RZ/A 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/A FSP.

Target Device

- RZ/A3UL

Supported Board Edition

- RZ/A3UL Evaluation Board Kit QSPI Edition.
- RZ/A3UL Evaluation Board Kit OCTAL-SPI Edition.

Contents

1. Prerequisites....6
2. Hardware and Software Requirements....6
3. Tool Installation....6
3.1 FSP and Tools Installation....6
4. Downloading and Running the Project....6
4.1 Downloading the Project....6
4.2 Running the Project....7
4.2.1 Importing the Project into e2studio....7
5. Change Board Edition Settings on the Project....10
5.1 Change to RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)....10
5.2 Change to RZ/A3UL Evaluation Board Kit OCTAL Edition (eXecute-In-Place)....12
6. About Examples....16
6.1 ADC....16
6.1.1 Project Overview....16
6.1.2 Hardware Requirements....16
6.1.3 Hardware Settings....16
6.1.4 Operation....17
6.2 CANFD....18
6.2.1 Project Overview....18
6.2.2 Hardware Requirements....18
6.2.3 Hardware Settings....18
6.2.4 Operation....19
6.3 RIIC Master....20
6.3.1 Project Overview....20
6.3.2 Hardware Requirements....20
6.3.3 Hardware Settings....20
6.3.4 Operation....21
6.4 RIIC Slave....22
6.4.1 Project Overview....22
6.4.2 Hardware Requirements....22
6.4.3 Hardware Settings....22
6.4.4 Operation....23
6.5 SSI....24
6.5.1 Project Overview....24
6.5.2 Hardware Requirements....24
6.5.3 Hardware Settings....24
6.5.4 Operation....25

6.6 RSPI 26
6.6.1 Project Overview 26
6.6.2 Hardware Requirements 26
6.6.3 Hardware Settings 26
6.6.4 Operation 27
6.7 SCIg 29
6.7.1 Project Overview 29
6.7.2 Hardware Requirements 29
6.7.3 Hardware Settings 29
6.7.4 Operation 30
6.8 SCIF 31
6.8.1 Project Overview 31
6.8.2 Hardware Requirements 31
6.8.3 Hardware Settings 31
6.8.4 Operation 32
6.9 USB HHID 33
6.9.1 Project Overview 33
6.9.2 Hardware Requirements 33
6.9.3 Hardware Settings 33
6.9.4 Operation 34
6.10 USB HMSC 35
6.10.1 Project Overview 35
6.10.2 Hardware Requirements 35
6.10.3 Hardware Settings 35
6.10.4 Operation 36
6.11 USB PCDC 37
6.11.1 Project Overview 37
6.11.2 Hardware Requirements 37
6.11.3 Hardware Settings 37
6.11.4 Operation 38
6.12 LCDC 39
6.12.1 Project Overview 39
6.12.2 Hardware Requirements 39
6.12.3 Hardware Settings 39
6.12.4 Operation 40
6.13 INTC IRQ 41
6.13.1 Project Overview 41
6.13.2 Hardware Requirements 41
6.13.3 Hardware Settings 41
6.13.4 Operation 42
6.14 INTC TINT 43

6.14.1 Project Overview 43
6.14.2 Hardware Requirements 43
6.14.3 Hardware Settings 43
6.14.4 Operation 44
6.15 WDT 45
6.15.1 Project Overview 45
6.15.2 Hardware Requirements 45
6.15.3 Hardware Settings 45
6.15.4 Operation 46
6.16 Ether 48
6.16.1 Project Overview 48
6.16.2 Hardware Requirements 48
6.16.3 Hardware Settings 49
6.16.4 Operation 49
6.17 FreeRTOS 50
6.17.1 Project Overview 50
6.17.2 Hardware Requirements 50
6.17.3 Hardware Settings 50
6.17.4 Operation 51
6.18 SDHI 52
6.18.1 Project Overview 52
6.18.2 Hardware Requirements 52
6.18.3 Hardware Settings 52
6.18.4 Operation 53
6.19 GTM 54
6.19.1 Project Overview 54
6.19.2 Hardware Requirements 54
6.19.3 Hardware Settings 54
6.19.4 Operation 55
6.20 MTU3a 56
6.20.1 Project Overview 56
6.20.2 Hardware Requirements 56
6.20.3 Hardware Settings 56
6.20.4 Operation 57
6.21 DMAC 58
6.21.1 Project Overview 58
6.21.2 Hardware Requirements 58
6.21.3 Hardware Settings 58
6.21.4 Operation 59

7. References 60

Revision History.....61

1. Prerequisites

1. Tool experience: It is assumed that the user has prior experience working with integrated development environments, such as e^2 studio and terminal emulation programs, such as Tera Term.
2. Subject knowledge: It is assumed that the user has basic knowledge about microprocessors, embedded systems, and FSP to modify the example projects. First time users are recommended to refer to *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 e^2 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/A FSP Example projects are designed to operate using Evaluation Board Kit for RZ/A3UL 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.

Operating Environment

- Windows® 10 operating system
- RZ/A FSP v3.0.0
- e^2 studio 2024-04

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/rza-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 e^2 studio.

4. Downloading and Running the Project

4.1 Downloading the Project

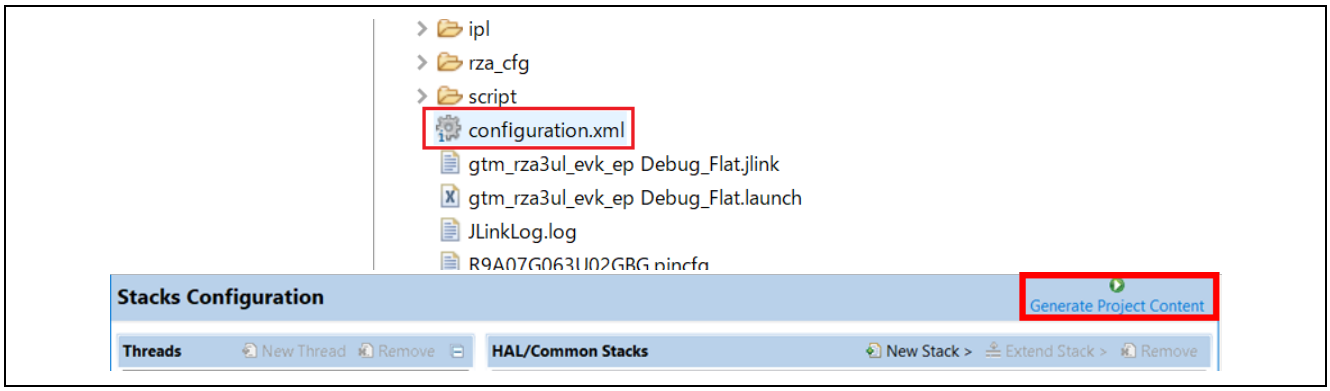
1. Download the example project which is "RZ/A3UL EVK Example Projects" from RZ/A3UL Software Package | Renesas.

4.2 Running the Project

4.2.1 Importing the Project into e2studio

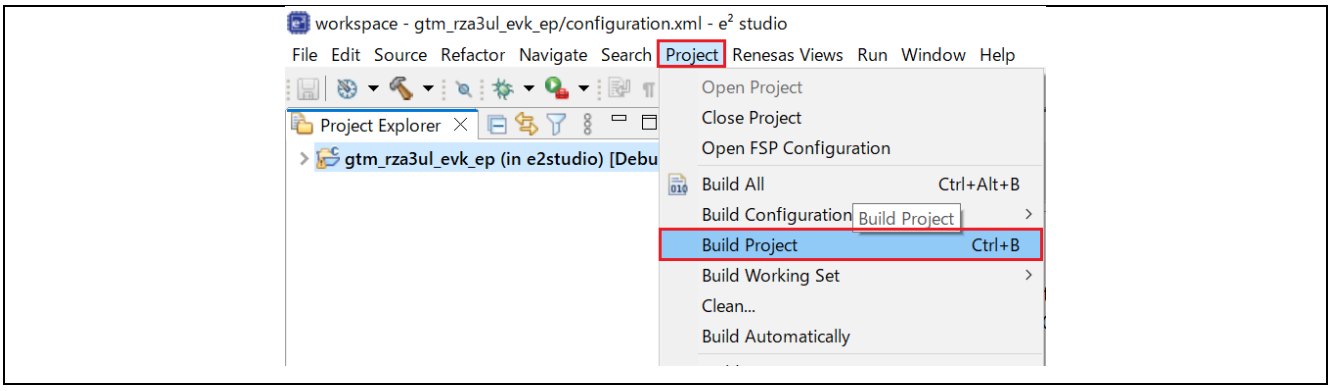
1. Import an existing project.
   Refer to the section *Importing an Existing Project into e2 studio* in *Getting Started with Flexible Software Package.*

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

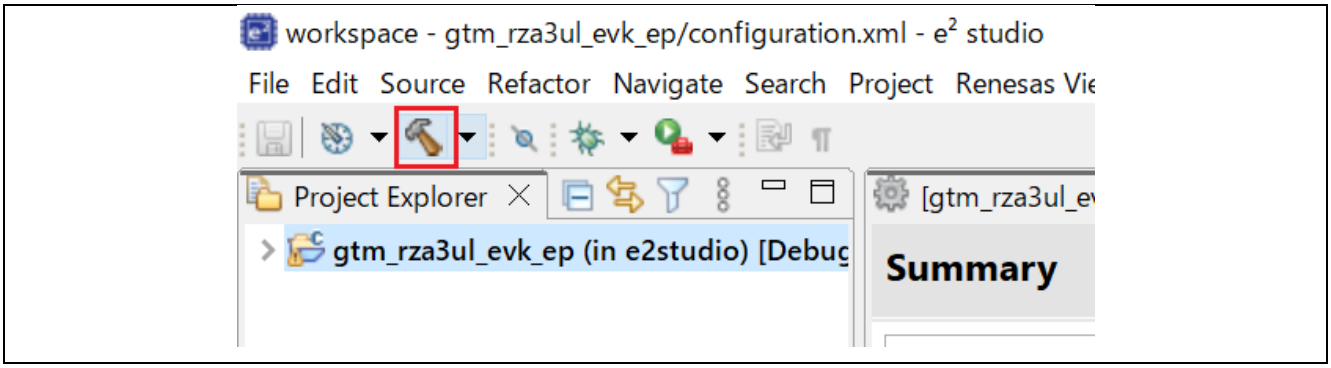


3. Build the project.
   There are three ways to build a project:

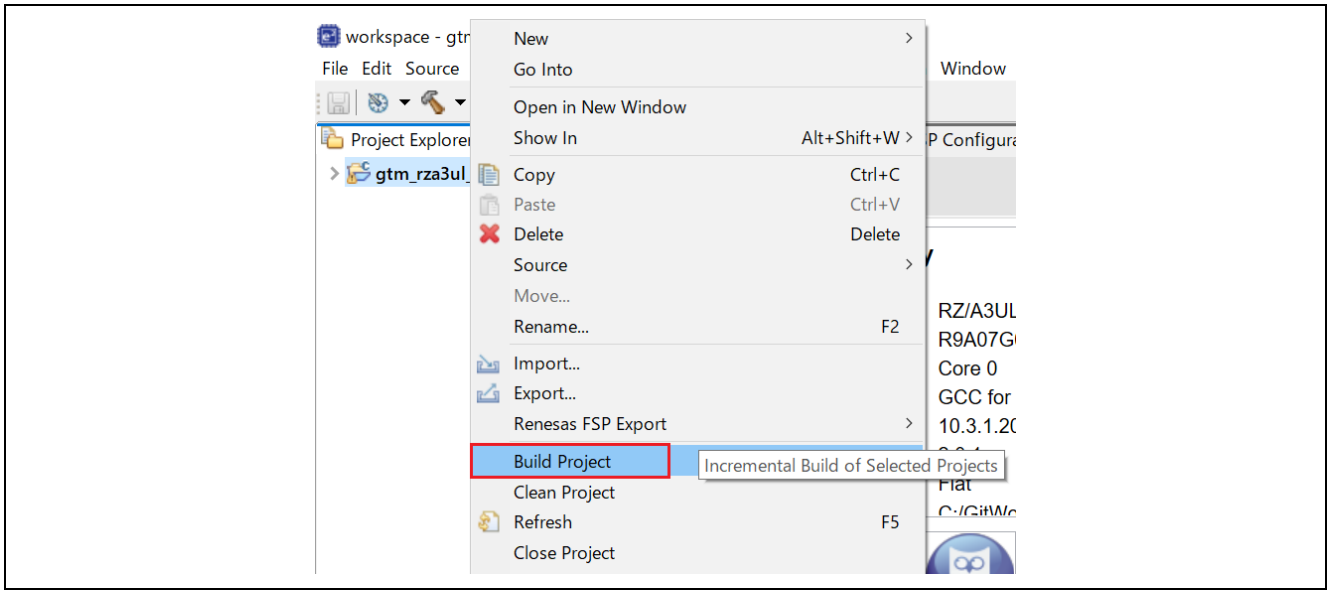
   a. Click on **Project** in the menu bar and select **Build Project**.



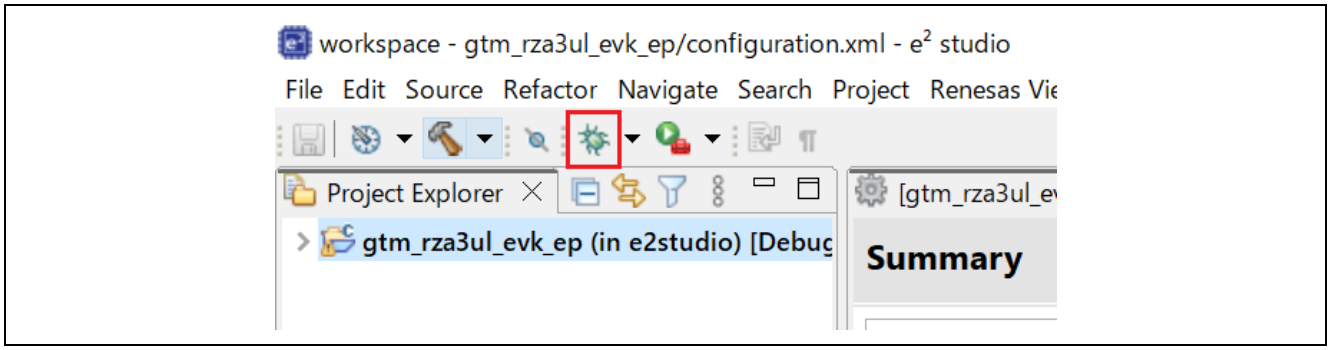
   b. Click on the hammer icon.



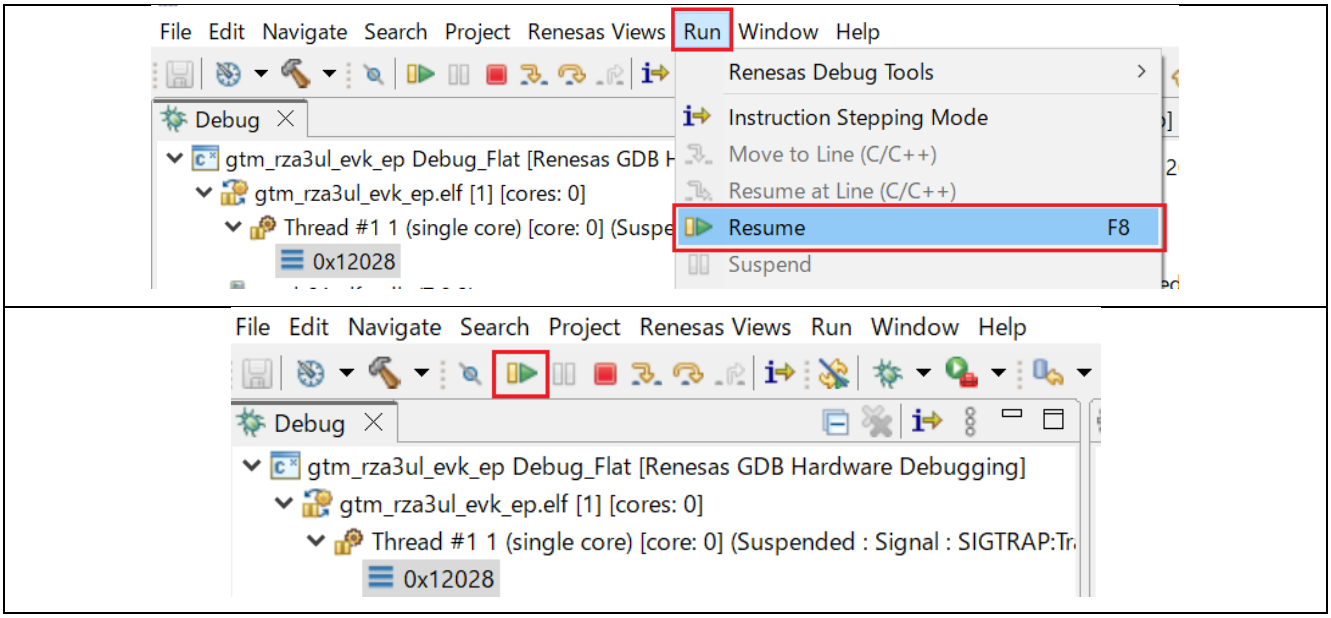
c. Right-click on the project and select **Build Project**.



4. Refer to the `readme.txt` file in the project folder and configure the hardware settings. Turn on the board and connect it to the PC. Set the configuration of Terminal Emulator.
   - Speed: 115200bps
   - Data: 8bit
   - Parity: None
   - Stop bits: 1bit
   - Flow control: None
5. Downloading the project image to the board.
   Click **Debug** to begin debugging the application.



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



7. Follow the instructions displayed on the Terminal Emulator as shown below.

Note:

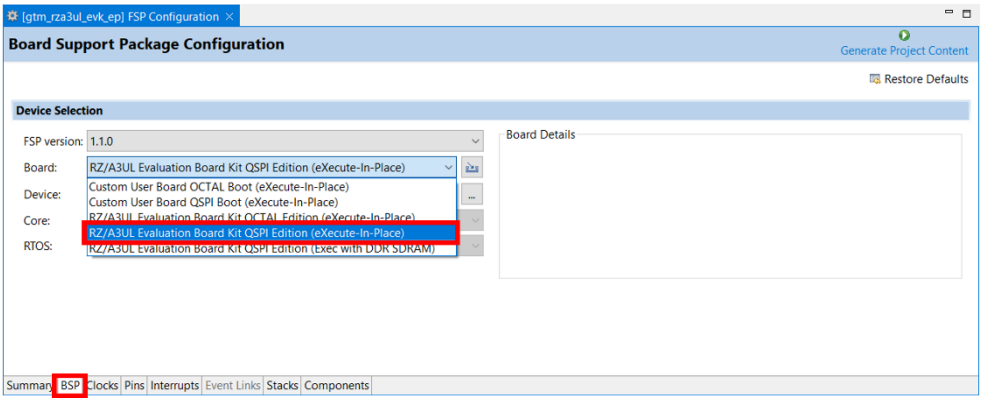
1. Example Projects do not support floating point or special characters or any non-numeric characters.
2. Example projects do not handle cases where the user input is greater than the expected input array size.

5. Change Board Edition Settings on the Project

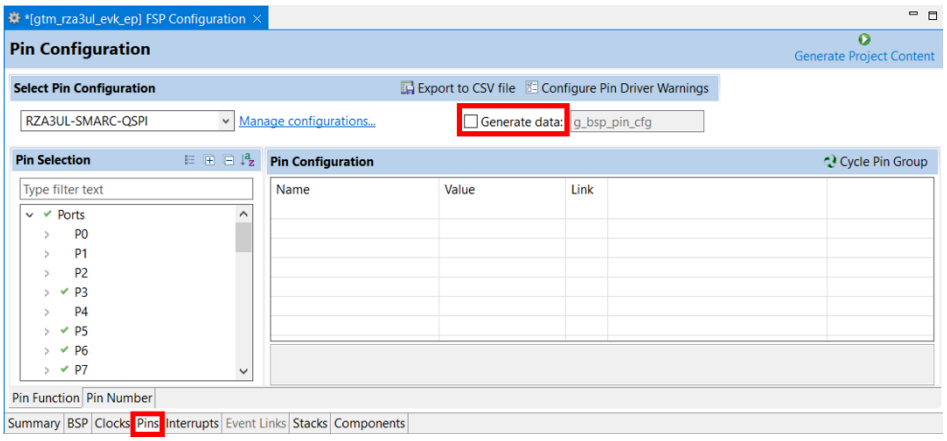
This example project sets RZ/A3UL Evaluation Board Kit QSPI Edition (Exec with DDR-SDRAM) as default. If you can use the project for other board edition, follow the below procedure.

5.1 Change to RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)

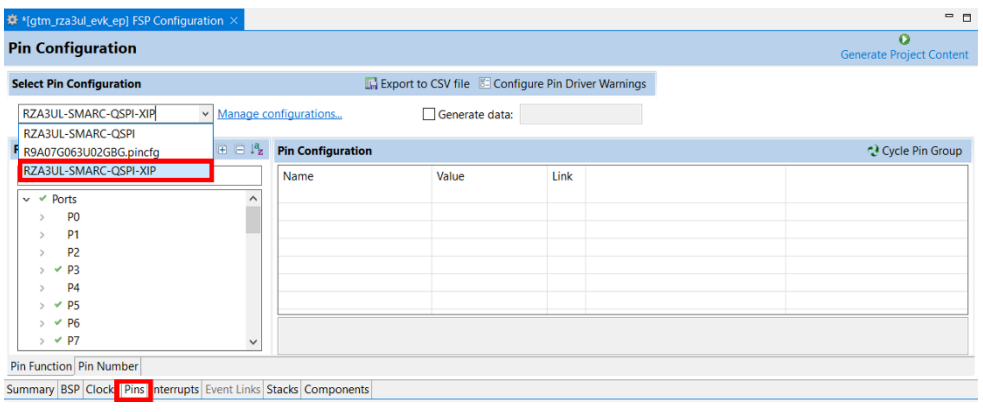
1. Change **Device Selection** to **RZ/A3UL Evaluation Board Kit QSPI Edition (eXecute-In-Place)** in FSP Configuration > BSP tab.



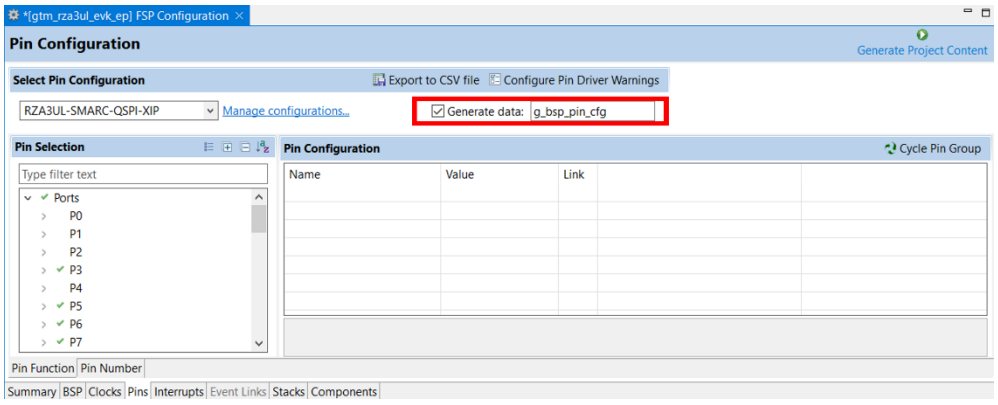
2. Uncheck the box in **Generate data** of FSP Configuration > Pins tab.



3. Change **Select Pin Configuration** to **RZA3UL-SMARC-QSPI-XIP** in FSP Configuration > Pins tab.



4. Check the box and enter **g_bsp_pin_cfg** in **Generate data** of FSP Configuration > Pins tab.

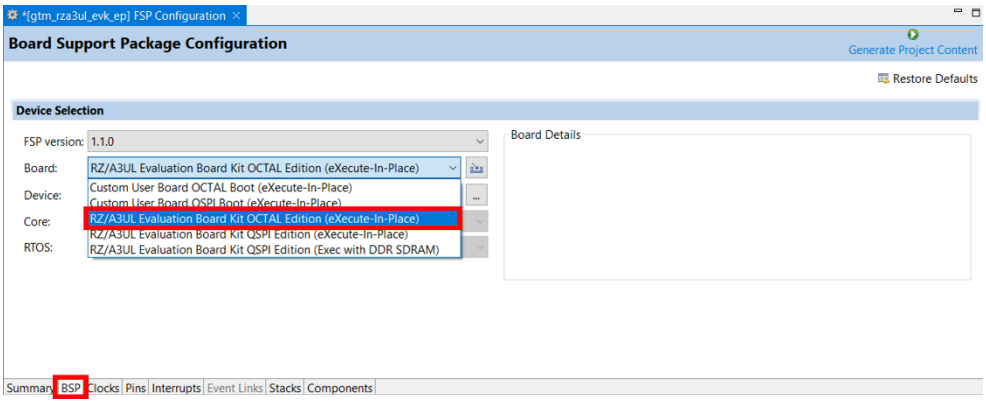


5. Click **Generate Project Content**.

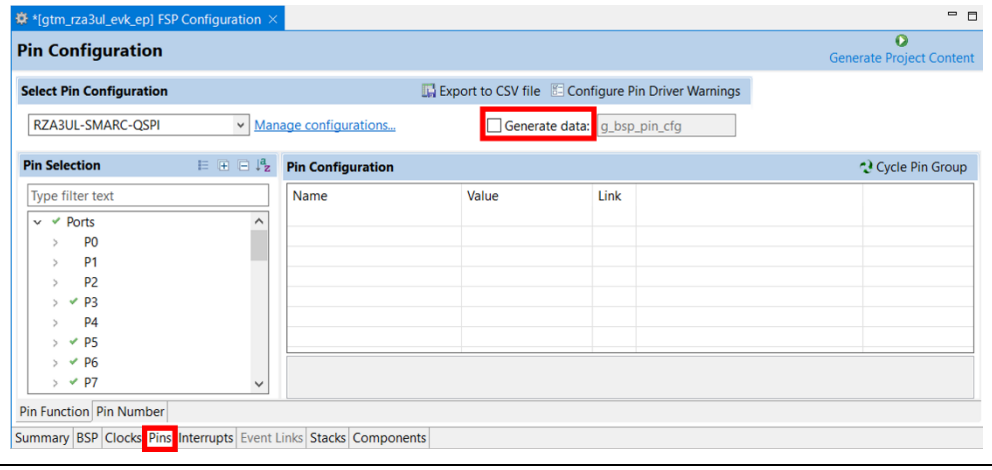
6. Build the project.

5.2 Change to RZ/A3UL Evaluation Board Kit OCTAL Edition (eXecute-In-Place)

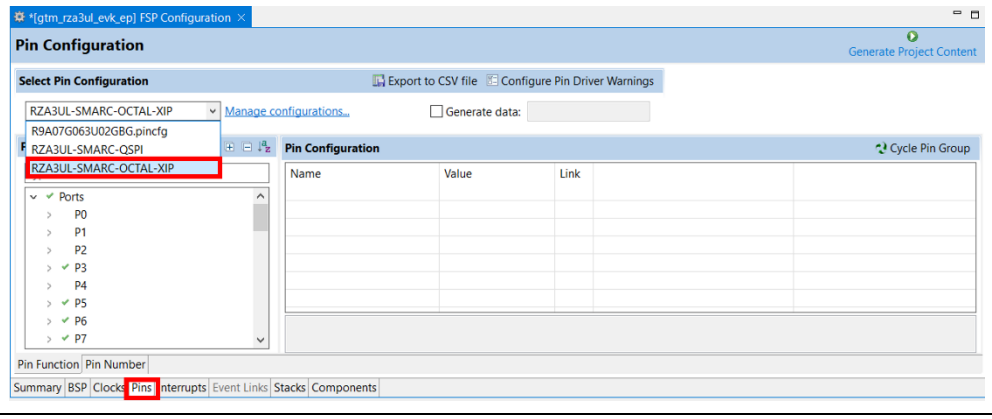
1. Change **Device Selection** to **RZ/A3UL Evaluation Board Kit OCTAL Edition (eXecute-In-Place)** in FSP Configuration > BSP tab.



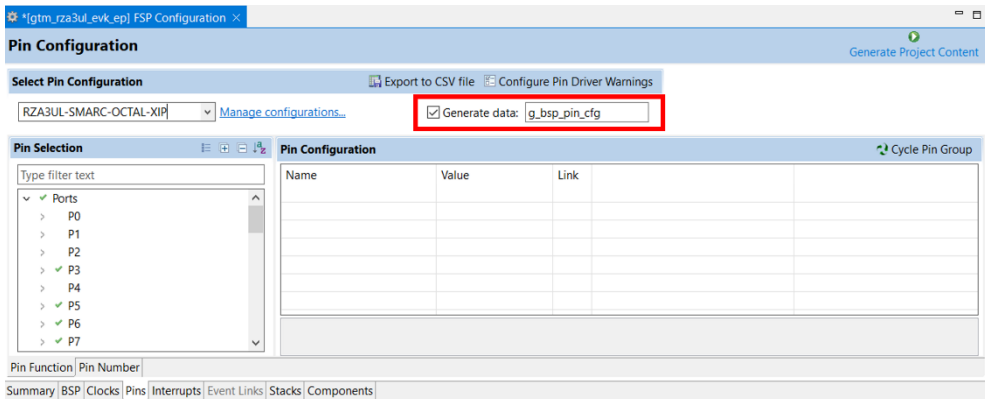
2. Uncheck the box in **Generate data** of FSP Configuration > Pins tab.



3. Change **Select Pin Configuration** to **RZA3UL-SMARC-OCTAL-XIP** in FSP Configuration > Pins tab.



4. Check the box and enter **g_bsp_pin_cfg** in **Generate data** of FSP Configuration > Pins tab.

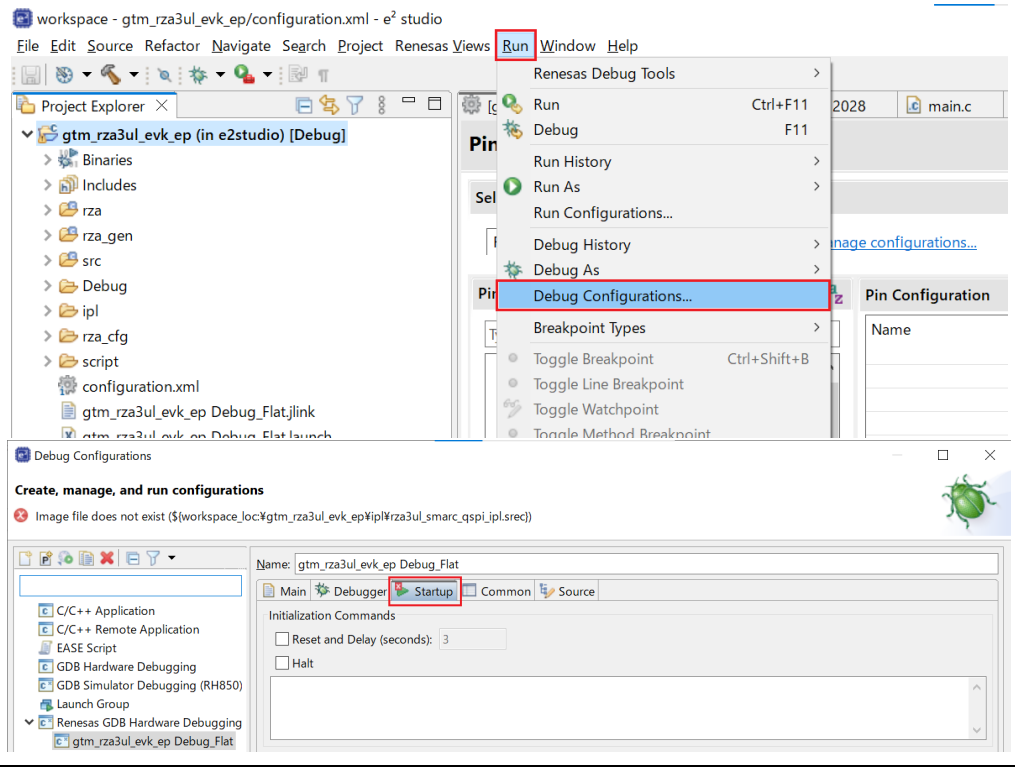


5. Click **Generate Project Content**.

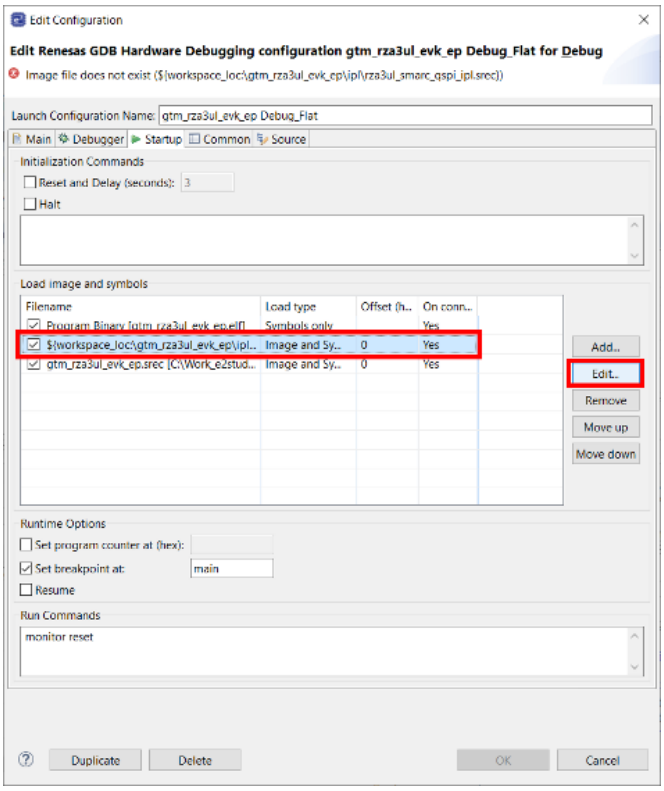
6. Build the project.

7. Change Loaded file of IPL for OCTAL edition.

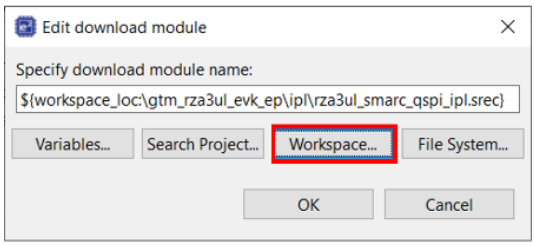
   1. Open **Debug Configurations** and move to **Startup** tab.

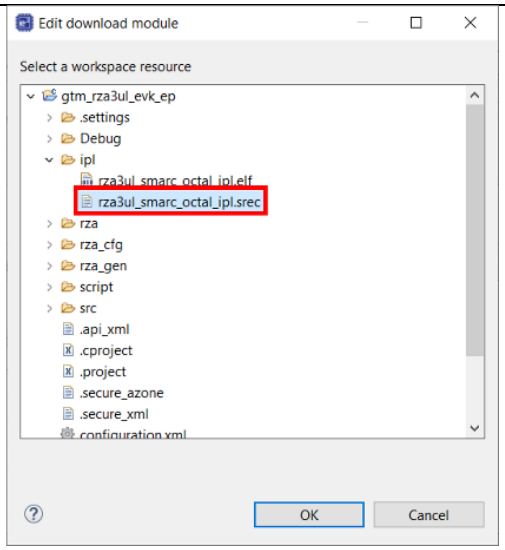


2. Click rza3ul_smarc_qspi_ipl.srec in **Load Image and Symbols** and click **Edit…**.

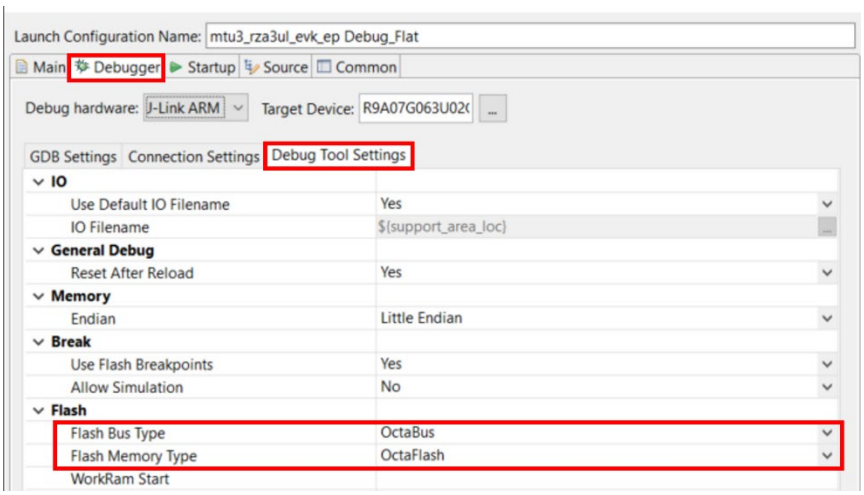


3. Click **Workspace…** and select project/ipl/rza3ul_smarc_octal_ipl.srec.





8. Then select the **Debug Tool Settings** tab inside the **Debugger** tab.
   Change **Flash Bus** Type to **OctaBus** (At this time, **Flash Memory Type** is automatically changed to **OctaFlash**).



9. Click OK and apply the Debug Configuration.

6. About Examples

6.1 ADC

6.1.1 Project Overview

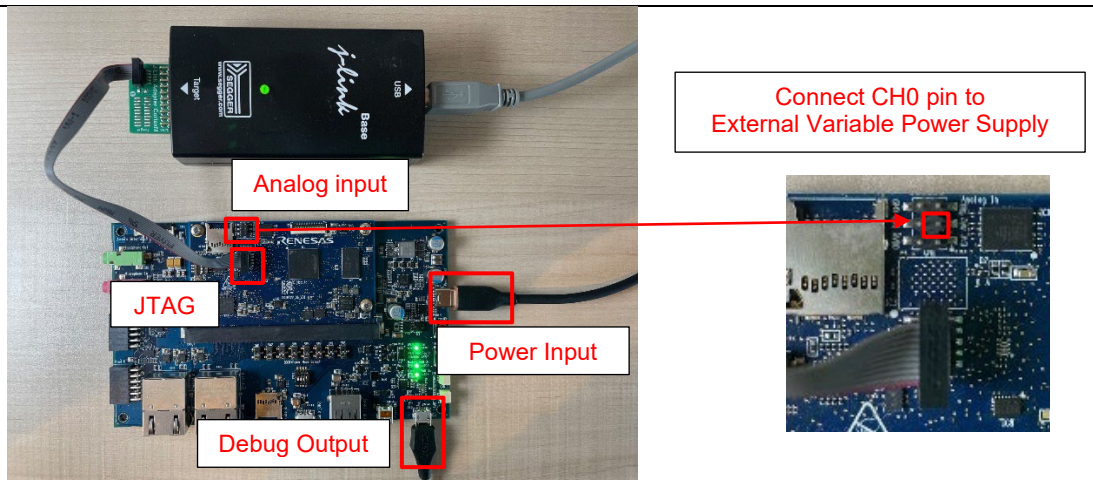
The example project demonstrates the typical use of the ADC HAL module APIs.
The project initializes the ADC in single scan or repeat scan mode based on user selection in RZ/A3UL configuration. Once initialized, the user can initiate the ADC scan and also stop the scan (in the case of repeat scan mode) using Tera Term by sending commands. Result and ADC status is displayed on the Tera Term.

6.1.2 Hardware Requirements

External hardware: External Variable Power Supply Unit (Rheostats 10K)

6.1.3 Hardware Settings

Please connect each hardware as below.



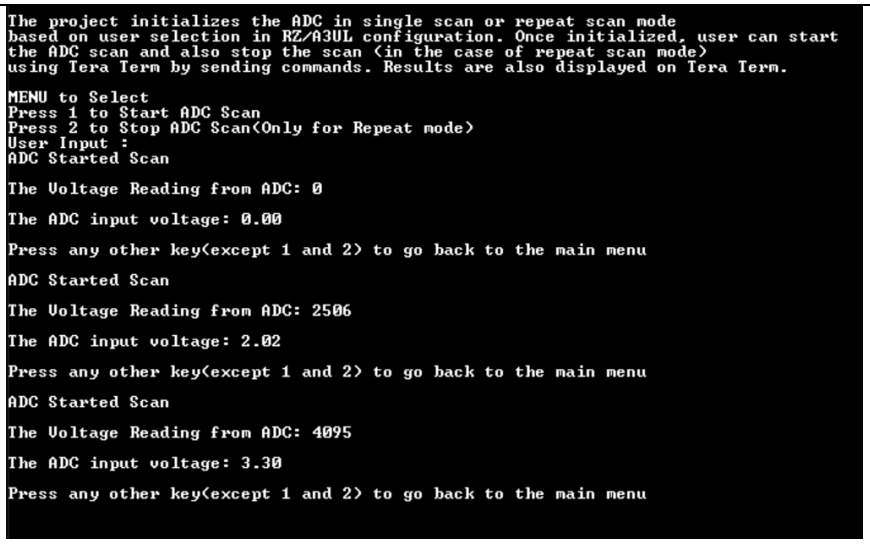
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.1.4 Operation

After running the example, the message below is displayed on console. Then, user can run each options.



6.2 CANFD

6.2.1 Project Overview

The example project shows the operation of CAN-FD running on Renesas RZ MCUs 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 the 4th transmission, Channel 1 transmits updated data to Channel 0 as ACK. Channel 0 displays the received data.

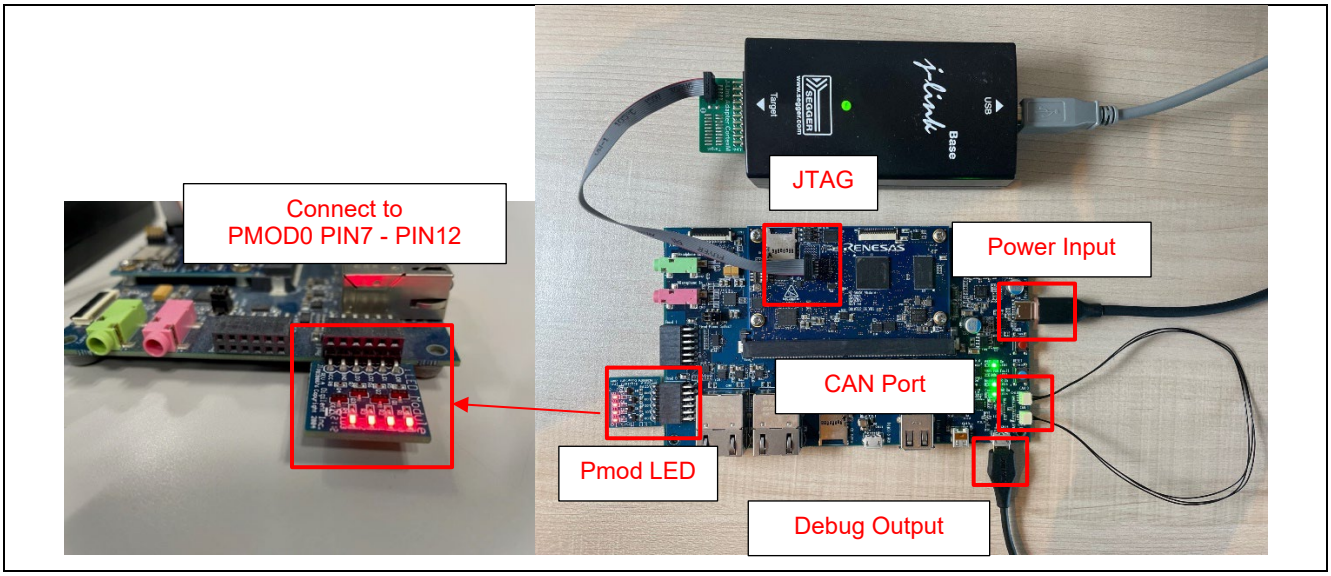
6.2.2 Hardware Requirements

External hardware: PMOD LED

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

6.2.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: OFF SW1-3: 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

6.2.4 Operation

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

```
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
```

6.3 RIIC Master

6.3.1 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 power up after establishing the connection of sensor with RZ/A3UL board, it displays accelerometer axis data on Terminal Emulator. Any API/event failure will be displayed on Terminal Emulator.
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.

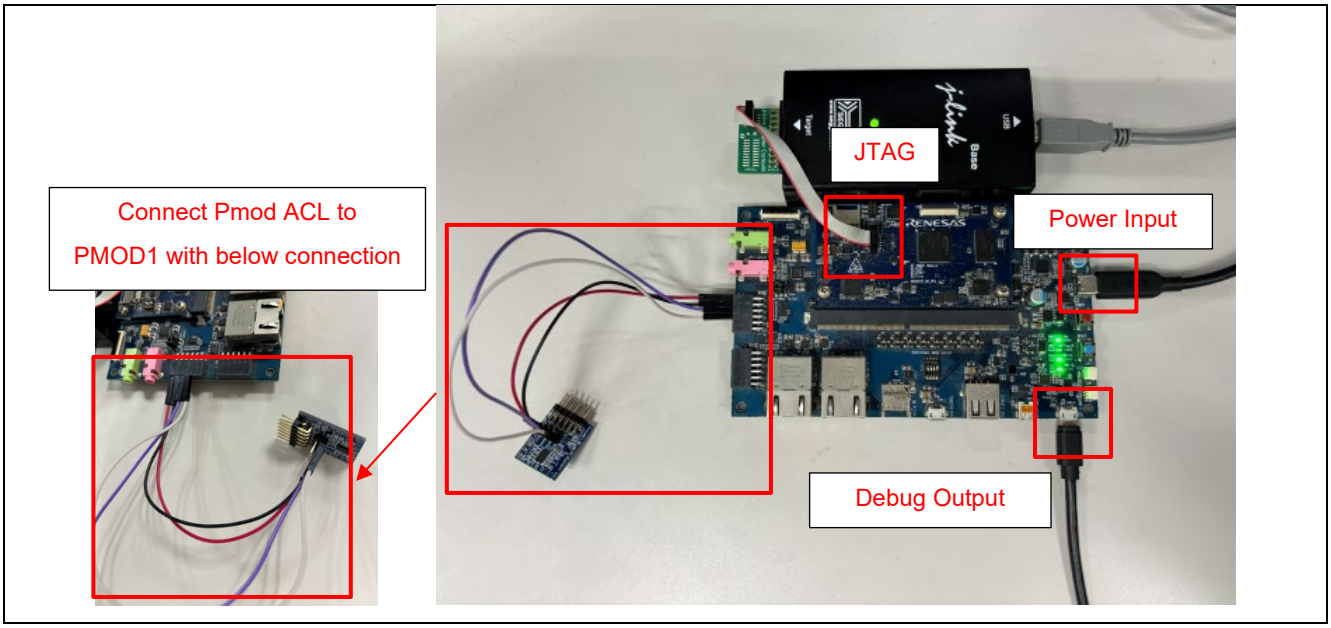
- Remove the DMAC sub stacks that linked to the main stack .

6.3.2 Hardware Requirements

External hardware: Pmod ACL

6.3.3 Hardware Settings

Please connect each hardware as below.



Board Setting

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

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.3.4 Operation

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

```
This EP 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 1 on Terminal Emulator to Open bus.

X-axis = 0.00, Y-axis = 0.00, Z-axis = 0.00
X-axis = 65530.00, Y-axis = 260.00, Z-axis = 65529.00
X-axis = 65533.00, Y-axis = 261.00, Z-axis = 65528.00
X-axis = 61.00, Y-axis = 173.00, Z-axis = 174.00
X-axis = 21.00, Y-axis = 65517.00, Z-axis = 245.00
X-axis = 65535.00, Y-axis = 259.00, Z-axis = 42.00
```

6.4 RIIC Slave

6.4.1 Project Overview

The example project needs two RZ/A3UL Evaluation Board Kit. One is called Slave Board, and the another is called Master Board.
The example project demonstrates typical use of the IIC slave HAL module APIs.
The project initializes IIC 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), transmitted and received data will be compared. Led blinks on data match else it is turned ON as sign of failure.
Output message for both boards corresponding slave operations will be displayed on Terminal Emulator.
Any API/event failure message will also be displayed.
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.

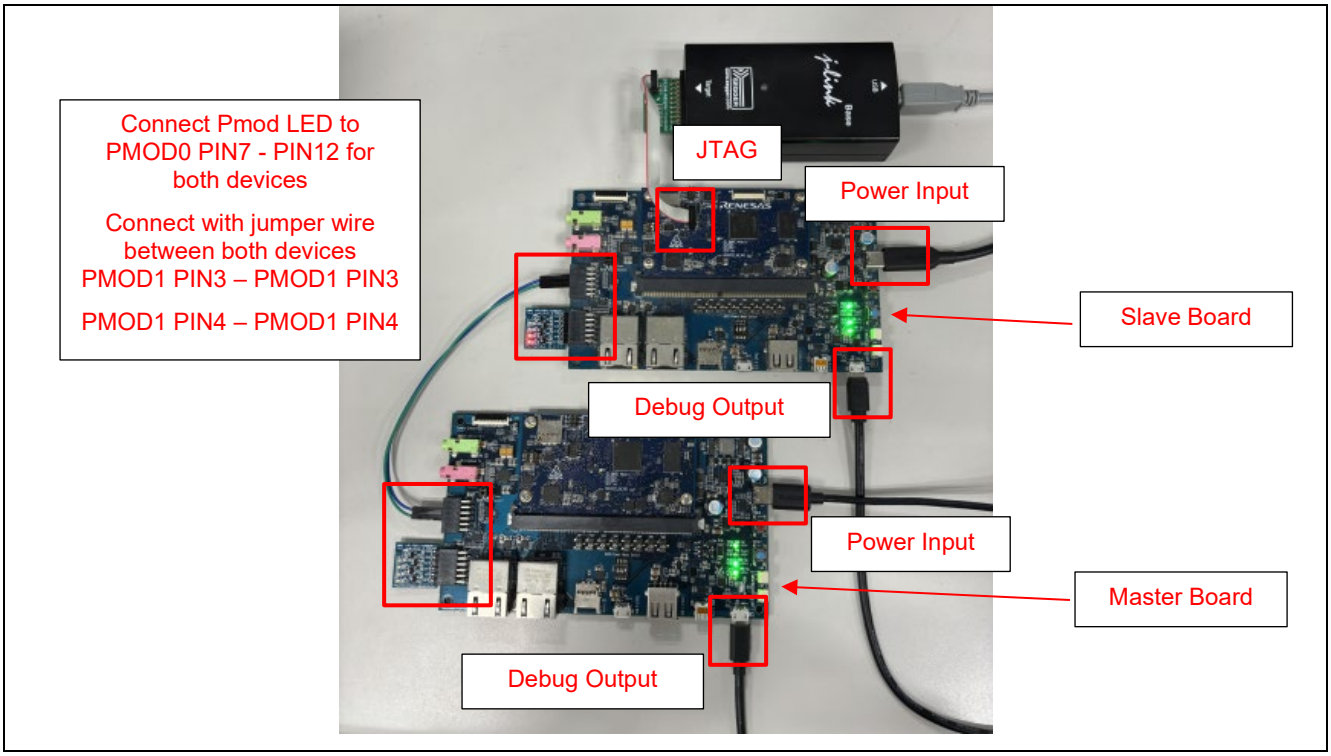
- Remove the DMAC sub stacks that linked to the main stack.

6.4.2 Hardware Requirements

External hardware: Two Pmod LED and two RZ/A3UL Evaluation Board Kit.

6.4.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper for both boards as below.

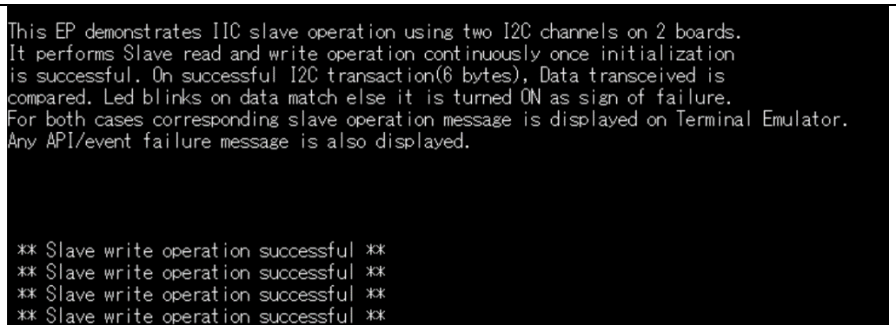
Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: 1-2. SW4: 1-2. SW5: Don't care. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4

6.4.4 Operation

Note: Need to run the example on Slave Board first, then run the example on Master Board. Make sure the slave board in operation when start running Master Board.

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

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



6.5 SSI

6.5.1 Project Overview

The example project demonstrates the typical use of the SSI HAL module APIs.
The project transfers sample audio data from source buffer to destination buffer by connecting Tx and Rx pins and compares the transferred data of two buffers.
The result of comparison will be displayed on Terminal Emulator.
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 Disable on the main stack of the driver in FSP Configuration.

- Remove the DMAC sub stacks that linked to the main stack .

Steps to enable again the DMAC feature after disabled it:

- Set DMAC Support to Enable on the main stack of the driver in FSP Configuration.

- Add Transfer Driver on r_dmac in the DMAC sub stacks that linked to the main stack

- In the g_transfer0 SSIF_DMA_TX0, set DMA Activation Request Source Select to Requested by a transfer destination module

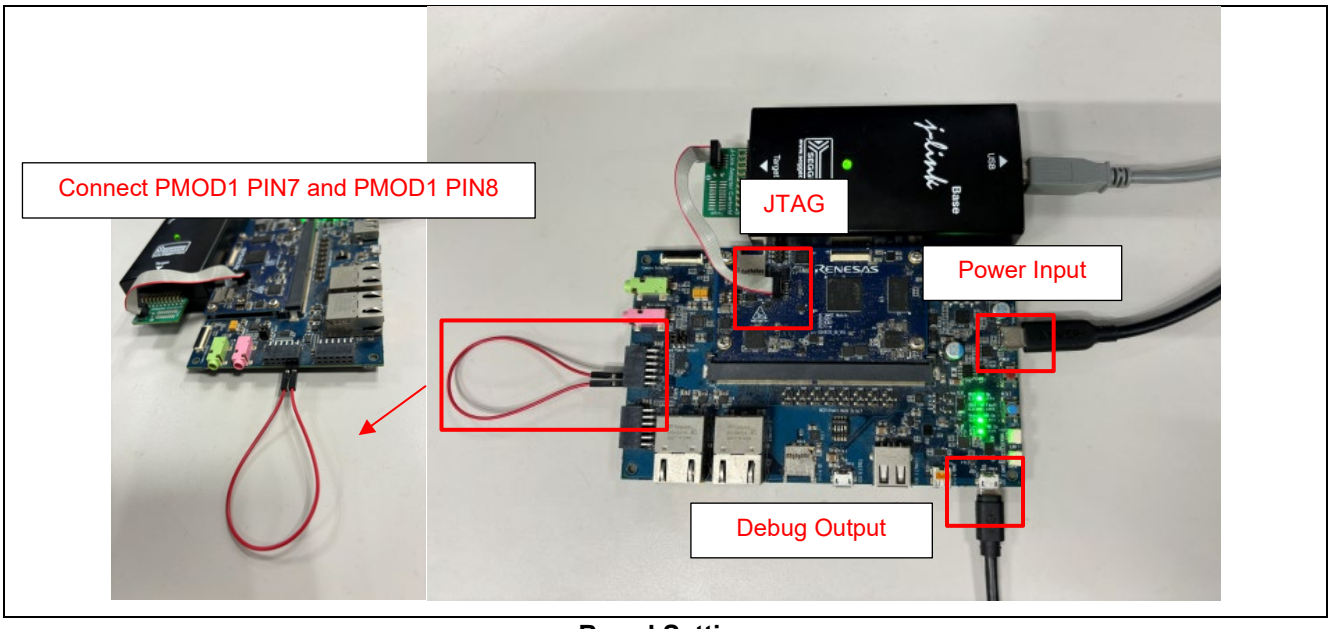
- In the g_transfer1 SSIF_DMA_RX0, set DMA Activation Request Source Select to Requested by a transfer source module

6.5.2 Hardware Requirements

External hardware: None

6.5.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: OFF
Carrier board	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

6.5.4 Operation

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

```
The project demonstrates SSI module by transmitting and receiving
the sample audio data in loop back connection and prints the status
by comparing the transmitted and received data buffers.

Start transmit data

Complete receive data

Compared the transmitted sample audio data of SSI with received data is successf
ul.
```

6.6 RSPI

6.6.1 Project Overview

The example project demonstrates the typical use of the RSPI HAL module APIs.
The project configures RSPI channels (Channel 1 and Channel 0) 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 Terminal Emulator.
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.

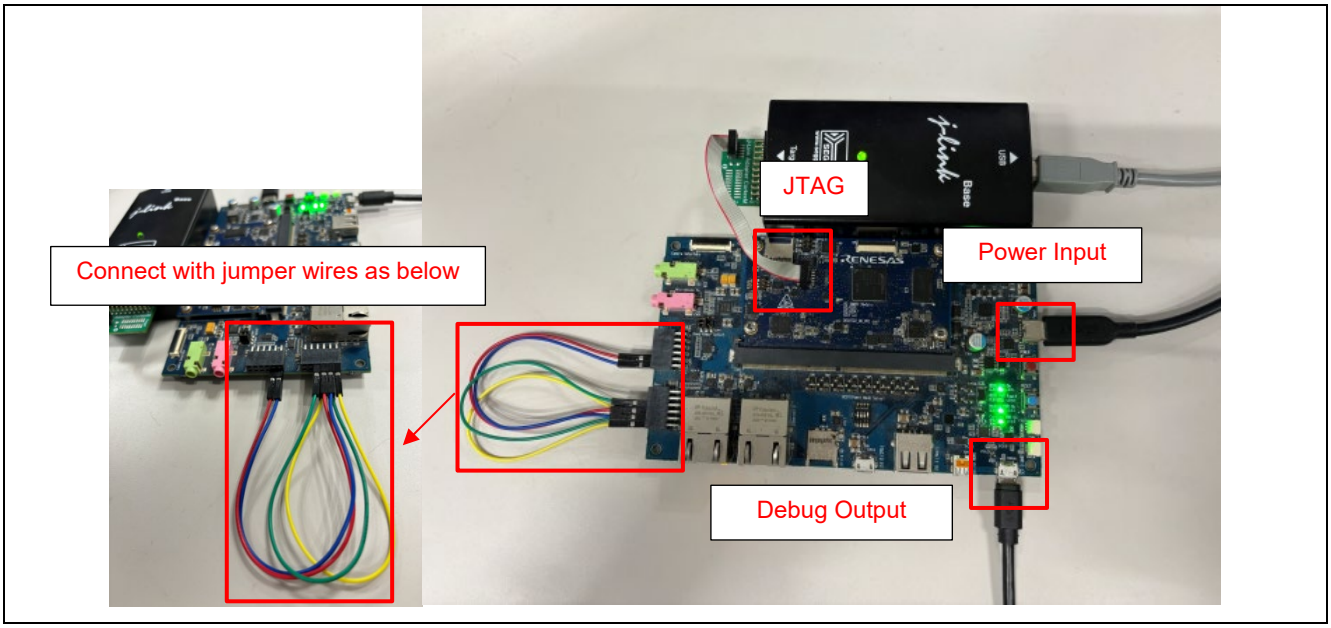
- Remove the DMAC sub stacks that linked to the main stack .

6.6.2 Hardware Requirements

External hardware: None

6.6.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set the Pin Connection as below.

RZ/A3UL EVK	
MISO:	PMOD0 PIN3 - PMOD1 PIN8
MOSI:	PMOD0 PIN2 - PMOD1 PIN7
CK:	PMOD0 PIN4 - PMOD0 PIN8
SSL:	PMOD0 PIN1 - PMOD0 PIN9

Please set each DIP switch and jumper as below.

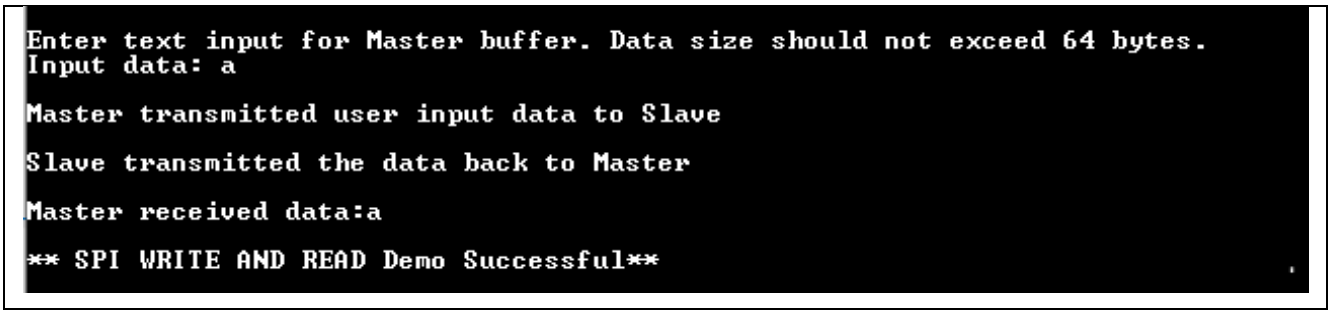
Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: 1-2. SW6: 1-2. SW7: 1-2. SW8: 1-2. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4

6.6.4 Operation

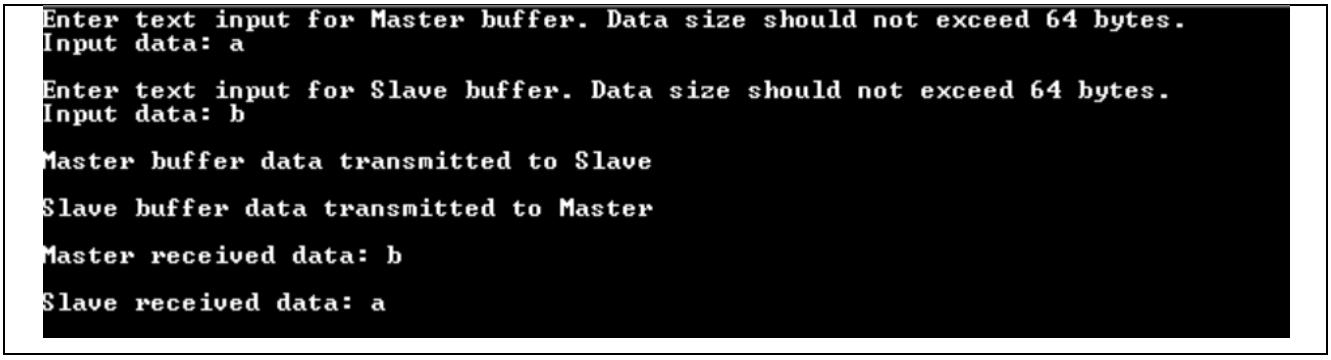
After running the example, the message below is displayed on console.

```
The project initializes SPI driver and configures SPI 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 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() example and transmission will be performed after input data from Master to Slave. Then, Slave sends back same data to Master.



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.



6.7 SCIg

6.7.1 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.
Using a Terminal Emulator user can provide a value & press enter key to set the cycle of the Pmod LED signal.
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 Disable on the main stack of the driver in FSP Configuration.

- Remove the DMAC sub stacks that linked to the main stack .

Steps to enable again the DMAC feature after disabled it:

- Set DMAC Support to Enable on the main stack of the driver in FSP Configuration.

- Add Transfer Driver on r_dmac in the DMAC sub stacks that linked to the main stack

- In the g_transfer0 SCIg_TXI0, set DMA Activation Request Source Select to Requested by a transfer destination module

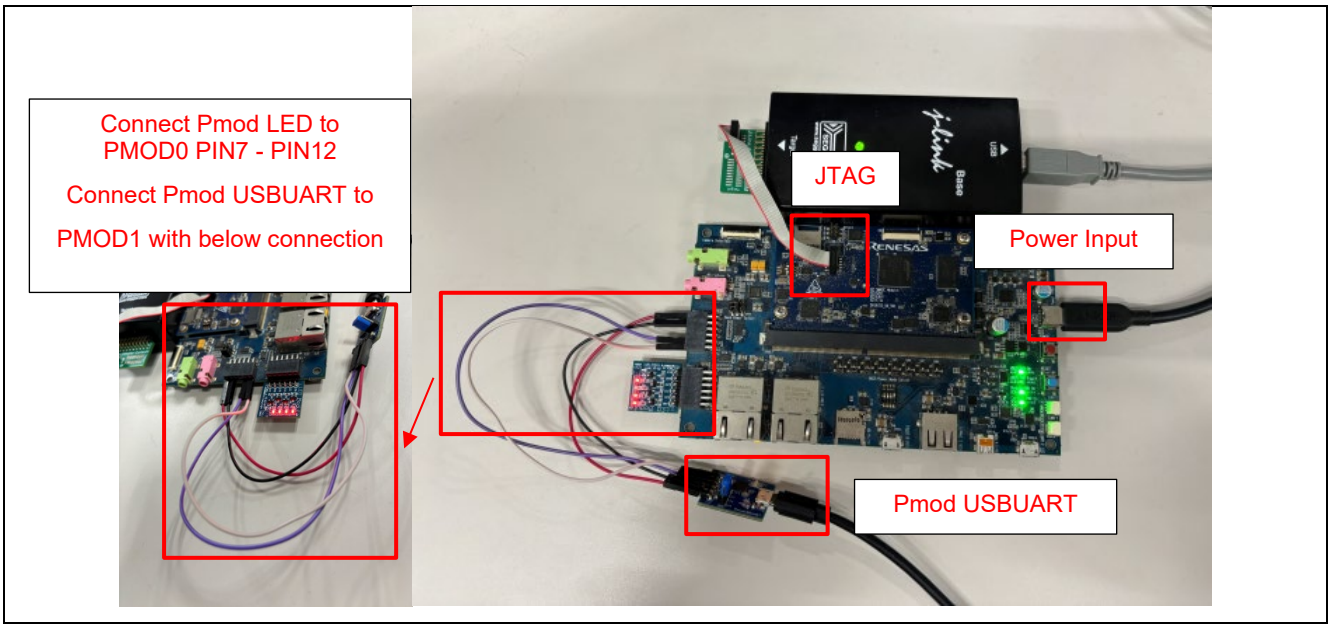
- In the g_transfer1 SCIg_RXI0, set DMA Activation Request Source Select to Requested by a transfer source module

6.7.2 Hardware Requirements

External hardware: Pmod USBUART and Pmod LED

6.7.3 Hardware Settings

Please connect each hardware as below.



Board Setting

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

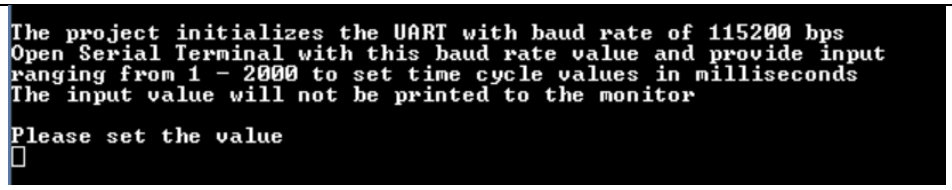
Connection of Pmod USBUART	
RXD:	Pmod USBUART J2 PIN2 – PMOD1 PIN7
TXD:	Pmod USBUART J2 PIN3 – PMOD1 PIN8
GND:	Pmod USBUART J2 PIN5 – PMOD1 PIN11
VCC:	Pmod USBUART J2 PIN6 – PMOD1 PIN12

Please set each DIP switch and jumper as below.

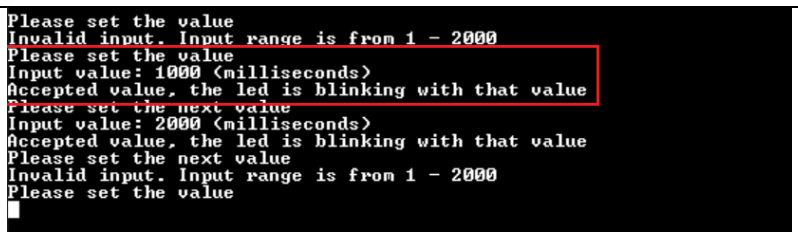
Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don't care. SW1-3: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: 1-2. SW6: 1-2. SW7: 1-2. SW8: 1-2. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4

6.7.4 Operation

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



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



6.8 SCIF

6.8.1 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.
Using a Terminal Emulator user can provide a value & press enter key to set the cycle of the Pmod LED signal.
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.

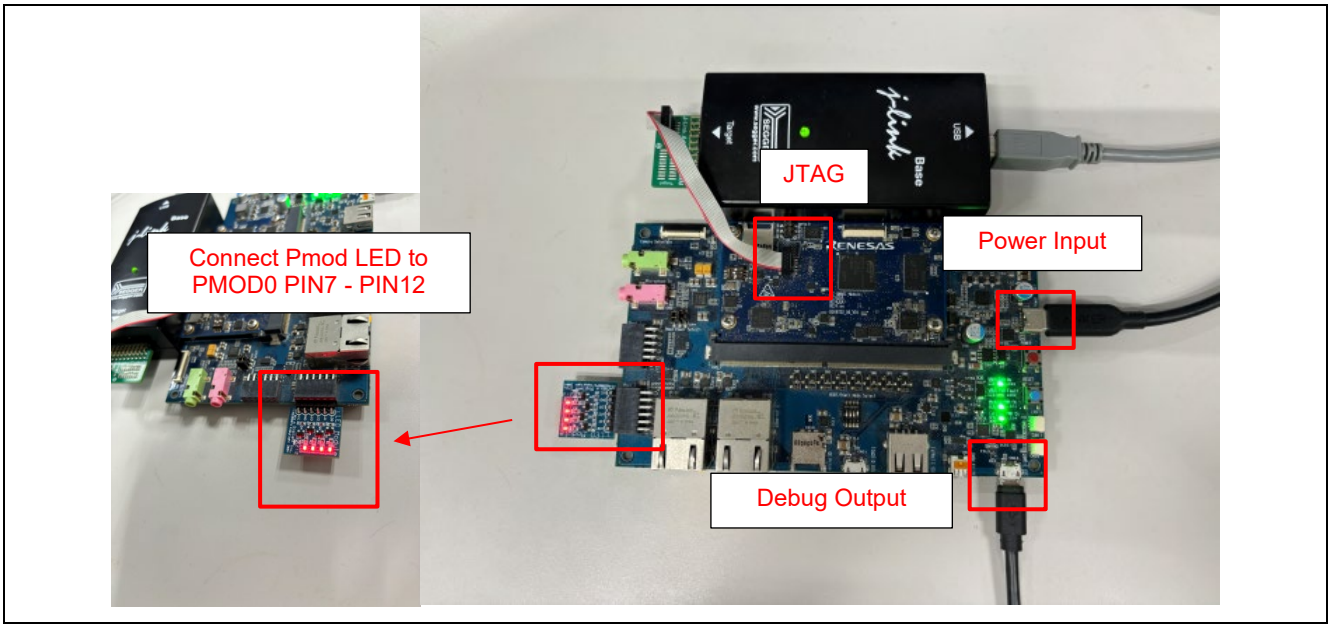
- Remove the DMAC sub stacks that linked to the main stack .

6.8.2 Hardware Requirements

External hardware: Pmod LED

6.8.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.8.4 Operation

After running the example, the message below is displayed on console. User can specify a value for LED blinking on Terminal Program.

```
The project initializes the UART with baud rate of 115200 bps
Open Serial Terminal with this baud rate value and
Provide input ranging from 1 - 2000 to set time cycle values in milliseconds
The input value will not be printed to the monitor

Accepted value, the led is blinking with that value
Please set the next value
```

6.9 USB HHID

6.9.1 Project Overview

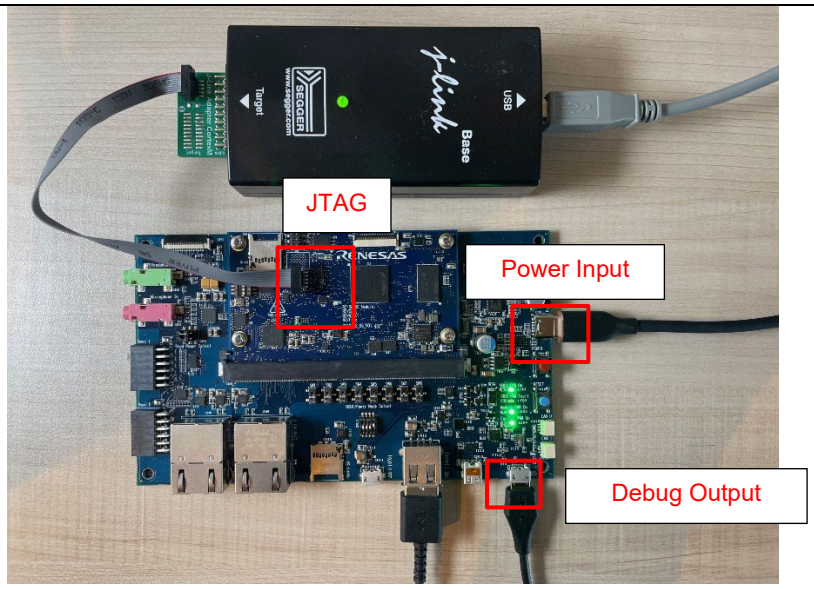
This example project demonstrates basic functionalities of USB HHID driver with FreeRTOS on Renesas RZ/A3UL.
MCUs based on Renesas FSP. USB HHID driver configures keyboard as a hhid device. On pressing key from the keyboard, received keycode is decoded and displayed on the Terminal Emulator. Error and info messages will be printed on Terminal Emulator.

6.9.2 Hardware Requirements

External hardware: USB PC Keyboard

6.9.3 Hardware Settings

Please connect each hardware as below.



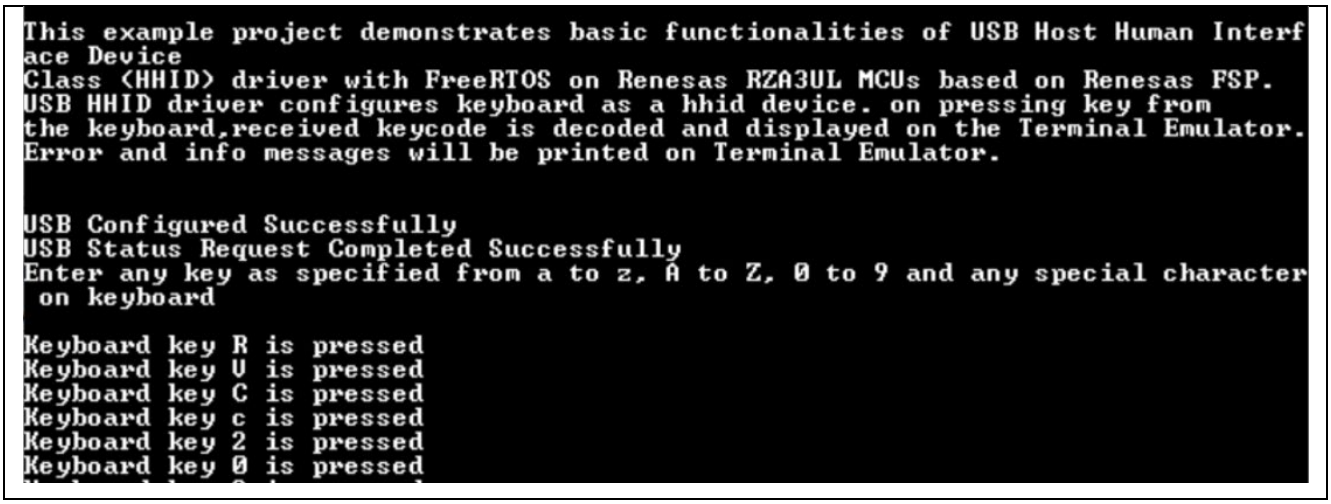
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.9.4 Operation

After running the example, the message below is displayed on console. Then, user can type some character on keyboard.



6.10 USB HMSC

6.10.1 Project Overview

The sample code accompanying this file shows the operation of USB_HMSC running on Renesas RZ MCUs using RZ/A3UL board, USB.
Format the USB driver with FAT32 file system before performing any operation
After successful USB connection with PC, the Terminal Emulator shows menu options with 4 options.

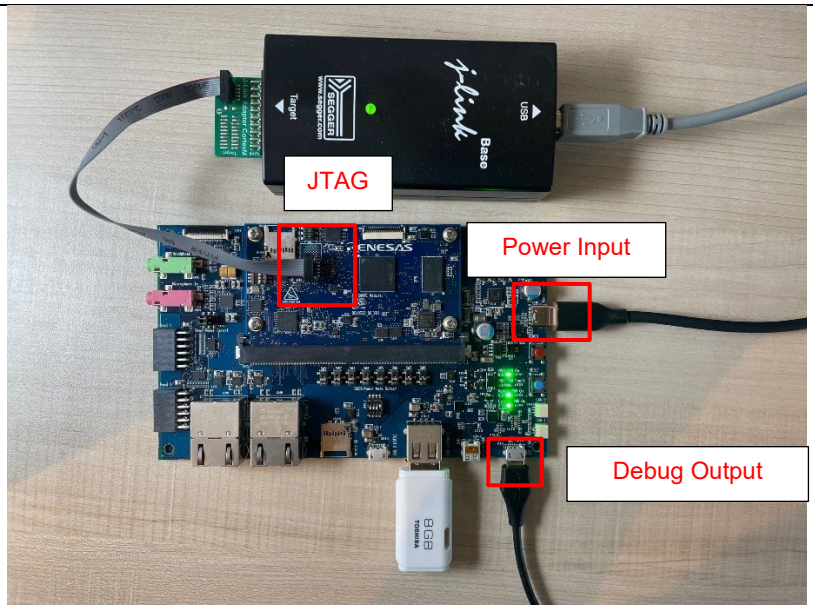
- The user select option 1 to write 10K data from app_buffer.
- The user enters data into the Terminal Emulator, and the data is saved to the rza_usb.txt file.
- The user select option 2 to Format USB Driver.
- The user select option 3 to Safely Eject the USB Driver before removing USB.
- The user select option 4 to Initialize FreeRTOS+FAT (Valid only after Safely_Eject option is executed)

6.10.2 Hardware Requirements

External hardware: USB MSC Device.

6.10.3 Hardware Settings

Please connect each hardware as below.



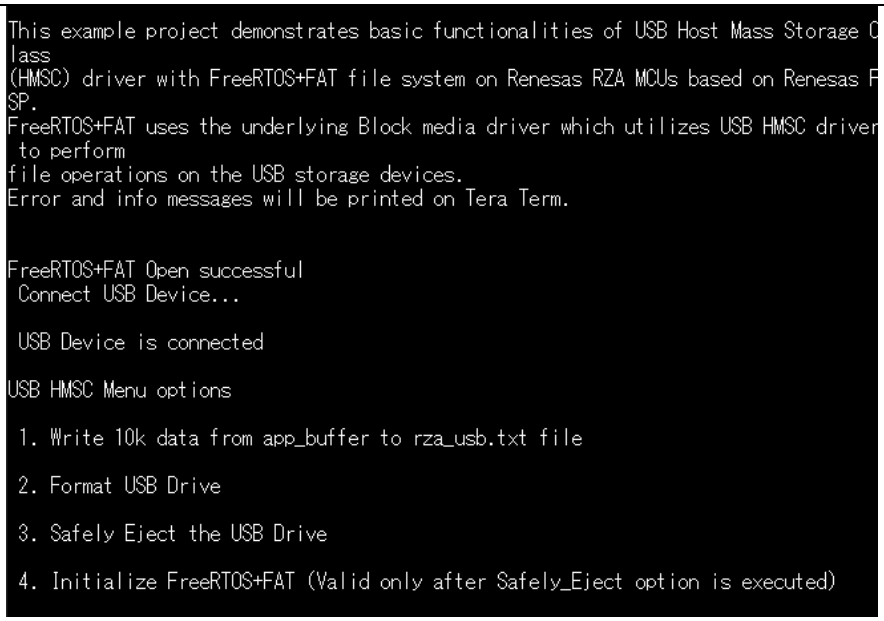
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.10.4 Operation

After running the example, the message below is displayed on console. Then, user can run each options.



6.11 USB PCDC

6.11.1 Project Overview

This Example Projects demonstrates the basic functionalities of the USB_PCDC driver on Renesas RZA MCUs based on Renesas FSP

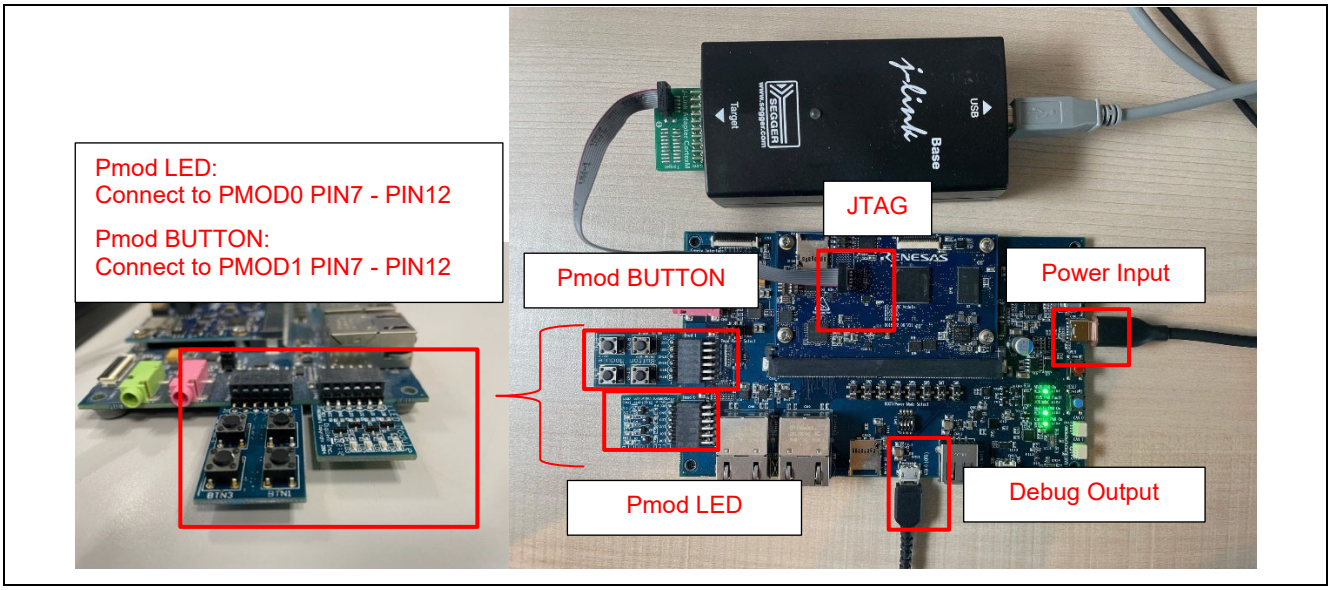
- On power up or RESET, all LEDs on PMOD LED start blinking at 1Hz.
- Press BTN0 button of PMOD BUTTON to change the blinking frequency of the PMOD LED.
  With every press BTN0 button, the frequency will switch from 1 Hz to 5 Hz to 10 Hz and cycle back.
- Open the serial port of the device using Terminal Emulator and press Enter key (keyboard on host PC).
  Press 1: Show kit information (show kit name, blinking frequency).
  Press 2: Show URLs to user visit.

6.11.2 Hardware Requirements

External hardware: PMOD LED, PMOD BUTTON.

6.11.3 Hardware Settings

Please connect each hardware as below.



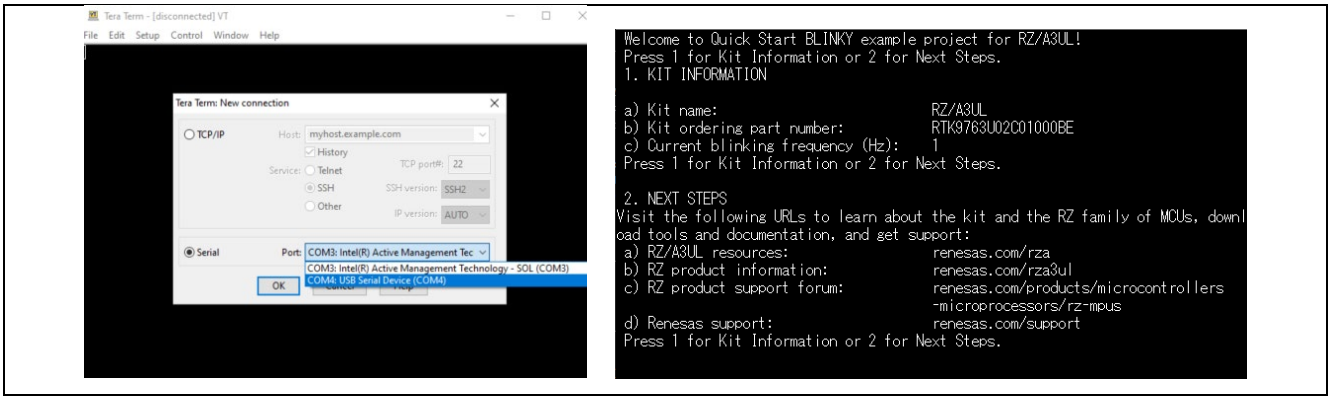
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.11.4 Operation

After running the example, plesae connect terminal with “USB Serial Device” as below.
After the connection, please input enter on the console, then user can see the message and run the options.
Also, the blinking frequency of the Pmod LED can be changed by pressing the BTN0 of Pmod BUTTON.
With every press BTN0 button, the frequency will switch from 1 Hz to 5 Hz to 10 Hz and cycle back.



6.12 LCDC

6.12.1 Project Overview

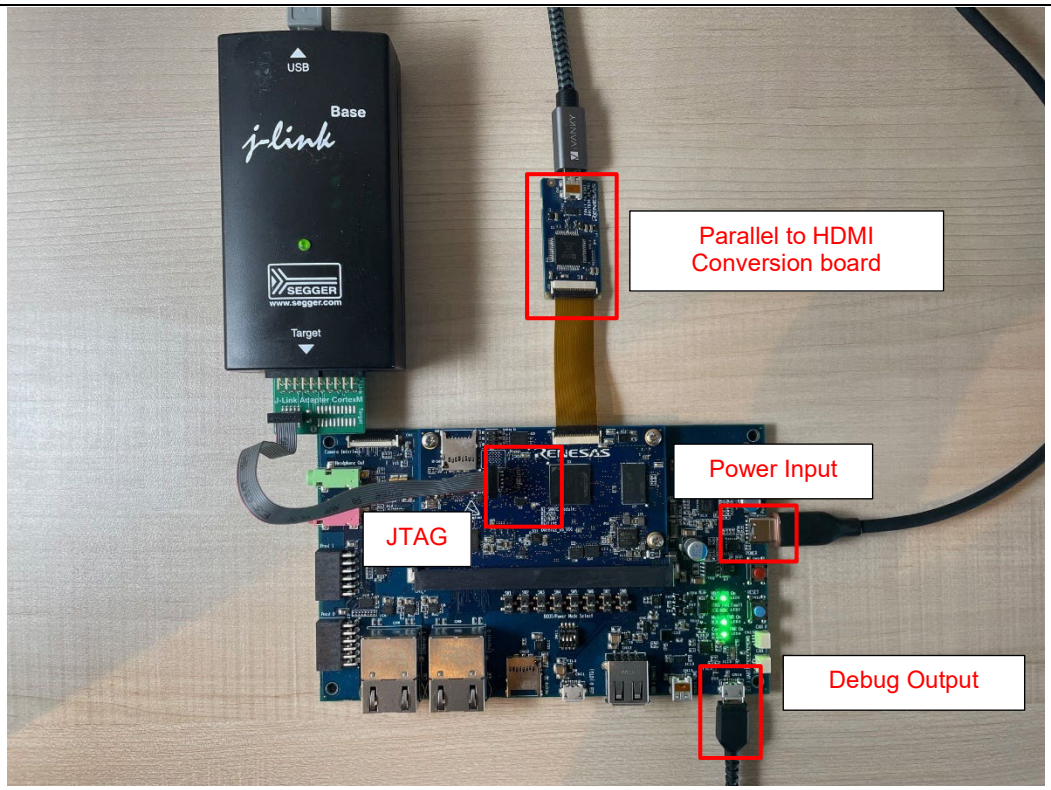
This Example Project demonstrates the basic functionalities of LCDC device driver on Renesas RZ/A3UL MCUs based on Renesas FSP.
On successful initialization of LCDC module, the EP will be able to read the display resolution and frame buffer set in the LCDC module and draw color bands to cover the entire screen of external LCD connected to RZ MCU. API Failure messages will be displayed on Terminal Emulator.

6.12.2 Hardware Requirements

External hardware: Mini HDMI to HDMI Cable, Parallel to HDMI Conversion board, A display supports HDMI port.

6.12.3 Hardware Settings

Please connect each hardware as below.



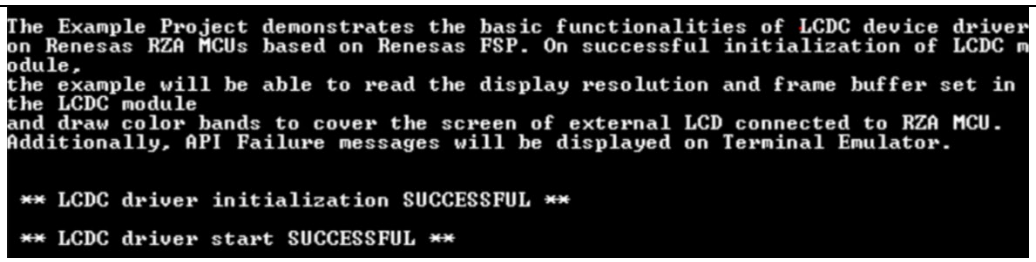
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.12.4 Operation

After running the example, the message below is displayed on console. And external display will show the color bar image.



6.13 INTC IRQ

6.13.1 Project Overview

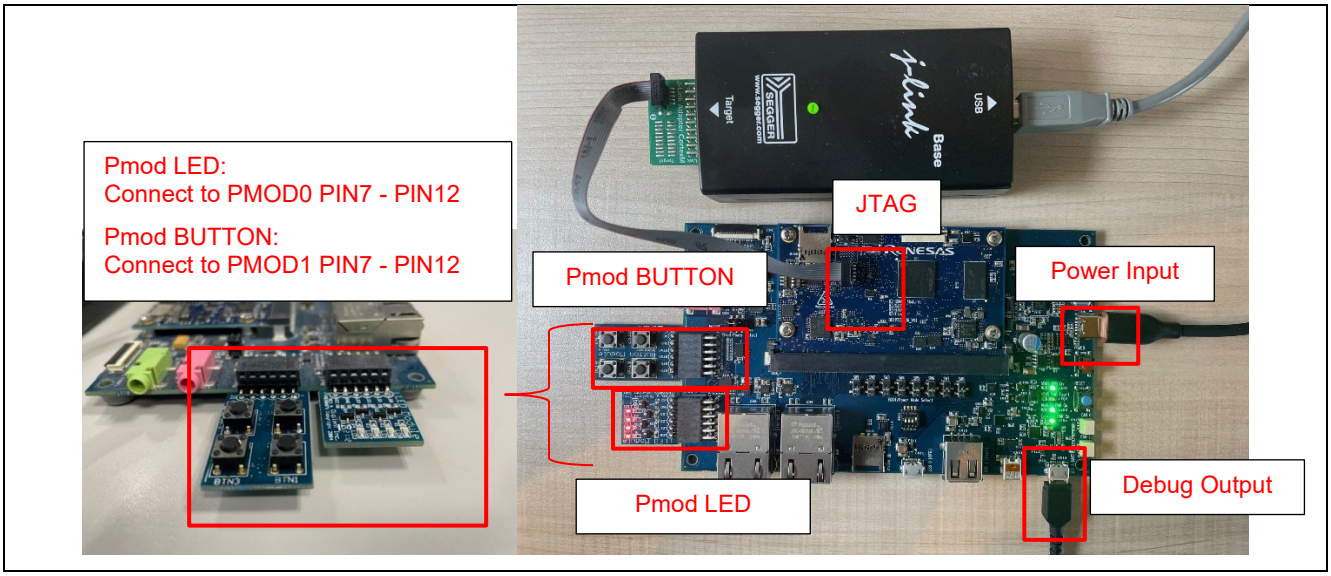
The example project demonstrates the typical use of the INTC IRQ module APIs.
The project initializes the IRQ interrupt in Interrupt Controller User is requested to press the push button to trigger the external irq and this then will start toggling of user LED.

6.13.2 Hardware Requirements

External hardware: PMOD LED, PMOD BUTTON

6.13.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.13.4 Operation

After running the example, the message below is displayed on console. When pressing the BTN0 button of Pmod BUTTON, the LED status is displayed.

```
This Example Project demonstrates the functionality of INTC_IRQ driver.
On pressing the user push button, an external IRQ is triggered, which toggles on
-board LED.

User Pushbutton Pressed
LED State: Low(OFF)

User Pushbutton Pressed
LED State: High(ON)

User Pushbutton Pressed
LED State: Low(OFF)

User Pushbutton Pressed
LED State: High(ON)

User Pushbutton Pressed
LED State: Low(OFF)

User Pushbutton Pressed
LED State: High(ON)

User Pushbutton Pressed
LED State: Low(OFF)

User Pushbutton Pressed
LED State: High(ON)

User Pushbutton Pressed
LED State: Low(OFF)
```

6.14 INTC TINT

6.14.1 Project Overview

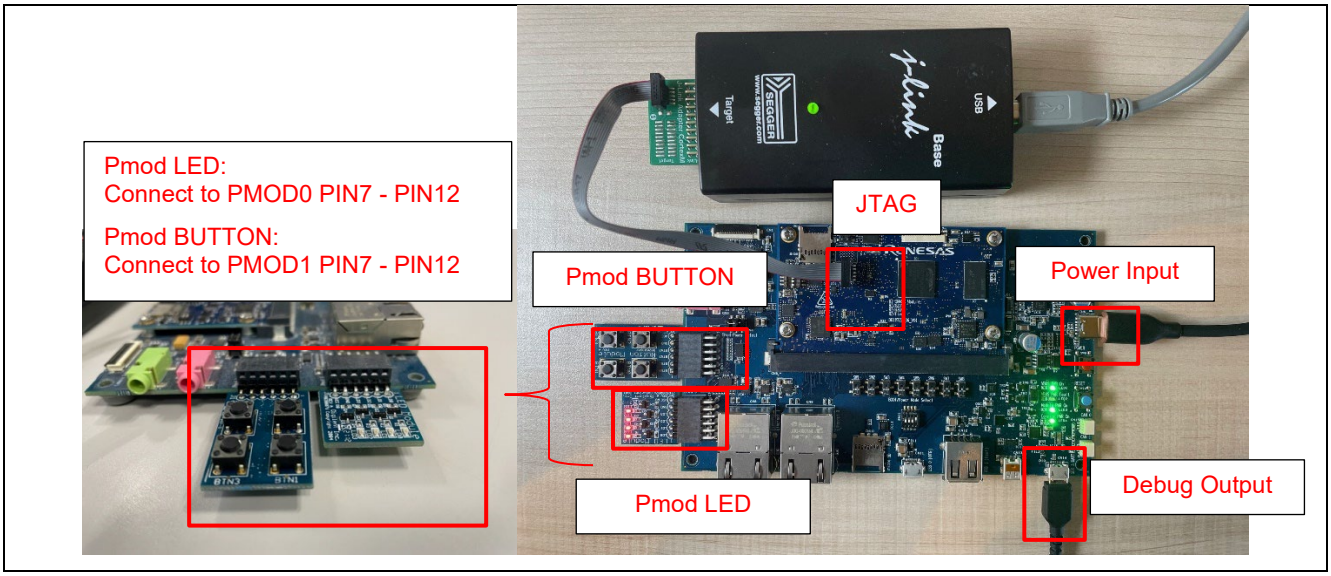
The example project demonstrates the typical use of the INTC TINT module APIs.
The project initializes the TINT interrupt in Interrupt Controller.
The user is requested to press the push-button to trigger the external tint and then will start toggling of LEDs.

6.14.2 Hardware Requirements

External hardware:

6.14.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.14.4 Operation

After running the example, the message below is displayed on console. When pressing the BTN0 button of Pmod BUTTON, the LED status is displayed.

```
This Example Project demonstrates the functionality of INTC_TINT driver.
On pressing the user push button, an external IRQ is triggered, which toggles on-board LED.

User Pushbutton Pressed
LED State: Low[OFF]

User Pushbutton Pressed
LED State: High[ON]

User Pushbutton Pressed
LED State: Low[OFF]

User Pushbutton Pressed
LED State: High[ON]
```

6.15 WDT

6.15.1 Project Overview

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 (LD0 on PMOD LED).

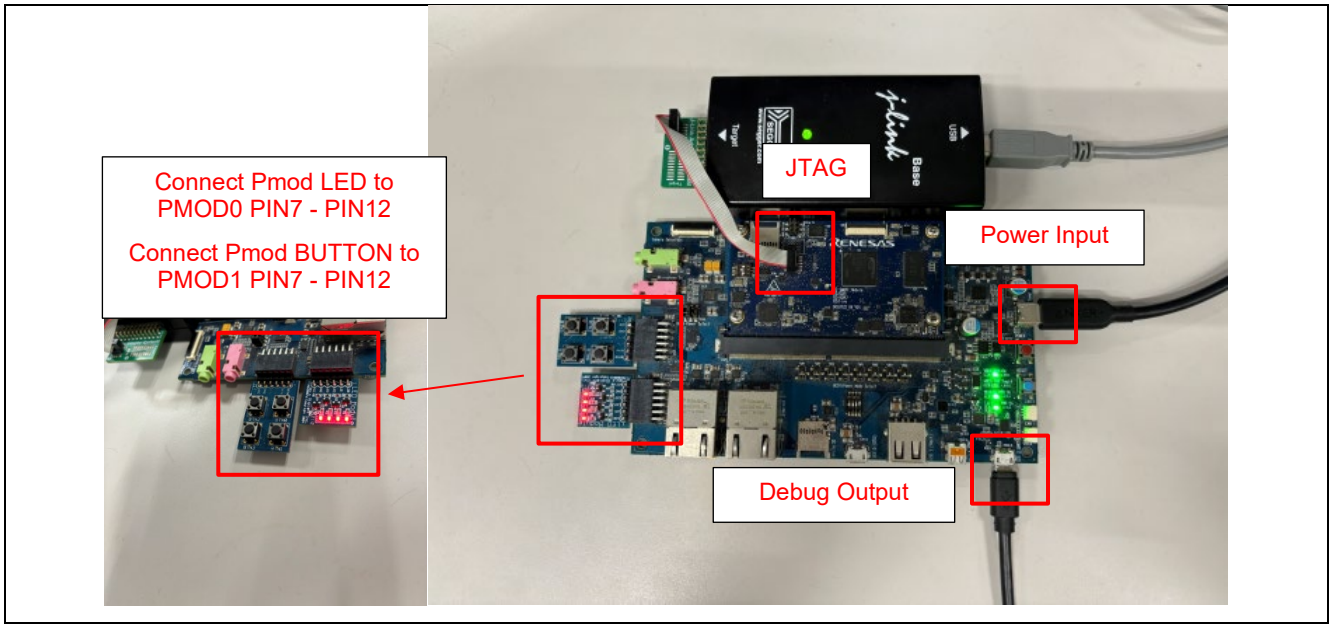
- If the button BTN0 (on PMOD BUTTON) is pressed, IRQ is triggered to stop GTM timer, and after 2 seconds, WDT will reset MPU.

6.15.2 Hardware Requirements

External hardware: Pmod LED and Pmod BUTTON

6.15.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

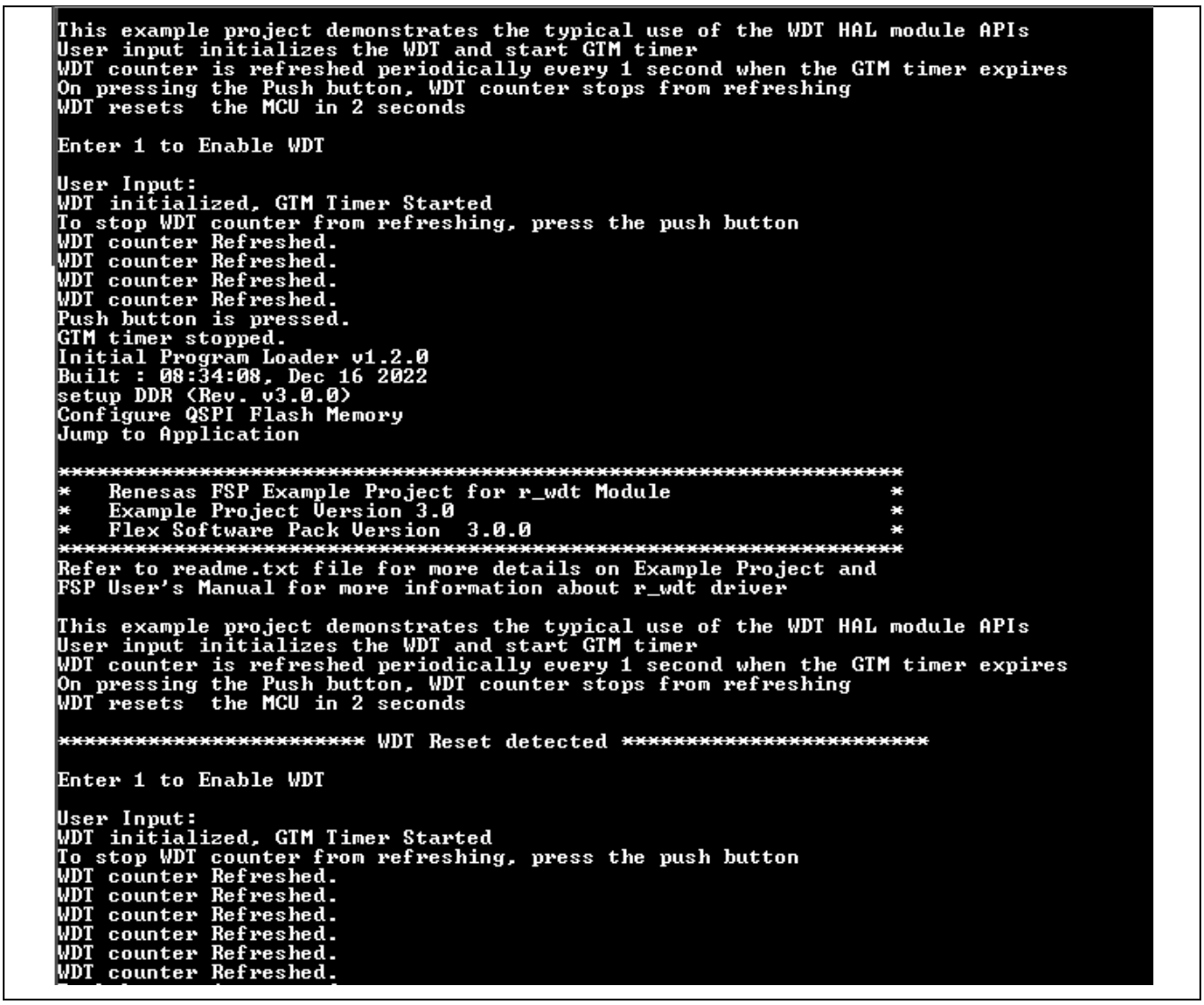
Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: OFF
Carrier board	SW1: Don't care. SW2: Don't care. SW3: Don't care. SW4: Don't care. SW5: 3-2. SW6: Don't care. SW7: Don't care. SW8: Don't care. SW11-1: OFF SW11-2: OFF SW11-3: OFF SW11-4: ON CN4: Jumper connects 1-3 Jumper connects 2-4

6.15.4 Operation

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

```
This example project demonstrates the typical use of the WDT HAL module APIs
User input initializes the WDT and start GTM timer
WDT counter is refreshed periodically every 1 second when the GTM timer expires
On pressing the Push button, WDT counter stops from refreshing
WDT resets  the MCU in 2 seconds

Enter 1 to Enable WDT

User Input:
```

After inputting “1” to initial WDT and start GTM timer, the GTM timer will refresh WDT counter and blink the LED2 of Pmod LED. When pressing the BTN3 of Pmod BUTTON, IRQ is triggered to stop GTM timer, and after 2 seconds, WDT will reset MPU. At the same time, the WDT Reset will be detected in the next launch.



6.16 Ether

6.16.1 Project Overview

The example project demonstrates the use of the FreeRTOS + TCP Module with Ethernet.
The example project demonstrates the usage of FreeRTOS DHCP Client, DNS Client and ICMP code.

- This Example Project has 2 modes for the user:
1. DHCP Mode, where the IP credentials are obtained from the DHCP Server on the Network.
2. Static IP Address mode, where in the Static IP address can be configured for the Ethernet based on the Availability of free address in the LAN.

Besides this example also checks, calculates, and shows the send and response time between the target board and server.

NOTE: User is expected to enter the static IP address according to the network settings of the LAN.
For example, set up Static IP Configuration in net_thread_entry.c as below:

```
static uint8_t ucMACAddress[6] = {0x00, 0x11, 0x33, 0x55, 0x77, 0x99};
static uint8_t ucIPAddress[4] = {192, 168, 3, 109};
static uint8_t ucNetMask[4] = {255, 255, 255, 0};
static uint8_t ucGatewayAddress[4] = {192, 168, 3, 1};
static uint8_t ucDNSServerAddress[4] = {8, 8, 8, 8};
```

In case of DHCP, User just needs to run the application and DHCP client will communicate to the DHCP server and gets the IP address.

NOTE: While running the application using DHCP or Static Address mode, The following settings needs to be done in the configurator. The same projects can be used for both the settings.

DHCP mode

At FreeRTOS+TCP config stack change to table below:

FreeRTOS+TCP	Setting
Use DHCP	Enable
DHCP Register Hostname	Enable
DHCP Uses Unicast	Enable
DHCP Send Discover After Auto IP	Enable
DHCP callback function	Enable

Static IP Address mode

After run DHCP Mode we change IPv4 Address, Subnet Mask, Default Gateway, DNS Servers for Static IP config depending DHCP config printed at the console. At FreeRTOS+TCP config stack change to table below:

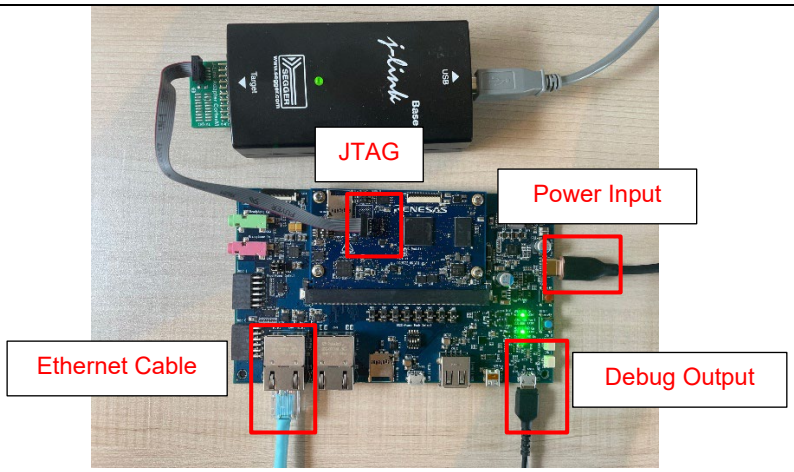
FreeRTOS+TCP	Setting
Use DHCP	Disable
DHCP Register Hostname	Disable
DHCP Uses Unicast	Disable
DHCP Send Discover After Auto IP	Disable
DHCP callback function	Disable

6.16.2 Hardware Requirements

External hardware: Ethernet Cable

6.16.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don't care. SW1-3: 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

6.16.4 Operation

After running the example, the message below is displayed on console. After connection successfully, the ping round trip time will be displayed.

```
Refer to readme.txt file for more details on Example Project and
RZA FSP Documentation for more information about Ethernet using FreeRTOS+TCP

********************************************************************************


--------------------------------------------------------------------------------
Ethernet adapter Configuration for Renesas RZ/A3UL-SMARC: Pre IP Init
--------------------------------------------------------------------------------


Ethernet adapter for Renesas RZ/A3UL-SMARC:
        Description . . . . . . . . . . . : Renesas RZ/A3UL-SMARC Ethernet
        Physical Address. . . . . . . . . : 00-11-33-55-77-99
        DHCP Enabled. . . . . . . . . . . : No
        IPv4 Address. . . . . . . . . . . : 0.0.0.0
        Subnet Mask . . . . . . . . . . . : 0.0.0.0
        Default Gateway . . . . . . . . . : 0.0.0.0
        DNS Servers . . . . . . . . . . . : 0.0.0.0


--------------------------------------------------------------------------------
Ethernet adapter Configuration for Renesas RZ/A3UL-SMARC: Post IP Init
--------------------------------------------------------------------------------


Network is Down.....................................................................
.....................................................................
Network is Up
Connecton: 100Mbps Full Duplex
```

6.17 FreeRTOS

6.17.1 Project Overview

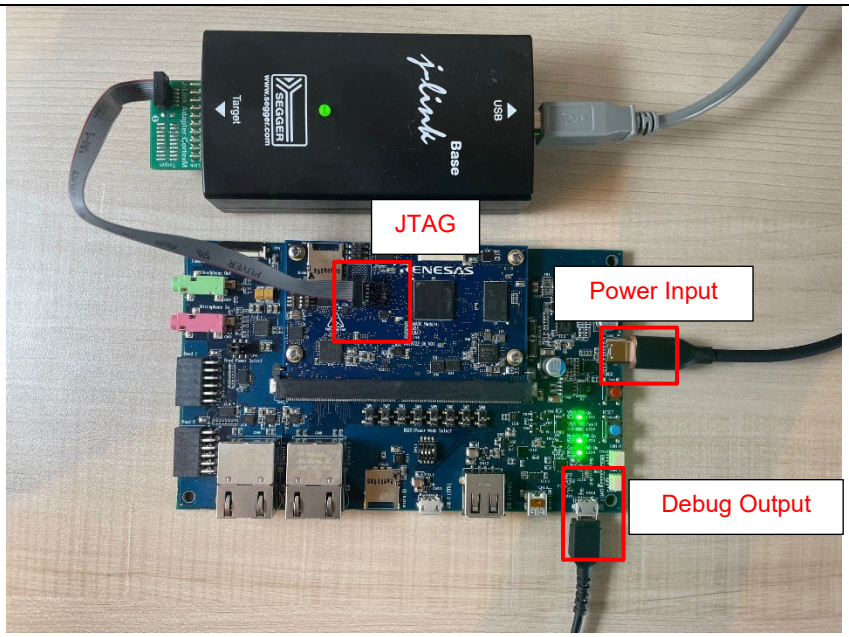
The example projects demonstrate Message Queue and Semaphore between tasks and interrupt. Message Queue is demonstrated between Tasks and between Task and interrupt. GTM timer periodically generates interrupt at 1000msec.
For the first few seconds, messages are shared between Sender and Receiver Tasks 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 sempahore until it is released by GTM ISR1. GTM ISR1 releases semaphore periodically at 1000msec. All the tasks run with equal priority level.

6.17.2 Hardware Requirements

External hardware: None.

6.17.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.17.4 Operation

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

```
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 : 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>  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
```

6.18 SDHI

6.18.1 Project Overview

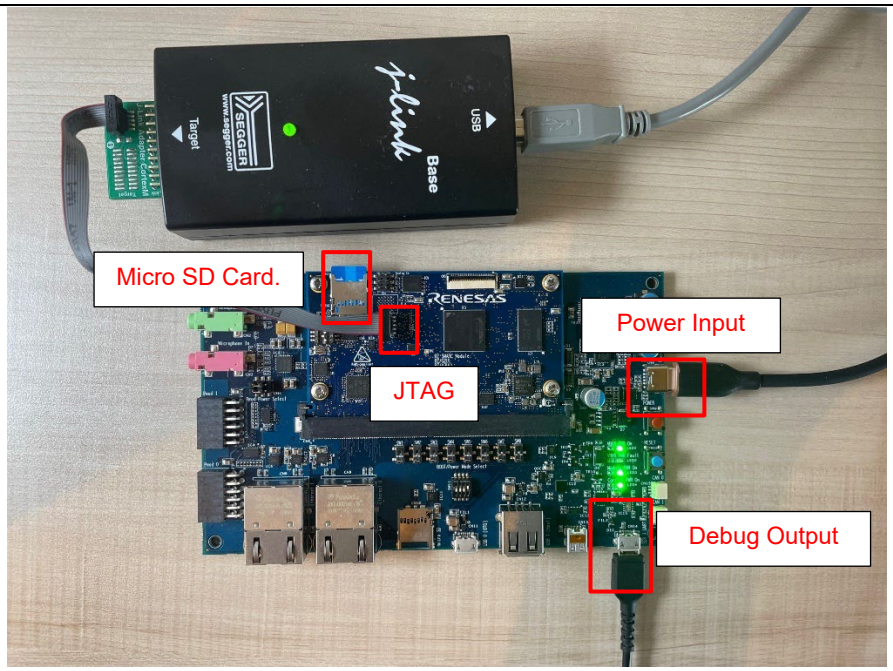
This example project demonstrates basic functionalities of sdhi driver with FreeRTOS+FAT file system on Renesas RZ/A3UL MPU based on Renesas FSP.
FreeRTOS+FAT uses the underlying Block media driver. The Block media driver utilizes the SDHI driver to perform file operations on the SD Card. Error and info messages will be printed on Terminal Emulator.

6.18.2 Hardware Requirements

External hardware: Micro SD Card.

6.18.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: ON SW1-3: 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

6.18.4 Operation

After running the example, the message below is displayed on console. User can select each option after connecting Micro SD card.

```
Refer to readme.txt file for more details on Example Project
for more information about SDHI driver
This example project demonstrates basic functionalities of sdhi
driver with FreeRTOS+FAT file system on Renesas RZ/A MPU based on Renesas FSP.
FreeRTOS+FAT uses the underlying Block media driver which utilizes SDHI driver to perform
file operations on the SD Card devices.
Error and info messages will be printed on Terminal Emulator.

Please remove and re-insert the SD Card after executing Safely Remove &
Initialize FreeRTOS+FAT command
FreeRTOS+FAT Open successful
 Connect SD Card...

SD Card Menu options
1. Write 10k data from app_buffer to rza_sdhi.txt file
2. Format SD Card Device
3. Safely Remove the SD Card
4. Display content of Current Working Directory
5. Initialize FreeRTOS+FAT (Valid only after Safely_Eject option is executed)
```

6.19 GTM

6.19.1 Project Overview

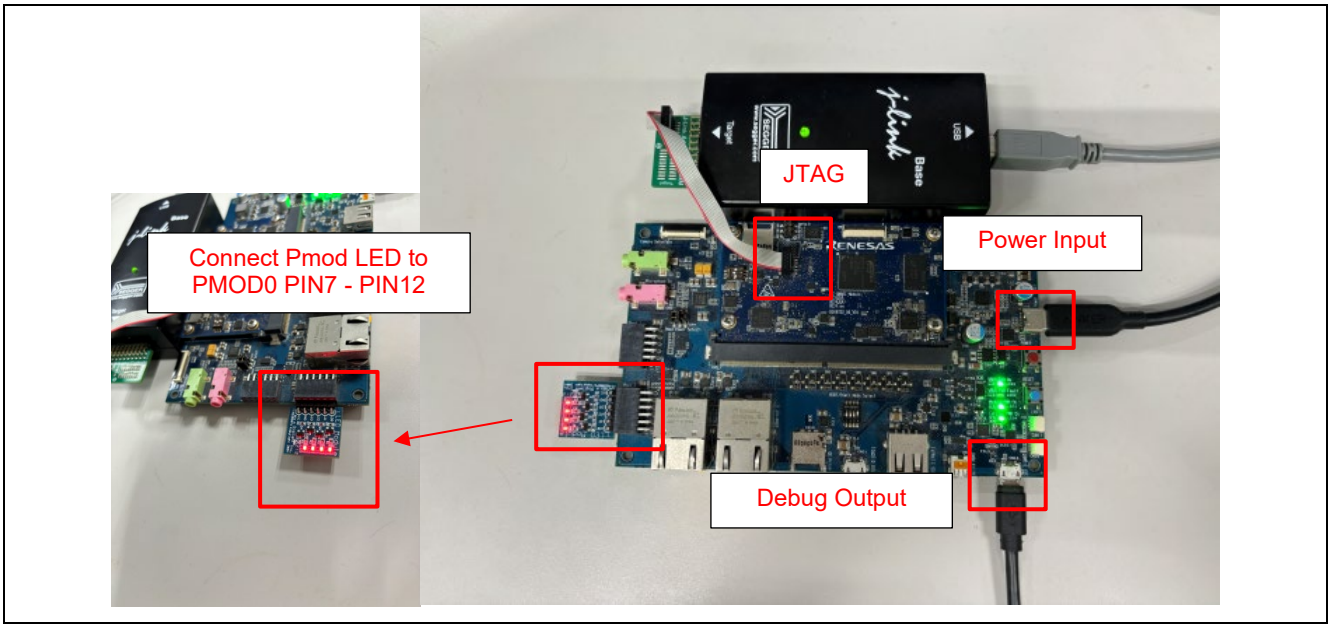
The example project demonstrates the function of GTM driver.
The project initializes the periodic Timer and one-shot Timer. (UART used to communicate.)
Using a Terminal Emulator, the user can provide a value & press enter key to set the period of the Pmod LED signal. The range of input values are displayed on the Terminal Emulator. Any failure will also be displayed using Terminal Emulator.

6.19.2 Hardware Requirements

External hardware: Pmod LED

6.19.3 Hardware Settings

Please connect each hardware as below.



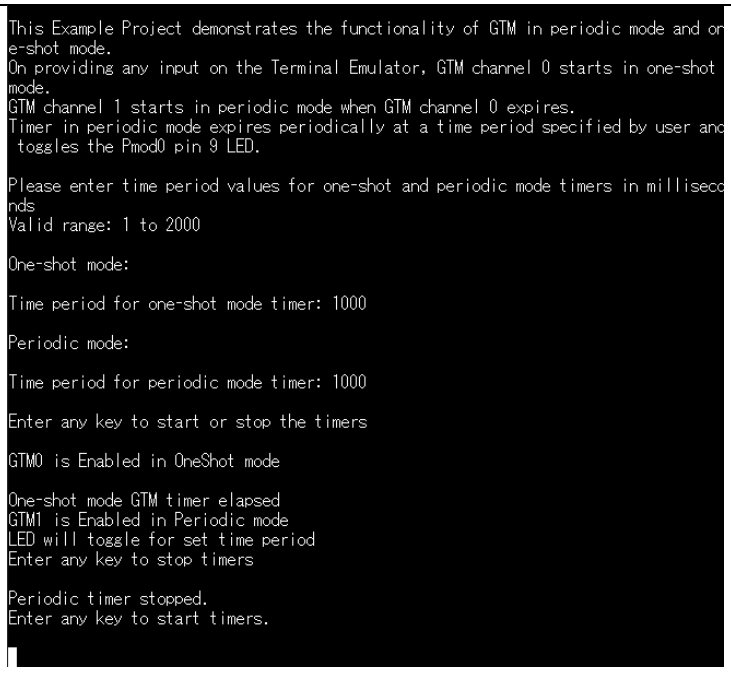
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.19.4 Operation

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



6.20 MTU3a

6.20.1 Project Overview

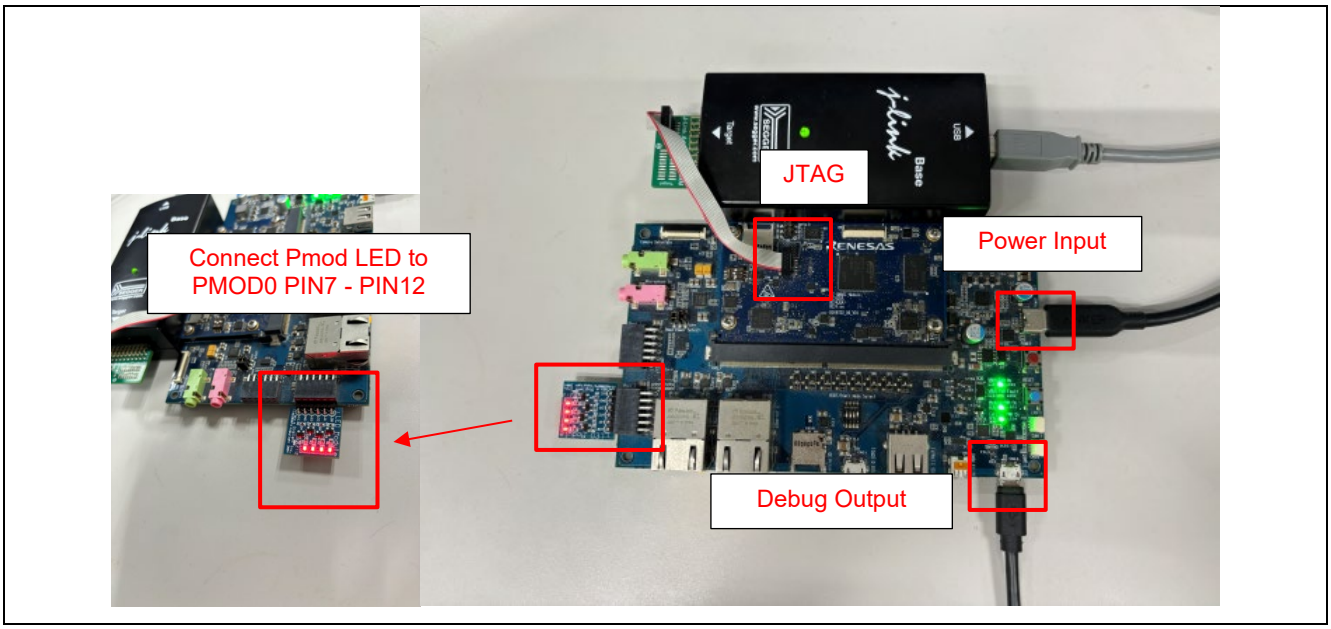
The example project demonstrates typical use of MTU3 HAL module APIs.
User has the provision to input value as per displayed menu through a terminal application to select different MTU3 supported modes (Periodic, PWM, One-Shot).
In periodic mode, the user can enter the time period within the permitted 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.

6.20.2 Hardware Requirements

External hardware: Pmod LED

6.20.3 Hardware Settings

Please connect each hardware as below.



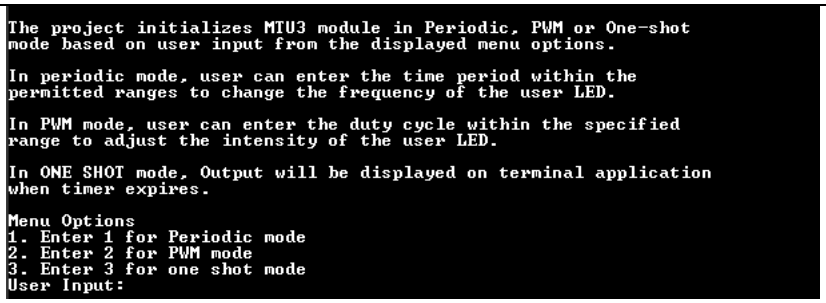
Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.20.4 Operation

After running the example, the message below is displayed on console. Users can select each option.



6.21 DMAC

6.21.1 Project Overview

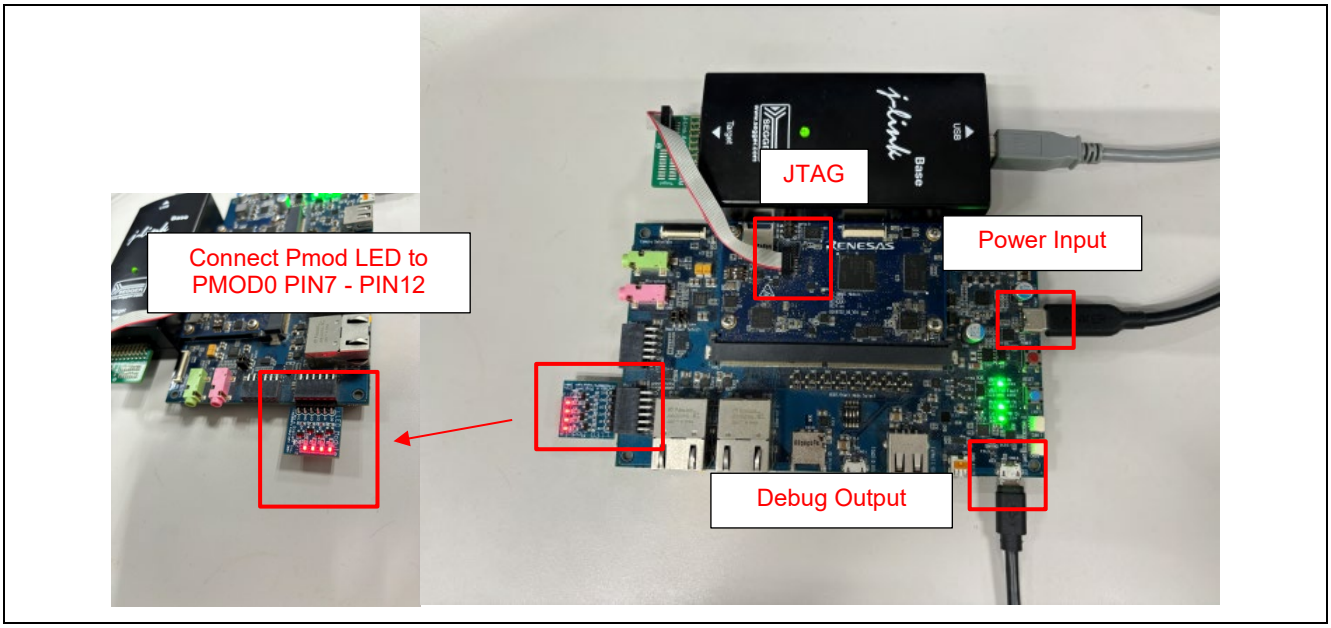
Project initializes 2 DMAC transfer instances. Upon successful initialization, a menu option is displayed. On selecting 1 from menu option, GTM generates 100ms interrupt to trigger a transfer from the source buffer to port control register for 60 times indicated by LED blinking for 60 times. On selecting 2 from menu ontion, DMAC (runs in Block mode) transfers data from the MTU counter register to destination which gets printed to Terminal Emulator.

6.21.2 Hardware Requirements

External hardware: PMOD LED

6.21.3 Hardware Settings

Please connect each hardware as below.



Board Setting

Please set each DIP switch and jumper as below.

Board	RZ/A3UL EVK
Module board	SW1-1: OFF SW1-2: Don’t care. SW1-3: 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

6.21.4 Operation

After running the example, the message below is displayed on console. Users can select each option.

```
Project initializes 2 DMAC transfer instances.Upon successful
initialization,a menu option is displayed. On selecting 1 from menu
option,GTM generates 100ms interrupt to trigger a transfer from the
source buffer to port control register for 60 times indicated by LED
blinking for 60 times. On selecting 2 from menu option,DMAC (runs in
Block mode) transfers data from the MTU counter register to destination
which gets printed to Terminal Emulator.


Press 1 to transfer source array to LED Port in normal mode.
Press 2 to transfer MTU current timer value.
```

7. References

FSP GitHub:	github.com/renesas/rza-fsp
FSP User Manual:	renesas.github.io/rza-fsp/
Getting Started Guide	Getting Started with RZ/A Flexible Software Package V2.01 (renesas.com)
FSP Example Projects:	RZ/A3UL Software Package | Renesas
Evaluation Kit Manuals:	RZ/A3UL-Evaluation-Board-Kit (renesas.com)
Knowledge Base:	Knowledge Base (renesas.com)
Renesas Support:	RZ/A3UL - Support (renesas.com)

Revision History

Rev.	Date	Description	
		Page	Summary
3.00	May.31.24	6	Updated the operating environment.
		16 to 59	Added the description of all examples.
1.10	Oct.27.23	3	Updated the version of FSP and e2 studio supported.
		4-6	Updated the section of running project.
1.01	Dec.9 22	–	Added a setting procedure to change the project to RZ/A3UL Evaluation Board Kit OCTAL Edition.
1.00	Nov.8.22	–	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/.

© 2023 Renesas Electronics Corporation. All rights reserved.