Getting started with the Renesas RX65N Cloud Kit

This tutorial provides instructions for getting started with the Renesas RX65N Cloud Kit. If you do not have the Renesas RX65N Cloud Kit, visit the AWS Partner Device Catalog, and purchase one from our partners.

This document explains 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 Hardware Requirement.

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

C Creating Policy for Device

D Device on AWS IoT Core

E Set up the Renesas RX65N Cloud Kit.

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

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

H Monitoring MQTT messages on the cloud.

A. Hardware Requirement

1. Renesas RX65N Cloud Kit (RTK5RX65NDSODOODBE):

   RX65N-Cloud-Kit - Renesas RX65N Cloud Kit | Renesas

2. Mini-B USB cables x2
   These cables can be used to connect the PC to the Renesas RX65N Cloud Kit

Go to Troubleshooting section to solve any issues.

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

Note: Host machine running Windows 8.1 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.
Note: Linux and MacOS are not supported.

To download and install the RX Family C/C++ V3.03.00 Compiler Package

1. Download RX Family C/C++ V3.03.00 Compiler Package.
2. Open the executable and install the compiler.

To download and install the GCC for Renesas 8.3.0.202004-GNURX Toolchain Package

3. Download GCC for Renesas 8.3.0.202004-GNURX Toolchain.
4. Open the executable and install the compiler.

To download Tera Term

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

C. Create a Policy for a Device

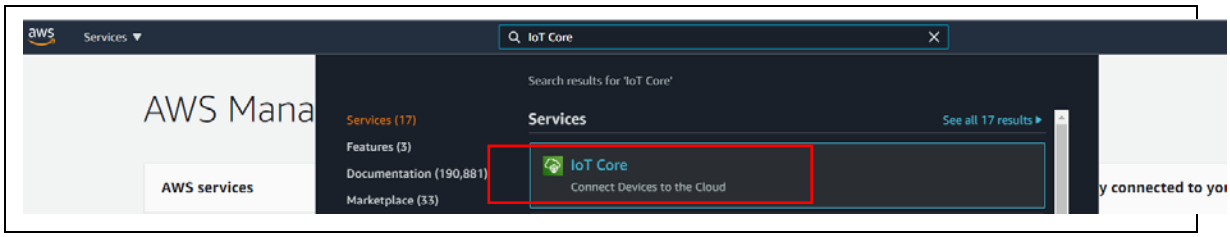
User needs to create AWS account. Refer to the instructions at Set up your AWS Account.
Follow the steps outlined in these sections to create your account and a user and get started:

- 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 AWS account already in the past, please skip this step.

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



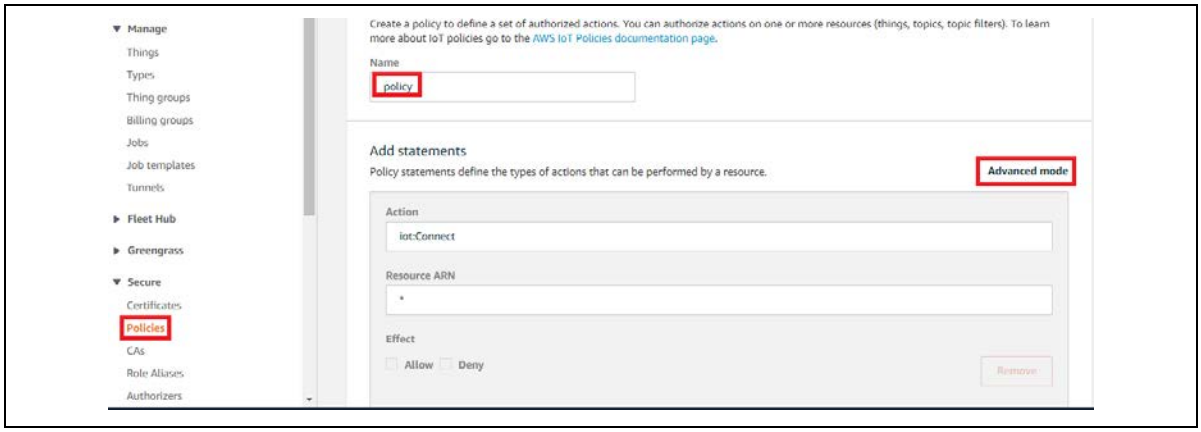
AWS IoT Core Selection

2. Go to Secure→Policies

Click on **Create** to create a policy

Create a policy

In the **Name** field, enter a name for the policy.
Then, change to **Advanced mode**



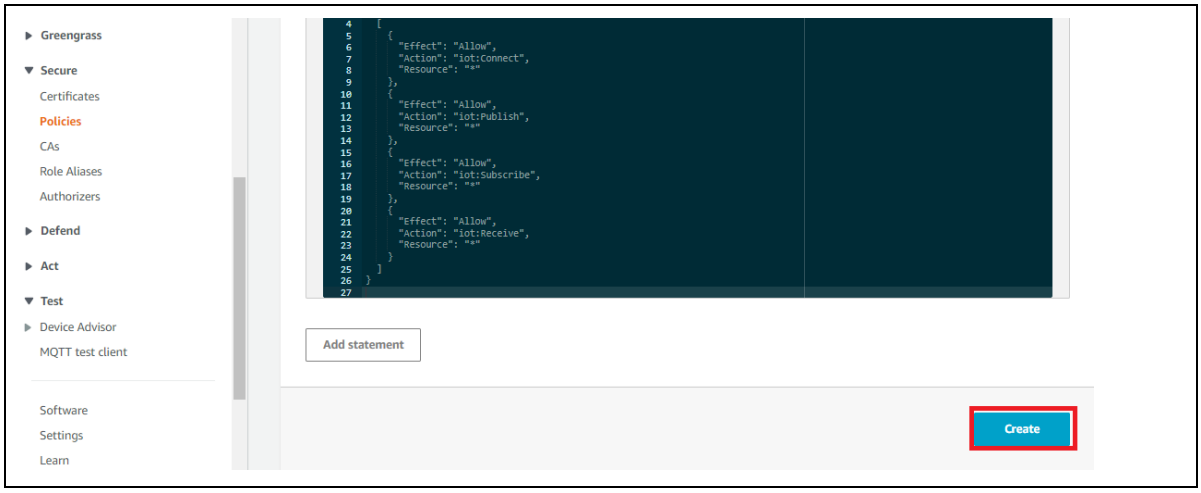
Give a policy name

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



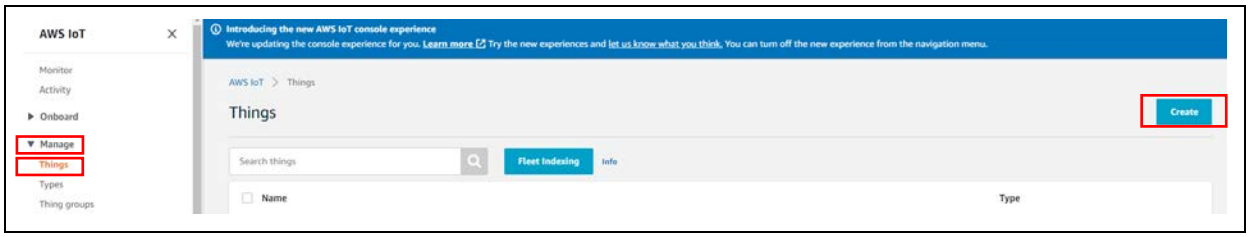
Create a policy

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

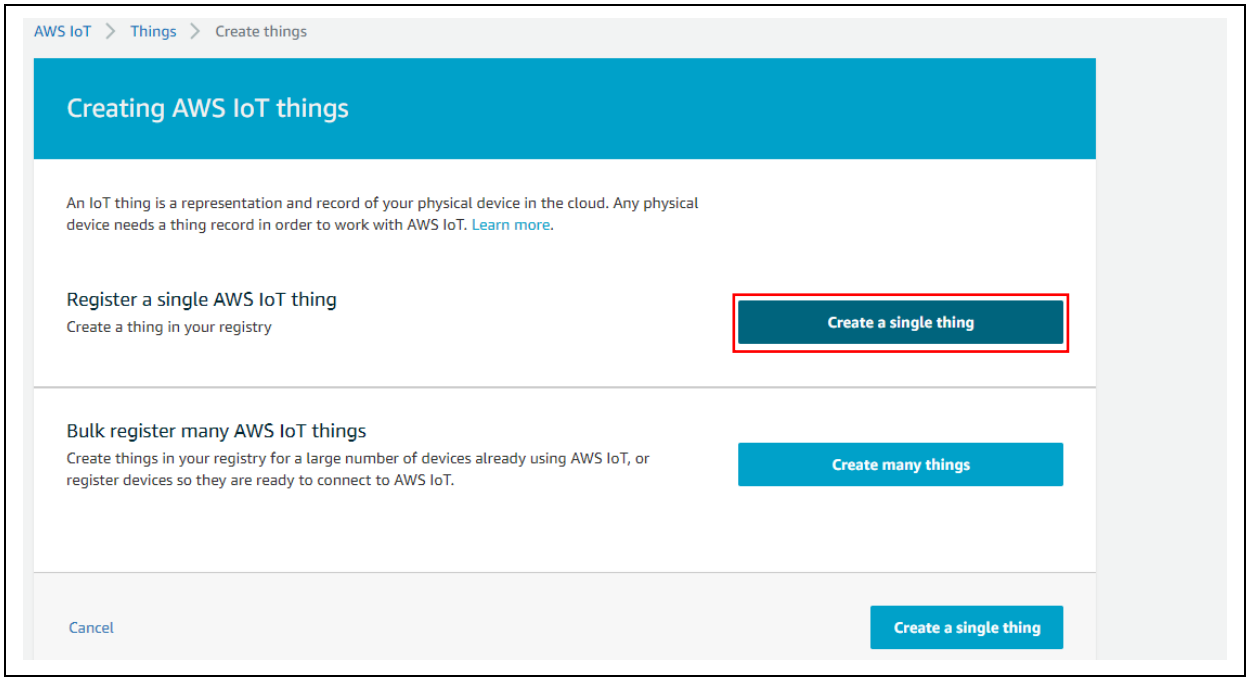
4. Create a Thing

Select **Manage→ Things→Create** to create a thing



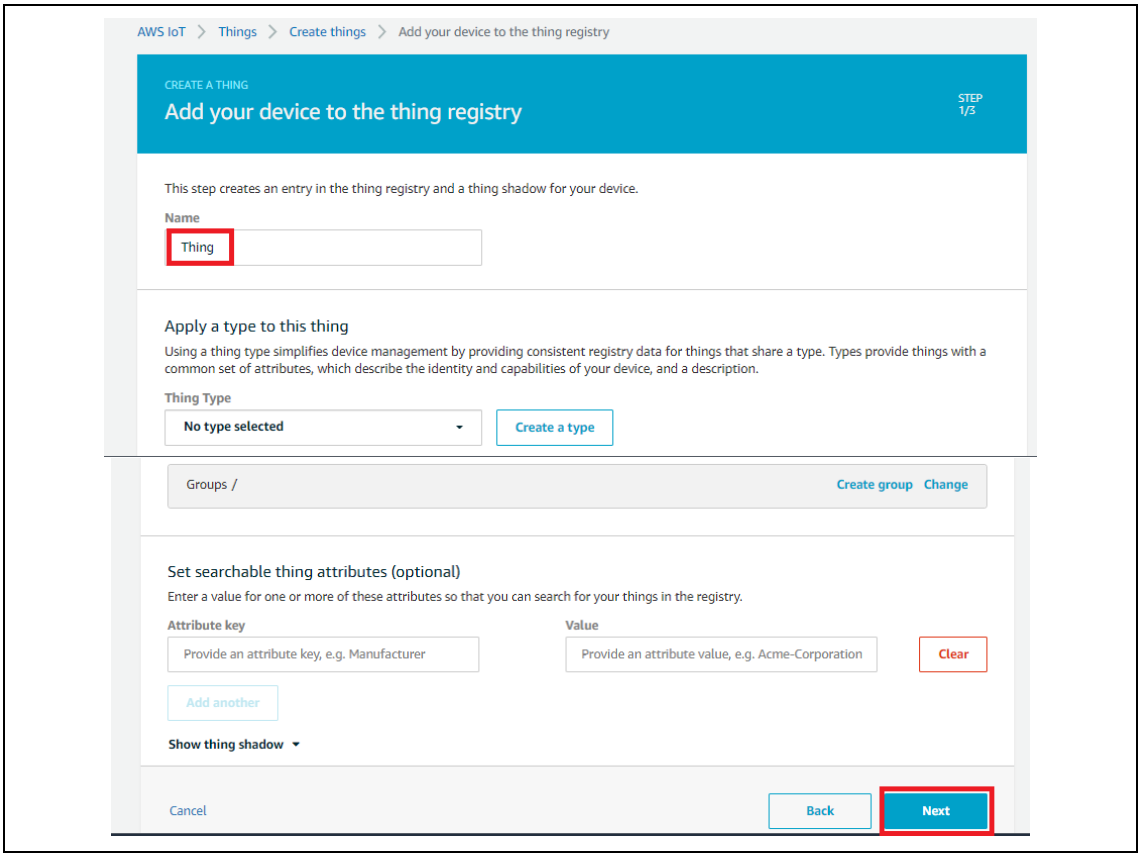
Create a thing

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



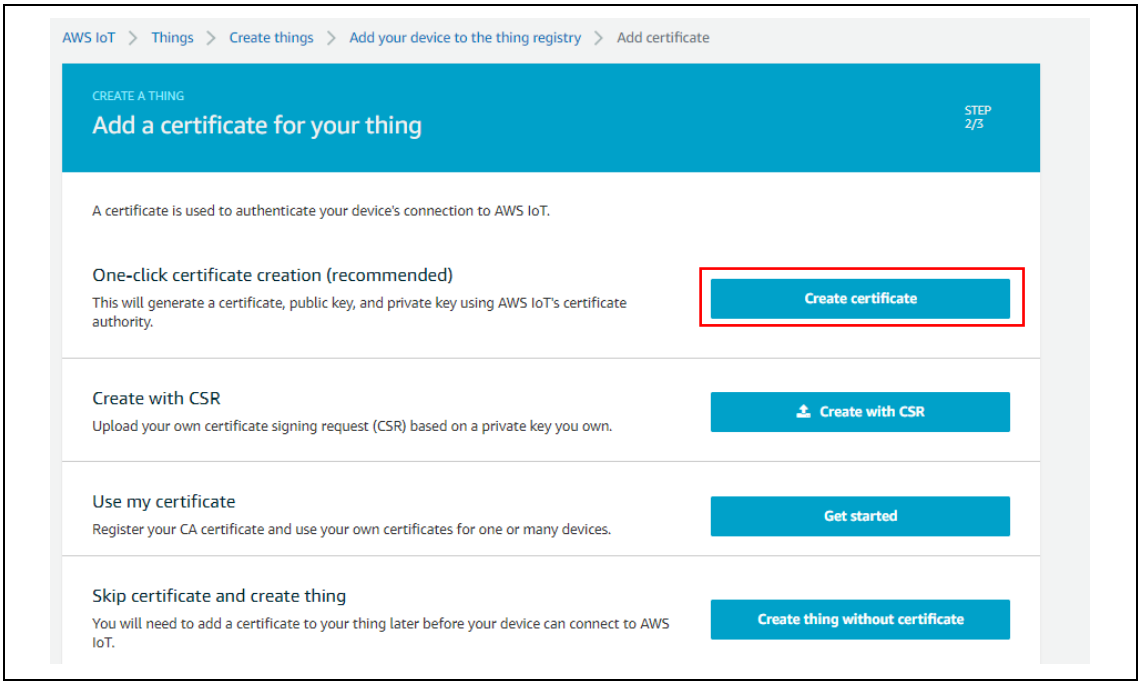
Create a single thing

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



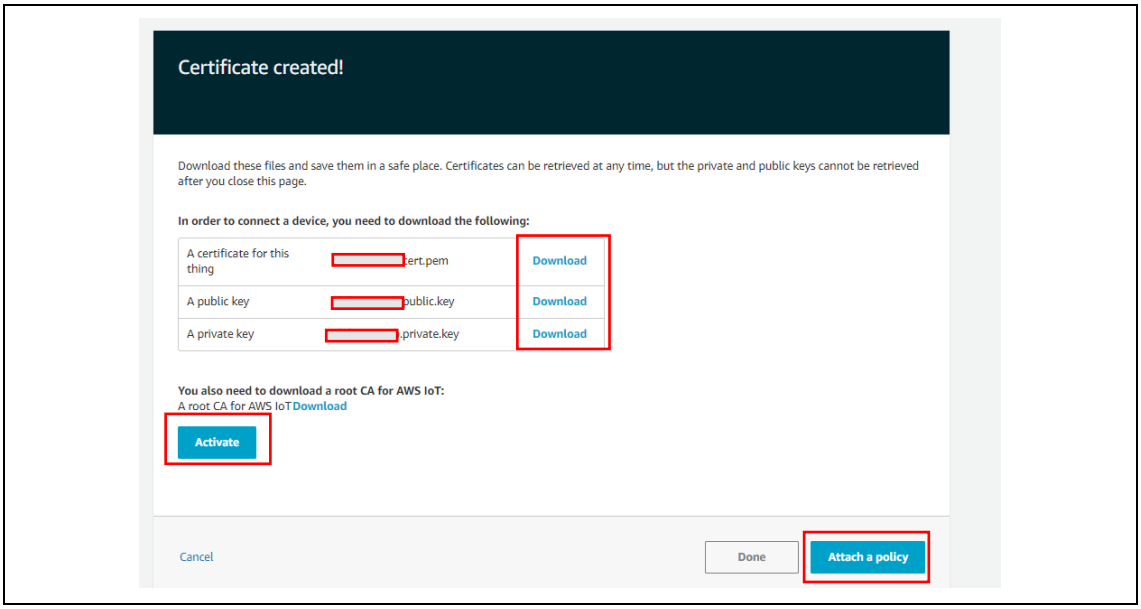
Add name to a single thing

7. Add a certificate for thing



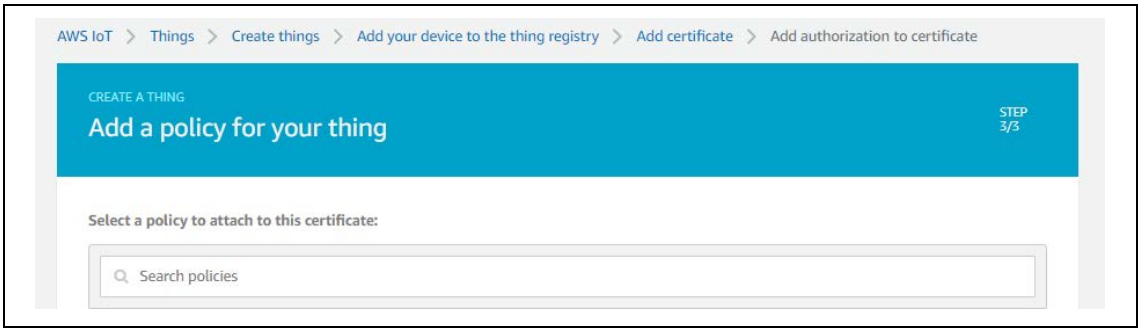
Create a certificate for thing

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** and choose the policy you created in section C.



Attach a policy

9. Register policy to thing

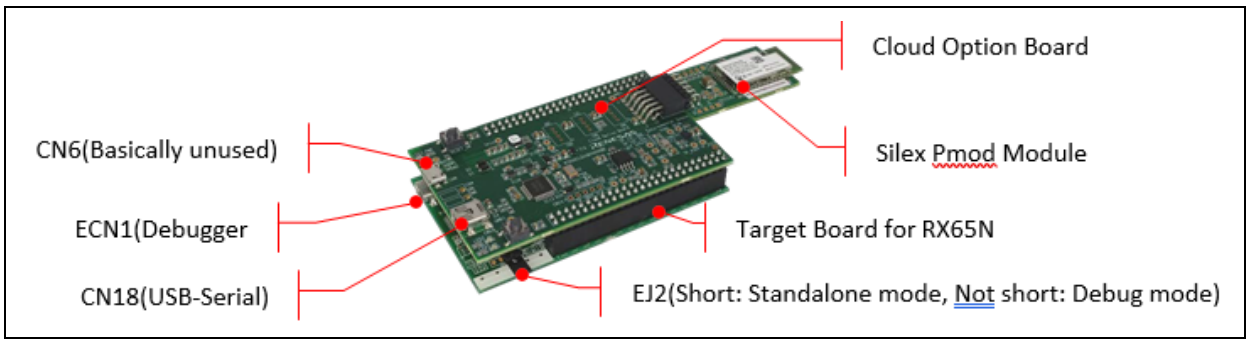


Register policy to thing

E. Set up the Renesas RX65N Cloud Kit

To confirm functionality on Renesas RX65N Cloud Kit

- Make sure to connect Silex PMOD Wi-Fi module to CN5.
- Remove the jumper pin (EJ2) to switch to debug mode.
- Connect USB cable form Cloud Option board (Top board) CN18 to any USB port on your PC.
- Connect USB cable from Target board connector ECN1 (Bottom board) to any USB port on your PC. This will connect to on-board E2 Lite debugger.
- The E2 Lite debugger drivers will now be installed. Note that this may take up to a minute.
  Note that administrator privileges are required to install the drivers.



Connect Renesas RX65N Cloud Kit to power PC

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

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

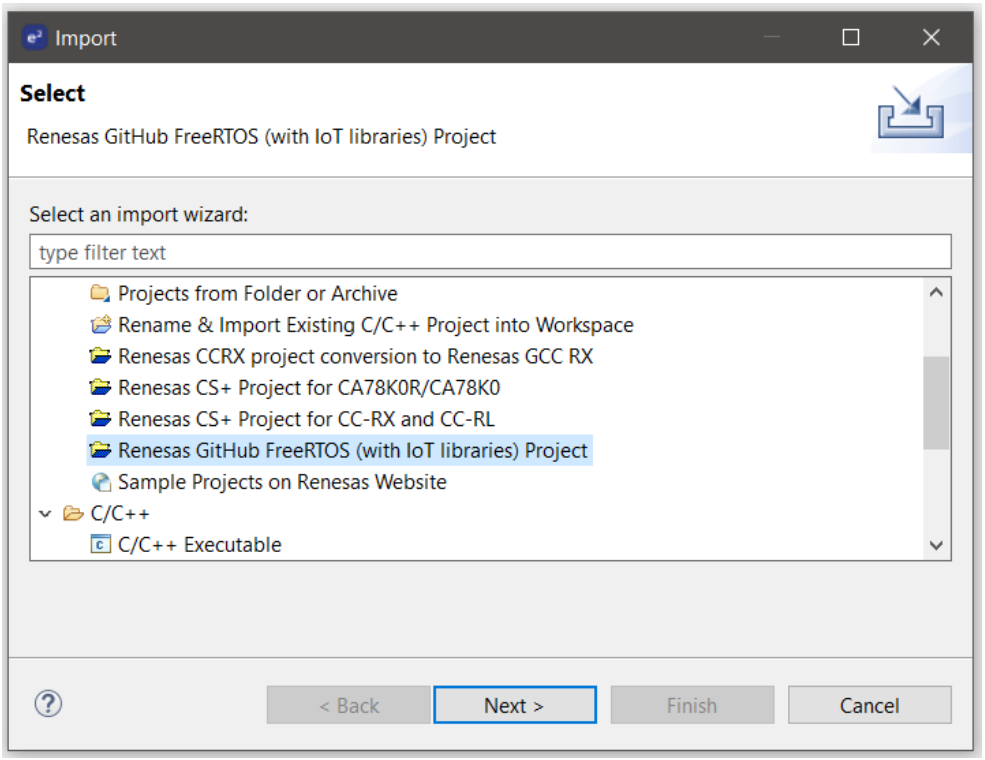
Build the FreeRTOS Demo in e²studio

To download and build the demo in e²studio

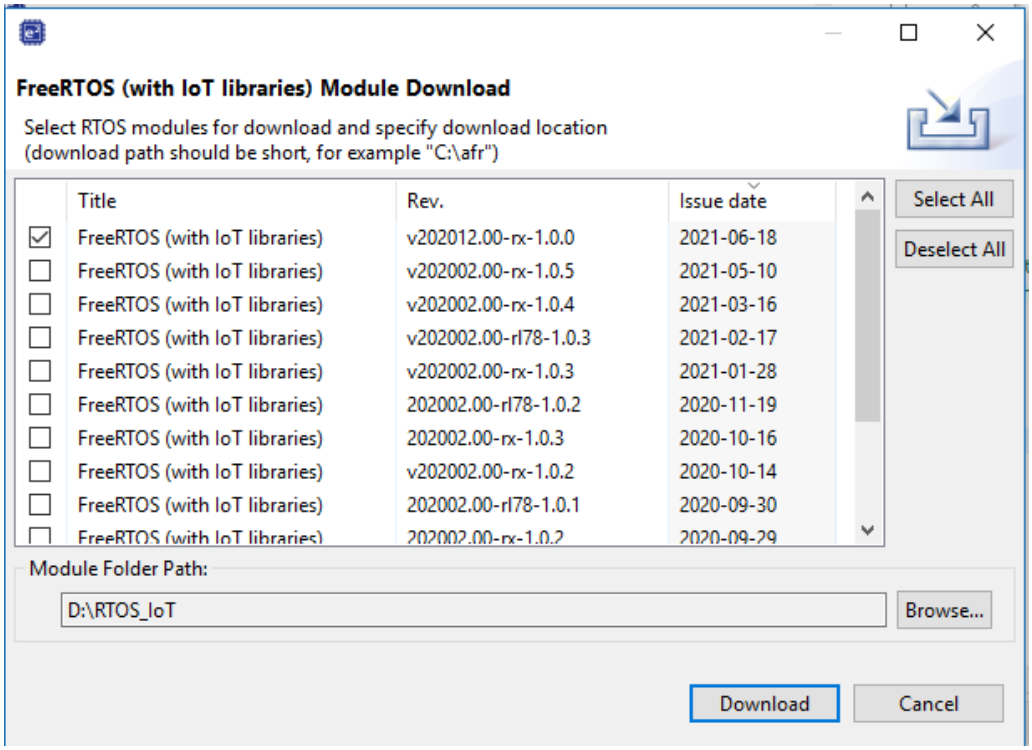
In this tutorial, the path to the FreeRTOS download directory is referred to as `freertos`.

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-RX v3.03.00`** or **`GCC for Renesas 8.3.0.202004-GNURX`** 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: v202012.00-rx-1.0.0) 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 to import:
    - To import CC-RX demo project, choose `${freertos}/projects/renesas/rx65n-cloud-kit-uart-sx-ulpgn/e2studio/aws_demos`), then choose **Finish**.
    - To import GNURX demo project, choose `${freertos}/projects/renesas/rx65n-cloud-kit-uart-sx-ulpgn/e2studio-gcc/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-RX compiler. To install the License Manger, see the License Manager download page.

G. 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 Renesas RX65N Cloud Kit.
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**.

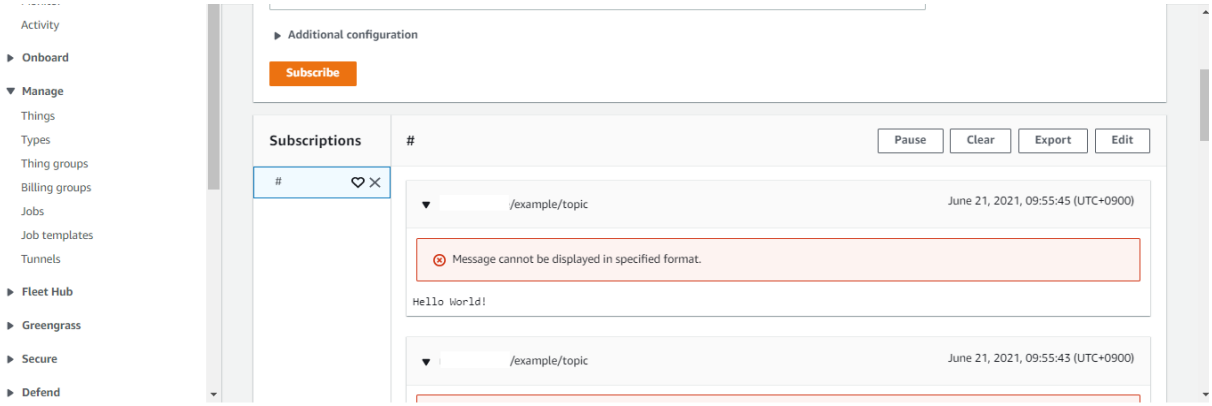
6. e²studio might ask to change to **Renesas Debug Perspective**. Choose **Yes**.
7. After the code is downloaded to the board, choose **Resume** to run the code up to the first line of the main function. Choose **Resume** again to run the rest of the code.

H. Monitoring MQTT messages in 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 test client.
3. In **Subscription topic**, enter #, 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.

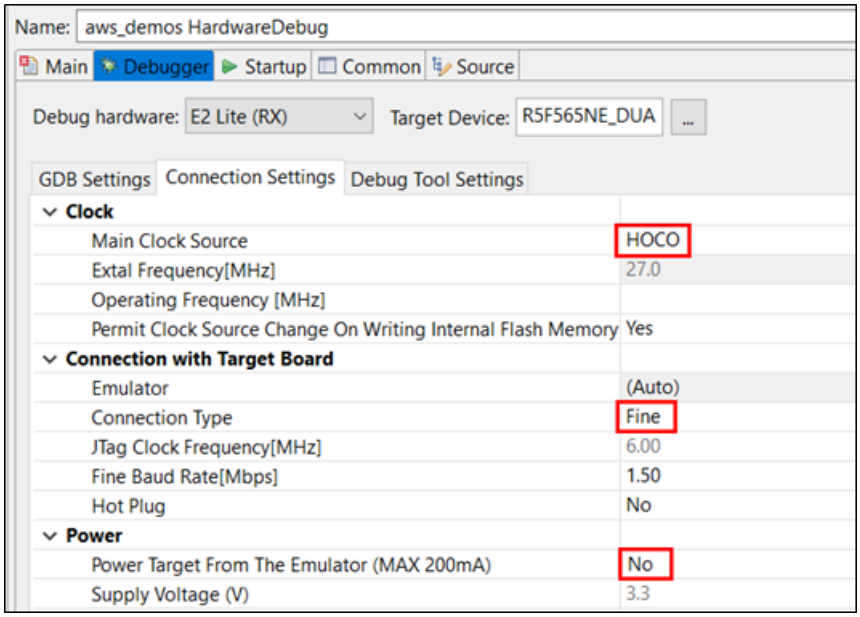
Troubleshooting

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

The following information is for debugging if any troubles.

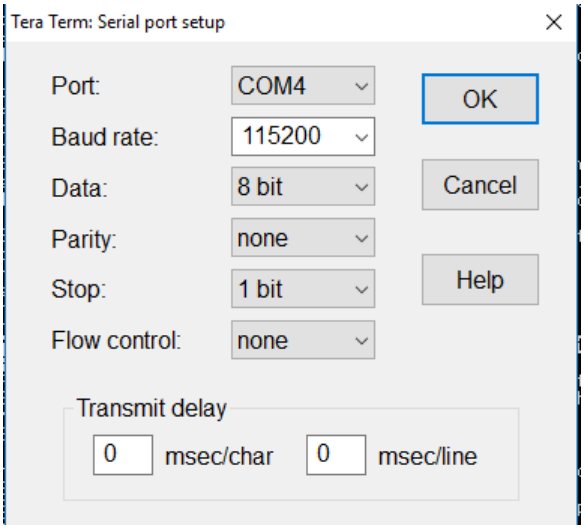
1. **Open e2studio to debug**

   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 v202012.00-rx-1.0.0 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.

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

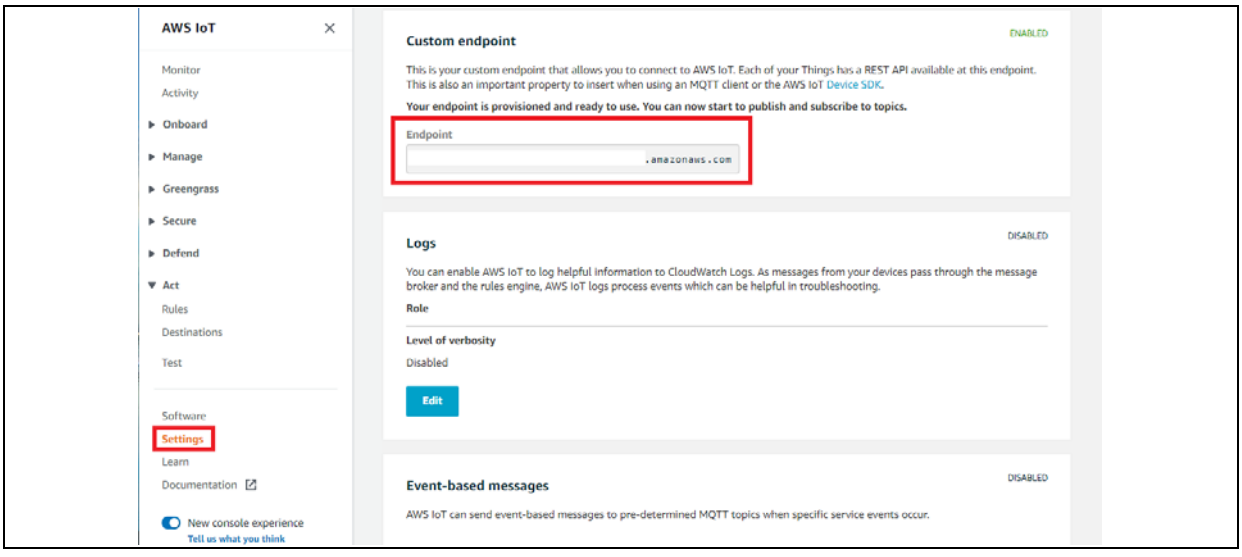
  clientcredentialWIFI_PASSWORD

For “clientcredentialIOT_THING_NAME”, input name of the thing you created in section D.

```
* 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