AmebaPro Amazon FreeRTOS-LTS - 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.

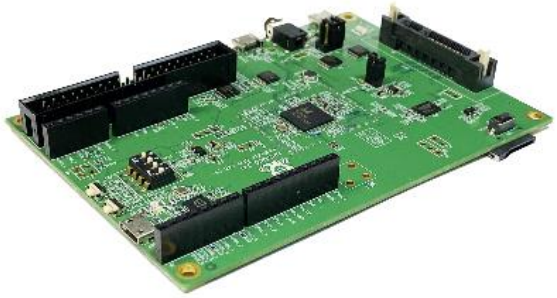
USING THIS DOCUMENT

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 AmebaPro RTL8715AD Board

1.1 AmebaPro Demo EVB

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



CPU

32-bit Arm v8M, up to 300MHz

32-bit Arm®Cortex®-M0, up to 4MHz

MEMORY

512KB RAM + 32MB LPDDR

KEY FEATURES

Integrated 802.11ac/n Wi-Fi SoC

Trustzone-M Security

Hardware SSL Engine

Root Trust Secure Boot

USB Host/Device

SD Host

LCDC

Codec

ISP

H.264

OTHER FEATURES

4 SPI interface

5 UART interface

2 I2S interface

4 I2C interface

11 ADC interface

16 PWM

2 PCM

Max 90 GPIO

1.2 PCB Layout Overview

The PCB layout of AmebaPro is shown in Fig 1-1.

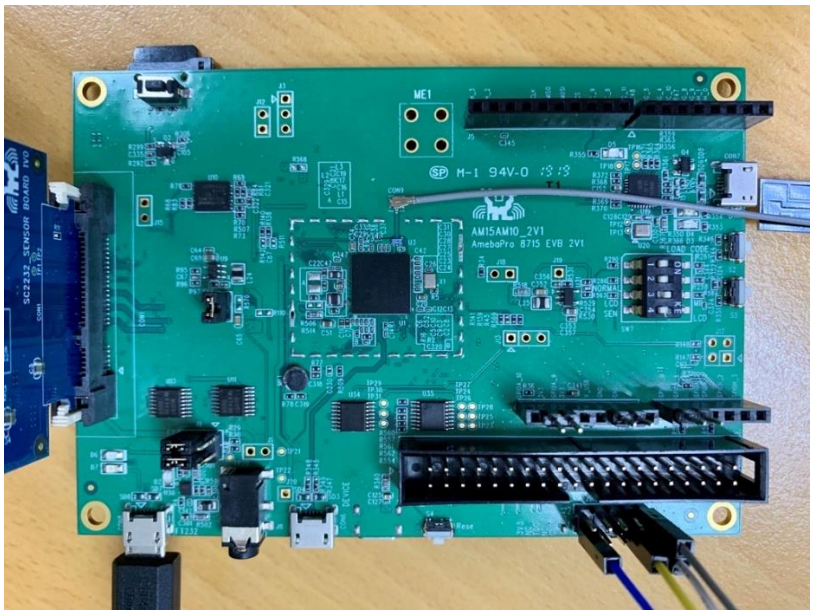


Fig 1-1 Demo board – PCB layout (2D)

1.3 Log UART

The log UART is shown in Fig 1-2.

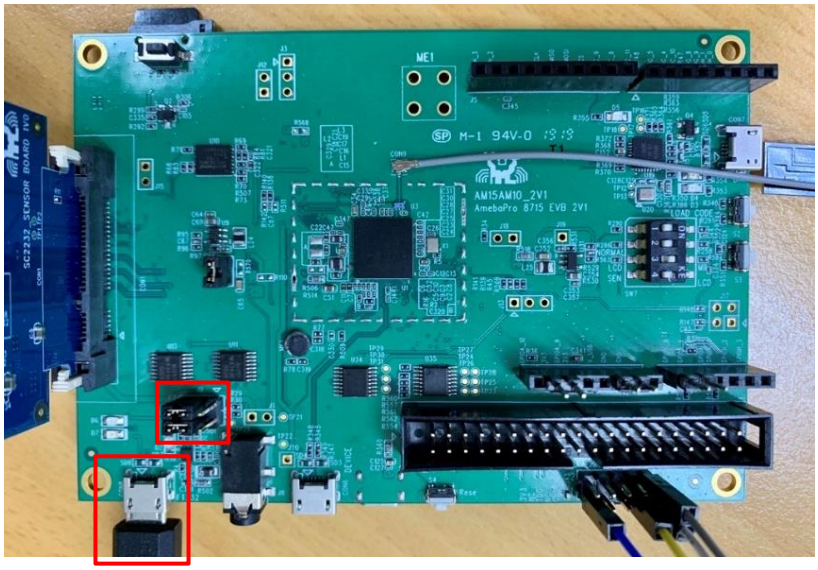


Fig 1-2 Demo board – log UART

1.4 JTAG/SWD

The SWD interface is shown in Fig 1-3.

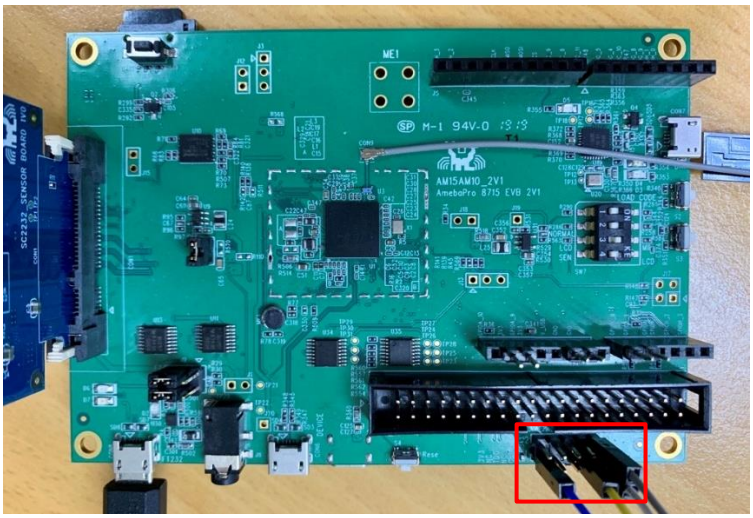


Fig 1-3 Demo board – JTAG/SWD

Note: If using 2V0、2V1 version AmebaPro. Please check SW7 pin 3 switch to ON before connection.

1.5 Image Sensor

There is an image sensor socket as shown in Fig 1-4.

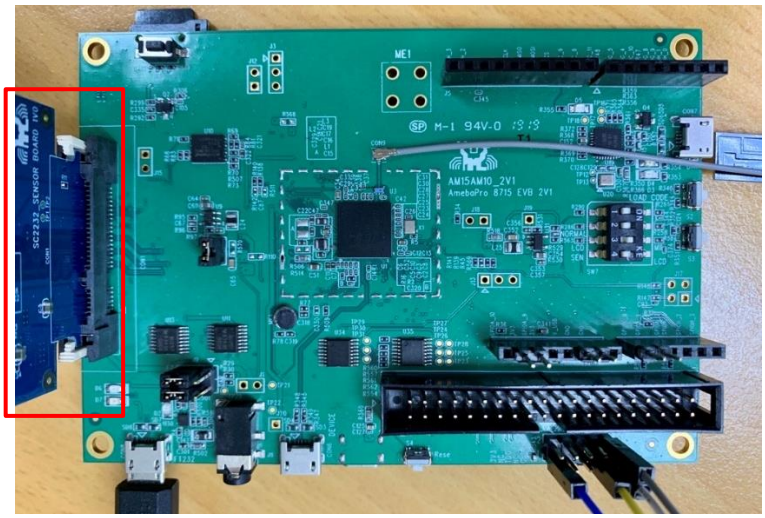


Fig 1-4 Demo board – image sensor

2 Configure AWS IoT Core

2.1 Set up your AWS account and Permissions

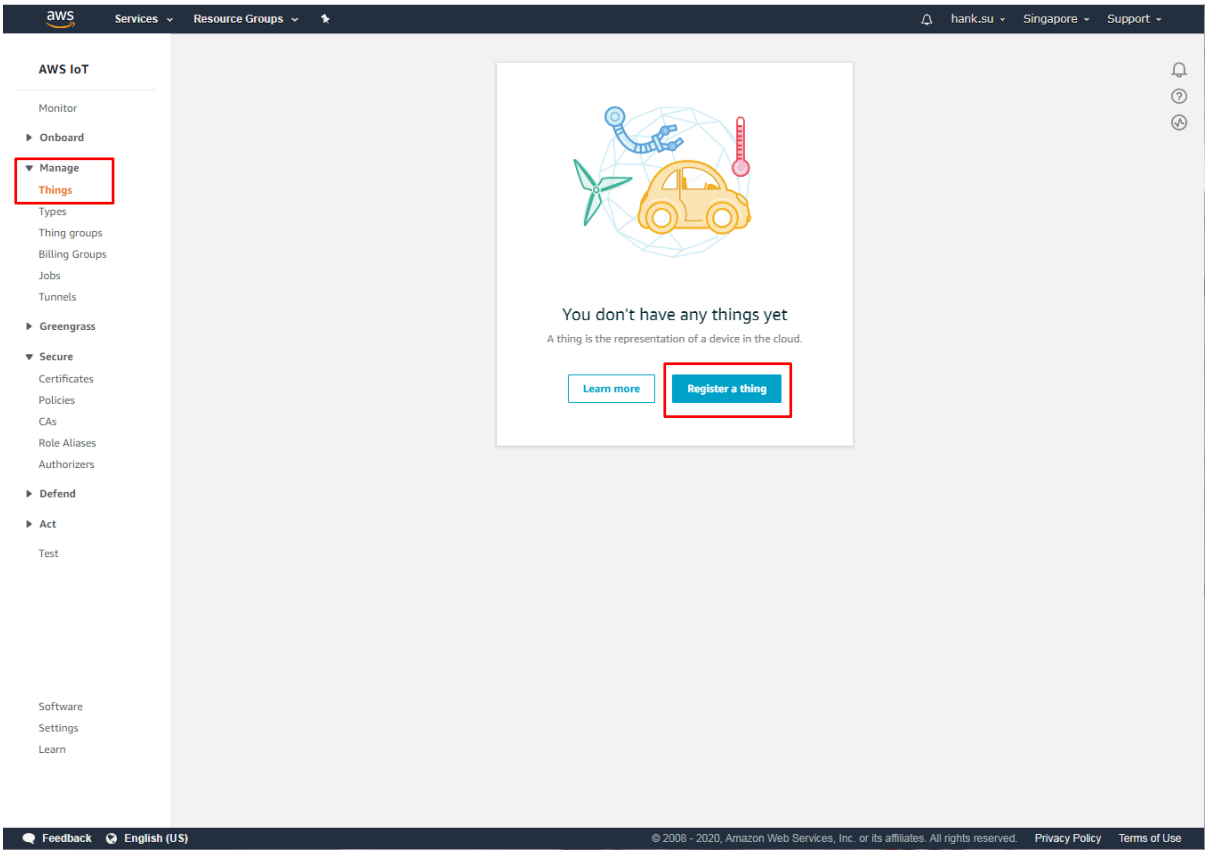
Refer to the instructions at Set up your AWS Account https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html. Follow the steps outlined in these sections to create your account and a user and get started:

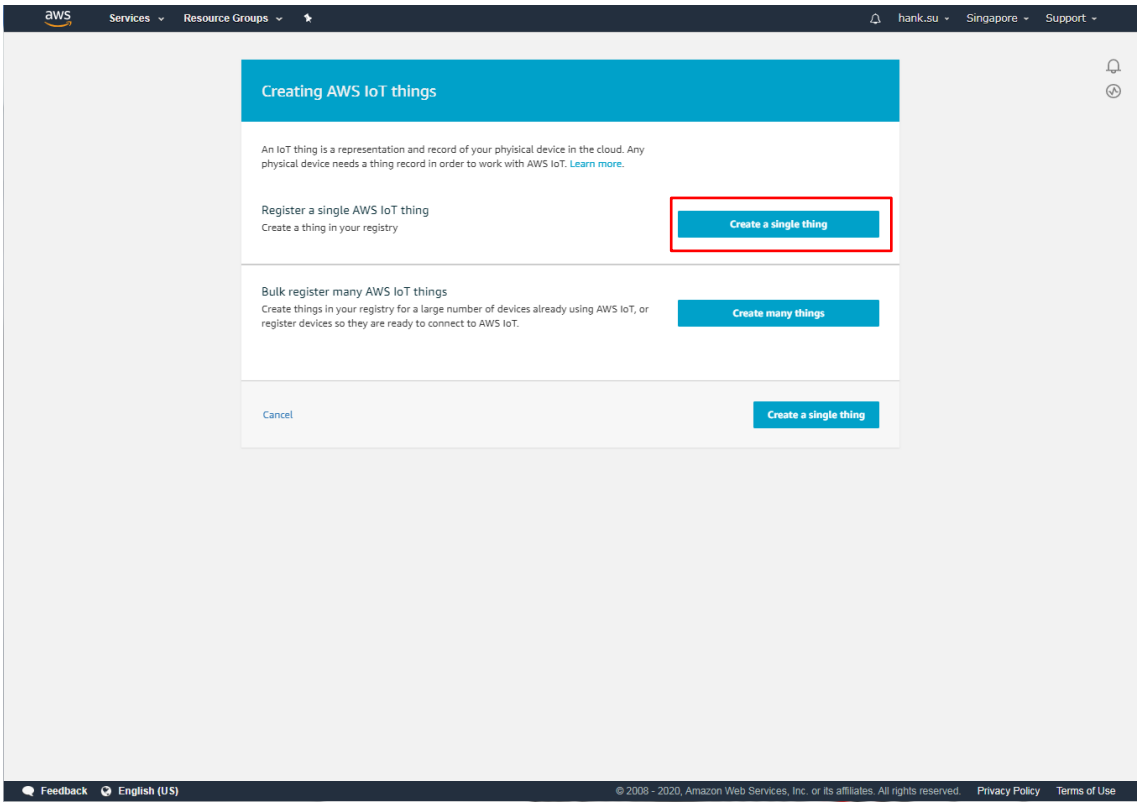
- Sign up for an AWS account
- Create a user and grant permissions
- Open the AWS IoT console

Please pay special attention to the Notes in AWS webpage.

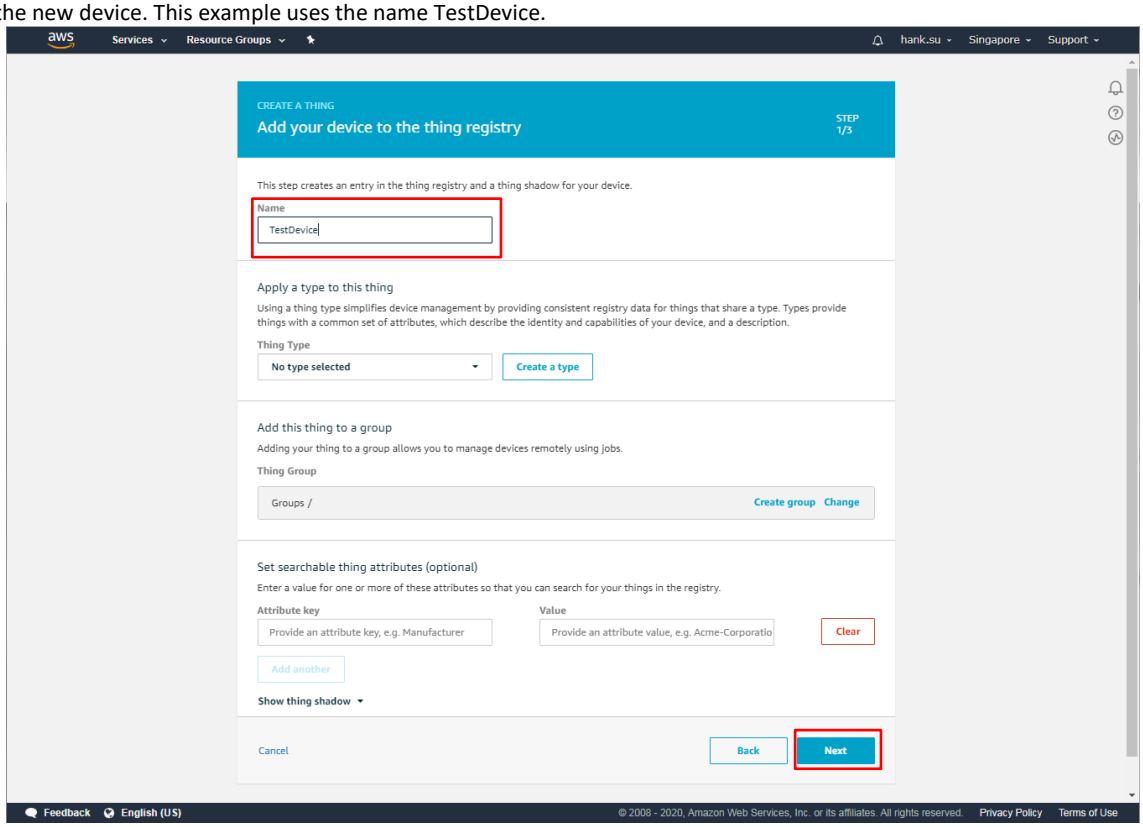
2.2 Create a New Device

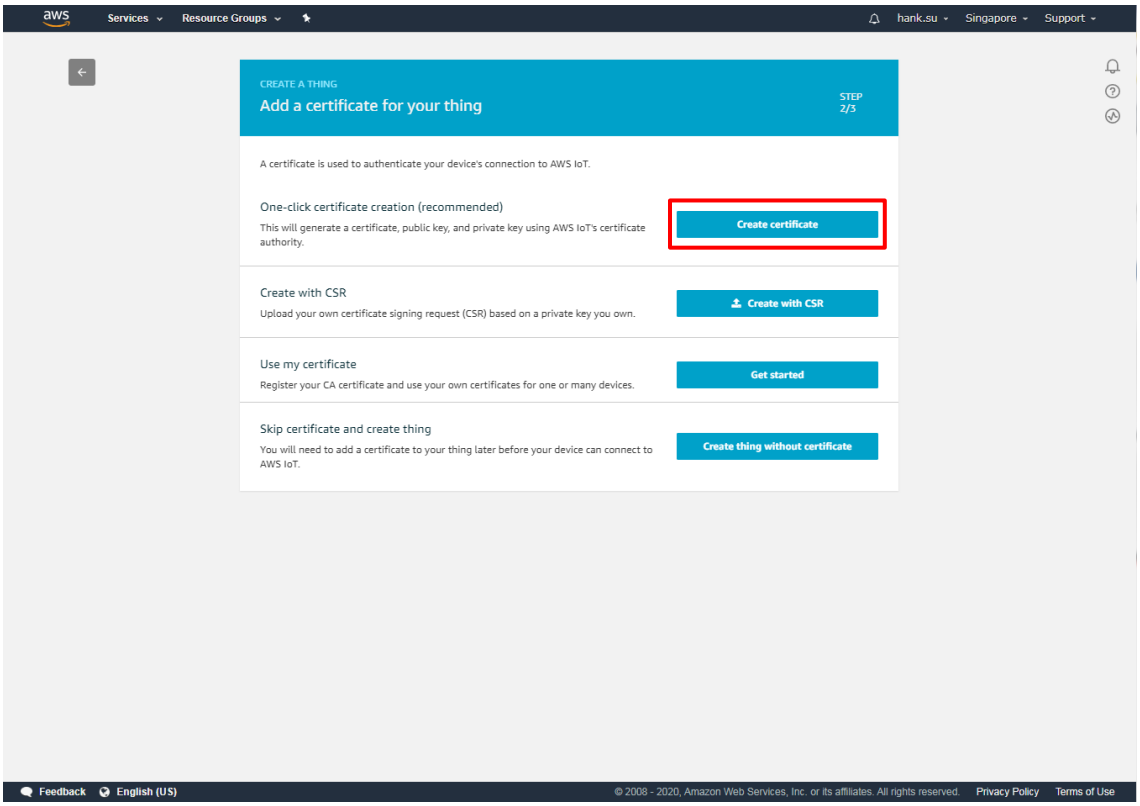
To create a new device, navigate to Manage -> Things in the left-hand navigation menu. Then click “Register a thing”.



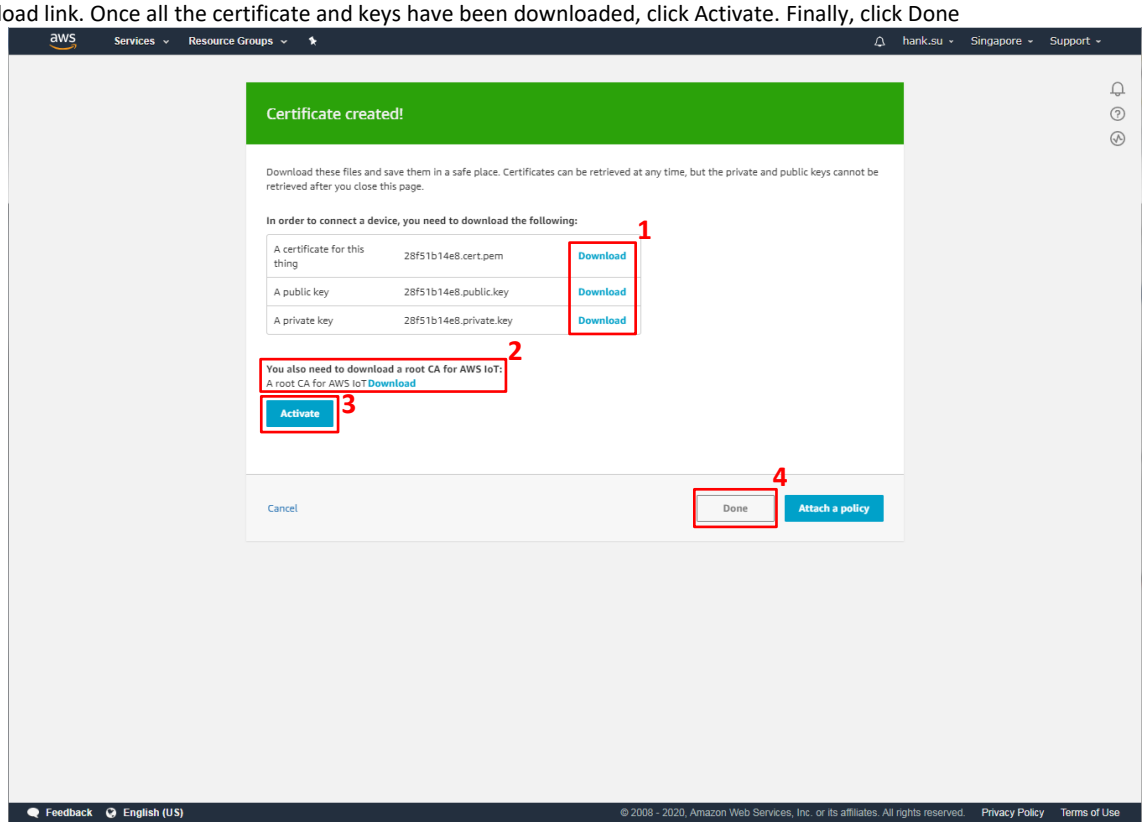


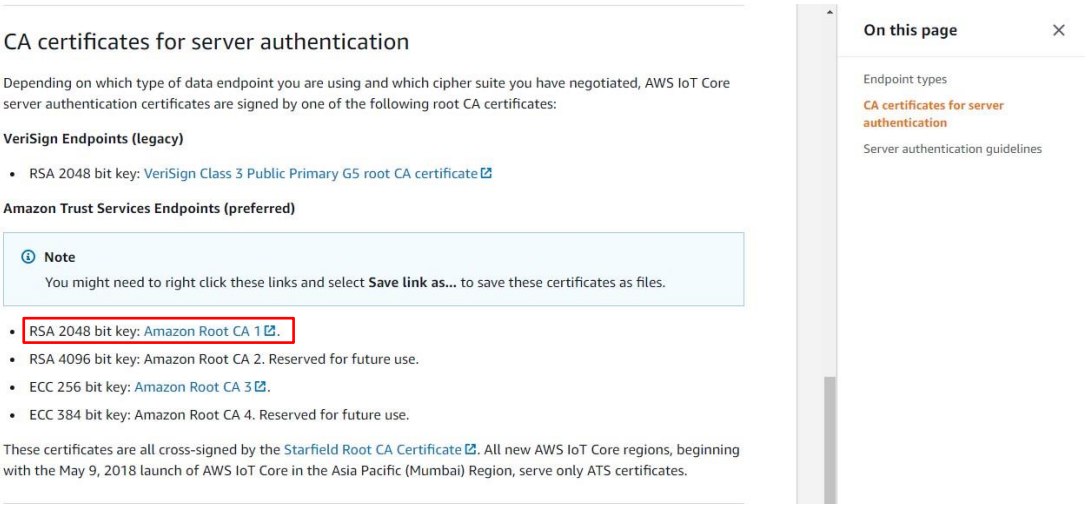
Then, name the new device. This example uses the name TestDevice.





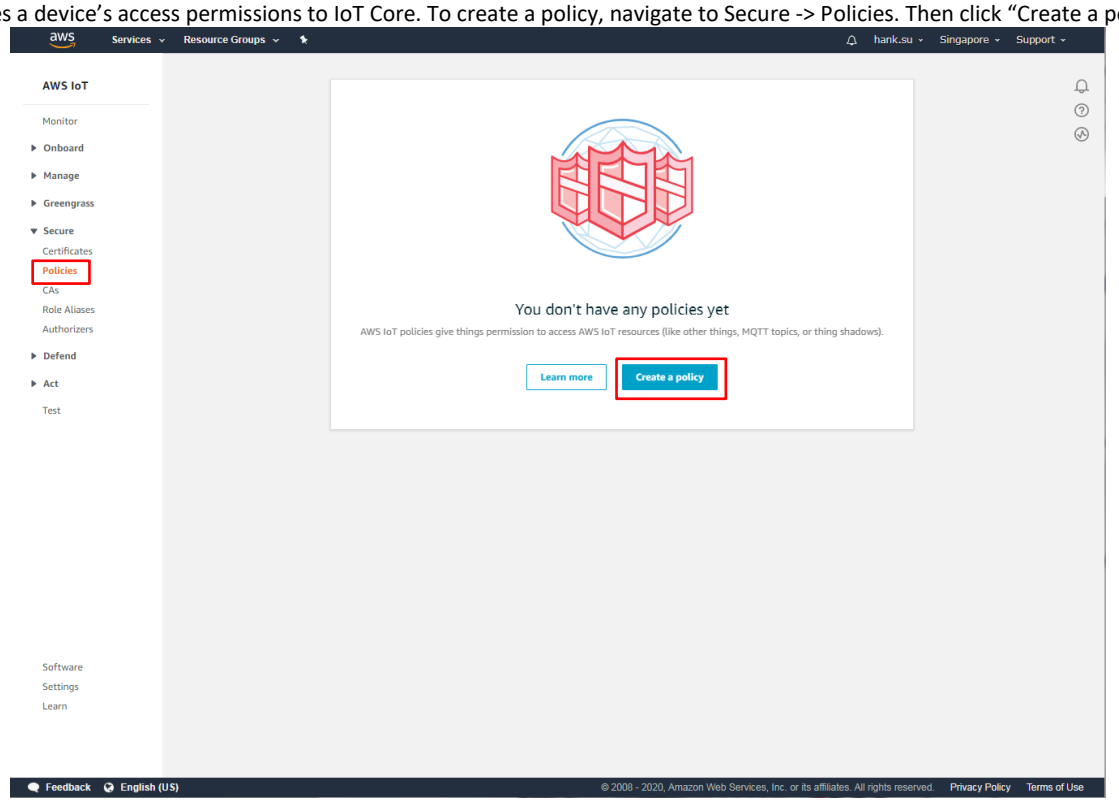
Download the certificate, public key, and private key for the device by clicking Download. Next, download the root CA for AWS IoT by clicking to the Download link. Once all the certificate and keys have been downloaded, click Activate. Finally, click Done



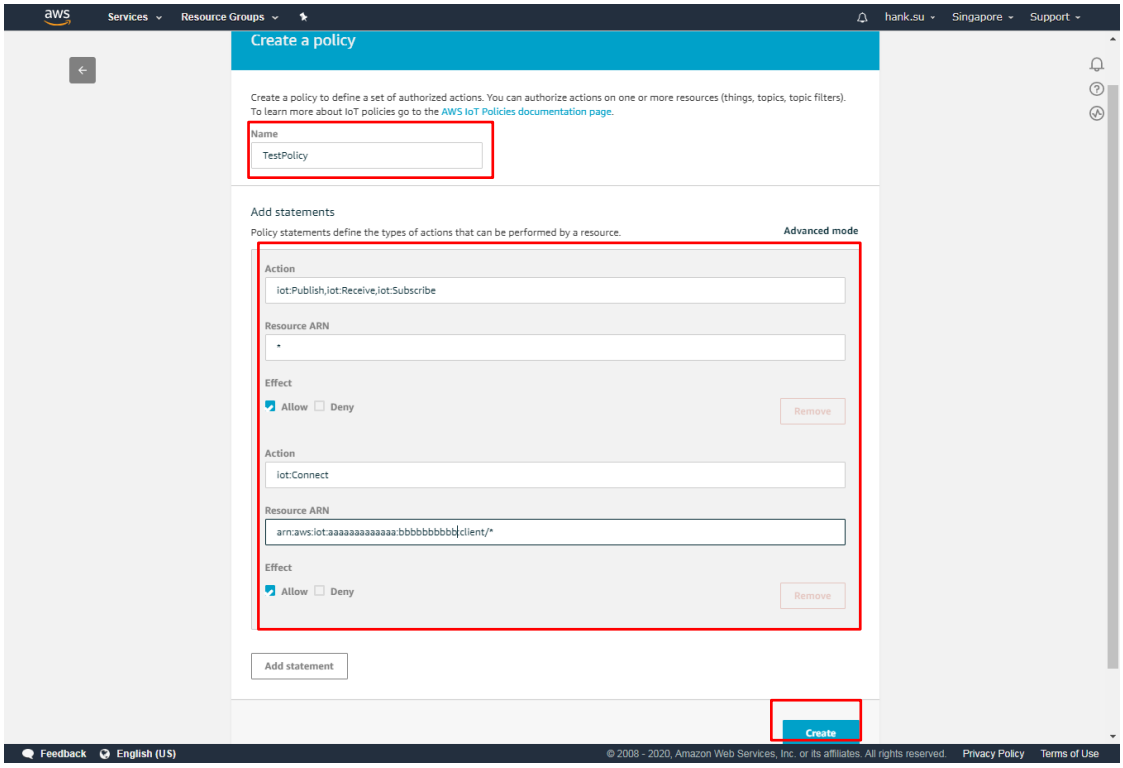


2.3 Create a policy

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

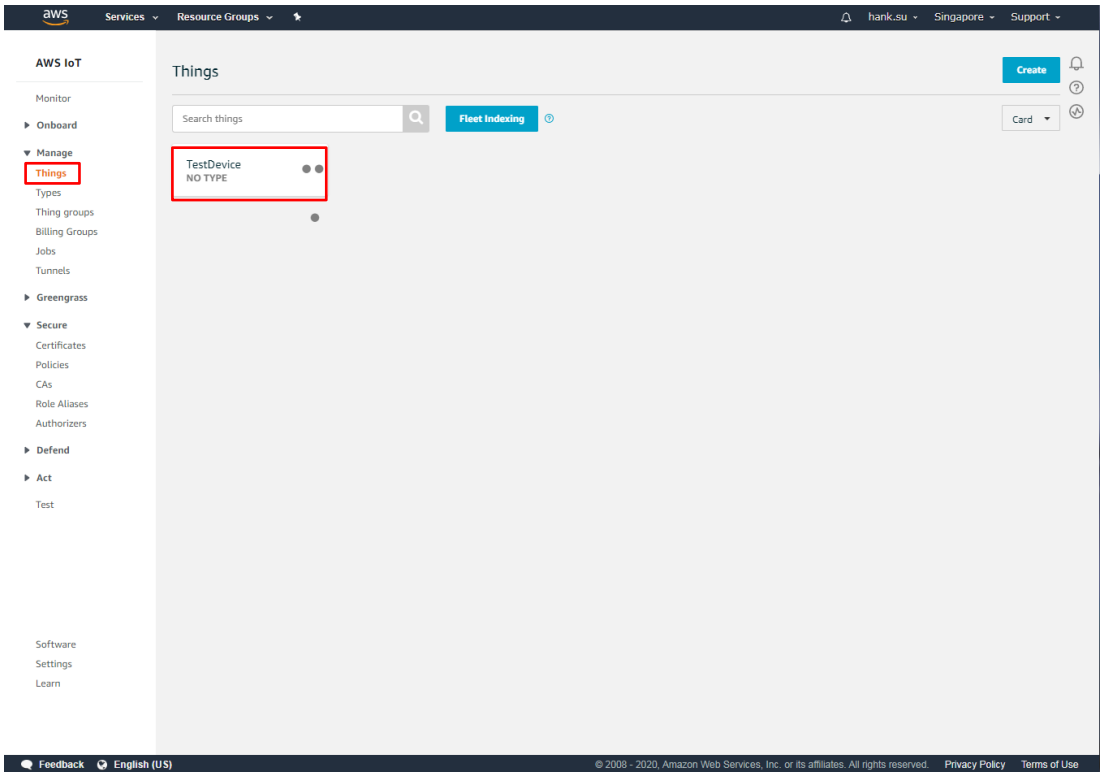


NOTE – this policy grants unrestricted access for all iot operations, and is to be used only in a development environment. For non-dev environments, all devices in your fleet must have credentials with privileges that authorize intended actions only, which include (but not limited to) AWS IoT MQTT actions 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

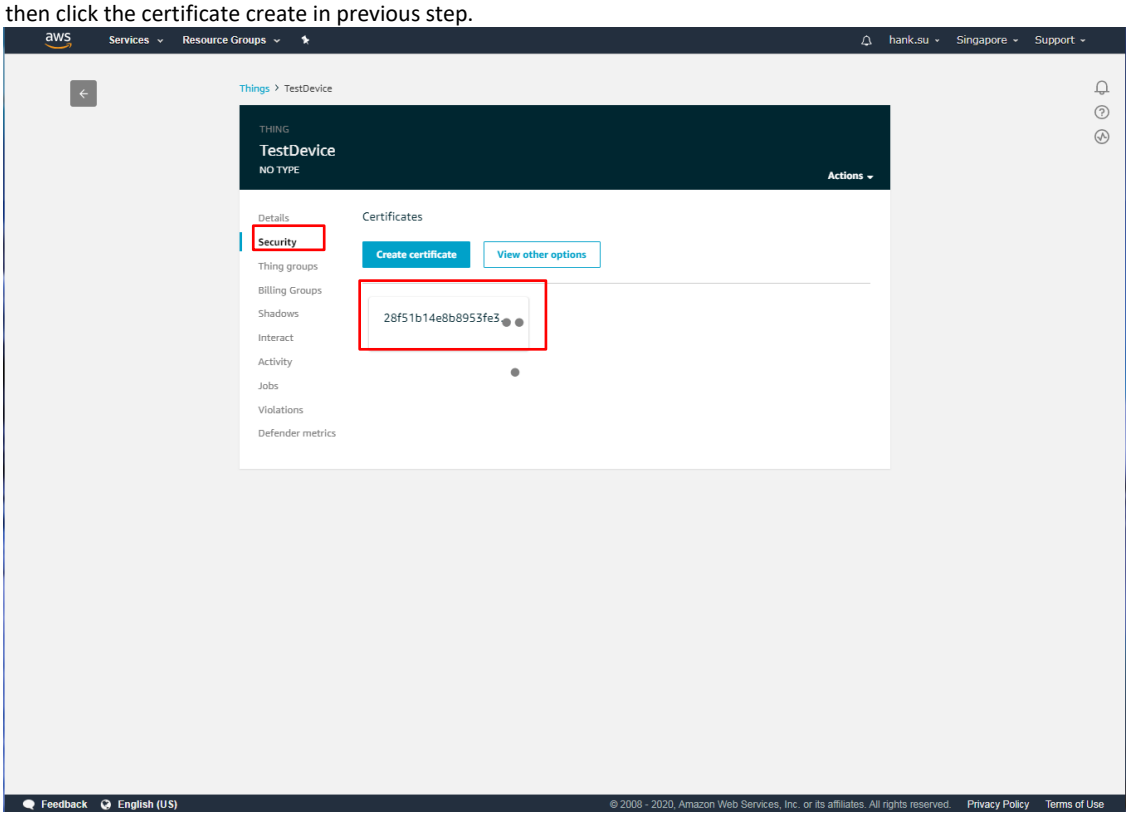


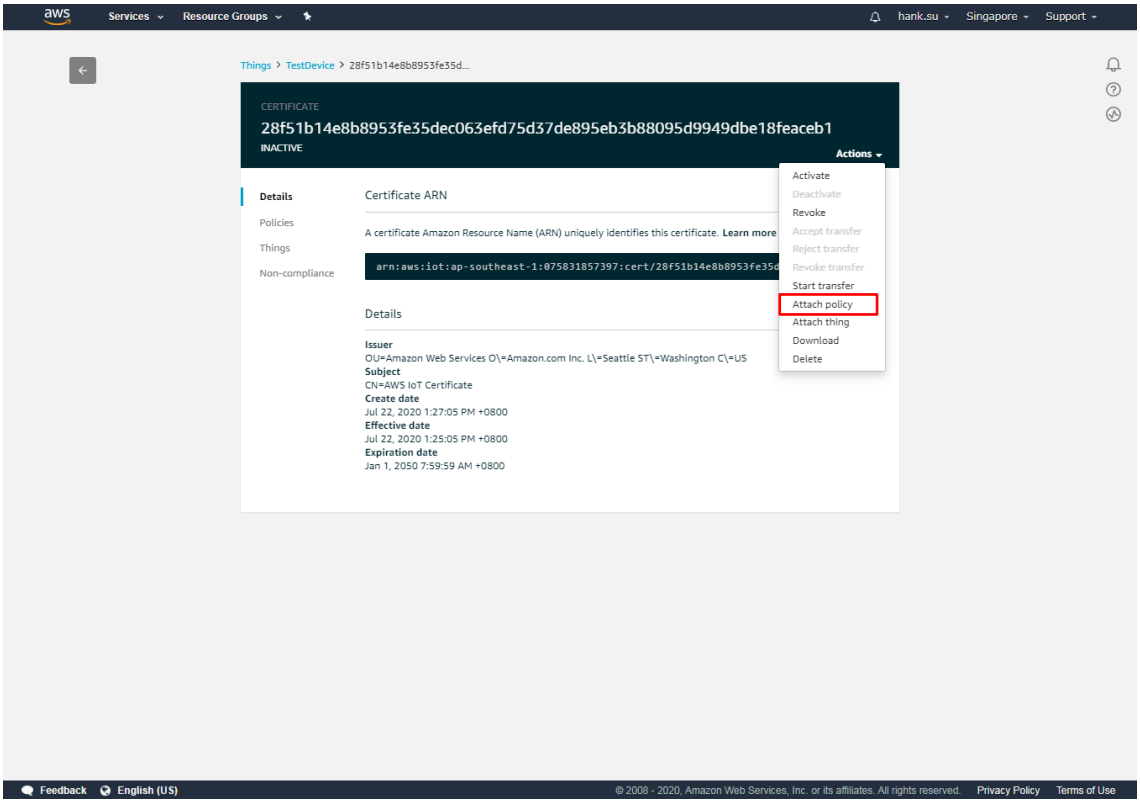
2.4 Attach Policy

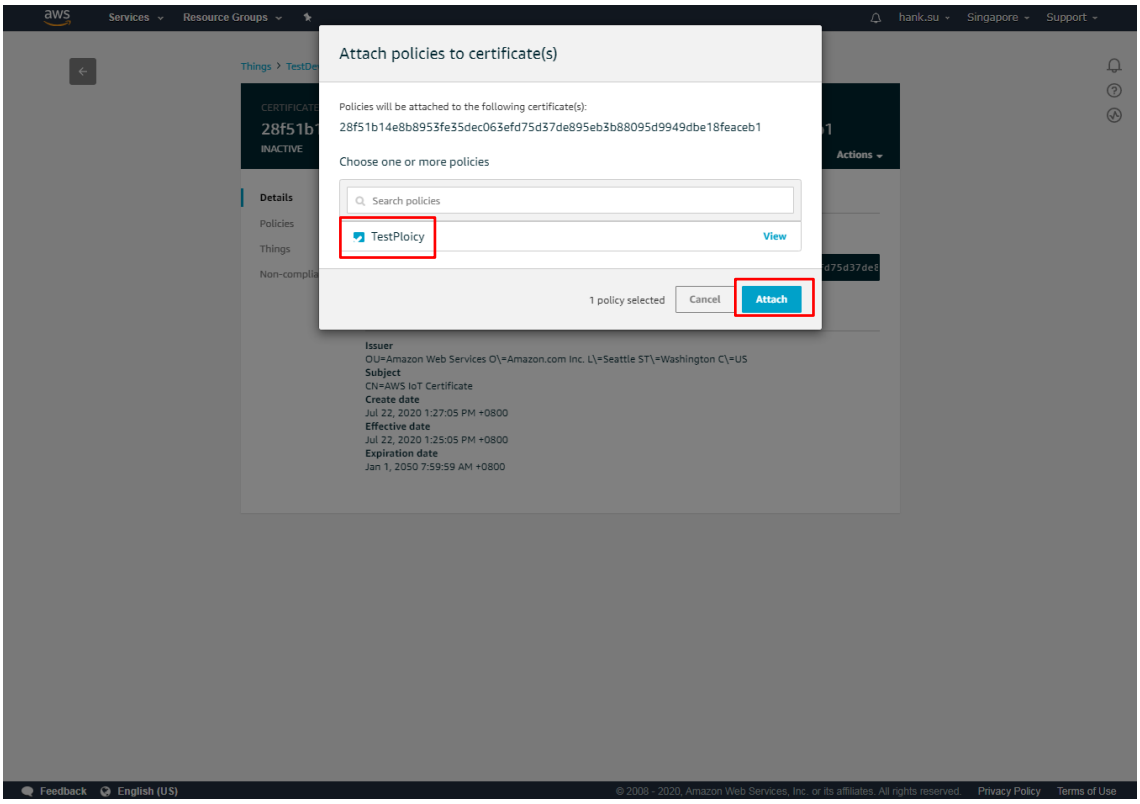
The last step to configuring the device is attaching a policy. To attach a policy to new device, navigate to Manage -> Things. Then click on the device which was created.



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



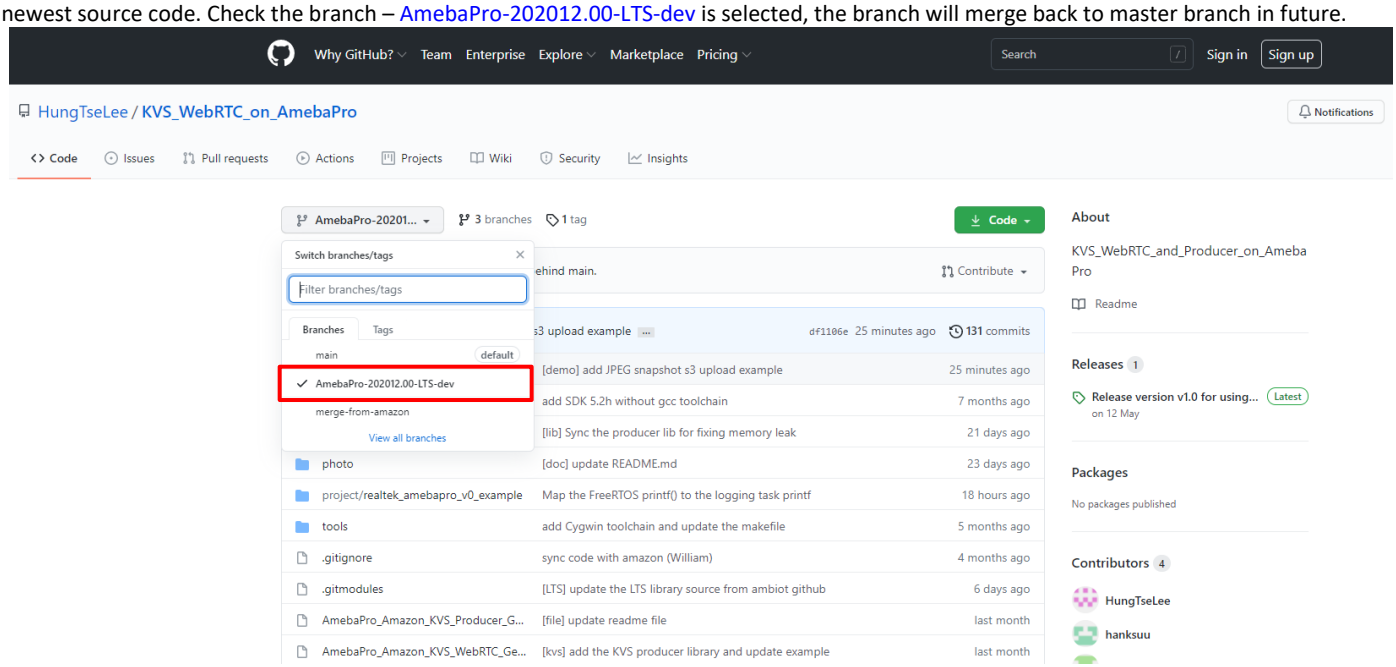




3 Configure AmebaPro Amazon FreeRTOS

3.1 Download Source Code from Github

Open source link: https://github.com/HungTseLee/KVS_WebRTC_on_AmebaPro/tree/AmebaPro-202012.00-LTS-dev and select master to get newest source code. Check the branch – AmebaPro-202012.00-LTS-dev is selected, the branch will merge back to master branch in future.



3.1.1 Download the Project

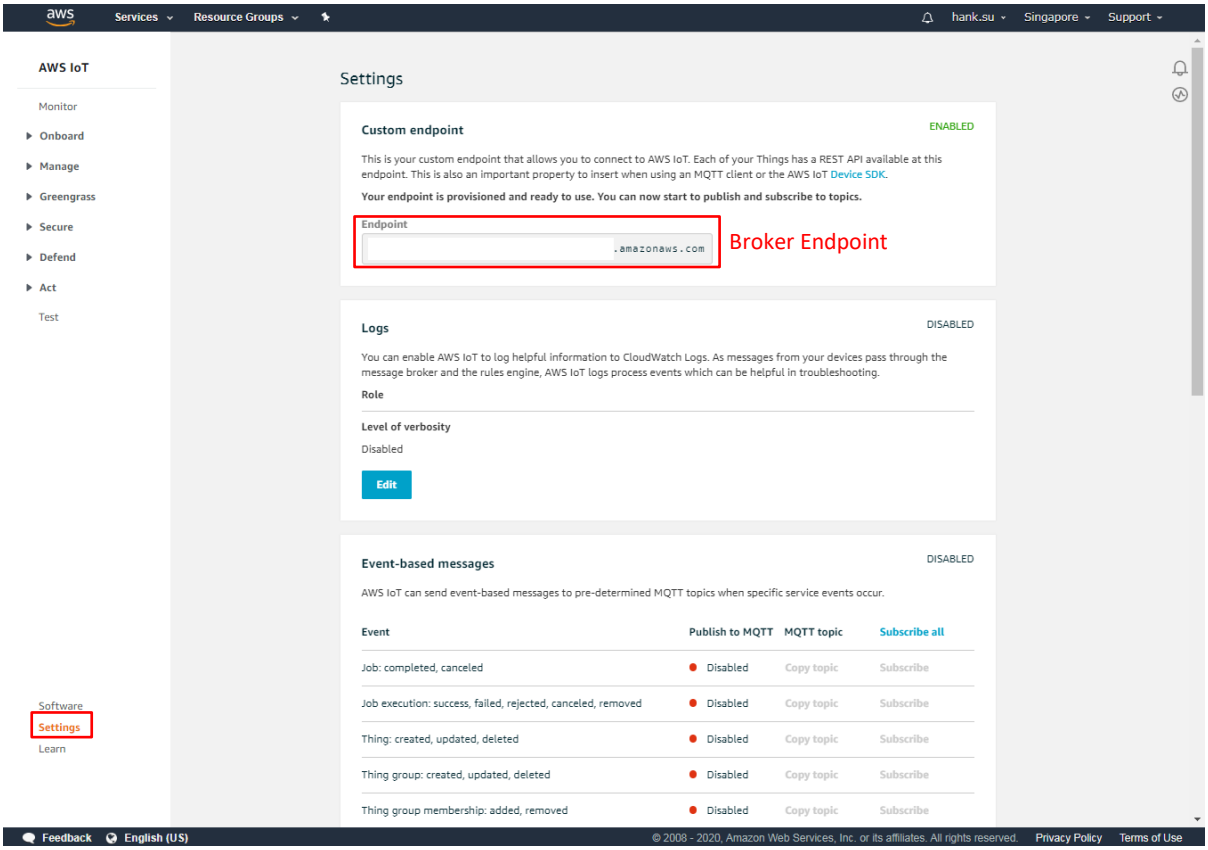
Run the command to download the whole project:

$ git clone -b AmebaPro-202012.00-LTS-dev --recurse-submodules https://github.com/HungTseLee/KVS_WebRTC_on_AmebaPro.git

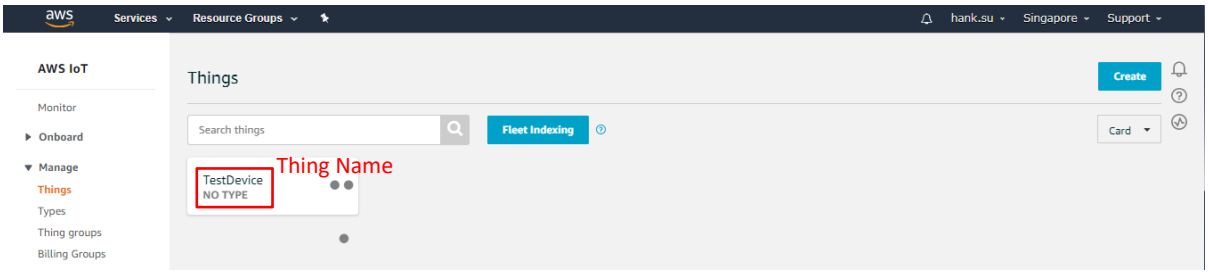
If you already have a checkout, run the following command to sync submodules:

$ git submodule update --init

3.2 Get Broker Endpoint by AWS IoT Core



3.3 Get Thing Name



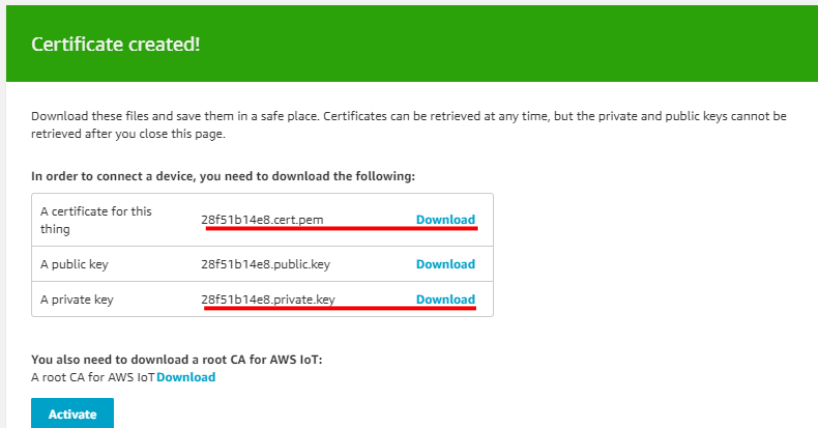
3.4 Setup IoT Core Information with AmebaPro Amazon FreeRTOS

Setup BROKER_ENDPOINT, THING_NAME, WIFI_SSID, PASSWORD in "component/common/application/amazon/amazon-freertos-202012.00/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

Fill keyCLIENT_CERTIFICATE_PEM and keyCLIENT_PRIVATE_KEY_PEM in "component/common/application/amazon/amazon-freertos-202012.00/demos/include /aws_clientcredential_keys.h" by xxxxxxxx-certifiacte.pem and xxxxxxxx-private.pem.key.



It can done by "component/common/application/amazon/amazon-freertos-202012.00/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"\
"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"\
"                                                                  \n"\
"                                                                  \n"\
"pOWEuLUuz2FAv1noAbN/6OQ8H/PT0AFJT/ghA04GnIUF0kjSzY60ehS2mVp6neP+\n"\
"AZjzZ6QJYlb5/PFz9oES448kpyaAoS2ke86+R4r4YOMBK+I5RVbfSQ==\n"\
"-----END RSA PRIVATE KEY-----\n"
```

3.4.2 Enable FreeRTOS demo on AmebaPro

Find platform_opts.h in "project/realtek_amebapro_v0_example/inc" and enable **CONFIG_EXAMPLE_AMAZON_FREERTOS**

```
/* For Amazon FreeRTOS LTS demo example */
#define CONFIG_EXAMPLE_AMAZON_FREERTOS          1
```

Find aws_demo_config.h in "component/common/application/amazon/amazon-freertos-202012.00/vendors/realtek/boards/amebaPro/aws_demos/config_files" and enable **CONFIG_CORE_MQTT_MUTUAL_AUTH_DEMO_ENABLED**

```
/* To run a particular demo you need to define one of these.
 * Only one demo can be configured at a time
 *
 *          CONFIG_CORE_HTTP_MUTUAL_AUTH_DEMO_ENABLED
 *          CONFIG_CORE_HTTP_S3_DOWNLOAD_DEMO_ENABLED
 *          CONFIG_CORE_HTTP_S3_DOWNLOAD_MULTITHREADED_DEMO_ENABLED
 *          CONFIG_CORE_HTTP_S3_UPLOAD_DEMO_ENABLED
 *          CONFIG_CORE_MQTT_MUTUAL_AUTH_DEMO_ENABLED
 *          CONFIG_CORE_MQTT_CONNECTION_SHARING_DEMO_ENABLED
 *          CONFIG_DEVICE_SHADOW_DEMO_ENABLED
 *          CONFIG_DEVICE_DEFENDER_DEMO_ENABLED
 *          CONFIG_JOBS_DEMO_ENABLED
 *          CONFIG_MQTT_BLE_DEMO_ENABLED
 *          CONFIG_GREENGRASS_DISCOVERY_DEMO_ENABLED
 *          CONFIG_TCP_ECHO_CLIENT_DEMO_ENABLED
 *          CONFIG_POSIX_DEMO_ENABLED
 *          CONFIG_OTA_UPDATE_DEMO_ENABLED
 *          CONFIG_BLE_GATT_SERVER_DEMO_ENABLED
 *          CONFIG_BLE_NUMERIC_COMPARISON_DEMO_ENABLED
 *
 *  These defines are used in iot_demo_runner.h for demo selection */

#define CONFIG_CORE_MQTT_MUTUAL_AUTH_DEMO_ENABLED
//#define CONFIG_OTA_UPDATE_DEMO_ENABLED
//#define CONFIG_DEVICE_SHADOW_DEMO_ENABLED
//#define CONFIG_CORE_HTTP_S3_UPLOAD_DEMO_ENABLED
//#define CONFIG_CORE_HTTP_MUTUAL_AUTH_DEMO_ENABLED
```

Now you can start to compile AmebaPro Amazon FreeRTOS

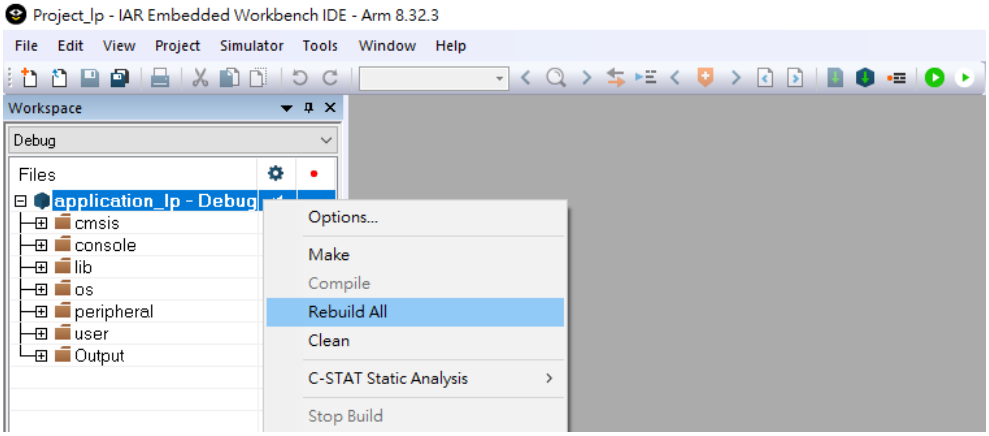
4 Compile AmebaPro Amazon FreeRTOS

4.1 IAR Embedded Workbench Build Environment Setup

AmebaPro use the newest Big-Little architecture. Since the big CPU will depend on the setting of small CPU, **it is necessary to compile the small CPU before the big CPU.**

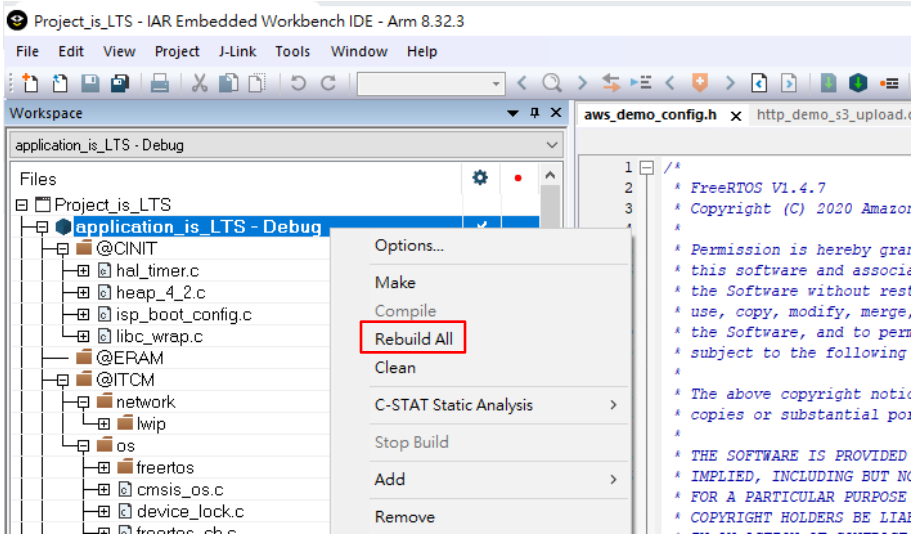
4.1.1 Compile Little CPU

step 1. Open SDK/project/realtek_amebapro_v0_example/EWARMRELEASE/**Project_lp.eww**.
step 2. Confirm application_lp in WorkSpace, right click application_lp and choose “Rebuild All” to compile.
step 3. Make sure there is no error after compile.



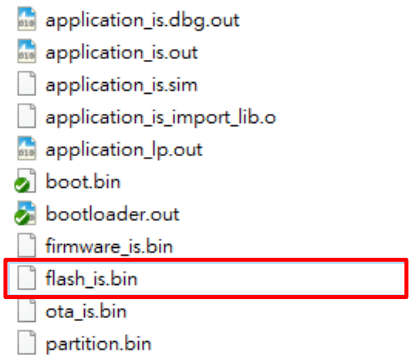
4.1.2 Compile Big CPU

step 1. Open SDK/project/realtek_amebapro_v0_example/EWARMRELEASE/**Project_is_LTS.eww**.
step 2. Confirm application_is_LTS in WorkSpace, right click application_is_LTS and choose “Rebuild All” to compile.
step 3. Make sure there is no error after compile.



4.1.3 Generating Image (Bin)

After compile, the images partition.bin, boot.bin, firmware_is.bin and flash_is.bin can be seen in the **EWARM-RELEASE\Debug\Exe**.
flash_is.bin links partition.bin, boot.bin and firmware_is.bin. Users need to choose **flash_is.bin** when downloading the image to board by Image Tool.



4.2 Compile Program with GCC Toolchain

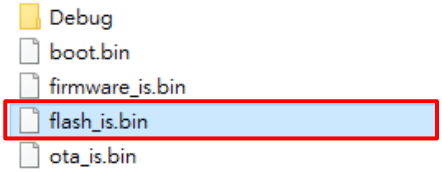
If using Linux environment or Cygwin on windows, follow the instructions below to build the project

$ cd project/realtek_amebapro_v0_example/GCC-RELEASE

Build the library and the example by running make in the directory

$ make -f Makefile_amazon_LTS all

If somehow it built failed, you can try to type **$ make -f Makefile_amazon_LTS clean** and then redo the make procedure.
After successfully build, there should be a directory named "application_is" created under GCC-RELEASE/ directory.
The image file **flash_is.bin** is located in "application_is" directory.



Note:
if there is compile error with shell script, you may need to run following command to deal with the problem

$ dos2unix component/soc/realtek/8195b/misc/gcc_utility/*

5 Image Tool

The tool **ImageTool.exe** can be find in **project\tools\AmebaPro\Image_Tool\ImageTool.exe**

5.1 Introduction

As show in the following figure, Image Tool has two tab pages:

- Download: used as image download server to transmit images to AmebaPro through UART
- Generate: concat separate images and generate a final image

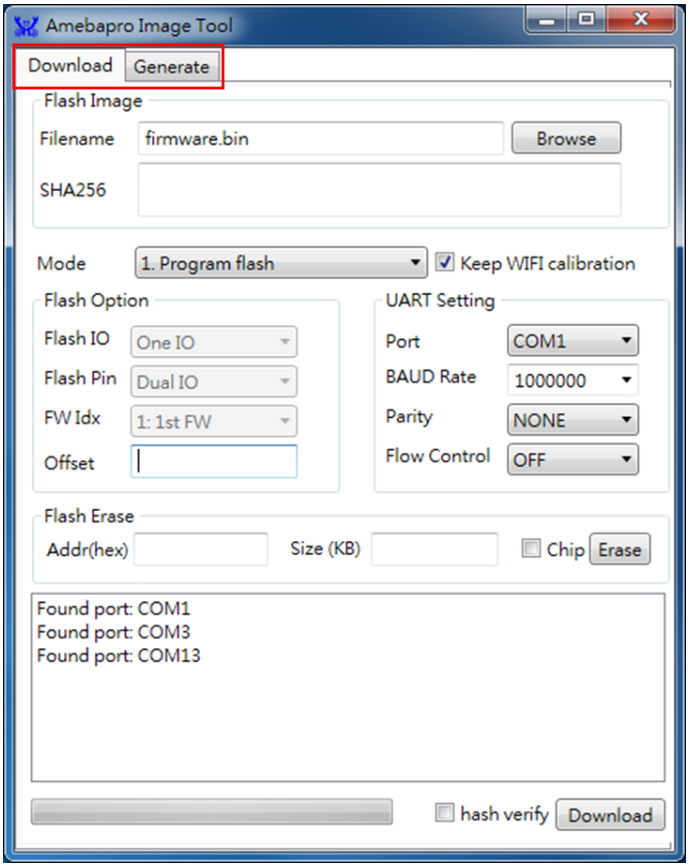


Fig 5-1 ImageTool UI

5.2 Environment Setup

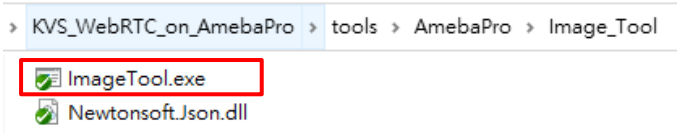
5.2.1 Hardware Setup

The hardware setup is shown in Fig 5-2.

Fig 5-2 Hardware setup

5.2.2 Software Setup

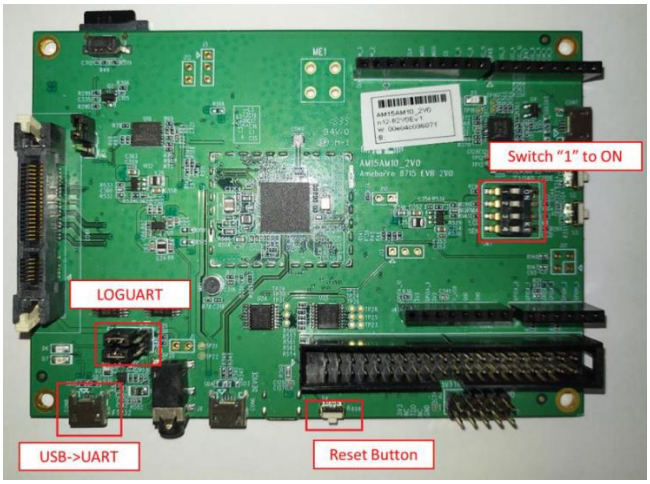
Execute **ImageTool.exe** from location **project\tools\AmebaPro\Image_Tool\ImageTool.exe**



5.3 Download

5.3.1 Enter the Download Mode to Ready

Image tool use UART to transmit image to AmebaPro board. Before performing image download function, AmebaPro need to enter UART_DOWNLOAD mode first. Please follow below steps to get AmebaPro into UART_DOWNLOAD mode:

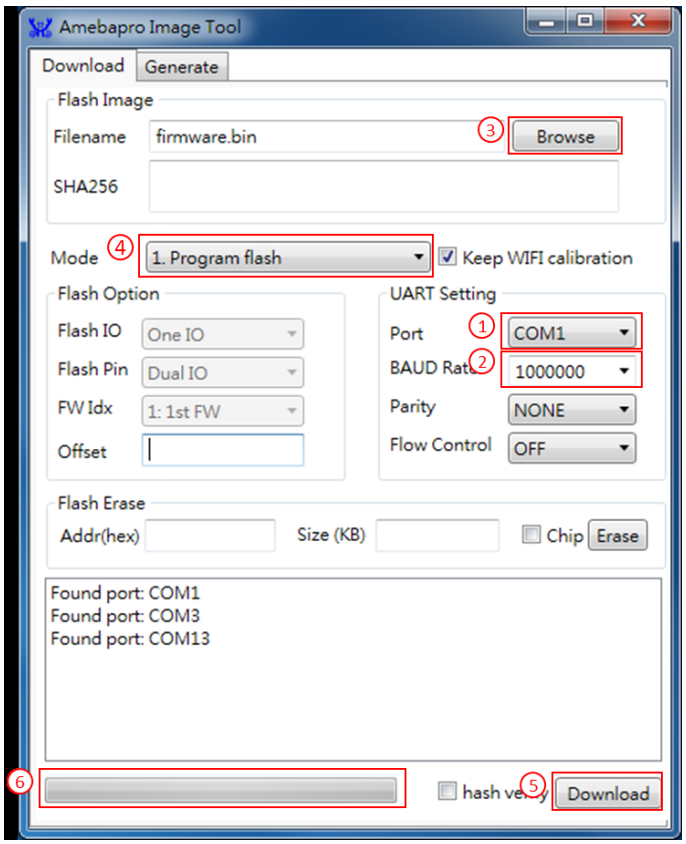


step 1. Connect LOGUART with FT pin by jumper cap.
step 2. Connect USB->UART to PC by using micro-USB wire.
step 3. Switch "1" to ON from SW7(2V0、2V1) or Switch "2" to ON from SW7(1V0)
step 4. Push reset button.

5.3.2 Download the Image to Flash

To download image through Image Tool, device need to enter UART_DOWNLOAD mode first.
Steps to download flash are as following:

step 1. Application will scan available UART ports. Please choose correct UART port. Please close other UART connection for the target UART port.
step 2. Choose desired baud rate between computer and AmebaPro.
step 3. Choose target flash binary image file “flash_xx.bin”
step 4. Check Mode is “1. Program flash”
step 5. Click “Download”
step 6. Progress will be shown on progress bar and result will be shown after download finish.
step 7. Switch “1” to OFF from SW7(2V0、2V1) or Switch “2” to OFF from SW7(1V0)
step 8. Push reset button to start the program.



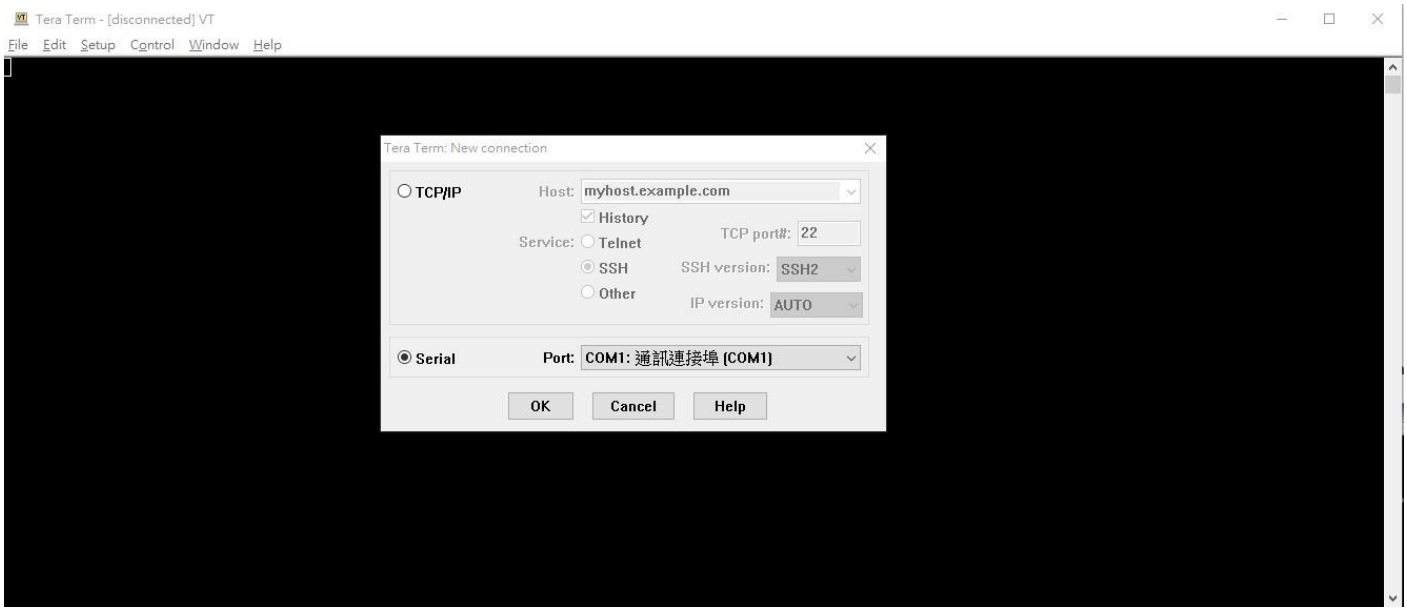
6 MQTT Demo

6.1 Get Device Log

Install **Tera Term** or other terminal emulator to get device log

Fig 6-1 Hardware setup

The serial port is same with ImageTool that get from 5.3.2 or use device manager to get the right serial port of device.



6.2 Run MQTT Demo

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

```
COM3:115200baud - Tera Term VT
File  Edit  Setup  Control  Window  Help

Initializing WIFI ...
RTL871X: [HALMAC]11692M
                      HALMAC_MAJOR_VER = 1
                                 HALMAC_PROTOTYPE_VER = 4
                                                 HALMAC_MINOR_VER = 14
                                                                 HALMAC_PATCH_VER = 0

RTL871X: efuse autoload en: 1

RTL871X: rfe_type=0x0, map=0x0

RTL871X: Download fw addr:9819ad80, size:115560
                                        - download_firmware_88xx HALMAC_RET_SUCCESS

RTL871X: fw: 24.4

RTL871X: RFE type = 0x0

Start LOG SERVICE MODE

# start_addr=(0x4000), end_addr=(0x8000), buffer_size=(0x4000), smp_number_max=(2048)

[Driver]: set ssid [lukai]
                        0 175 [example_a] Write certificate...
1 264 [iot_threa] [INFO ][DEMO][264] ---------STARTING DEMO---------

2 271 [iot_threa] [INFO ][INIT][271] SDK successfully initialized.

L
```

...

```
COM3:115200baud - Tera Term VT
File  Edit  Setup  Control  Window  Help
16 6811 [iot_threa] [INFO] [MQTT] [core_mqtt_serializer.c:912] 17 6815 [iot_threa] Connection accepted.18 6820 [iot_threa]

19 6820 [iot_threa] [INFO] [MQTT] [core_mqtt.c:1563] 20 6825 [iot_threa] Received MQTT CONNACK successfully from broker.21
 6832 [iot_threa]
22 6833 [iot_threa] [INFO] [MQTT] [core_mqtt.c:1829] 23 6838 [iot_threa] MQTT connection established with the broker.24 68
43 [iot_threa]
25 6846 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:820] 26 6852 [iot_threa] An MQTT connection is
established with a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.27 6862 [iot_threa]
28 6863 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:885] 29 6871 [iot_threa] Attempt to subscribe t
o the MQTT topic ameba-ota/example/topic.30 6877 [iot_threa]
31 6880 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:899] 32 6886 [iot_threa] SUBSCRIBE sent for top
ic ameba-ota/example/topic to broker.33 6895 [iot_threa]
34 7006 [iot_threa] [INFO] [MQTT] [core_mqtt.c:886] 35 7010 [iot_threa] Packet received. ReceivedBytes=3.36 7014 [iot_thre
a]
37 7017 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:1053] 38 7023 [iot_threa] Subscribed to the top
ic ameba-ota/example/topic with maximum QoS 1.39 7032 [iot_threa]
40 8032 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:533] 41 8037 [iot_threa] Publish to the MQTT to
pic ameba-ota/example/topic.42 8044 [iot_threa]
43 8047 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:543] 44 8054 [iot_threa] Attempt to receive pub
lish message from broker.45 8058 [iot_threa]
46 8284 [iot_threa] [INFO] [MQTT] [core_mqtt.c:886] 47 8287 [iot_threa] Packet received. ReceivedBytes=2.48 8292 [iot_thre
a]
49 8293 [iot_threa] [INFO] [MQTT] [core_mqtt.c:1162] 50 8299 [iot_threa] Ack packet deserialized with result: MQTTSuccess.
51 8304 [iot_threa]
52 8307 [iot_threa] [INFO] [MQTT] [core_mqtt.c:1175] 53 8311 [iot_threa] State record updated. New state=MQTTPublishDone.5
4 8317 [iot_threa]
55 8318 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:1031] 56 8326 [iot_threa] PUBACK received for p
acket Id 2.57 8330 [iot_threa]
58 8333 [iot_threa] [INFO] [MQTT] [core_mqtt.c:886] 59 8336 [iot_threa] Packet received. ReceivedBytes=39.60 8342 [iot_thr
ea]
```

...

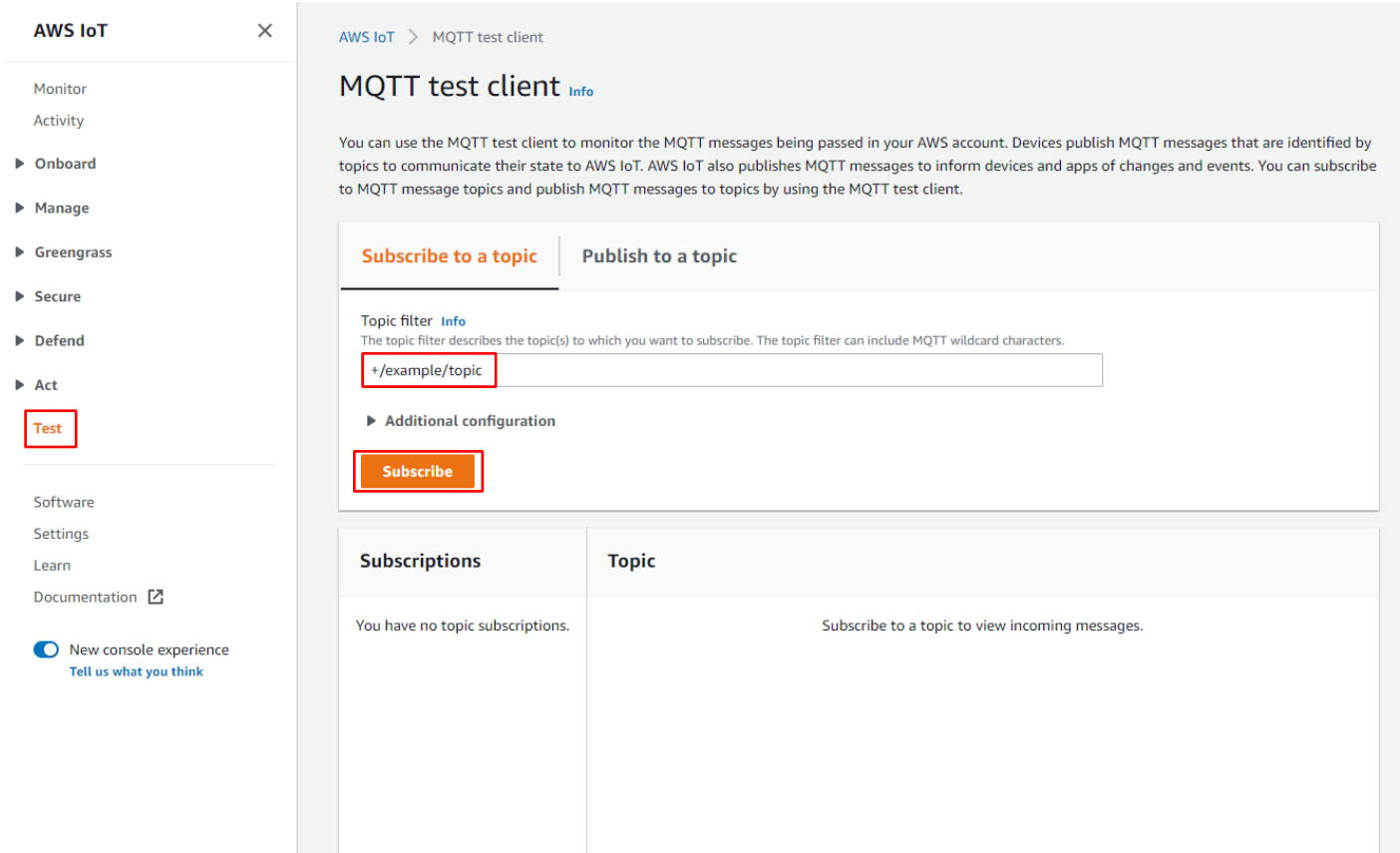
```
COM3:115200baud - Tera Term VT
File  Edit  Setup  Control  Window  Help
Topic Name: ameba-ota/example/topic matches subscribed topic.Incoming Publish Message : Hello World!698 68052 [iot_threa]
699 68554 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:553] 700 68559 [iot_threa] Keeping Connection
 Idle...701 68565 [iot_threa]
702 70566 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:561] 703 70571 [iot_threa] Unsubscribe from t
he MQTT topic ameba-ota/example/topic.704 70579 [iot_threa]
705 70707 [iot_threa] [INFO] [MQTT] [core_mqtt.c:886] 706 70710 [iot_threa] Packet received. ReceivedBytes=2.707 70716 [io
t_threa]
708 70717 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:1062] 709 70725 [iot_threa] Unsubscribed from
 the topic ameba-ota/example/topic.710 70730 [iot_threa]
711 71733 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:583] 712 71738 [iot_threa] Disconnecting the
MQTT connection with a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.713 71749 [iot_threa]
714 71752 [iot_threa] [INFO] [MQTT] [core_mqtt.c:2149] 715 71756 [iot_threa] Disconnected from the broker.716 71760 [iot_t
hrea]
717 71763 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:612] 718 71770 [iot_threa] Demo completed an
iteration successfully.719 71775 [iot_threa]
720 71778 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:613] 721 71784 [iot_threa] Demo iteration 3 c
ompleted successfully.722 71790 [iot_threa]
723 71791 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:625] 724 71799 [iot_threa] Short delay before
 starting the next iteration.... 725 71805 [iot_threa]
726 76808 [iot_threa] [INFO] [MQTT_MutualAuth_Demo] [mqtt_demo_mutual_auth.c:636] 727 76813 [iot_threa] Demo run is succes
sful with 3 successful loops out of total 3 loops.728 76822 [iot_threa]
729 77823 [iot_threa] [INFO ][DEMO][77823] Demo completed successfully.

LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
[deinit_timer_wrapper] Need to delete 1 timer_entry
WIFI deinitialized730 78085 [iot_threa] [INFO ][INIT][78085] SDK cleanup done.
731 78090 [iot_threa] [INFO ][DEMO][78090] -------DEMO FINISHED-------
```

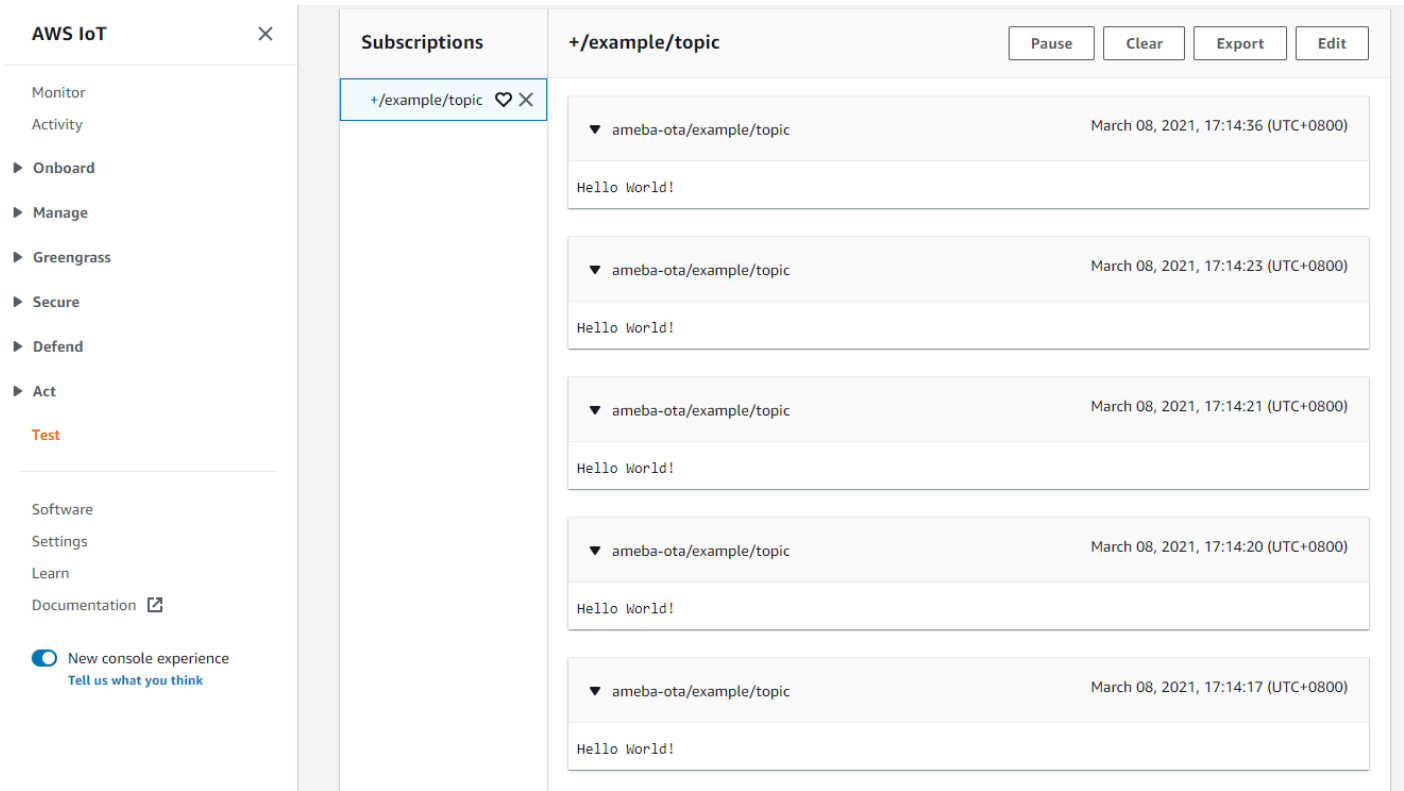
Monitor connection summary.

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 “+/example/topic”, 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 ERROR: Invalid Key

Please check **WIFI_SSID** and **WIFI_PASSWORD** in in "component/common/application/amazon/amazon-freertos-202012.00/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 ...
```

7.2 Failed to establish new MQTT connection

Please check **clientcredentialMQTT_BROKER_ENDPOINT** in "component/common/application/amazon/amazon-freertos-202012.00/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-------
```

7.3 TLS_Connect fail

Please check **keyCLIENT_CERTIFICATE_PEM** and **keyCLIENT_PRIVATE_KEY_PEM** in "component/common/application/amazon/amazon-freertos-202012.00/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-------
```