RZ/V 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/V FSP.

Target Device

- RZ/V2L
- RZ/V2H

Supported Kit

- RZ/V2L Evaluation Board Kit.
- RZ/V2H 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/V FSP Example projects are designed to operate using Evaluation Board Kit for RZ/V2L and RZ/V2H 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
- Ubuntu 20.04 LTS
- RZ/V FSP v2.0.1
- e² studio 2024-07
- SEGGER J-Link RTT Viewer v7.96e

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/rzv-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 7.96e 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/V FSP Example Projects" from below.

GitHub - renesas/rzv-fsp-examples

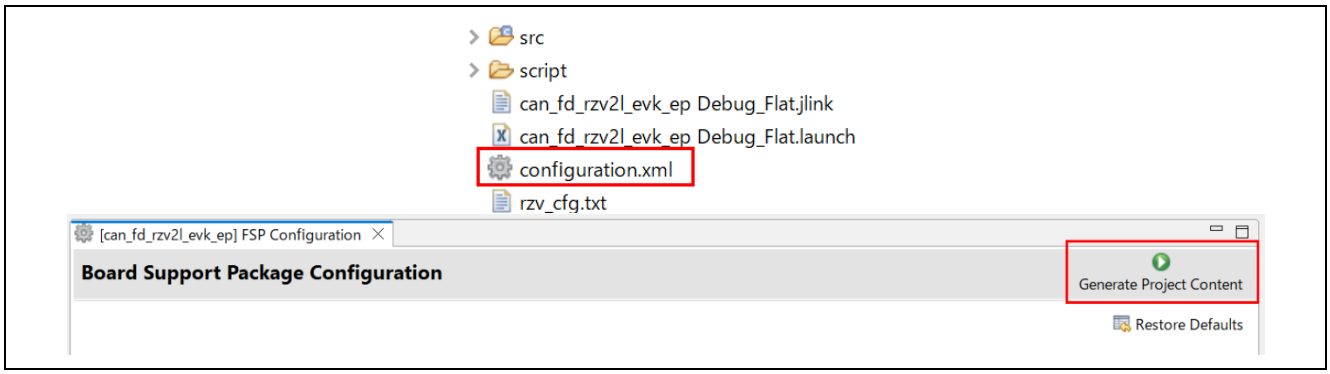
4.2 Running the project

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.*

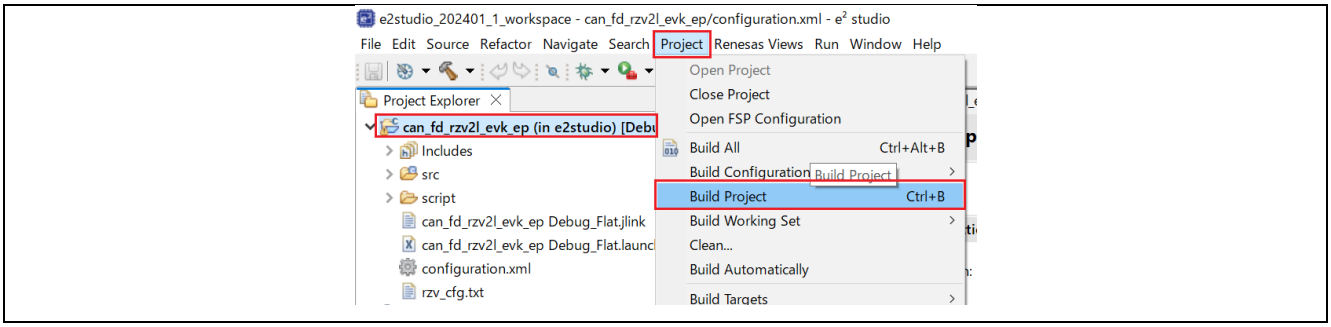
   Note: In the case of RZ/V2H, it is required to build “preceding_rzv2h_evk_cm33_ep” before build a CR8 project. Hence, please import and build the “.. \rzv2h_evk\preceding\preceding_rzv2h_evk_cm33_ep” when trying out the example for CR8.

2. Generate project content.
   Double clicks to open **configuration.xml** and then click **Generate Project Content**.

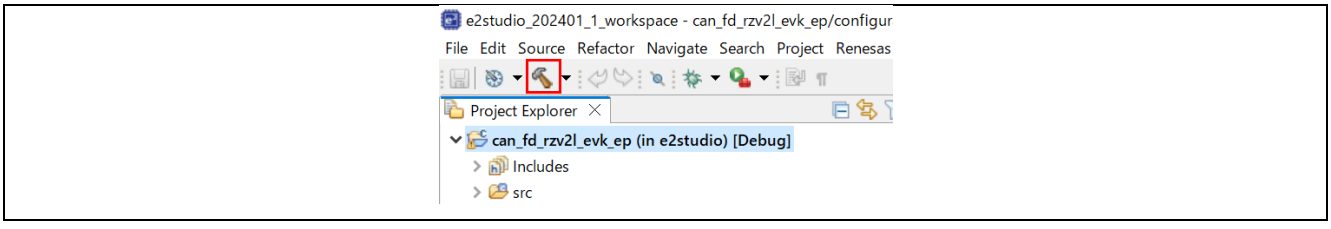


3. Build the project.
   There are three ways to build a project:

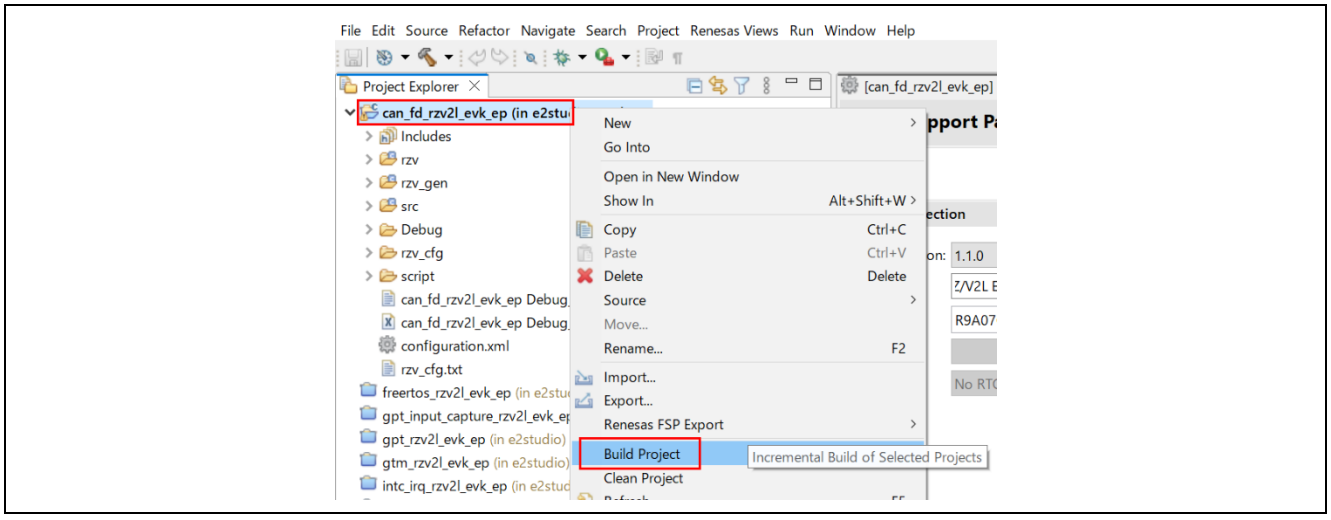
   a. Click on **Project** in the menu bar and select **Build Project**.



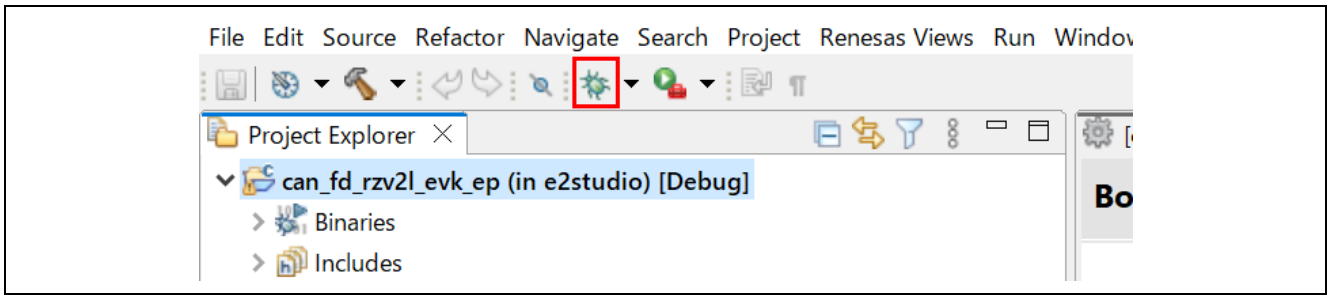
   b. Click on the hammer icon.



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

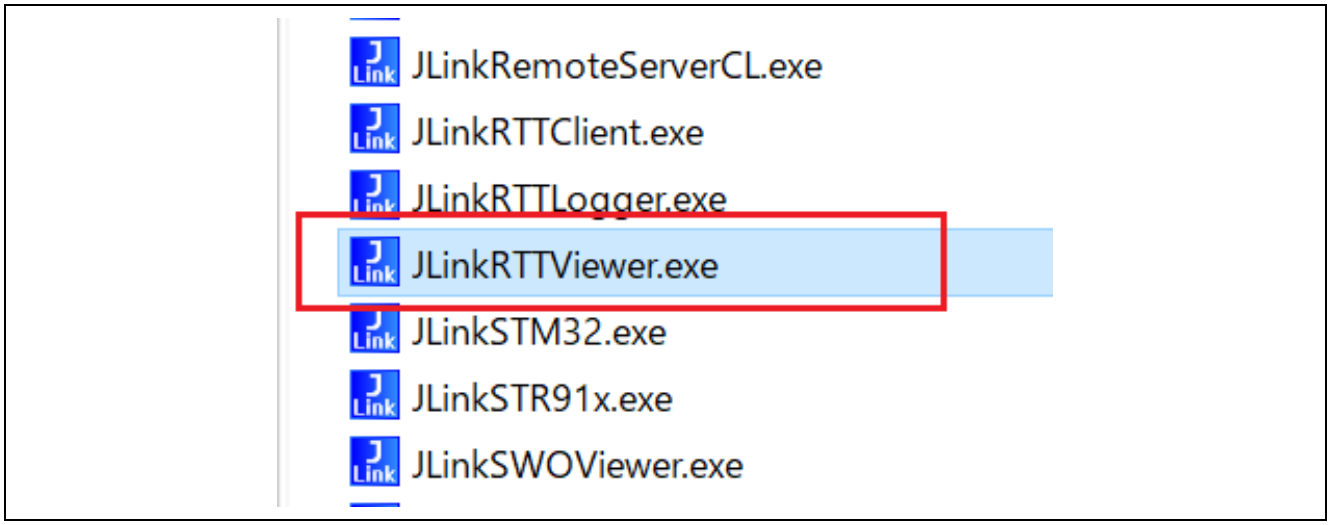


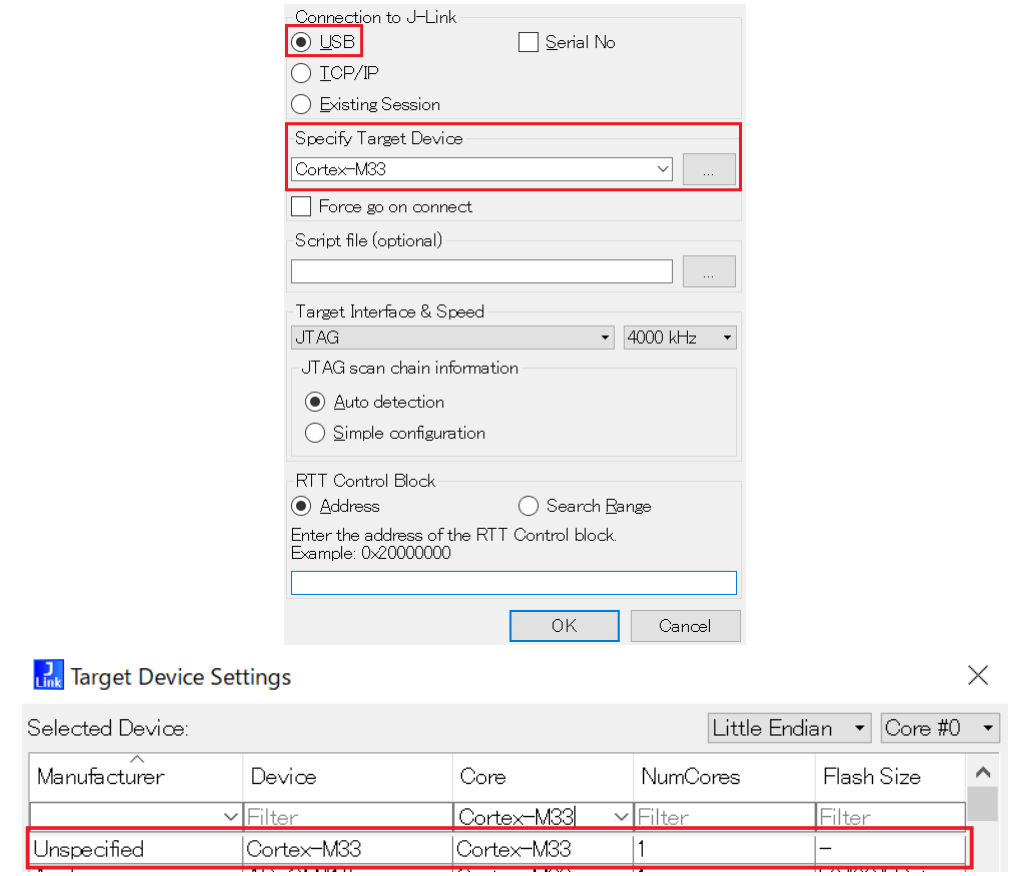
4. Downloading the project image to the board.
Click **Debug** to begin debugging the application.



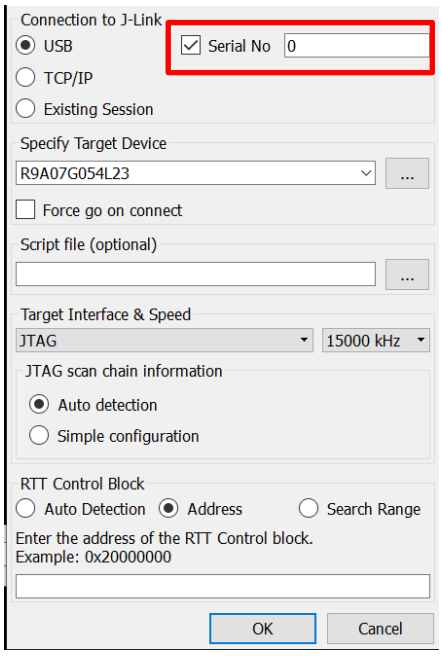
4.2.2 Connecting with J-Link RTT Viewer

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



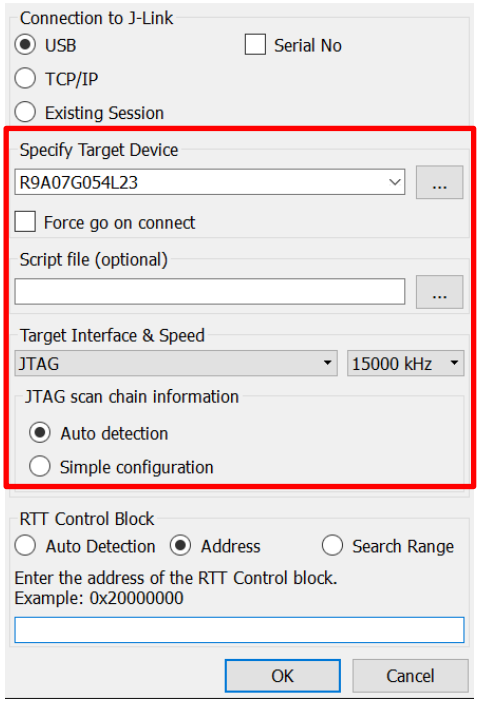


2. (This setting is optional.) If multiple kits are connected to the PC, make sure to choose the corresponding serial number. The default is 0.



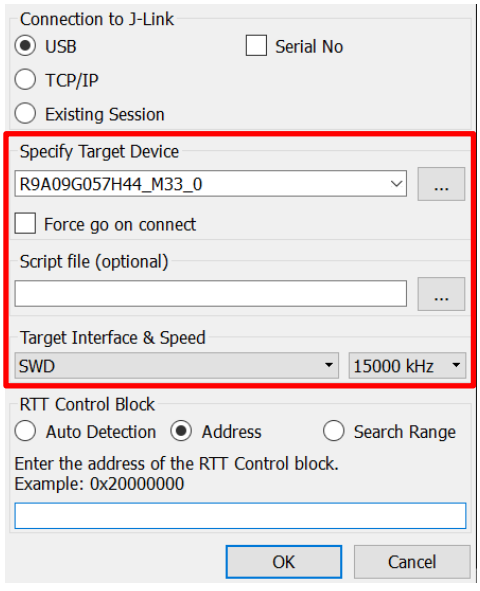
3. Select Target Interface & Speed

With respect to the RZ/V2L examples, please configure as below.

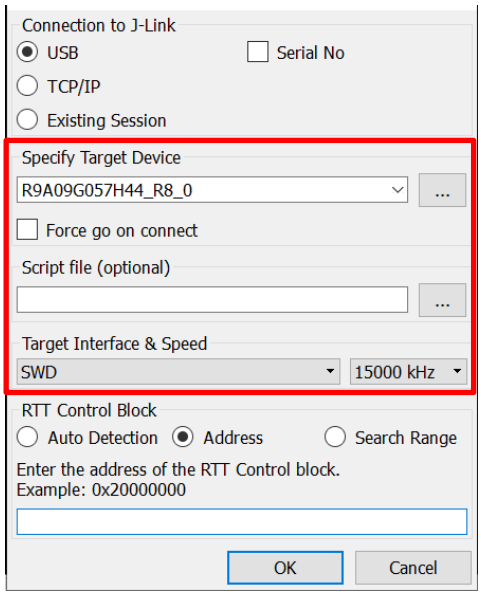


With respect to the other RZ/V2H examples, please configure as below.

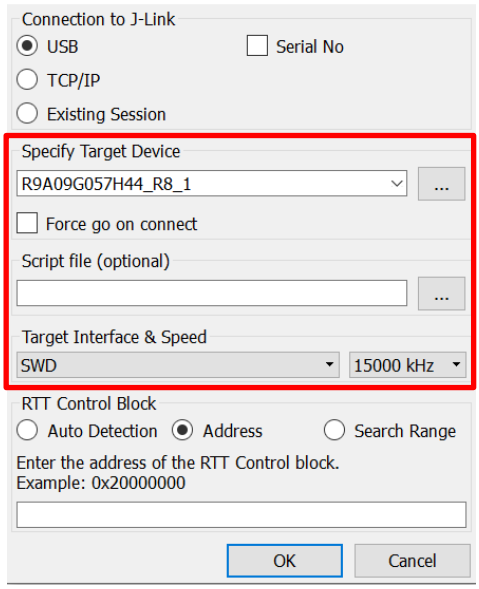
For CM33:



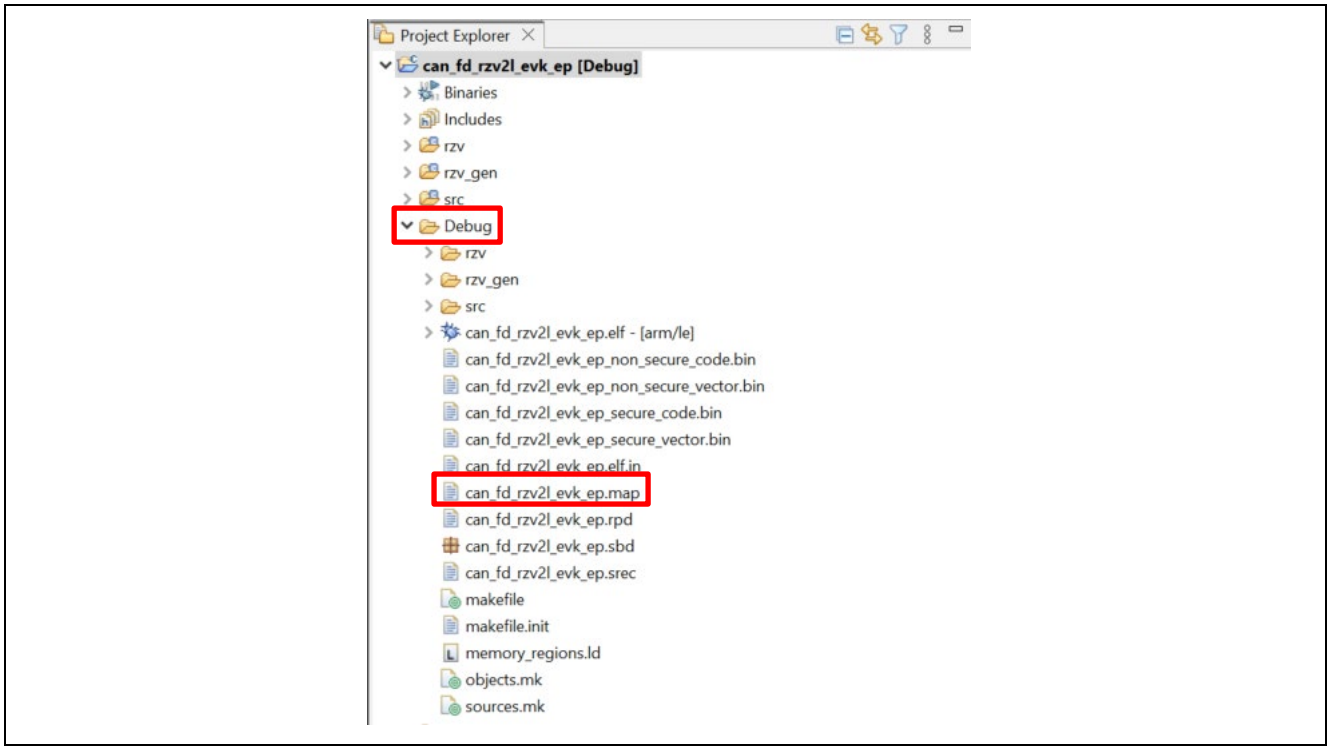
For CR8_0:

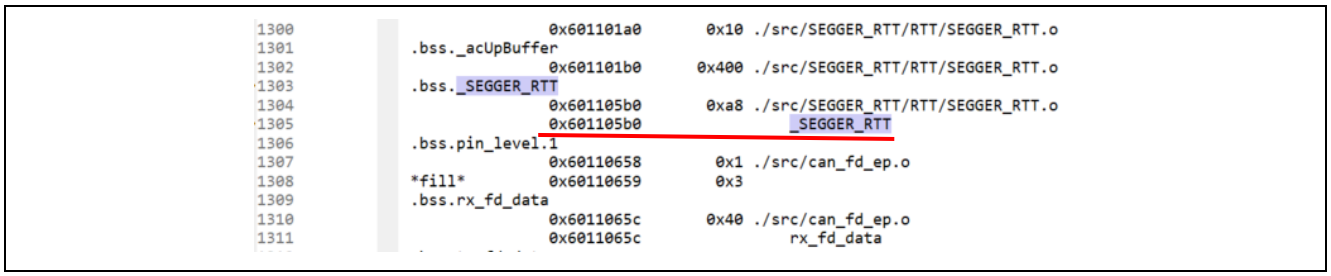


For CR8_1:

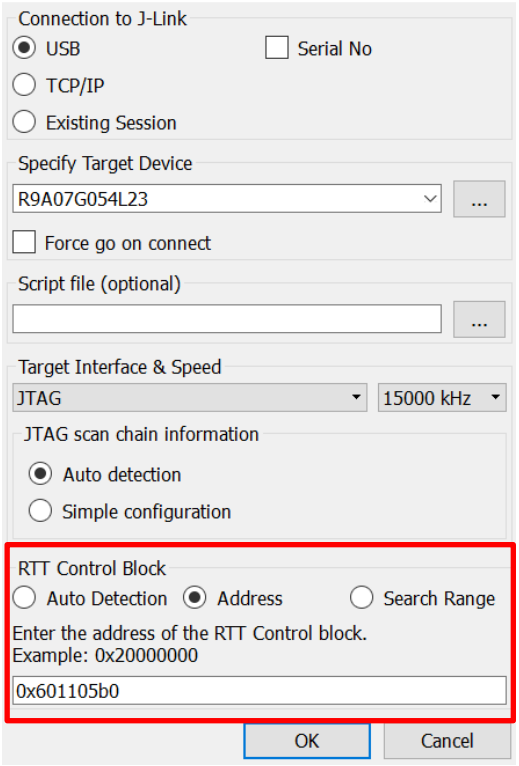


4. Configurate RTT Control Block.
   1. Segger RTT block address is required to connect J-Link RTT Viewer. Search **_SEGGER_RTT** variable in the map file, generated upon successfully building a configuration of an example project, which is by default located in the address space for SDRAM.

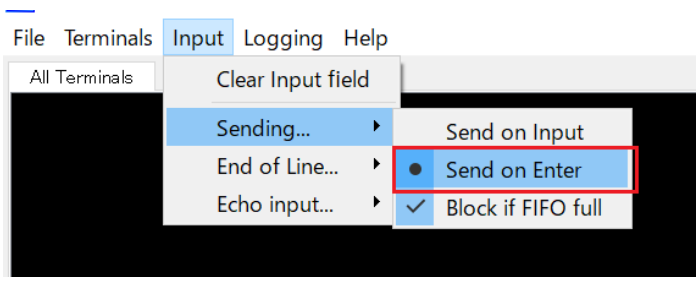




2. Select the Address and enter the address of RTT Control Block into textbox.

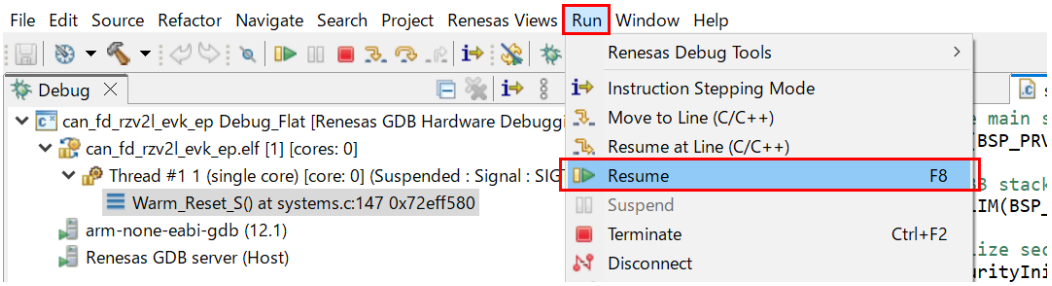


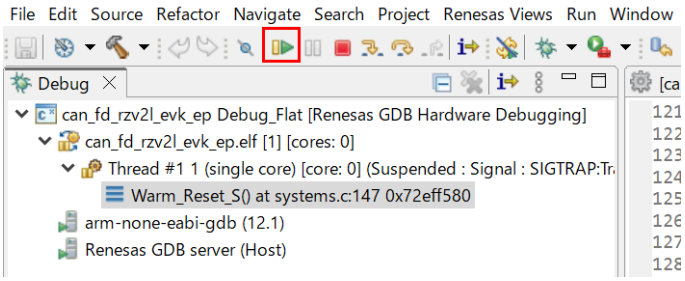
3. Click OK.
4. Click on the Input tab and change Sending option to Send on Enter. Every time input in entered, you must either press the Enter or Enter tab on the RTT viewer.



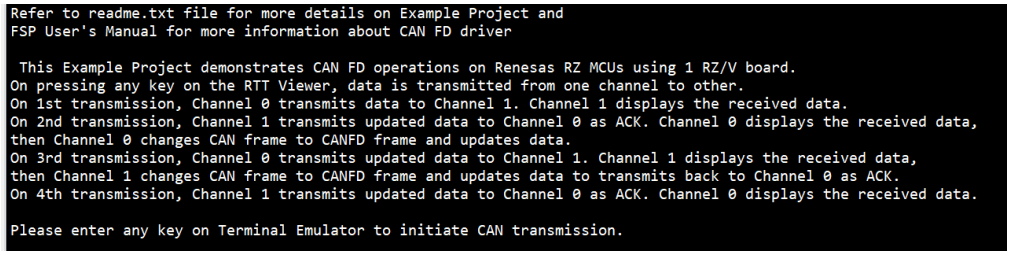
4.2.3 Running the project

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





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.



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 ADC_C

5.1.1 Project Overview

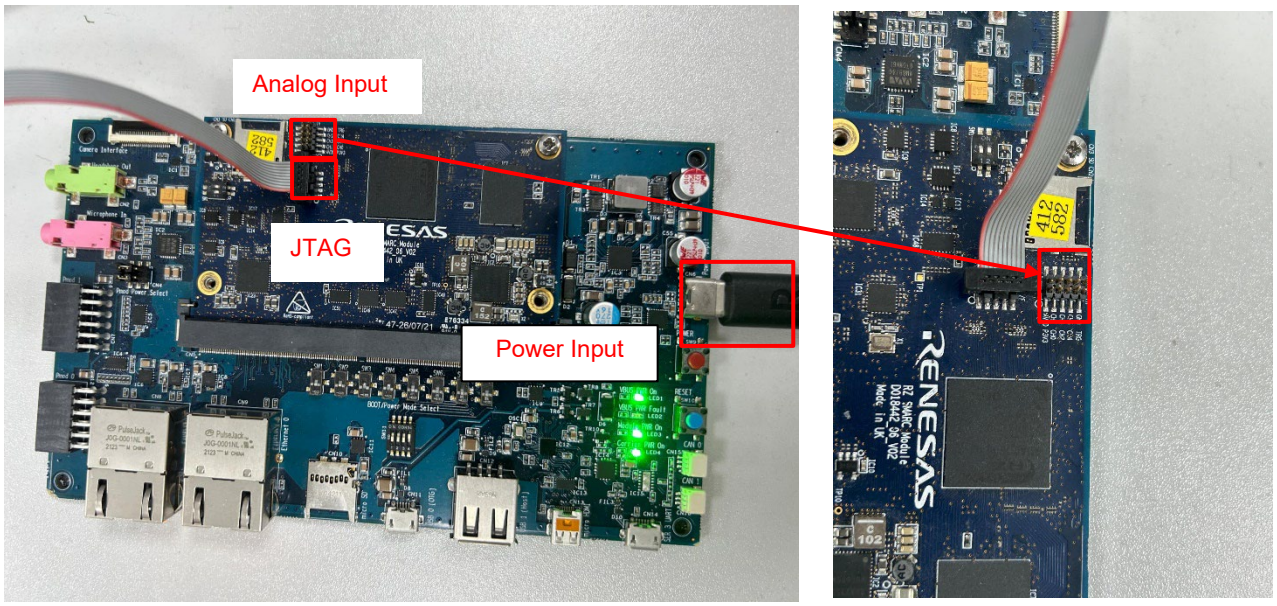
The example project demonstrates the typical use of the ADC HAL module APIs. The project initializes the ADC in single scan in accordance with FSP configuration. Result and ADC status is displayed on the RTTViewer.

5.1.2 Hardware Requirements

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

5.1.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK
Module board	SW1-1: OFF SW1-2: Don’t care.
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

5.1.4 Operation

Press 1 to Start ADC Scan via RTT viewer software and the result will be showed as below:

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_adc_c driver
00>
00> The project initializes the ADC in single scan mode in accordance
00> with FSP configuration. Results are displayed on JLinkRTTViewer.
00>
00> MENU to Select
00> Press 1 to Start ADC Scan
00> Press 2 to Stop ADC Scan(Only for Repeat mode)
00> User Input :
  < 1
00>
00> ADC Started Scan
00>
00> The Voltage Reading from ADC: 4095
00>
00> The ADC input voltage: 3.30
00>
```

5.2 ADC_E

5.2.1 Project Overview

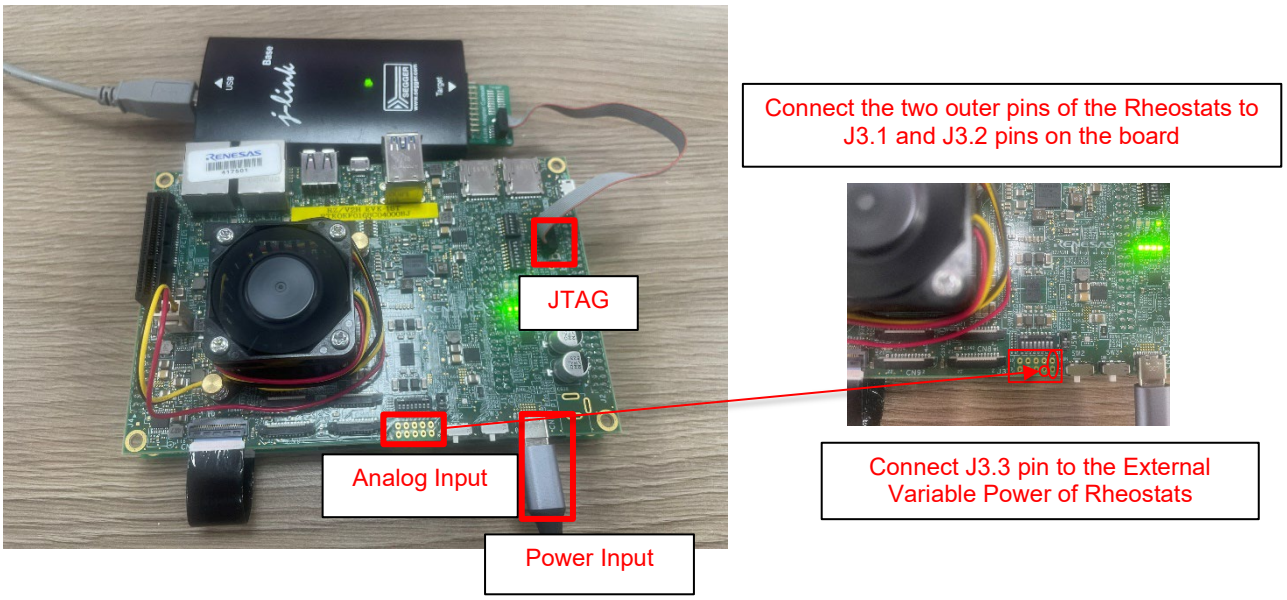
The example project demonstrates the typical use of the ADC HAL module APIs. The project initializes the ADC in single scan or repeat scan mode in accordance with FSP configuration. Result and ADC status is displayed on the RTTViewer.

5.2.2 Hardware Requirements

External hardware: Rheostats 10K

5.2.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

I

Please set each DIP switch and jumpers as below.

Board	RZ/V2H EVK
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

5.2.4 Operation

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

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_adc_e driver
00>
00> The project initializes the ADC in single scan mode in accordance
00> with FSP configuration. Results are displayed on JLinkRTTViewer.
00>
00> MENU to Select
00> Press 1 to Start ADC Scan
00> Press 2 to Stop ADC Scan(Only for Repeat mode)
00> User Input :
  < 1
00>
00> ADC Started Scan
00>
00> The Voltage Reading from ADC: 4095
00>
00> The ADC input voltage: 1.80
```

5.3 CANFD

5.3.1 Project Overview

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.

5.3.2 Hardware Requirements

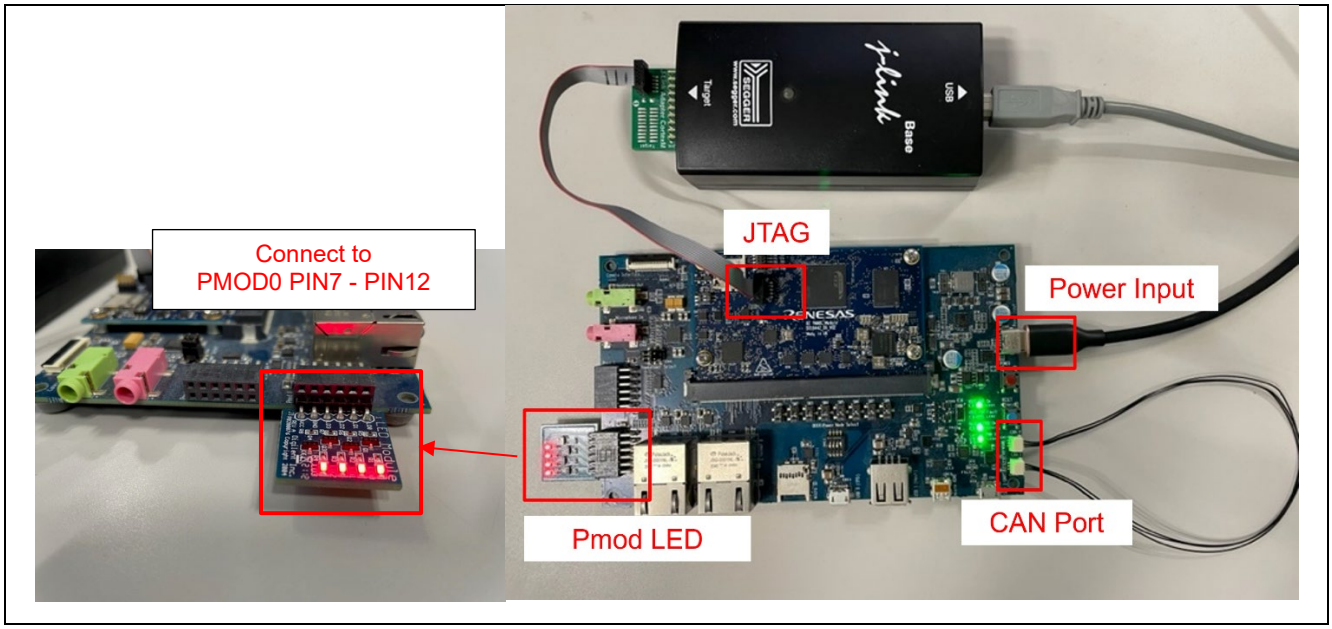
External hardware: Pmod LED

Also, The Evaluation Board Kit must have IC15 (to support CAN port) is on the carrier board.

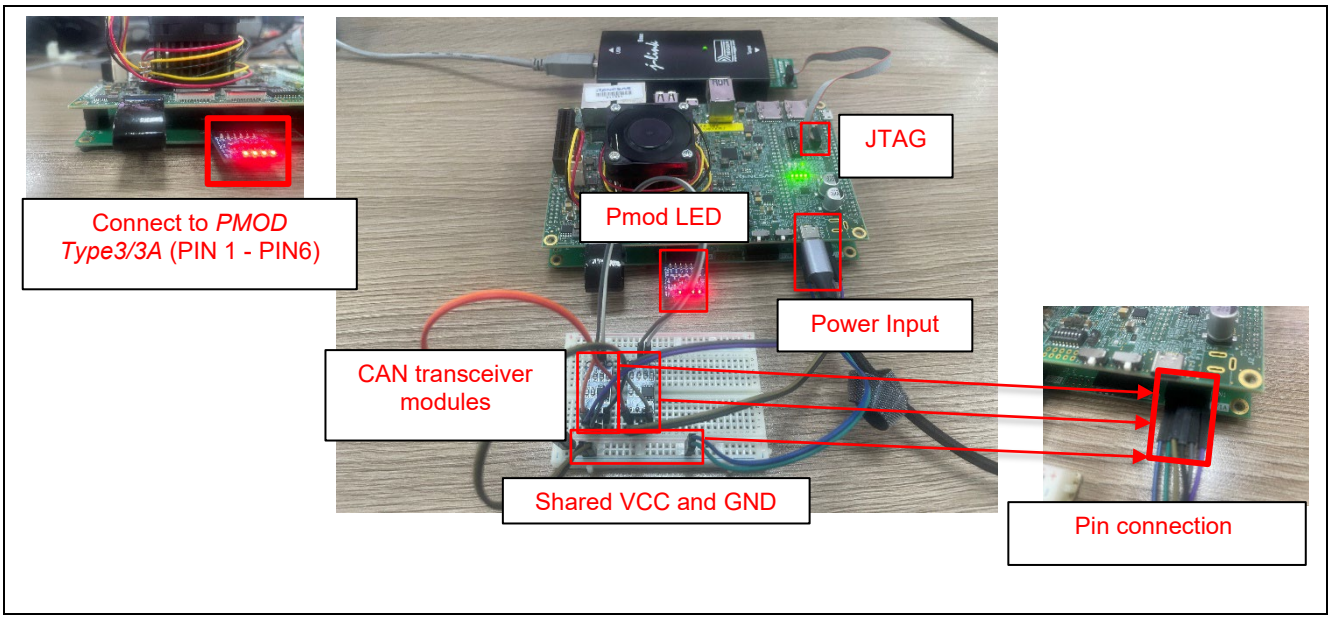
For RZ/V2H, two high-speed CAN transceiver modules/ICs are required.

5.3.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)



Board Setting (RZ/V2H)

Please set the connection between RZ/V2H EVK Expansion board and CAN transceiver modules as below.

RZ/V2H EVK
CTX0: Pmod 1/1A PIN1 - Tx Pin of CAN transceiver module 1.
CRX0: Pmod 1/1A PIN2 - Rx Pin of CAN transceiver module 1.
CTX1: Pmod 1/1A PIN3 - Tx Pin of CAN transceiver module 2.
CRX1: Pmod 1/1A PIN4 - Rx Pin of CAN transceiver module 2.
VCC: Pmod 1/1A PIN12 – VCC of 2 CAN transceiver modules.
GND: Pmod 1/1A PIN11 – GND of 2 CAN transceiver modules
CANH: Can_H Pin of CAN transceiver module 1- Can_H Pin of CAN transceiver module 2.
CANL: Can_L Pin of CAN transceiver module 1 - Can_L Pin of CAN transceiver module 2.

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.3.4 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/V 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
00>
00>
00> Please enter any key on Terminal Emulator to initiate CAN transmission.
```

5.4 RIIC Master

5.4.1 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.

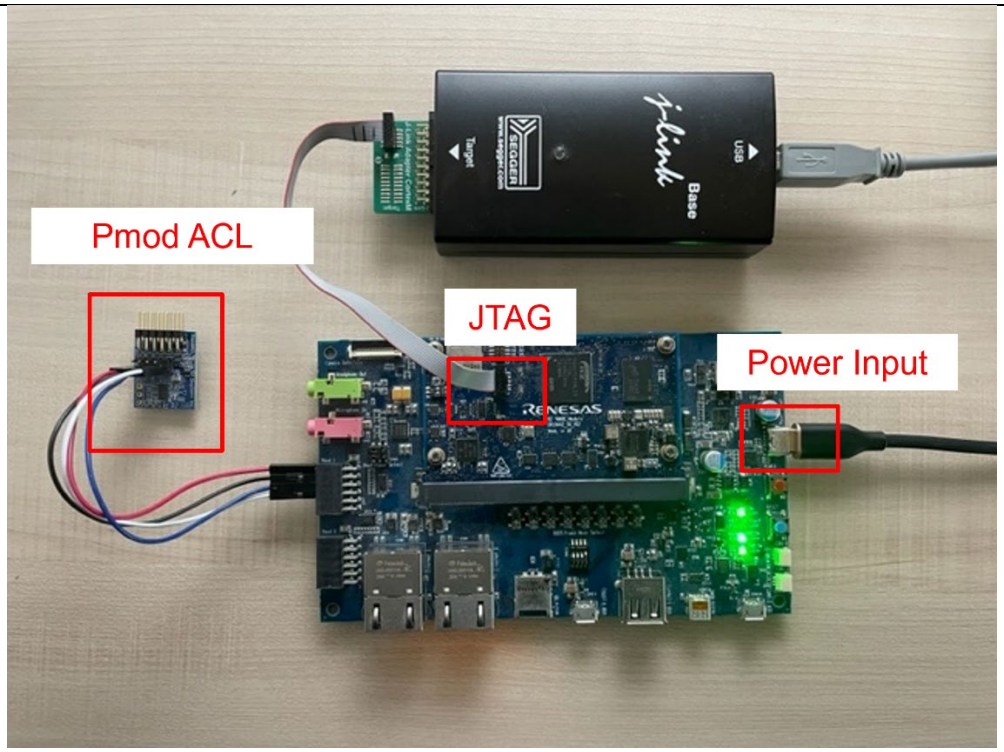
On powers up after establishing the connection of sensor with RZ/V board, it displays accelerometer axis data on RTTviewer. Any API/event failure will be displayed on RTTviewer.

5.4.2 Hardware Requirements

External hardware: Pmod ACL

5.4.3 Hardware Settings

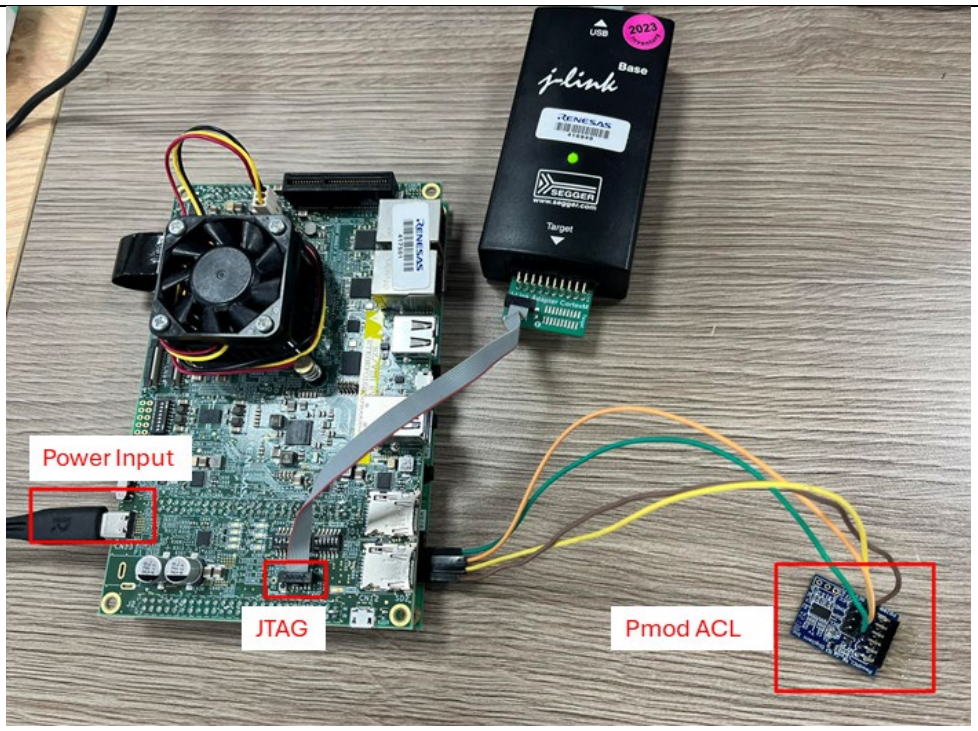
Please connect each hardware as below.



Board Setting (RZ/V2L)

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

Connection of Pmod ACL – RZV2L	
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



Board Setting (RZ/V2H)

Please set the connection between Pmod ACL and PMOD 6/6A on board as below.

Connection of Pmod ACL – RZV2H	
SCL:	Pmod ACL J2 PIN5 – PMOD 6/6A PIN3
SDA:	Pmod ACL J2 PIN6 – PMOD 6/6A PIN4
GND:	Pmod ACL J2 PIN7 – PMOD 6/6A PIN5
VCC:	Pmod ACL J2 PIN8 – PMOD 6/6A PIN6

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don't care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.4.4 Operation

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

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_riic_master driver
00>
00> This project utilizes PMOD ACL sensor as iic slave device
00> Upon successful initialization, MPU displays sensor axis data
00>
00> If SDA line is kept in LOW by any error
00> Please input 1 on J-Link RTT Viewer to Open bus.
00>
00>
00> X-axis = 254.00, Y-axis = 65533.00, Z-axis = 65510.00
00> X-axis = 65534.00, Y-axis = 65535.00, Z-axis = 65535.00
```

5.5 RIIC Slave

5.5.1 Project Overview

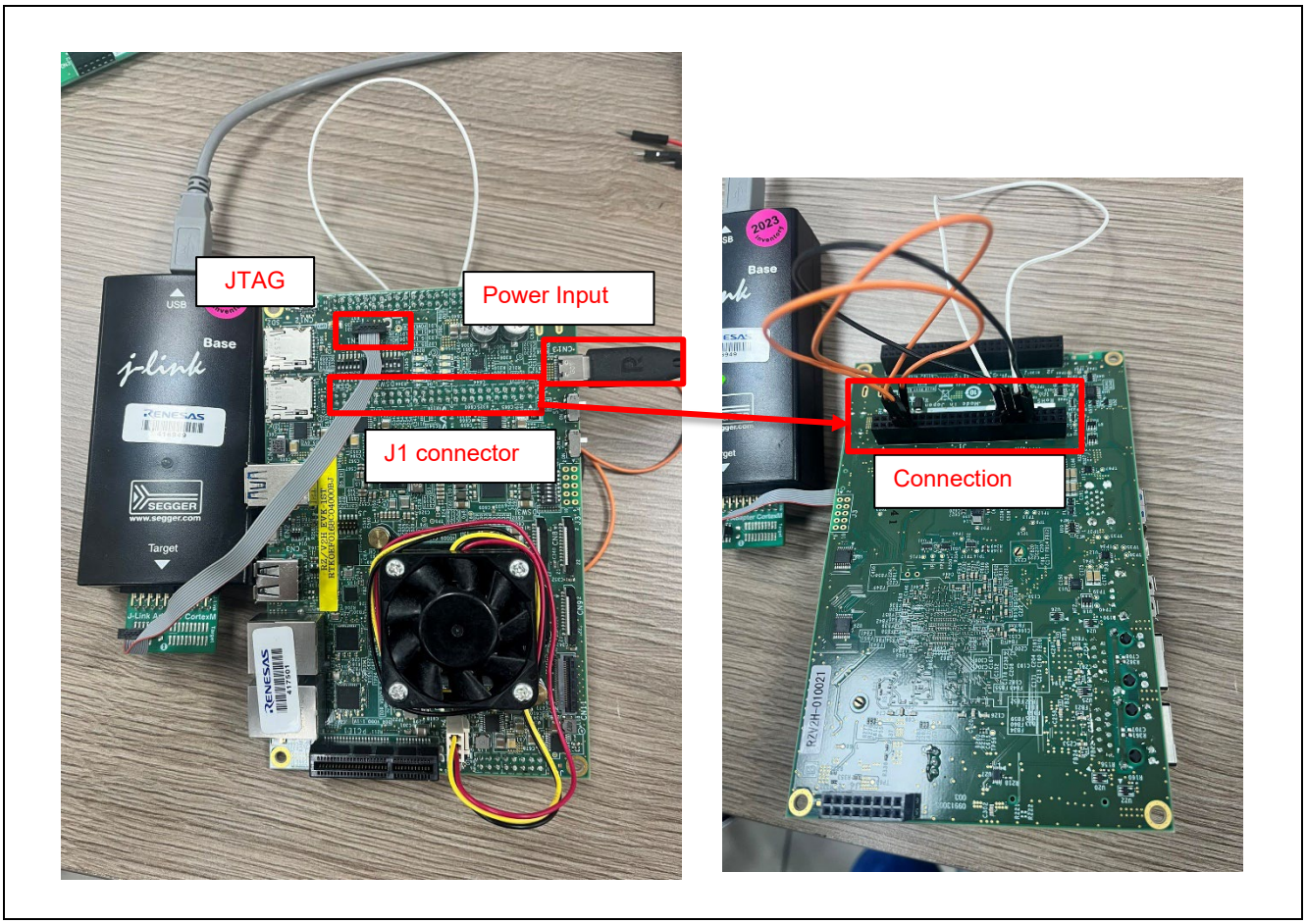
The example project demonstrates typical use of the RIIC slave HAL module APIs. The project initializes RIIC slave and IIC master module with standard rate and is made interfaced with loop-back mechanism. It performs Slave read and write operation continuously once initialization is successful. On successful 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.

5.5.2 Hardware Requirements

External hardware: None

5.5.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

Please set the Pin connection in GPIO Connector J1 RZV2H_Expansion Board as below.

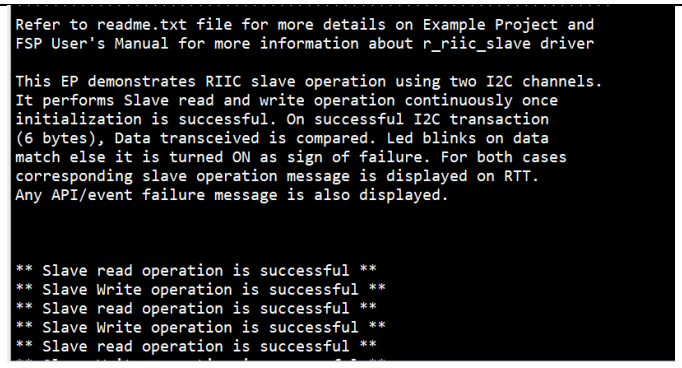
RZ/V2H EVK
Un-Connect the Secure Evaluation Board with Expansion Board. For Master and Slave : - GPIO Connector J1-41 - GPIO Connector J1-37 - GPIO Connector J1-39 - GPIO Connector J1-35. For LEDs on board : - GPIO Connector J1-7 - GPIO Connector J1-8 - GPIO Connector J1-9 - GPIO Connector J1-10.

Please set each DIP switch as below.

Board	**RZ/V2H EVK**
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

5.5.4 Operation

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



5.6 RSPI

5.6.1 Project Overview

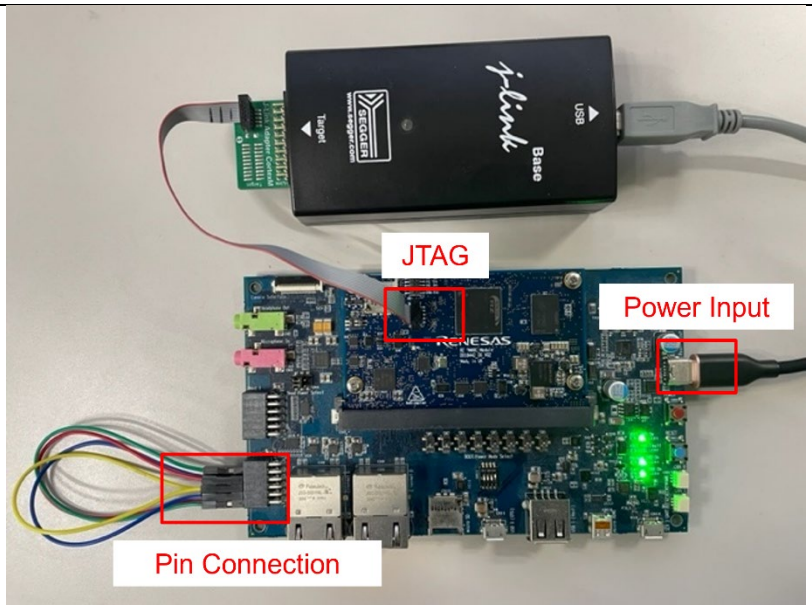
The example project demonstrates the typical use of the RSPI HAL module APIs. The project configures RSPI channels (Channel 0 and Channel 1) in Master and Slave mode. Once the module is initialized and the channels are configured, Master and Slave can transmit and receive data based on commands from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

5.6.2 Hardware Requirements

External hardware: None

5.6.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)

Please set the Pin Connection as below.

RZ/V2L EVK	
MISO:	PMOD0 PIN3 - PMOD0 PIN8
MOSI:	PMOD0 PIN2 - PMOD0 PIN10
CK:	PMOD0 PIN4 - PMOD0 PIN7
SSL:	PMOD0 PIN1 - PMOD0 PIN11

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK
Module board	SW1-1: OFF SW1-2: Don't care.
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

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_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
```

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> Select from the below Menu options
00>
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.
  < hello
00>
00> MASTER SEND: hello
00>
00> SLAVE RECEIVE: hello
00>
00> Master transmitted user input data to Slave
00>
00> Slave transmitted the data back to Master
00>
00> Master received data: hello
00>
00>
00> ** RSPI WRITE AND READ Demo Successful**
00>
00> Enter any other key to go back to the main menu
```

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.
  < Hello World !
00>
00> Enter text input for Slave buffer. Data size should not exceed 64 bytes.
  < This is RSPI example !
00>
00> Master buffer data transmitted to Slave
00>
00> Slave buffer data transmitted to Master
00>
00> Master received data: This is RSPI ex
00>
00> Slave received data: Hello World !
00>
00> ** RSPI WRITE_READ Demo Successful**
00>
00> Enter any other key to go back to the main menu
```

Enter "3" to exit the demo.

```
00> Press 1 for Write() and Read()
00> Press 2 for WriteRead()
00> Press 3 to Exit
  < 3
00> ** RSPI EXIT Demo Successful **
```

5.7 SCI_B_UART

5.7.1 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.

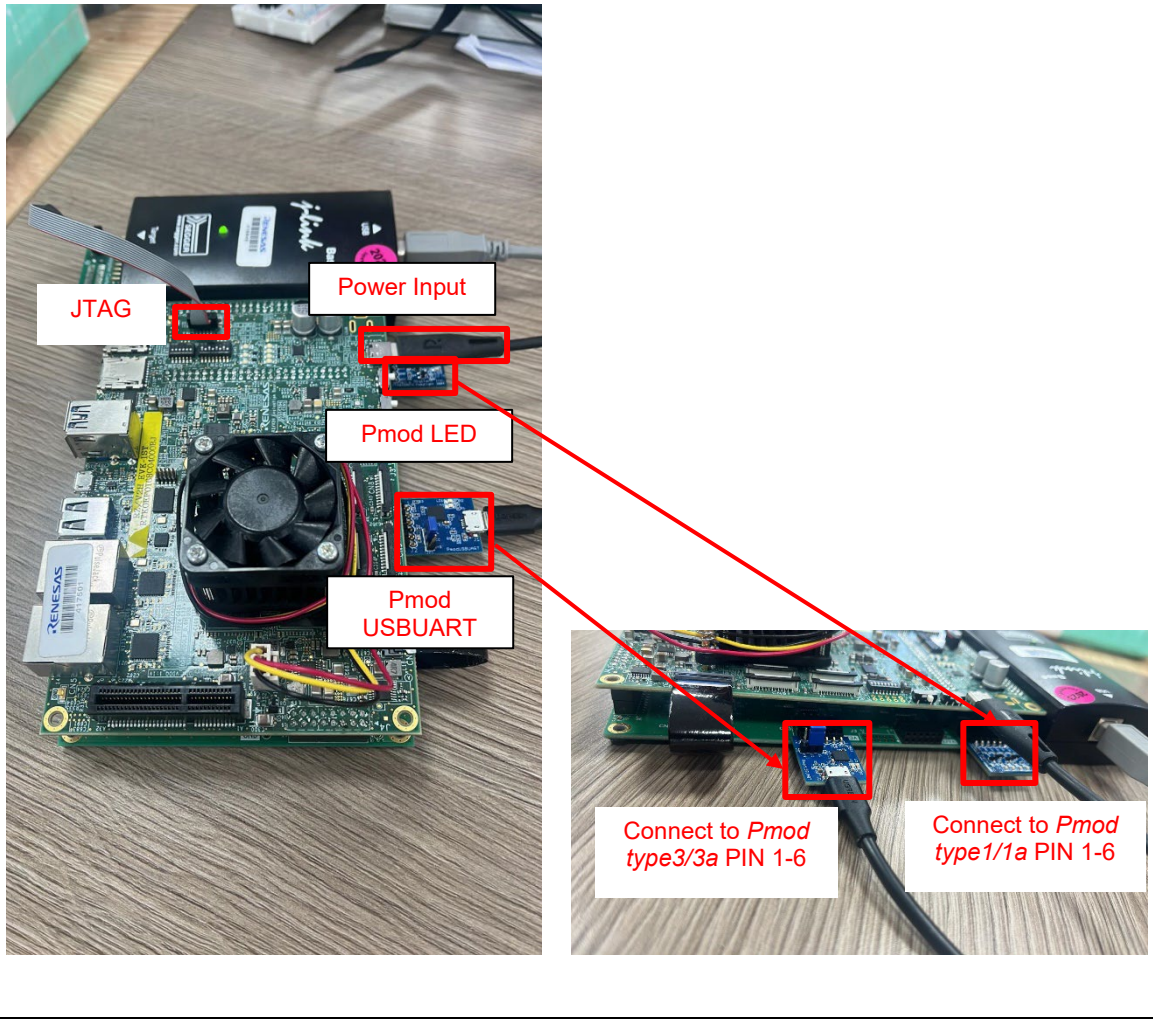
To see user input values on Serial terminal, enable local echo option.

5.7.2 Hardware Requirements

External hardware: Pmod USBUART and Pmod LED

5.7.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

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

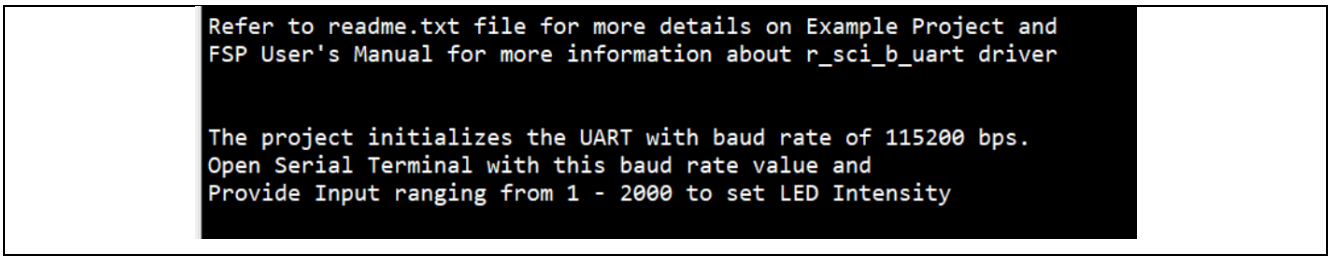
Connection of Pmod USBUART
RXD: Pmod USBUART J2 PIN2 – PMOD Type3/3A PIN2
TXD: Pmod USBUART J2 PIN3 – PMOD Type3/3A PIN3
GND: Pmod USBUART J2 PIN5 – PMOD Type3/3A PIN5
VCC: Pmod USBUART J2 PIN6 – PMOD Type3/3A PIN6

Please set each DIP switch as below.

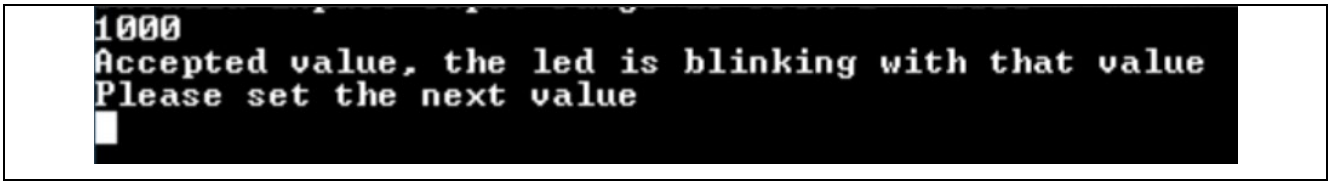
Board	**RZ/V2H EVK**
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

5.7.4 Operation

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



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



The configuration of Terminal Program is as below.

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

5.8 SCIF_UART

5.8.1 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.

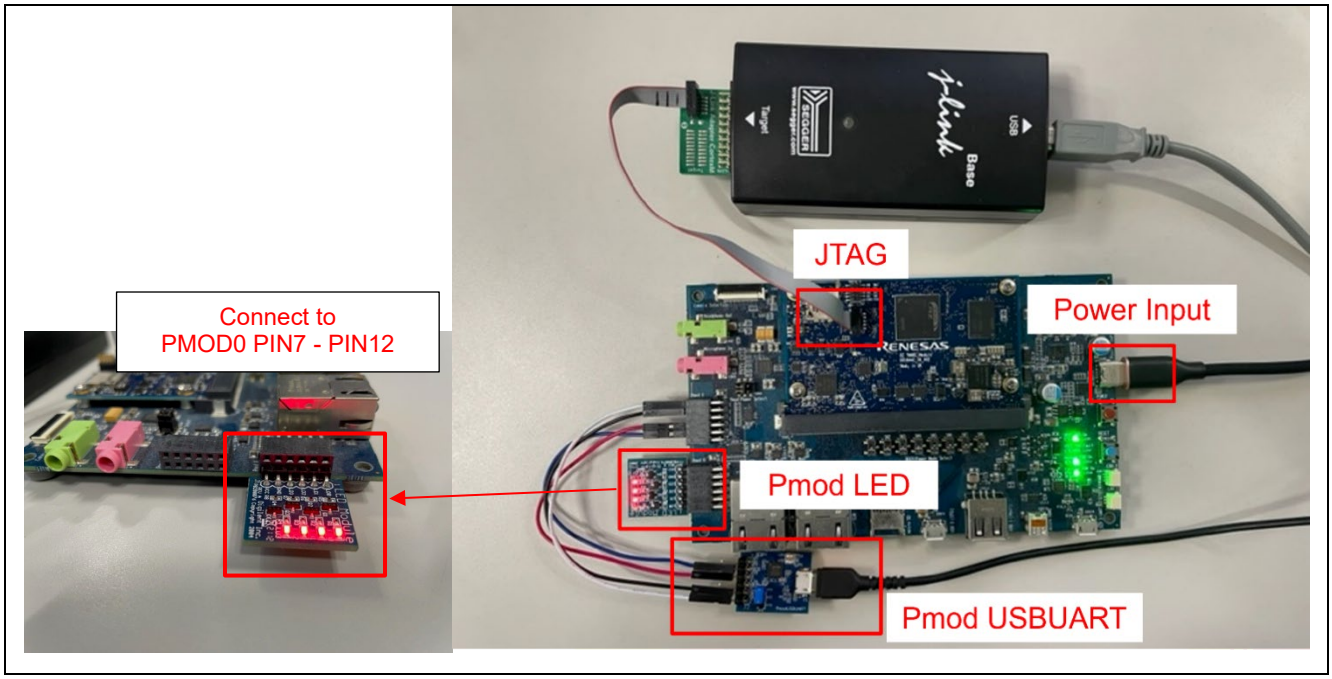
The range of input values are displayed on the JLinkRTTViewer. Any failure will also be displayed using JLinkRTTViewer. To see user input values on Serial terminal, enable local echo option.

5.8.2 Hardware Requirements

External hardware: Pmod USBUART and Pmod LED

5.8.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)

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

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK
Module board	SW1-1: OFF SW1-2: Don't care.
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

5.8.4 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
```

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

```
0
Invalid input. Input range is from 1 - 2000
2001
Invalid input. Input range is from 1 - 2000
299
Accepted value, the led is blinking with that value
Please set the next value
50
Accepted value, the led is blinking with that value
Please set the next value
20
Accepted value, the led is blinking with that value
Please set the next value
```

The configuration of Terminal Program is as below.

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

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

5.9 SPI_B

5.9.1 Project Overview

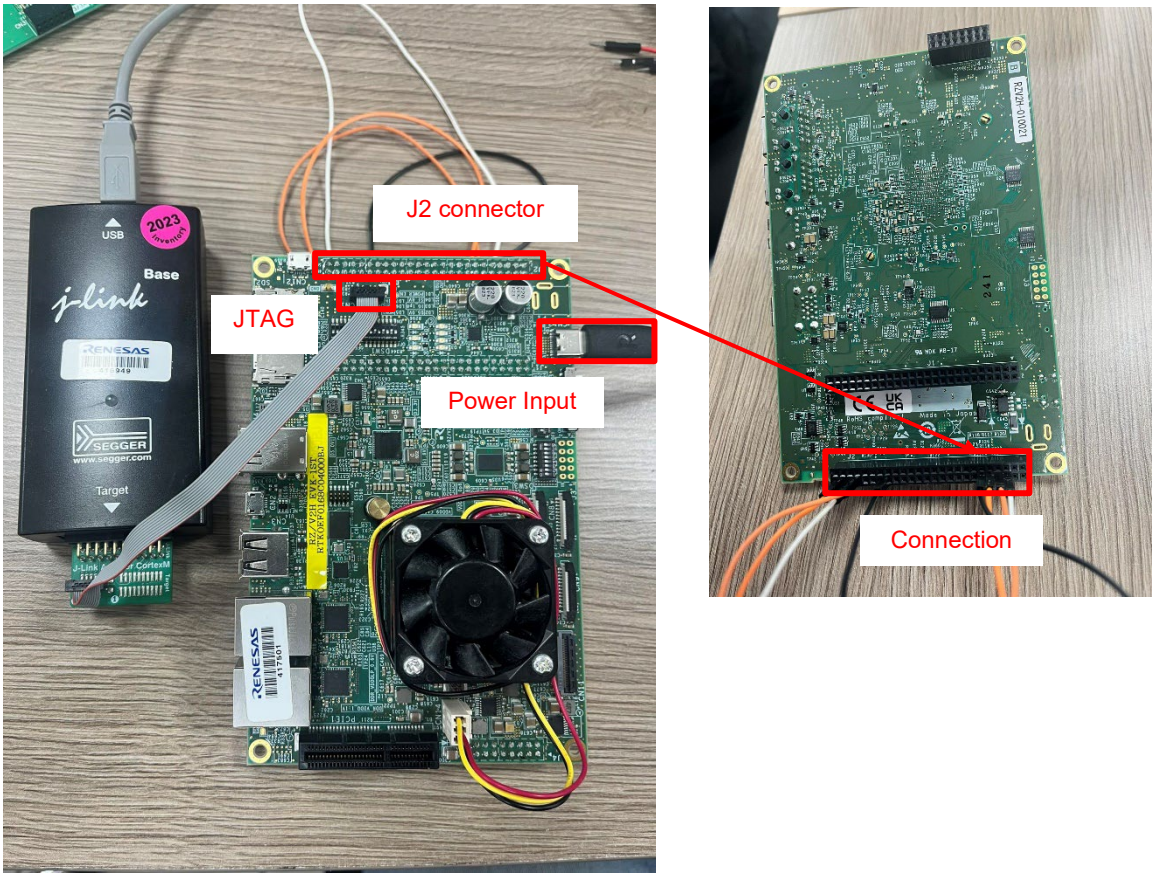
The example project demonstrates the typical use of the SPI_B HAL module APIs.The project configure SPI_B channels (Channel 0 and Channel 2) in Master and Slave mode. Once the module is initialised and the channels are configured, Master and Slave can transmit and receive data based on commands from user sent through JLinkRTTViewer. This project supports the DMAC feature. In cases where this feature is not used in the application, please invalid the DMAC feature by the following steps. - Set DMAC Support to Disabled on the main stack of the driver in FSP Configuration. - Remove the DMAC sub stacks that linked to the main stack.

5.9.2 Hardware Requirements

External hardware: None

5.9.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

Please set each DIP switch as below.

Board	RZ/V2H EVK
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

Please set the Pin connection in GPIO Connector J1 RZV2H_Expansion Board as below.

RZ/V2H EVK			
Un-Connect the Secure Evaluation Board with Expansion Board. On the J2 header GPIO connector (Secure Evaluation board):			
		Master (SPI 0)	Slave (SPI 2)
MISOA	---->	P91(Pin 8) -	PB3 (Pin 44)
MOSIA	---->	P90(Pin 6) -	PB4 (Pin 46)
RSPCKA	---->	P92(Pin 10) -	PB5 (Pin 48)
SSLA0	---->	P93(Pin 12) -	PA7 (Pin 36)

5.9.4 Operation

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

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

The project initializes SPI_B driver and configures SPI_B channels
in Master and Slave mode. After initialization, master and slave
can transmit and receive data based on the commands from user.
Refer to the MPU User Manual for valid bit rates and corresponding
clock settings.

 ** SPI_B INIT SUCCESSFUL **

Select from the below Menu options

Press 1 for Write() and Read()
Press 2 for WriteRead()
Press 3 to Exit
```

Enter "1" to run Write() and Read().

Enter a data to use the demo.

Output a data you enter.

If the demo is succeeded, output the data Master received and the message "** SPI_B WRITE AND READ Demo Successful **".

```
1

Enter text input for Master buffer. Data size should not exceed 64 bytes.
a

MASTER SEND: a

SLAVE RECEIVE: a

Master transmitted user input data to Slave

Slave transmitted the data back to Master

Master received data: a


** SPI_B WRITE AND READ Demo Successful**
```

Enter "2" to run WriteRead().

Enter data to use the demo for Master and Slave.

Output data you enter.

If the demo is succeeded, output the data Master and Slave received and the message "** SPI_B WRITE AND READ Demo Successful **".

```
2

Enter text input for Master buffer. Data size should not exceed 64 bytes.
abcdefghijklmnopqrstuvwxyz012345

Enter text input for Slave buffer. Data size should not exceed 64 bytes.
ABCDEFGHIJKLMNOPQRSTUVWXYZ012345

Master buffer data transmitted to Slave

Slave buffer data transmitted to Master

Master received data: ABCDEFGHIJKLMNOPQRSTUVWXYZ012345

Slave received data: abcdefghijklmnopqrstuvwxyz012345
** SPI_B WRITE_READ Demo Successful**
```

Enter "3" to exit the demo.

```
3
** SPI_B EXIT Demo Successful **

Exiting SPI_B demo....
```

5.10 GPT (Input Capture)

5.10.1 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.

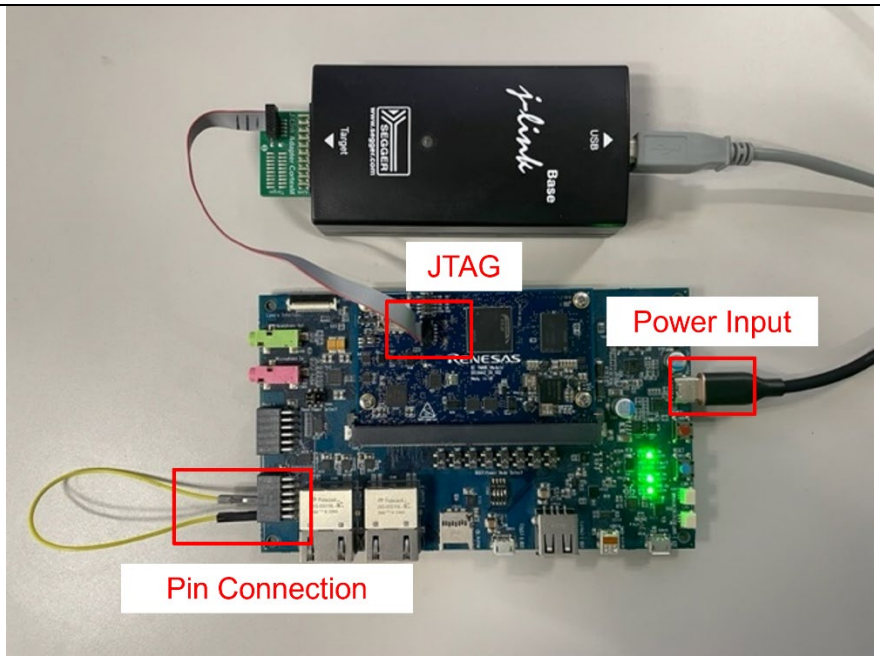
GPT3 counts the event pulse received at its input. Based on the period and capture event, the time period of pulse is calculated and displayed on RTTViewer.

5.10.2 Hardware Requirements

External hardware: None

5.10.3 Hardware Settings

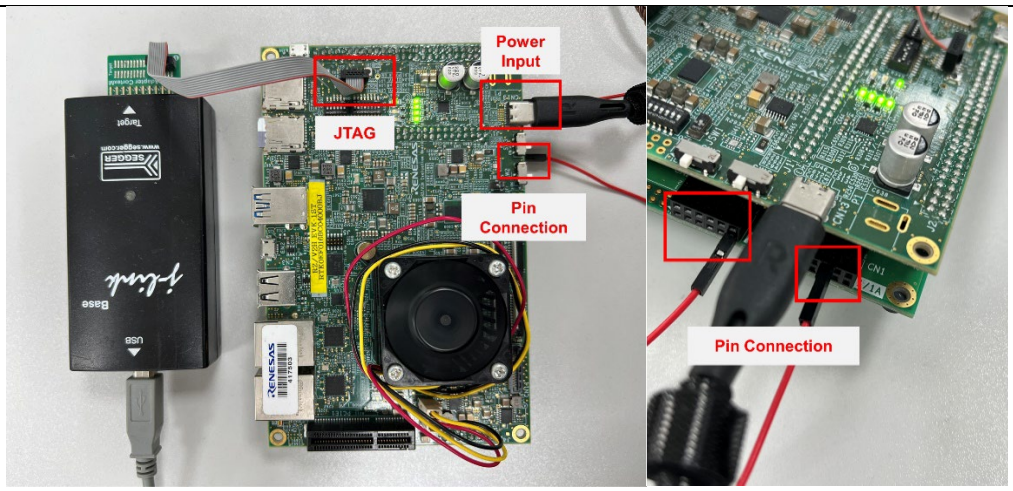
Please connect each hardware as below.



Board Setting (RZ/V2L)

Please set the Pin Connection as below.

RZ/V2L EVK
PMOD0 PIN7 (GPT ch4) - PMOD0 PIN9 (GPT ch3)



Board Setting (RZ/V2H)

Please set the Pin Connection as below.

RZ/V2H EVK Ver1
PMOD1/1A PIN 3 (GPT ch3) - PMOD2/2A pin 7 (GPT ch4)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.10.4 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 Example Project demonstrates the functionality of GPT Input
00> 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
```

5.11 GPT (PWM)

5.11.1 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.

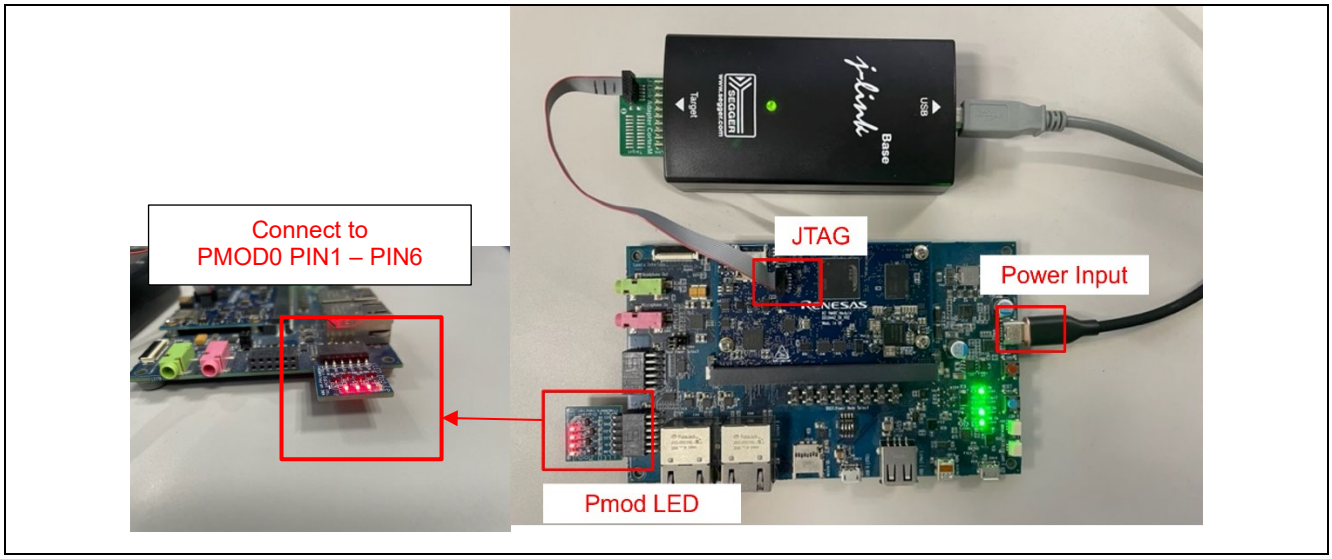
Any failure will also be displayed on JLinkRTTViewer.

5.11.2 Hardware Requirements

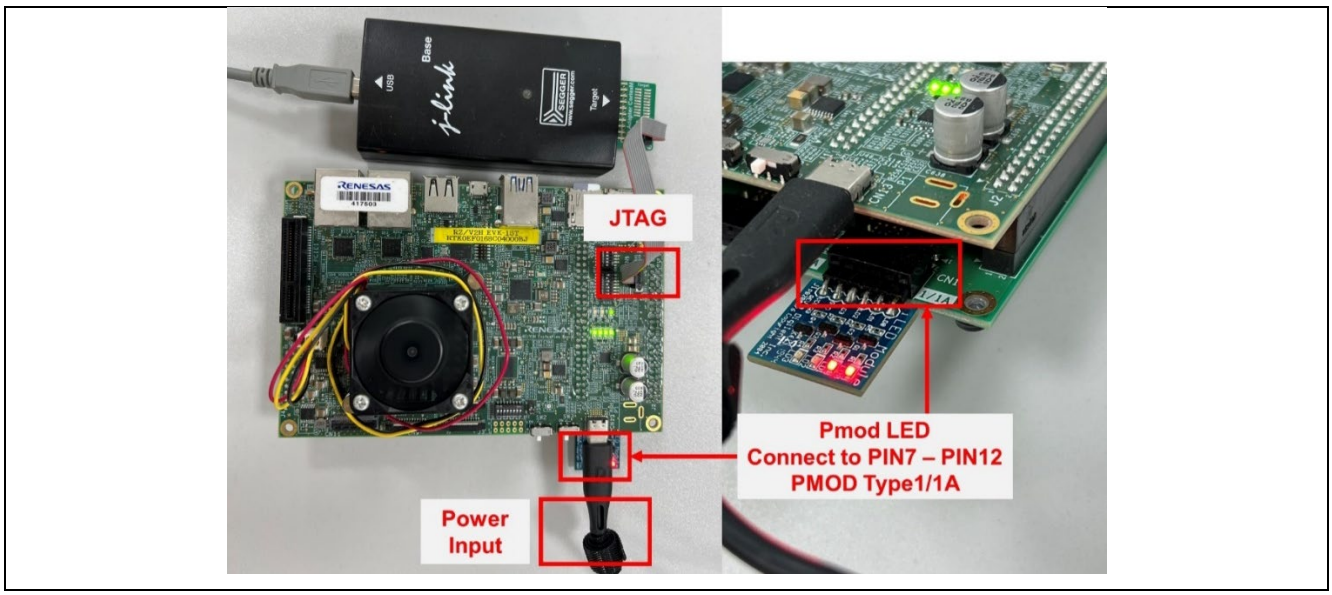
External hardware: Pmod LED

5.11.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)



Board Setting (RZ/V2H)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.11.4 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
00> mode based on user input from the displayed menu options.
00> In periodic mode, user can enter the time period within the
00> permitted ranges to change the frequency of the user LED.
00> In PWM mode, user can enter the duty cycle within the
00> specified range to adjust the intensity of the user LED.
00> In ONE SHOT mode, Output will be displayed on JlinkRTTViewer
00> 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
```

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
00>
00> Menu Options
```

In PWM 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>
```

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>
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> ONE-SHOT Timer Instance Closed, If Already Opened
00> Opened Timer in ONE-SHOT Mode
00> Started Timer in ONE-SHOT Mode
00>
00>  Timer Expired in One-Shot Mode
00>
```

5.12 GTM

5.12.1 Project Overview

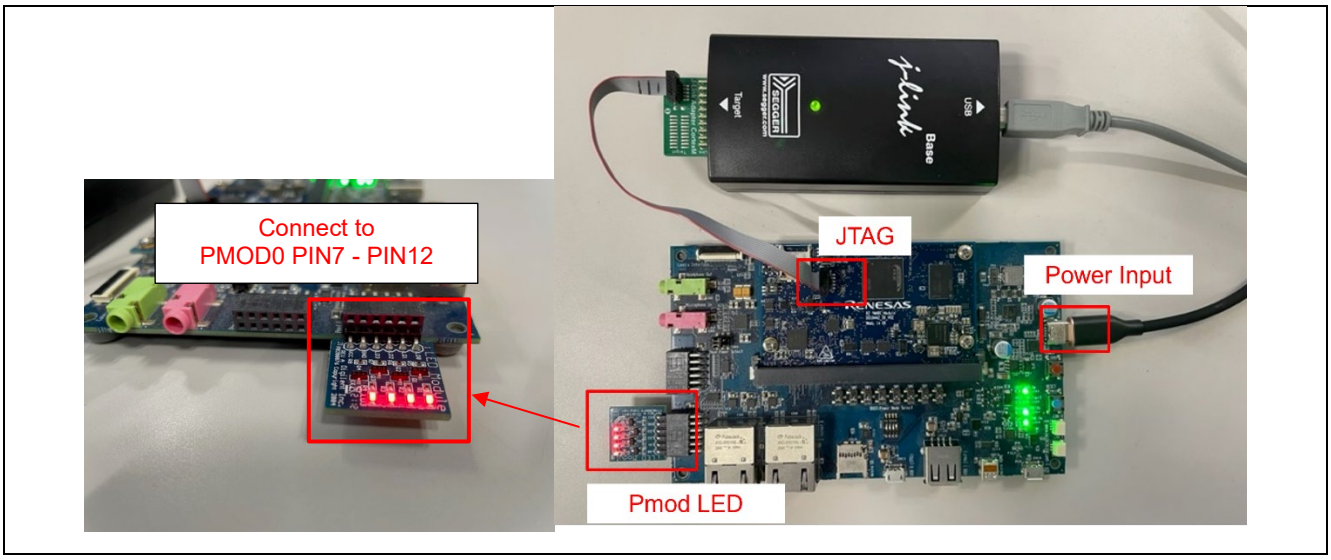
This Example Project demonstrates the functionality of GTM in periodic mode and one-shot mode. On providing any input on the RTTviewer, GTM channel 2 starts in one-shot mode. GTM channel 1 starts in periodic mode when GTM channel 2 expires. Timer in periodic mode expires periodically at a time period specified by user and toggles the LED1 on Pmod LED.

5.12.2 Hardware Requirements

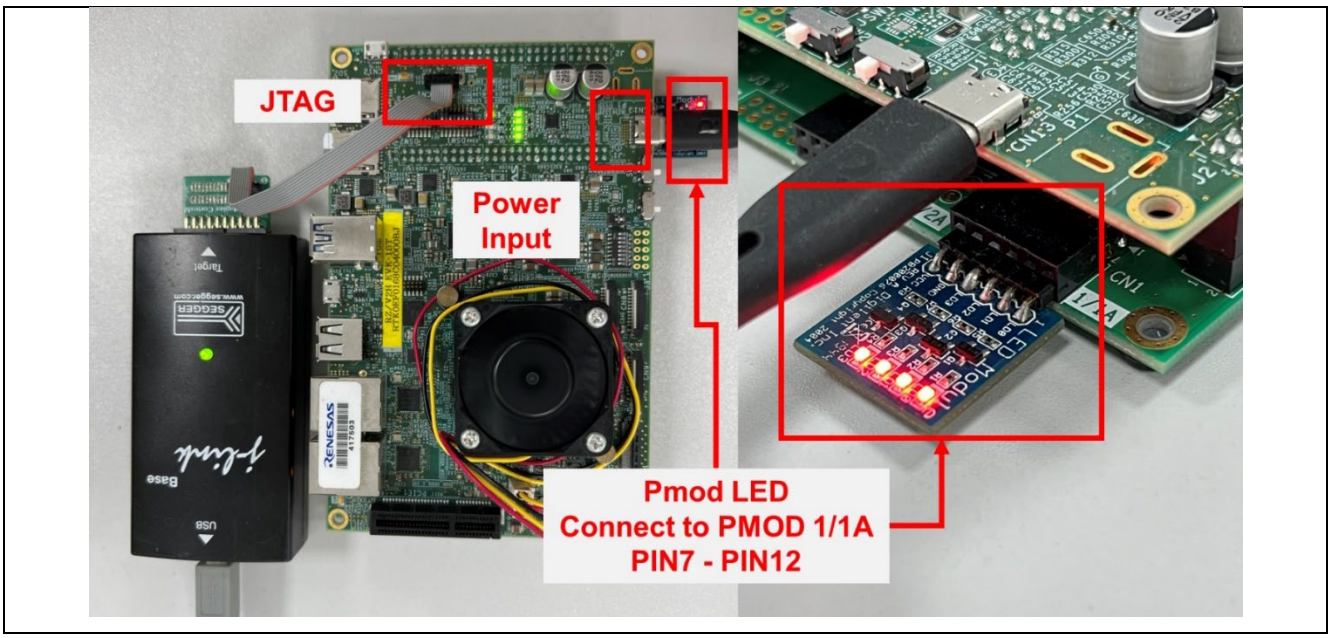
External hardware: Pmod LED

5.12.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)



Board Setting (RZ/V2H)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.12.4 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> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_gtm driver
00>
00> This Example Project demonstrates the functionality of GTM in
00> periodic mode and one-shot mode. On providing any input on the
00> RTTviewer, GTM channel 2 starts in one-shot mode. GTM channel 1
00> starts in periodic mode when GTM channel 2 expires. Timer in
00> periodic mode expires periodically at a time period specified
00> by user and toggles the Pmod0 pin 8 LED.
00>
00> Please enter time period values for one-shot and periodic mode
00> timers in milliseconds. 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:
  < 200
00>
00> Time period for periodic mode timer: 200
00>
```

If input wrong value, the message “Invalid input. Please enter valid input” will be printed out on the terminal.

```
00> One-shot mode:
  < 0
00>
00> Invalid input. Please enter valid input
  < 2001
00>
00> Invalid input. Please enter valid input
```

If input valid value, The message ” Time period for one-shot mode timer: ” will be printed out on the terminal.

```
00> Invalid input. Please enter valid input
  < 500
00>
00> Time period for periodic mode timer: 500
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
```

Then message "Periodic mode" will be printed, and the LED run with the time period for periodic mode timer set above.

```
00>
00> GTM1 is Enabled in Periodic mode
00> LED will toggle for set time period
00> Enter any key to stop timers
  <
00>
00> Periodic timer stopped. Enter any key to start timers.
```

5.13 I3C_B

5.13.1 Project Overview

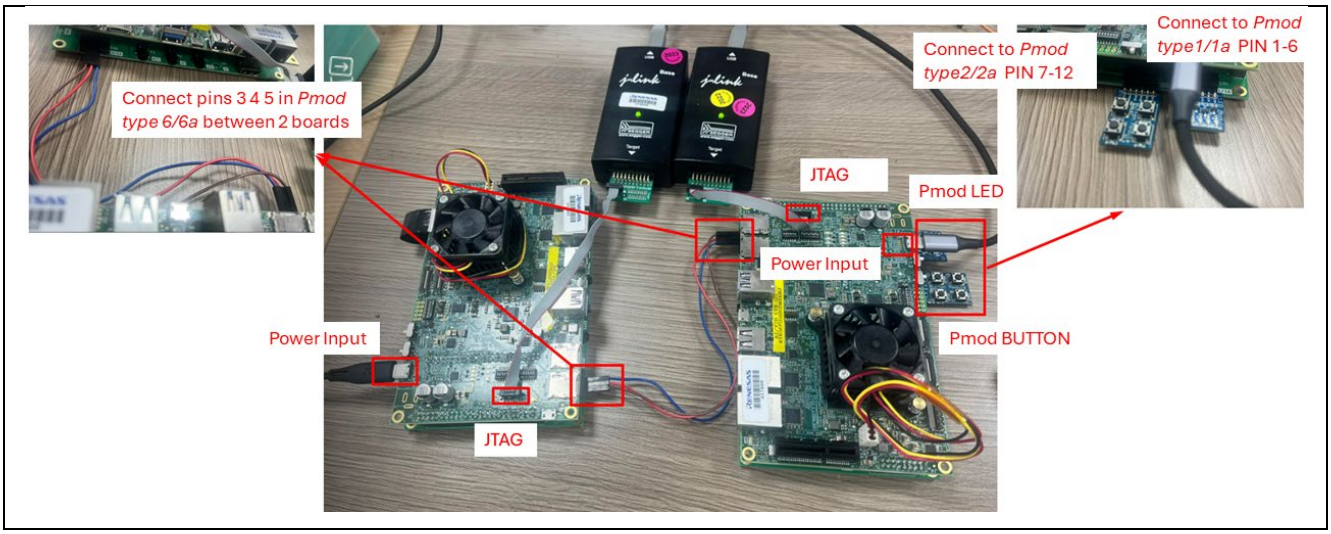
This example project demonstrates the typical use of the I3C Driver on Renesas RZ MCUs based on Renesas FSP. The I3C master on RZ/V2H demonstrates operations associated with an I3C slave running on another RZ/V2H board. Once initialization is successful, I3C Master device assigns slave address to I3C 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 switch BTN0 on the Pmod BUTTON is pressed on the slave side, it will initiate an IBI transfer request. Error and info messages will be printed on Jlink RTTViewer.

5.13.2 Hardware Requirements

External hardware: Pmod BTN and Pmod LED

5.13.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

Connect Pins on RZV2H_Expansion Board between Master board and Slave board as follows:

RZ/V2H EVK		
	Master board	Slave board
•	P21_SCL (Pin 3 Pmod 6/6A)	--- P21_SCL (Pin 3 Pmod 6/6A)
•	P20_SDA (Pin 4 Pmod 6/6A)	--- P20_SDA (Pin 4 Pmod 6/6A)
•	GND (Pin 5 Pmod 6/6A)	--- GND (Pin 5 Pmod 6/6A)

Please set each DIP switch as below.

Board	**RZ/V2H EVK**
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

5.13.4 Operation

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

Note: Please start the master first.

Master board:

```
This example project demonstrates the typical use of the I3C Driver as master
on Renesas RZ MCUs based on Renesas FSP. The I3C master demonstrates operation
with I3C slave connected on another RZ board. once initialization is
successful, it will start assigning dynamic addresses to slave devices present on the bus.
The EP performs write/read operation and displays device information based on user input.
Error and info messages will be printed on JlinkRTTViewer.
Refer to readme.txt file for more details on Example Project and
FSP User's Manual for more information about r_i3c_b driver

INFO : I3C Initialized successfully in master mode.

INFO : Sending CCC broadcast signal for Dynamic address assignment.
INFO : CCC Dynamic Address Assignment transfer completed successfully.

I3C Master operations :
1. Display I3C slave device Information if exists on I3C bus
2. I3C Write Read operation

1
ERROR : No Slave device exists on I3C bus,
Sending broadcast common command code to check for slave Hot Join requests
Please re-check again
INFO : CCC Dynamic Address Assignment transfer completed successfully.

INFO : Please re-check with menu option 1 to view dynamic address assigned to slave before using menu option 2
I3C Master operations :
1. Display I3C slave device Information if exists on I3C bus
2. I3C Write Read operation

INFO : A hot Join event is received, Initiate DAA using CCC transmission.
INFO : DAA using CCC transmission completed,
Please check by pressing user input 1 (available at menu option) for slave information
1
INFO : number of I3C device on bus: 1 **

*********************************************
*                I3C Slave Info             *
*********************************************
- Dynamic Address:   0x71
- BCR:               0x06
- DCR:               0x00
- PID:               0x04CC00050000

I3C Master operations :
1. Display I3C slave device Information if exists on I3C bus
2. I3C Write Read operation

2
INFO : Data written to I3C slave is read back and matching - SUCCESS
INFO: Data Transfer size 0x20

I3C Master operations :
1. Display I3C slave device Information if exists on I3C bus
2. I3C Write Read operation

2
INFO : Data written to I3C slave is read back and matching - SUCCESS
INFO: Data Transfer size 0x20

I3C Master operations :
1. Display I3C slave device Information if exists on I3C bus
2. I3C Write Read operation

2
INFO : Data written to I3C slave is read back and matching - SUCCESS
INFO: Data Transfer size 0x20
```

Slave board:

Below are the observations of the LED operation on slave board :

- When Dynamic Address Assignment completes, LED LD0 (PMOD LED) will be toggled.
- When write transfer completes, LED LD1 (PMOD LED) will be toggled.
- When read transfer completes, LED LD2 (PMOD LED) will be toggled.
- When IBI write transfer completes, LED LD1 (PMOD LED) will be toggled.

5.14 FreeRTOS

5.14.1 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.

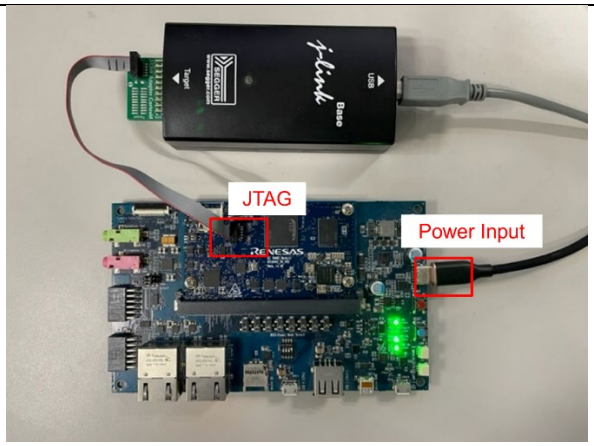
All the tasks run with equal priority level.

5.14.2 Hardware Requirements

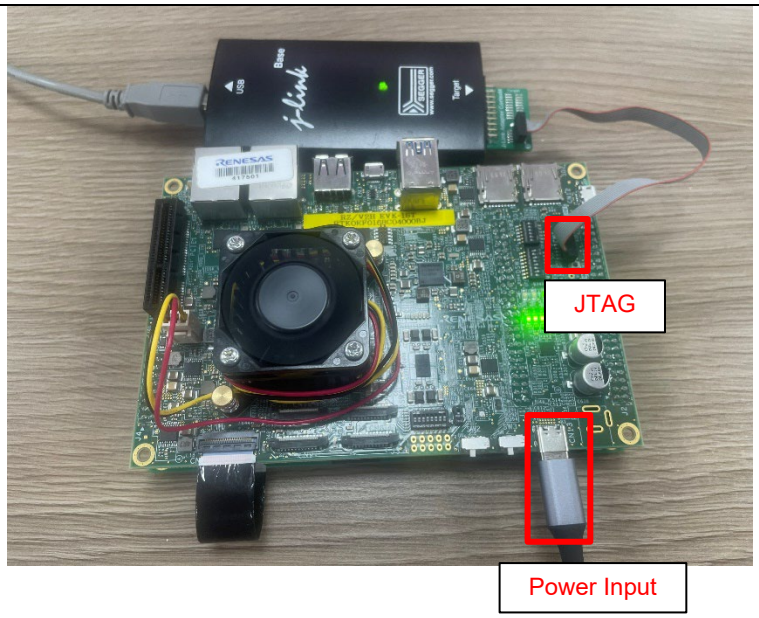
External hardware: None

5.14.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)



Board Setting (RZ/V2H)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.14.4 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 FreeRTOS Message Queue & Semaphore driver
00>
00> Messages are shared between Sender_Task, ISR and Receiver_Task for the first few seconds.
00> Later, Sender and Receiver Tasks will be suspended timer will be stopped. Semaphore is acquired
00> and released between Semaphore Task and ISR for the next few seconds and Semaphore Task is suspended.
00> To restart the application, power cycle the board.
00>
00>  Sender_Task : Starting g_periodic_timer_msgq timer
00> GTM Timer Started successfully
00>
00>  Sender_Task : After delay of 500ms
00>
00>  Sender_Task : Message posted on Queue successfully
00>  Sender_Task : Going on delay for 500ms
00> Receiver Task : Message received Successfully
00>     Data   : 100
00>     Sender : Sender_Task
00>
00>  Sender_Task : After delay of 500ms
00>
00>  Sender_Task : Message posted on Queue successfully
00>  Sender_Task : Going on delay for 500ms
00> Receiver Task : Message received Successfully
00>     Data   : 100
00>     Sender : Sender_Task
00>
00> Receiver Task : Message received Successfully
00>     Data   : 200
00>     Sender : GTM Callback
00>
00>  Sender_Task : After delay of 500ms
00>
00>  Sender_Task : Message posted on Queue successfully
00>  Sender_Task : Going on delay for 500ms
00> Receiver Task : Message received Successfully
00>     Data   : 100
00>     Sender : Sender_Task
00>
00>  Sender_Task : After delay of 500ms
00>
00>  Sender_Task : Message posted on Queue successfully
00>  Sender_Task : Going on delay for 500ms
00> Receiver Task : Message received Successfully
00>     Data   : 100
00>     Sender : Sender_Task
```

5.15 INTC_IRQ

5.15.1 Project Overview

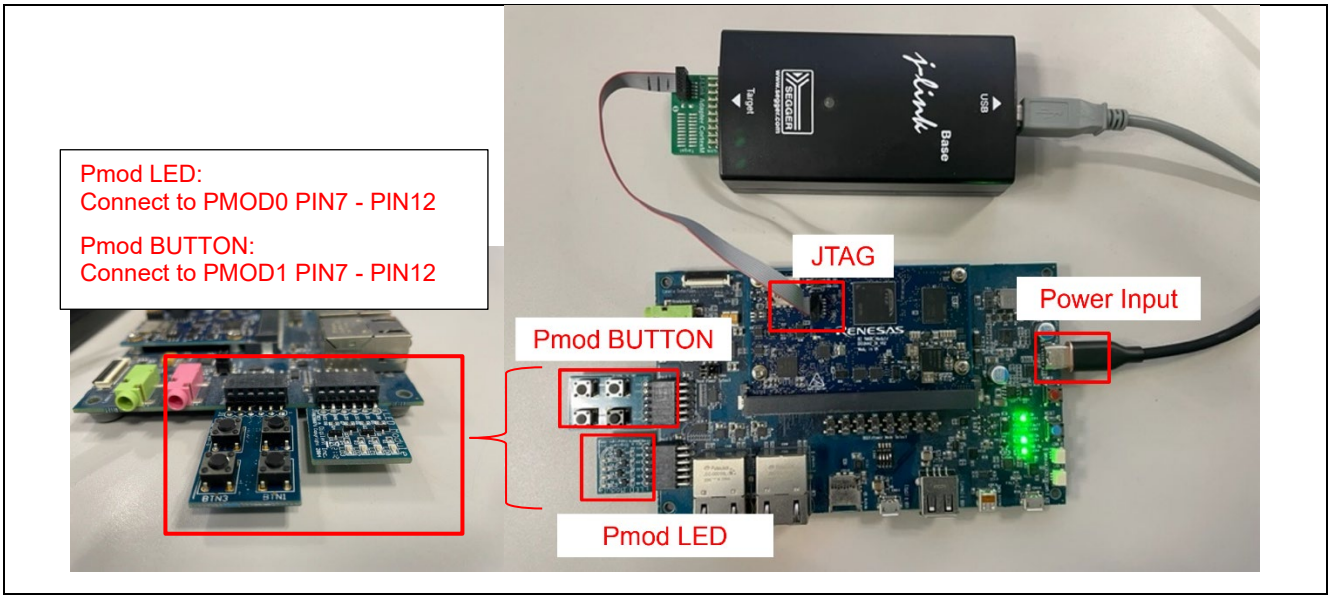
The example project demonstrates the typical use of the INTC IRQ module APIs. The project initializes the IRQ interrupt in Interrupt Controller. User is requested to press the push button to trigger the external IRQ and this then will start toggling of user LED.

5.15.2 Hardware Requirements

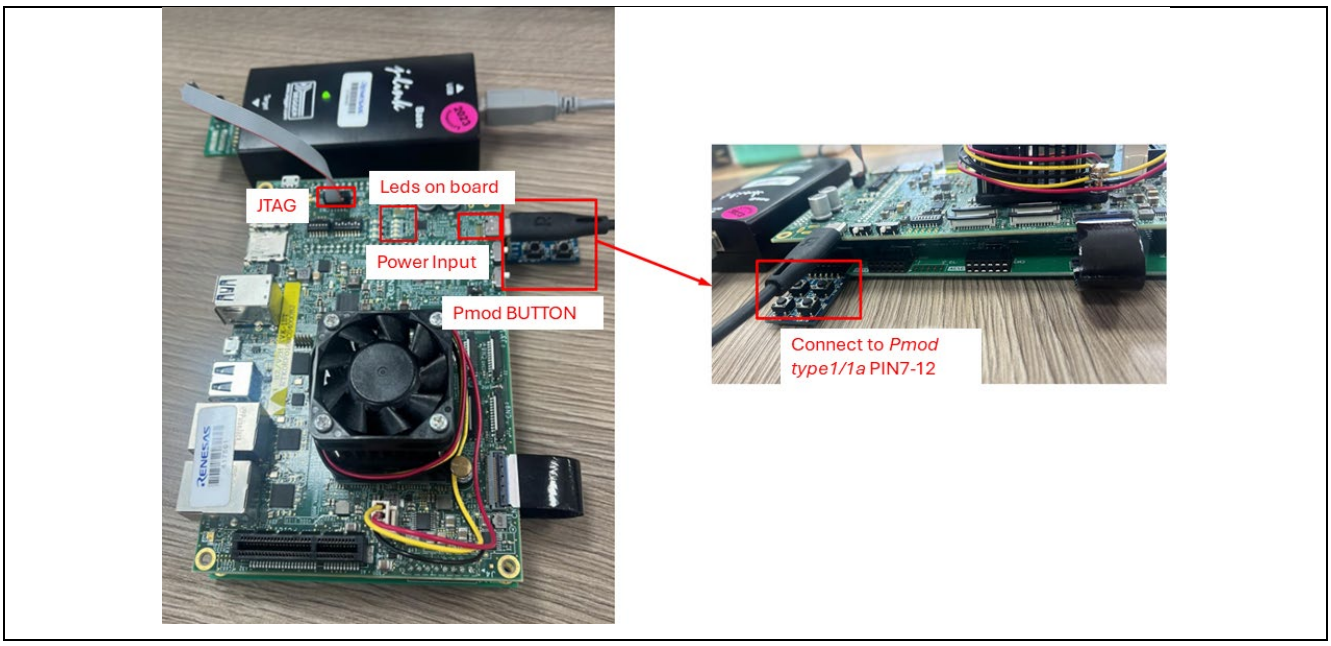
External hardware: Pmod BTN and Pmod LED

5.15.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)



Board Setting (RZ/V2H)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.15.4 Operation

After running the example, the message below is displayed on console. LED ON/OFF status is displayed when pressing the BTN0 of Pmod BUTTON.

```
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}
```

5.16 INTC_TINT

5.16.1 Project Overview

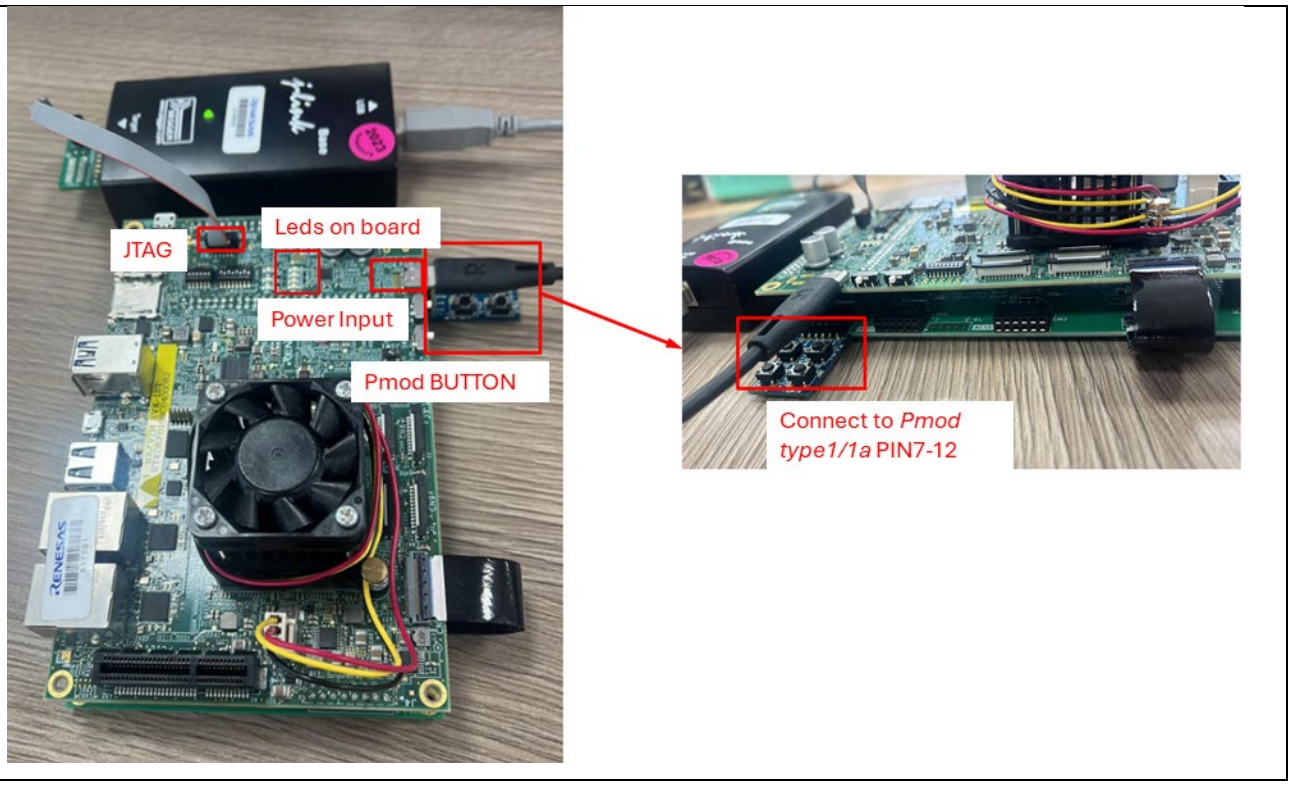
The example project demonstrates the typical use of the INTC TINT module APIs. The project initializes the TINT interrupt in Interrupt Controller User is requested to press the push button to trigger the external TINT and this then will start toggling of user LED.

5.16.2 Hardware Requirements

External Hardware : Pmod BTN

5.16.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

Please set each DIP switch and jumpers as below.

Board	RZ/V2H EVK
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

5.16.4 Operation

After running the example, the message below is displayed on console. LED ON/OFF status is displayed when pressing the BTN0 of Pmod BUTTON.

```
00> Refer to readme.txt file for more details on Example Project and
00> RZA FSP Documentation for more information about r_intc_tint driver
00>
00> This Example Project demonstrates the functionality of INTC_TINT 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: Low{OFF}
00>
00> User Pushbutton Pressed
00> LED State: High{ON}
00>
```

5.17 MTU3

5.17.1 Project Overview

The example project demonstrates typical use of MTU3 HAL module APIs.

User has the provision to input value as per displayed menu through a terminal application to select different MTU3 supported modes(Periodic, PWM, One-Shot).

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

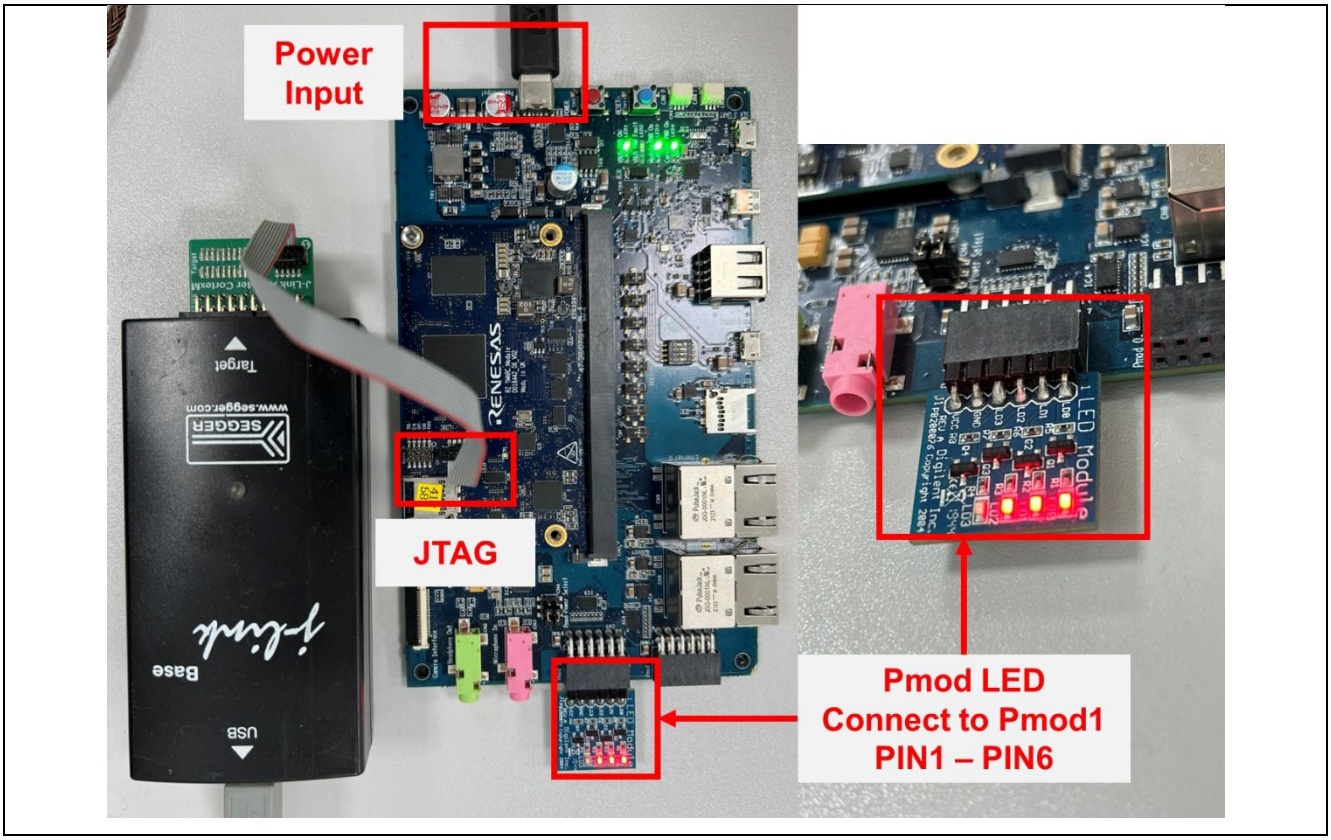
Any failure will also be displayed on a terminal application.

5.17.2 Hardware Requirements

External Hardware : Pmod LED

5.17.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)

Please set each DIP switch and jumpers as below.

Board	RZ/V2L EVK
Module board	SW1-1: OFF SW1-2: Don’t care.
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. 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

5.17.4 Operation

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

```
00>
00> ****************************************************************
00> *   Renesas FSP Example Project for r_mtu3 Module               *
00> *   Example Project Version 2.0                                *
00> *   Flex Software Pack Version  2.0.1                          *
00> ****************************************************************
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_mtu3 driver
00> The project initializes MTU3 module in Periodic, PWM or One-shot
00> mode based on user input from the displayed menu options.
00>
00> In periodic mode, user can enter the time period within the
00> permitted ranges to change the frequency of the user LED.
00>
00> In PWM mode, user can enter the duty cycle within the specified
00> range to adjust the intensity of the user LED.
00>
00> In ONE SHOT mode, Output will be displayed on terminal application
00> 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
```

Enter 1 for Periodic Mode and enter "500" to set a period of the timer. LED blinks at a cycle of 500ms and output the message as follows.

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

Enter 2 for PWM Mode and enter "20" to set a Duty Cycle. LED blinks at a duty ratio of 20% and output the message as follows.

```
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:
00>
  < 2
00> Periodic Timer Instance Closed, Start PWM Timer Instance
00> Enter The Desired Duty Cycle in Percentage of Range 0 - 100:
00>
  < 20
00> Opened Timer in PWM Mode
00> Started Timer in PWM Mode
```

Enter 3 for One Shot Mode. The LED illuminates only once. After that, display that an one-shot pulse was output on Tera Term.

```
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:
00>
  < 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> Stopped and reseted Timer in ONE-SHOT Mode
00> Timer Expired in One-Shot Mode
```

5.18 POEG

5.18.1 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.

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

RZ/V2H:

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

RZ/V2L:

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

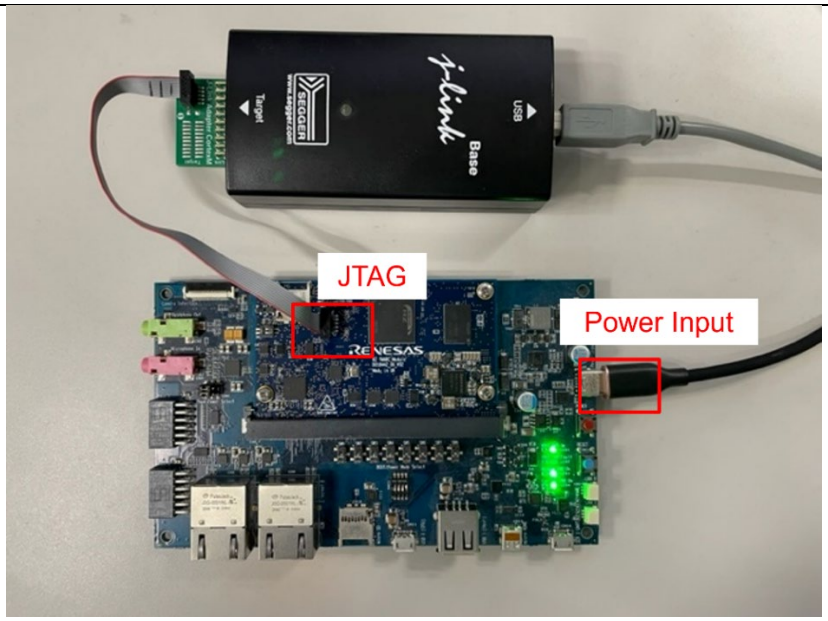
Based on the user's selection, the corresponding trigger mechanism is activated to stop the GPT signal output or reset POEG.

5.18.2 Hardware Requirements

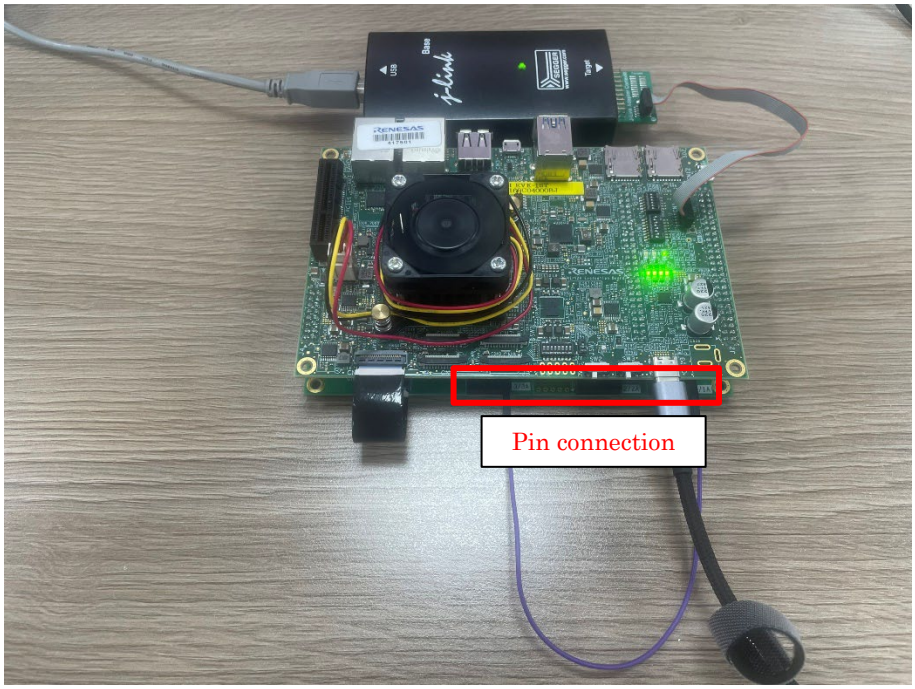
External hardware: Pmod LED (for RZ/V2L)

5.18.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2L)



Board Setting (RZ/V2H)

Please set the Pin Connection as below.

RZ/V2H EVK
Connect P57 (PMOD3_PIN7) to P80 (PMOD1_PIN1) on RZ/V2H EVK Expansion Board.

Please set each DIP switch and jumpers as below.

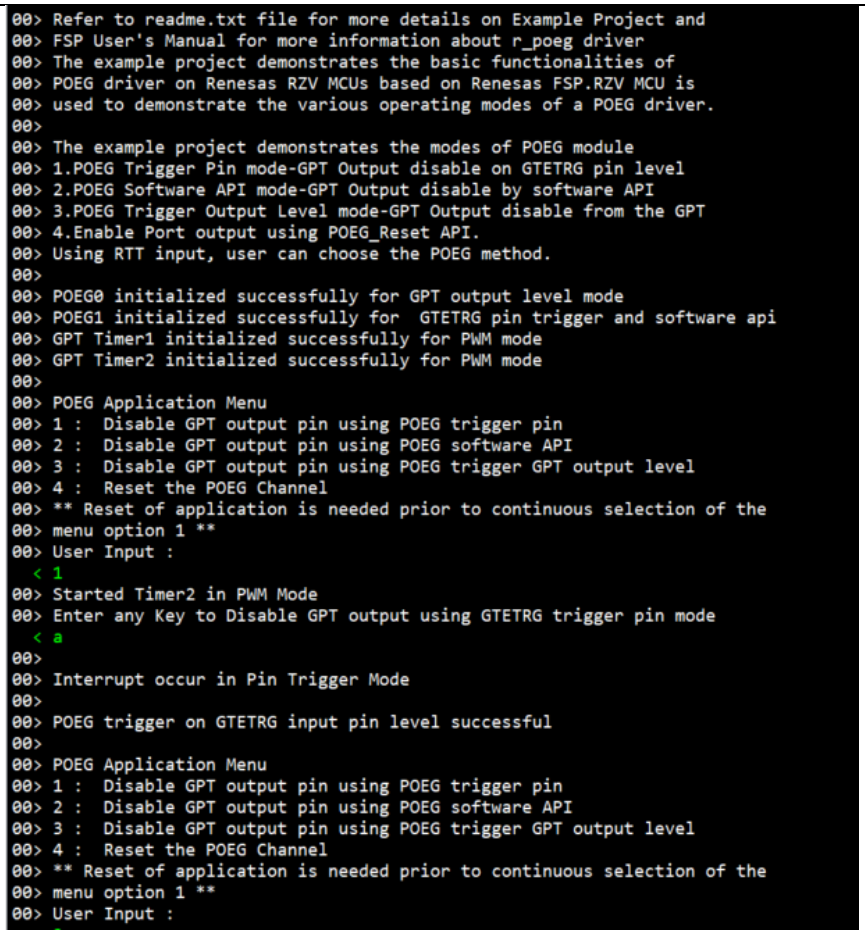
Board	RZ/V2L EVK	RZ/V2H EVK
Module board (For RZ/V2L) RZ/V2H Secure Evaluation Board (For RZ/V2H)	SW1-1: OFF SW1-2: Don’t care.	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
Carrier board(For RZ/V2L) RZ/V2H EVK Expansion Board. (For RZ/V2H)	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	None

5.18.4 Operation

After running the example, the following message is displayed on the console. The user is prompted to enter a number to select the desired trigger mechanism. After selecting the trigger, the corresponding action is performed accordingly:

```
00> Refer to readme.txt file for more details on Example Project and
00> FSP User's Manual for more information about r_poeg driver
00> The example project demonstrates the basic functionalities of
00> POEG driver on Renesas RZV MCUs based on Renesas FSP.RZV MCU is
00> used to demonstrate the various operating modes of a POEG driver.
00>
00> The example project demonstrates the modes of POEG module
00> 1.POEG Software API mode-GPT Output disable by software API
00> 2.POEG Trigger Output Level mode-GPT Output disable from the GPT
00> 3.Enable Port output using POEG_Reset API.
00> Using RTT input, user can choose the POEG method.
00>
00> POEG0 initialized successfully for GPT output level mode
00> POEG1 initialized successfully for software api
00> GPT Timer1 initialized successfully for PWM mode
00> GPT Timer2 initialized successfully for PWM mode
00>
00> POEG Application Menu
00> 1 :  Disable GPT output pin using POEG software API
00> 2 :  Disable GPT output pin using POEG trigger GPT output level
00> 3 :  Reset the POEG Channel
00> ** Reset of application is needed prior to continuous selection of the
00> menu option 1 **
00> User Input :
  < 1
00>
00> Started Timer2 in PWM Mode
00> Enter any Key to Disable GPT output pins using POEG software API
  < a
00>
00> POEG output disable using software API successful
00>
00> POEG Application Menu
00> 1 :  Disable GPT output pin using POEG software API
00> 2 :  Disable GPT output pin using POEG trigger GPT output level
00> 3 :  Reset the POEG Channel
00> ** Reset of application is needed prior to continuous selection of the
00> menu option 1 **
00> User Input :
```

Operation for RZ/V2L



Operation for RZ/V2H

5.19 WDT

5.19.1 Project Overview

User can give input through RTT Viewer to start the WDT. WDT gets refreshed periodically through GTM timer.

User can press BTN0 button of PMOD BUTTON to stop the GTM timer which in turn stops refreshing WDT timer.

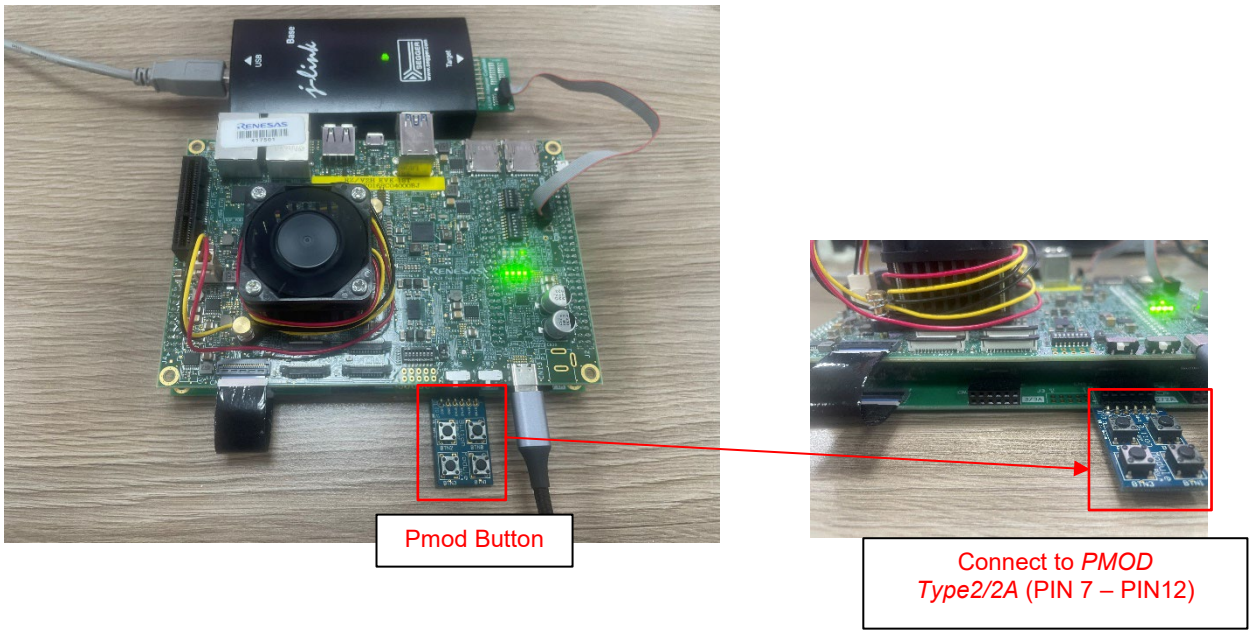
Approximately after 2 seconds, WDT resets the MPU and turn on LED on board until user gives the input once again.

5.19.2 Hardware Requirements

External hardware: Pmod BTN

5.19.3 Hardware Settings

Please connect each hardware as below.



Board Setting (RZ/V2H)

Please set each DIP switch as below.

Board	**RZ/V2H EVK**
RZ/V2H Secure Evaluation Board	DSW1-1: ON DSW1-2: OFF DSW1-3: ON DSW1-4: OFF DSW1-5: ON DSW1-6: OFF DSW1-7: ON DSW1-8: OFF DSW2-1:OFF DSW2-2:OFF DSW2-3:OFF DSW2-4:OFF DSW2-5:OFF DSW3-1:ON DSW3-2:ON DSW3-3:ON DSW3-4:ON DSW3-5:ON DSW3-6:ON DSW3-7:ON DSW3-8:ON
RZ/V2H EVK Expansion Board.	None

5.19.4 Operation

After running the example, the message below is displayed on console. After inputting “1”, the operation will be performed accordingly. Pressing the button to stop WDT counter from refresing.

```
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, WDT counter stops from refreshing
00> WDT resets the MCU in 2 seconds
00>
00> Enter 1 to Enable WDT
00> User Input:
  < 1
00>
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.
00>
00> ******************************************************************
00> *   Renesas FSP Example Project for r_wdt Module                  *
00> *   Example Project Version 2.0                                   *
00> *   Flex Software Pack Version  2.0.1                             *
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, WDT counter stops from refreshing
00> WDT resets the MCU in 2 seconds
00>
00> ********************** WDT Reset detected ***********************
00>
00> Enter 1 to Enable WDT
00> User Input:
  < 1
00>
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.
```

References

FSP GitHub:	github.com/renesas/rzv-fsp
FSP User Manual:	renesas.github.io/rzv-fsp/
Getting Started Guide:	Getting Started with RZ/V Flexible Software Package (renesas.com)
FSP Example Projects:	RZ/V Multi-OS Package | Renesas
Evaluation Kit Manual (RZ/V2L):	RZ/G2L, RZ/V2L SMARC Module Board User's Manual: Hardware (renesas.com) RZ SMARC Series Carrier Board User's Manual: Hardware
Evaluation Kit Manual (RZ/V2H):	RZ/V2H Evaluation Board Kit (renesas.com)
Knowledge Base:	Knowledge Base (renesas.com)
Renesas Support:	RZ/V2L - Support (renesas.com) RZ/V2H - Support (renesas.com)

Revision History

Rev.	Date	Description	
		Page	Summary
2.00	Aug.30.24	1	Added support for RZ/V2H.
		5 to 10	Updated the description and figure based on the latest development environment.
		14 to 72	Added description for RZ/V2H EVK environment and new supported examples.
1.00	Mar.29.24	–	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/.