RZ/G series

FSP Example Project Usage Guide

Introduction

This Flexible Software Package (FSP) Example Project Usage Guide provides steps and guidelines for operating example projects which use the RZ/G FSP.

Target Device

- RZ/G2L
- RZ/G2LC
- RZ/G2UL
- RZ/G3S
- RZ/G3E

Supported Kit

- RZ/G2L Evaluation Board Kit.
- RZ/G2LC Evaluation Board Kit.
- RZ/G2UL Evaluation Board Kit.
- RZ/G3S Evaluation Board Kit.
- RZ/G3E Evaluation Board Kit.

Contents

1. Prerequisites

1. Tool experience: It is assumed that the user has prior experience working with integrated development environments, such as e2 studio, SEGGER J-Link RTT Viewer and terminal emulation programs, such as Tera Term.

2. Subject knowledge: It is assumed that the user has basic knowledge about microcontrollers, embedded systems, and FSP to modify the example projects. First time users are recommended to refer to FSP User Manual for Tutorial on *Getting Started with Flexible Software Package*, paying special attention to sections as follow.

   - Set up a SMARC EVK
   - Tutorial: Your First RZ MPU Project – Blinky
   - Importing an Existing Project into e2 studio

3. The screen shots provided throughout this document are for reference. The actual screen content may differ depending on the version of software and development tools used.

2. Hardware and Software Requirements

RZ/G FSP Example projects are designed to operate using Evaluation Board Kit for RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/G3S and RZ/G3E MPU officially supported by Renesas.
Refer to the readme.txt file in the specific module folder of */example_projects* folder for additional hardware and software requirements for running the projects.

Note:

Some projects may require external hardware as mentioned in the respective readme.txt files.

Software Requirements

- Windows® 10 operating system
- RZ/G FSP v3.1.0
- e² studio 2025-07
- SEGGER J-Link RTT Viewer v7.96j

3. Tool Installation

3.1 FSP and tools installation

Download and install the latest version of FSP and tools from FSP GitHub repository.

1. Open FSP GitHub repository: https://github.com/renesas/rzg-fsp
2. Go to the ***Releases*** section of Git and navigate to latest FSP section.
3. Follow the instructions on installing and using FSP and e² studio.

3.2 J-Link RTT Viewer installation

Download and install SEGGER J-Link Software for Windows from

https://www.segger.com/downloads/jlink#J-LinkSoftwareAndDocumentationPack.

Default install path is `C:\Program Files\SEGGER\JLink`.

Note: Select version 7.96j from the drop-down menu in Version tab.

4. Importing and Running the Project

4.1 Downloading the Project

Download the example project which is "RZ/G FSP Example Projects" from RZ/G Multi-OS Package | Renesas.

4.2 Running the project

4.2.1 Importing the project into e2studio

1. Import an existing project.
   Refer to the section *Importing an Existing Project into e2 studio* of *Getting Started with Flexible Software Package.*
2. Generate project content.
   Double clicks to open **configuration.xml** and then click **Generate Project Content**.

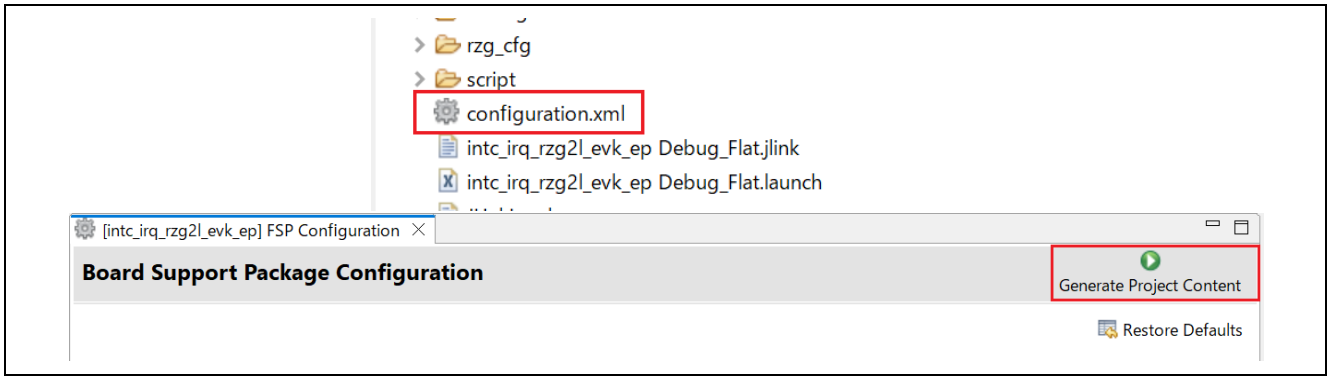


Figure 1: Generate project content

3. Build the project.
   There are three ways to build a project:
   a. Click on **Project** in the menu bar and select **Build Project**.

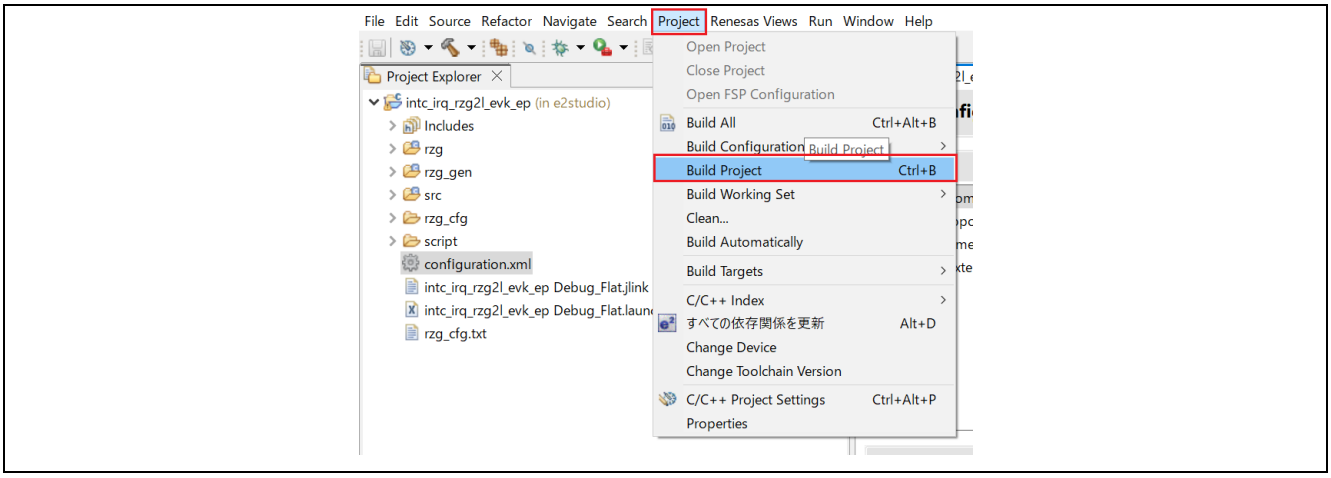


Figure 2: Build Project selection

   b. Click on the hammer icon.

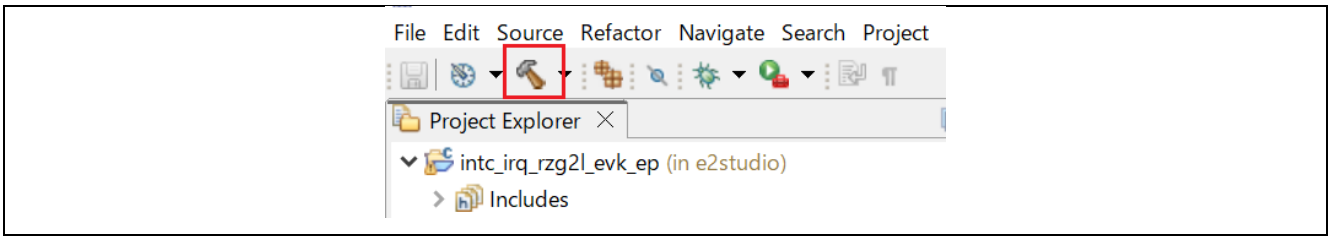


Figure 3: Click hammer icon

c. Right-click on the project and select **Build Project**.

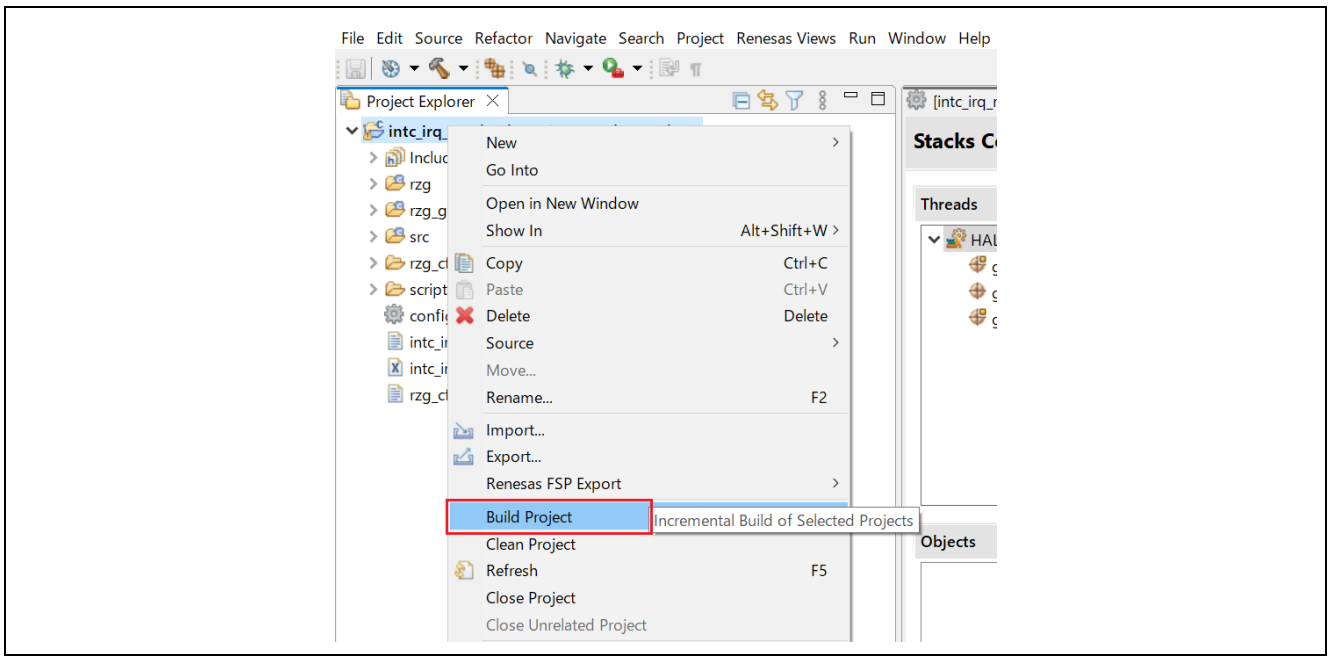


Figure 4: Build Project selection

4. Downloading the project image to the board.
   Click **Debug** to begin debugging the application.

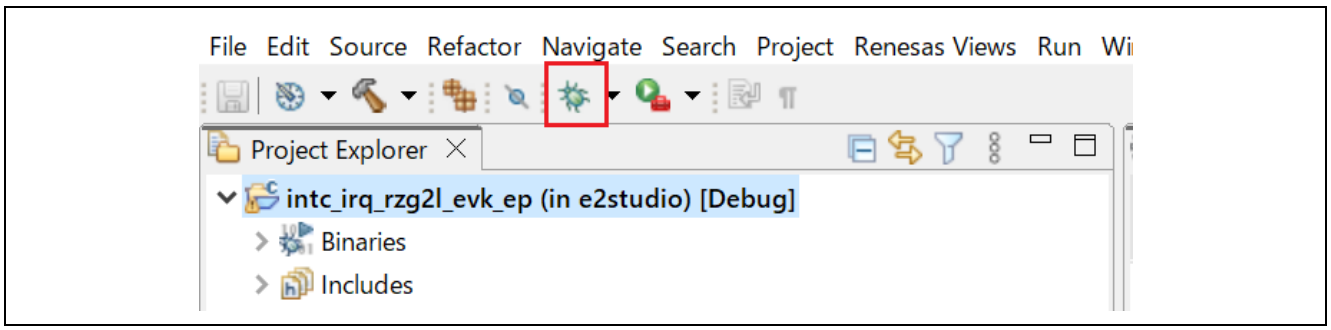


Figure 5: Debugging the application

4.2.2 Connecting with J-Link RTT Viewer

1. Open RTT Viewer by double clicking **JLinkRTTViewer.exe** in the installed /SEGGER/Jlink folder.

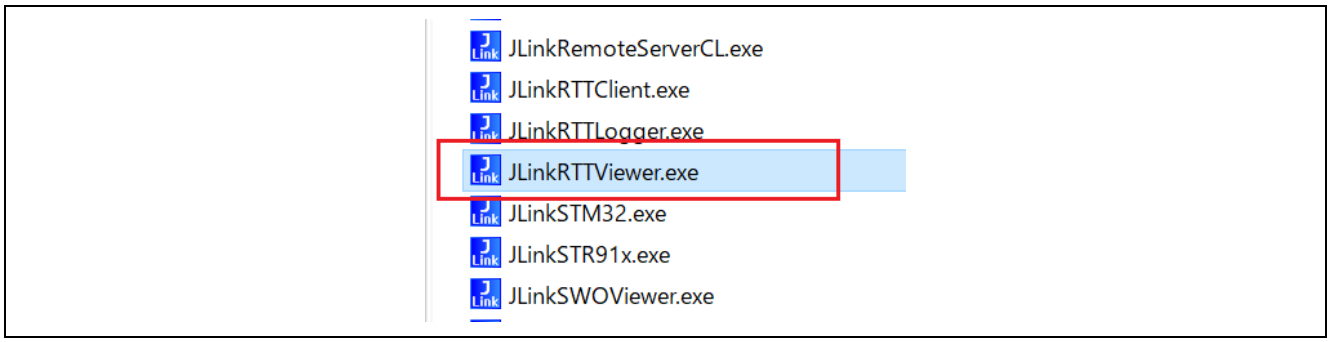


Figure 6: JLinkRTTViewer

2. On opening, the field **Specify Target Device** shows up as **unspecified**. Click on the "…" tab to select the Cortex-M33 device (Cortex-M33 Core) as follows.

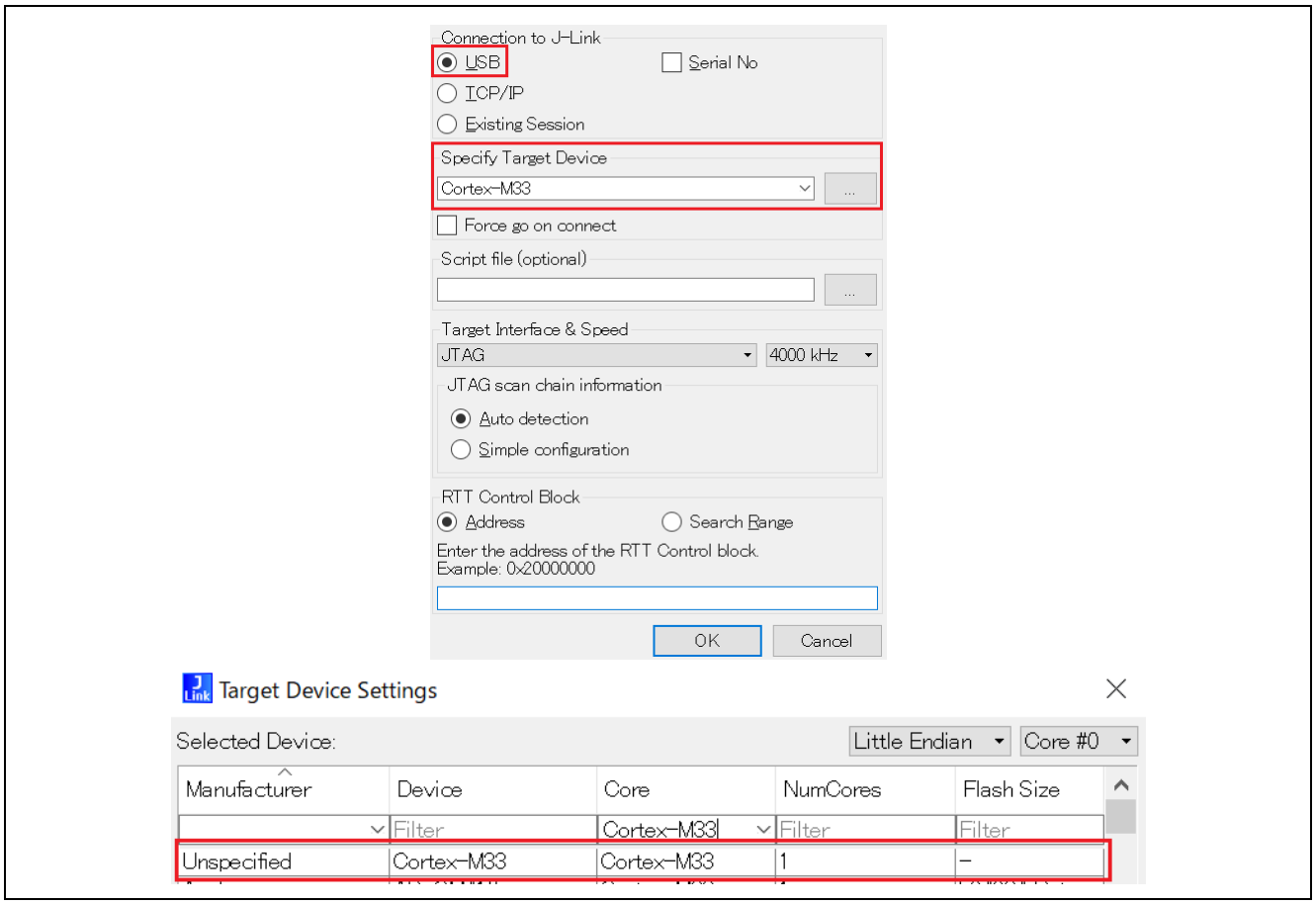


Figure 7: JLinkRTTViewer

3. (This setting is optional.) If multiple kits are connected to the PC, make sure to choose the corresponding serial number. The default is 0.

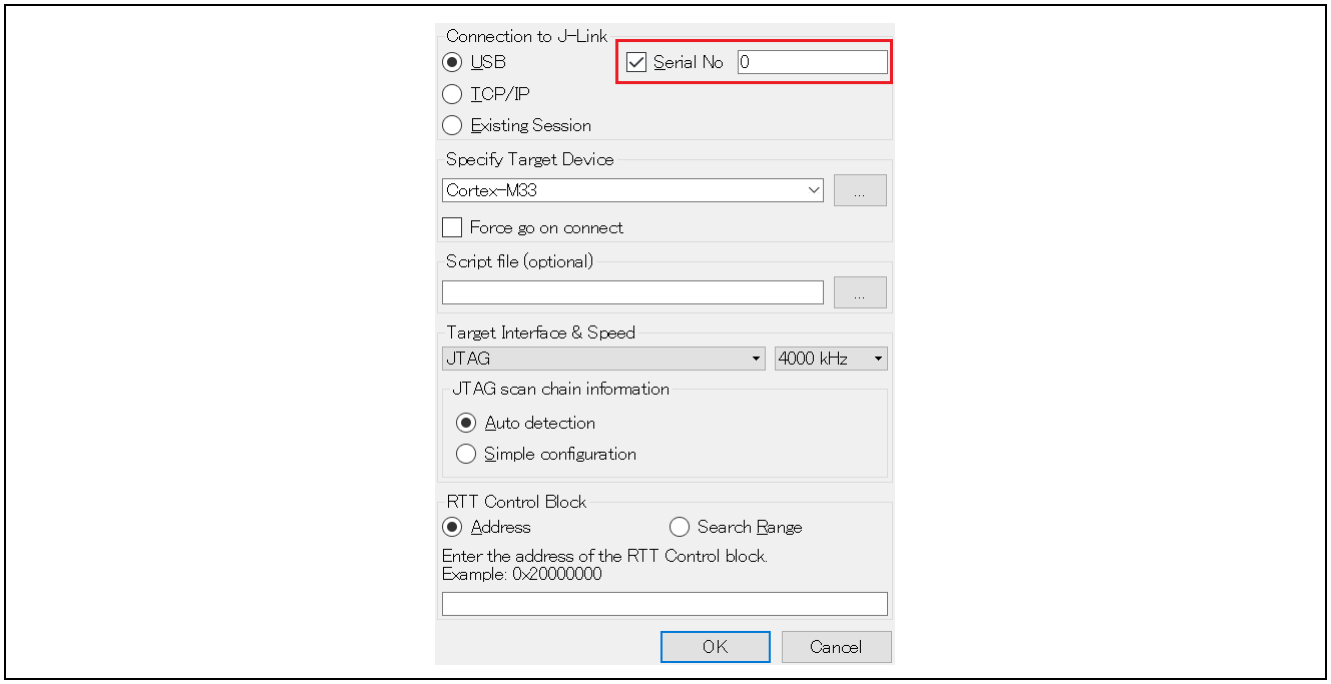


Figure 8: Setting RTT Viewer with option multiple kits connected

4. Select Target Interface & Speed.

   With respect to the RZ/G2LC, RZ/G2LC, RZ/G2UL examples, please configure as below.

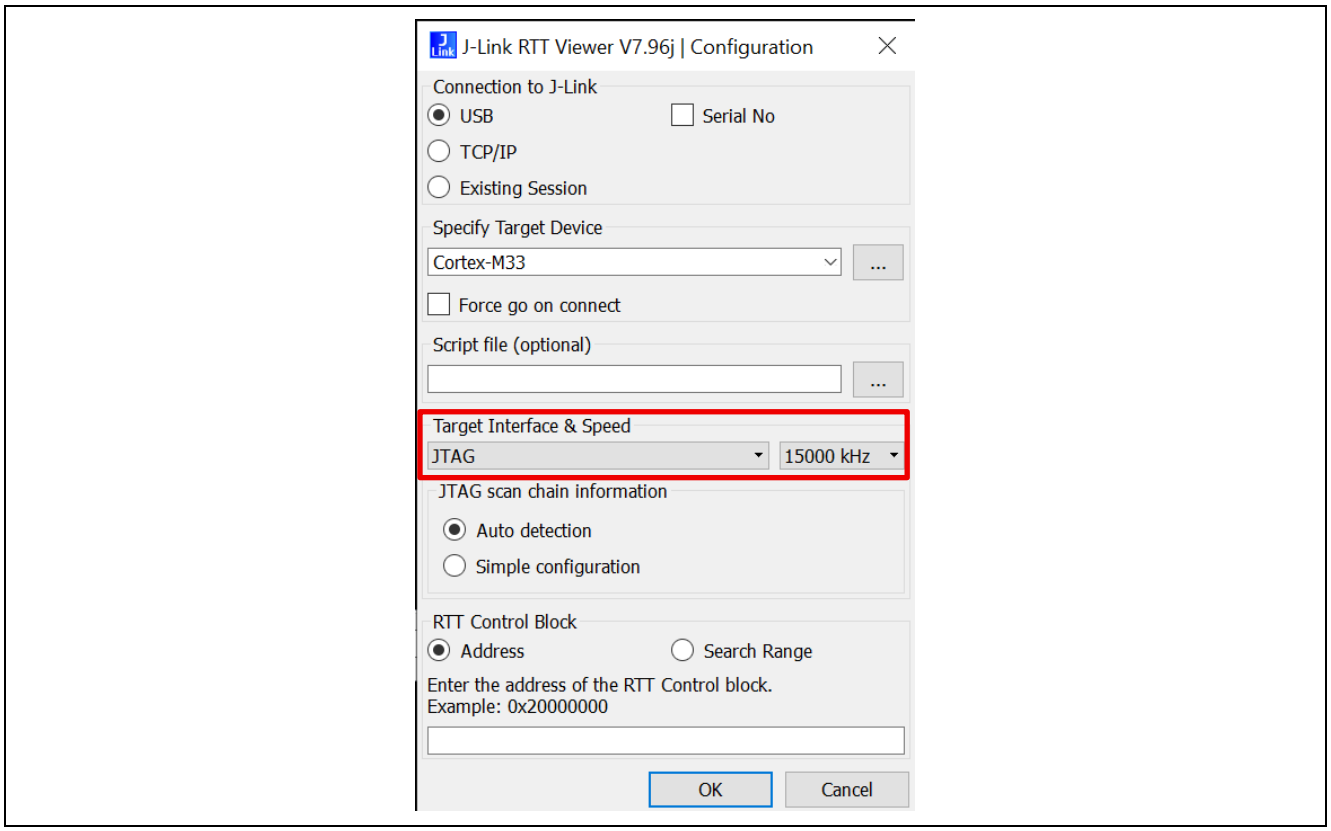


Figure 9: Setting RTT Viewer for RZ/G2 boards

With respect to the RZ/G3S and RZ/G3E examples, please configure as below.

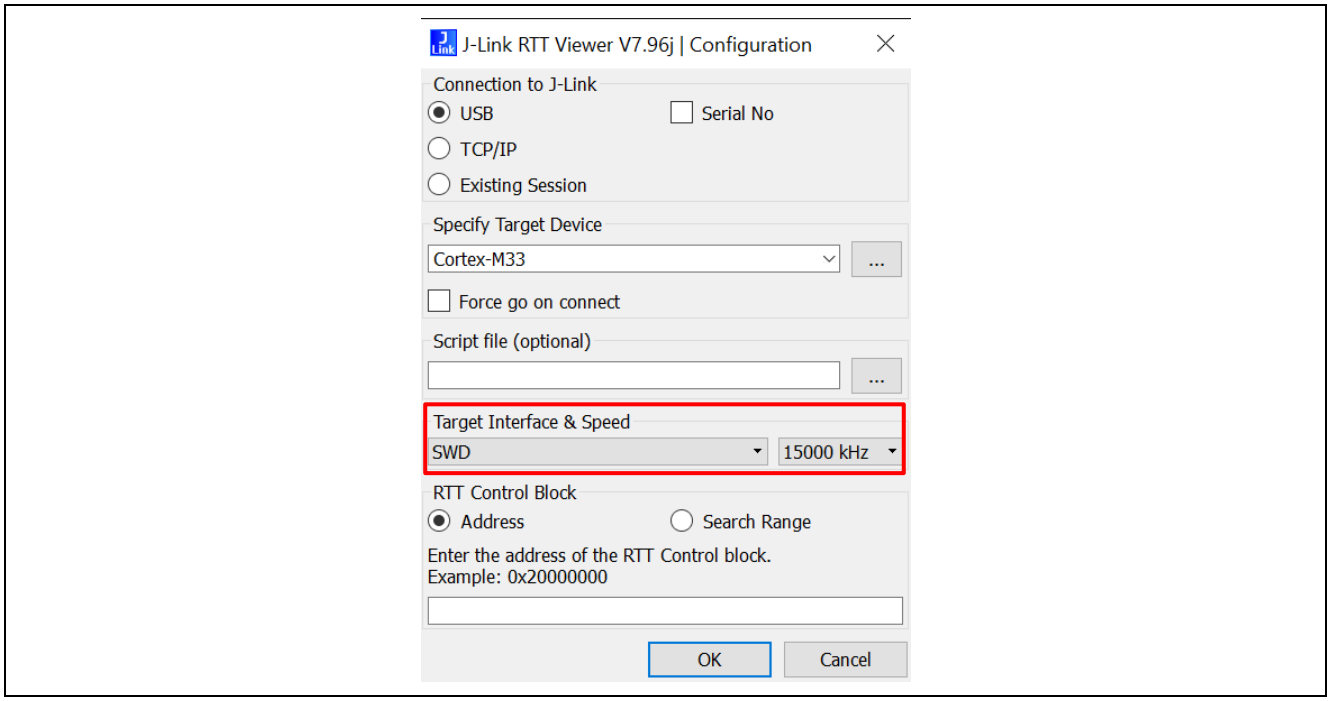


Figure 10: Setting RTT Viewer for RZ/G3 boards

5. Configurate RTT Control Block.
   1. Segger RTT block address is required to connect J-Link RTT Viewer. Search **_SEGGER_RTT** variable in the map file, generated upon successfully building a configuration of an example project, which is by default located in the address space for SDRAM.

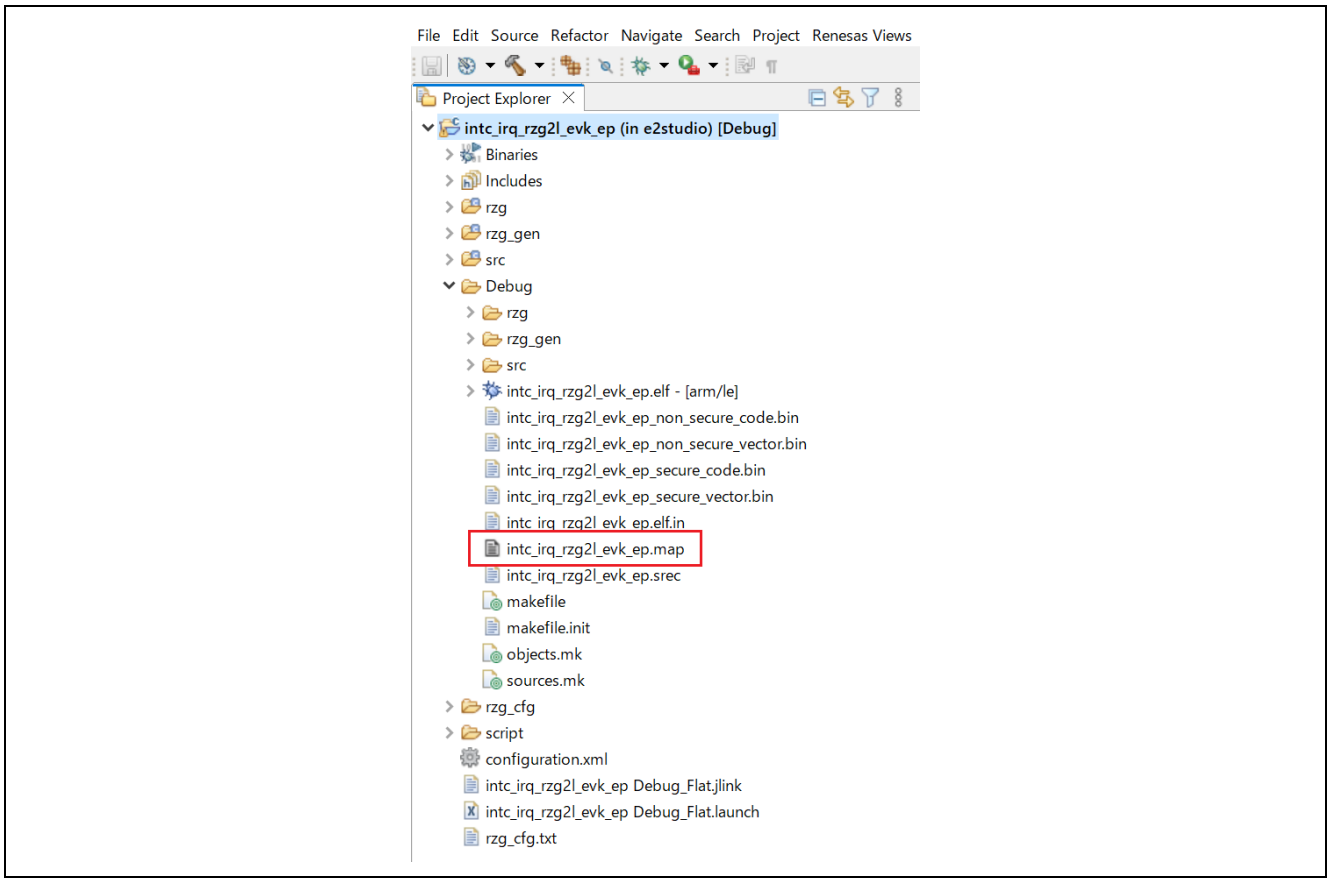


Figure 11: Search RTT block address in map file

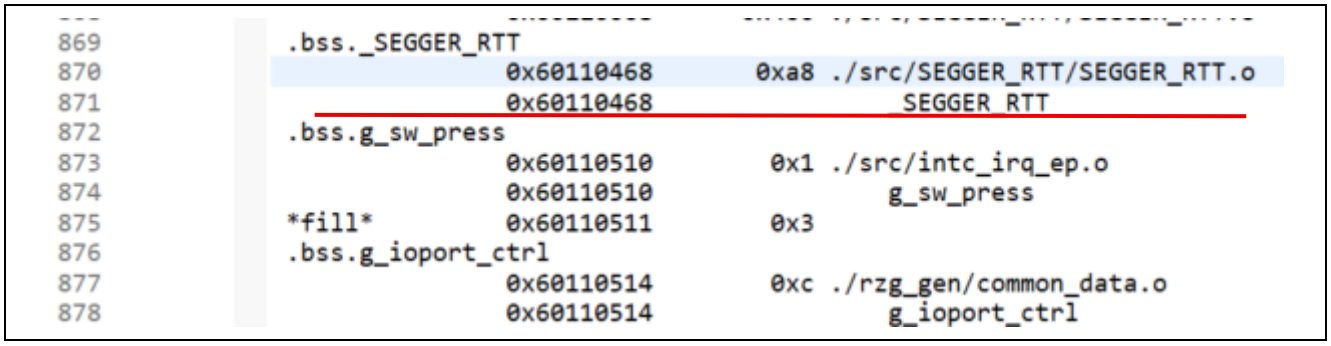


Figure 12: Get RTT block address in map file

2. Select the Address and enter the address of RTT Control Block into textbox.

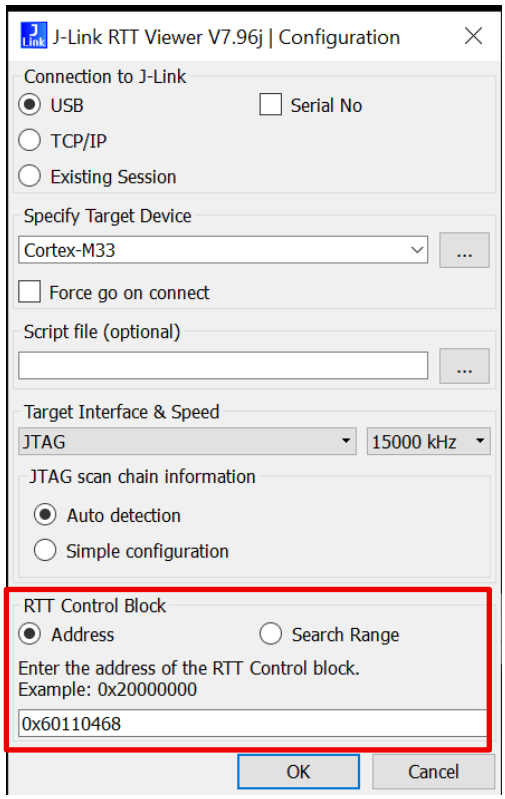


Figure 13: Enter the address of RTT Control Block into textbox

3. Click OK.
4. Click on the Input tab and change Sending option to Send on Enter. Every time input in entered, you must either press the Enter or Enter tab on the RTT viewer.

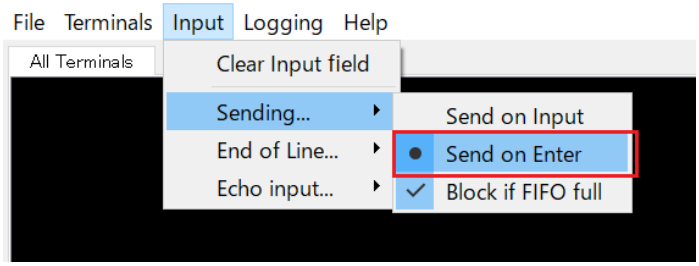


Figure 14: Sending option to Send on Enter

4.2.3 Running the project

1. In Debug mode, click **Run** > **Resume** or click on the **Play** icon twice.

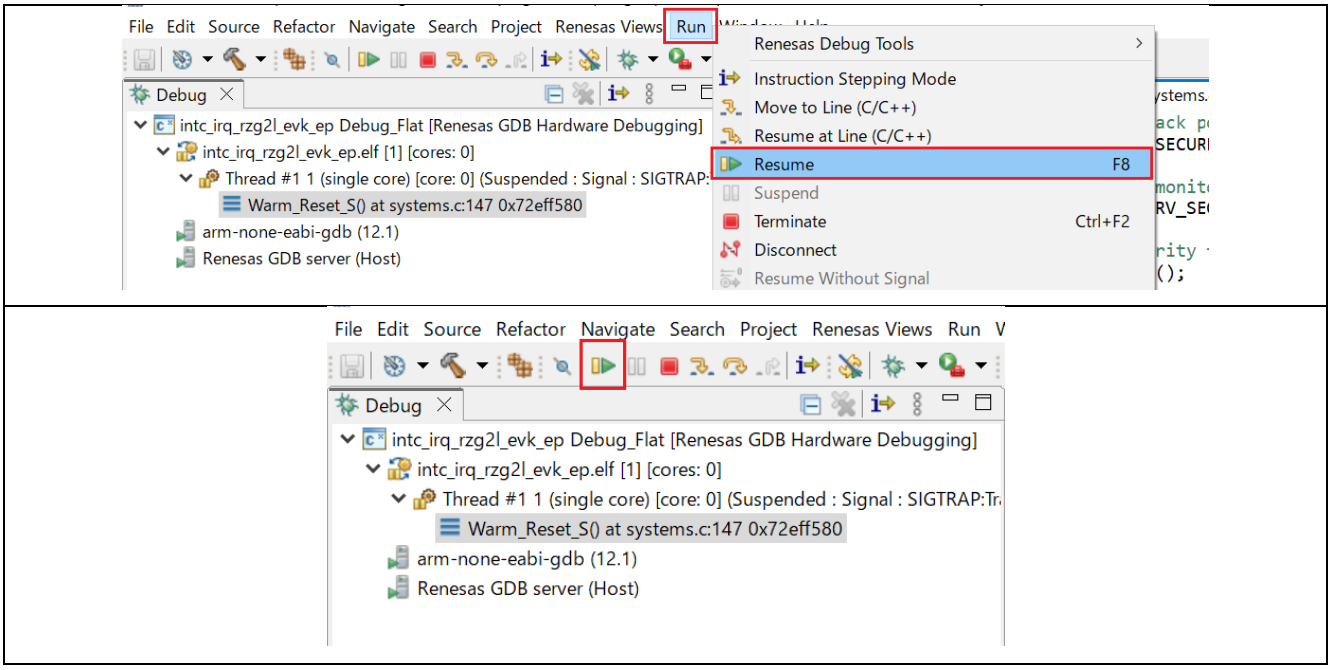


Figure 15: Run the project

2. Follow the instructions displayed on the RTT Viewer as shown below. Also refer to `readme.txt` file in the project folder to run the project.

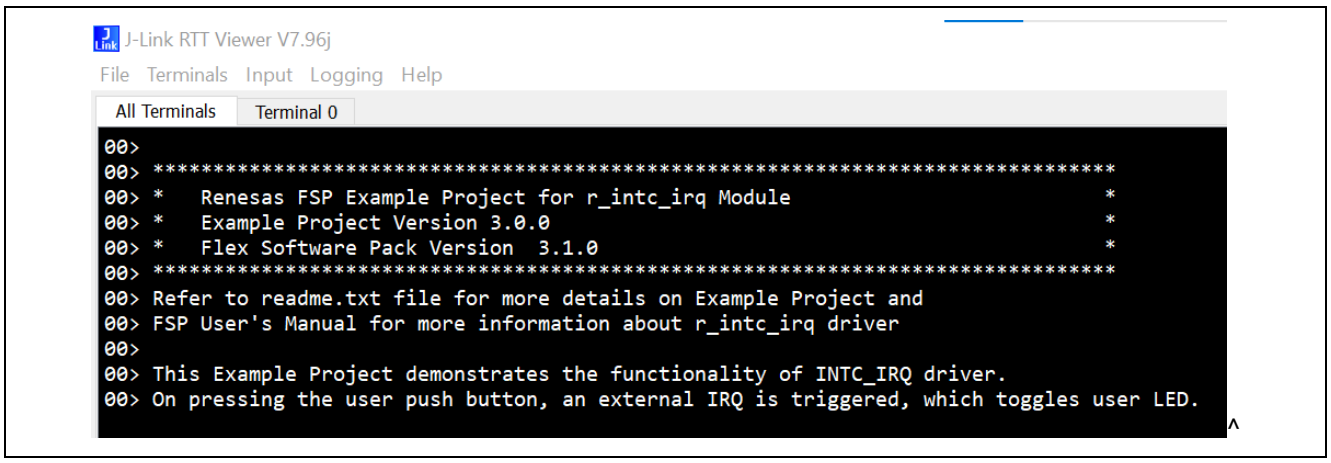


Figure 16: Follow the instructions displayed on the RTT Viewer

Note:

1. Example Projects do not support floating point or special characters or any non-numeric characters.
2. Example projects do not handle cases where the user input is greater than the expected input array size.

5. About Examples

5.1 CANFD

5.1.1 Project Overview

For RZ/G2L, RZ/G2UL and RZ/G3S:

The example project shows the operation of CAN-FD running on Renesas RZ MPUs using channel 0 and channel 1 on board. On pressing any key on the Terminal Emulator, data is transmitted from one channel to another.

On the 1st transmission, Channel 0 transmits data to Channel 1. Channel 1 displays the received data.
On the 2nd transmission, Channel 1 transmits updated data to Channel 0 as ACK. Channel 0 displays the received data. Then, Channel 0 changes CAN frame to CANFD frame and updates data.
On the 3rd transmission, Channel 0 transmits updated data to Channel 1. Channel 1 displays the received data. Then, Channel 1 changes CAN frame to CANFD frame and updates data to transmits back to Channel 0 as ACK.
On 4th transmission, Channel 1 transmits updated data to Channel 0 as ACK. Channel 0 displays the received data.

For RZ/G3E:

The example project shows the operation of CAN-FD running on Renesas RZ MPUs using channel 4 and channel 1 on board. On pressing any key on the Terminal Emulator, data is transmitted from one channel to another.

On the 1st transmission, Channel 4 transmits data to Channel 1. Channel 1 displays the received data.
On the 2nd transmission, Channel 1 transmits updated data to Channel 4 as ACK. Channel 4 displays the received data. Then, Channel 4 changes CAN frame to CANFD frame and updates data.
On the 3rd transmission, Channel 4 transmits updated data to Channel 1. Channel 1 displays the received data. Then, Channel 1 changes CAN frame to CANFD frame and updates data to transmits back to Channel 4 as ACK.
On 4th transmission, Channel 1 transmits updated data to Channel 4 as ACK. Channel 4 displays the received data.

5.1.2 Hardware Requirements

External hardware: Pmod LED

For RZ/G2L and RZ/G2UL board, make sure that IC15 (to support CAN port) is on the carrier board.
For RZ/G3S and RZ/G3E board, make sure that IC41 (to support CAN port) is on the carrier board.

5.1.3 Hardware Settings

Please connect each hardware as below.

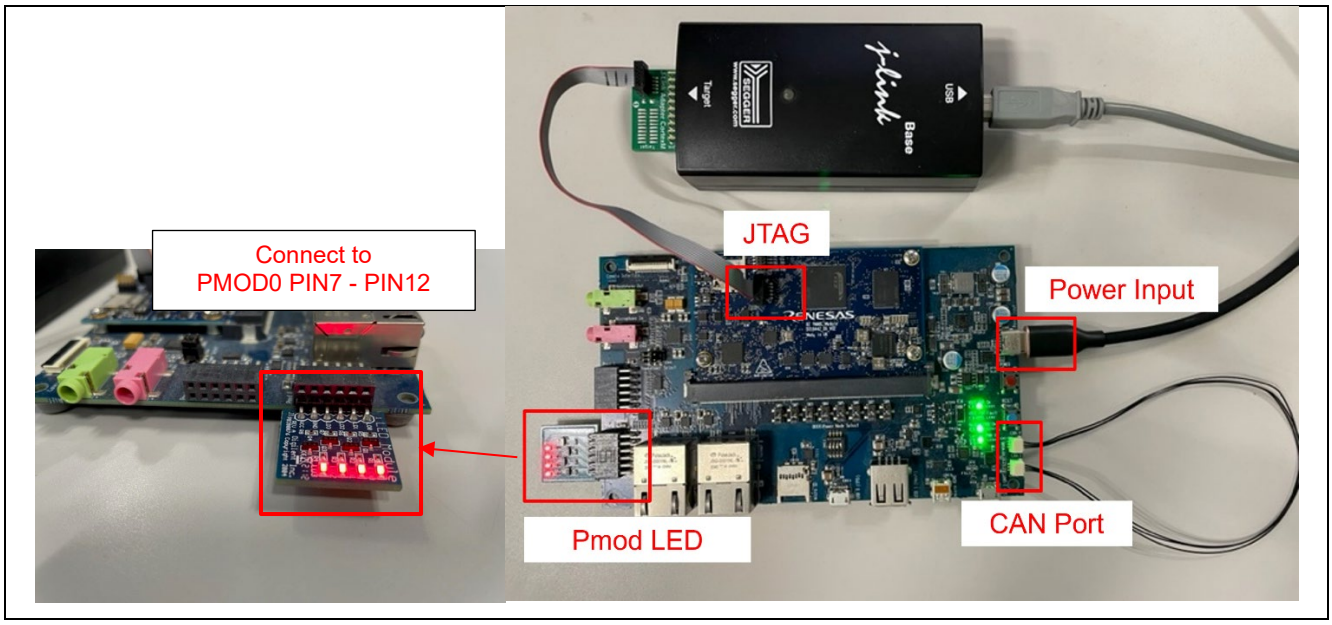


Figure 17: CANFD example - Board Setting (RZ/G2L, RZ/G2UL)

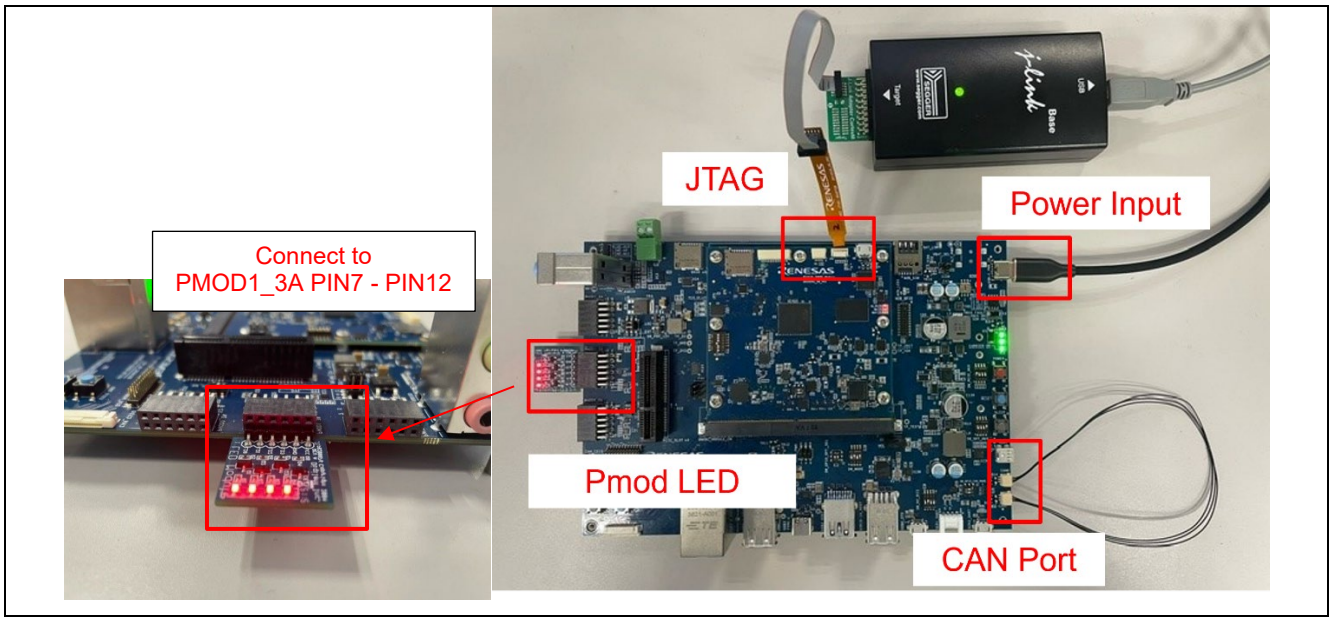


Figure 18: CANFD example - Board Setting (RZ/G3S)

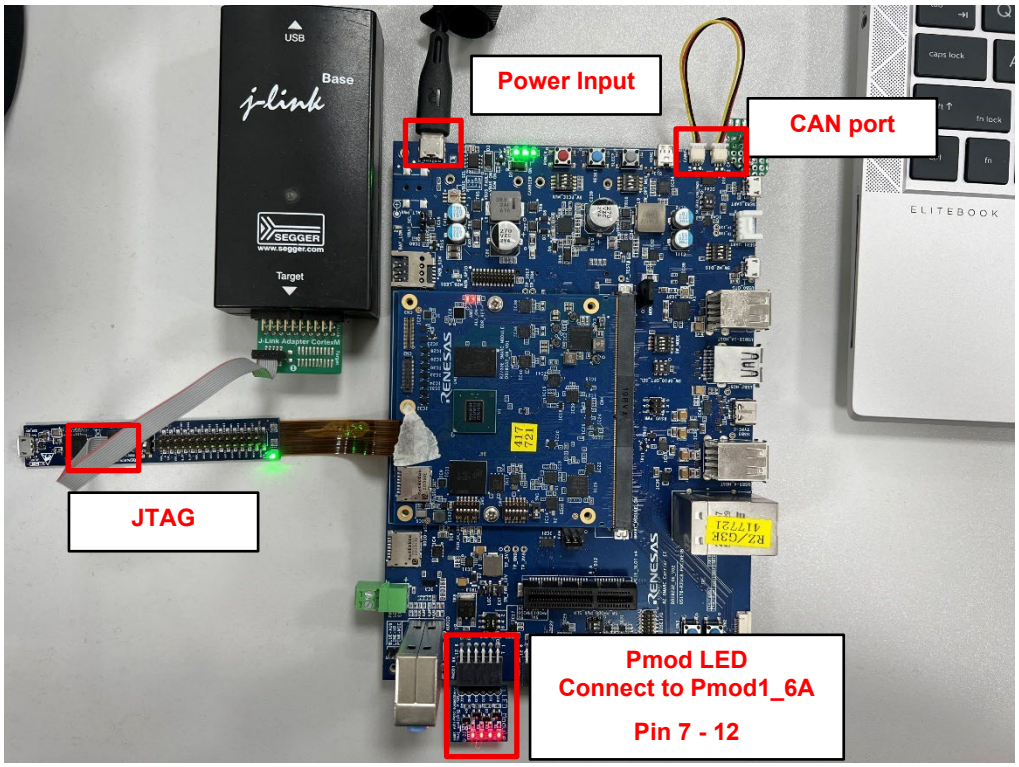


Figure 19: CANFD example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2UL EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: 1-2 SW8: 1-2 SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4		SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_M2_DIS_4: ON PMOD_PWR_SEL: Jumper 2-4 SW_GPIO_CAN_PMOD: SW1: 2-3 SW2: 5-6	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_M2_DIS_4: ON PMOD_PWR_SEL: Jumper 1-3 Jumper 2-4 SW_GPIO_CAN_PMOD: SW1: 2-3 SW2: 5-6

5.1.4 Operation

After running the example, the message below is displayed on console. After inputting any key, the transmission will be performed accordingly.

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about CAN FD driver
00>
00>  This Example Project demonstrates CAN FD operations on Renesas RZ MCUs using 1 RZ/G board.
00> On pressing any key on the RTT Viewer, data is transmitted from one channel to other.
00> On 1st transmission, Channel 0 transmits data to Channel 1. Channel 1 displays the received data.
00> On 2nd transmission, Channel 1 transmits updated data to Channel 0 as ACK. Channel 0 displays the received data,
00> then Channel 0 changes CAN frame to CANFD frame and updates data.
00> On 3rd transmission, Channel 0 transmits updated data to Channel 1. Channel 1 displays the received data,
00> then Channel 1 changes CAN frame to CANFD frame and updates data to transmits back to Channel 0 as ACK.
00> On 4th transmission, Channel 1 transmits updated data to Channel 0 as ACK. Channel 0 displays the received data.
00>
00> Please enter any key on Terminal Emulator to initiate CAN transmission.
  < h
00>
00> User input: h
00>
00>  1. Transmission of data over classic CAN Frame
00>
00> Classic CAN transmission is successful
00> Received 'TX__MESG' on classic frame.
00>  2. Responding with 'RX__MESG' using classic CAN frame
00>
00> CAN transmission after receive is successful. Sent back the ACK using classic CAN frame
00> Received Acknowledgement for Classic CAN Frame transmission.
00> CAN operation Successful. Data length = 8
00>
00>  3. Data transmission over FD frame
00>
00> CAN transmission on FD Frame after receiving classic frame ACK is successful
00> Received data over FD Frame.
00> CAN operation Successful. Data length = 16
00>
00>  4. Sending modified data over FD Frame now as acknowledgement for received FD data.
00>
00> CAN transmission on FD Frame as acknowledgement is successful
00> Received Acknowledgement for FD Frame.
00> CAN operation Successful. Data length = 16
```

Figure 20: CANFD example - Operation (RZ/G2, RZ/G3S)

```
00>
00> ********************************************************************
00> *   Renesas FSP Example Project for r_canfd Module                 *
00> *   Example Project Version 3.0.0                                  *
00> *   Flex Software Pack Version  3.1.0                              *
00> ********************************************************************
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_canfd driver
00>
00>  This Example Project demonstrates CAN FD operations on Renesas RZ MPUs using 1 RZ/G board.
00> On pressing any key on the RTT Viewer, data is transmitted from one channel to other.
00> On 1st transmission, Channel 4 transmits data to Channel 1. Channel 1 displays the received data.
00> On 2nd transmission, Channel 1 transmits updated data to Channel 4 as ACK. Channel 4 displays the received data,
00> then Channel 4 changes CAN frame to CANFD frame and updates data.
00> On 3rd transmission, Channel 4 transmits updated data to Channel 1. Channel 1 displays the received data,
00> then Channel 1 changes CAN frame to CANFD frame and updates data.
00> On 4th transmission, Channel 1 transmits updated data to Channel 4 as ACK. Channel 4 displays the received data.
00>
00> Please enter any key on Terminal Emulator to initiate CAN transmission.
  < h
00>
00> User input: h
00>
00>  1. Transmission of data over classic CAN Frame
00>
00> Classic CAN transmission is successful
00> Received 'TX__MESG' on classic frame.
00>  2. Responding with 'RX__MESG' using classic CAN frame
00>
00> CAN transmission after receive is successful. Sent back the ACK using classic CAN frame
00> Received Acknowledgement for Classic CAN Frame transmission.
00> CAN operation Successful. Data length = 8
00>
00>  3. Data transmission over FD frame
00>
00> CAN transmission on FD Frame after receiving classic frame ACK is successful
00> Received data over FD Frame.
00> CAN operation Successful. Data length = 16
00>
00>  4. Sending modified data over FD Frame now as acknowledgement for received FD data.
00>
00> CAN transmission on FD Frame as acknowledgement is successful
00> Received Acknowledgement for FD Frame.
00> CAN operation Successful. Data length = 16
00>
00>
00> Please enter any key on Terminal Emulator to initiate CAN transmission.
```

Figure 21: CANFD example - Operation (RZ/G3E)

5.2 RIIC Master

5.2.1 Project Overview

The example project demonstrates the typical use of the RIIC master HAL module APIs. The project initializes RIIC master module with fast mode and interfaces with PmodACL™ Board for ADXL345.

On powers up after establishing the connection of sensor with RZ/G board, it displays accelerometer axis data on RTTviewer. Any API/event failure will be displayed on RTTviewer.

5.2.2 Hardware Requirements

External hardware: Pmod ACL

5.2.3 Hardware Settings

Please connect each hardware as below.

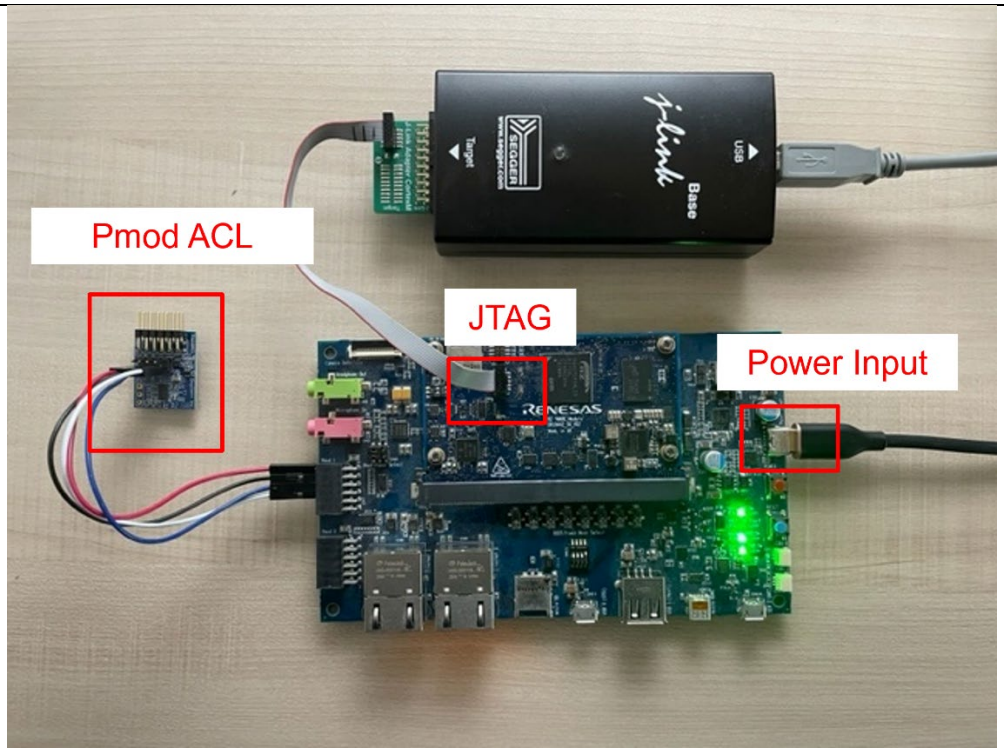


Figure 22: RIIC Master - Board Setting (RZ/G2L, RZ/G2LC, RZ/G2UL)

Please set the connection between Pmod ACL and PMOD1 on board as below.

Connection of Pmod ACL	
SCL:	Pmod ACL J2 PIN5 – PMOD1 PIN3
SDA:	Pmod ACL J2 PIN6 – PMOD1 PIN4
GND:	Pmod ACL J2 PIN7 – PMOD1 PIN5
VCC:	Pmod ACL J2 PIN8 – PMOD1 PIN6

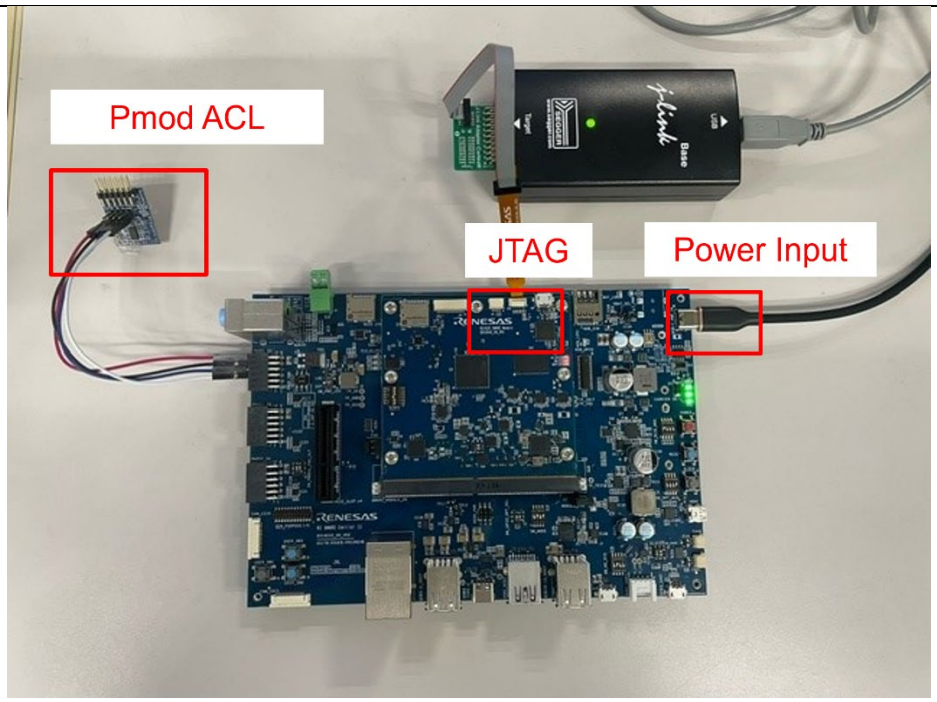


Figure 23: RIIC Master - Board Setting (RZ/G3S)

Please set the connection between Pmod ACL and PMOD1_6A on board as below.

Connection of Pmod ACL	
SCL:	Pmod ACL J2 PIN5 – PMOD1_6A PIN3
SDA:	Pmod ACL J2 PIN6 – PMOD1_6A PIN4
GND:	Pmod ACL J2 PIN7 – PMOD1_6A PIN5
VCC:	Pmod ACL J2 PIN8 – PMOD1_6A PIN6

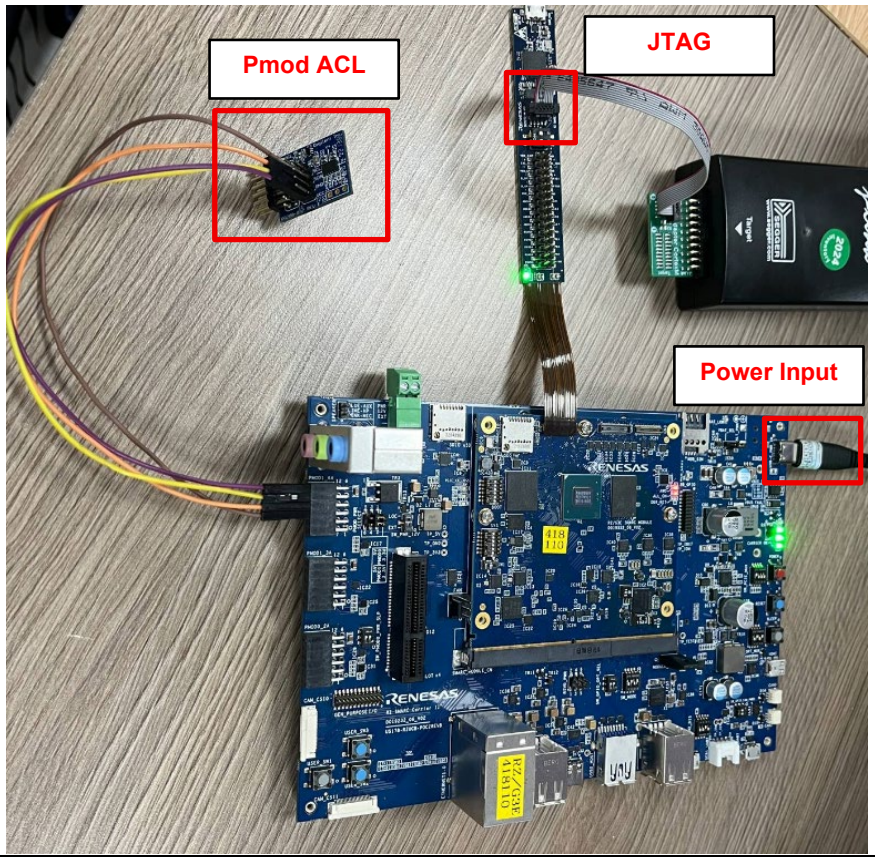


Figure 24: RIIC Master - Board Setting (RZ/G3E)

Please set the connection between Pmod ACL and PMOD1_6A on board as below.

Connection of Pmod ACL	
SCL:	Pmod ACL J2 PIN5 – PMOD1_6A PIN3
SDA:	Pmod ACL J2 PIN6 – PMOD1_6A PIN4
GND:	Pmod ACL J2 PIN7 – PMOD1_6A PIN5
VCC:	Pmod ACL J2 PIN8 – PMOD1_6A PIN6

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G2UL EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: 1-2 SW4: 1-2 SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4			SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.2.4 Operation

After running the example, the message below is displayed on console. The sensor information is shown.

```
Refer to readme.txt file for more details on Example Project and
FSP User's Manual for more information about r_riic_master driver

This project utilizes PMOD ACL sensor as iic slave device
Upon successful initialization, MPU displays sensor axis data

If SDA line is kept in LOW by any error
Please input any character on J-Link RTT Viewer to Open bus.


X-axis = 243.00, Y-axis = 4.00, Z-axis = 65488.00
X-axis = 245.00, Y-axis = 3.00, Z-axis = 65490.00
X-axis = 244.00, Y-axis = 5.00, Z-axis = 65488.00
```

Figure 25: RIIC Master Example - Operation

5.3 RSPI

5.3.1 Project Overview

The RSPI example for RZ/G2L, RZ/G2UL specification is different from RZ/G3S.

For RZ/G2L, RZ/G2UL:
The example project demonstrates the typical use of the RSPI HAL module APIs. The project configures RSPI channels (Channel 0 and Channel 1) in Master and Slave mode. Once the module is initialized and the channels are configured, Master and Slave can transmit and receive data based on commands from user sent through JLinkRTTViewer.

For RZ/G3S:
The example project demonstrates the typical use of the RSPI HAL module APIs. The project configures RSPI channel 0 in Master and Pmod SF3 in Slave. Once the module is initialized and the channels are configured, Master can write data to Slave and read it back based on commands from user sent through JLink RTT Viewer.

5.3.2 Hardware Requirements

External hardware: Pmod SF3 (RZ/G3S environment only.)

5.3.3 Hardware Settings

Please connect each hardware as below.

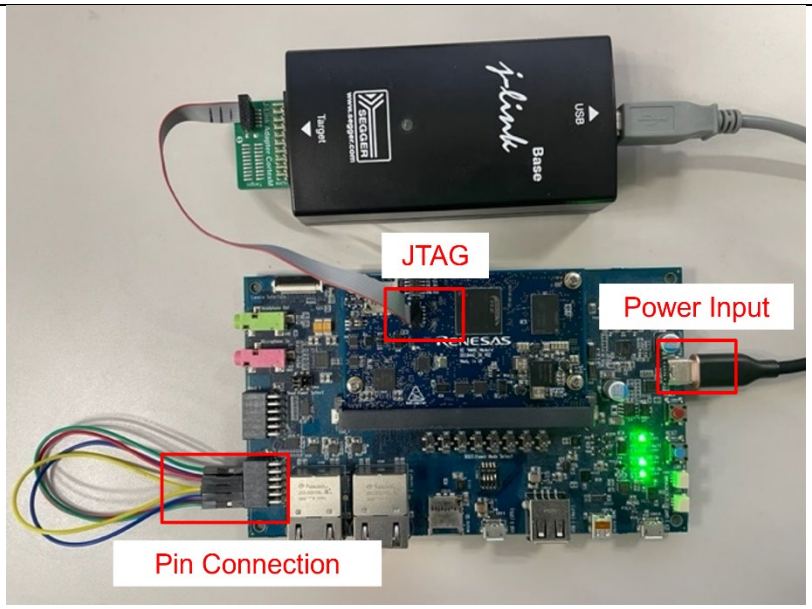


Figure 26: RSPI Example - Board Setting (RZ/G2L, RZ/G2UL)

Please set the Pin Connection as below.

RZ/G2L EVK	RZ/G2UL EVK
MISO: PMOD0 PIN3 - PMOD0 PIN8 MOSI: PMOD0 PIN2 - PMOD0 PIN10 CK: PMOD0 PIN4 - PMOD0 PIN7 SSL: PMOD0 PIN1 - PMOD0 PIN11	MISO: PMOD0 PIN3 - PMOD1 PIN8 MOSI: PMOD0 PIN2 - PMOD1 PIN7 CK: PMOD0 PIN4 - PMOD0 PIN8 SSL: PMOD0 PIN1 - PMOD0 PIN9

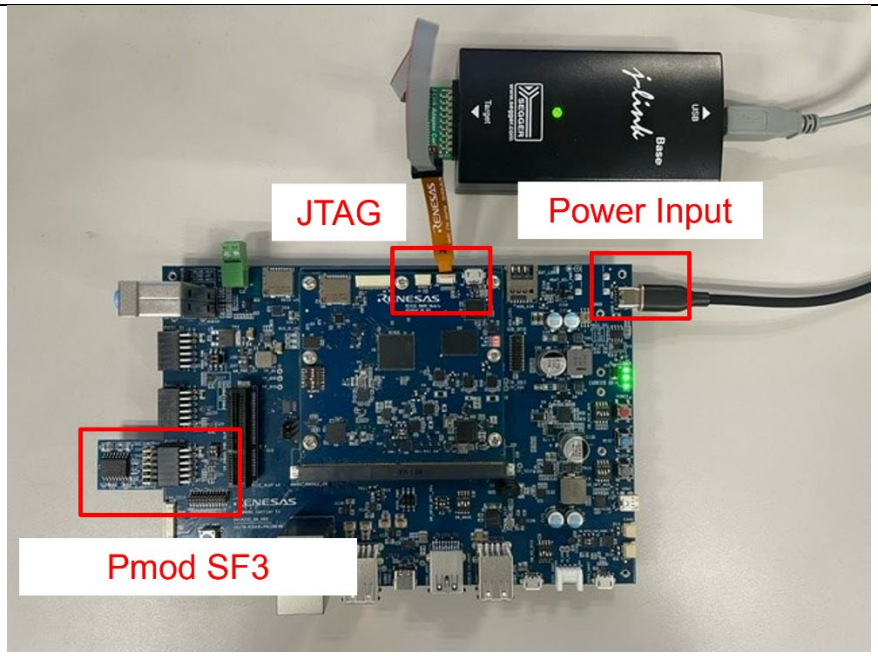


Figure 27: RSPI Example Board Setting (RZ/G3S)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2UL EVK	RZ/G3S EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: 1-2 SW6: 1-2 SW7: 1-2 SW8: 1-2 SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3

5.3.4 Operation

In the case of RZ/G2L, RZ/G2UL, after running the example, the message below is displayed on console.

```
00> ********************************************************************************
00> *   Renesas FSP Example Project for r_rspi Module                               *
00> *   Example Project Version 3.0.0                                              *
00> *   Flex Software Pack Version  3.1.0                                          *
00> ********************************************************************************
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_rspi driver
00>
00> The project initializes RSPI driver and configures RSPI channels
00> in Master and Slave mode. After initialization, master and slave
00> can transmit and receive data based on the commands from user.
00> Refer to the MPU User Manual for valid bit rates and corresponding
00> clock settings.
00>
00>  ** RSPI INIT SUCCESSFUL **
00>
00> Select from the below Menu options
00>
00> Press 1 for Write() and Read()
00> Press 2 for WriteRead()
00> Press 3 to Exit
```

Figure 28: RSPI Example – Operation (banner - RZ/G2)

Enter “1” to run Write() and Read() example and transmission will be performed after input data from Master to Slave. Then, Slave sends back same data to Master.

```
00> Press 1 for Write() and Read()
00> Press 2 for WriteRead()
00> Press 3 to Exit
  < 1
00>
00> Enter text input for Master buffer. Data size should not exceed 64 bytes.
  < aaa
00>
00> MASTER SEND: aaa
00>
00> SLAVE RECEIVE: aaa
00>
00> Master transmitted user input data to Slave
00>
00> Slave transmitted the data back to Master
00>
00> Master received data: aaa
00>
00>
00> ** SPI WRITE AND READ Demo Successful**
00>
00> Enter any other key to go back to the main menu
```

Figure 29: RSPI Example – Operation (option 1 – RZ/G2)

Enter “2” to run WriteRead() example. User can specify the data for Master and Slave. Transmission will be performed after inputting each data. Master received data and Slave received data will be shown on console once the transmission complete.

```
00> Select from the below Menu options
00>
00> Press 1 for Write() and Read()
00> Press 2 for WriteRead()
00> Press 3 to Exit
  < 2
00>
00> Enter text input for Master buffer. Data size should not exceed 64 bytes.
  < abcdefghijklmnopqrstuvwxyz012345
00>
00> Enter text input for Slave buffer. Data size should not exceed 64 bytes.
  < abcdefghijklmnopqrstuvwxyz012345
00>
00> Master buffer data transmitted to Slave
00>
00> Slave buffer data transmitted to Master
00>
00> Master received data: abcdefghijklmnopqrstuvwxyz012345
00>
00> Slave received data: abcdefghijklmnopqrstuvwxyz012345
00> ** SPI WRITE_READ Demo Successful**
00>
00> Enter any other key to go back to the main menu
```

Figure 30: RSPI Example – Operation (option 2 – RZ/G2)

In the case of RZ/G3S, after running the example, the message below is displayed on console.

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_rspi driver
00>
00> The project initializes RSPI driver and configures RSPI channel 0
00> as Master and Pmod SF3 as Slave. After initialization, master
00> can transmit and receive data based on the commands from user.
00> Refer to the MPU User Manual for valid bit rates and corresponding
00> clock settings.
00>
00>  ** RSPI INIT SUCCESSFUL **
00>
00> Select from the below Menu options
00>
00> Press 1 for Write() to Pmod SF3
00> Press 2 for Read() from Pmod SF3
00> Press 3 to Exit
```

Figure 31: RSPI Example – Operation (banner - RZ/G3S)

Enter “1” to run Write() to Pmod SF3 example and transmission will be performed after input data from Master to Slave.

```
Select from the below Menu options

Press 1 for Write() to Pmod SF3
Press 2 for Read() from Pmod SF3
Press 3 to Exit
1

Enter text input to write into Pmod SF3. Data size should not exceed 256 bytes.
Hello from Renesas!

Data write to Pmod SF3: Hello from Renesas!

** RSPI WRITE to Pmod SF3 Demo Successful**

Enter any other key to go back to the main menu
```

Figure 32: RSPI Example – Operation (option 1 - RZ/G3S)

Enter “2” to run Read() from Pmod SF3 example. Reading data from Pmod SF3 will be performed and received data will be shown on console once the transmission complete.

```
Select from the below Menu options

Press 1 for Write() to Pmod SF3
Press 2 for Read() from Pmod SF3
Press 3 to Exit
2

Data read from Pmod SF3: Hello from Renesas!

** RSPI READ from Pmod SF3 Demo Successful**
```

Figure 33: RSPI Example – Operation (option 2 - RZ/G3S)

5.4 SCIF

5.4.1 Project Overview

The example project demonstrates the typical use of the UART HAL module APIs. The project initializes the UART with Baud rate of 115200 bps and GTM module. Using a Terminal Program (like Tera Term) user can provide a value & press enter key to set the period of the Pmod LED signal.

The range of input values are displayed on the JLinkRTTViewer. Any failure will also be displayed using JLinkRTTViewer. To see user input values on Serial terminal, enable local echo option.

5.4.2 Hardware Requirements

External hardware:

RZ/G2L, RZ/G2lC, RZ/G3S: Pmod USBUART and Pmod LED.

RZ/G3E: Pmod LED.

5.4.3 Hardware Settings

Please connect each hardware as below.

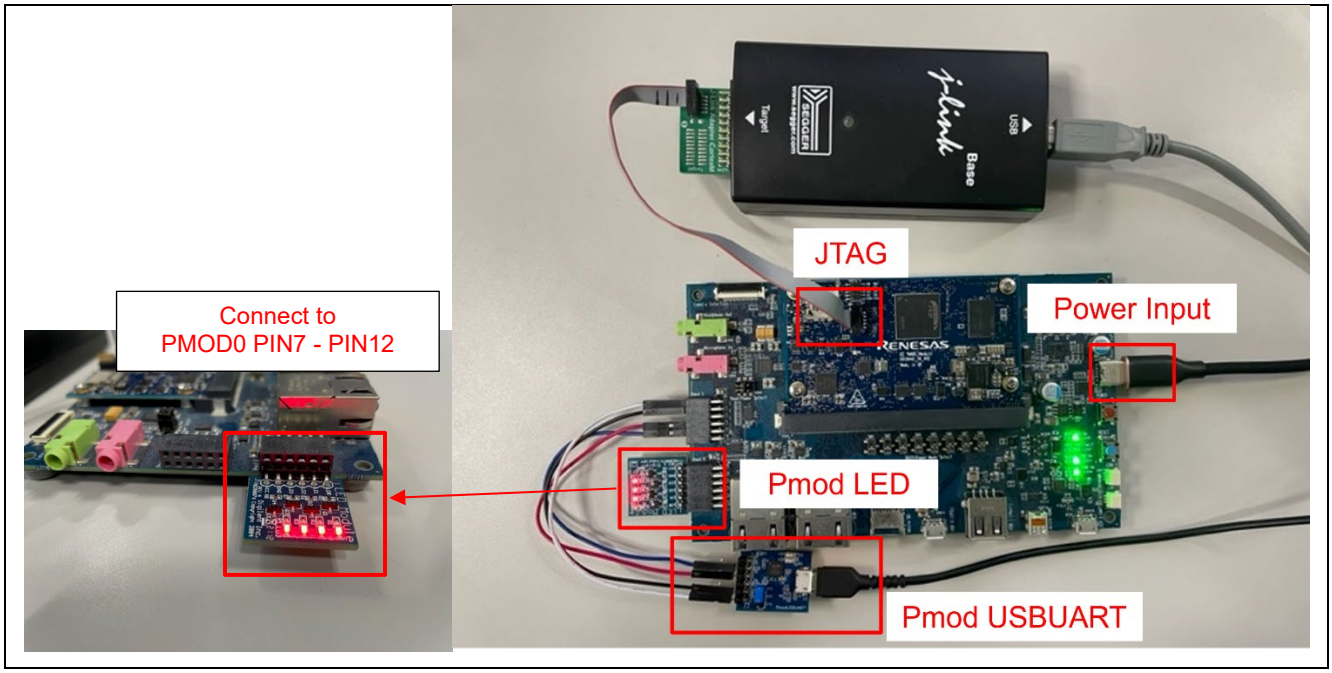


Figure 34: SCIF UART Example - Board Setting (RZ/G2L, RZ/G2LC)

Please set the connection between Pmod USBUART and PMOD1 on the board below.

Connection of Pmod USBUART	
RXD:	Pmod USBUART J2 PIN2 – PMOD1 PIN2
TXD:	Pmod USBUART J2 PIN3 – PMOD1 PIN3
GND:	Pmod USBUART J2 PIN5 – PMOD1 PIN5
VCC:	Pmod USBUART J2 PIN6 – PMOD1 PIN6

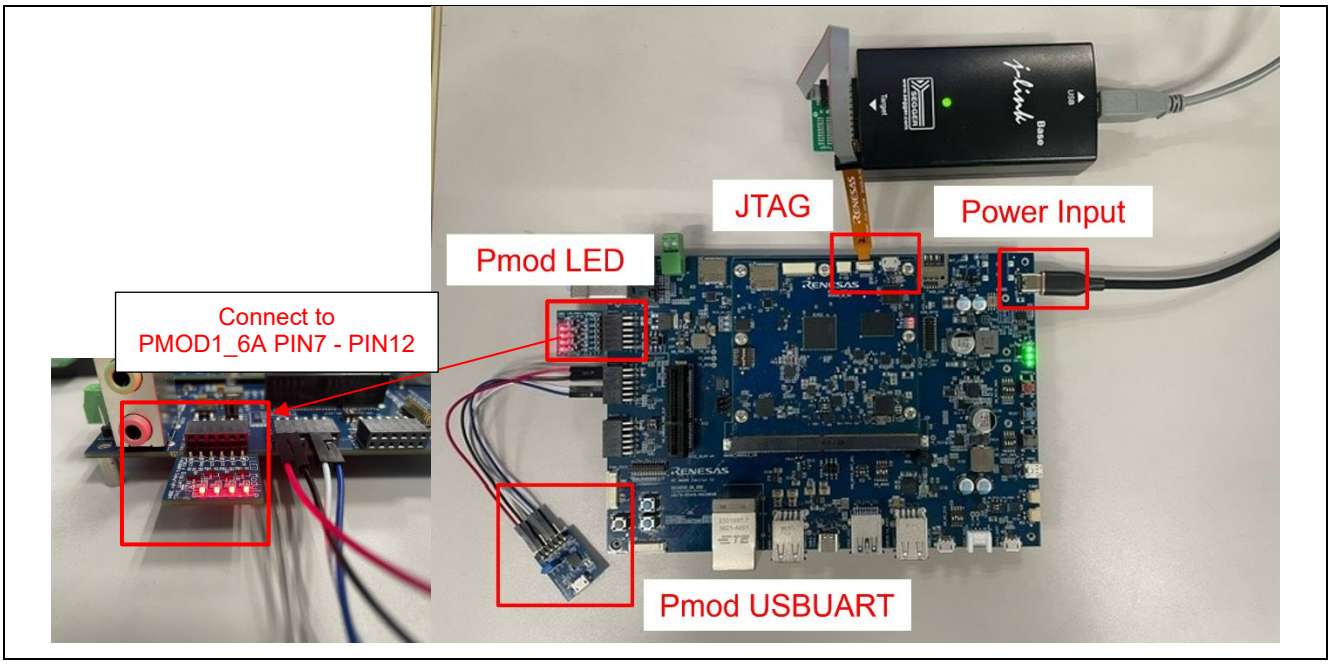


Figure 35: SCIF UART Example - Board Setting (RZ/G3S)

Please set the connection between Pmod USBUART and PMOD1_3A on the board below.

Connection of Pmod USBUART	
RXD:	Pmod USBUART J2 PIN2 – PMOD1_3A PIN2
TXD:	Pmod USBUART J2 PIN3 – PMOD1_3A PIN3
GND:	Pmod USBUART J2 PIN5 – PMOD1_3A PIN5
VCC:	Pmod USBUART J2 PIN6 – PMOD1_3A PIN6

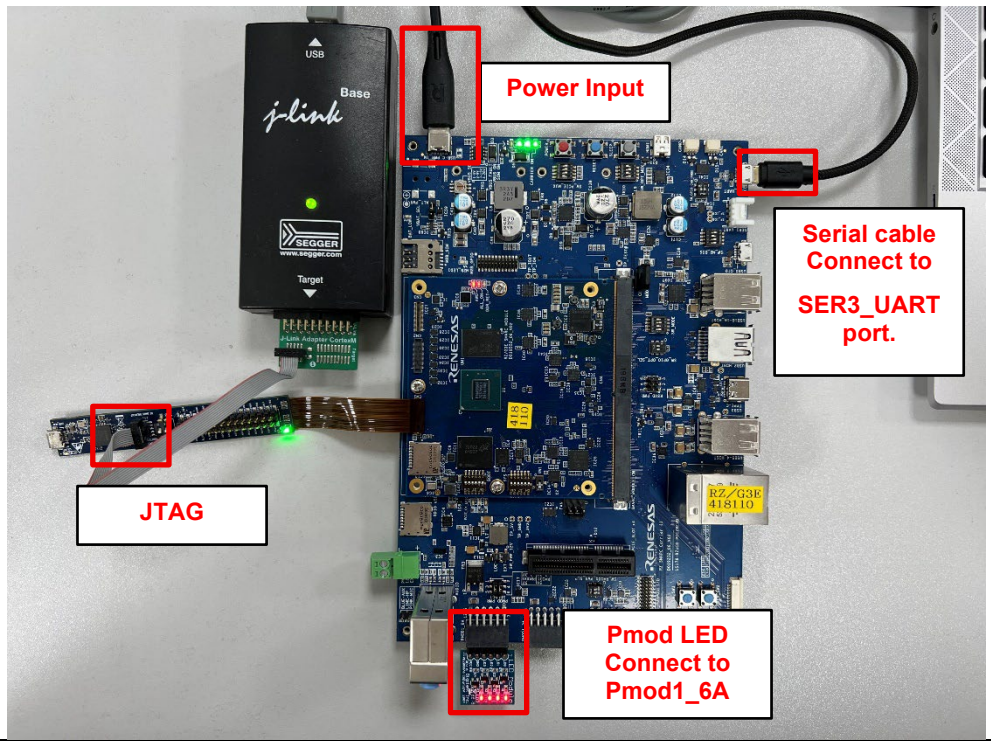


Figure 36: SCIF UART Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: ON SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: 2-3 SW3: 2-3 SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4		SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_M2_DIS_4: ON PMOD_PWR_SEL: Jumper connects 2-4 SW_OPT_MUX_4: ON	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.4.4 Operation

After running the example, the message below is displayed on console.

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_scif_uart driver
00>
00>
00> The project initializes the UART with baud rate of 115200 bps.
00> Open Serial Terminal with this baud rate value and
00> Provide Input ranging from 1 - 2000 to set LED Intensity
```

Figure 37: SCIF UART Example – Operation (console)

User can specify a value for LED blinking on Terminal Program.

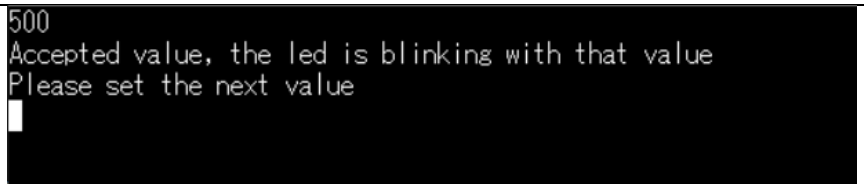


Figure 38: SCIF UART Example – Operation (specify a value)

The configuration of Terminal Program is as below.

- Speed: 115200bps
- Data: 8bit
- Parity: None
- Stop bits: 1bit
- Flow control: None

Also, enable local echo option to see user input values on Serial terminal.

5.5 GPT (Input Capture)

5.5.1 Project Overview

The Example Project demonstrates the functionality of GPT Input capture module. GPT4 is used to generate periodic pulses of 500msec duration and provided as input to GPT3 used as Input capture.

GPT3 counts the event pulse received at its input. Based on the period and capture event, the time period of pulse is calculated and displayed on RTTViewer.

5.5.2 Hardware Requirements

External hardware: None

5.5.3 Hardware Settings

Please connect each hardware as below.

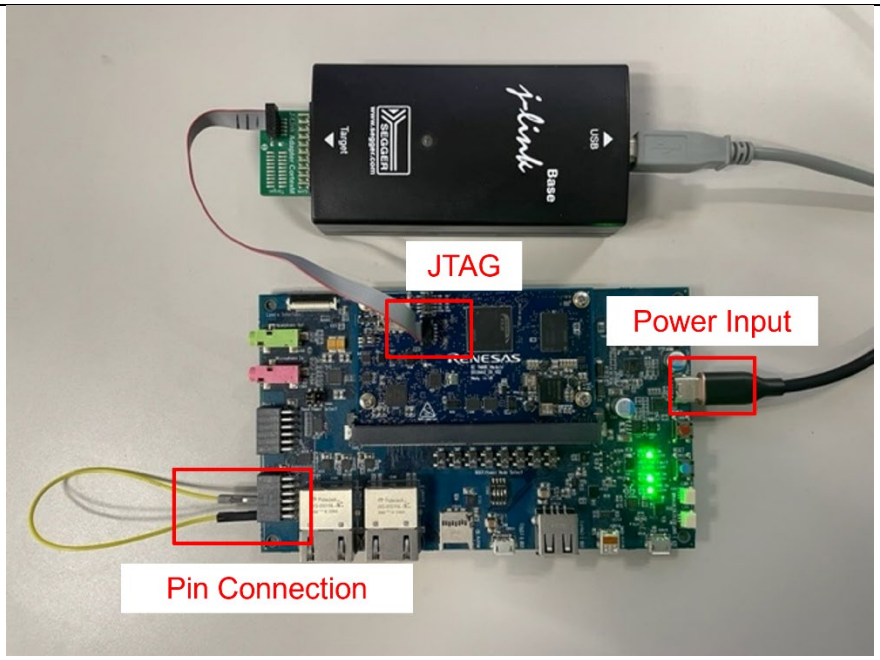


Figure 39: GPT Example - Board Setting (RZ/G2L, RZ/G2LC)

Please set the Pin Connection as below.

RZ/G2L EVK	RZ/G2LC EVK
PMOD0 PIN7 - PMOD0 PIN9	PMOD0 PIN7 – PMOD1 PIN1

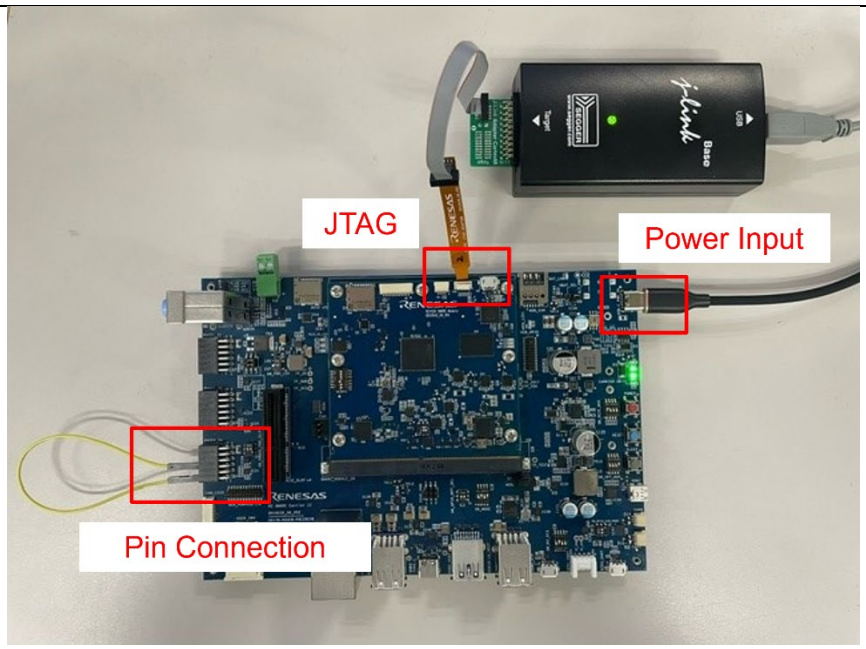


Figure 34: GPT Input Cpture Example - Board Setting (RZ/G3S)

Please set the Pin Connection as below.

RZ/G3S EVK
PMOD0_2A PIN7 - PMOD0_2A PIN9

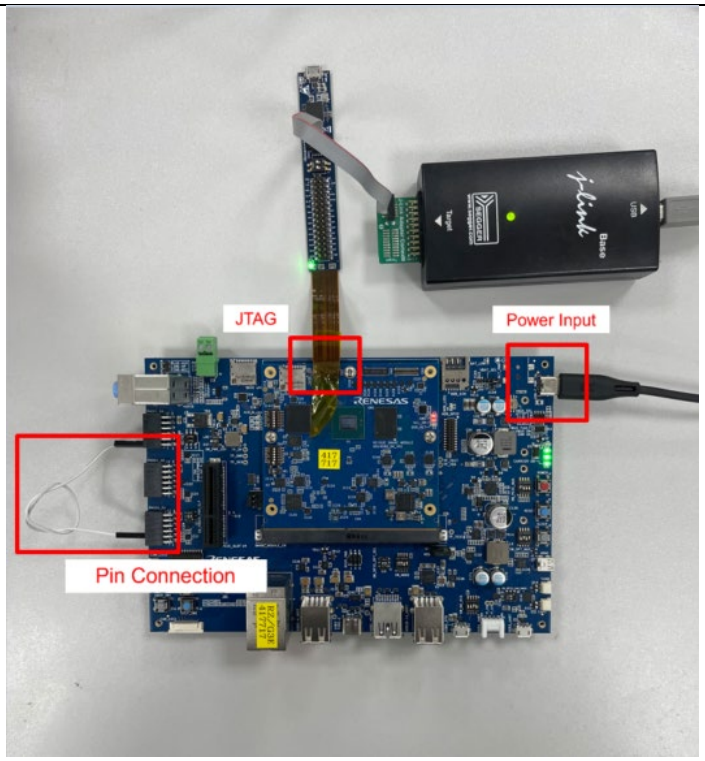


Figure 35: GPT Input Capture Example - Board Setting (RZ/G3E)

Please set the Pin Connection as below.

RZ/G3E EVK
PMOD1_6A PIN7 - PMOD0_2A PIN8

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4	SW1: 2-3 SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_PMOD0_PWR_SLP_1: Short 3-2 PMOD_PWR_SELECT: Jumper connects 1-3	SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_GPIO_CAN_PMOD: Short 3-2 PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4

5.5.4 Operation

After running the example, the message below is displayed on console.

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_gpt driver
00>
00> The EP demonstrates the functionality of GPT Input capture module.
00> GPT4 is used to generate periodic pulses of 500msec duration and
00> provided as input to GPT Input capture(GPT3).GPT3 counts the event
00> pulse received at its input. Based on the period and capture event,
00> the time period of pulse is calculated and displayed on RTTViewer.
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
00> Pulse width measurement value(in second) - 0.25000
00>
```

Figure 36: GPT Input Capture Example - Operation

5.6 GPT (PWM)

5.6.1 Project Overview

The example project demonstrates typical use of GPT HAL module APIs. Users have the provision to input value as per displayed menu through JLinkRTTViewer to select different GPT supported modes (Periodic, PWM, One-Shot).

In periodic mode, the user can enter the period within the permitted ranges to change the frequency of the user Pmod LED. In PWM mode, the user can enter the duty cycle within the specified range to adjust the intensity of the user Pmod LED.In One-Shot mode, output will be displayed on JlinkRTTViewer.

Any failure will also be displayed on JlinkRTTViewer.

5.6.2 Hardware Requirements

External hardware: Pmod LED

5.6.3 Hardware Settings

Please connect each hardware as below.

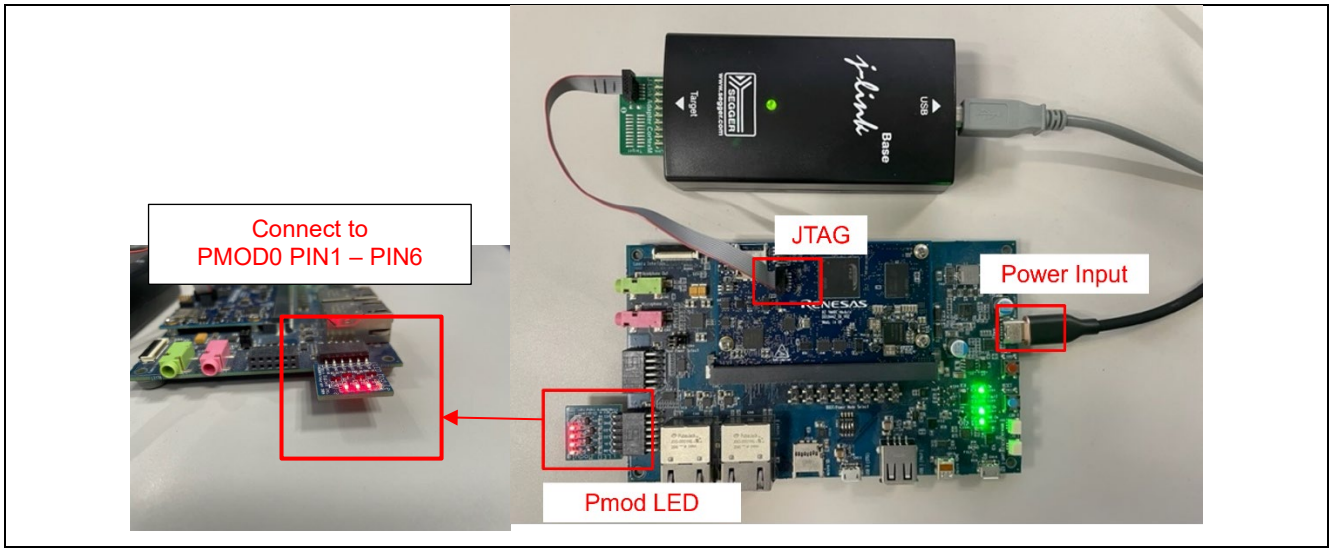


Figure 37: GPT Example - Board Setting (RZ/G2L, RZ/G2LC)

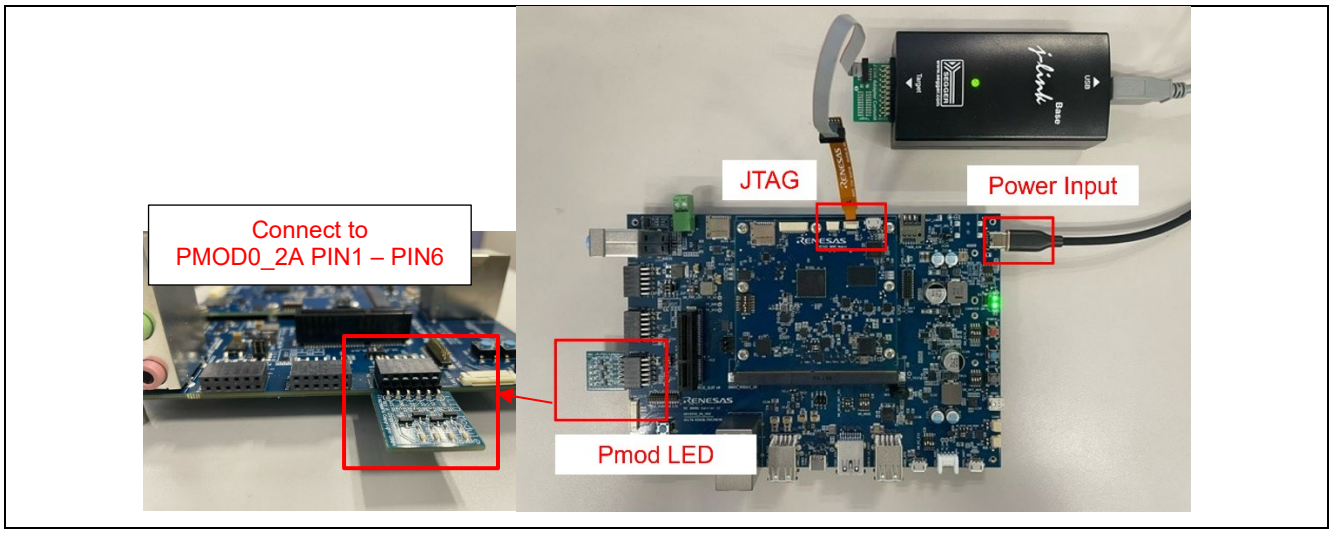


Figure 38: GPT Example - Board Setting (RZ/G3S)

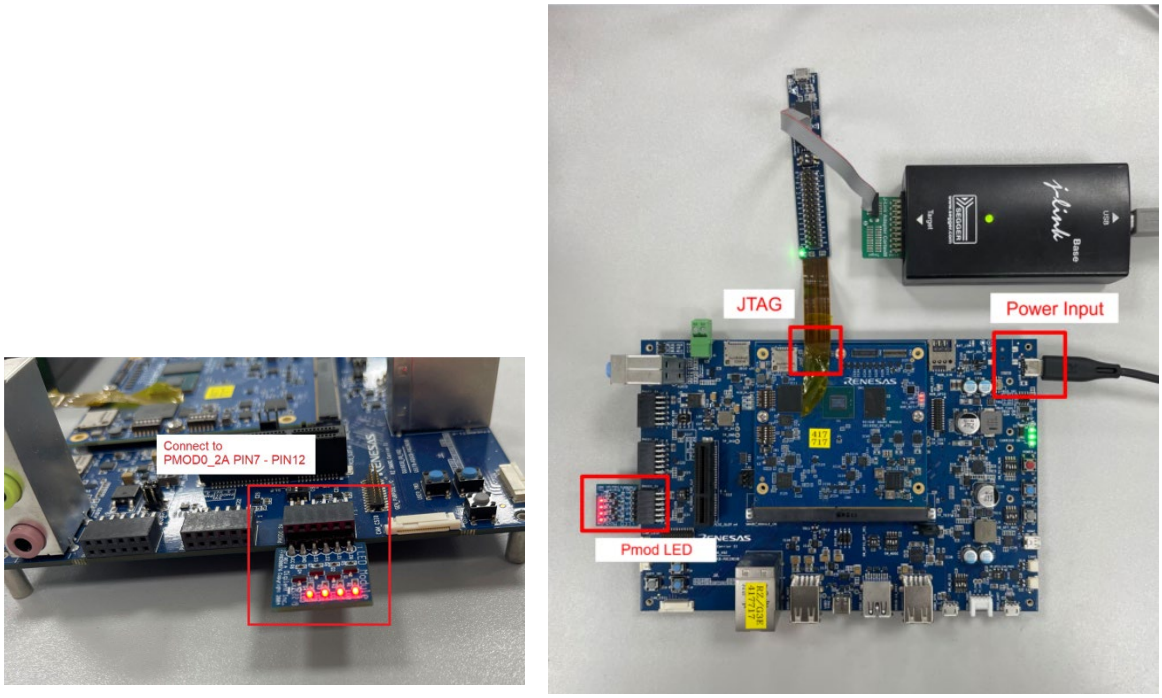


Figure 39: GPT Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G3S EVK	RZ/G3E
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: ON SW1-5: Don't care. SW1-6: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4		SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3

5.6.4 Operation

After running the example, the message below is displayed on console.

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_gpt driver
00>
00> The project initializes GPT module in Periodic, PWM or One-shot mode based on user input from the displayed menu options.
00> In periodic mode, user can enter the time period within the permitted ranges to change the frequency of the user LED.
00> In PWM mode, user can enter the duty cycle within the specified range to adjust the intensity of the user LED.
00> In ONE SHOT mode, Output will be displayed on JlinkRTTViewer when timer expires.
00>
00> Menu Options
00> 1. Enter 1 for Periodic mode
00> 2. Enter 2 for PWM mode
00> 3. Enter 3 for one shot mode
00> User Input:
```

Figure 40: GPT Example – Operation (Console)

In periodic mode, the user can enter period of the timer, then LED starts blink.

```
00> Menu Options
00> 1. Enter 1 for Periodic mode
00> 2. Enter 2 for PWM mode
00> 3. Enter 3 for one shot mode
00> User Input:
  < 1
00>
00>  Enter the desired period in millisecond
  < 500
00> Opened Timer in Periodic Mode
00> Started Timer in Periodic Mode
```

Figure 41: GPT Example – Operation (option 1)

In periodic mode, the user can enter the duty cycle, then LED starts blink.

```
00> Menu Options
00> 1. Enter 1 for Periodic mode
00> 2. Enter 2 for PWM mode
00> 3. Enter 3 for one shot mode
00> User Input:
  < 2
00> Periodic Timer Instance Closed, Start PWM Timer Instance
00>
00> Enter The Desired Duty Cycle in Percentage of Range 0 - 100:
  < 30
00> Opened Timer in PWM Mode
00> Started Timer in PWM Mode
00>
```

Figure 42: GPT Example – Operation (option 2)

In One-Shot mode, One-shot timer is started after selecting this mode.

```
00> Menu Options
00> 1. Enter 1 for Periodic mode
00> 2. Enter 2 for PWM mode
00> 3. Enter 3 for one shot mode
00> User Input:
  < 3
00> PWM Timer Instance Closed, Start ONE-SHOT Timer Instance
00> Opened Timer in ONE-SHOT Mode
00> Started Timer in ONE-SHOT Mode
00>
00>  Timer Expired in One-Shot Mode
00>
```

Figure 43: GPT Example – Operation (option 3)

5.7 GTM

5.7.1 Project Overview

This Example Project demonstrates the functionality of GTM in periodic mode and one-shot mode. On providing any input on the RTTviewer, GTM channel 2 starts in one-shot mode. GTM channel 1 starts in periodic mode when GTM channel 2 expires. Timer in periodic mode expires periodically at a time period specified by user and toggles the LED0 on Pmod LED.

5.7.2 Hardware Requirements

External hardware: Pmod LED

5.7.3 Hardware Settings

Please connect each hardware as below.

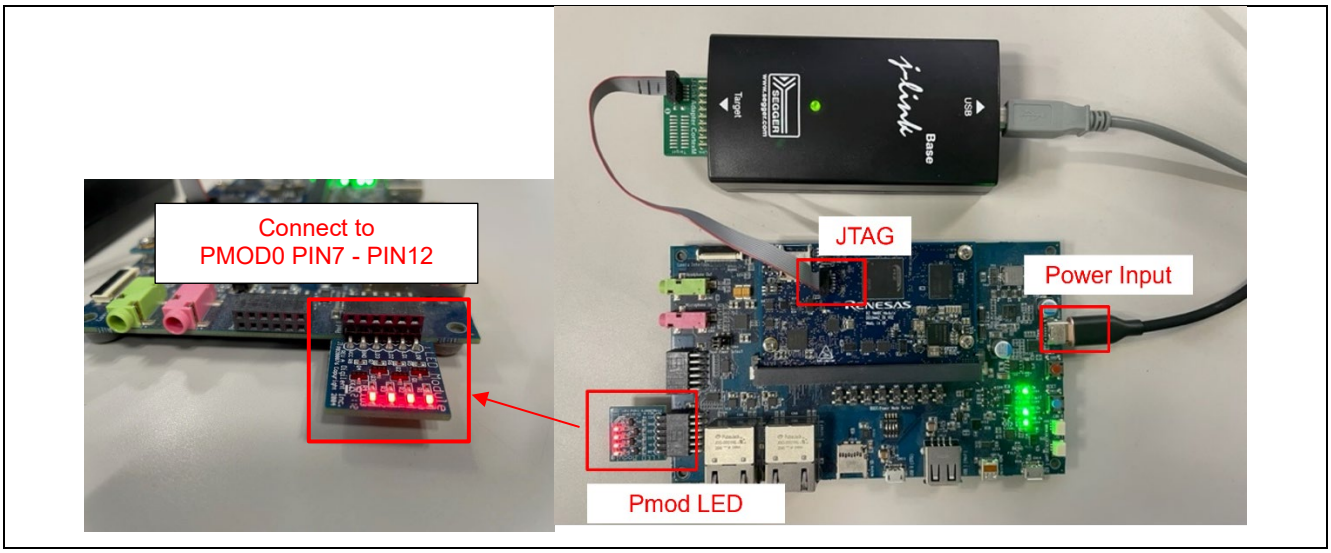


Figure 44: GTM Example - Board Setting (RZ/G2L, RZ/G2LC, RZ/G2UL)

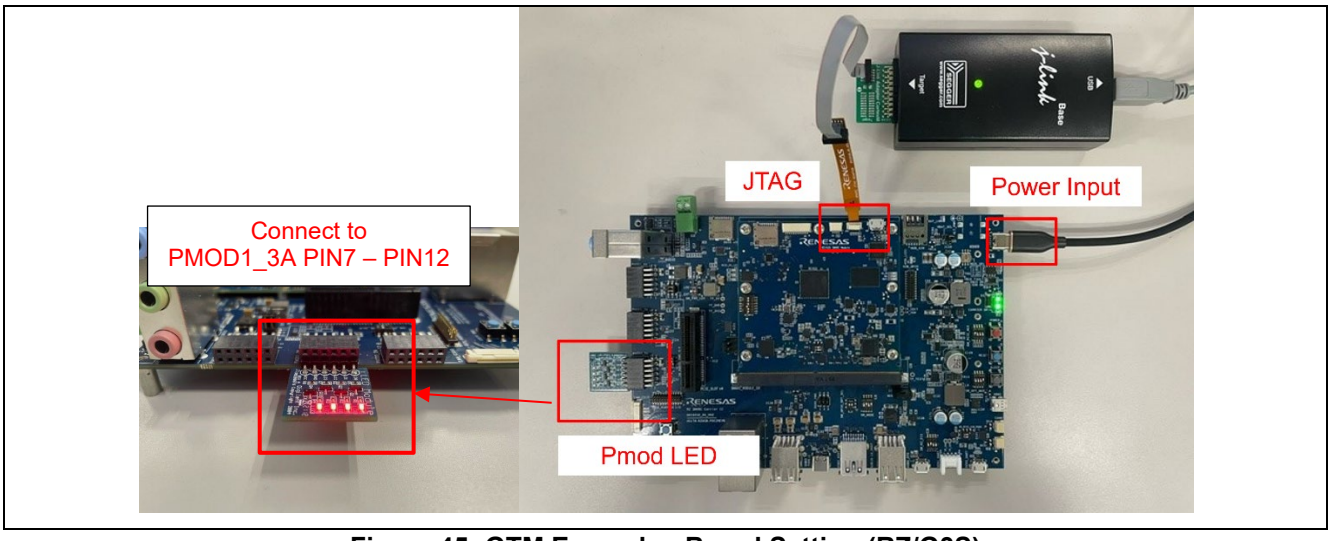


Figure 45: GTM Example - Board Setting (RZ/G3S)

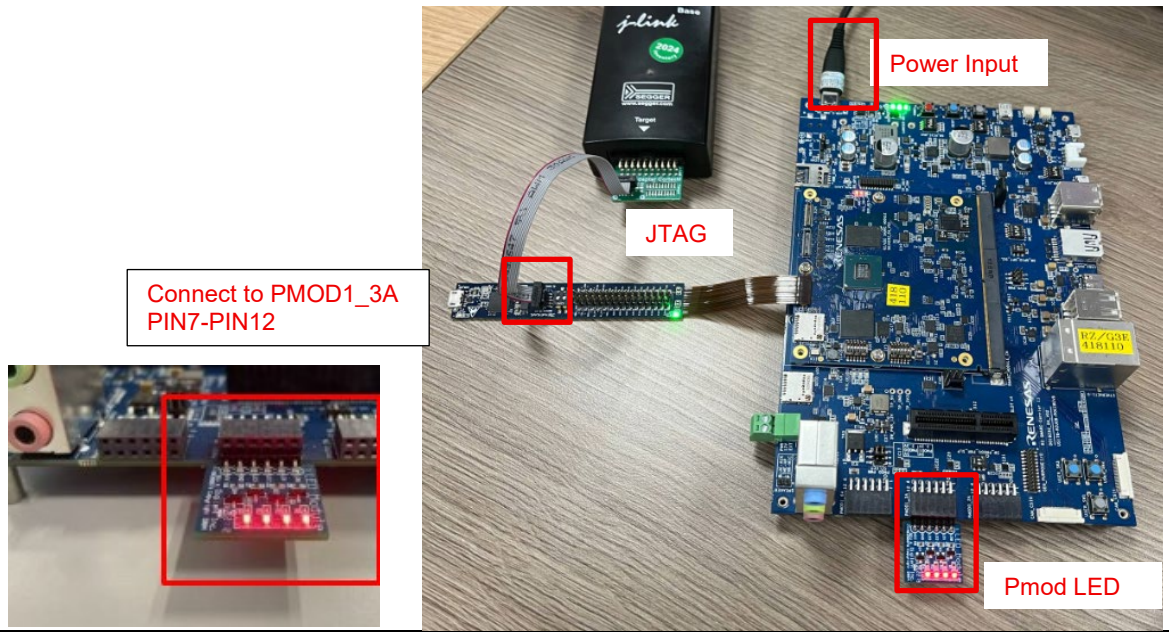


Figure 46: GTM Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G2UL EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4			SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_M2_DIS_4: ON PMOD_PWR_SEL: Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.7.4 Operation

After running the example, the message below is displayed on console. Users can input the period for One-shot mode and Periodic mode.

```
00> This Example Project demonstrates the functionality of GTM in periodic mode and one-shot mode.
00> On providing any input on the RTTviewer, GTM channel 2 starts in one-shot mode. GTM channel 1
00> starts in periodic mode when GTM channel 2 expires. Timer in periodic mode expires periodically
00> at a time period specified by user and toggles the Pmod0 pin 7 LED.
00>
00> Please enter time period values for one-shot and periodic mode timers in milliseconds
00>  Valid range: 1 to 2000
00>
00> One-shot mode:
  < 1000
00>
00> Time period for one-shot mode timer: 1000
00>
00> Periodic mode:
  < 1000
00>
00> Time period for periodic mode timer: 1000
00>
00> Enter any key to start or stop the timers
  <
00>
00> GTM2 is Enabled in OneShot mode
00>
00> One-shot mode GTM timer elapsed
00>
00> GTM1 is Enabled in Periodic mode
00> LED will toggle for set time period
00> Enter any key to stop timers
```

Figure 47: GTM Example - Operation

5.8 FreeRTOS

5.8.1 Project Overview

The example projects demonstrate Message Queue and Semaphore between tasks and interrupt. Message Queue is demonstrated between Tasks and between Task and interrupt. GTM timer periodically generates interrupt at 1000msec. For the first few seconds, messages are shared between Sender and Receiver Tasks and GTM ISR0. Receiver task pends on Message Queue, receives and displays message received on RTTViewer periodically at 500msec. For the next few seconds, Semaphore Task waits for semaphore until it is released by GTM ISR1. GTM ISR1 releases semaphore periodically at 1000msec.

All the tasks run with equal priority level.

5.8.2 Hardware Requirements

External hardware: None

5.8.3 Hardware Settings

Please connect each hardware as below.

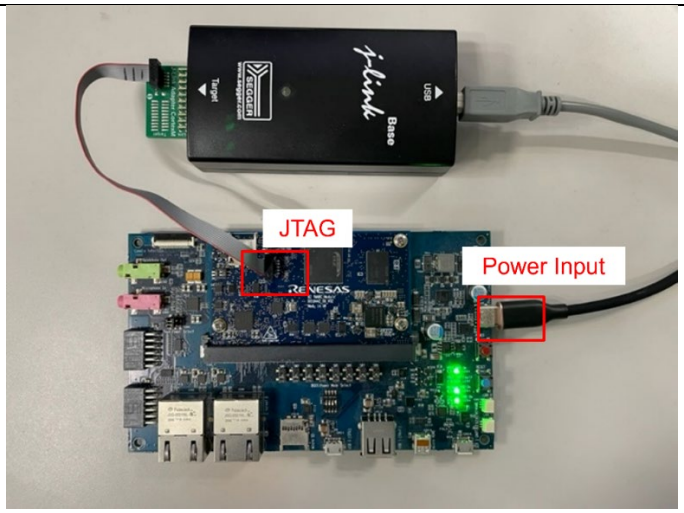


Figure 48: FreeRTOS Example - Board Setting (RZ/G2L, RZ/G2LC, RZ/G2UL)

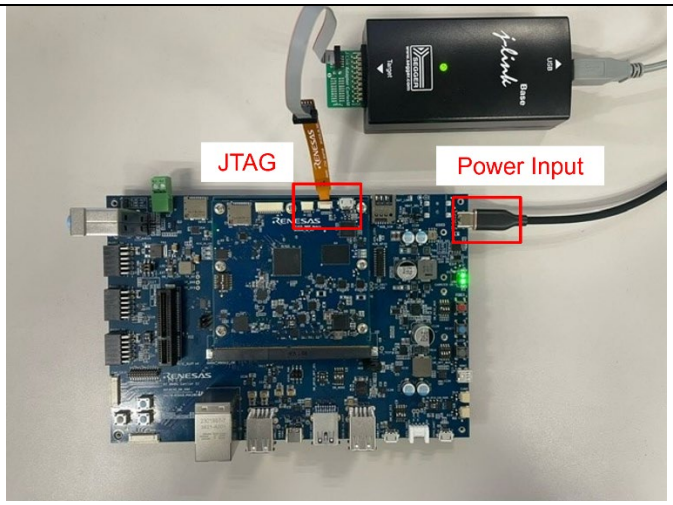


Figure 49: FreeRTOS Example - Board Setting (RZ/G3S)

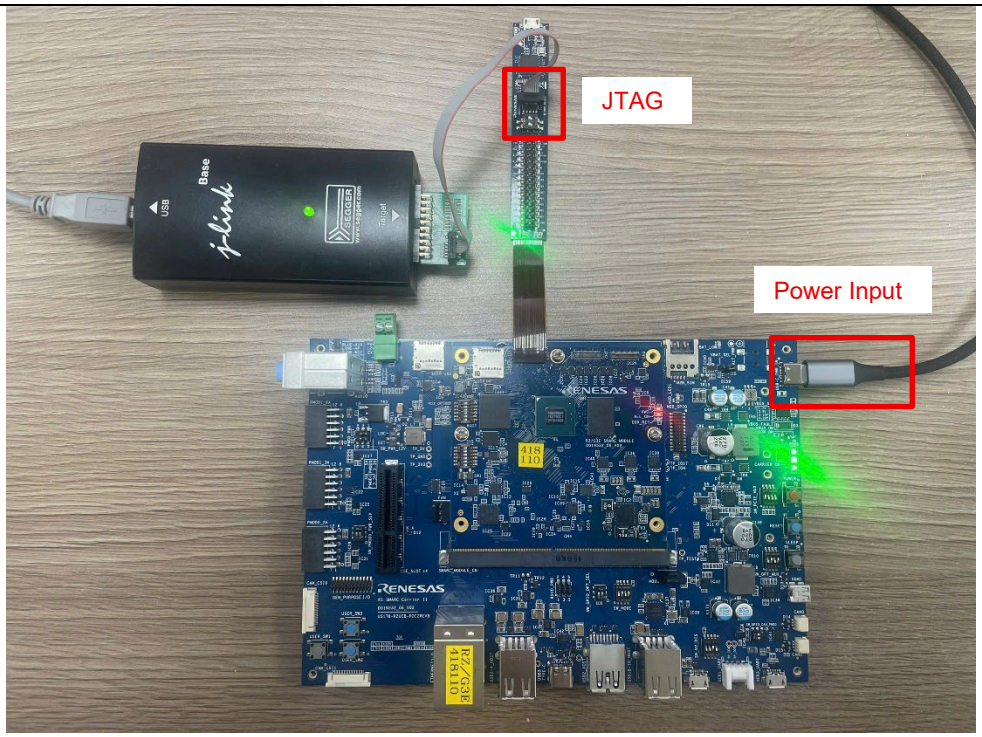


Figure 50: FreeRTOS Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G2UL EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4			SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SELECT: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON

5.8.4 Operation

After running the example, the message below is displayed on console.

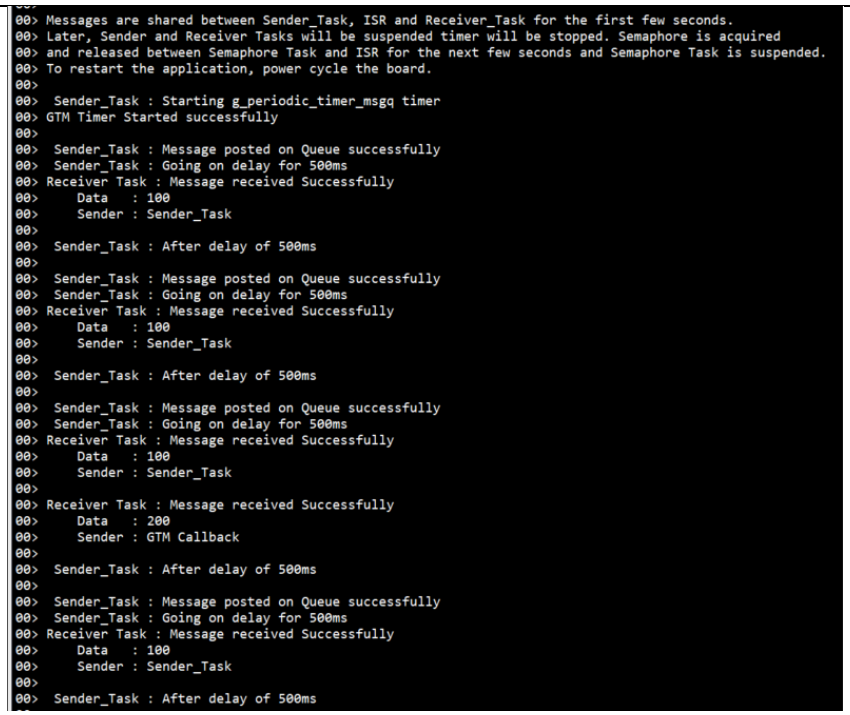


Figure 51: FreeRTOS Example - Operation

5.9 INTC_IRQ

5.9.1 Project Overview

The example project demonstrates the typical use of the INTC IRQ module APIs. The project initializes the IRQ interrupt in Interrupt Controller. User is requested to press the push button to trigger the external IRQ and this then will start toggling of user LED.

5.9.2 Hardware Requirements

External hardware:

RZ/G2 boards: Pmod BUTTON and Pmod LED

RZ/G3 boards: Pmod LED

5.9.3 Hardware Settings

Please connect each hardware as below.

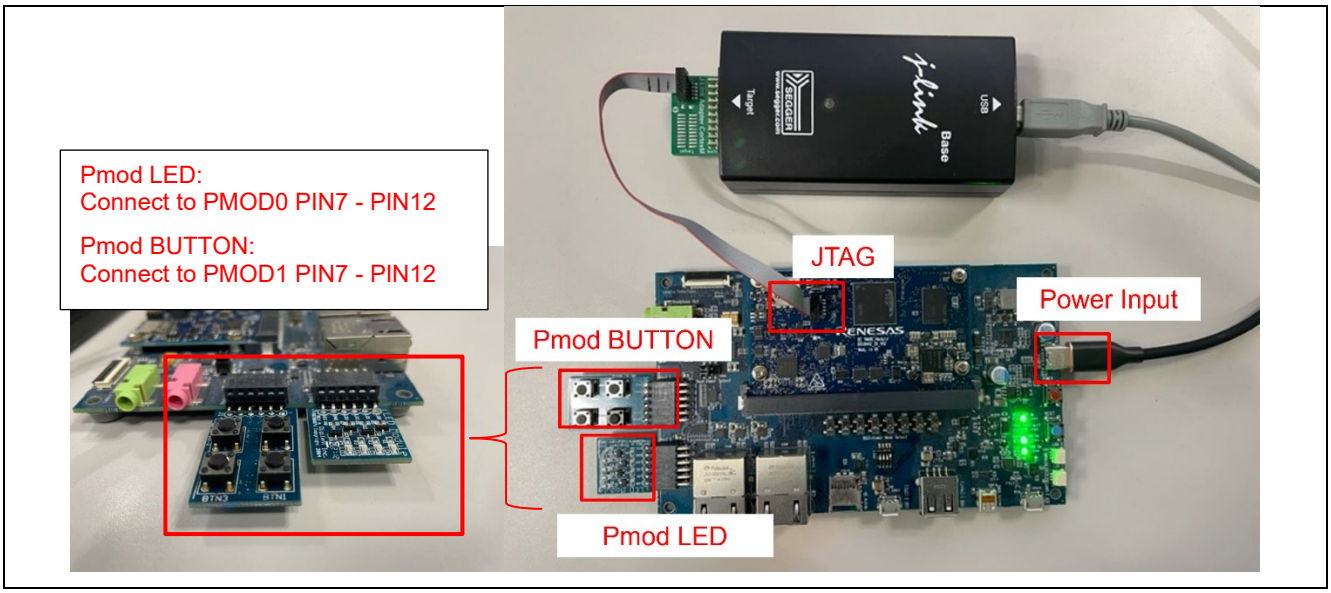
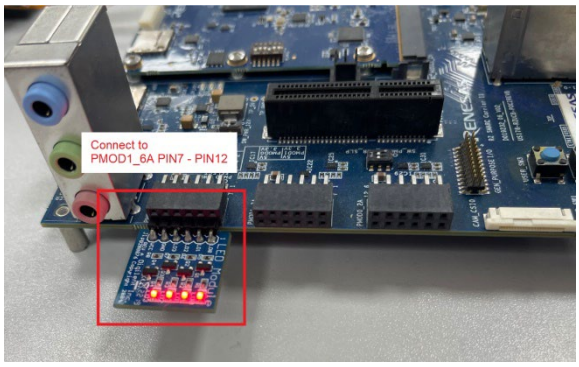


Figure 52: INTC IRQ Example - Board Setting (RZ/G2L, RZ/G2LC, RZ/G2UL)



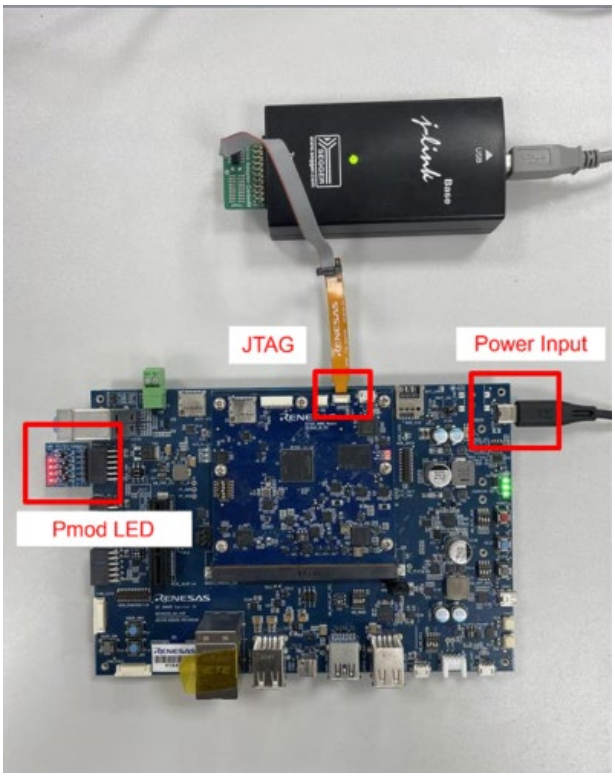
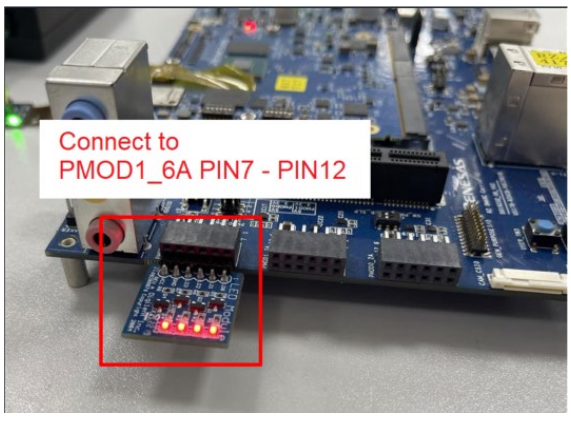


Figure 53: INTC IRQ Example - Board Setting (RZ/G3S)



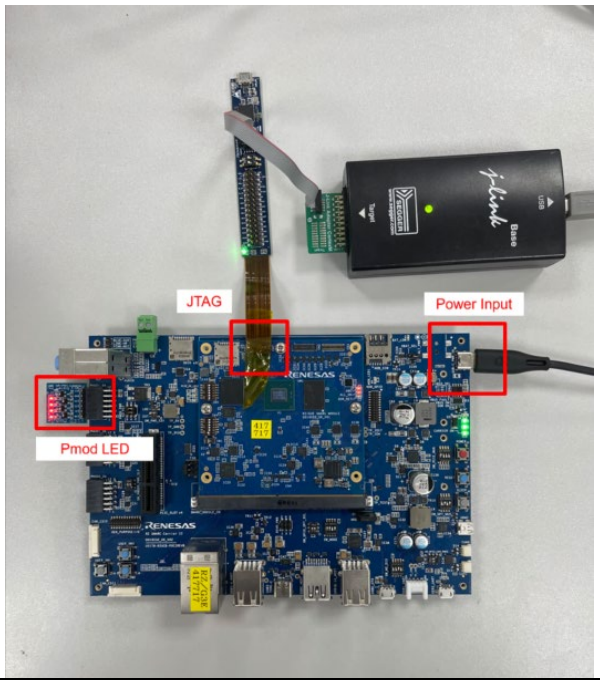


Figure 54: INTC IRQ Example - Board Setting (RZ/G3E)

Please set each DIP switches and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G2UL EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: 3-2 SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4			SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_M2_DIS_4: ON PMOD_PWR_SELECT: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_M2_DIS_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.9.4 Operation

After running the example, the message below is displayed on console. LED ON/OFF status is displayed when pressing the button (RZ/G3S, RZ/G3E: USER_SW1 button on Carrier Board, others: BTN0 of PMOD)

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_intc_irq driver
00>
00> This Example Project demonstrates the functionality of INTC_IRQ driver.
00> On pressing the user push button, an external IRQ is triggered, which toggles on-board LED.
00>
00> User Pushbutton Pressed
00> LED State: High{ON}
00>
00> User Pushbutton Pressed
00> LED State: Low{OFF}
```

Figure 55: INTC IRQ Example - Operation

5.10 WDT

5.10.1 Project Overview

For RZ/G2L, RZ/G2UL, RZ/G2LC and RZ/G3S:

The example project demonstrates the use of WDT module with the collaboration of module IRQ, GTM, WDT and button/led. The operation of this example is as below:

- Start GTM timer having a callback every 1 second, to refresh WDT and blink the led Pmod LED
- The WDT timeout is configured to 2 seconds.
- If the button (on Pmod BUTTON) is pressed, the WDT reset will issued.

For RZ/G3E:

The example project demonstrates the use of WDT module with the collaboration of module IRQ, GTM, WDT and button/led. The operation of this example is as below:

- Start GTM timer having a callback every 50 millliseconds, to refresh WDT and blink the led Pmod
- The WDT timeout is configured to 0.175 seconds.
- If the button (on Pmod BUTTON) is pressed, the WDT reset will issued.

5.10.2 Hardware Requirements

External hardware: Pmod BUTTON and Pmod LED

5.10.3 Hardware Settings

Please connect each hardware as below.

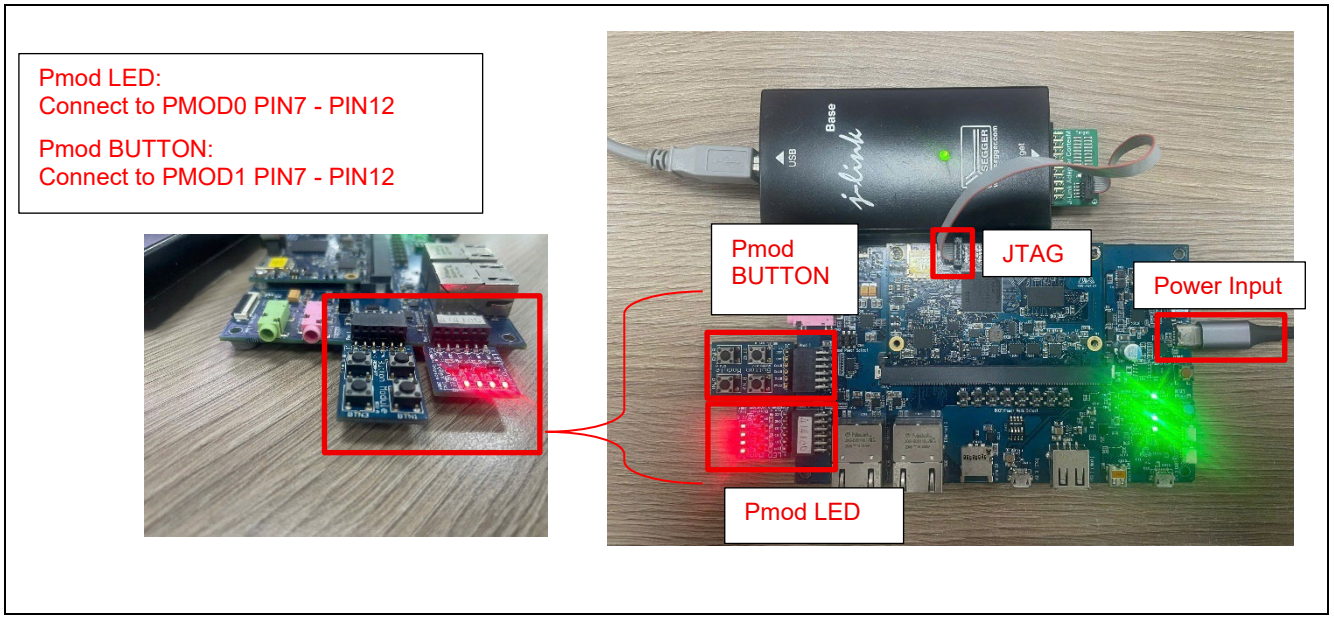


Figure 56: WDT Example - Board Setting (RZ/G2L, RZ/G2LC, RZ/G2UL)

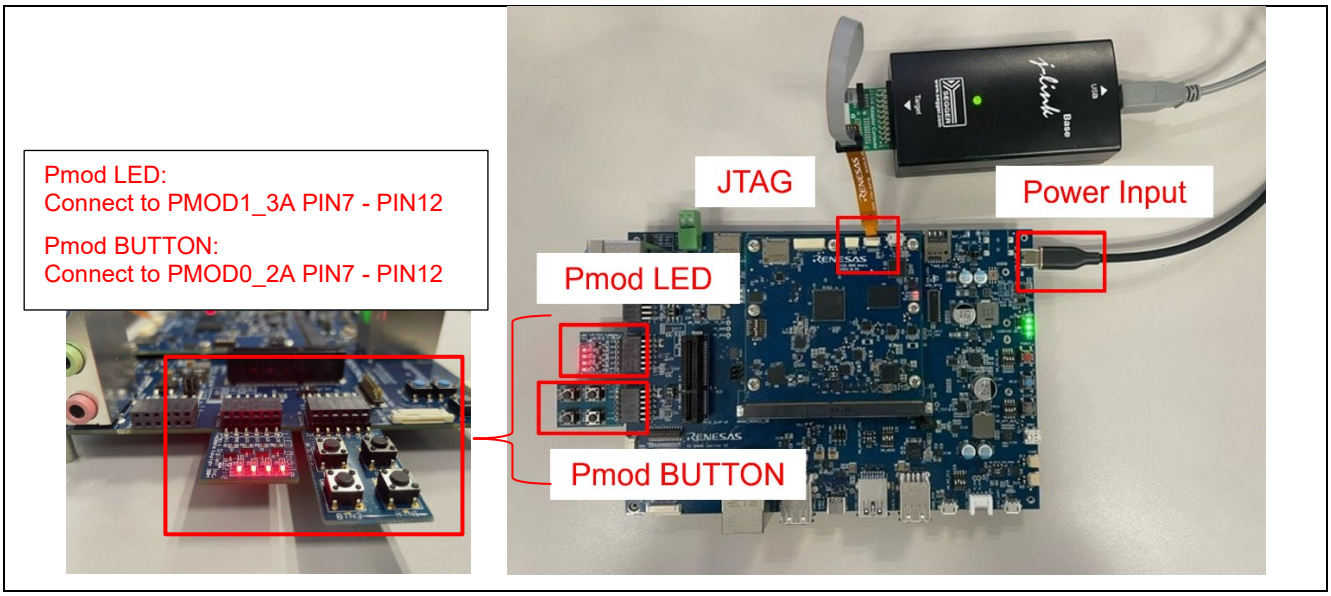


Figure 57: WDT Example - Board Setting (RZ/G3S)

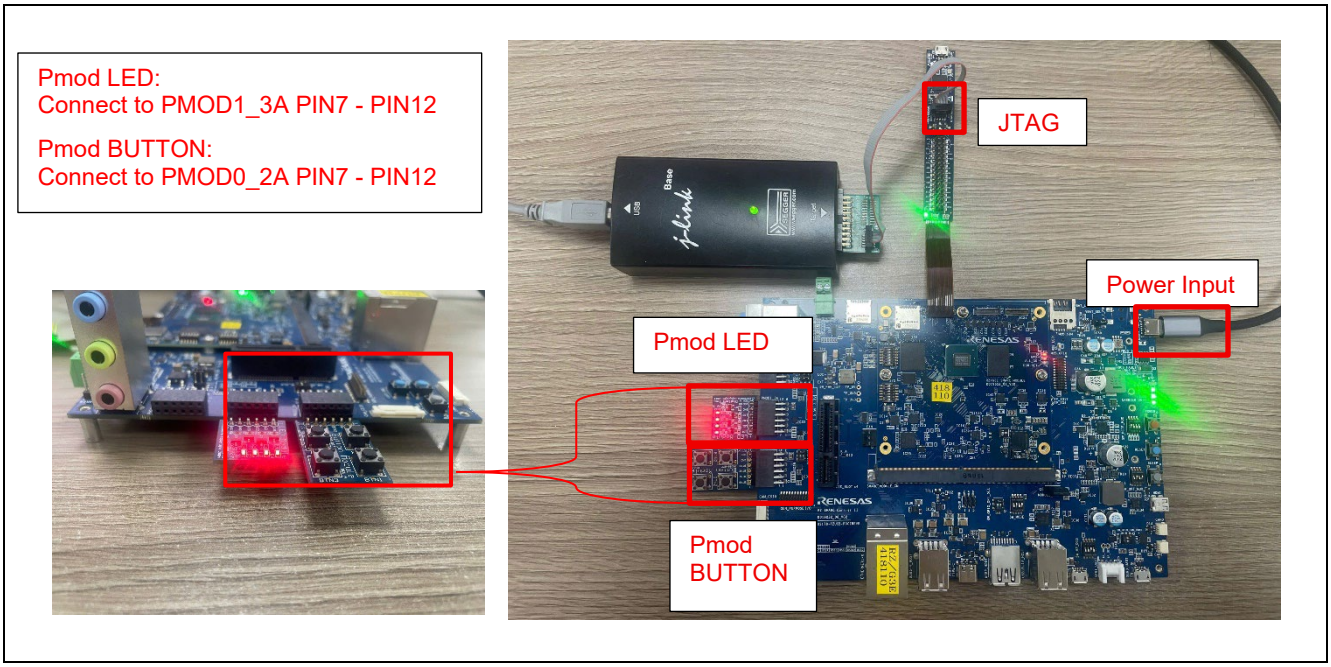


Figure 58: WDT Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2LC EVK	RZ/G2UL EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: 3-2 SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4			SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON SW_PMOD0_PWR_SLP: 5-6 PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4

5.10.4 Operation

For RZ/G2L, RZ/G2UL, RZ/G2LC boards, RZ/G3S board:

When J-Link is connected to the board, the message below is displayed on console. After inputting “1” to initial WDT and start GTM timer, the GTM timer will refresh WDT counter every 1 second and blink the LED of Pmod LED. When pressing the BTN of Pmod BUTTON, IRQ is triggered to stop GTM timer, and after that, WDT will issued.

For RZ/G3E board:

When J-Link is connected to the board, the message below is displayed on console. After inputting “1” to initial WDT and start GTM timer, the GTM timer will refresh WDT counter every 50 milliseconds and blink the LED of Pmod LED. When pressing the BTN of Pmod BUTTON, IRQ is triggered to stop GTM timer, and after that, WDT will issued.

For RZ/G2L, RZ/G2UL, RZ/G2LC and RZ/G3S board:

```
00>
00> ********************************************************************************
00> *   Renesas FSP Example Project for r_wdt Module                                *
00> *   Example Project Version 3.0.0                                              *
00> *   Flex Software Pack Version  3.1.0                                          *
00> ********************************************************************************
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_wdt driver
00>
00> This example project demonstrates the typical use of the WDT HAL module APIs
00> User input initializes the WDT and start GTM timer
00> WDT counter is refreshed periodically every 1 second when the GTM timer expires
00> On pressing the Push button, the WDT reset will issued
00>
00> Enter 1 to Enable WDT
00>
00> User Input:
  < 1
00> WDT initialized, GTM Timer Started
00> To stop WDT counter from refreshing, press the push button
00> WDT counter Refreshed.
00> WDT counter Refreshed.
00> WDT counter Refreshed.
00> Push button is pressed.
00> GTM timer stopped.
```

Figure 59: WDT Example - Operation (RZ/G2L, RZ/G2UL, RZ/G2LC and RZ/G3S)

For RZ/G3E

```
00>
00> ********************************************************************************
00> *   Renesas FSP Example Project for r_wdt Module                                *
00> *   Example Project Version 3.0.0                                              *
00> *   Flex Software Pack Version  3.1.0                                          *
00> ********************************************************************************
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_wdt driver
00>
00> This example project demonstrates the typical use of the WDT HAL module APIs
00> User input initializes the WDT and start GTM timer
00> WDT counter is refreshed periodically every 50 milliseconds when the GTM timer expires
00> On pressing the Push button, the WDT reset will issued
00>
00> Enter 1 to Enable WDT
00>
00> User Input:
  < 1
00> WDT initialized, GTM Timer Started
00> To stop WDT counter from refreshing, press the push button
00> WDT counter Refreshed.
00> WDT counter Refreshed.
00> WDT counter Refreshed.
00> WDT counter Refreshed.
00> Push button is pressed.
00> GTM timer stopped.
```

Figure 60: WDT Example - Operation (RZ/G3E)

5.11 ADC_E

5.11.1 Project Overview

The example project demonstrates the function of r_adc_e driver. The example project demonstrates the typical use of the ADC HAL module APIs. The project initializes the ADC in single scan mode in accordance with FSP configuration. Result and ADC status is displayed on the JLink RTT Viewer.

5.11.2 Hardware Requirements

External Hardware : External Variable Power Supply Unit (Rheostats 10K).

5.11.3 Hardware Settings

Please connect each hardware as below.

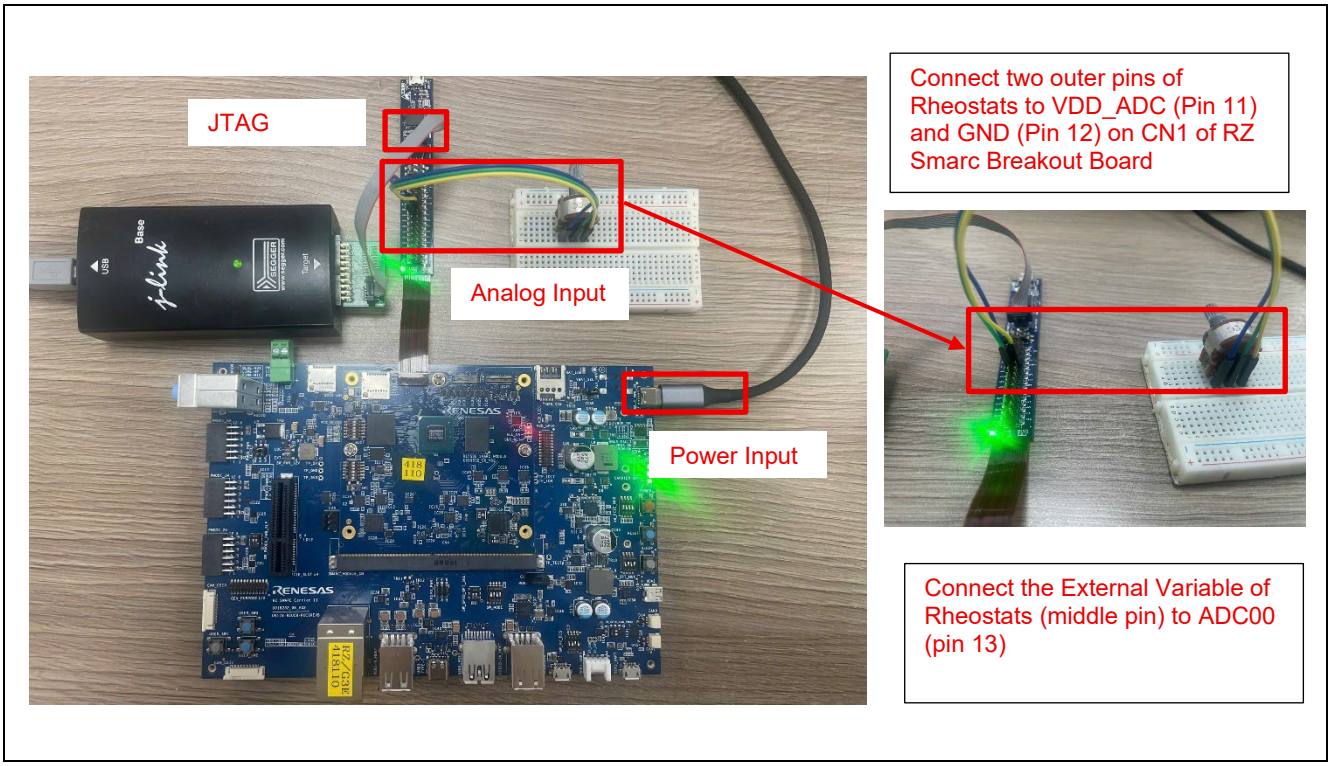


Figure 61: ADC_E Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON

5.11.4 Operation

Press 1 to Start ADC Scan via RTTViewer software and the result will be showed as below.

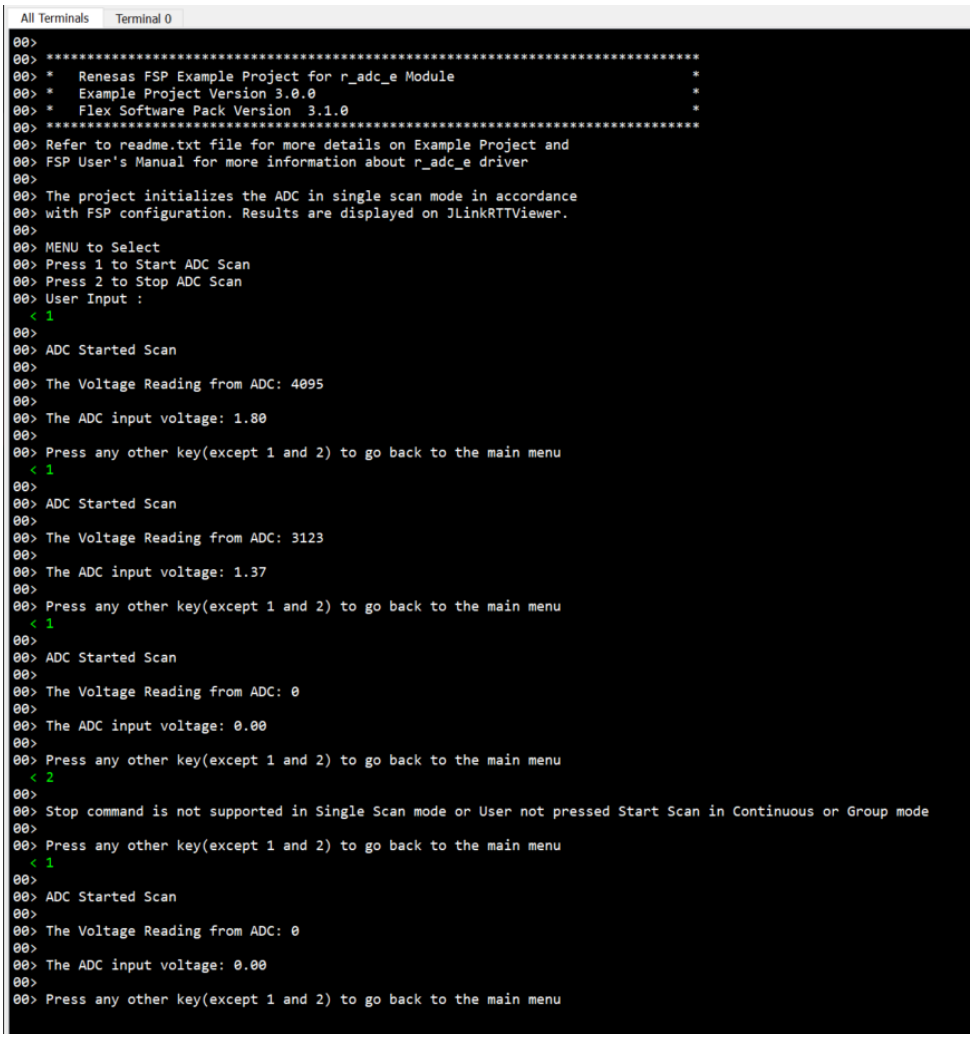


Figure 62: ADC_E Example - Operation

5.12 RIIC_Slave

5.12.1 Project Overview

The example project demonstrates typical use of the riic slave HAL module APIs. The project initializes riic slave and riic master module with standard rate and is made interfaced with loop-back mechanism. It performs slave read and write operation continuously once initialization is successful. On successful i2c transaction (6 bytes), Data transceived is compared. Led blinks on data match else it is turned on as sign of failure. Output message for both corresponding slave operations is displayed on RTT Viewer. Any API/event failure message is also displayed.

5.12.2 Hardware Requirements

External Hardware : Pmod LED

5.12.3 Hardware Settings

Please connect each hardware as below.

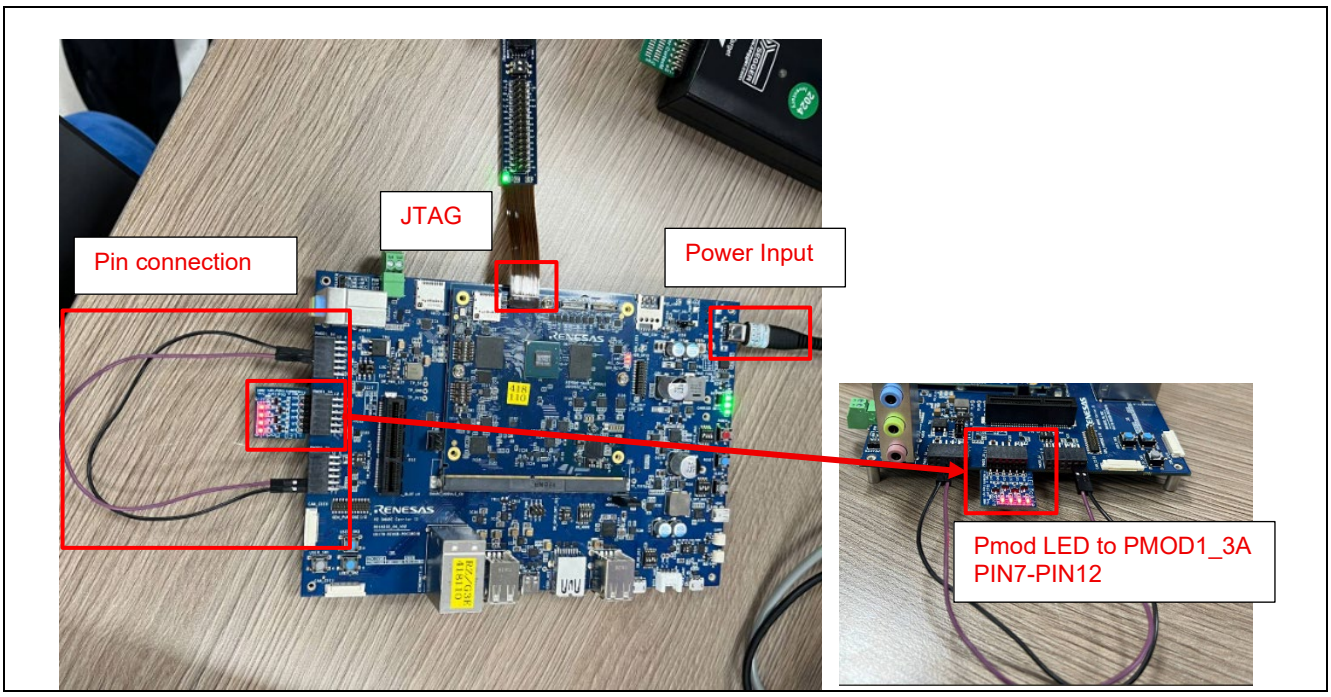


Figure 63: RIIC Slave Example - Board Setting (RZ/G3E)

Please set the Pin connection as below.

RZ/G3E
PIN 7 (PMOD0_2A) - PIN 4 (PMOD1_6A)
PIN 8 (PMOD0_2A) - PIN 3 (PMOD1_6A)

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4

5.12.4 Operation

After running the example, the message below is displayed on console.

```
********************************************************************************
*   Renesas FSP Example Project for r_riic_slave Module                        *
*   Example Project Version 3.0.0                                              *
*   Flex Software Pack Version  3.1.0                                          *
********************************************************************************
Refer to readme.txt file for more details on Example Project and
FSP User's Manual for more information about r_riic_slave driver

This EP demonstrates riic slave operation using two riic channels.
It performs slave read and write operation continuously once
initialization is successful. On successful riic transaction
(6 bytes), Data transceived is compared. Led blinks on data
match else it is turned ON as sign of failure. For both cases
corresponding slave operation message is displayed on RTT.
Any API/event failure message is also displayed.


** Slave read operation is successful **
** Slave Write operation is successful **
** Slave read operation is successful **
** Slave Write operation is successful **
** Slave read operation is successful **
** Slave Write operation is successful **
```

Figure 64: RIIC Slave Example - Operation

5.13 INTC_NMI

5.13.1 Project Overview

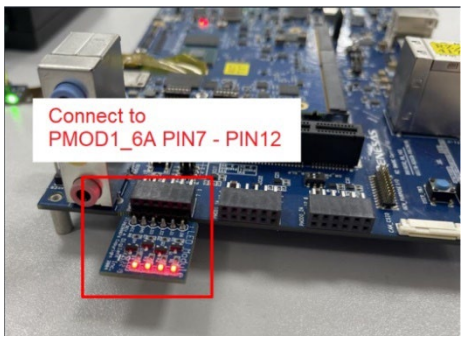
The example project demonstrates the typical use of the INTC NMI module APIs. The project initializes the IRQ interrupt in Interrupt Controller User is requested to press the SLEEP push button on Carrier Board to trigger the external irq and this then will start toggling of user LED.

5.13.2 Hardware Requirements

External Hardware : Pmod LED

5.13.3 Hardware Settings

Please connect each hardware as below.



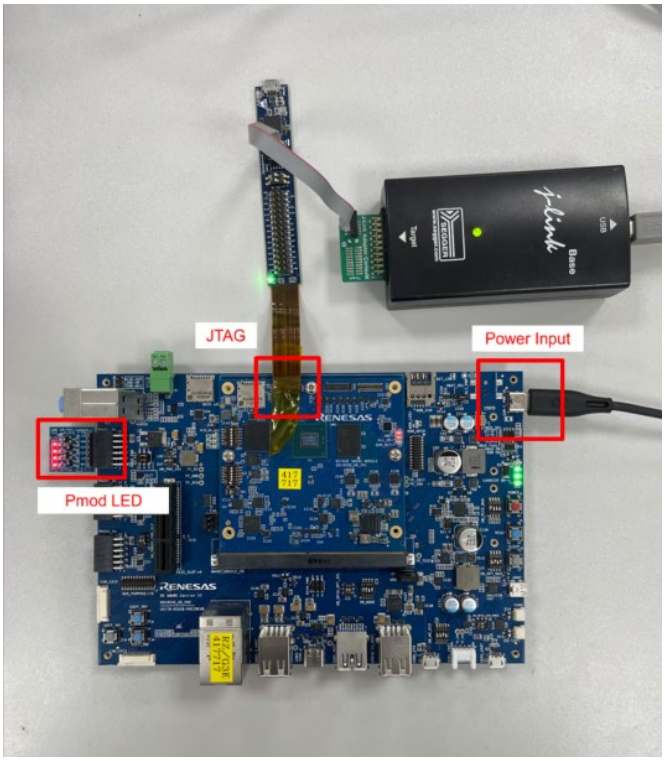


Figure 65: INTC NMI Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.13.4 Operation

After running the example, the message below is displayed on console. LED ON/OFF status is displayed when pressing the button (RZ/G3E: SLEEP button on Carrier Board)

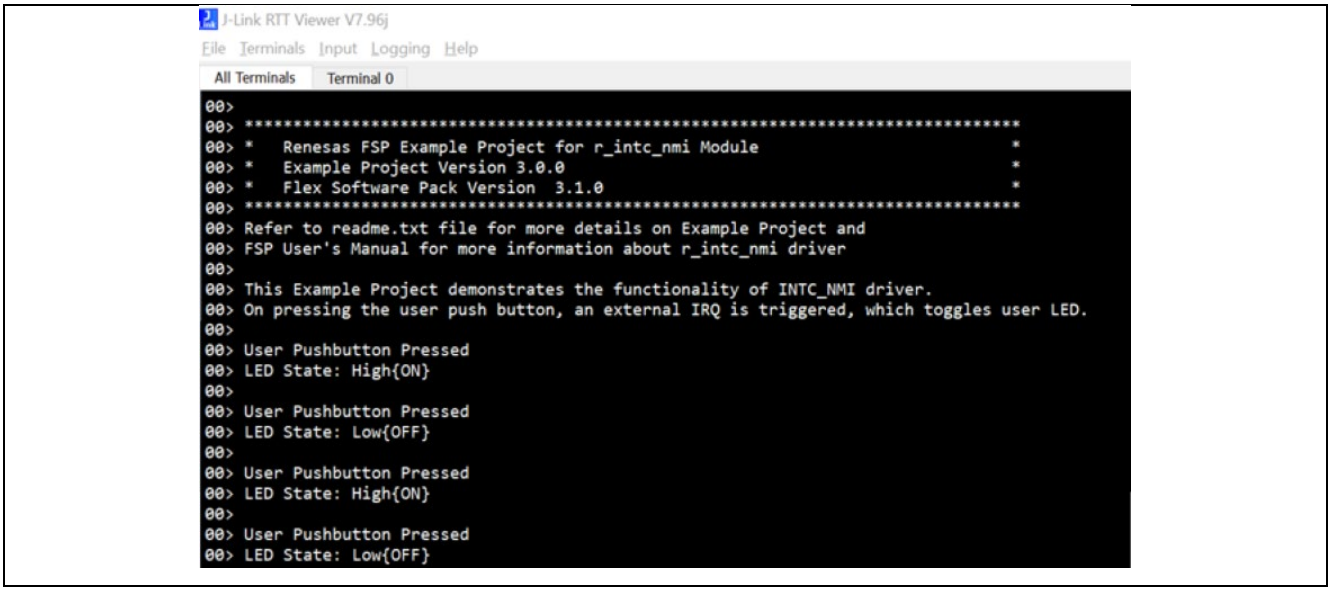


Figure 66: INTC NMI Example - Operation

5.14 INTC_TINT

5.14.1 Project Overview

The example project demonstrates the typical use of the INTC TINT module APIs. The project initializes the IRQ interrupt in Interrupt Controller User is requested to press the USER_SW1 push button on Carrier Board to trigger the external irq and this then will start toggling of user LED.

5.14.2 Hardware Requirements

External Hardware : Pmod LED

5.14.3 Hardware Settings

Please connect each hardware as below.

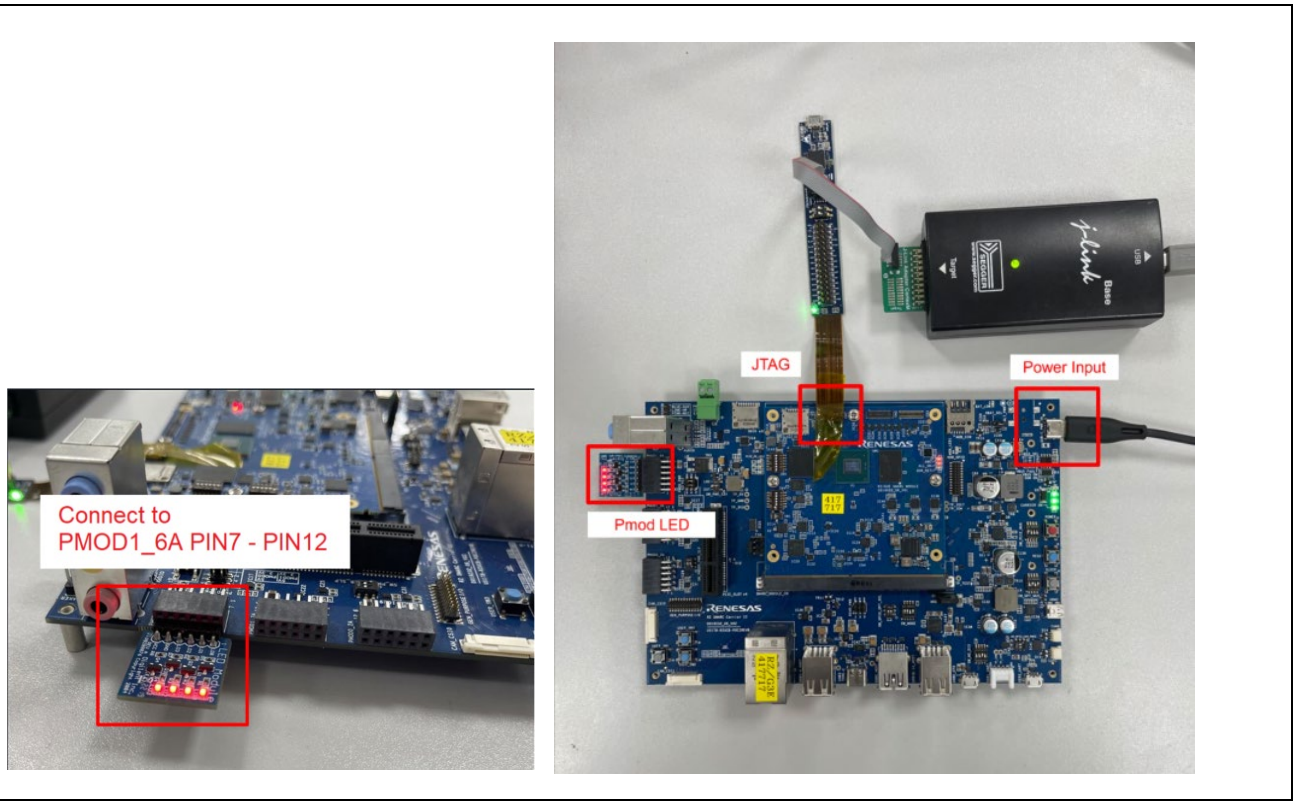


Figure 67: INTC TINT Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.14.4 Operation

After running the example, the message below is displayed on console. LED ON/OFF status is displayed when pressing the button (RZ/G3E: USER_SW1 button on Carrier Board)

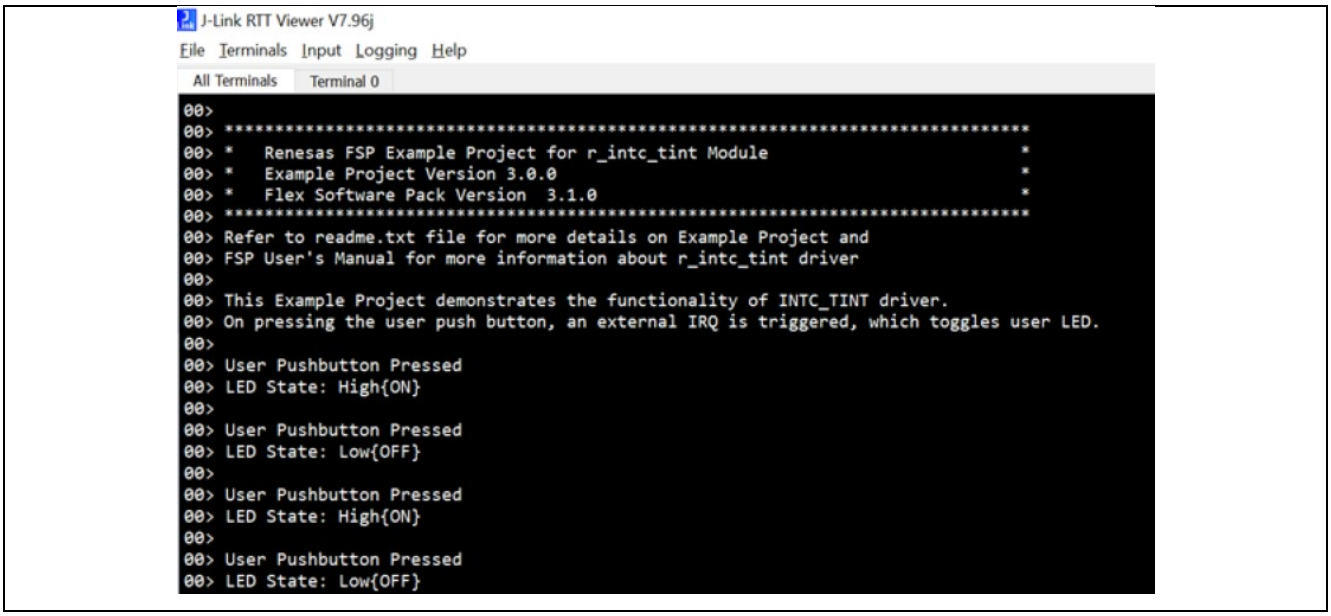


Figure 68: INTC TINT Example - Operation

5.15 POEG

5.15.1 Project Overview

This example demonstrates the main functions of the POEG module. The user is prompted to enter a number to select the desired trigger:

RZ/G3S, RZ/G3E:

- 1 for POEG Trigger Pin mode-GPT Output disable on GTETRG pin level
- 2 for POEG Register control Output disable by R_POEG_OutputDisable() software API request
- 3 for POEG Trigger Output Level mode-GPT Output disable from the GPT
- 4 for Enable Port output using POEG_Reset API

RZ/G2L, RZ/G2LC:

- 1 for POEG Register control Output disable by R_POEG_OutputDisable() software API request
- 2 for POEG Trigger Output Level mode-GPT Output disable from the GPT
- 3 for Enable Port output using POEG_Reset API

Based on the user's selection, the corresponding trigger mechanism is activated to stop the GPT signal output or reset POEG.

5.15.2 Hardware Requirements

External Hardware : Pmod LED

5.15.3 Hardware Settings

Please connect each hardware as below.

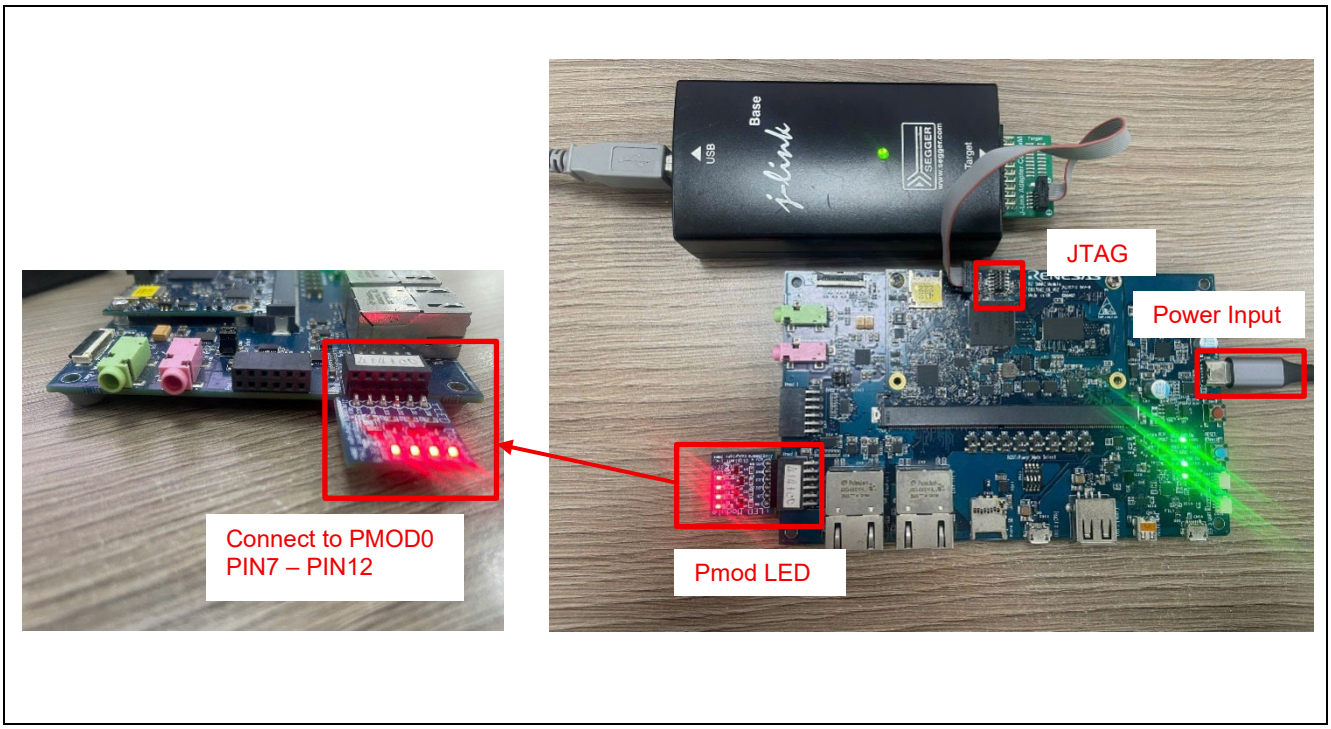


Figure 69: POEG Example - Board Setting (RZ/G2L, RZ/G2LC)

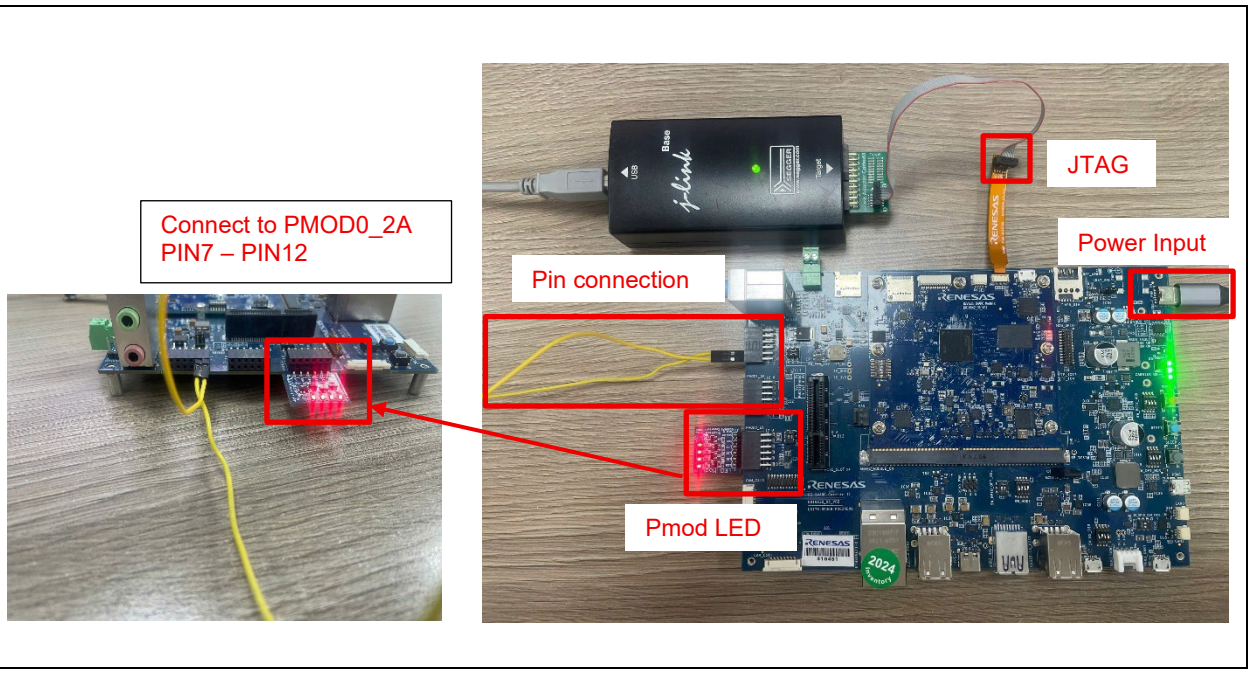


Figure 70: POEG Example - Board Setting (RZ/G3S)

Please set the Pin Connection as below.

RZ/G3S EVK
Connect P18_1 (PMOD1_6a PIN1) to P14_2 (PMOD1_6a PIN2) on SMARC Carrier Board

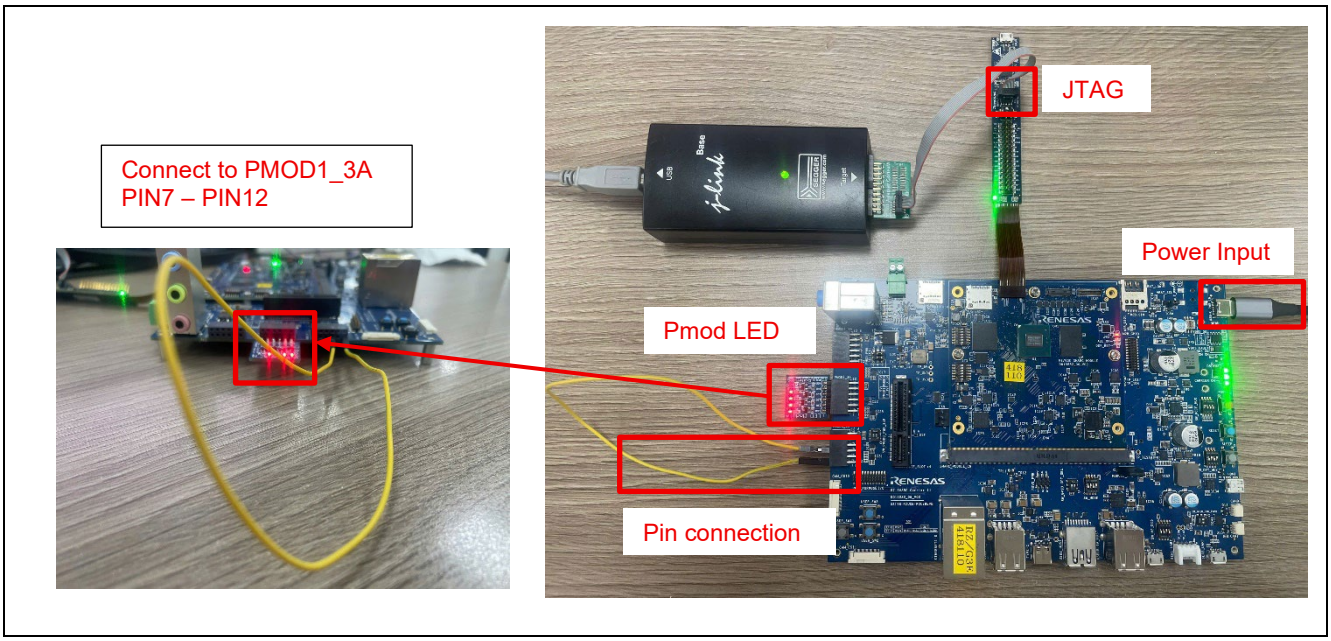


Figure 71: POEG Example - Board Setting (RZ/G3E)

Please set the Pin Connection as below.

RZ/G3E EVK
Connect P84 (PMOD0_2a PIN9) to P31 (PMOD0_2a PIN7) on SMARC Carrier Board

Please set each DIP switch and jumpers as below.

Board	RZ/G2L EVK	RZ/G2UL EVK	RZ/G2LC EVK	RZ/G3S EVK	RZ/G3E EVK
Module board	SW1-1: OFF SW1-2: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care.	SW1-1: OFF SW1-2: Don't care. SW1-3: Don't care. SW1-4: Don't care. SW1-5: Don't care. SW1-6: Don't care.	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: ON SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4		SW1: 3-2. SW2: 3-2. SW3: 3-2. SW4: 3-2. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4

5.15.4 Operation

After running the example, the message below is displayed on console.

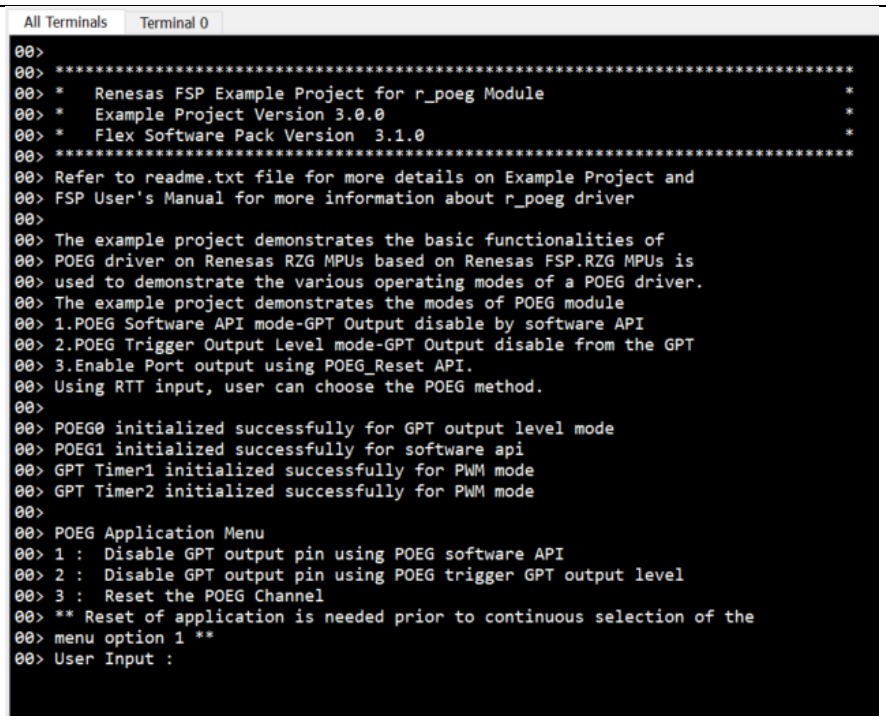


Figure 72: POEG Example – Operation for RZ/G2L and RZ/G2LC

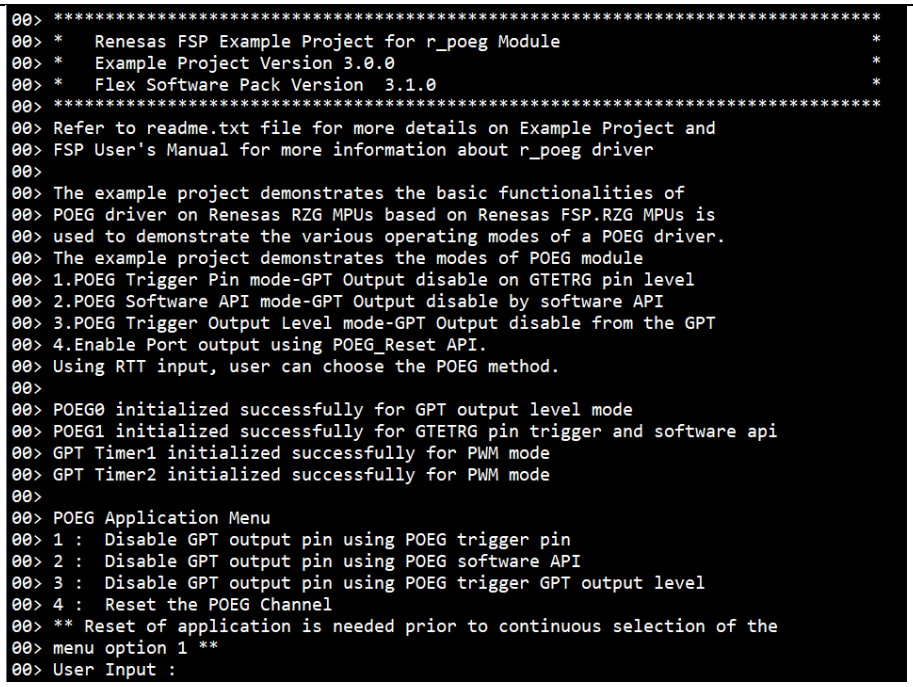


Figure 73: POEG Example – Operation for RZ/G3S

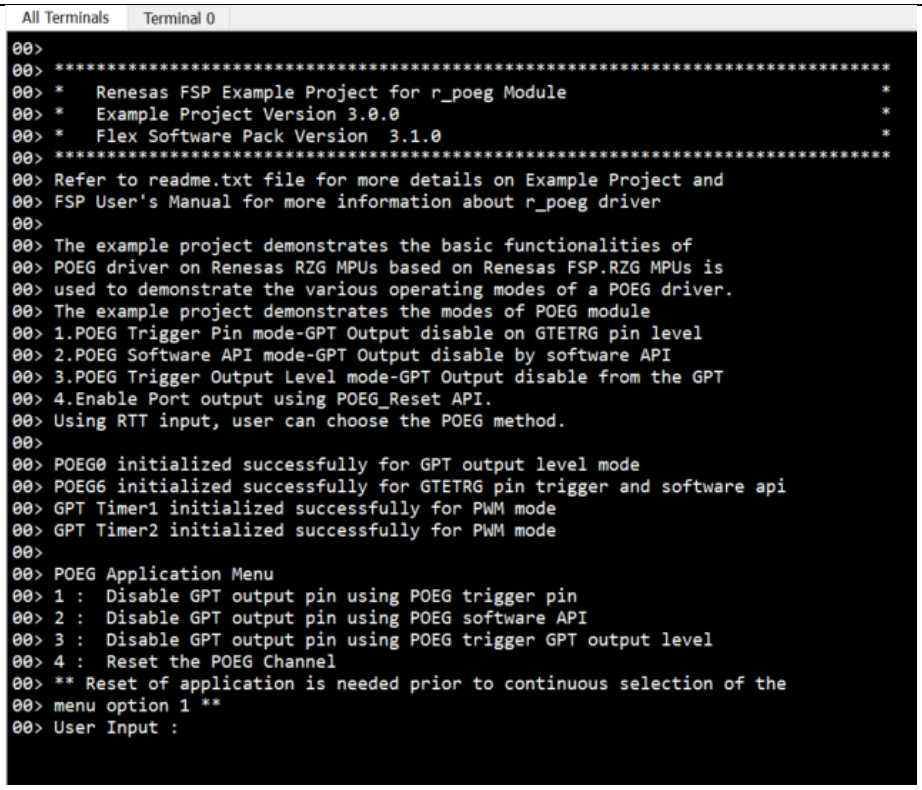


Figure 74: POEG Example – Operation for RZ/G2L and RZ/G2LC

5.16 SPI_B

5.16.1 Project Overview

The example project demonstrates the typical use of the SPI_B HAL module APIs. The project configure RSPI channel 0 in Master and Pmod SF3 in Slave. Once the module is initialised and the channels are configured, Master can write data to Slave and read it back based on commands from user sent through JLink RTT Viewer.

5.16.2 Hardware Requirements

External Hardware : Pmod SF3

5.16.3 Hardware Settings

Please connect each hardware as below.

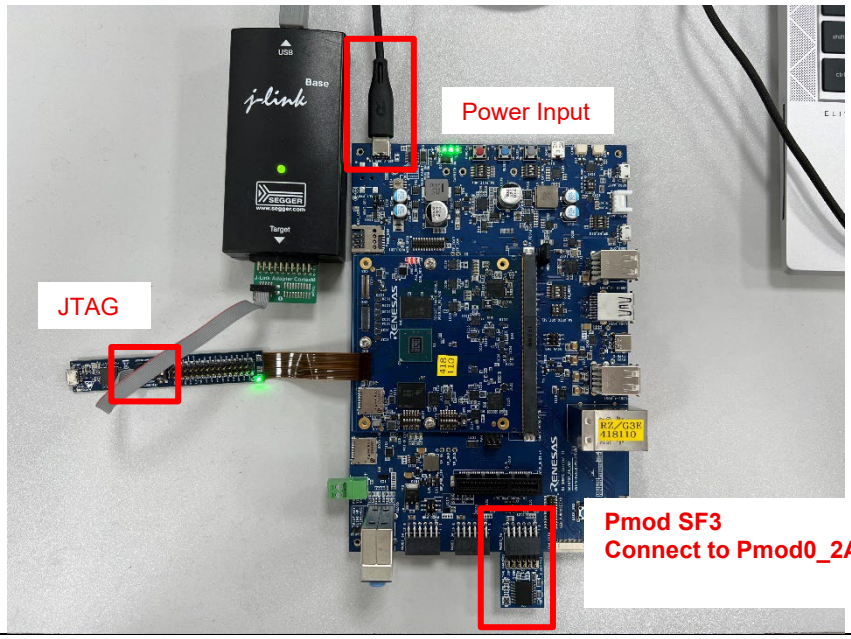


Figure 75: SPI_B Example - Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3

5.16.4 Operation

Display the message of the project overview and the number for the option.

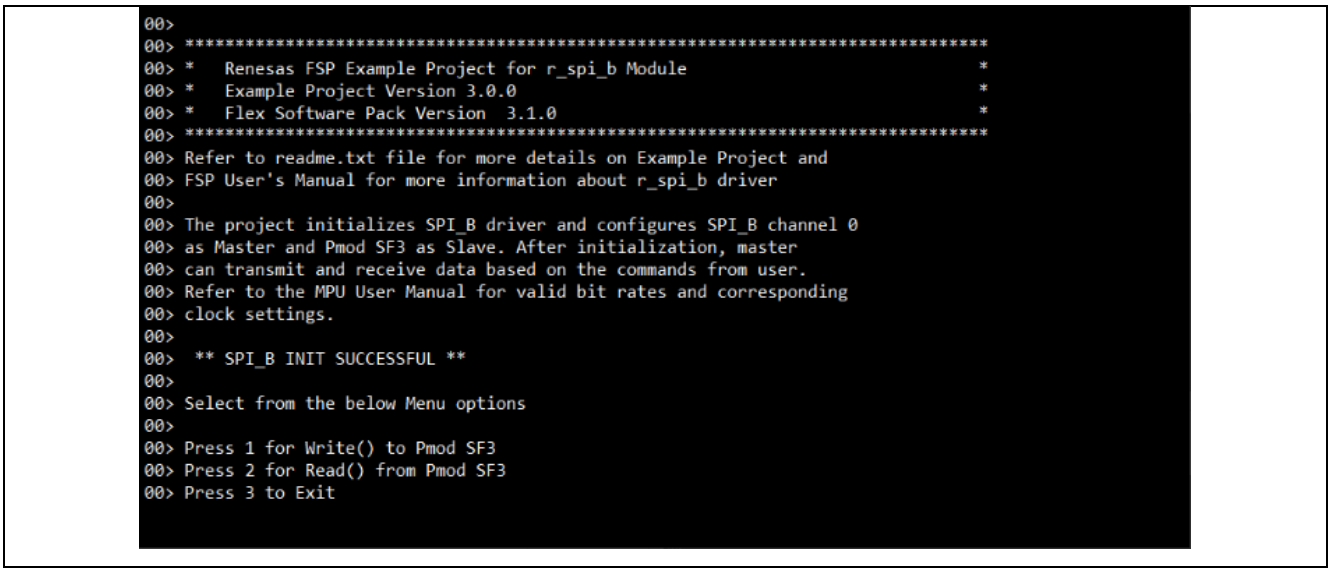


Figure 76: SPI_B Example – Operation (Console)

Input 1 to write operation. After that, input 32 bytes data ("abcdefghijklmnopqrstuvwxyz012345") for writing to Pmod SF3. Press Enter to turn back Menu Selection. Then, input 2 to read operation. Finally, confirm reading text is as same as writing text.

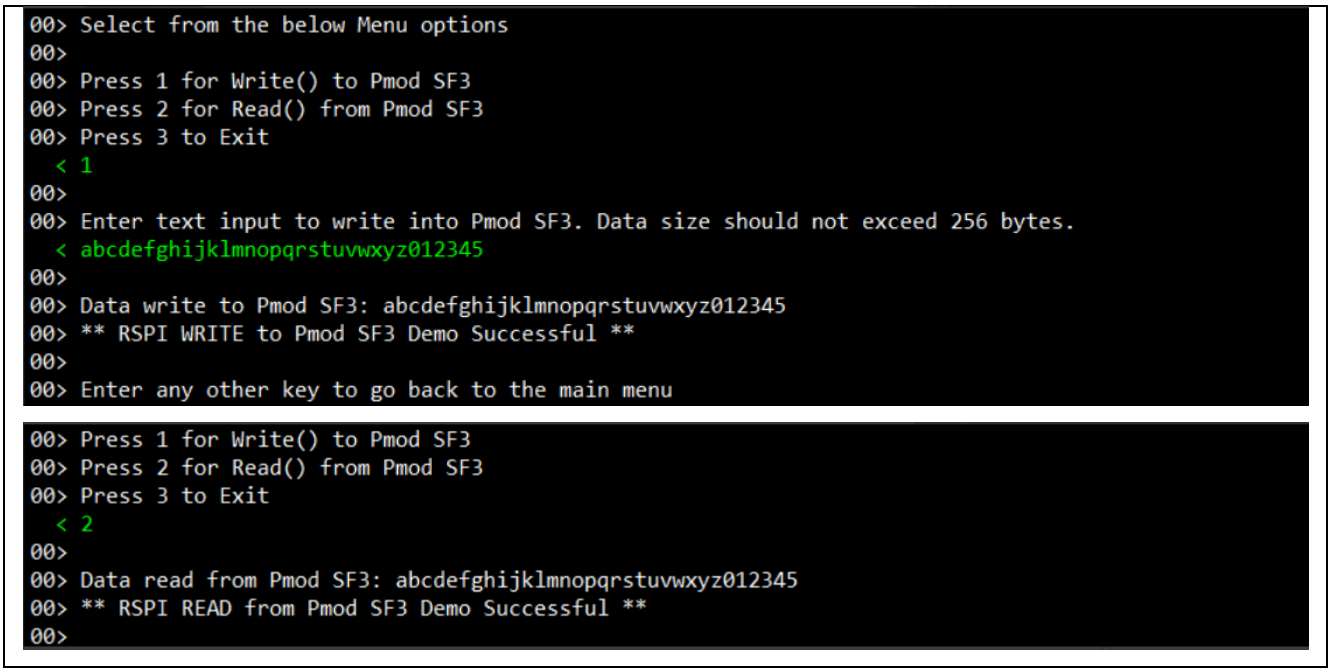


Figure 77: SPI_B Example – Operation (option 1 and 2)

Enter "3" to exit the demo.

```
00> Press 1 for Write() to Pmod SF3
00> Press 2 for Read() from Pmod SF3
00> Press 3 to Exit
  < 3
00> ** RSPI EXIT Demo Successful **
00>
00> Exiting RSPI demo....
```

Figure 78: SPI_B Example – Operation (option 3)

5.17 SCI_B_UART

5.17.1 Project Overview

The example project demonstrates the typical use of the SCI_B_UART HAL module APIs. The project initializes the UART with Baud rate of 115200 bps and GTM module. Using a Terminal Program (like Tera Term) user can provide a value & press enter key to set the period of the Pmod LED signal. The range of input values are displayed on the JLink RTT Viewer. Any failure will also be displayed using JLink RTT Viewer. To see user input values on Serial terminal, enable local echo option.

5.17.2 Hardware Requirements

External Hardware : Pmod LED, Pmod USBUART

5.17.3 Hardware Settings

Please connect each hardware as below.

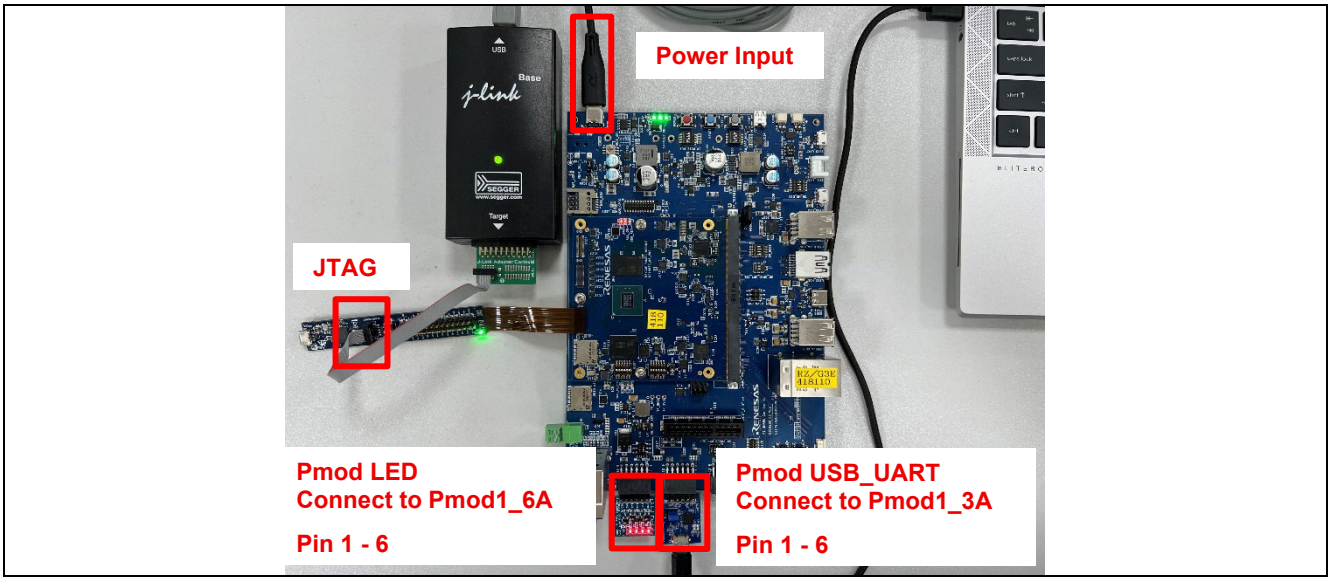


Figure 79: SCI_B UART Example - Board Setting (RZ/G3E)

Please set the connection between Pmod USBUART and SMARC Carrier Board as follows.

Connection of Pmod USBUART
RXD: Pmod USBUART J2 PIN2 – PMOD1_3A PIN2
TXD: Pmod USBUART J2 PIN3 – PMOD1_3A PIN3
GND: Pmod USBUART J2 PIN5 – PMOD1_3A PIN5
VCC: Pmod USBUART J2 PIN6 – PMOD1_3A PIN6

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3 Jumper connects 2-4

5.17.4 Operation

After running the example, the message below is displayed on console.

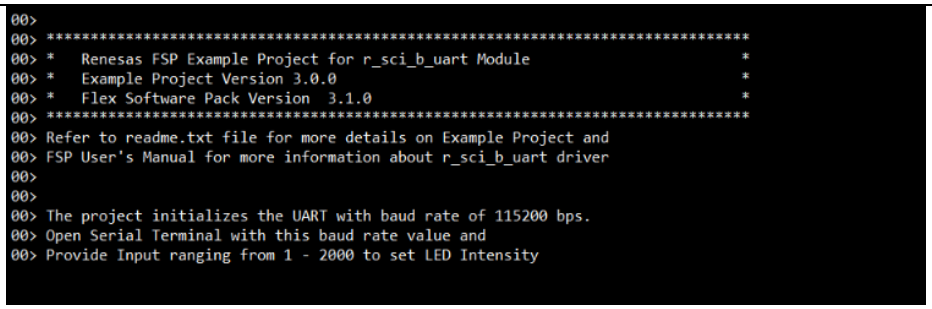


Figure 80: SCI_B UART Example – Operation (Console)

User can specify a value for LED blinking on Terminal Program.

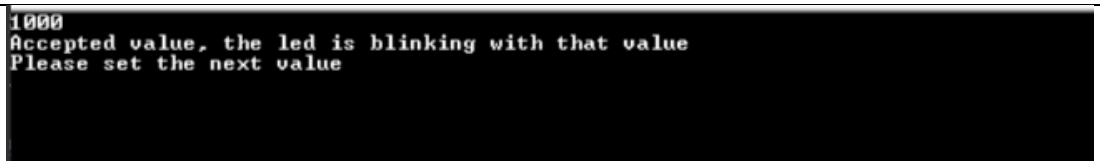


Figure 81: SCI_B UART Example – Operation (specify a value)

The configuration of Terminal Program is as below.
- Speed: 115200bps
- Data: 8bit
- Parity: None
- Stop bits: 1bit
- Flow control: None

5.18 ELC

5.18.1 Project Overview

The example project demonstrates the typical use of the ELC HAL module APIs. ELC Software Event, GPT0 and GPT1 events are linked using ELC. The start source for GPT0 and GPT1 is ELC Software Event and the stop source for GPT0 is GPT1 counter overflow. GPT0 runs in PWM mode and GPT1 runs in one-shot mode. On giving valid RTT input, an ELC Software Event is generated that triggers LED blinking. LED stops blinking after 5 sec when GPT0 expires.

5.18.2 Hardware Requirements

External Hardware : Pmod LED

5.18.3 Hardware Settings

Please connect each hardware as below.

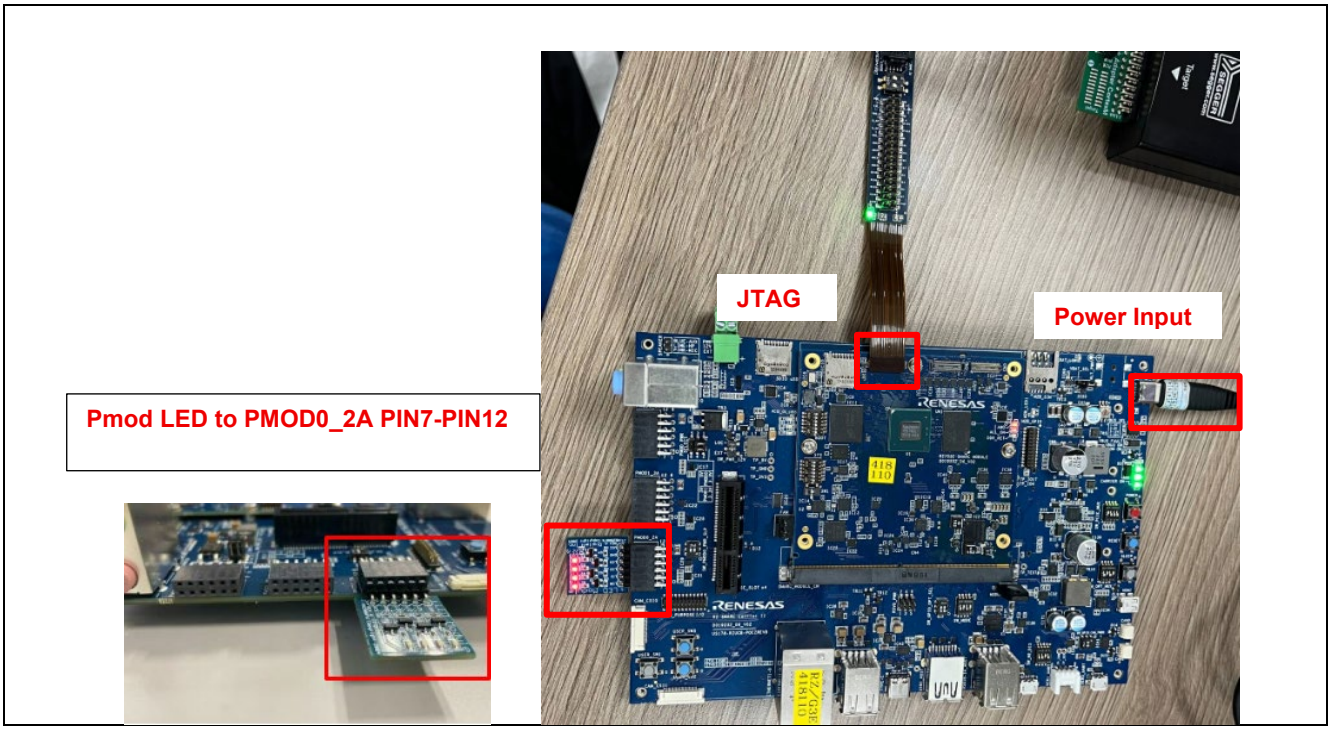


Figure 82: ELC Example – Board Setting (RZ/G3E)

Please set each DIP switch and jumpers as below.

Board	RZ/G3E EVK
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 1-3

5.18.4 Operation

After running the example, the message below is displayed on console.

Type any character on the input console (RTT Viewer) and check the status of LEDs.

```
*********************************************************************************
*   Renesas FSP Example Project for r_elc Module                                *
*   Example Project Version 3.0.0                                               *
*   Flex Software Pack Version  3.1.0                                           *
*********************************************************************************
Refer to readme.txt file for more details on Example Project and
FSP User's Manual for more information about r_elc driver

This Example Project demonstrates the basic usage of  ELC driver.
ELC Software Event, GPT0 and GPT1 events are linked using ELC.
The start source for GPT0 and GPT1 is ELC Software Event and the stop source for GPT0 is GPT1 counter overflow.
GPT0 runs in PWM mode and GPT1 runs in one-shot mode.
On giving valid RTT input, an ELC Software Event is generated that triggers LED blinking.
LED stops blinking after 5 seconds when GPT0 expires.

Enter any key as RTT input. LED blinks for 5 seconds and stopsq

Enter any key as RTT input. LED blinks for 5 seconds and stops
```

Figure 83: ELC Example – Operation

5.19 I3C_B

5.19.1 Project Overview

This example project demonstrates the typical use of the I3C Driver on Renesas RZ MPUs based on Renesas FSP. The I3C master on RZ/G3E demonstrates operations associated with an I3C slave running on another RZ/G3E board. Once initialization is successful, I3C Master device assigns slave address to I3C slave device through DAA (dynamic address assignment) using I3C common command code (CCC). Once Dynamic Address Assignment is completed, the Master EP performs write/read operations, and displays slave device information based on user input. If the the on-board switch is pressed on the slave side, it will initiate an IBI transfer request. Error and info messages will be printed on Jlink RTTViewer.

5.19.2 Hardware Requirements

External Hardware : Pmod LED

5.19.3 Hardware Settings

Please connect each hardware as below.

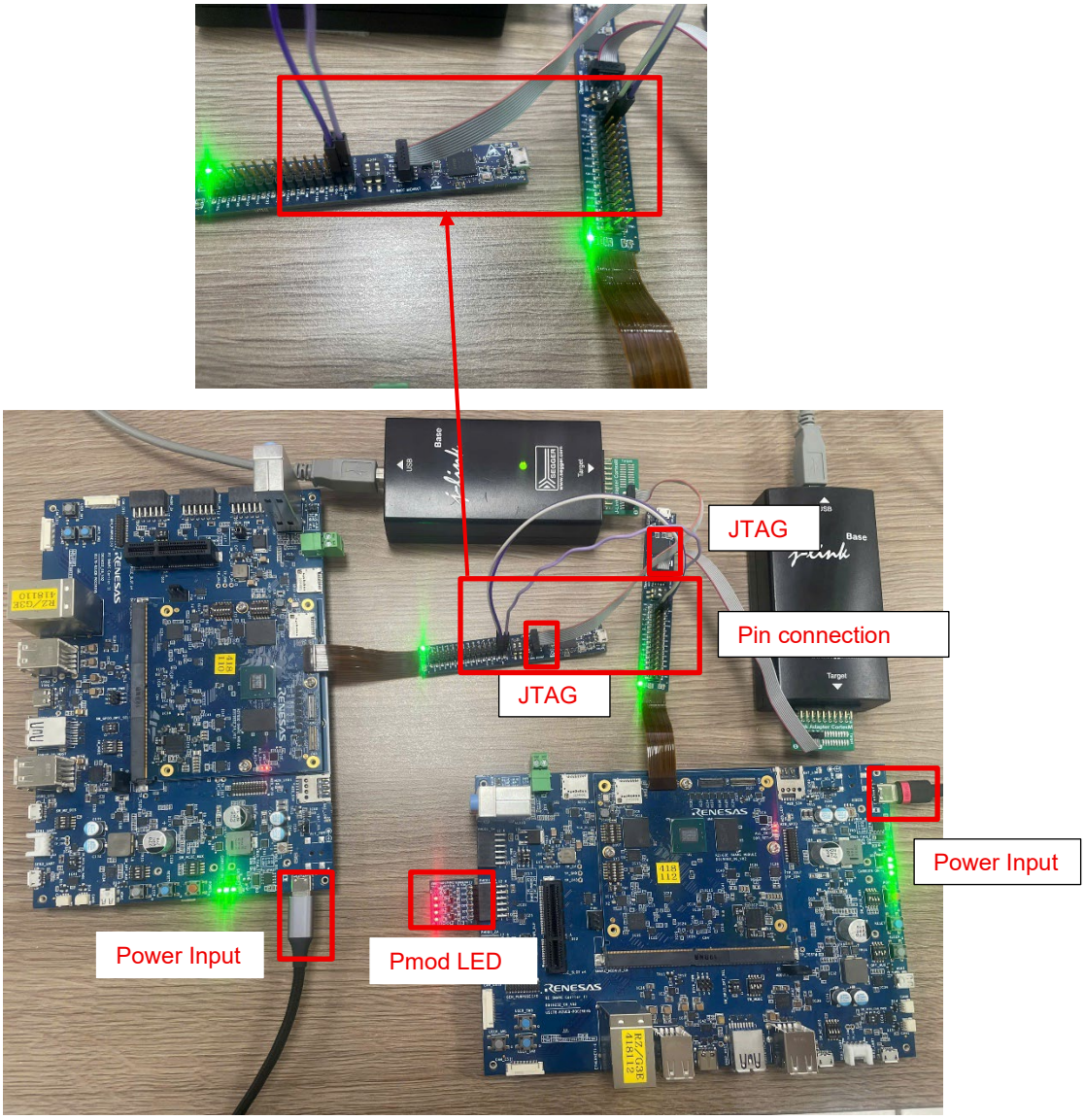


Figure 84: I3C_B Example - Board Setting (RZ/G3E)

Connect Pins on RZ Smarc Breakout between Master board and Slave board as follows:

RZ/G3E EVK		
Master board		Slave board
• Pin 3 - SCL (CN1)	---	Pin 3 - SCL (CN1)
• Pin 4 - SDA (CN1)	---	Pin 4 - SDA (CN1)
• Pin 2 - GND (CN1)	---	Pin 2 - GND (CN1)

Please set each DIP switch and jumpers as below.

Board	**RZ/G3E EVK (Master board)**	**RZ/G3E EVK (Slave board)**
Module board	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF	SW_CONFIG_1: OFF SW_CONFIG_2: OFF SW_CONFIG_3: OFF SW_CONFIG_4: OFF SW_CONFIG_5: OFF SW_CONFIG_6: OFF
Carrier board	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON	SW_MODE_1: OFF SW_MODE_2: ON SW_MODE_3: OFF SW_MODE_4: ON PMOD_PWR_SEL: Jumper connects 2-4

5.19.4 Operation

After running the example, the message below is displayed on console.

Run the master project after starting the slave project.
#Note : Please start the master within ten seconds after starting the slave.

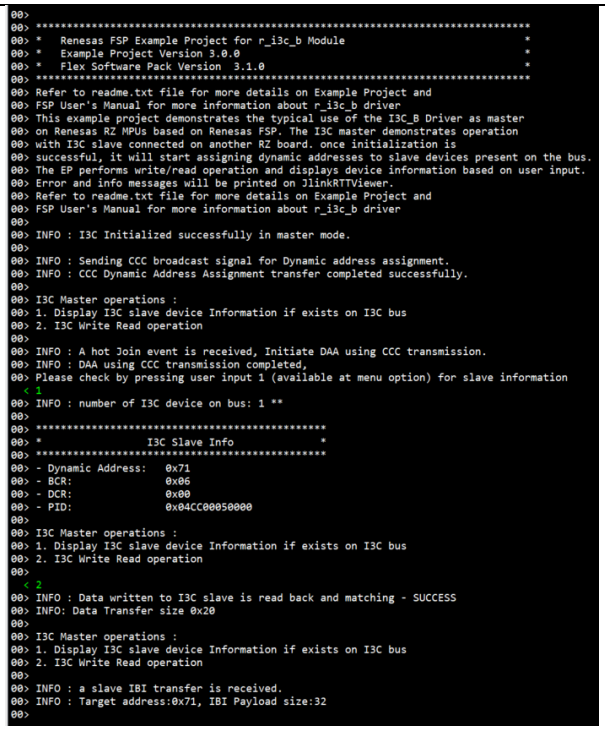


Figure 85: I3C_B Example – Operation (master board)

After pushing the USER_SW1 on the slave board, the IBI transfer request will be initiated to the master.
The following messages will be displayed:
INFO : User Pushbutton Pressed.
INFO : Initiate an IBI transfer request.
INFO : IBI transfer request initiated successfully.
INFO : IBI Write complete, transfer size: 0x20

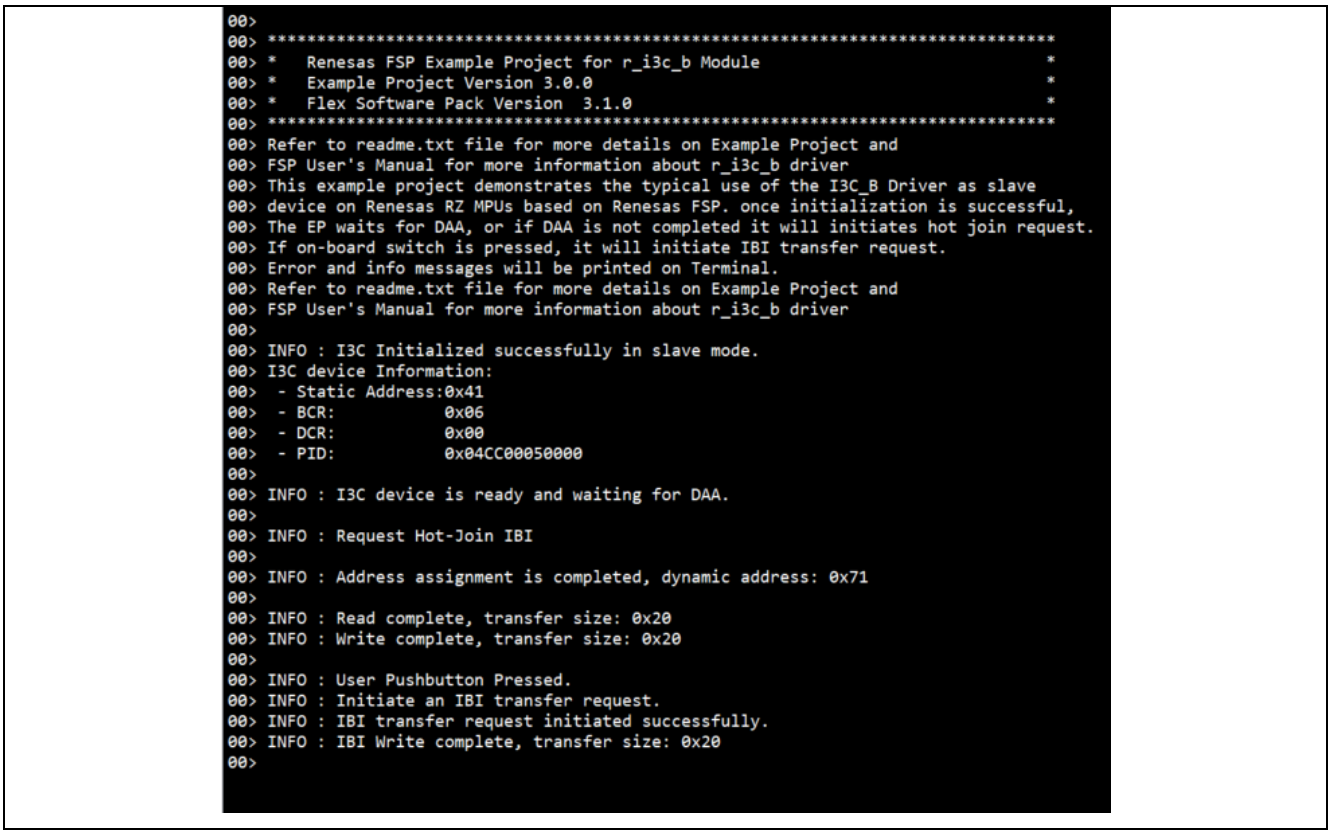


Figure 86: I3C_B Example – Operation (slave board)

6. References

FSP GitHub:	github.com/renesas/rzg-fsp
FSP User Manual:	renesas.github.io/rzg-fsp/
Getting Started Guide	Getting Started with RZ/G Flexible Software Package
FSP Example Projects:	RZ/G Multi-OS Package | Renesas
Evaluation Kit Manual:	RZ/G2L, RZ/V2L SMARC Module Board User's Manua
	RZ/G2LC SMARC Module Board RTK9744C22C01000BE User's Manual
	RZ/G2UL, RZ/A3UL, RZ/Five SMARC Module Board User's Manual
	RZ/G3S SMARC Module Board User's Manual
	RZ/G3E SMARC Module Board User's Manual
Knowledge Base:	Knowledge Base (renesas.com)
Renesas Support:	RZ/G2L Support (renesas.com)
	RZ/G2LC Support (renesas.com)
	RZ/G2UL Support (renesas.com)
	RZ/G3S Support (renesas.com)
	RZ/G3E Support (renesas.com)

Revision History

Rev.	Date	Description	
		Page	Summary
3.00	Jul.22.25	1	Added support for RZ/G3E.
		5, 10	Updated the description and figure based on the latest development environment.
		14 to 70	Added description for RZ/G3E EVK environment and new supported examples.
2.00	Feb.29.24	1	Added support for RZ/G3S.
		4, 8	Updated the description and figure based on the latest development environment.
		13, 14	Added description how to update the pin configuration in the case of changing the target board.
		17 to 48	Added description for RZ/G3S EVK environment.
1.00	Nov.27.23	–	First release document.

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
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 products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

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

Contact information

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