

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 <u>AWS Partner Device Catalog</u>, 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
- 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 <u>Renesas e²studio installer</u> 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 $\underline{e^2studio}$ 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 <u>RX Family C/C++ V3.03.00 Compiler Package</u>.
- 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 <u>Set up your AWS Account</u>. 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	Services 🔻		Q IoT Core	×
	AWS Mana	Services (17) Features (3)	Search results for 'IoT Core' Services	See all 17 results >
	AWS services	Documentation (190,881) Marketplace (33)	IoT Core Connect Devices to the Cloud	y connected to yo

AWS IoT Core Selection

2. Go to Secure → Policies

Click on Create to create a policy

AWS IOT	AWS Int > Policies	
Monitor	Policies	Create
Activity Onboard	Search policies Q	
 Manage Fleet Hub 	Name	
 Fleet Hub Greengrass 	any and a second sec	
▼ Secure	a supergrame	
Certificates Policies	 manuficial mails 	
CAs	C manyants	

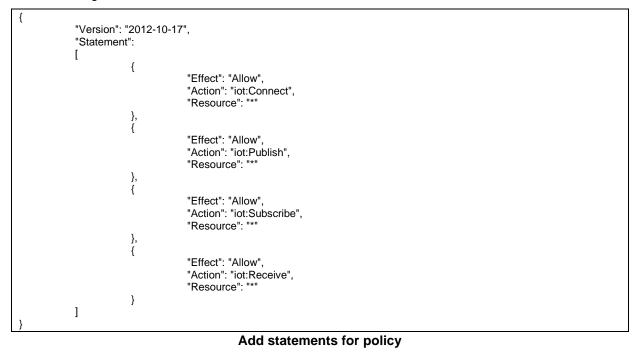
Create a policy

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

♥ Manage Things Types	Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page. Name policy
Thing groups Billing groups Jobs Job templates Tunnets	Add statements Policy statements define the types of actions that can be performed by a resource. Advanced mode
Fleet Hub	Action
▶ Greengrass	iot:Connect
▼ Secure	Resource ARN
Certificates	
Policies CAs Role Aliases	Effect Allow Deny Remove
Authorizers	

Give a policy name

Add following text to Advanced mode



3. Create a policy

 Greengrass Secure Certificates Policies CAs Role Aliases Authorizers Defend Act Test Device Advisor MQTT test client 	4 [* Ffect*: *Allow", *Action": *int:connect*, *Resource*: *** 9 }, *effect*: *Allow", *action": *int:subscribe", *action": *int:subscribe", *action: *int:subscribe", *action: *int:subscribe", *effect*: *Allow", *action: *int:subscribe", *action: *int:subscr
Software Settings Learn	Creste

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 <u>Example policies</u> and <u>Security Best practices</u>.

D. Creating Device on AWS IoT Core

4. Create a Thing

Select Manage→ Things→Create to create a thing

AWS IoT ×	O Introducing the new AWS IoT console experience We're updating the console experience for you. Learn more [2] Try the new experiences and let us know what you think. You can turn off the new experience from the navigation menu.	
Monitoe Activity ► Onboard	AWS INT > Things Things	Create
Manage Things Types Thing groups	Search things	

Create a thing

5. Select the Create a single thing

Creating AWS IoT things	
An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. Learn more.	
Register a single AWS IoT thing Create a thing in your registry	Create a single thing
Bulk register many AWS IoT things	
Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.	Create many things
Cancel	Create a single thing

Create a single thing

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

AWS IoT > Things > Create things > Add your dev	vice to the thing registry
CREATE A THING Add your device to the thing rec	gistry I/3
This step creates an entry in the thing registry and a the Name	thing shadow for your device.
Thing	
Apply a type to this thing Using a thing type simplifies device management by p common set of attributes, which describe the identity	providing consistent registry data for things that share a type. Types provide things with a r and capabilities of your device, and a description.
Thing Type No type selected	Create a type
Groups /	Create group Change
Set searchable thing attributes (optional) Enter a value for one or more of these attributes so th	nat you can search for your things in the registry.
Attribute key	Value
Provide an attribute key, e.g. Manufacturer	Provide an attribute value, e.g. Acme-Corporation
Add another	
Add another Show thing shadow	

Add name to a single thing

7. Add a certificate for thing

Add a certificate for your thing	STEP 2/3
A certificate is used to authenticate your device's connection to AWS IoT.	
One-click certificate creation (recommended) This will generate a certificate, public key, and private key using AWS IoT's certificate authority.	Create certificate
Create with CSR Upload your own certificate signing request (CSR) based on a private key you own.	Create with CSR
Use my certificate Register your CA certificate and use your own certificates for one or many devices.	Get started
Skip certificate and create thing	Create thing without certificate

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.

Certificate created!		
Download these files and save them in a safe place. Certificates ca after you close this page.		ny time, but the private and public keys cannot be retrieved
In order to connect a device, you need to download the followin	g:	
A certificate for this thing	Download	
A public key public.key	Download	
A private keyprivate.key	Download	
You also need to download a root CA for AWS IoT: A root CA for AWS IoTDownload		
Cancel		Done Attach a policy

Attach a policy

9. Register policy to thing

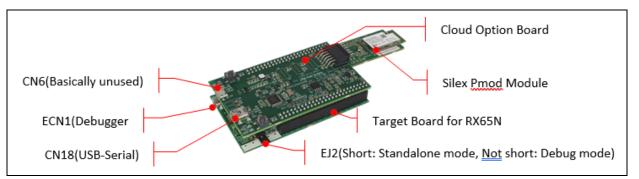
بسيميد سمك بمثلمه ماما		
d a policy for your	thing	STEP 3/3
	umiy	
ect a policy to attach to this cert	ficate	

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.

e Import	—		×
Select Renesas GitHub FreeRTOS (with IoT libraries) Project		Ľ	4
Select an import wizard:			
type filter text			
 Projects from Folder or Archive Rename & Import Existing C/C++ Project into Workspace Renesas CCRX project conversion to Renesas GCC RX Renesas CS+ Project for CA78K0R/CA78K0 Renesas CS+ Project for CC-RX and CC-RL Renesas GitHub FreeRTOS (with IoT libraries) Project Sample Projects on Renesas Website C/C++ C/C++ Executable 			~
? < Back Next > Finish		Cancel	

- 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: <u>v202012.00-rx-1.0.0</u>) you want to work on by clicking the checkbox, then choose **Download**.

elec	RTOS (with IoT libraries) Mod tt RTOS modules for download and mload path should be short, for ex	specify download location			Ľ
	Title	Rev.	Issue date	^	Select All
/	FreeRTOS (with IoT libraries)	v202012.00-rx-1.0.0	2021-06-18		Deselect A
	FreeRTOS (with IoT libraries)	v202002.00-rx-1.0.5	2021-05-10		
	FreeRTOS (with IoT libraries)	v202002.00-rx-1.0.4	2021-03-16		
	FreeRTOS (with IoT libraries)	v202002.00-rl78-1.0.3	2021-02-17		
	FreeRTOS (with IoT libraries)	v202002.00-rx-1.0.3	2021-01-28		
	FreeRTOS (with IoT libraries)	202002.00-rl78-1.0.2	2020-11-19		
	FreeRTOS (with IoT libraries)	202002.00-rx-1.0.3	2020-10-16		
	FreeRTOS (with IoT libraries)	v202002.00-rx-1.0.2	2020-10-14		
	FreeRTOS (with IoT libraries)	202002.00-rl78-1.0.1	2020-09-30		
	FreeRTOS (with InT libraries)	202002.00-rx-1.0.2	2020-09-29	¥	
/loc	lule Folder Path:				
	D:\RTOS_loT				Browse
L					cronsen

- 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/rx65ncloud-kit-uart-sx-ulpgn/e2studio/aws_demos), then choose Finish.
 - To import GNURX demo project, choose \${freertos}/projects/renesas/rx65ncloud-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 loT 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

Activity		Addition	nal configur	ation	
Onboard		Subscri	_		
Manage		Subser			
Things					
Types		Subscrip	tions	#	Pause Clear Export Edit
Thing groups					
Billing groups		#	$\diamond \times$		
obs				▼ /example/topic	June 21, 2021, 09:55:45 (UTC+0900)
ob templates					
unnels				Message cannot be displayed in specified format.	
Fleet Hub				Hello World!	
Greengrass					
Secure				▼ /example/topic	June 21, 2021, 09:55:43 (UTC+0900)
Defend	*				

For the latest projects released by Renesas, see the renesas fork of the amazon-freertos repository on <u>GitHub</u>.

Troubleshooting

For general troubleshooting information about Getting Started with FreeRTOS, see <u>Troubleshooting</u><u>getting started</u>.

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.

Name: aws_demos HardwareDebug	
🖺 Main 🎋 Debugger 🕨 Startup 🗔 Common 🧤 Source	
Debug hardware: E2 Lite (RX) V Target Device: R5F565	5NE_DUA
GDB Settings Connection Settings Debug Tool Settings	
✓ Clock	
Main Clock Source	HOCO
Extal Frequency[MHz]	27.0
Operating Frequency [MHz]	
Permit Clock Source Change On Writing Internal Flash Mer	mory Yes
Connection with Target Board	
Emulator	(Auto)
Connection Type	Fine
JTag Clock Frequency[MHz]	6.00
Fine Baud Rate[Mbps]	1.50
Hot Plug	No
✓ Power	
Power Target From The Emulator (MAX 200mA)	No
Supply Voltage (V)	3.3

2. Tera term

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

Т	Tera Term: Serial port setup			×
	Port:	COM4	~	OK
	Baud rate:	115200	~	
	Data:	8 bit	\sim	Cancel
	Parity:	none	\sim	
	Stop:	1 bit	\sim	Help
	Flow control:	none	\sim	
	Transmit dela	y c/char 0	ms	ec/line

3. The Build errors

Make sure that <u>v202012.00-rx-1.0.0</u> 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.
<pre> #ifndefAWS_CLIENTCREDENTIAL_H #defineAWS_CLIENTCREDENTIAL_H</pre>
<pre> * @brief MQTT Broker endpoint.[] #define clientcredentialMQTT_BROKER_ENDPOINT "" </pre>
<pre>* @brief Host name #define clientcredentialIOT_THING_NAME ""</pre>
* @brief Port number the Green Grass Discovery use for JSON retrieval from cloud is using. #define clientcredentialGREENGRASS_DISCOVERY_PORT 8443
<pre> * @brief <u>Wi-Fi</u> network to join #define clientcredentialWIFI_SSID "" </pre>
<pre> * @brief Password needed to join <u>Wi-Fi</u> network.[] #define clientcredentialWIFI_PASSWORD "" </pre>

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:

AWS IOT X	Custom endpoint DHALED
Monitor Activity	This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.
 Onboard 	Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics. Endpoint
 Manage 	. amazonaws.com
Greengrass	
Secure Defend	Logs DISABLED
▼ Act	You can enable AWS loT to log helpful information to CloudWatch Logs. As messages from your devices pass through the message broker and the rules engine, AWS loT logs process events which can be helpful in troubleshooting.
Rules Destinations	Role
Test	Level of verbosity Disabled
Software	Edit
Settings	
Learn Documentation	Event-based messages DISARLED
New console experience Tell us what you think	AWS IOT can send event-based messages to pre-determined MQTT topics when specific service events occur.

The endpoint in AWS IoT