Getting started with the RL78/G14 Fast Prototyping Board / Wi-Fi-Pmod-Expansion-Board

This tutorial provides instructions for getting started with the Renesas RL78/G14 Fast Prototyping Board / Wi-Fi-Pmod-Expansion-Board. If you do not have the RL78/G14 Fast Prototyping Board / Wi-Fi-Pmod-Expansion-Board, visit the AWS Partner Device Catalog, and purchase one from our partners.

This document show user how to configure AWS IoT Core and FreeRTOS to connect your device to the AWS Cloud.

Overview

This tutorial contains instructions for the following getting started steps:

- **A Purchase RL78/G14 Fast Prototyping Board and Wi-Fi-Pmod-Expansion-Board.**
- **B Installing tool and software on the host machine for developing.**
- **C Creating Policy for Device**
- **D Device on AWS IoT Core**
- **E The steps to write the private key and certificate into Wi-Fi-Pmod-Expansion-Board.**
- **F Set up the RL78/G14 Fast Prototyping Board / Wi-Fi-Pmod-Expansion-Board.**
- **G Cross compiling a FreeRTOS demo application to a binary image.**
- **H Loading the application binary image to your board, and then running the application.**
- **I Monitoring MQTT messages on the cloud.**

A. Purchase RL78/G14 Fast Prototyping Board / Wi-Fi-Pmod-Expansion-Board

The evaluation kit compose of RL78/G14 Fast Prototyping Board and Wi-Fi-Pmod-Expansion-Board. Also, need Digilent Pmod USBUART to connect Wi-Fi module to PC to set up credentials, or to connect RL78/G14 for logging/debugging info.

1. RL78/G14 Fast Prototyping Board

https://www.renesas.com/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus/rl78g14-fast-prototyping-board-rl78g14-fast-prototyping-board

2. Wi-Fi-Pmod-Expansion-Board

https://www.renesas.com/products/microcontrollers-microprocessors/ra-cortex-m-mcus/wi-fi-pmod-expansion-board-80211bgn-24g-wi-fi-pmod-expansion-board

Wi-Fi-Pmod-Expansion-Board implements Silex SX-ULPGN wifi module on board.

https://www.silextechnology.com/connectivity-solutions/embedded-wireless/sx-ulpgn

3. Digilent Pmod USBUART

https://reference.digilentinc.com/reference/pmod/pmodusbuart/start

4. USB cables x2

5. Generic cables to connect between the Digilent Pmod USBUART and Wi-Fi-Pmod-Expansion-Board x3.

(These cables can be used to connect the Digilent Pmod USBUART to the RL78/G14 Fast Prototyping Board)

B. Installing software and tool on the host machine for developing

Note: Host machine running Windows 7, 8 or 10.

To download and install e²studio

1. Go to the Renesas e²studio installer download page and download the offline installer.
2. You are directed to a Renesas Login page.

   If you have an account with Renesas, enter your username and password and then choose **Login**.

   If you do not have an account, choose **Register now**, and follow the first registration steps. You should receive an email with a link to activate your Renesas account. Follow this link to complete your registration with Renesas, and then login to Renesas.
3. After you log in, download the e²studio installer to your computer.
4. Open the installer and follow the steps to completion.

For more information, see the e²studio on the Renesas website.

To download and install the RL78 Family C Compiler Package

1. Go to the RL78 Family C Compiler Package download page, and download the v1.09.00 package.
2. Open the executable and install the compiler.

For more information, see the C Compiler Package for RL78 Family on the Renesas website.

> **Note**
>
> The compiler is available free for evaluation version only and valid for 60 days. On the 61st day, you need to get a License Key. For more information, see Evaluation Software Tools.

To download SharkSSL

Following free software program to convert certificate data to the required format. Go to https://realtimelogic.com/downloads/sharkssl/ to download the software.

To download Tera Term

Write the converted certificate and CA list (binary files) to the Wi-Fi-Pmod-Expansion-Board. Go to

https://ttssh2.osdn.jp/index.html.en to download the software.

C. Create a Policy for a Device

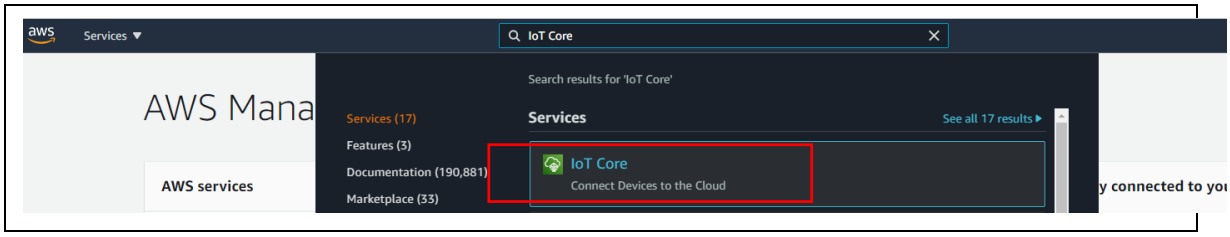
User needs to create AWS account. Refer to the instructions at Set up your AWS Account. Follow the steps outlined in these sections to create your account and a user and get started:

- Sign up for an AWS account.
- Create a user and grant permissions.
- Open the AWS IoT console.

Pay special attention to the Notes.

If user created account already in the past, please skip this step.

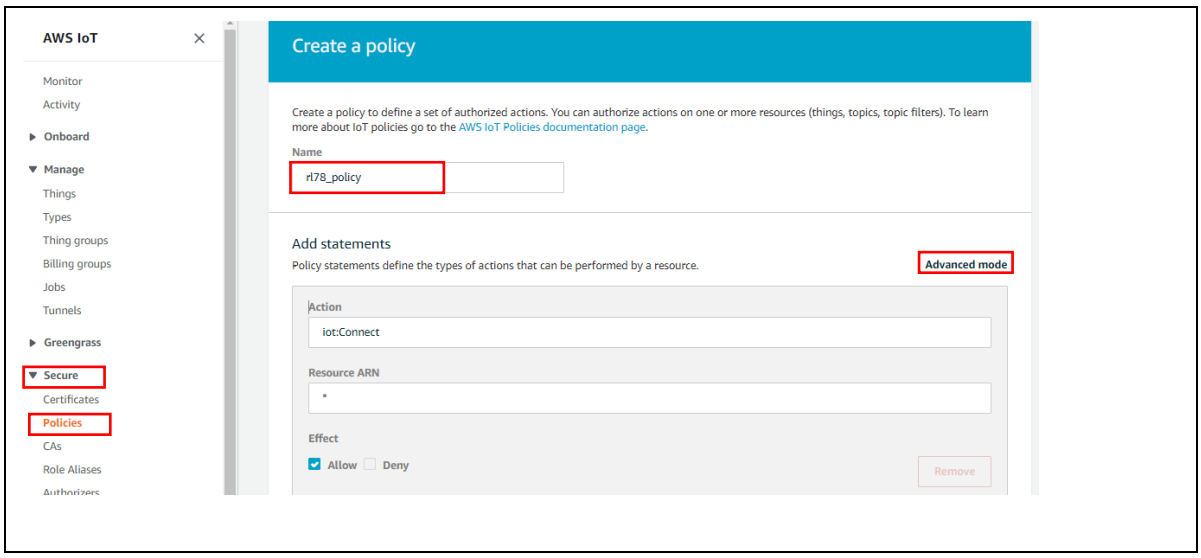
1. Typing IoT Core in search bar and click **IoT Core**



AWS IoT Core Selection

2. Go to Secure→Policies

   Add **policy name** and change to **Advanced mode**



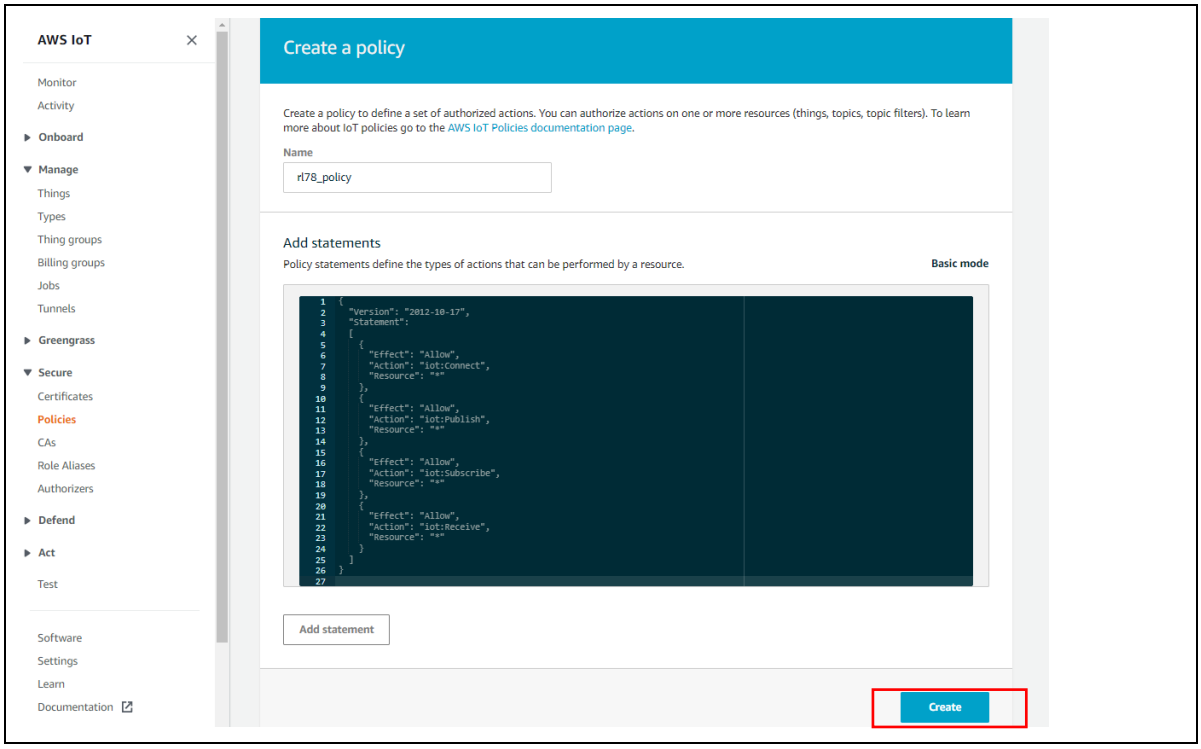
Give a policy name

Add following text to **Advanced mode**

```
{
        "Version": "2012-10-17",
        "Statement":
        [
                {
                        "Effect": "Allow",
                        "Action": "iot:Connect",
                        "Resource": "*"
                },
                {
                        "Effect": "Allow",
                        "Action": "iot:Publish",
                        "Resource": "*"
                },
                {
                        "Effect": "Allow",
                        "Action": "iot:Subscribe",
                        "Resource": "*"
                },
                {
                        "Effect": "Allow",
                        "Action": "iot:Receive",
                        "Resource": "*"
                }
        ]
}
```

Add statements for policy

3. Create a policy



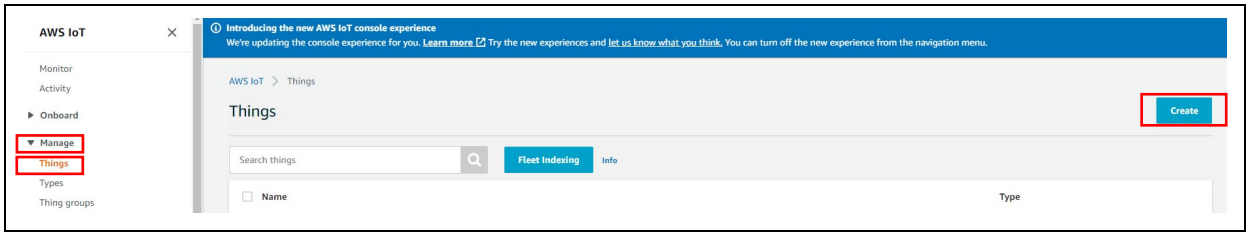
Create a policy

Note: The examples in this document are intended only for dev environments. All devices in your fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to Example policies and Security Best practices.

D. Creating Device on AWS IoT Core

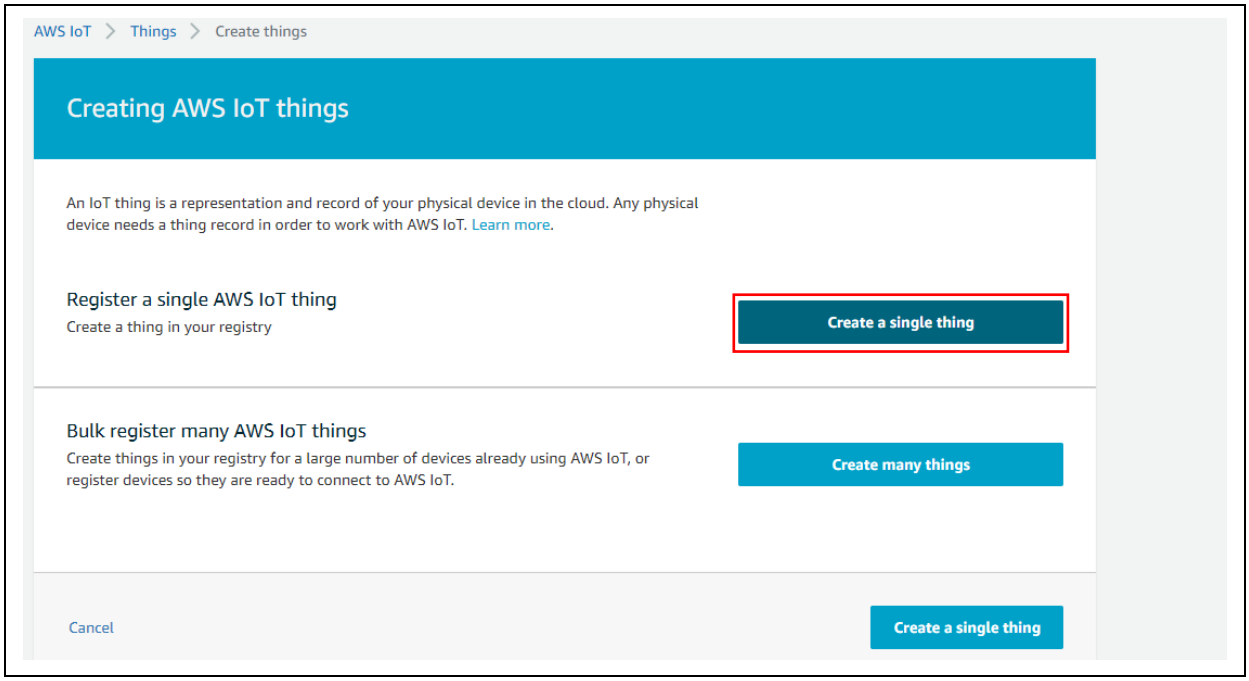
4. **Create a Thing**

Select **Manager→ Thing→Create** to create a thing



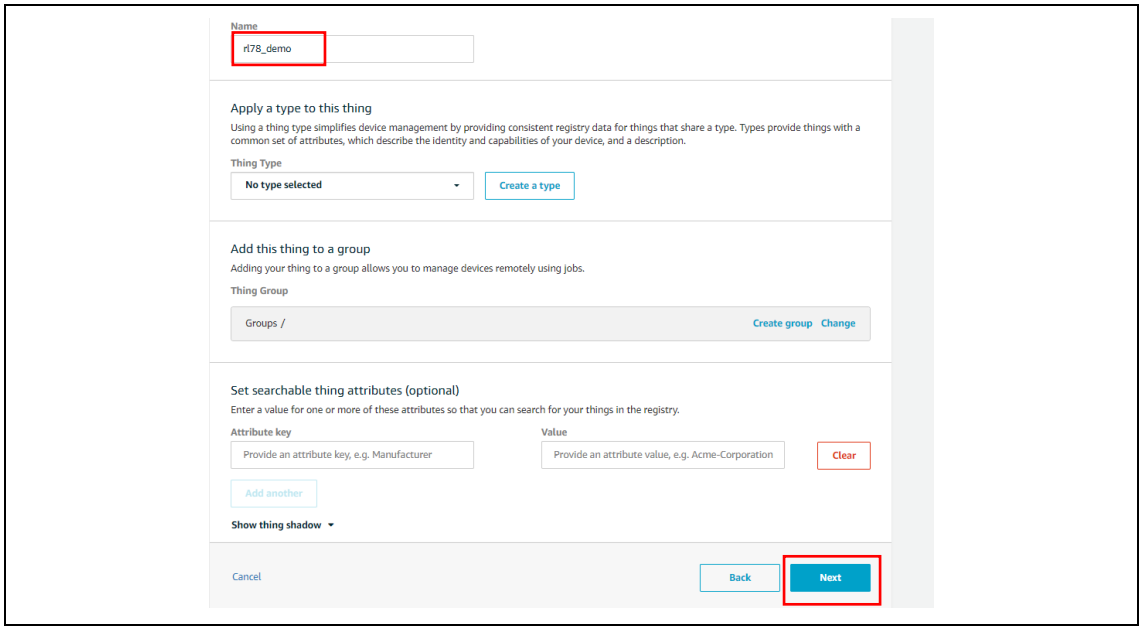
Create a thing

5. Select the **Create a single thing**



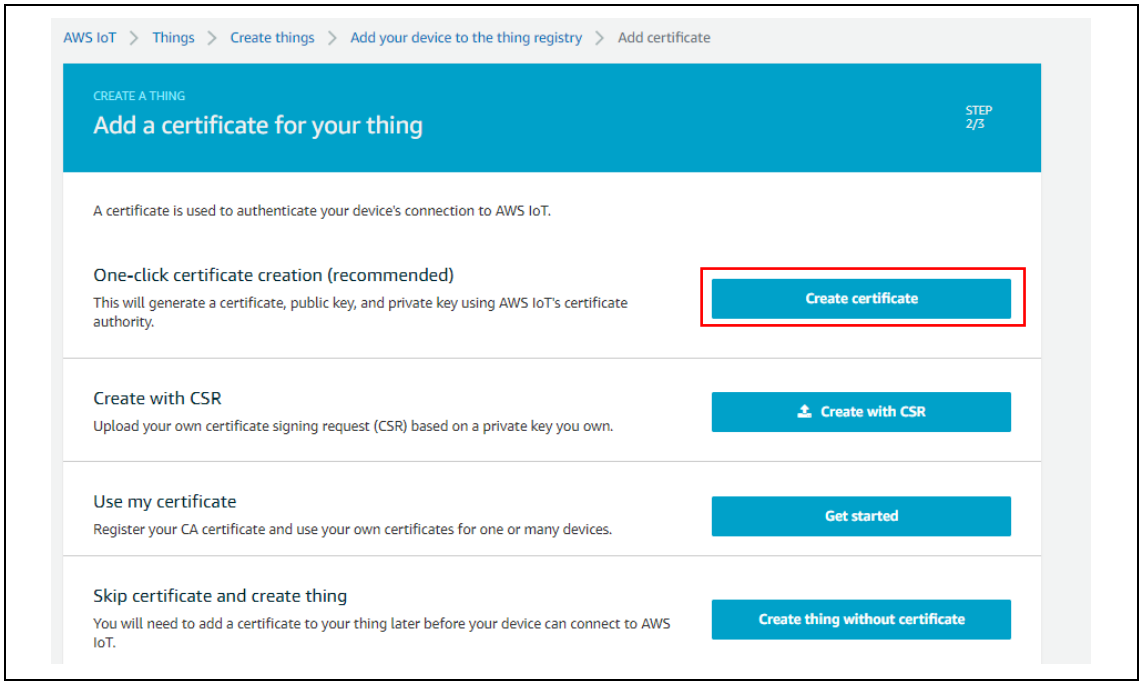
Create a single thing

6. Add name to thing and **Next**



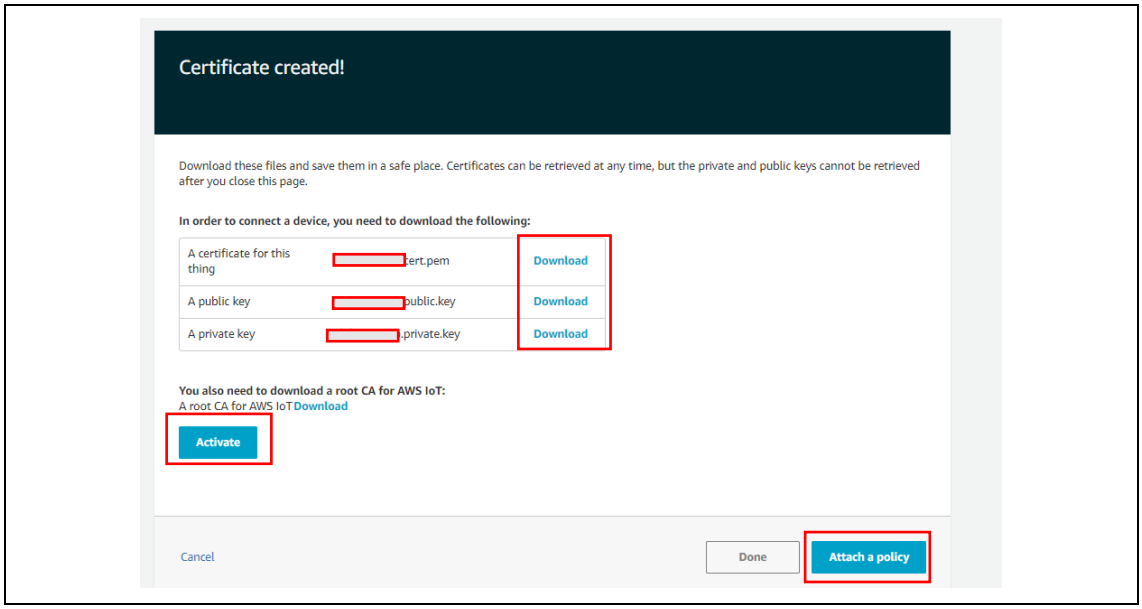
Add name to a single thing

7. Add a certificate for thing



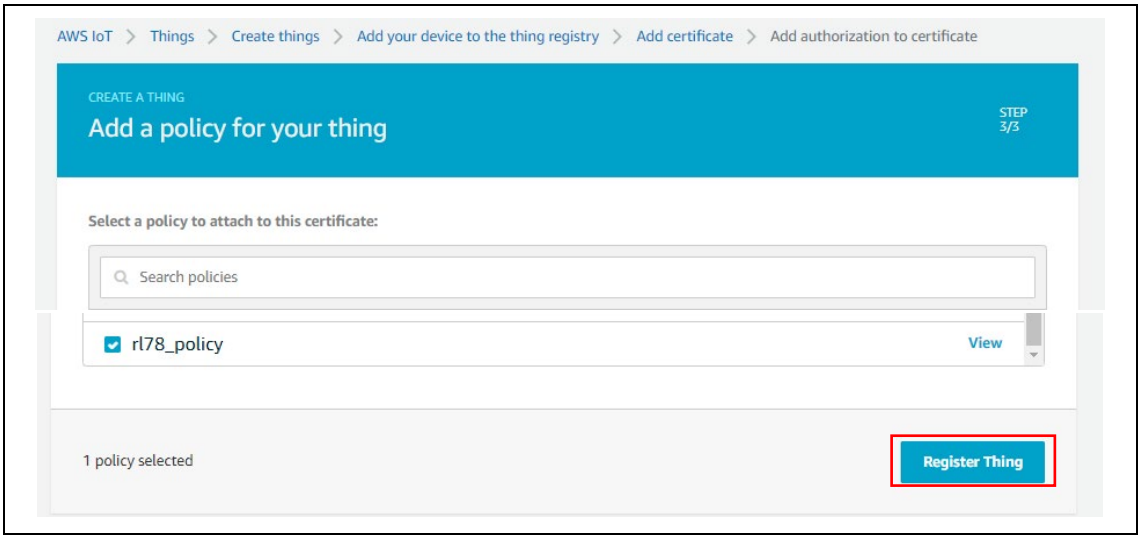
Create a certificate for thing

8. Attach a policy to thing
   - Click the **Download** button next to each of the certificates, keys and save in local PC or host machine.
   - Click the **Activate** button to activate the certificate.
   - Select **Attach a policy**



Attach a policy

9. Register policy to thing



Register policy to thing

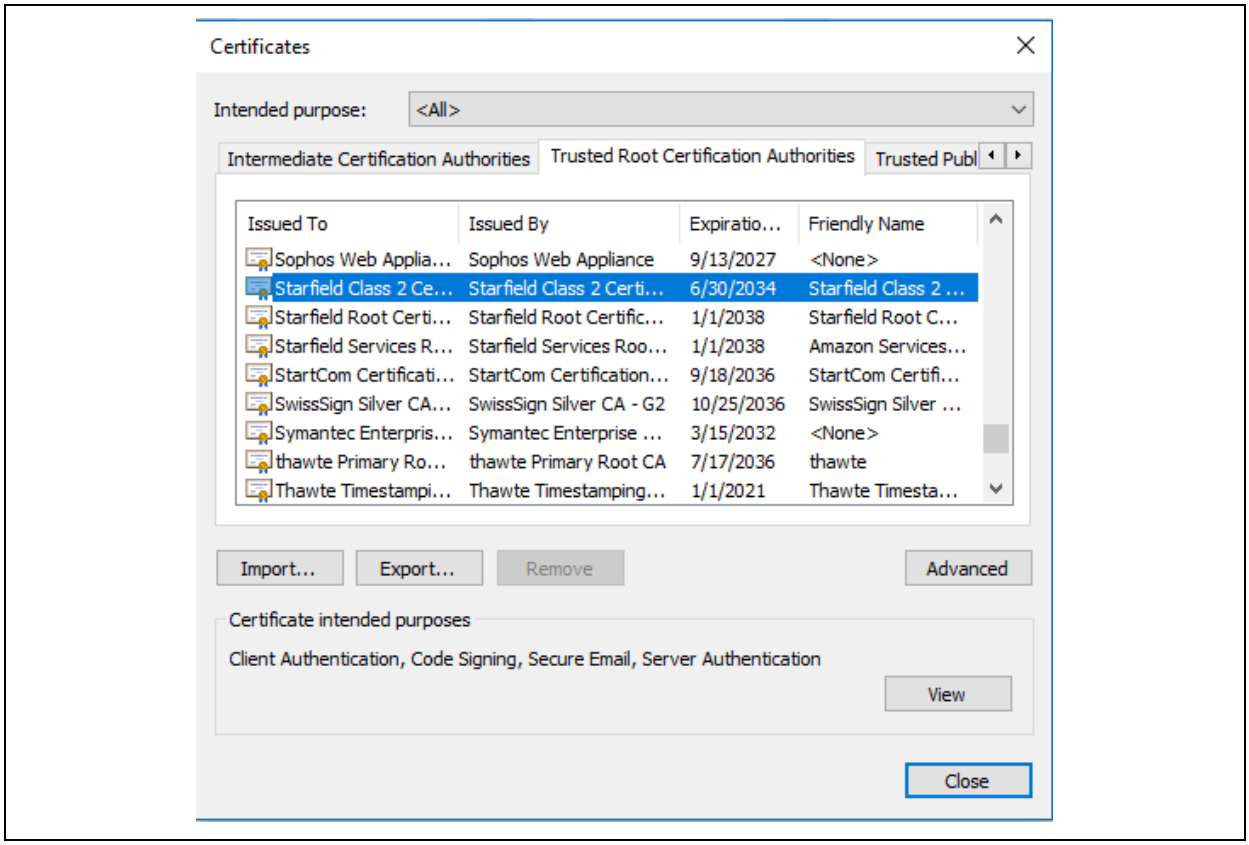
E. The steps to write the private key and certificate into Wi-Fi-Pmod-Expansion-Board

To write the private key and certificate into Wi-Fi-Pmod-Expansion-Board

We need to write the certificate and private key to Wi-Fi module to connect to AWS servers. The TCP/IP and SSL/TLS with mutual authentication and secure storage are offloaded to the Wi-Fi module. This architecture allows to build secure connected Internet of Things (IoT) devices using small MCU like RL78.

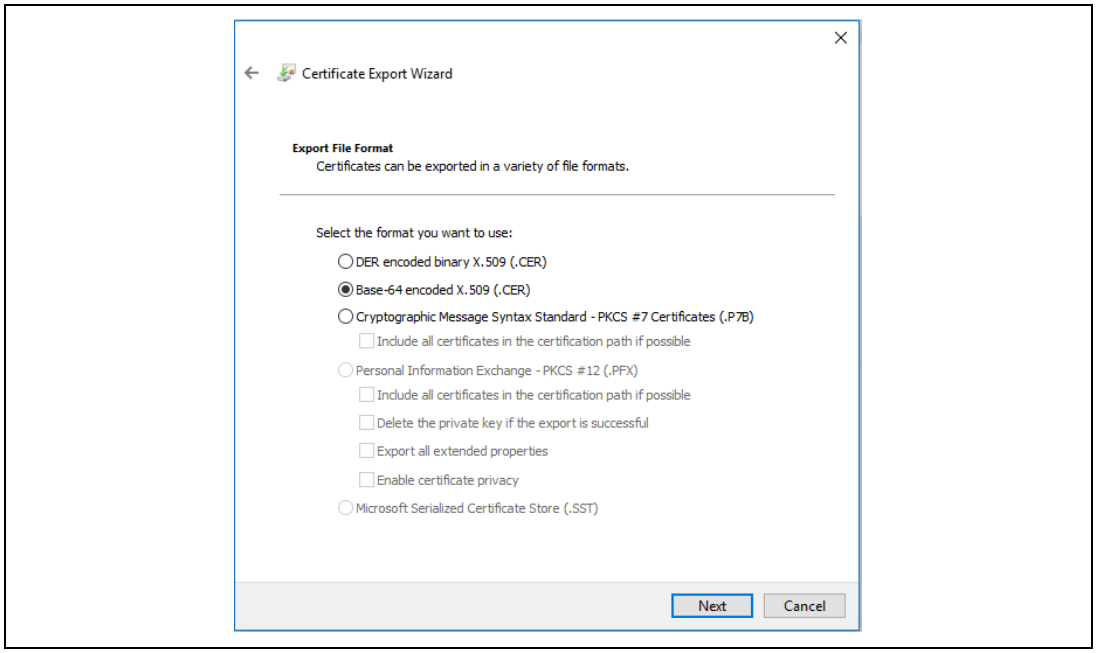
1. **Obtaining a CA List (Class 2 Root CA)**
   - In Internet Explorer, select Tool tab → Internet Options → Content → Certificates → Trusted Root Certification Authorities, then export Starfield Class 2 Certification Authority.

     Note: Class 2 Root CA is created by Internet Explorer. We have not tested with other browsers or method yet.



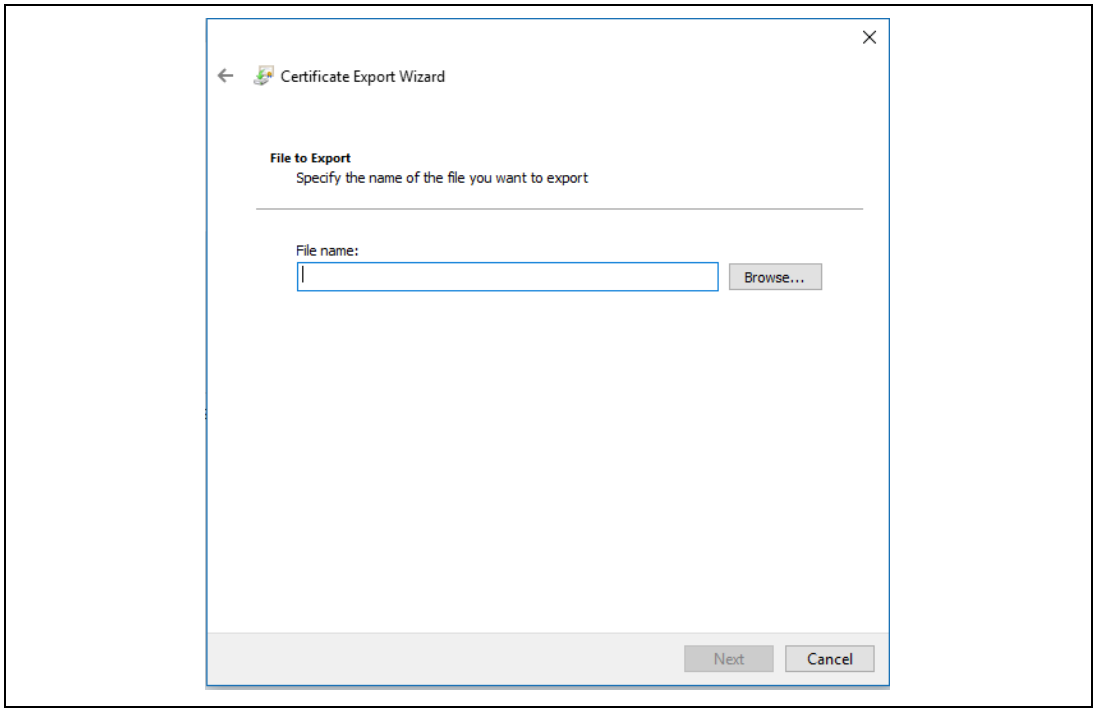
Export Starfield Class 2 Certification Authority

- Select Base 64 encoded X.509 (.CER).



Select Base 64 encoded X.509 (.CER)

- Enter a file name of your choice, and export the certificate



Enter a file name

2. **Convert the Certificate and Secret Key to SharkSSL Binary Format**

   Run the following command from the command prompt to convert the certificate and secret key to SharkSSL binary format.

   SharkSSLParseCert xxxxx-certificate.pem.crt xxxxx-private.pem.key –b xxxxx-certificate.bin

   xxxxx represents the file name

3. **Convert the CA List to SharkSSLPerseCAList Binary Format**

   Run the following command from the command prompt to convert the CA list to SharkSSLPerseCAList binary format.

   SharkSSLParseCAList.exe -b yyyyy.bin zzzzz.cer

   yyyyy, zzzzz represents the file name.

4. **Connect Wi-Fi module to PC to write the Certificate to the Wi-Fi-Pmod-Expansion-Board**

   Write the converted certificate and CA list (binary files) to the Wi-Fi-Pmod-Expansion-Board. Connect the PC to the TX and RX pins of the Wi-Fi-Pmod-Expansion-Board via a USB-to-serial converter and use AT commands to write the data. Use a baud rate of 115,200 bps.

   As an example, settings for writing the certificate and CA list using a terminal emulator (Tera Term) are given below. Make sure to use version 4.105 or later of Tera Term.

   [Serial port settings in Setup tab]

   Baud rate: 115,200 bps

   Data: 8 bits

   Parity: none

   Stop: 1 bit

   Flow control: none

[Terminal settings in Setup tab]

New line code

Receive: CR

Transmit: CR

Local echo: Unchecked

As an example, connections between the Digilent Pmod USBUART and Wi-Fi-Pmod-Expansion-Board are shown below. The connector on the Wi-Fi-Pmod-Expansion-Board has two rows. Connect wires from the Digilent Pmod USBUART to the **top-row connectors** on the Wi-Fi-Pmod-Expansion-Board.

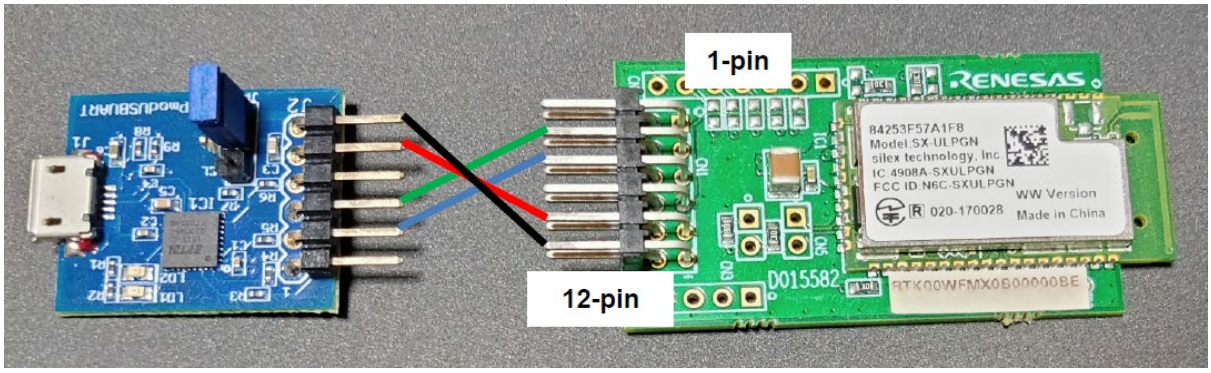
On the Digilent Pmod USBUART, use a jumper to short VCC and SYS (power supply from Digilent Pmod USBUART to Wi-Fi-Pmod-Expansion-Board).

Connect pin 2 (RxD) on Digilent Pmod USBUART to pin 3 (TxD) on Wi-Fi-Pmod-Expansion-Board.

Connect pin 3 (TxD) on Digilent Pmod USBUART to pin 2 (RxD) on Wi-Fi-Pmod-Expansion-Board.

Connect pin 5 (GND) on Digilent Pmod USBUART to pin 5 (GND) on Wi-Fi-Pmod-Expansion-Board.

Connect pin 6 (VCC) on Digilent Pmod USBUART to pin 6 (VCC) on Wi-Fi-Pmod-Expansion-Board.



Connect between the Digilent Pmod USBUART and Wi-Fi-Pmod-Expansion-Board

5. Write the Certificate to the Wi-Fi-Pmod-Expansion-Board

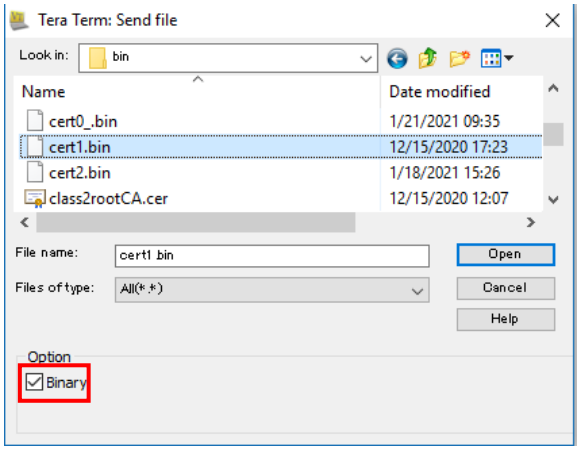
- Run the following command to write certificate and private key.

  ATNSSLCERT=cert1.crt,< binary file size of converted certificate >

  Example: ATNSSLCERT=cert1.crt,1768

  Within 30 seconds, send the binary file converted as described in “Converting the Certificate and Secret Key to SharkSSL Binary Format” by file transfer from Tera Term.

  Note: Make sure that Binary is checked under Option.



Transfer .bin file to Wi-Fi-Pmod-Expansion-Board

- Run the following command to write **CA List (Class 2 Root CA)**

  ATNSSLCERT= calist1.crt,< binary file size of converted CA list >

  Example: ATNSSLCERT=calist1.crt,1059

  Within 30 seconds, send the binary file converted as described in “Converting the CA List to SharkSSLPerseCAList Binary Format” by file transfer from Tera Term.

  Note: Make sure that Binary is checked under Option.

  ➢ To confirm certificate/private key and CA List are written properly.

  Run the ATNSSLCERT=? command, and confirm that the following lines are displayed.

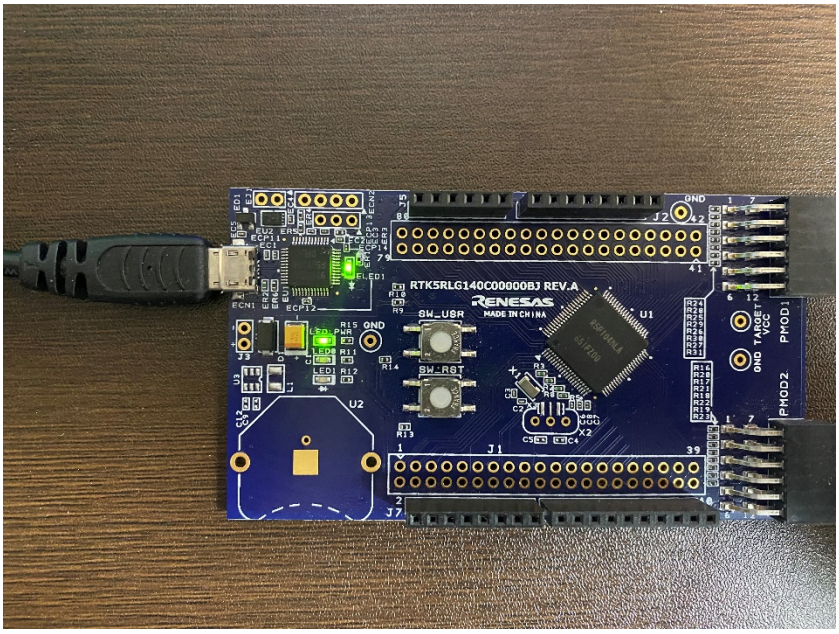
  calist1.crt

  cert1.crt

F. Set up the RL78/G14 Fast Prototyping Board / Wi-Fi-Pmod-Expansion-Board

To confirm functionality on RL78/G14 Fast Prototyping Board and E2 Lite Debugger module on board

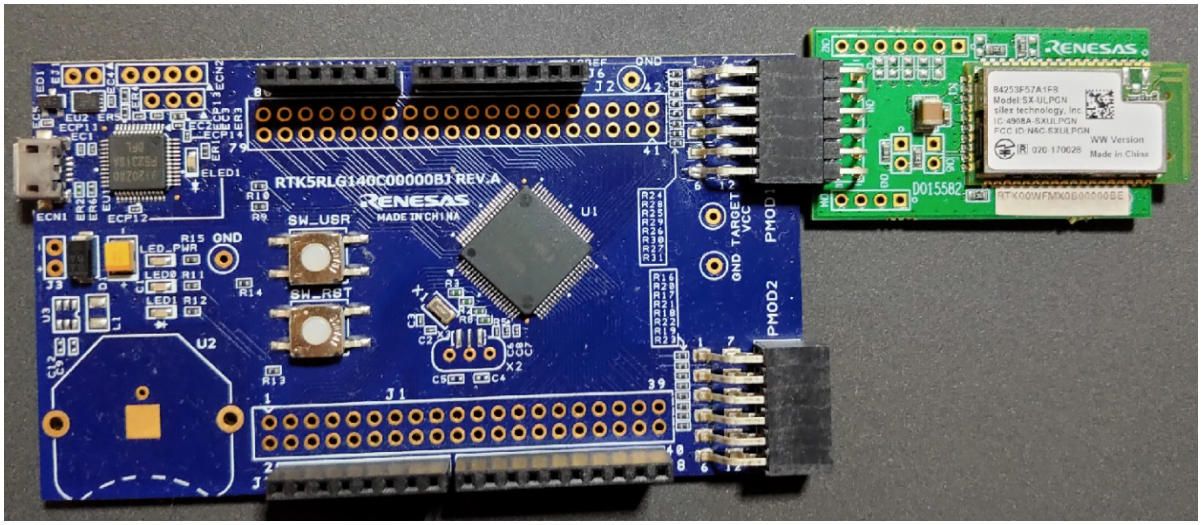
Connect ECN1 (USB Micro-B) on RL78/G14 Fast Prototyping Board to power source USB port (PC,etc).



Connect RL78/G14 Fast Prototyping Board to power PC

To connect the Wi-Fi-Pmod-Expansion-Board

Connect the Wi-Fi-Pmod-Expansion-Board to the RL78/G14 Fast Prototyping Board. The Wi-Fi-Pmod-Expansion-Board connects to PMOD1.



Connect the Wi-Fi-Pmod-Expansion-Board to the RL78/G14 Fast Prototyping Board

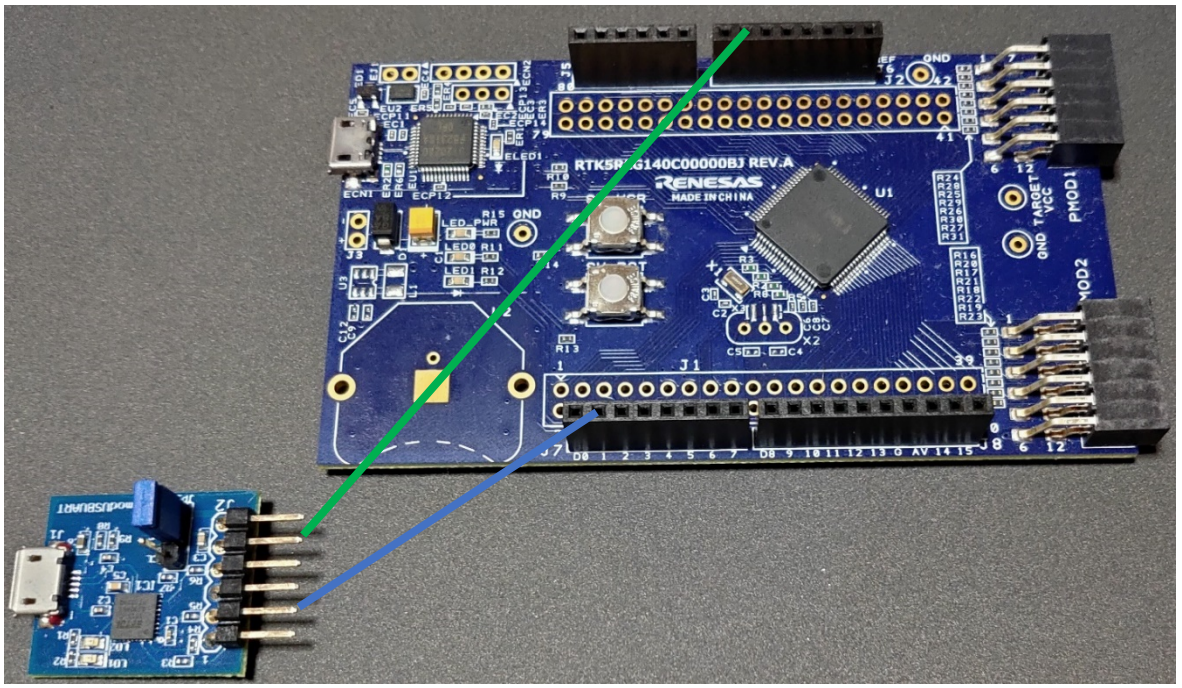
To receive Debug Logs

The demo outputs debug logs via the SCI port. To check the debug logs, use a terminal emulator (Tera Term, etc.) to connect to the serial port used by the SCI driver. As an example, connection of the Digilent Pmod USBUART and RL78/G14 Fast Prototyping Board is shown below.

Connect pin 2 (RxD) on Digilent Pmod USBUART to pin 2 (TxD) on RL78/G14 Fast Prototyping Board.

Connect pin 3 (GND) on Digilent Pmod USBUART to pin 3 (GND) on RL78/G14 Fast Prototyping Board.

Power is supplied from the PC to the RL78/G14 Fast Prototyping Board via a USB cable, so there is no need to supply power from the Digilent Pmod USBUART. In addition, it is not necessary to send data from the Digilent Pmod USBUART because debug logs are only received, not sent.



Connect the Digilent Pmod USBUART to the RL78/G14 Fast Prototyping Board

G. Cross compiling a FreeRTOS demo application to a binary image

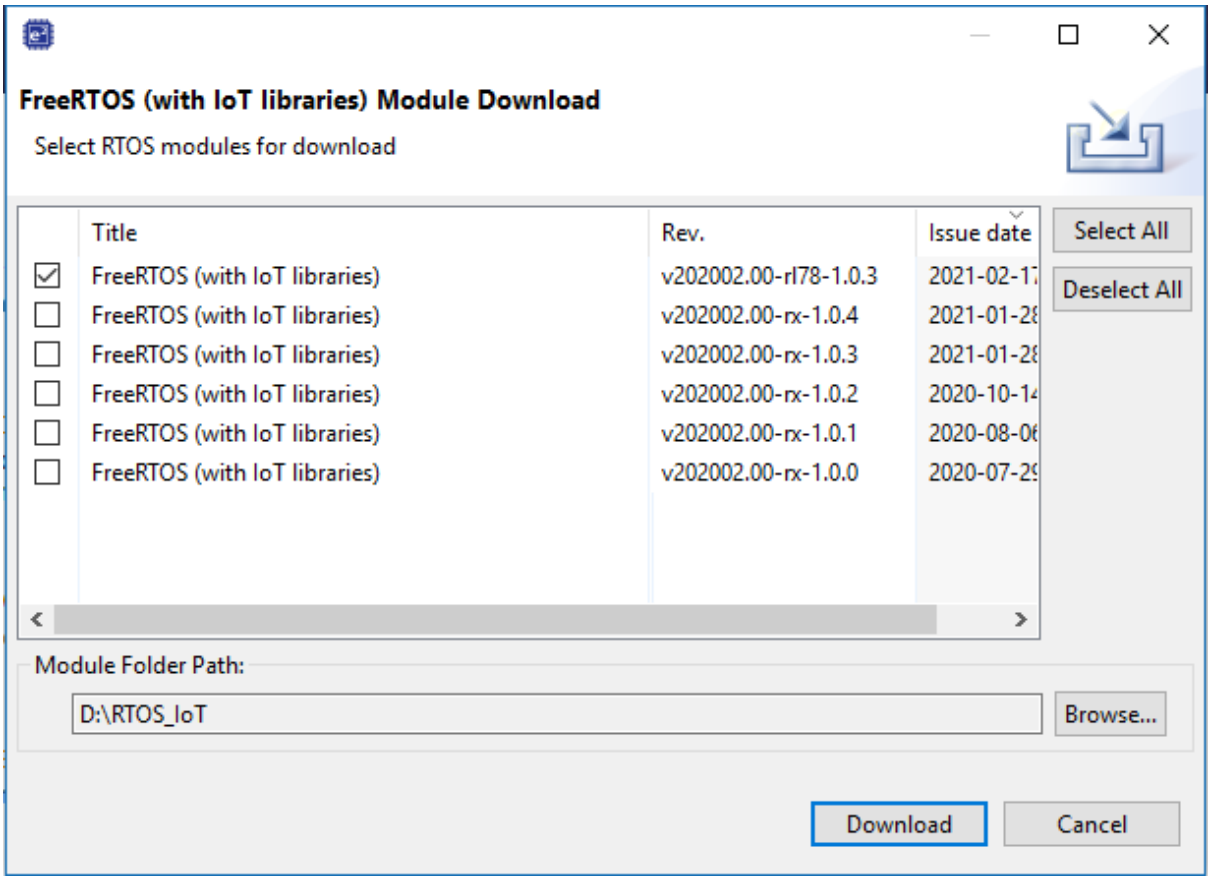
Now that you have configured the demo project, you are ready to build and run the project on your board.

Build the FreeRTOS Demo in e²studio

To download and build the demo in e²studio

1. Launch e²studio from the Start menu.
2. On the **Select a directory as a workspace** window, browse to the folder that you want to work in, and choose **Launch**.
3. The first time you open e2studio, the **Toolchain Registry** window opens. Choose **Renesas Toolchains** and confirm that `CC-RL v1.09.00` is selected. Choose **Register**, and then choose **OK**.
4. If you are opening e²studio for the first time, the **Code Generator Registration** window appears. Choose **OK**.
5. The **Code Generator COM component register** window appears. Under **Please restart e²studio to use Code Generator**, choose **OK**.
6. The **Restart e²studio** window appears. Choose **OK**.
7. e²studio restarts. On the **Select a directory as a workspace** window, choose **Launch**.
8. On the e²studio welcome screen, choose the **Go to the e²studio workbench** arrow icon.
9. Right-click the **Project Explorer** window and choose **Import**.

10. In the import wizard, choose **General**, **Renesas GitHub FreeRTOS (with IoT libraries) Project**, and the choose **Next**.
11. Choose **Browse** to specify a folder to copy downloaded RTOS content in order to import project.
12. In RTOS version setting, choose **Check for more version…** to see a list of all supported RTOS version. On the **FreeRTOS (with IoT libraries) Module Download** window, select the FreeRTOS version (recommended: v202002.00-rl78-1.0.3) you want to work on by clicking the checkbox, then choose **Download**.



13. Once download is completed, choose **Next** in the **Renesas GitHub FreeRTOS (with IoT libraries) Project** window.
14. If you are *not* using an empty folder, the **Copy Resources** warning message appears. Choose **Yes**.
15. Choose the project aws_demos (`${FOLDER_DIR}/projects/renesas/rl78g14-fpb-sx-ulpgn/e2studio/aws_demos`), then choose **Finish**.
16. From **Project** menu, choose **Build All**.

The build console issues a warning message that the License Manager is not installed. You can ignore this message unless you have a license key for the CC-RL compiler. To install the License Manger, see the License Manager download page.

H. Loading the application binary image to your board, and then running the application

To run the project in e²studio

1. Confirm that you have connected your computer to the USB-to-serial port on RL78/G14 Fast Prototyping Board.
2. From the top menu, choose **Run**, **Debug Configurations…**.
3. Expand **Renesas GDB Hardware Debugging** and choose **aws_demos HardwareDebug**.
4. Choose the **Debugger** tab, and then choose the **Connection Settings** tab. Confirm that your connection settings are correct.
5. Choose **Debug** to download the code to your board and begin debugging.

   You might be prompted by a firewall warning for `e2-server-gdb.exe`. Check **Private networks, such as my home or work network**, and then choose **Allow access**.
6. e²studio might ask to change to **Renesas Debug Perspective**. Choose **Yes**.

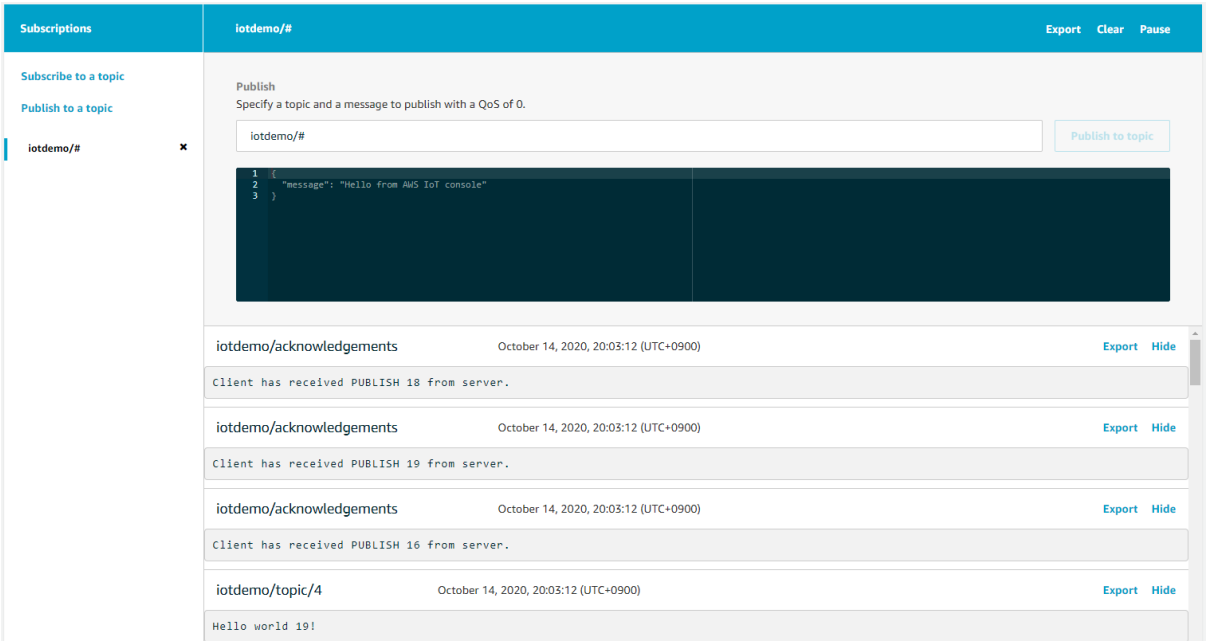
   The flashing green ELED1 on RL78/G14 Fast Prototyping Board illuminates.
7. After the code is downloaded to the board, choose **Resume** to run the code up to the first line of the main function. Choose **Resume** again to run the rest of the code.

I. Monitoring MQTT messages on the cloud

You can use the MQTT client in the AWS IoT console to monitor the messages that your device sends to the AWS Cloud.

To subscribe to the MQTT topic with the AWS IoT MQTT client

1. Sign in to the AWS IoT console.
2. In the navigation pane, choose **Test** to open the MQTT client.
3. In **Subscription topic**, enter `iotdemo/#`, and then choose **Subscribe to topic**.
4. Successful demo run looks like following the picture



For the latest projects released by Renesas, see the `renesas` fork of the `amazon-freertos` repository on GitHub.

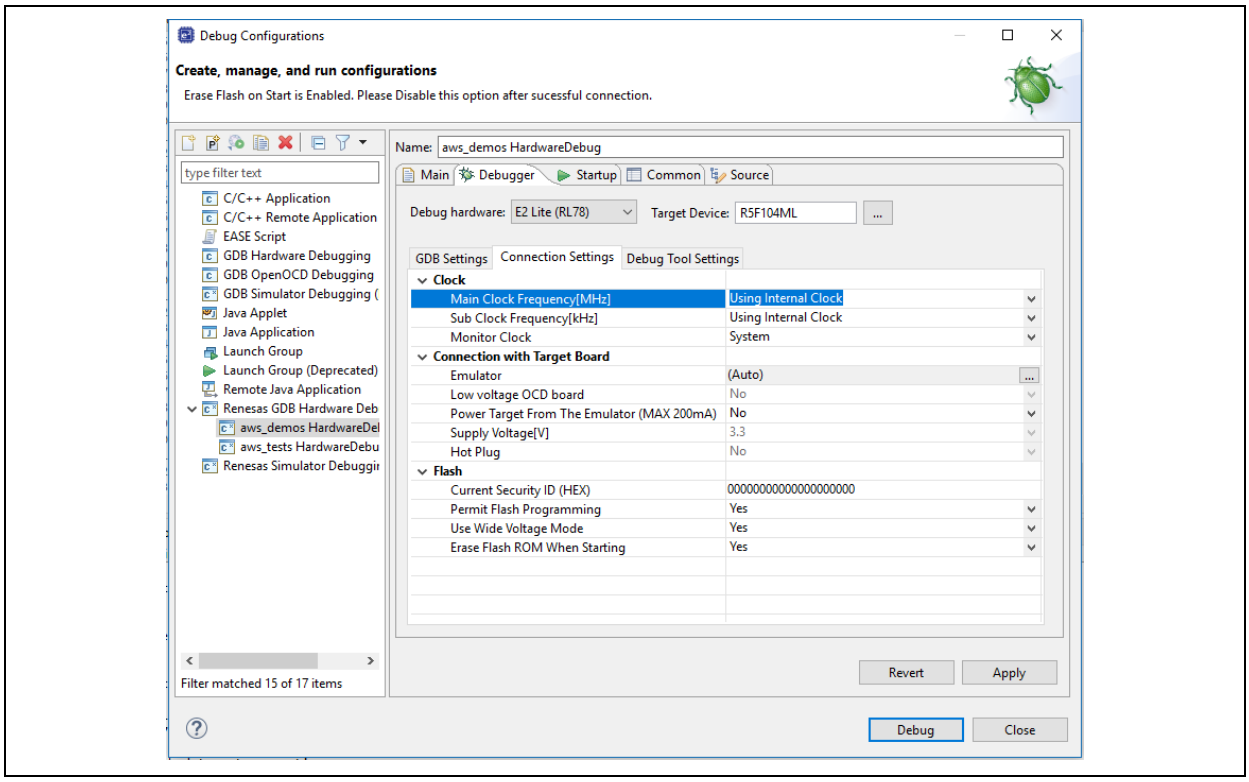
Troubleshooting

For general troubleshooting information about Getting Started with FreeRTOS, see Troubleshooting getting started.

The following information is for debugging if any troubles.

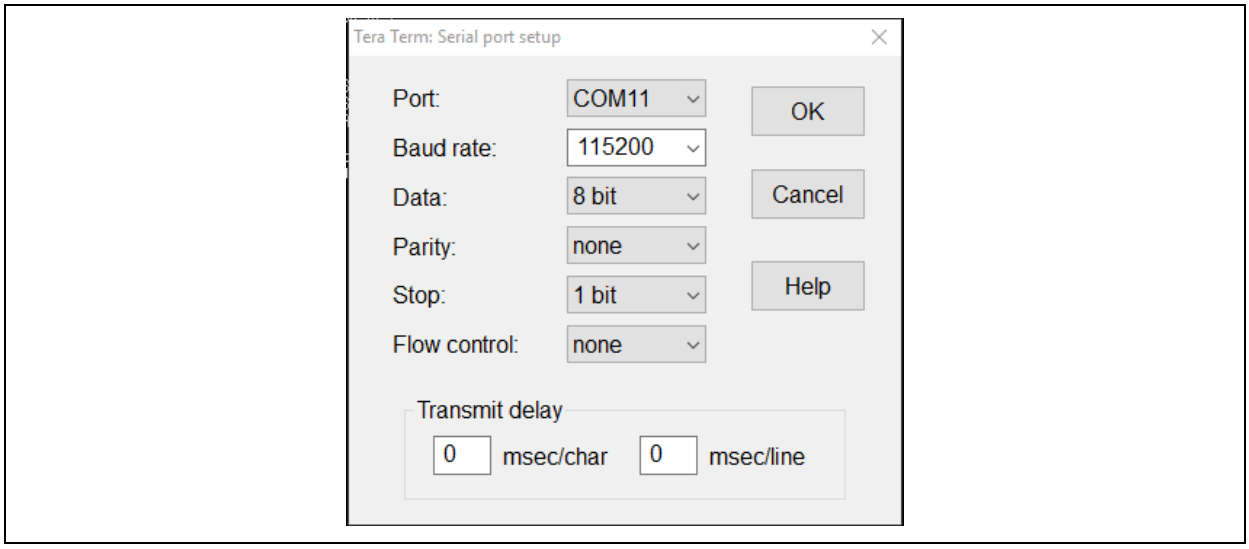
1. **Open e2studio to debug**

   Make sure that debug configuration is same as the following setting.



2. **Tera term**

   Open tera term to check port, baud rate, Data, Parity, Stop and Flow control.



3. The Build errors

- Make sure that v202002.00-rl78-1.0.3 is located to C: or D: drive or etc. Windows has a path length limitation of 260 characters. The path structure of FreeRTOS is many levels deep, so if you are using Windows, keep your file paths under the 260-character limit. The build will be passed if file paths under the 260-character.
- Check `#define configTOTAL_HEAP_SIZE` in FreeRTOSConfig.h if following error occurs

```
amazon-freertos/freertos_kernel/portable/MemMang/heap_4.c(65):E0520095:Array is too large
```

4. Can not connect to AWS IoT Core

- Check aws_demos/demos/include/aws_clientcredential.h and confirm 4 settings:

  clientcredentialMQTT_BROKER_ENDPOINT

  clientcredentialIOT_THING_NAME

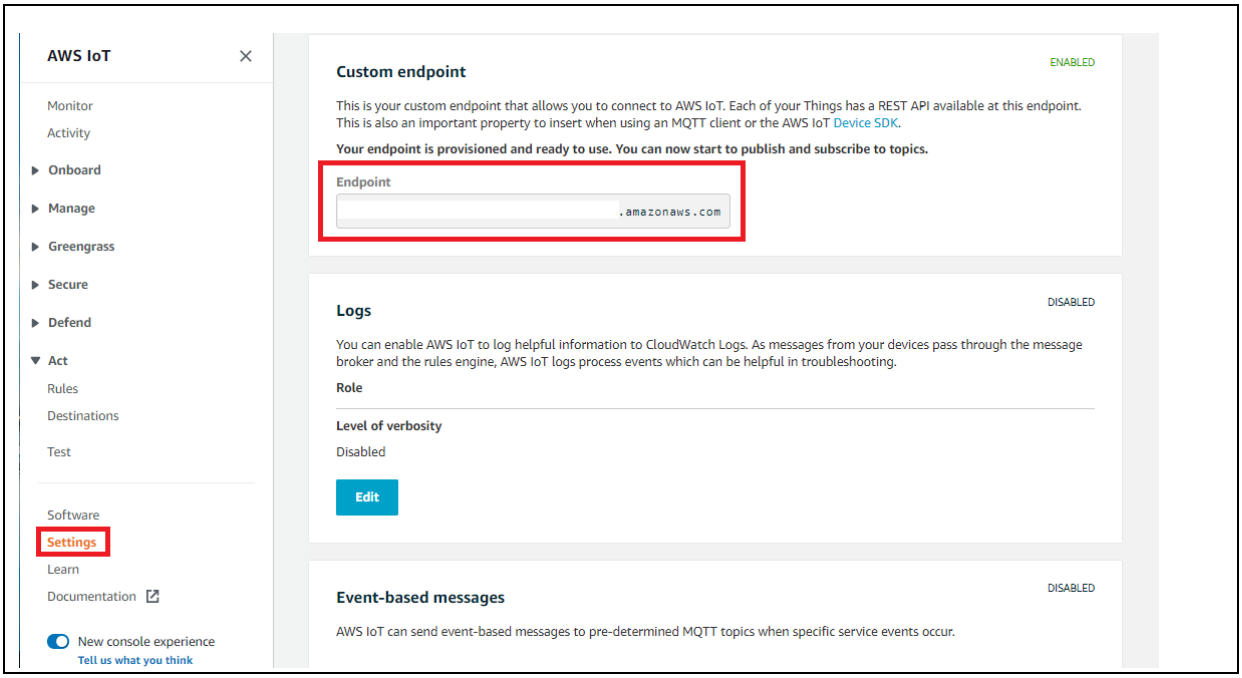
  clientcredentialWIFI_SSID

  clientcredentialWIFI_PASSWORD

```
* FreeRTOS V202002.00

#ifndef __AWS_CLIENTCREDENTIAL__H__
#define __AWS_CLIENTCREDENTIAL__H__

* @brief MQTT Broker endpoint.
#define clientcredentialMQTT_BROKER_ENDPOINT ""

* @brief Host name.
#define clientcredentialIOT_THING_NAME ""

* @brief Port number the MQTT broker is using.
#define clientcredentialMQTT_BROKER_PORT              8883

* @brief Port number the Green Grass Discovery use for JSON retrieval from cloud is using.
#define clientcredentialGREENGRASS_DISCOVERY_PORT     8443

* @brief Wi-Fi network to join.
#define clientcredentialWIFI_SSID                     ""

* @brief Password needed to join Wi-Fi network.
#define clientcredentialWIFI_PASSWORD                 ""
```

aws_clientcredential.h

To find the endpoint for your account, use the AWS IoT console at console.aws.amazon.com/iot. In the left panel, choose Settings. The endpoint is listed under Custom endpoint as following snapshot:



The endpoint in AWS IoT

- Make sure that writing the private key and certificate into Wi-Fi-Pmod-Expansion-Board successfully. If not, back to step E. The steps to write the private key and certificate into Wi-Fi-Pmod-Expansion-Board to rewrite to Wi-Fi.