AmebaZ2 Amazon FreeRTOS Getting Started Guide

Realtek Semiconductor Corp.
No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan
Tel.: +886-3-578-0211. Fax: +886-3-577-6047
www.realtek.com

DISCLAIMER

Please Read Carefully:

Realtek Semiconductor Corp., (Realtek) reserves the right to make corrections, enhancements, improvements and other changes to its products and services. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Reproduction of significant portions in Realtek data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Realtek is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions.

Buyers and others who are developing systems that incorporate Realtek products (collectively, “Customers”) understand and agree that Customers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Customers have full and exclusive responsibility to assure the safety of Customers' applications and compliance of their applications (and of all Realtek products used in or for Customers’ applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Customer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Customer agrees that prior to using or distributing any applications that include Realtek products, Customer will thoroughly test such applications and the functionality of such Realtek products as used in such applications.

Realtek’s provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation kits, (collectively, “Resources”) are intended to assist designers who are developing applications that incorporate Realtek products; by downloading, accessing or using Realtek’s Resources in any way, Customer (individually or, if Customer is acting on behalf of a company, Customer’s company) agrees to use any particular Realtek Resources solely for this purpose and subject to the terms of this Notice.

Realtek’s provision of Realtek Resources does not expand or otherwise alter Realtek’s applicable published warranties or warranty disclaimers for Realtek’s products, and no additional obligations or liabilities arise from Realtek providing such Realtek Resources. Realtek reserves the right to make corrections, enhancements, improvements and other changes to its Realtek Resources. Realtek has not conducted any testing other than that specifically described in the published documentation for a particular Realtek Resource.

Customer is authorized to use, copy and modify any individual Realtek Resource only in connection with the development of applications that include the Realtek product(s) identified in such Realtek Resource. No other license, express or implied, by estoppel or otherwise to any other Realtek intellectual property right, and no license to any technology or intellectual property right of Realtek or any third party is granted herein, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Realtek products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of Realtek Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from Realtek under the patents or other Realtek’s intellectual property.

Realtek’s Resources are provided “as is” and with all faults. Realtek disclaims all other warranties or representations, express or implied, regarding resources or use thereof, including but not limited to accuracy or completeness, title, any epidemic failure warranty and any implied warranties of merchantability, fitness for a particular purpose, and non-infringement of any third party intellectual property rights.

Realtek shall not be liable for and shall not defend or indemnify Customer against any claim, including but not limited to any infringement claim that related to or is based on any combination of products even if described in Realtek Resources or otherwise. In no event shall Realtek be liable for any actual, direct, special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of Realtek’s Resources or use thereof, and regardless of whether Realtek has been advised of the possibility of such damages. Realtek is not responsible for any failure to meet such industry standard requirements.

Where Realtek specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Customers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may not use any Realtek products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death. Such equipment includes, without limitation, all medical devices identified by the U.S.FDA as Class III devices and equivalent classifications outside the U.S.

Customers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Customers' own risk. Customers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection.

Customer will fully indemnify Realtek and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

USING THIS DOCUMENT

Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide.

1 AmebaZ2 RTL8720CM Board

1.1 AmebaZ2 Demo EVB

Ameba Demo board home page: https://www.amebaiot.com/amebaz2/

Ameba RTL8720CM Board (AMB 31)

Manual / Schematic / Layout

Buy it

CPU

- 32-bit Arm®Cortex®-M4, up to 100MHz

Memory

- 256KB SRAM + 4MB PSRAM

Key Features

- Integrated 802.11n Wi-Fi SoC
- Hardware SSL Engine
- Root Trust Secure Boot
- BLE4.2

1.2 PCB Layout Overview

RTL8720C embedded on Ameba-ZII DEV demo board, which consists of various I/O interfaces. For the details of the HDK, please contact us for further reference.

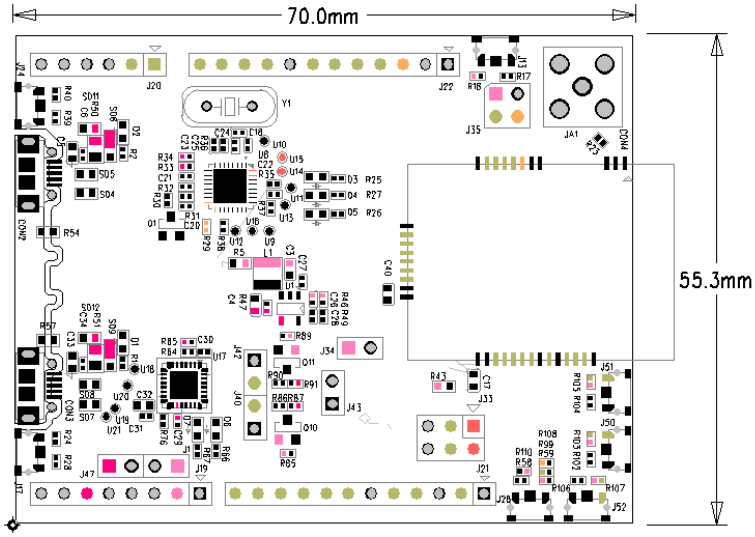


Figure 1-1 Top View of Ameba-ZII 2V0 Dev Board

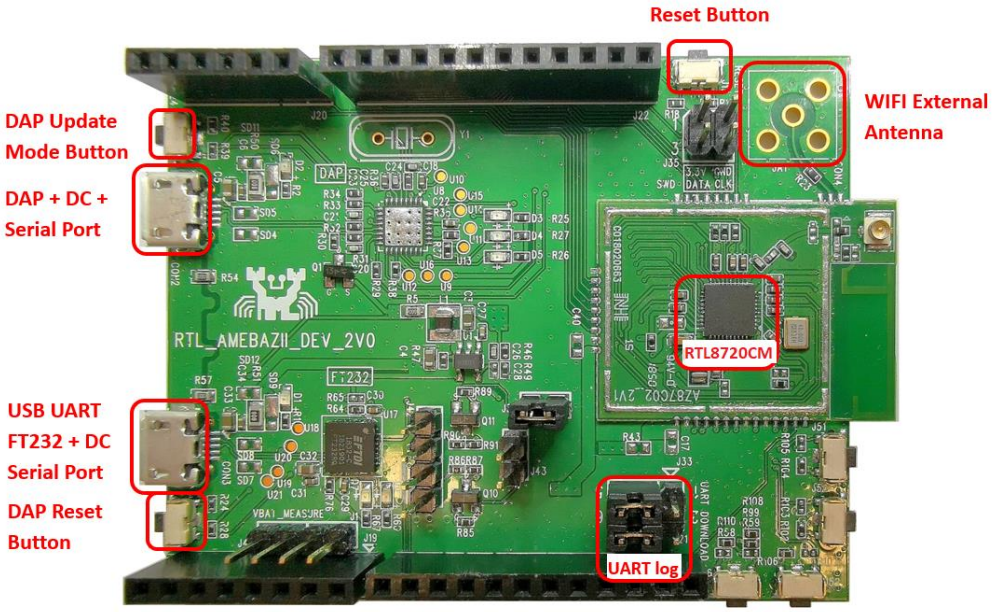
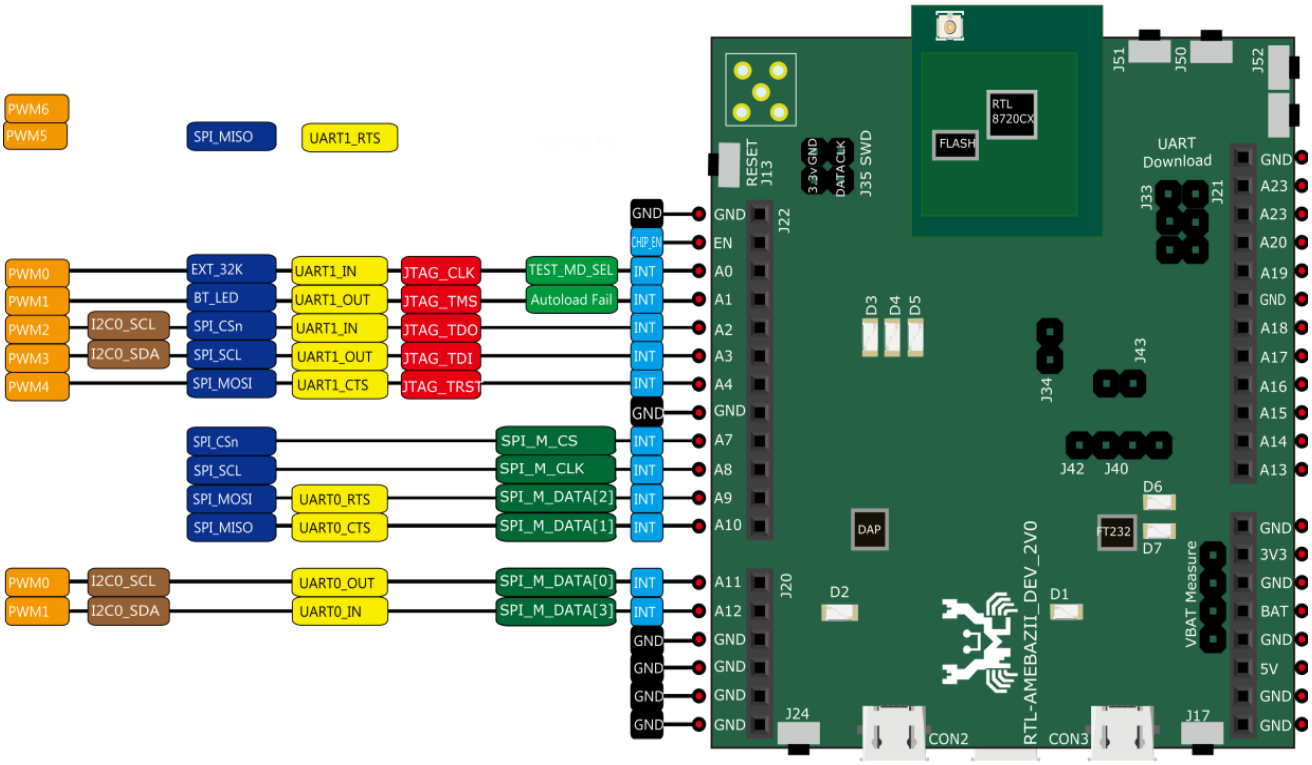


Figure 1-2 Ameba-ZII 2V0 Dev Board PCB Layout

1.3 Pin-Out Reference



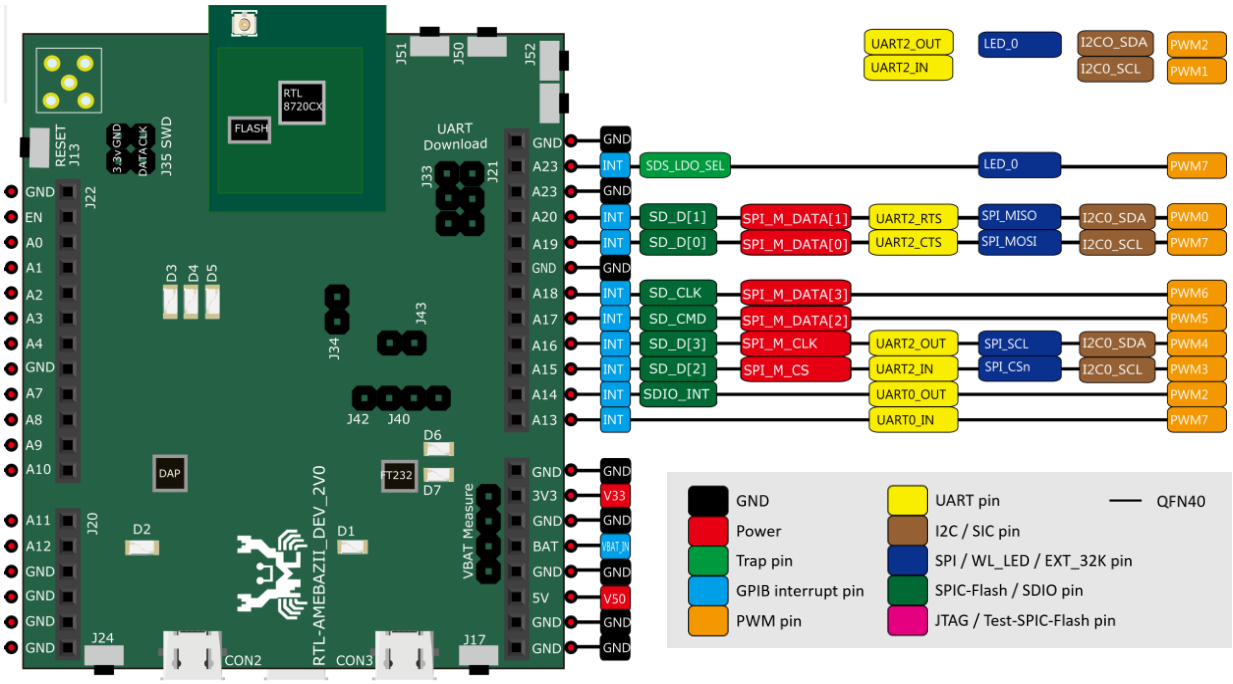
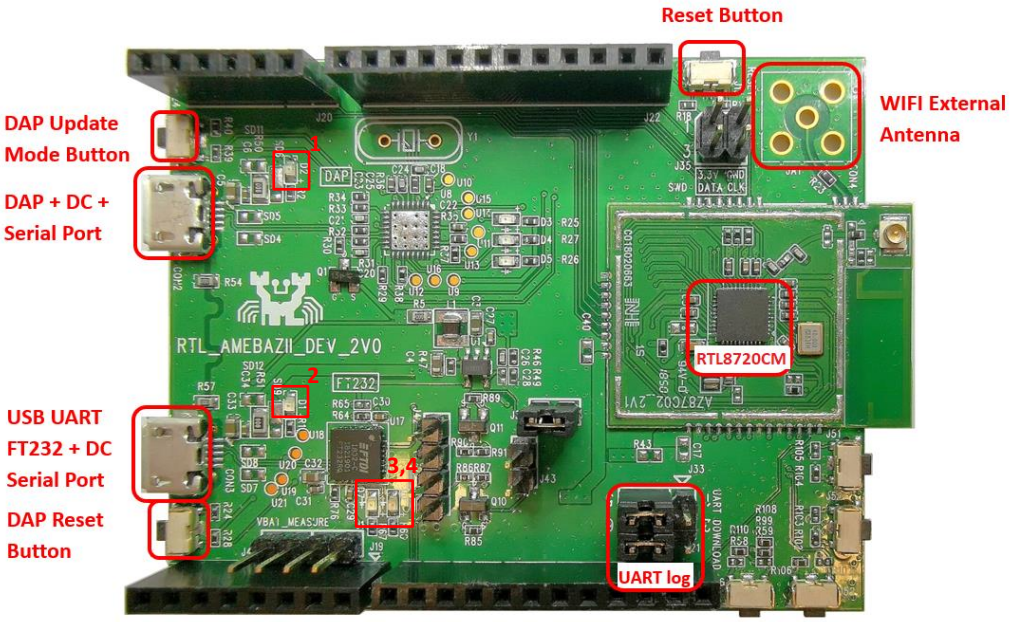


Figure 1-3 Pin Out Reference for DEV_2V0

1.4 LED State

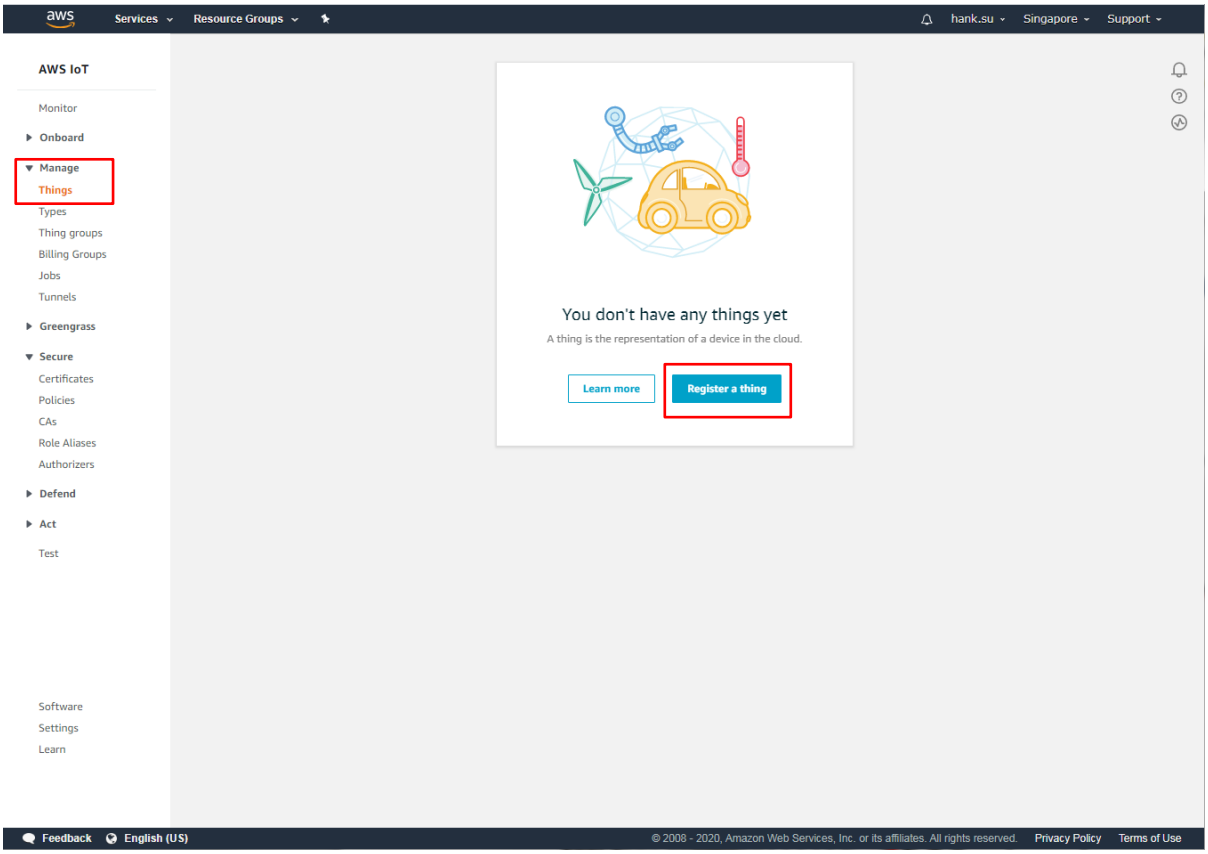
There are four LED on the AmebaZ2 EVB. LED1 lights steady green and LED2 steady red when device have power. LED3 and LED4 go with log uart, they flash red and green when uart communicating.

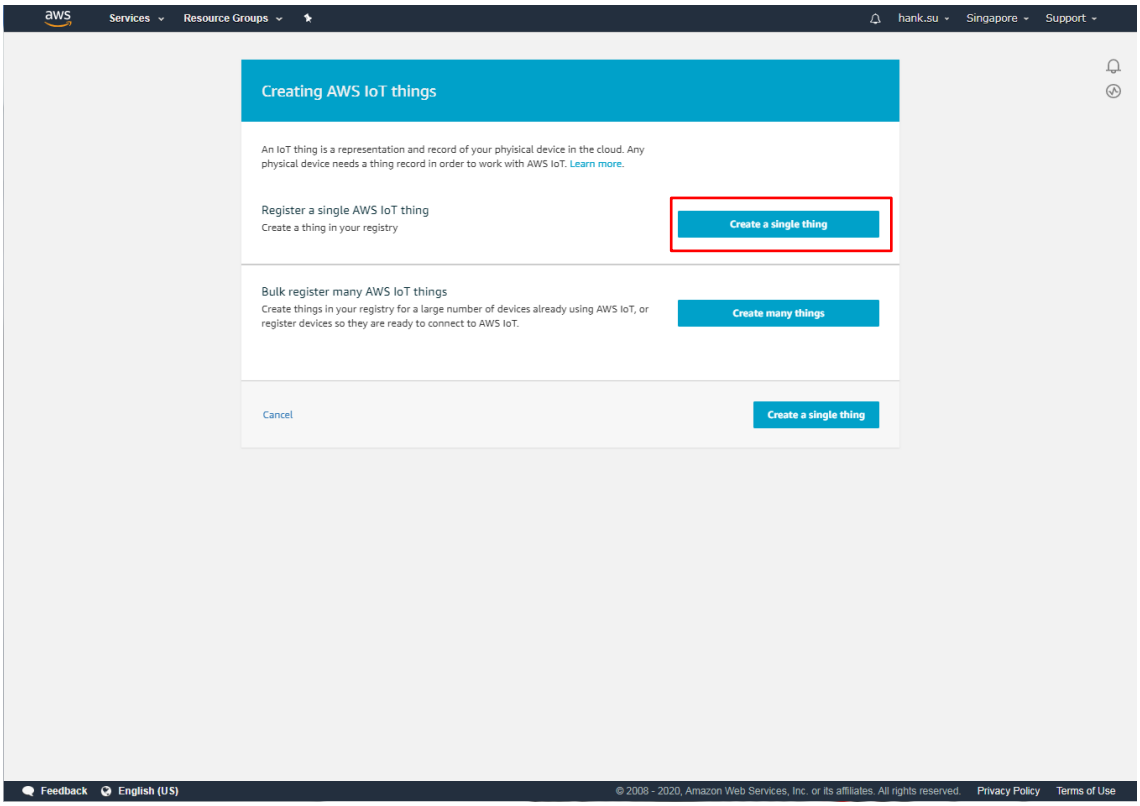


2 Configure AWS IoT Core

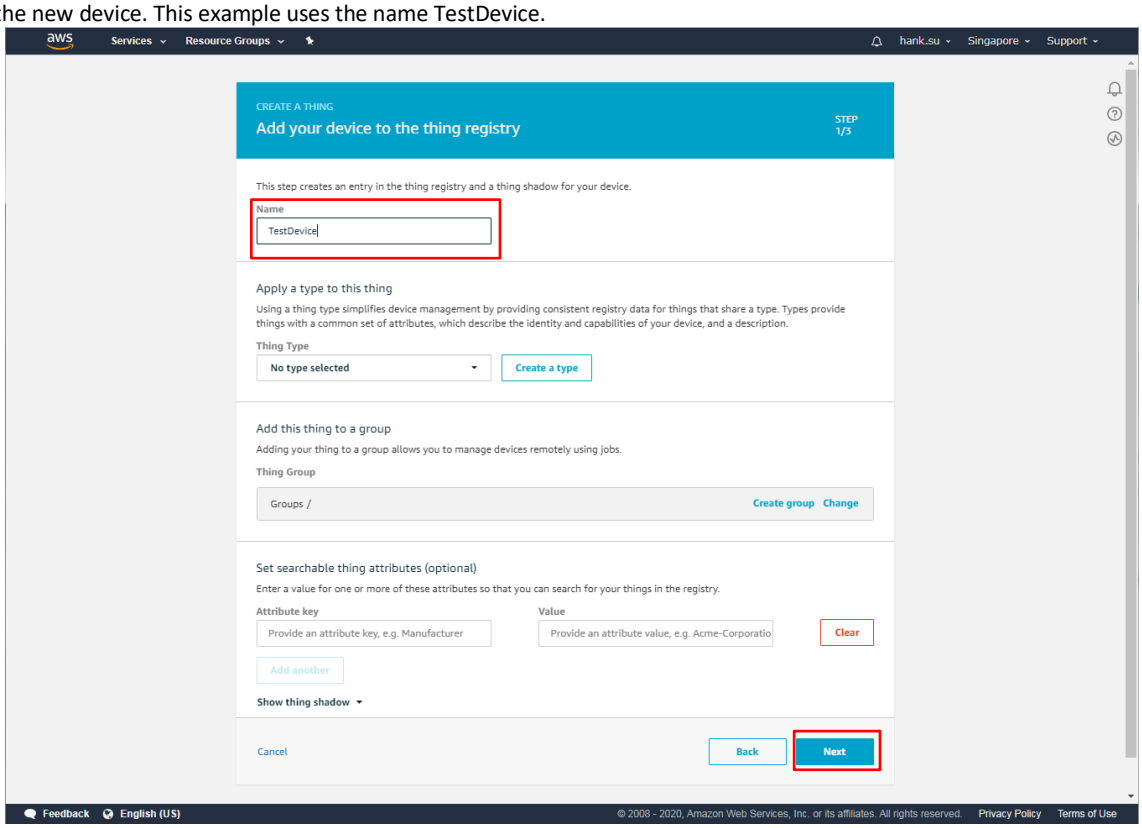
2.1 Create a New Device

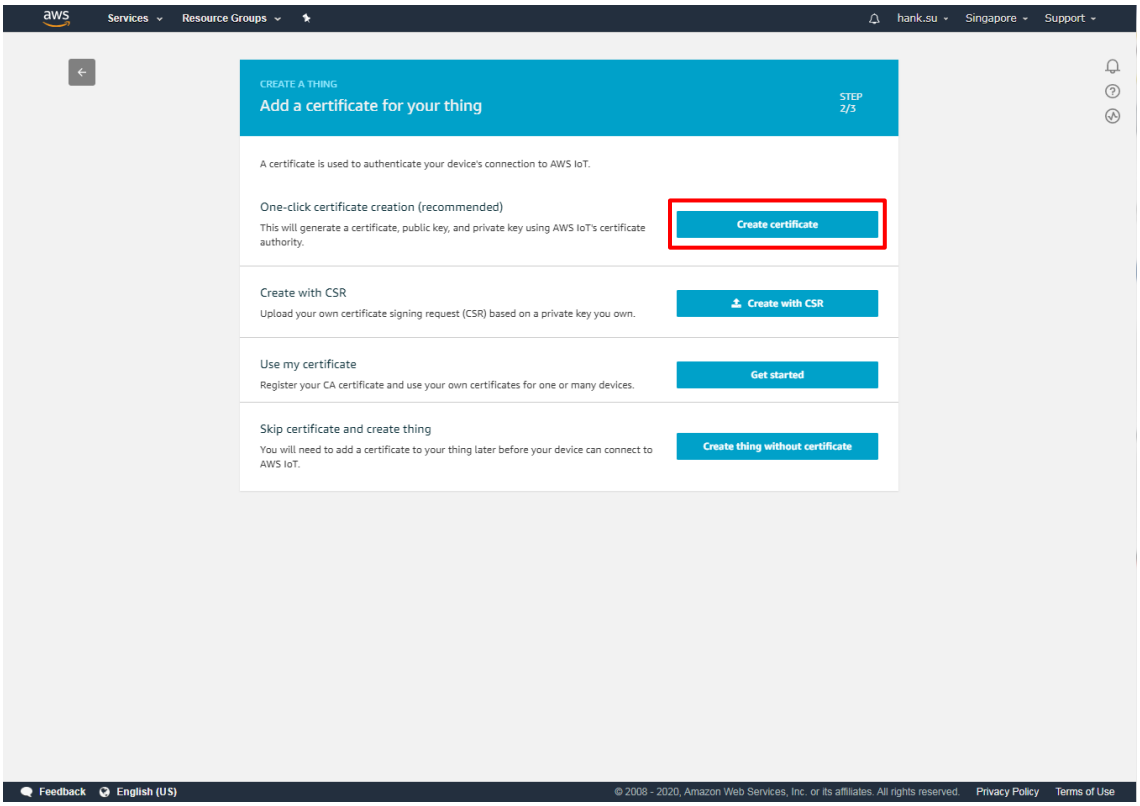
To create a new device, navigate to Manage -> Things in the left-hand navigation menu. Then click “Register a thing”.



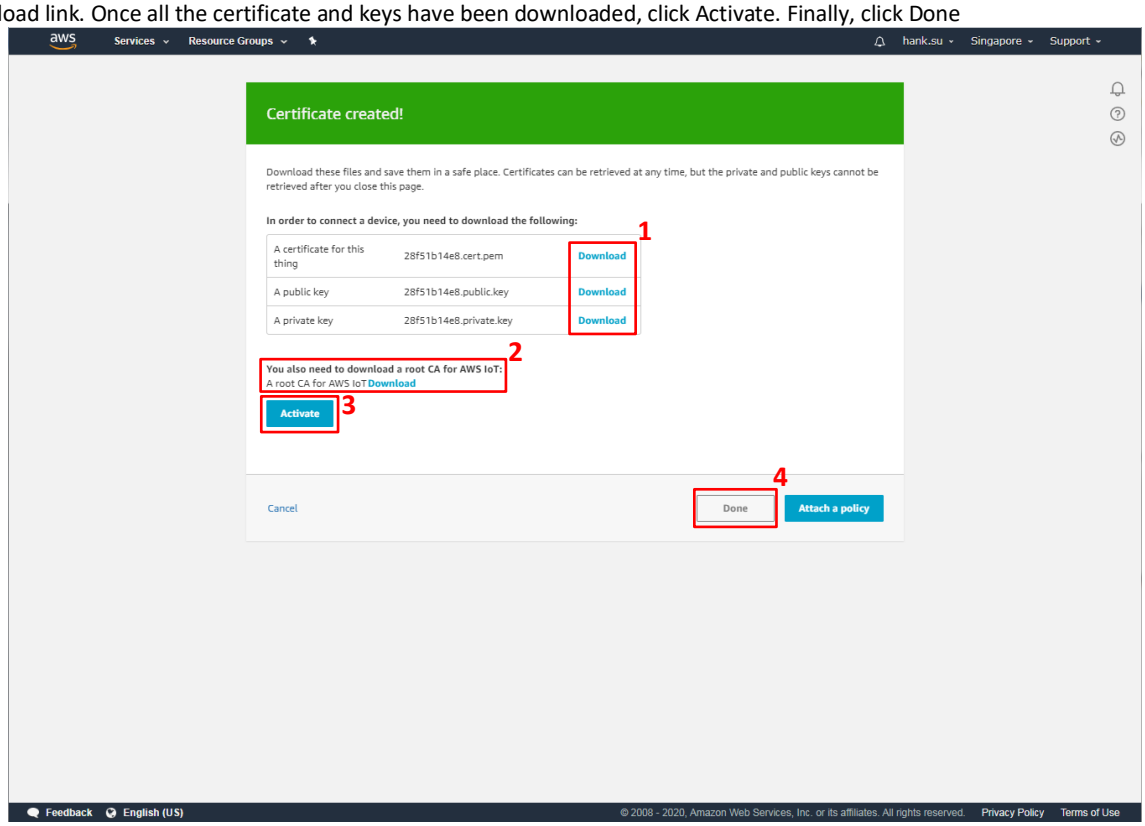


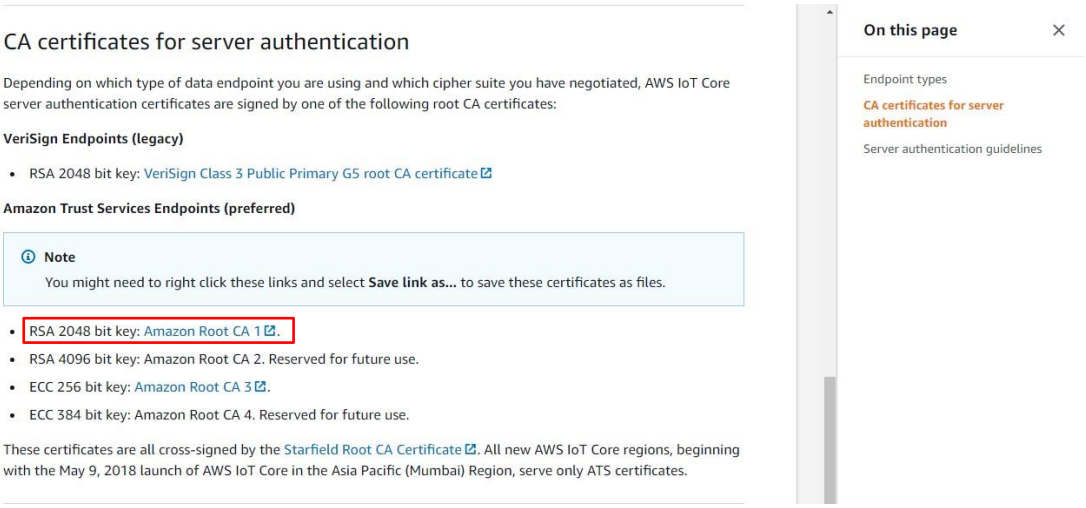
Then, name the new device. This example uses the name TestDevice.





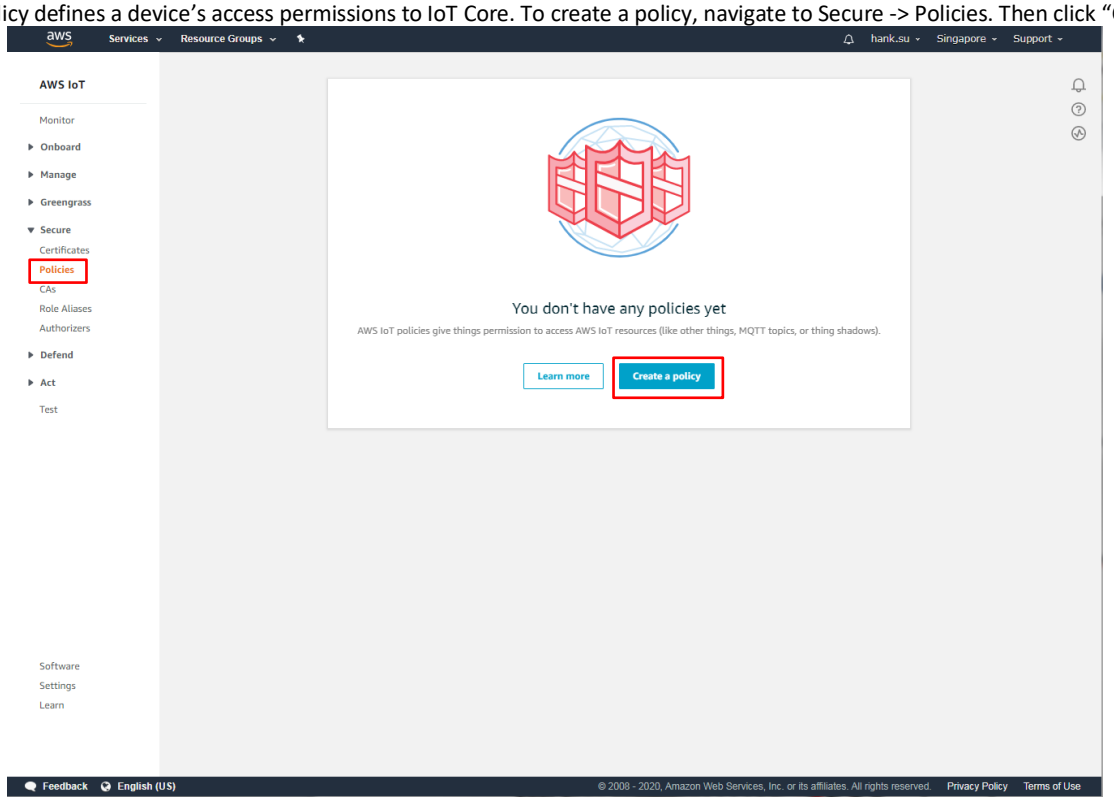
Download the certificate, public key, and private key for the device by clicking Download. Next, download the root CA for AWS IoT by clicking to the Download link. Once all the certificate and keys have been downloaded, click Activate. Finally, click Done



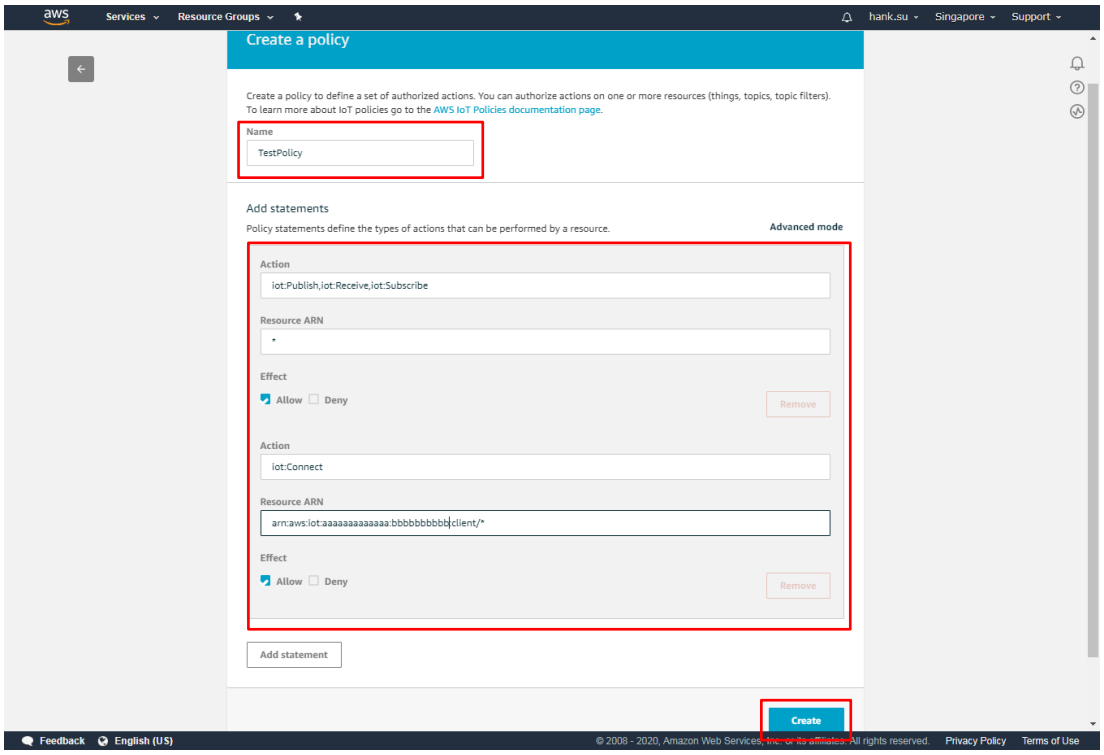


2.2 Create a policy

A policy defines a device's access permissions to IoT Core. To create a policy, navigate to Secure -> Policies. Then click "Create a policy"

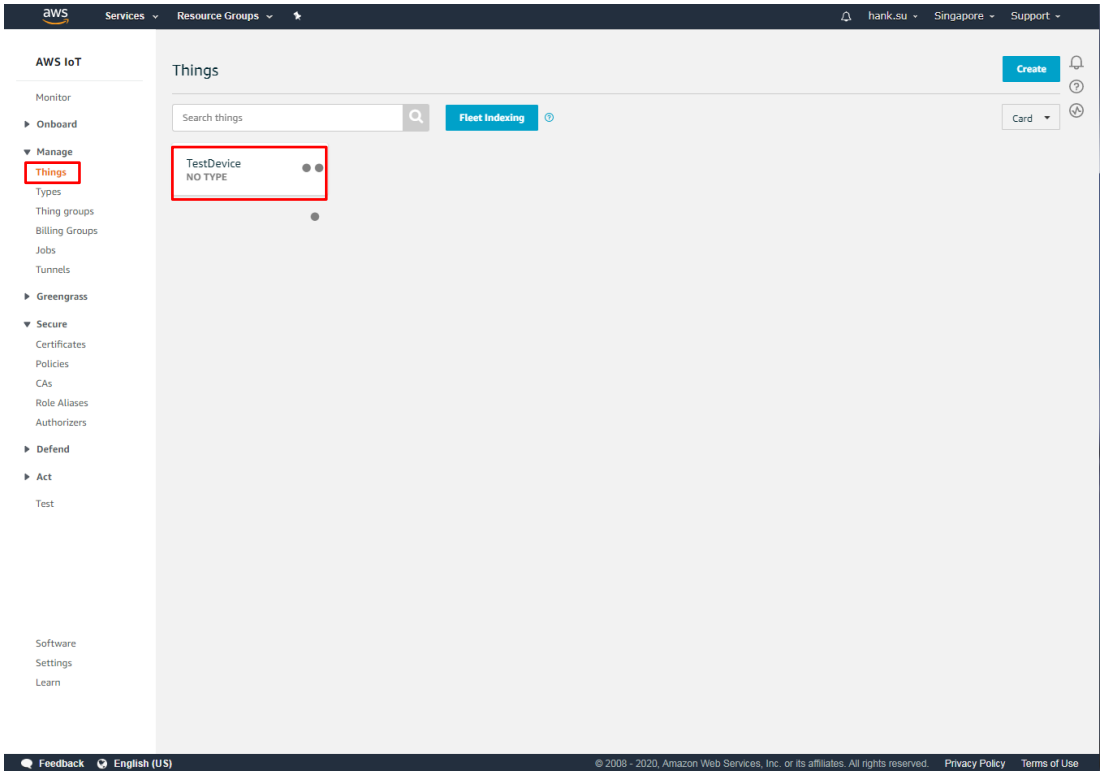


NOTE – this policy grants unrestricted access for all iot operations, and is to be used only in a development environment. For non-dev environments, all devices in your fleet must have credentials with privileges that authorize intended actions only, which include (but not limited to) AWS IoT MQTT actions such as publishing messages or subscribing to topics with specific scope and context. The specific permission policies can vary for your use cases. Identify the permission policies that best meet your business and security requirements.
For sample policies, refer to https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html.
Also refer to https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html

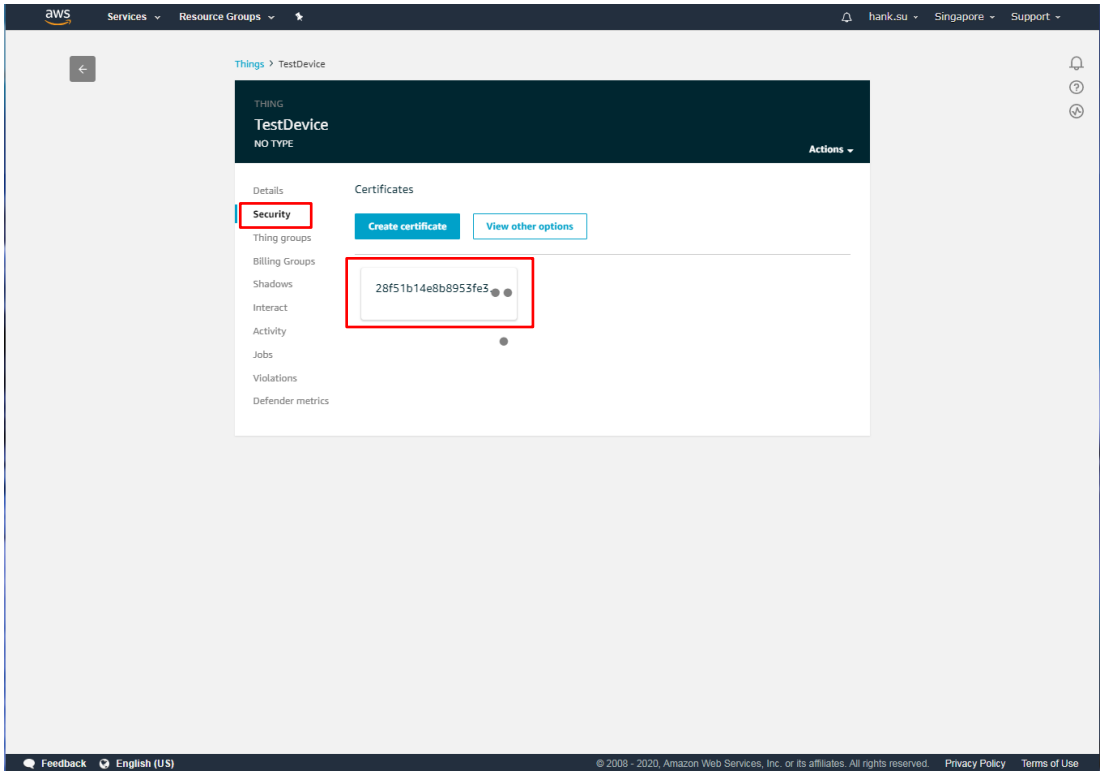


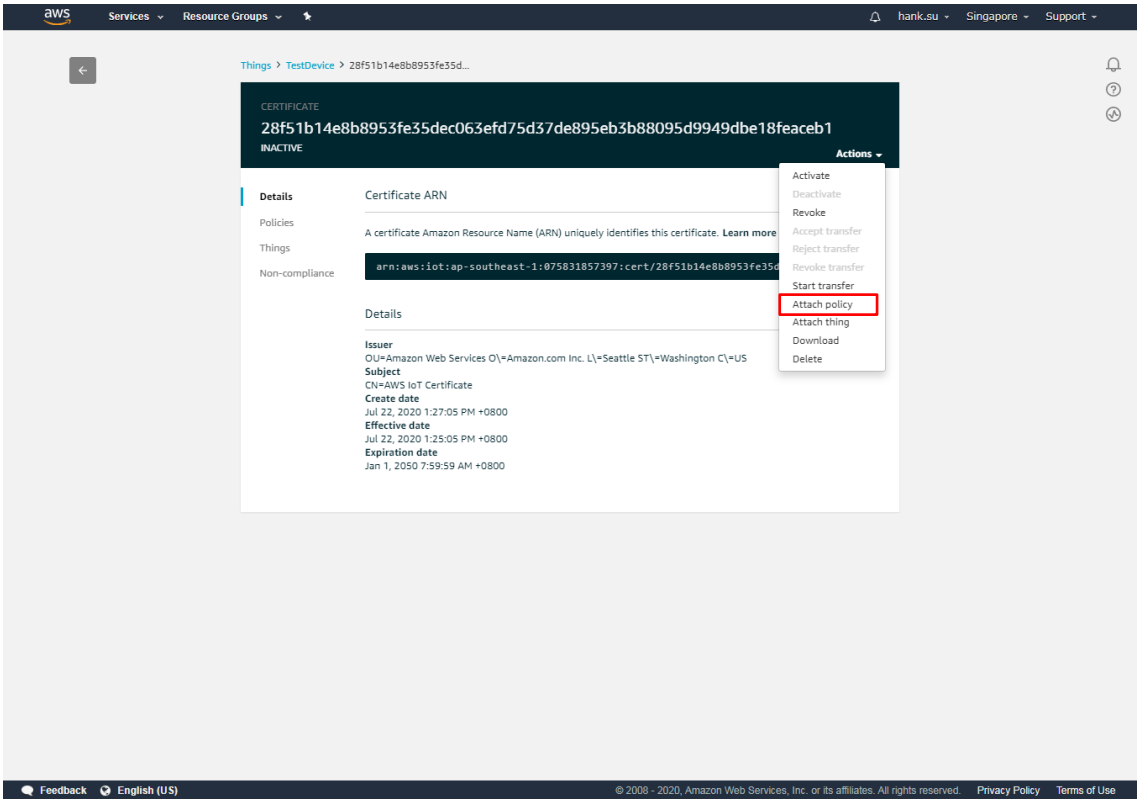
2.3 Attach Policy

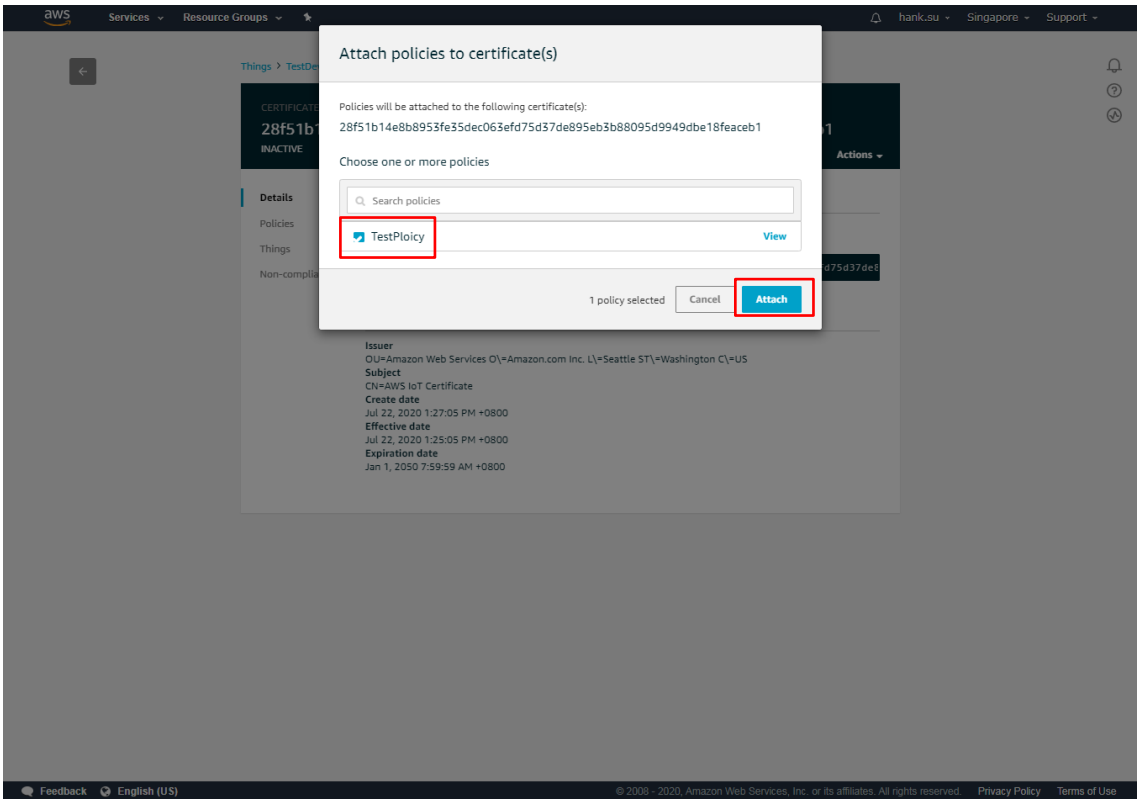
The last step to configuring the device is attaching a policy. To attach a policy to new device, navigate to Manage -> Things. Then click on the device which was created.



Click Security, then click the certificate create in previous step.



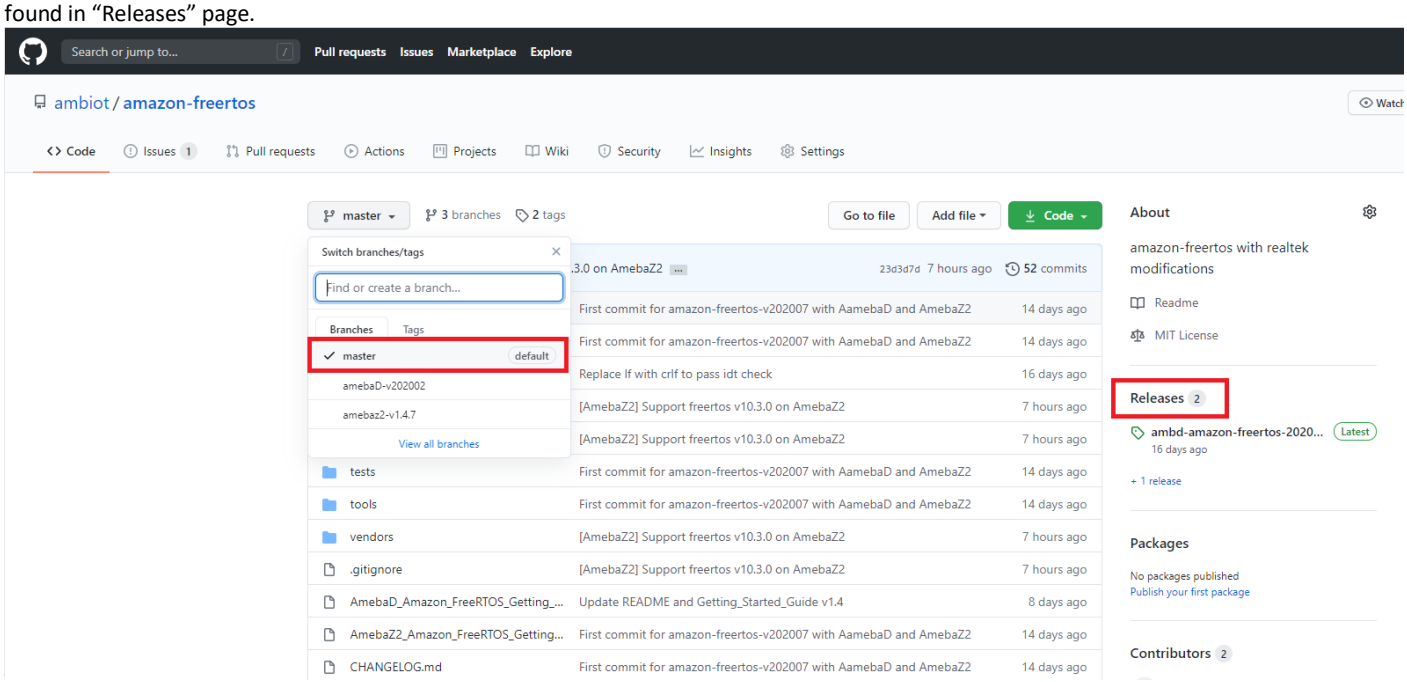




3 Configure AmebaZ2 Amazon FreeRTOS

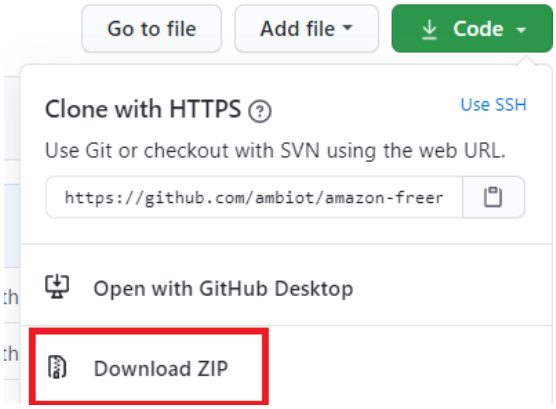
3.1 Download Source Code from github

Open source link: https://github.com/ambiot/amazon-freertos and select master for get newest source code. The stable version could be found in “Releases” page.



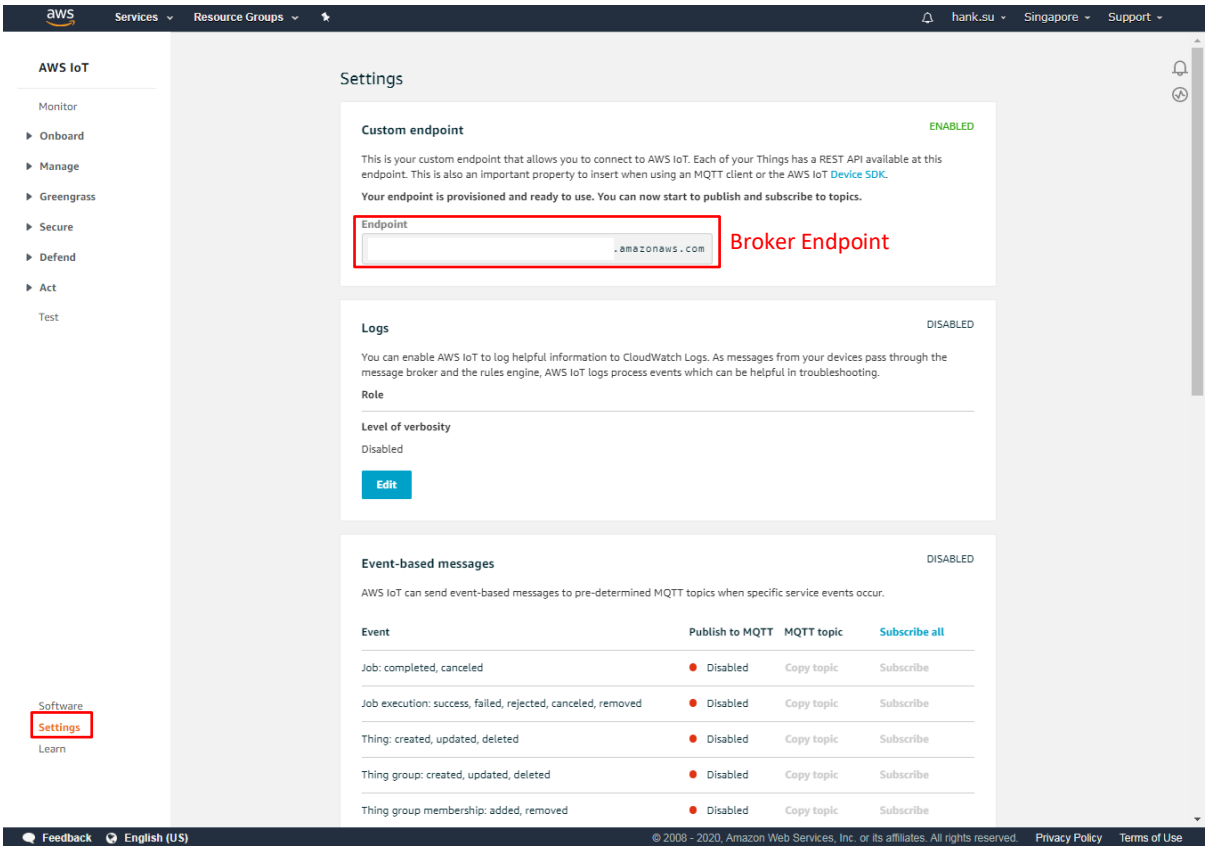
3.1.1 Cloning a repository by Download ZIP

1. On GitHub, navigate to the main page of the repository.
2. Above the list of files, click **Code**.
3. Click **Donwload ZIP** to get source code.

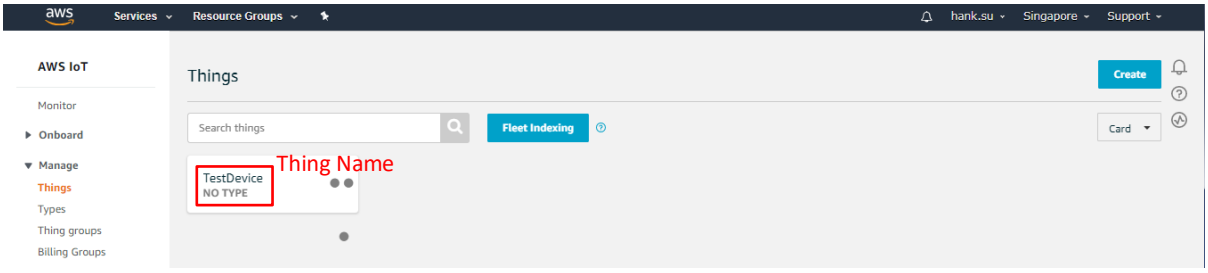


For more information, please refer "Cloning a repository from GitHub to GitHub Desktop."

3.2 Get Broker Endpoint by AWS IoT Core



3.3 Get Thing Name



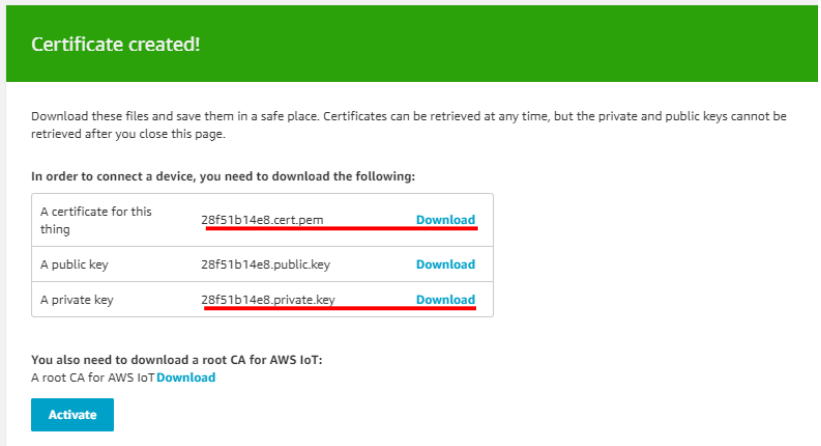
3.4 Setup IoT Core Information with AmebaZ2 Amazon FreeRTOS

Setup BROKER_ENDPOINT, THING_NAME, WIFI_SSID, PASSWORD in “amazon-freertos/blob/master/demos/include/aws_clientcredential.h”

```
#define clientcredentialMQTT_BROKER_ENDPOINT         "xxxxxxxxxxxxxxx.amazonaws.com"

/*
 * @brief Host name.
 *
 * @todo Set this to the unique name of your IoT Thing.
 */
#define clientcredentialIOT_THING_NAME               "TestDevice"

/*
 * @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.
 *
 * @todo If you are using Wi-Fi, set this to your network name.
 */
#define clientcredentialWIFI_SSID                    "TestAP"

/*
 * @brief Password needed to join Wi-Fi network.
 * @todo If you are using WPA, set this to your network password.
 */
#define clientcredentialWIFI_PASSWORD                "password"

/*
 * @brief Wi-Fi network security type.
 *
 * @see WIFISecurity_t.
 *
 * @note Possible values are eWiFiSecurityOpen, eWiFiSecurityWEP, eWiFiSecurityWPA,
 * eWiFiSecurityWPA2 (depending on the support of your device Wi-Fi radio).
 */
#define clientcredentialWIFI_SECURITY                eWiFiSecurityWPA2

#endif /* ifndef __AWS_CLIENTCREDENTIAL__H__ */
```

3.4.1 Setup Thing’s Private Key and Certificate

Filled keyCLIENT_CERTIFICATE_PEM and keyCLIENT_PRIVATE_KEY_PEM in “amazon-freertos/blob/master/demos/include/aws_clientcredential_keys.h” by xxxxxxxx-certifiacte.pem and xxxxxxxx-private.pem.key.



It can done by amazon-freertos/tools/certificate_configuration/CertificateConfigurator.html

Final aws_clientcredential_keys.h overview.

```
/*
 * @brief PEM-encoded client certificate.
 *
 * @todo If you are running one of the FreeRTOS demo projects, set this
 * to the certificate that will be used for TLS client authentication.
 *
 * @note Must include the PEM header and footer:
 * "-----BEGIN CERTIFICATE-----\n"\
 * "...base64 data...\n"\
 * "-----END CERTIFICATE-----\n"
 */
#define keyCLIENT_CERTIFICATE_PEM \
"-----BEGIN CERTIFICATE-----\n"\
"MIIDWjCCAkKgAwIBAgIVAIDLSSoG+EARSbBprT4Im1uu8j2vMA0GCSqGSIb3DQEB\n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"k5+NsBroU/YdvOUmzKn6XfI4nX4hLQJ2TbhAT8aq1ounGk6ZGqCbxt4mg5bB0w==\n"\
"-----END CERTIFICATE-----"
```

```
/*
 * @brief PEM-encoded client private key.
 *
 * @todo If you are running one of the FreeRTOS demo projects, set this
 * to the private key that will be used for TLS client authentication.
 *
 * @note Must include the PEM header and footer:
 * "-----BEGIN RSA PRIVATE KEY-----\n"\
 * "...base64 data...\n"\
 * "-----END RSA PRIVATE KEY-----\n"
 */
#define keyCLIENT_PRIVATE_KEY_PEM \
"-----BEGIN RSA PRIVATE KEY-----\n"\
"MIIEpAIBAAKCAQEAwop96WNucGebARFjD8O+CLsqcBNn/AHyhEcozLZC8qoECUOn\n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"                                                                  \n"\
"p0WEuLUuz2FAv1noAbN/6OQ8H/PT0AFJT/ghA04GnIUF0kjSzY60ehS2mVp6neP+\n"\
"AZjzZ6QJYlb5/PFz9oES448kpyaAoS2ke86+R4r4YOMBK+I5RVbfSQ==\n"\
"-----END RSA PRIVATE KEY-----\n"
```

3.4.2 Enable FreeRTOS demo on AmebaZ2

Find platform_opts.h in amazon-freertos\vendors\realtek\boards\amebaZ2\aws_demos\config_files and enable **CONFIG_EXAMPLE_AMAZON_FREERTOS**

```
/* For Amazon FreeRTOS SDK example */
#define CONFIG_EXAMPLE_AMAZON_FREERTOS   1
```

Fine aws_demo_config.h in amazon-freertos\vendors\realtek\boards\amebaZ2\aws_demos\config_files and add **CONFIG_MQTT_DEMO_ENABLED**

```
/* To run a particular demo you need to define one of these.
 * Only one demo can be configured at a time
 *
 *          CONFIG_MQTT_DEMO_ENABLED
 *          CONFIG_SHADOW_DEMO_ENABLED
 *          CONFIG_OTA_UPDATE_DEMO_ENABLED
 *
 *  These defines are used in iot_demo_runner.h for demo selection */

#define CONFIG_MQTT_DEMO_ENABLED
```

Now you can start to compile AmebaZ2 Amazon FreeRTOS

4 Compile AmebaZ2 Amazon FreeRTOS

4.1 Pre-Requisite

- Required source code. (https://github.com/ambiot/amazon-freertos)
- AmebaZ2 Demo board
- Realtek Image Tool
- IAR Embedded Workbench ver.8.30.1

4.2 IAR Build Environment Setup

The IAR IDE (integrated development environment) only supports Windows OS, this section is applicable for **Windows OS only**.

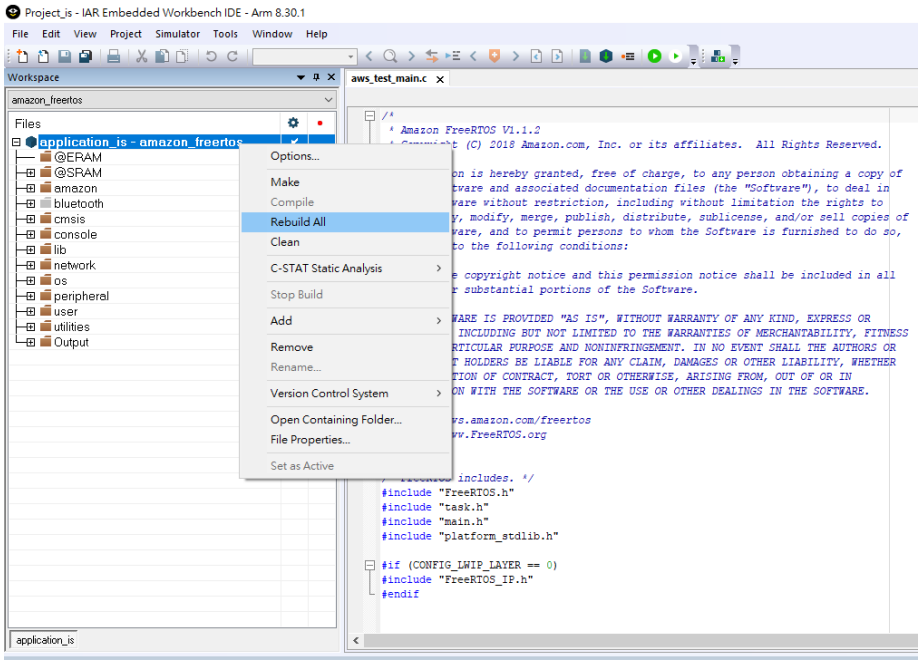
4.3 Install IAR IDE

IAR IDE provides the toolchain for Ameba-ZII. It allows users to write programs, compile and upload them to your board. Also, it supports step-by-step debug function.
User can visit the official website of **IAR Embedded Workbench** and install the IDE by following its instructions.

Note: Please use IAR version **8.30** or above.

4.4 Compilation

1) Open amazon-freertos/projects/realtek/amebaZ2/IAR/aws_demos/Project_is.eww.
2) Confirm 'application_is' in Work Space, right click 'application_is' and choose "**Rebuild All**" to compile.
3) Make sure there is no error after compile.



4.5 Generate Image Binary

After compile, the images **partition.bin**, **bootloader.bin**, **firmware_is.bin** and **flash_is.bin** can be seen in the amazon-freertos/projects/realtek/amebaZ2/IAR/aws_demos/Debug/Exe.

1) **partition.bin** stores partition table, recording the address of Boot image and firmware image;
2) **bootloader.bin** is bootloader image;
3) **firmware_is.bin** is application image;
4) flash_is.bin links partition.bin, bootloader.bin and firmware_is.bin. Users need to choose flash_is.bin when downloading the image to board by Image Tool

5 ImageTool

The tool can be find in amazon-freertos/vendors/realtek/tools/AmebaZ2_PGTool_v1.2.8

5.1 Introduction

This chapter introduces how to use Image Tool to generate and download images. As show in picture below, Image Tool has two menu pages:

- Download: used as image download server to transmit images to Ameba through UART.

Note: If you need to download code via external uart, must use FT232 USB to connect UART dongle.

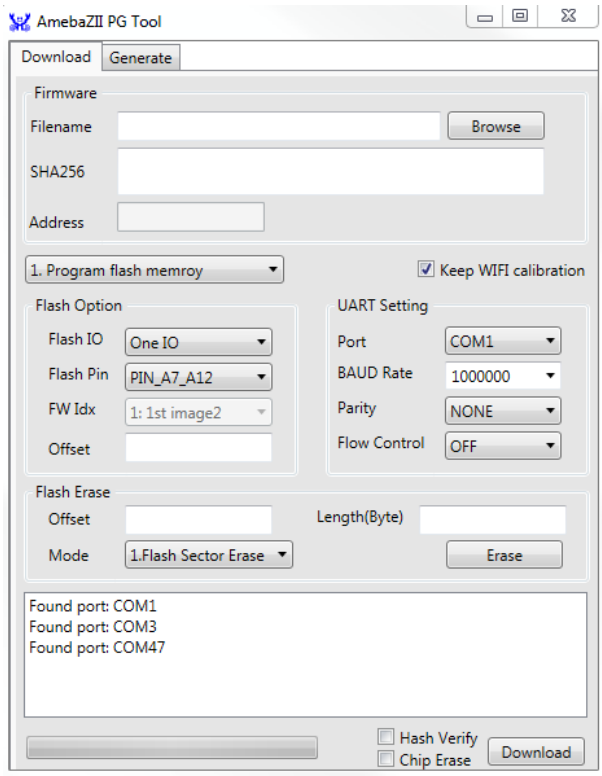


Fig 5-1 AmebaZII ImageTool UI

5.2 Environment Setup

5.2.1 Hardware Setup

User needs to connect CON3 to user's PC via a Micro USB cable. Add jumpers for J34 and J33 (J33 is for log UART which has two jumpers) if there is no connection.

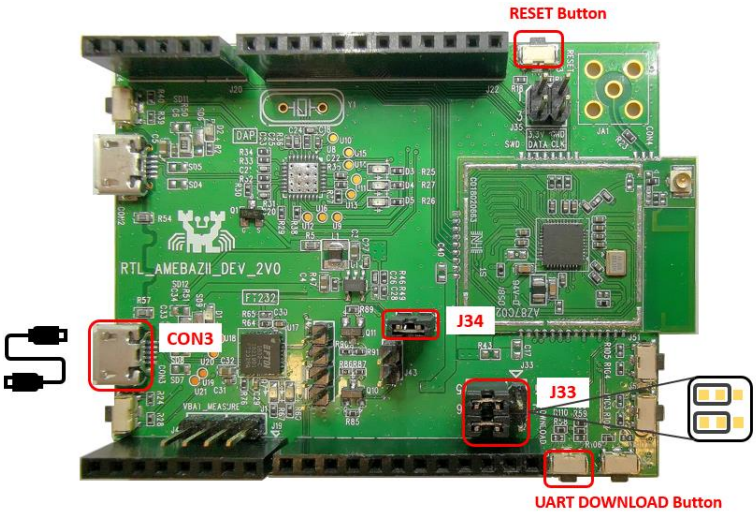


Fig 5-2 Ameba-ZII EVB V2.0 Hardware Setup

5.2.2 Software Setup

- Environment Requirements: EX. WinXP, Win 7 Above, Microsoft .NET Framework 3.5
- AmebaZII_PGTool_v1.2.8.exe

5.3 Image Download

User can download the image to demo board by following steps:

1) Trigger Ameba-ZII chip enter **UART download mode** by:

a. Press and hold the **UART DOWNLOAD** button then press the **RESET** button and release both buttons. And make sure the log UART is connected properly.

b. If the chip enters **download mode**, the below log should be shown on log UART console.

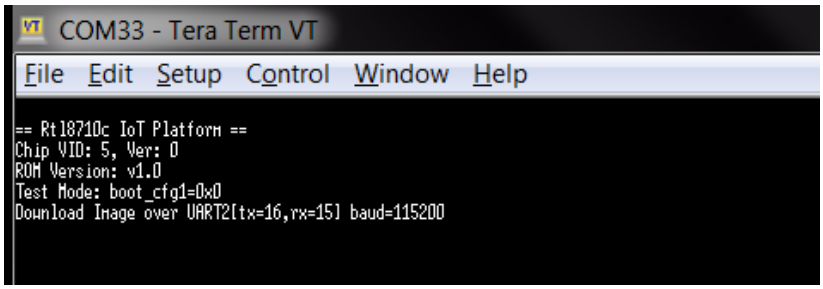
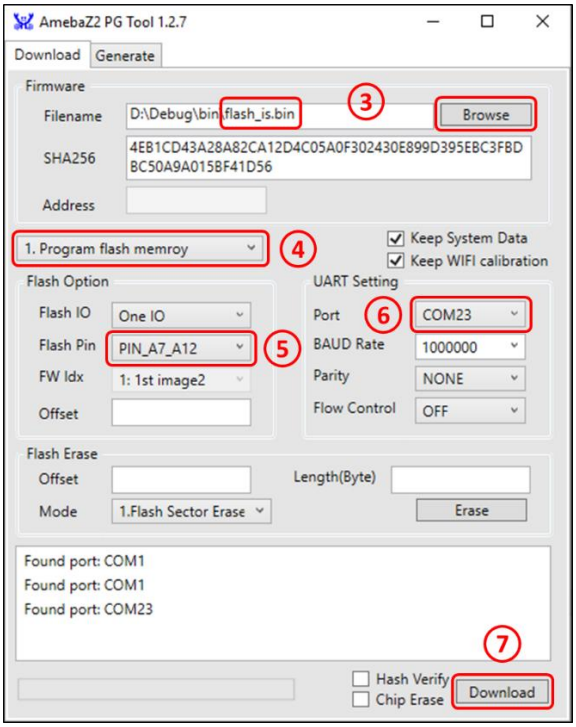


Fig 5-3 Ameba-ZII UART download mode

c. After confirming it is in download mode, **remember to disconnect the log UART console before using Image Tool to download**, because the tool will also need to connect to this log UART port.

2) Open **AmebaZ2 PG Tool**



3) "**Browse**" to choose the image to be downloaded (amazon-freertos/projects/realtek/amebaZ2/IAR/aws_demos/Debug/Exe /flash_is.bin)
4) Choose "**1. Program flash memory**"
5) Choose correct "**Flash Pin**" according to the IC part number

Flash Pin	**IC part number**
PIN_A7_A12	RTL8710CX/RTL8720CM
PIN_B6_B12	RTL8720CF

6) Choose the correct **UART port** (use **rescan** to update the port list)
7) Click "**Download**" to start downloading image. While downloading, the status will be shown on the left bar.

Note: It's recommended to use the default settings unless user is familiar with them.

6 MQTT Demo

6.1 Get Device Log

Install Tera Term to get device log

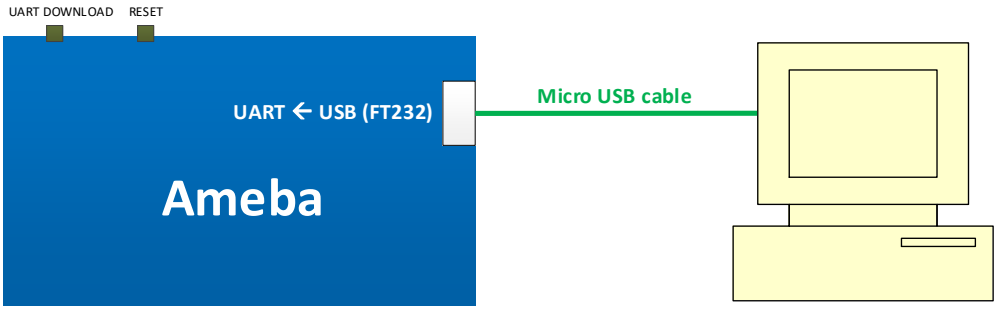
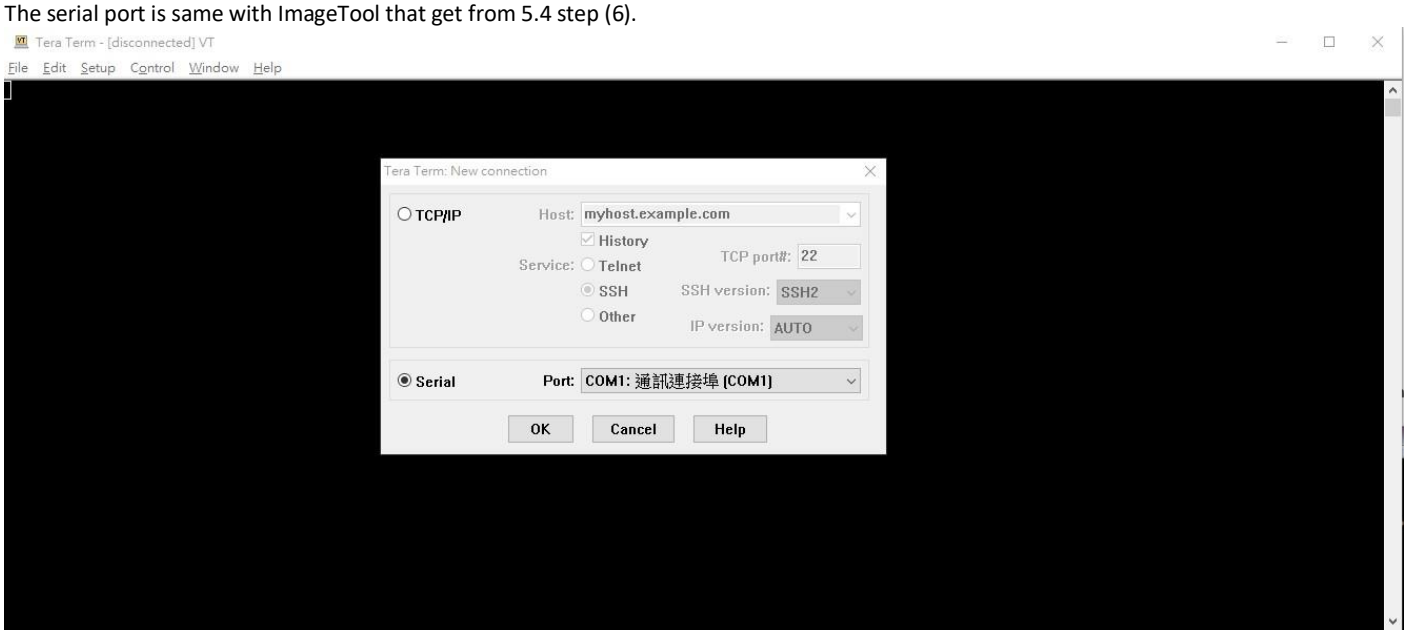


Fig 6-1 Hardware setup

The serial port is same with ImageTool that get from 5.4 step (6).



6.2 Run MQTT Demo

Default setting of SDK are enable MQTT demo. Once the AmebaD EVB has rebooted, the application will automatically start run MQTT demo and communicate to IoT Core.

```
COM6 - Tera Term VT
File  Edit  Setup  Control  Window  Help
#calibration_ok:[2:19:11]
#interface 0 is initialized
interface 1 is initialized

Initializing WIFI ...
WIFI is not running
WIFI initialized

init_thread(58), Available heap 0x24ac0
0 56 [example_a] Wi-Fi module initialized. Connecting to AP...
WIFI is already running
Joining BSS by SSID RealEZ-2.4G...

RTL8721D[Driver]: set ssid [RealEZ-2.4G]

RTL8721D[Driver]: rtw_set_wpa_ie[1136]: AuthKeyMgmt = 0x2

RTL8721D[Driver]: rtw_restruct_sec_ie[3763]: no pmksa cached

RTL8721D[Driver]: start auth to 80:2a:a8:d4:93:c4

RTL8721D[Driver]: auth alg = 2

RTL8721D[Driver]:
OnAuthClient:algthm = 0, seq = 2, status = 0, sae_msg_len = 0

RTL8721D[Driver]: auth success, start assoc

RTL8721D[Driver]: association success(res=4)
wlan1: 1 DL RSVD page success! DLBcnCount:01, poll:00000001

RTL8721D[Driver]: ClientSendEAPOL[1522]: no use cache pmksa

RTL8721D[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4)

RTL8721D[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:2

1 8000 [example_a] Wi-Fi Connected to AP. Creating tasks which use network...
2 8007 [example_a] IP Address acquired 192.168.89.151
3 8019 [example_a] Write certificate...
4 8080 [iot_threa] [INFO ][DEMO][8079] ---------STARTING DEMO---------

5 8086 [iot_threa] [INFO ][INIT][8086] SDK successfully initialized.
```

…

```
6 15504 [iot_threa] [INFO ][DEMO][15504] Successfully initialized the demo. Network type for the demo: 1
7 15513 [iot_threa] [INFO ][MQTT][15513] MQTT library successfully initialized.
8 15522 [iot_threa] [INFO ][DEMO][15522] MQTT demo client identifier is ameba-ota (length 9).
9 17272 [iot_threa] [INFO ][MQTT][17272] Establishing new MQTT connection.
Interface 0 IP address : 192.168.89.15110 17283 [iot_threa] [INFO ][MQTT][17283] Anonymous metrics (SDK language, SDK version) will be provided to AWS IoT. Recompil
e with AWS_IOT_MQTT_ENABLE_METRICS set to 0 to disable.
11 17302 [iot_threa] [INFO ][MQTT][17302] (MQTT connection 100337e0, CONNECT operation 100339a0) Waiting for operation completion.
12 17421 [iot_threa] [INFO ][MQTT][17421] (MQTT connection 100337e0, CONNECT operation 100339a0) Wait complete with result SUCCESS.
13 17433 [iot_threa] [INFO ][MQTT][17433] New MQTT connection 100381d0 established.
14 17443 [iot_threa] [INFO ][MQTT][17443] (MQTT connection 100337e0) SUBSCRIBE operation scheduled.
15 17452 [iot_threa] [INFO ][MQTT][17452] (MQTT connection 100337e0, SUBSCRIBE operation 100339e0) Waiting for operation completion.
16 17612 [iot_threa] [INFO ][MQTT][17612] (MQTT connection 100337e0, SUBSCRIBE operation 100339e0) Wait complete with result SUCCESS.
17 17624 [iot_threa] [INFO ][DEMO][17624] All demo topic filter subscriptions accepted.
18 17632 [iot_threa] [INFO ][DEMO][17632] Publishing messages 0 to 1.
19 17640 [iot_threa] [INFO ][MQTT][17640] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
20 17650 [iot_threa] [INFO ][MQTT][17650] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
21 17659 [iot_threa] [INFO ][DEMO][17659] Waiting for 2 publishes to be received.
22 17752 [iot_threa] [INFO ][DEMO][17752] MQTT PUBLISH 0 successfully sent.
23 17784 [iot_threa] [INFO ][DEMO][17784] Incoming PUBLISH received:
Subscription topic filter: iotdemo/topic/1
Publish topic name: iotdemo/topic/1
Publish retain flag: 0
Publish QoS: 1
Publish payload: Hello world 0!
24 17804 [iot_threa] [INFO ][MQTT][17804] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
25 17814 [iot_threa] [INFO ][DEMO][17814] Acknowledgment message for PUBLISH 0 will be sent.
26 17825 [iot_threa] [INFO ][DEMO][17825] MQTT PUBLISH 1 successfully sent.
27 17841 [iot_threa] [INFO ][DEMO][17840] Incoming PUBLISH received:
Subscription topic filter: iotdemo/topic/2
Publish topic name: iotdemo/topic/2
Publish retain flag: 0
Publish QoS: 1
Publish payload: Hello world 1!
28 17861 [iot_threa] [INFO ][MQTT][17861] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
29 17870 [iot_threa] [INFO ][DEMO][17870] Acknowledgment message for PUBLISH 1 will be sent.
30 17883 [iot_threa] [INFO ][DEMO][17883] 2 publishes received.
31 17889 [iot_threa] [INFO ][DEMO][17889] Publishing messages 2 to 3.
32 17897 [iot_threa] [INFO ][MQTT][17897] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
33 17907 [iot_threa] [INFO ][MQTT][17907] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
34 17916 [iot_threa] [INFO ][DEMO][17916] Waiting for 2 publishes to be received.
35 18021 [iot_threa] [INFO ][DEMO][18021] MQTT PUBLISH 3 successfully sent.
36 18030 [iot_threa] [INFO ][DEMO][18029] MQTT PUBLISH 2 successfully sent.
37 18039 [iot_threa] [INFO ][DEMO][18038] Incoming PUBLISH received:
Subscription topic filter: iotdemo/topic/4
Publish topic name: iotdemo/topic/4
```

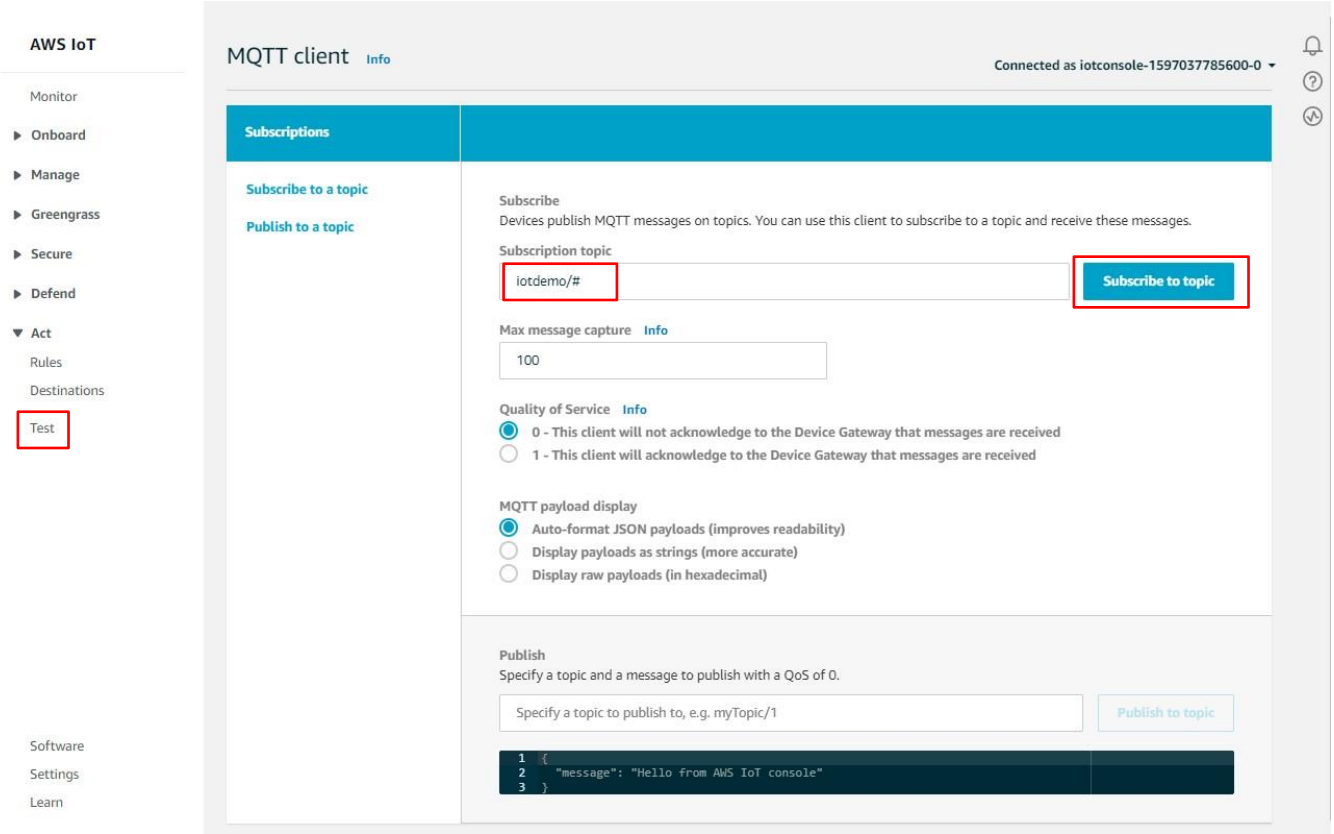
…

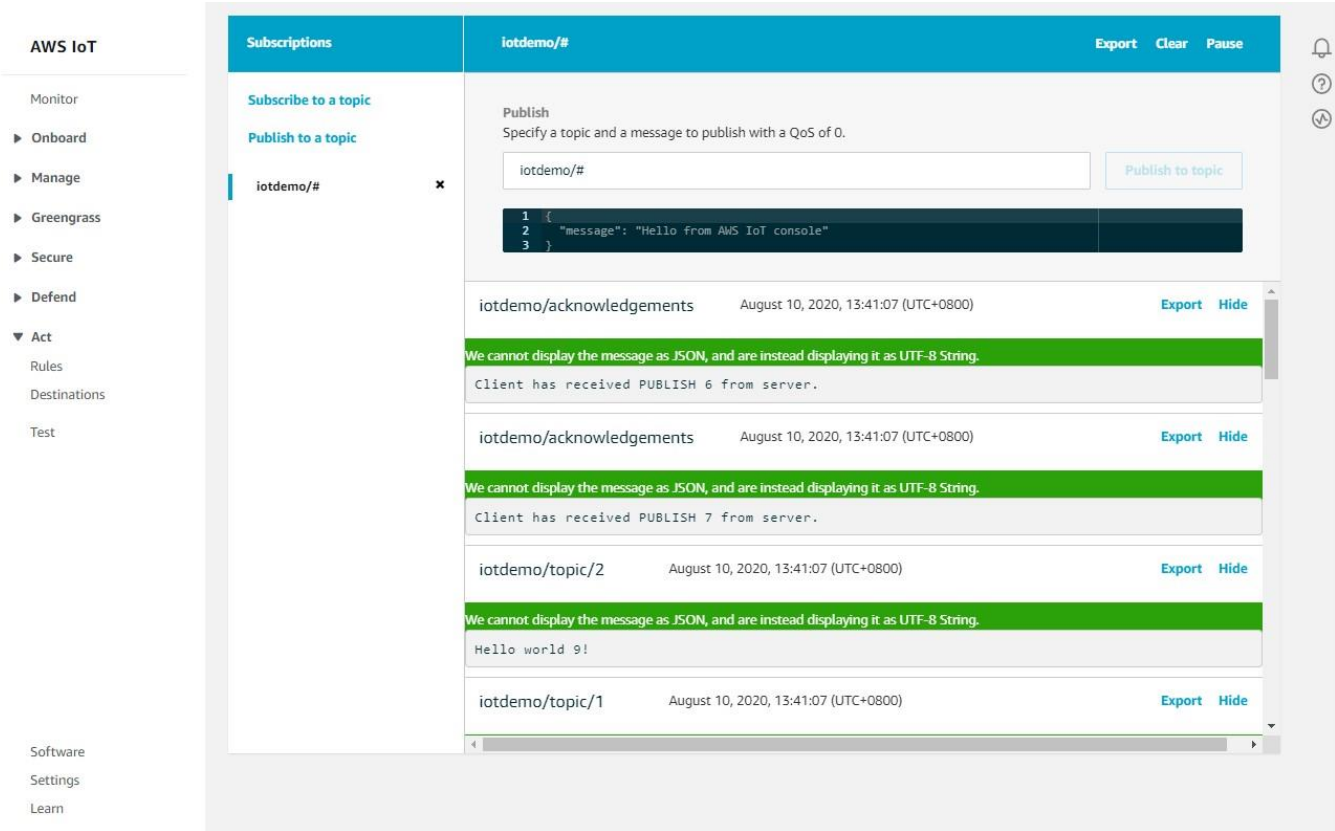
```
Publish payload: Hello world 16!
132 19827 [iot_threa] [INFO ][MQTT][19827] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
133 19837 [iot_threa] [INFO ][DEMO][19837] Acknowledgment message for PUBLISH 16 will be sent.
134 19851 [iot_threa] [INFO ][DEMO][19851] 2 publishes received.
135 19857 [iot_threa] [INFO ][DEMO][19857] Publishing messages 18 to 19.
136 19865 [iot_threa] [INFO ][MQTT][19865] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
137 19876 [iot_threa] [INFO ][MQTT][19876] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
138 19885 [iot_threa] [INFO ][DEMO][19885] Waiting for 2 publishes to be received.
139 19953 [iot_threa] [INFO ][DEMO][19953] MQTT PUBLISH 18 successfully sent.
140 19980 [iot_threa] [INFO ][DEMO][19980] Incoming PUBLISH received:
Subscription topic filter: iotdemo/topic/3
Publish topic name: iotdemo/topic/3
Publish retain flag: 0
Publish QoS: 1
Publish payload: Hello world 18!
141 20001 [iot_threa] [INFO ][MQTT][20001] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
142 20011 [iot_threa] [INFO ][DEMO][20011] Acknowledgment message for PUBLISH 18 will be sent.
143 20053 [iot_threa] [INFO ][DEMO][20053] MQTT PUBLISH 19 successfully sent.
144 20069 [iot_threa] [INFO ][DEMO][20068] Incoming PUBLISH received:
Subscription topic filter: iotdemo/topic/4
Publish topic name: iotdemo/topic/4
Publish retain flag: 0
Publish QoS: 1
Publish payload: Hello world 19!
145 20089 [iot_threa] [INFO ][MQTT][20089] (MQTT connection 100337e0) MQTT PUBLISH operation queued.
146 20099 [iot_threa] [INFO ][DEMO][20099] Acknowledgment message for PUBLISH 19 will be sent.
147 20108 [iot_threa] [INFO ][DEMO][20108] 2 publishes received.
148 20116 [iot_threa] [INFO ][MQTT][20116] (MQTT connection 100337e0) UNSUBSCRIBE operation scheduled.
149 20129 [iot_threa] [INFO ][MQTT][20128] (MQTT connection 100337e0, UNSUBSCRIBE operation 100339e0) Waiting for operation completion.
150 20322 [iot_threa] [INFO ][MQTT][20321] (MQTT connection 100337e0, UNSUBSCRIBE operation 100339e0) Wait complete with result SUCCESS.
151 20335 [iot_threa] [INFO ][MQTT][20335] (MQTT connection 100337e0) Disconnecting connection.
152 20347 [iot_threa] [INFO ][MQTT][20347] (MQTT connection 100337e0, DISCONNECT operation 100339e0) Waiting for operation completion.
153 20359 [iot_threa] [INFO ][MQTT][20359] (MQTT connection 100337e0, DISCONNECT operation 100339e0) Wait complete with result SUCCESS.
154 20371 [iot_threa] [INFO ][MQTT][20371] (MQTT connection 100337e0) Connection disconnected.
155 20380 [iot_threa] [INFO ][MQTT][20380] (MQTT connection 100337e0) Network connection closed.
156 21622 [iot_threa] [INFO ][MQTT][21622] (MQTT connection 100337e0) Network connection destroyed.
157 21631 [iot_threa] [INFO ][MQTT][21631] MQTT library cleanup done.
158 21637 [iot_threa] [INFO ][DEMO][21637] Demo completed successfully.

LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
159 21772 [iot_threa] [INFO ][INIT][21772] SDK cleanup done.
160 21777 [iot_threa] [INFO ][DEMO][21777] -------DEMO FINISHED-------
```

6.3 Monitoring MQTT messages on the 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.





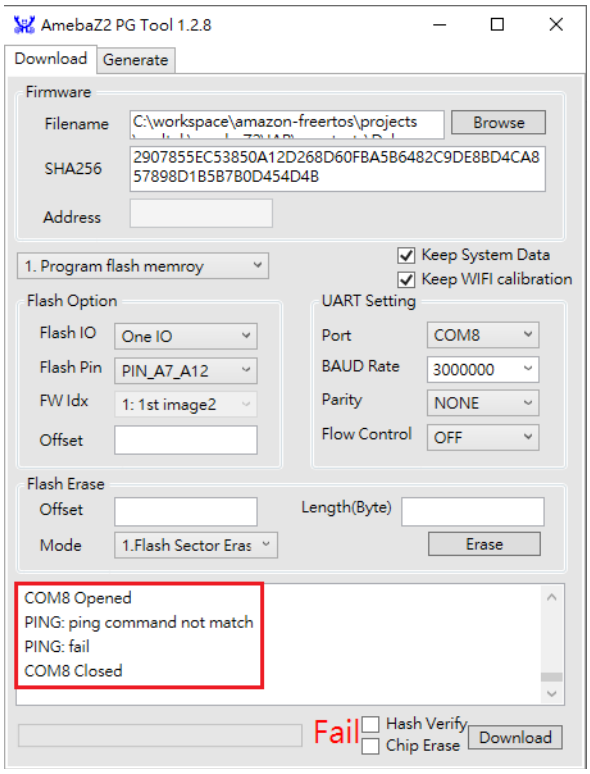
7 Troubleshooting

If these steps don't work, look at the device log in the serial terminal. You should see some text that indicates the source of the problem.

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

7.1 Image Tool Download Fail

Please check device in UART_DOWNLOAD mode or not. Refer 5.3 for more detail.



7.2 ERROR: Invalid Key

Please check **WIFI_SSID** and **WIFI_PASSWORD** in in ambd_amazon-freertos/blob/master/demos/include/aws_clientcredential.h

```
Enter SSID for Soft AP started
3 1098 [example_a] Wi-Fi configuration successful.
4 1108 [iot_threa] [INFO ][DEMO][1108] ---------STARTING DEMO---------

5 1115 [iot_threa] [INFO ][INIT][1115] SDK successfully initialized.

LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
WIFI deinitialized
Initializing WIFI ...
WIFI initialized

Joining BSS by SSID ...

ERROR:Invalid Key
ERROR: Can't connect to AP
Joining BSS by SSID ...

ERROR:Invalid Key
ERROR: Can't connect to AP
Joining BSS by SSID ...
```

7.3 Failed to establish new MQTT connection

Please check **clientcredentialMQTT_BROKER_ENDPOINT** in ambd_amazon-freertos/blob/master/demos/include/aws_clientcredential.h

```
6 12508 [iot_threa] [INFO ][DEMO][12508] Successfully initialized the demo. Network type for the demo: 1
7 12517 [iot_threa] [INFO ][MQTT][12517] MQTT library successfully initialized.
8 12524 [iot_threa] [INFO ][DEMO][12524] MQTT demo client identifier is ameba-ota (length 9).
9 12624 [iot_threa] [ERROR][NET][12624] Failed to resolve [redacted].amazonaws.com.
10 12934 [iot_threa] [ERROR][MQTT][12934] Failed to establish new MQTT connection, error NETWORK ERROR.
11 12943 [iot_threa] [ERROR][DEMO][12943] MQTT CONNECT returned error NETWORK ERROR.
12 12951 [iot_threa] [INFO ][MQTT][12950] MQTT library cleanup done.
13 12957 [iot_threa] [ERROR][DEMO][12957] Error running demo.
Interface 0 IP address : 192.168.90.185
LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
14 13094 [iot_threa] [INFO ][INIT][13094] SDK cleanup done.
15 13099 [iot_threa] [INFO ][DEMO][13099] -------DEMO FINISHED-------
```

7.4 TLS_Connect fail

Please check **keyCLIENT_CERTIFICATE_PEM** and **keyCLIENT_PRIVATE_KEY_PEM** in ambd_amazon-freertos/blob/master/demos/include/aws_clientcredential_keys.h

```
8 13501 [iot_threa] [INFO ][DEMO][13501] Successfully initialized the demo. Network type for the demo: 1
9 13511 [iot_threa] [INFO ][MQTT][13511] MQTT library successfully initialized.
10 13518 [iot_threa] [INFO ][DEMO][13518] MQTT demo client identifier is ameba-ota (length 9).
11 20102 [iot_threa] ERROR: Private key not found. 12 20107 [iot_threa] TLS_Connect fail (0x7d4, [redacted].amazonaws.com)
13 20115 [iot_threa] [ERROR][NET][20115] Failed to establish new connection. Socket status: -1.
14 20424 [iot_threa] [ERROR][MQTT][20424] Failed to establish new MQTT connection, error NETWORK ERROR.
15 20433 [iot_threa] [ERROR][DEMO][20433] MQTT CONNECT returned error NETWORK ERROR.
16 20441 [iot_threa] [INFO ][MQTT][20441] MQTT library cleanup done.
17 20447 [iot_threa] [ERROR][DEMO][20447] Error running demo.
Interface 0 IP address : 192.168.90.185
LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
18 20586 [iot_threa] [INFO ][INIT][20586] SDK cleanup done.
19 20591 [iot_threa] [INFO ][DEMO][20591] -------DEMO FINISHED-------
```