Getting Started for Edit and Write Programs with the Renesas RX65N Cloud Kit

Required Resources
To build and run the RX65N Cloud kit example, you will need following resources:

Development tools & software

- e² studio IDE v7.4.0 (e2studio download)
- Renesas CC-RX compiler v3.01 (CC-RX Compiler download)

Hardware

- Renesas RX65N Cloud kit, P/N: RTK5RX65NDSODOODBE
  Renesas RX65N Cloud kit is installed Application Example “Sensor Data Upload Program” by default. (www.renesas.com/rx-cloud)
- PC running Windows 7 or 10; the Tera Term console, or similar application; and an installed web browser (Google Chrome, Internet Explorer, Microsoft Edge, Mozilla Firefox, or Safari).
- Wi-Fi internet access

Before you begin, see Prerequisites.

If you do not have an RX65N Cloud Kit, you can order one from Renesas.

Setting Up Your Environment

FreeRTOS for the RX65N Cloud kit uses e² studio IDE and CC-RX compiler. Before you begin, install the IDE and compiler to your machine:

To install e² studio:

1. Browse to e² studio and choose **Download Software. Make sure to use** e² studio version 7.4.0 or later.
2. Unzip and run the installer. Follow the prompts for the installation.

To install CC-RX Compiler:

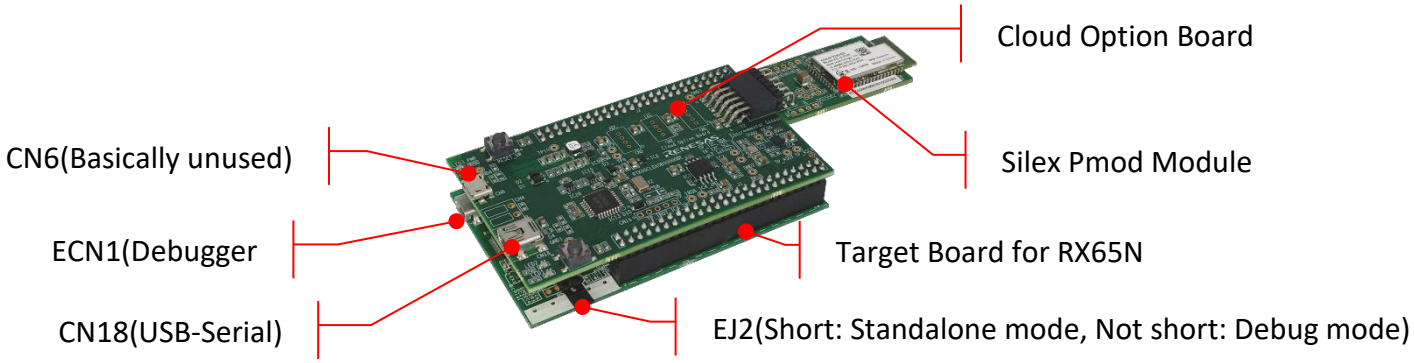
1. Browse to CC-RX Compiler and choose **Download Software. Make sure to use** CC-RX compiler version 3.01.00.
2. Unzip and run the installer. Follow the prompts.

Note: You will need to register on the Renesas website to download the software. The CC-RX compiler is evaluation version only and valid for 60 days.

If you experience issues during installation, contact Renesas Support

Connecting a Debugger

1. Assemble the kit as below.



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

Download and Build FreeRTOS

After your environment is set up, you can download ”MQTT Communication Test Program” and run the demo code.

Download FreeRTOS

1. Browse to the Renesas Cloud Solution page and download the code ”MQTT Communication Test Program” for RX65N Cloud Kit.

2. Unzip the downloaded file to a folder and make a note of the folder path. In this tutorial, this folder is referred to as `BASE_FOLDER`.

Note: The e² studio doesn't support long path names. To accommodate the files in the FreeRTOS projects, make sure the path to the directory is less than 260 characters and does not contain spaces or special characters.

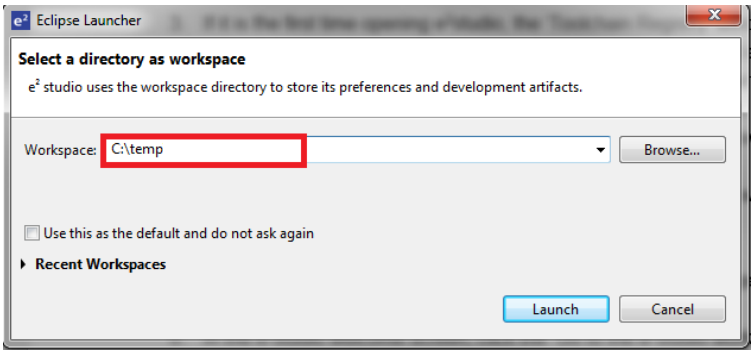
Import the FreeRTOS Demo Code into Your IDE

To import the code "MQTT Communication Test Program" into e² studio IDE

1. e² studio integrates various tools such as compiler, an assembler, debugger and an editor into a common graphical user interface. Start e² studio:

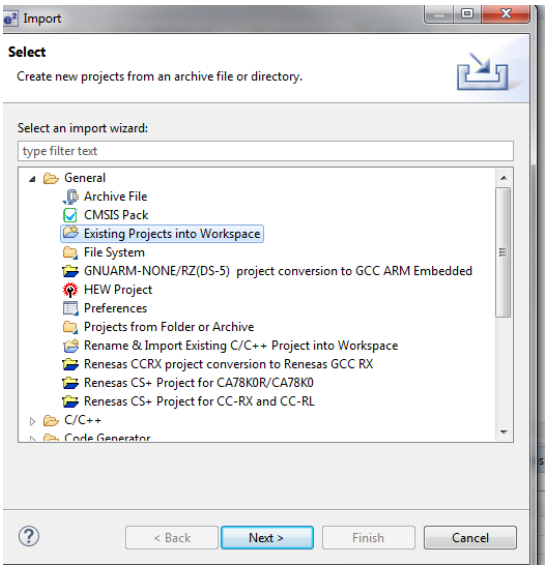
   ```
   Windows™ 7: Start Menu>All Programs>Renesas Electronics e2studio>e2studio
   Windows™ 10: Start Menu>All Apps> Renesas Electronics e2studio>e2studio
   ```

2. In the 'Select a workspace' folder that appears, browse to the folder "…`BASE_FOLDER`". Click 'OK' to continue.

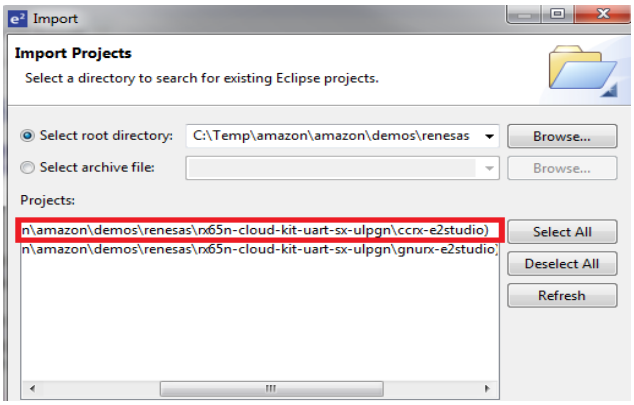


3. If it is the first time opening e² studio, the 'Toolchain Registry' window will open. In the 'Toolchain Registry' dialog select Renesas Toolchains and ensure that 'CC-RX v3.01.00' is selected. Click 'Register'. A dialog will appear "Selected Toolchains were successfully integrated with e² studio ". Click 'OK'.
4. In the 'Code Generator Registration' dialog click 'OK'. This window opens up first time only after installation.
5. A 'Code Generator COM component register' dialog will pop-up with the text "Please restart e² studio to use Code Generator". Click 'OK'.
6. In the 'Restart e² studio dialog, click 'OK'.

7. Once e^2 studio is restarted, then ‘Select a workspace’ window appears again with the folder path selected in step 2. Click ‘OK’.
8. In the e^2 studio welcome screen, click ‘Go to the e^2 studio workbench’ arrow icon, on the far right.
9. Right click in the Project Explorer window, and select ‘Import’.
10. In the import wizard, select General > Existing Projects into Workspace, and click ‘Next’.



11. Click the ‘Browse’ button, and locate the following directory ‘*<BASE_FOLDER>*\demos\renesas\rx65n-cloud-kit-uart-sx-ulpgn\ccrx-e2studio.



12. Click “Finish”.
13. In the **Project** menu, choose **Project->Build All**. The project should build with no errors.

Note: There will be a warning message in build console showing that the "License manager is not installed". You can ignore this warning unless you have the License key for the CC-RX compiler and ready to install the License manager (License Manager download).

Configure Your Project

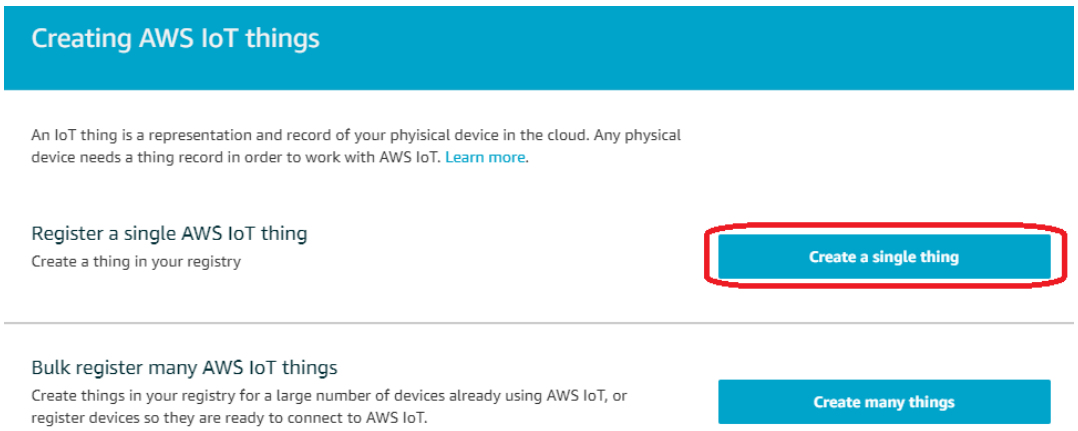
To configure your project, you need to know your AWS IoT endpoint and Thing name that represents your board.

Configure AWS IoT endpoint and Wi-Fi Credentials

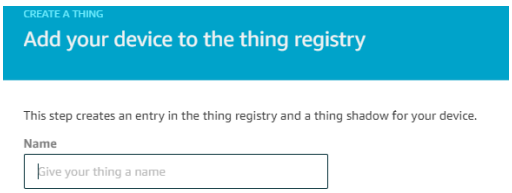
1. Login to aws account and Click on IoT Core services.
2. In the left navigation pane, choose **Settings**.
3. Copy your AWS IoT endpoint from the **Endpoint** text box. It should look like `<1234567890123>.iot.<us-east-1>.amazonaws.com`.
4. Open `aws_demos/application_code/common_demos/include/aws_clientcredential.h` and set `clientcredentialMQTT_BROKER_ENDPOINT` to your AWS IoT endpoint.

```
static const char clientcredentialMQTT_BROKER_ENDPOINT[] = "Paste AWS IoT Broker endpoint here.";
```

5. In the left navigation pane, Click on Manage-> Things, and then Click on 'Create" to create a new Thing.
6. In the next window, click on "Create a single thing".



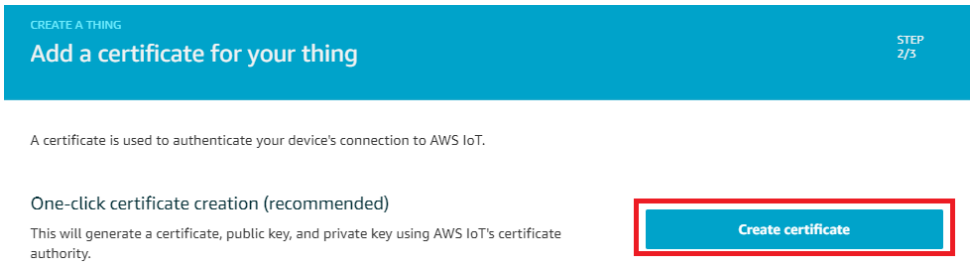
7. Enter thing Name for your IoT board.



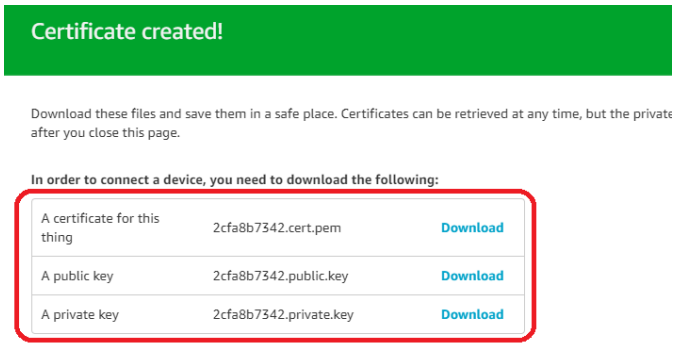
8. Open `aws_demos\application_code\common_demos\include\aws_clientcredential.h`. Specify AWS IoT thing for your board in the following `#define` constants from **Thing** pane in AWS IoT console.

```
#define clientcredentialIOT_THING_NAME "Paste AWS IoT Thing name here."
```

9. Click next. In next window click on “Create Certificate”



10. Download the certificate.



11. Open `aws_demos\application_code\common_demos\include\aws_clientcredential.h`. Enter SSID and password for the network in the following #define statement.

```
#define clientcredentialWIFI_SSID       "Paste Wi-Fi SSID here."

#define clientcredentialWIFI_PASSWORD   "Paste Wi-Fi password here."
```

Make sure to save your changes.

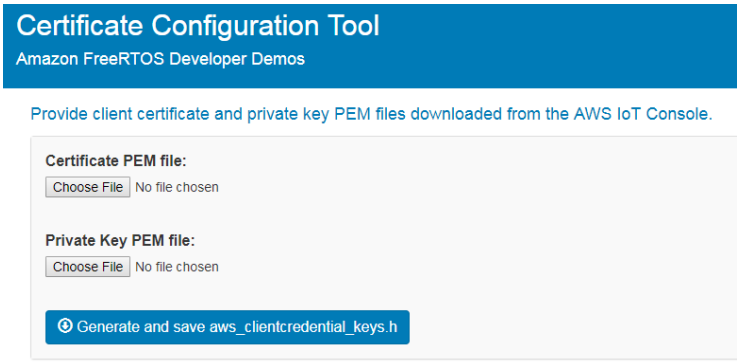
Configure certificate and private key

The certificate and private key must be hard-coded into the FreeRTOS demo code. This is for demo purposes only. Production level applications should store these files in a

secure location. FreeRTOS is a C language project, and the certificate and private key must be specially formatted to be added to the project.

To format your certificate and private key

1. In a browser window, open certificate configuration tool from project `<BASE_FOLDER>\tools\certificate_configuration\CertificateConfigurator.html`.



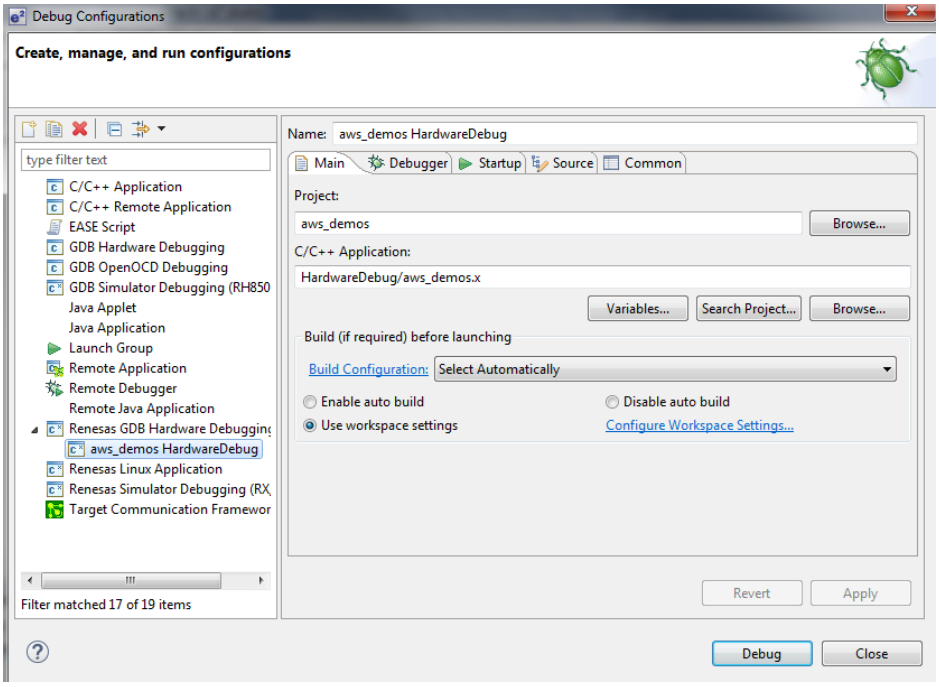
2. Under **Certificate PEM file**, choose `certificate.pem.crt` you downloaded from the AWS IoT console in previous step.
3. Under **Private Key PEM file**, choose `private.pem.key` you downloaded from the AWS IoT console in previous step.
4. Choose **Generate and save aws_clientcredential_keys.h**, and then save the file in`<BASE_FOLDER>\demos\common\include`. This overwrites the file aws_clientcredential_keys.h in the directory.

Run the FreeRTOS Demo

To run the FreeRTOS demos on the RX65N Cloud Kit:

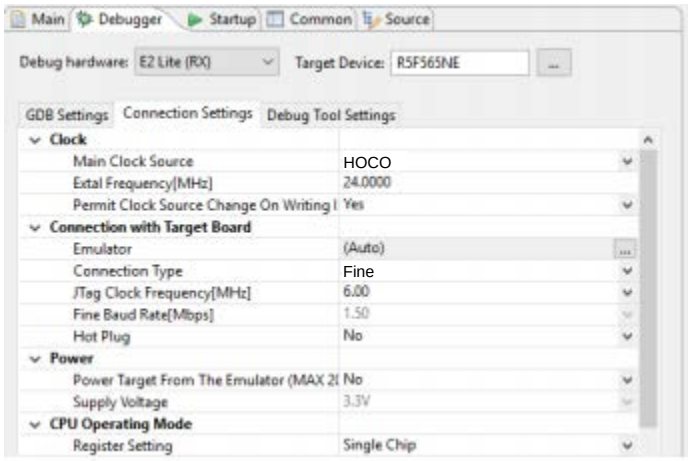
1. Sign in to the AWS IoT console.
2. In the left navigation pane, choose **Test** to open the MQTT client.
3. In the **Subscription topic** text box, type '#' that is wildcard, and then choose **Subscribe to topic**.
4. Rebuild the project, "**Project->Build All**".
5. Connect RX65N Cloud kit board, on board E2 Lite Debugger to PC using USB cable.

6. The debugging can be started by clicking the ‘Run-> Debug Configuration’. Click the symbol **“aws_demos HardwareDebug”** under ‘Renesas GDB Hardware Debugging’ by expanding the list.



7. Click the **‘Debugger**’ tab, then the ‘**Connection Settings**’ secondary tab. Review the settings listed in the screenshot below.

i.



8. Click the ‘**Debug**’ button to download the code to the target board to begin debugging. A firewall warning may be displayed for ‘e2-server-gdb.exe’. Select

the check-box for ‘Private networks, such as my home or work network’, and click ‘Allow access’.

9. e^2 studio may ask you to change to the ‘Renesas Debug Perspective’. Click ‘Yes’.
10. Once the code has been downloaded, click the ‘Resume’ button to run the code up to the first line of the main function. Click ‘Resume’ button again to run the target through the rest of the code.

In the AWS IOT console MQTT client, you should see the MQTT messages sent by your device.

Note:

Please visit the following GitHub repository to get the latest projects (prototype), but not yet certified for other Renesas devices, compilers, and target boards.

https://github.com/renesas-rx/amazon-freertos

Troubleshooting

If no messages appear in the AWS IoT console, try the following:

1. Check that your network credentials are valid.
2. Verify the Wi-Fi connection and key to make sure the connection is working.