1. If you need to change the RYZ012 to a different PMOD connector, use a different RA kit, or use UART comms instead of SPI with the RYZ012, then:

#### To change RYZ012 Communications Interface Mode

 To change the RYZ012 comms interface to UART see section 5.4, Add and Configure the SPP BLE Module. When adding the SPP BLE Abstraction Stack, add the UART for "Transport Interface for communicating with the module".

Similar to configuring SPI, see section 5.4.2, Configure Peripherals for SPP BLE Abstraction Driver, to configure the UART Ports and Pins required for the PMOD connector on your board. In the step where Input/Output Pins are assigned as specified in Figure 34, Pin Configuration > SCI, set the Operation Mode to **Asynchronous UART**.

As a reference, see Figure 35, Pin Configuration > Ports, the PMOD Mode Select GPIO (UART or SPI mode is set by GPIO PMODx\_IOx) must be configured to **UART** by setting the Pin Configuration > Mode to **Output mode (Initial High).** 



in Configuration				
elect Pin Configuration		E.	Export to CSV fil	e 🖺 Configure Pin Driver Warnin
RA4M2 EK	✓ Manage config	urations	🗹 Generate d	lata: g_bsp_pin_cfg
Pin Selection 🔡 🕀 🖨 🕌	Pin Configuration			
Type filter text	Name	Value	Link	
D and DO	Symbolic Name	PMOD1_IO1		
2 Y FU 71	Comment			
D1	Connent			
> V P1	Mode	Output mode (Initial H	ligh)	
> V P1 > V P2	Mode Pull up	Output mode (Initial H None	ligh)	
> ✓ P1 > ✓ P2 > ✓ P3	Mode Pull up IRQ	Output mode (Initial H None None	ligh)	
> \left P1 > \left P2 > \left P3 \left P4 \left P400	Mode Pull up IRQ Drive Capacity	Output mode (Initial H None None Low	ligh)	
> \left P1 > \left P2 > \left P3 \left P4 \left P400 \left P401	Mode Pull up IRQ Drive Capacity Output type	Output mode (Initial H None Low CMOS	ligh)	
<ul> <li>&gt; ✓ P1</li> <li>&gt; ✓ P2</li> <li>&gt; ✓ P3</li> <li>✓ P4</li> <li>✓ P400</li> <li>✓ P401</li> <li>✓ P402</li> </ul>	Mode Pull up IRQ Drive Capacity Output type V Input/Output	Output mode (Initial H None Low CMOS	ligh)	

Figure 41. Pin Configuration > Mode

Consult the schematic for your particular board to determine the correct Ports / Pins for the PMOD connector signals used in your application. Configure and confirm the **Peripherals > SCIx settings** and **Ports > Px > Pin Configuration** settings for all the PMOD signals.

If the PMOD signal pins do not appear as available in the Pin Configuration or Peripheral SCI channel, it may be required to disable unused SCI channels to gain access to the required PMOD Pins on your desired SCIx channel. Use **Peripherals > SCIx > Operation mode = Disabled** 

#### To change RYZ012 PMOD Connector

— To use a different PMOD connector other than PMOD1 (J26):

Consult the schematic for your particular board to determine the correct Ports / Pins for the PMOD connector signals used in your application.

Follow section 5.4.2, Configure Peripherals for SPP BLE Abstraction Driver to configure the Ports & Pins and comms mode (UART or SPI) required for the PMOD connector on your board.

Configure and confirm the **Peripherals** > **SCIx** settings and **Ports** > **Px** > **Pin Configuration** settings for all the PMOD signals.

If the PMOD signal pins do not appear as available in the Pin Configuration or Peripheral SCI channel, it may be required to disable unused SCI channels to gain access to the required PMOD Pins on your desired SCIx channel. Use **Peripherals > SCIx > Operation mode = Disabled** 

#### To change the RA MCU kit or Custom Board

 To use a different RA MCU kit or custom board design it may be required to reassign the communications interface Port / Pins for the RYZ012 PMOD connector on your board. You will also be required to change the e<sup>2</sup> studio FSP Configuration BSP settings for the RA MCU kit or custom board Device used.

#### See To change RYZ012 PMOD Connector above.

It may be required to change the board LED port and pin assignments.

See section 5.7 Add General PWM Timers for LED Indications to change the LED pins. Keep in mind that your RA MCU must have enough Code Flash to contain the entire RYZ012 firmware image. See Item 4 below for more details.

2. To convert this e<sup>2</sup> studio project to use with IAR or Keil see the App Note for converting the project to IAR. <u>Converting Applications from e<sup>2</sup> studio to IAR or Keil for RA – Application Note (renesas.com)</u>



3. Renesas FSP v3.8.0 and later must be used to support the RYZ012 (FW SPP SDK v5.4 and later) for MCU based firmware update.

The XMODEM file transfer depends on the MCU code flash size available. The MCU code flash size requirement can change as future RYZ012 firmware versions are released.

The MCU Flash size must be large enough to hold the entire RYZ012A firmware binary image file. For the EK-RA4M2, in this sample project (See file src/common\_init.h), the code flash used for the RYZ012 firmware image starts at

#define RYZ012\_FWIMAGE\_START\_ADDRESS 0x00060000 which is above the user application space. Make sure you adjust the RYZ012\_FWIMAGE\_START\_ADDRESS and RYZ012\_FWIMAGE\_END\_ADDRESS address for your custom application.

Currently, the largest RYZ012A firmware image used in this sample application is **102436 bytes** for **8258\_moduleV5\_5.bin**. The RYZ012A firmware image size can increase in future RYZ012 FW releases as new features are added. Be certain to increase the number of MCU Flash Blocks required to store the firmware image file as the RYZ012A image size increases. See file src/common\_init.h **#define** RYZ012\_FWIMAGE\_END\_ADDRESS 0x00080000 and #define RYZ012\_FWIMAGE\_NUM\_BLOCKS.

Make sure the #define RYZ012\_FWIMAGE\_START\_ADDRESS resides in free space above the MCU application code flash used. Otherwise, you may erase MCU application code during the XMODEM file transfer flash erase step.



# Website and Support

Visit the following vanity 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 RA™ EK-RA4M2 kit RYZ012x1 Bluetooth LE Module PMOD Expansion Board RYZ012x1 Renesas Support www.renesas.com/ra www.renesas.com/ra/forum www.renesas.com/FSP renesas.com/ra/ek-ra4m2 renesas.com/ryz012x1-bluetooth-le-module renesas.com/pmod-expansion-board-ryz012x1 www.renesas.com/support



# **Revision History**

		Description	
Rev.	Date	Page	Summary
1.00	Feb.10.23	—	Initial release
1.01	Sep.26.23	17	Revised section 2.4 Firmware Update Considerations



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

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

#### 1. Precaution against Electrostatic Discharge (ESD)

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

#### 2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

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

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

#### Notice

- Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
- Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
- 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
- 4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
- 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.
- 6. 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

#### www.renesas.com

#### Trademarks

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

# **Contact information**

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