



AmebaDplus 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

COPYRIGHT

©2019 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

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 Configure AWS IoT Core

1.1 Set up your AWS account and Permissions

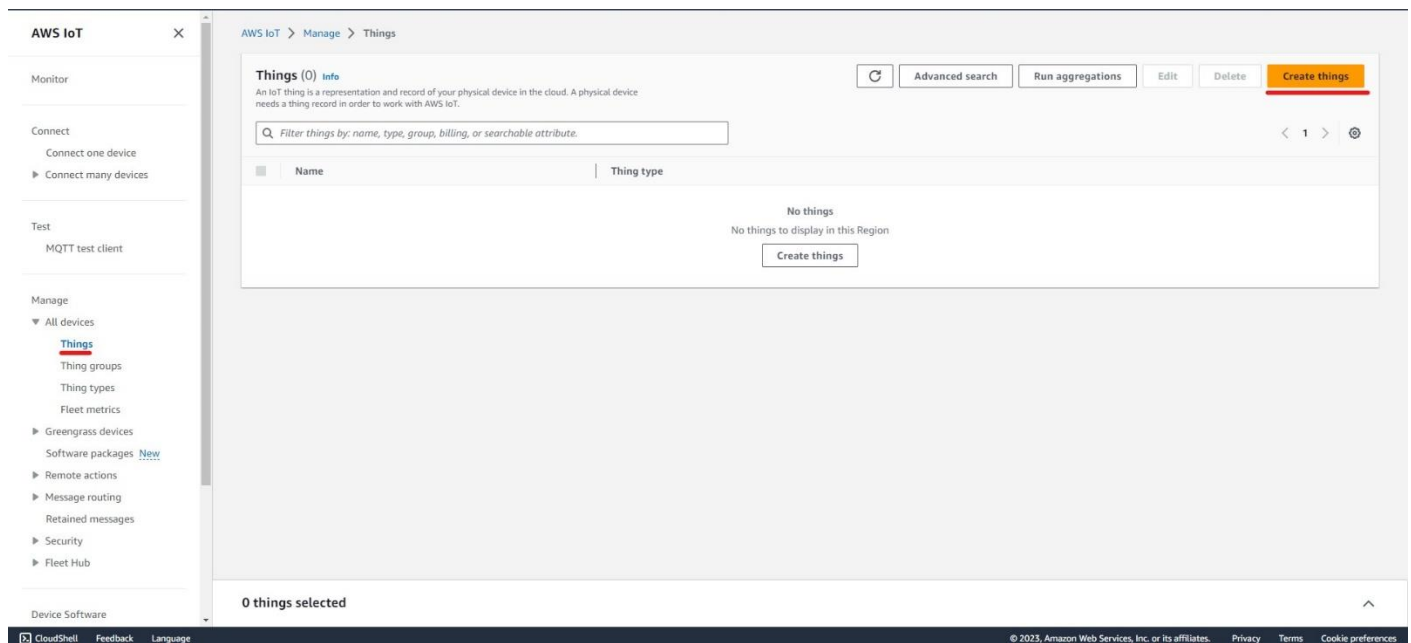
Refer to the instructions at Set up your AWS Account <https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html>. Follow the steps outlined in these sections to create your account and a user and get started:

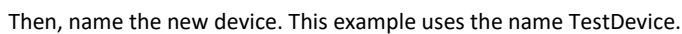
- Sign up for an AWS account (<https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html#sign-up-for-aws>)
- Create an administrative user (<https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html#create-an-admin>)
- Open the AWS IoT console(<https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html#iot-console-signin>)

Please pay special attention to the Notes in AWS webpage.

1.2 Create a New Device

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





Select "Auto-generate a new certificate (recommended)"

Getting Started Guide

1.3 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"

The screenshot shows the AWS IoT console interface. On the left, a sidebar contains navigation links: 'Specify thing properties', 'Configure device certificate', and 'Attach policies to certificate'. The main content area is titled 'Attach policies to certificate - optional' with an 'Info' link. Below the title, a text box explains that AWS IoT policies grant or deny access to AWS IoT resources. A section titled 'Policies (0)' indicates that up to 10 policies can be attached. A search bar labeled 'Filter policies' is present. Below the search bar, a table with the header 'Name' is shown, but it is empty, displaying 'No policies' and a message 'No policies could be found in ap-northeast-2.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create thing'.

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>.

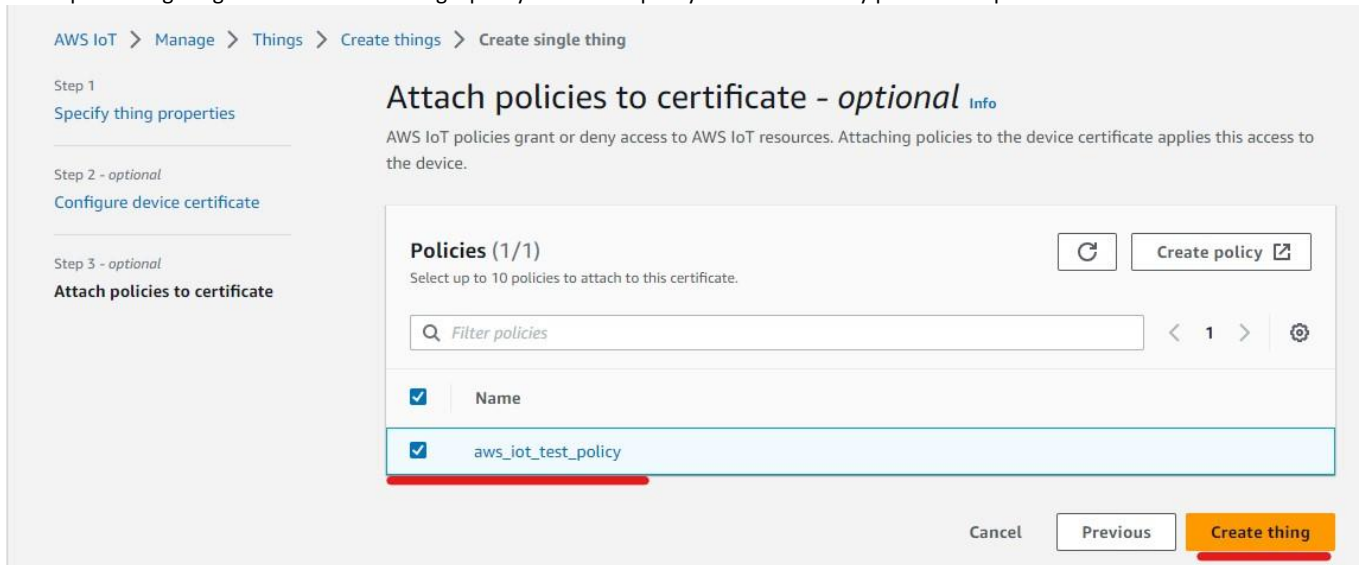
Also refer to <https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html>

The screenshot shows the 'Create policy' page in the AWS IoT console. The left sidebar has a 'Security' section expanded, showing 'Policies' as the selected option. The main content area is titled 'Create policy' with an 'Info' link. Below the title, a text box explains that AWS IoT Core policies allow you to manage access to the AWS IoT Core data plane operations. A section titled 'Policy properties' contains a 'Policy name' field with the value 'aws_iot_test_policy' and a note that a policy name is an alphanumeric string. Below this is a 'Tags - optional' section. The 'Policy statements' tab is selected, showing a 'Policy document' section. This section includes a 'Policy effect' dropdown set to 'Allow', a 'Policy action' dropdown set to '*', and a 'Policy resource' dropdown set to '*'. There is a 'Remove' button next to the resource field and an 'Add new statement' button. At the bottom right, there are 'Cancel' and 'Create' buttons.

Important: The policy is for reference. Please identify the permission policies that best meet your business and security requirements

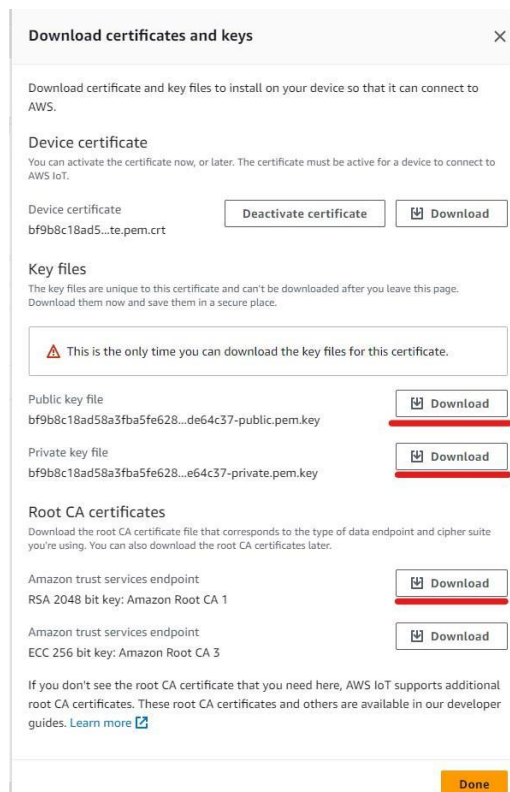
1.4 Attach Policy

The last step to configuring the device is attaching a policy. Select the policy which created by previous step.



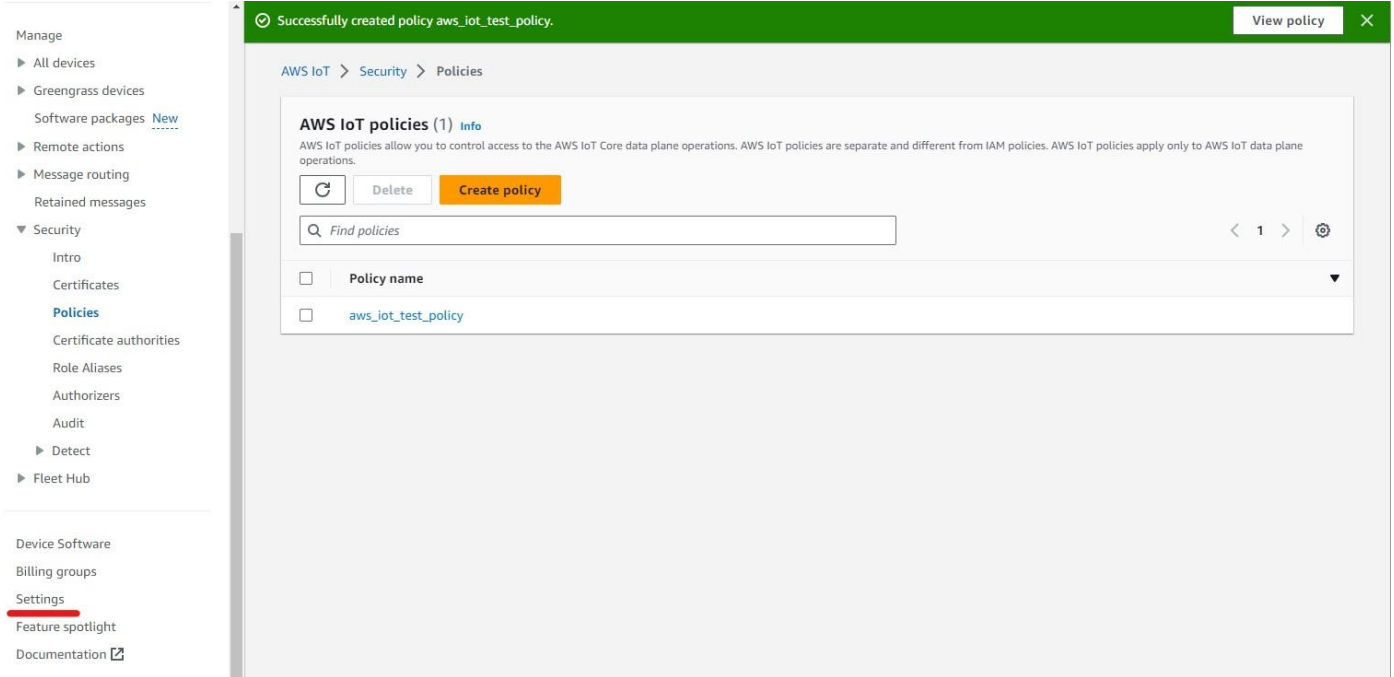
1.5 Download Certificates

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

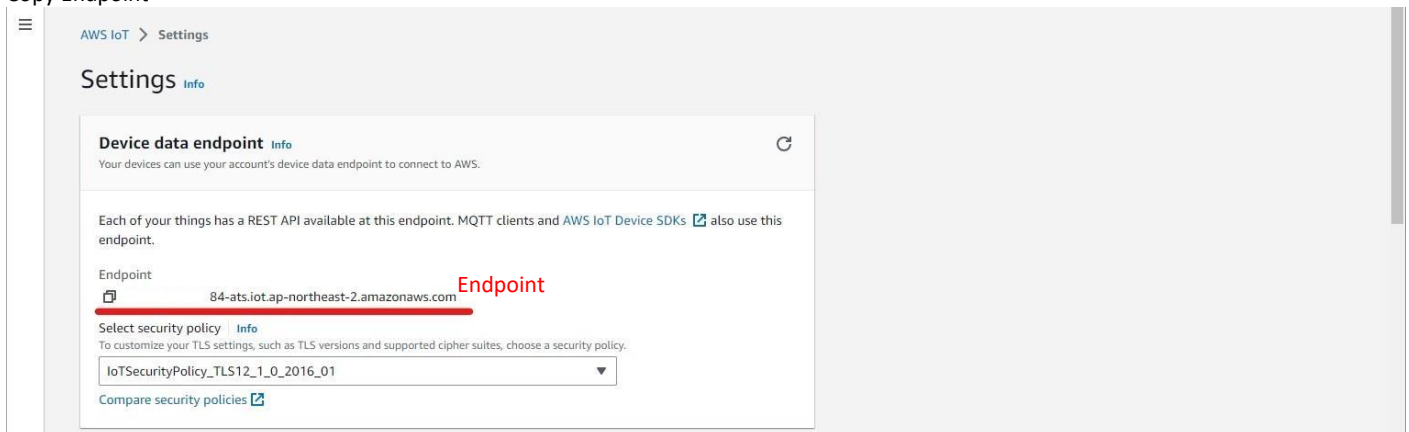


2 Configure AmebaDplus Amazon FreeRTOS

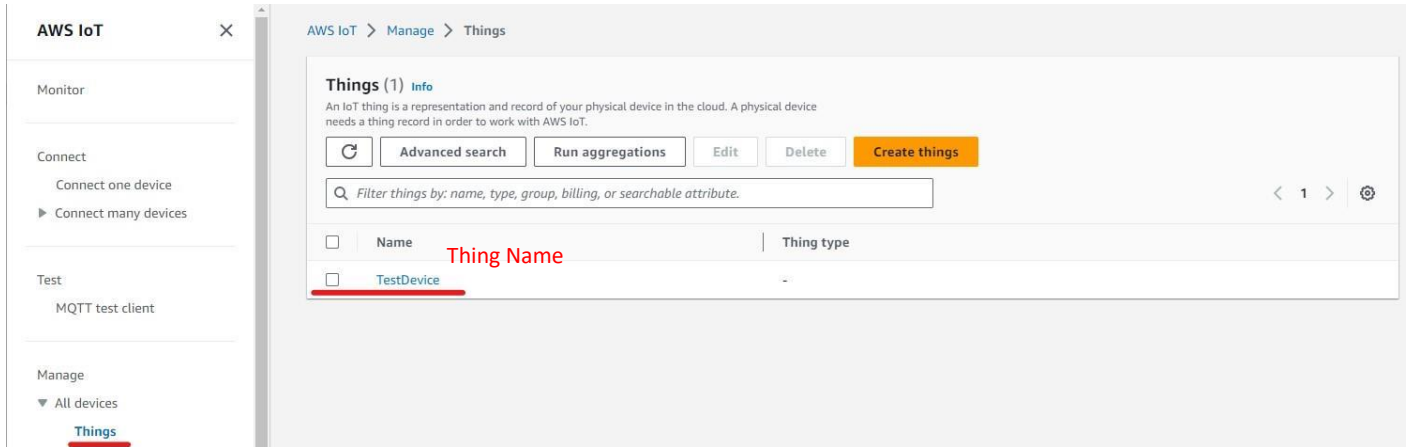
2.1 Get Endpoint by AWS IoT Core



Copy Endpoint



2.2 Get Thing Name



2.3 Setup IoT Core Information with AmebaDplus Amazon FreeRTOS

Setup BROKER_ENDPOINT, THING_NAME, WIFI_SSID, PASSWORD in "component/application/amazon/amazon-freertos/demos/include/aws_clientcredential.h"

```
#define clientcredentialMQTT_BROKER_ENDPOINT      "xxxxxxxxxxxxx.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__ */
```

Fill keyCLIENT_CERTIFICATE_PEM and keyCLIENT_PRIVATE_KEY_PEM in “component/application/amazon/amazon-freertos/demos/include/aws_clientcredential_keys.h” by xxxxxxxx-public.pem.key and xxxxxxxx-private.pem.key.

It can be done by "component/application/amazon/amazon-freertos/tools/certificate_configuration/CertificateConfigurator.html"

Final aws_clientcredential_keys.h overview.

Getting Started Guide

2.3.2 Enable FreeRTOS demo on AmebaDplus

Find `aws_main.c` in `component/application/amazon/amazon-freertos/ports/amebaDplus/aws_main.c` and enable **RunCoreMqttMutualAuthDemo**

```
int aws_main( void )
{
    /* Create tasks that are not dependent on the Wi-Fi being initialized. */
    xLoggingTaskInitialize( mainLOGGING_TASK_STACK_SIZE,
                           tskIDLE_PRIORITY+6,
                           mainLOGGING_MESSAGE_QUEUE_LENGTH );

    CRYPTO_ConfigureThreading();

    //mqtt mutual auto demo
    RunCoreMqttMutualAuthDemo(0, NULL, NULL, NULL, NULL);

    //http mutual auto demo
    //RunCoreHttpMutualAuthDemo(0, NULL, NULL, NULL, NULL);

    //device shadow demo
    //RunDeviceShadowDemo(0, NULL, NULL, NULL, NULL);

    //device defender demo
    //RunDeviceDefenderDemo(0, NULL, NULL, NULL, NULL);

    // ota over mqtt demo
    //RunOtaCoreMqttDemo(0, NULL, NULL, NULL, NULL);

    return 0;
}
```

Now you can start to compile AmebaDplus Amazon FreeRTOS

3 MQTT Demo

3.1 Get Device Log

Install Tera Term to get device log

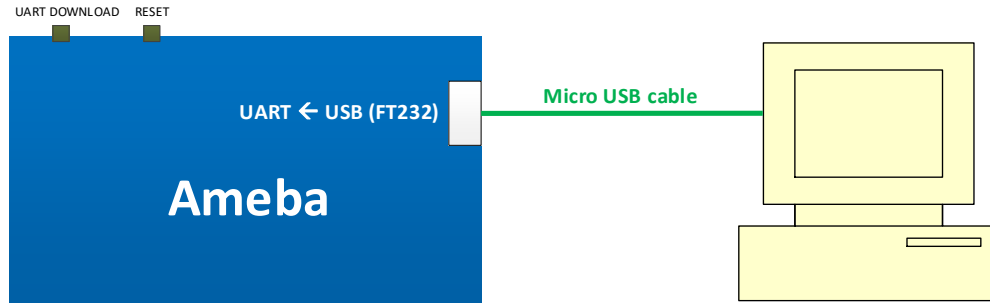
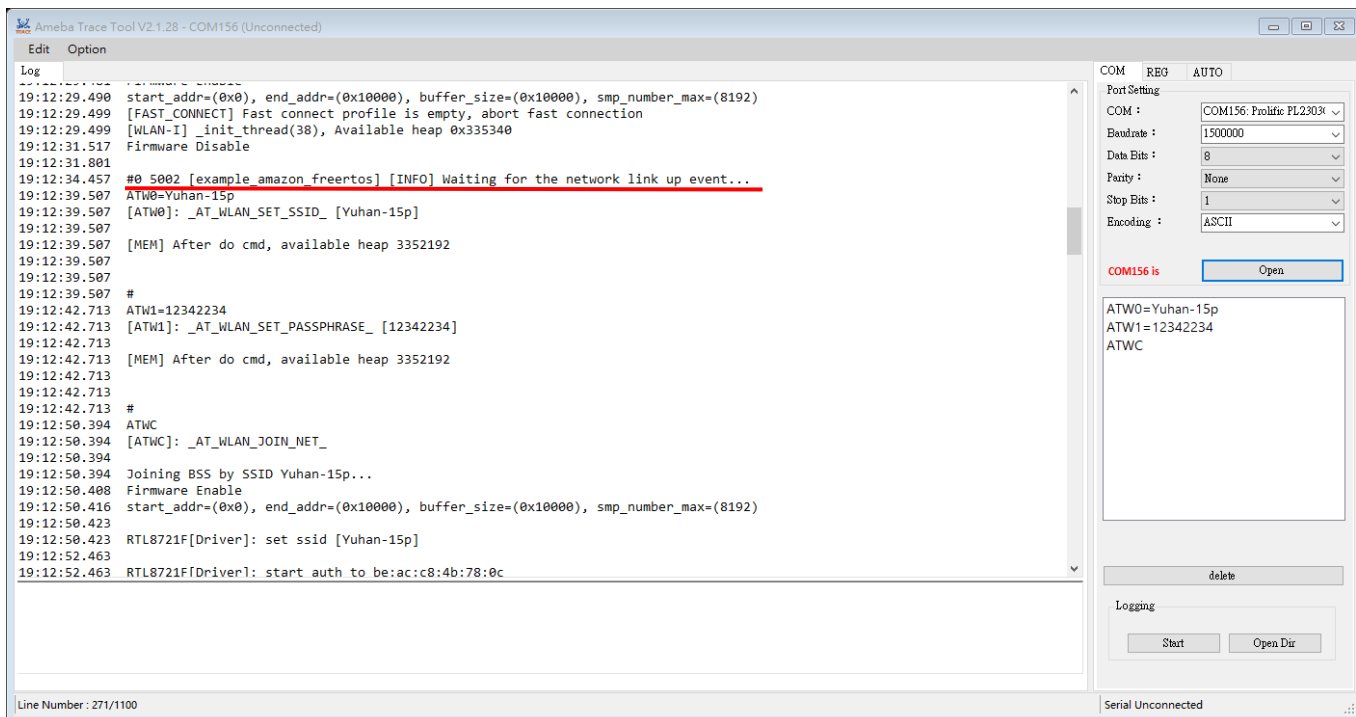


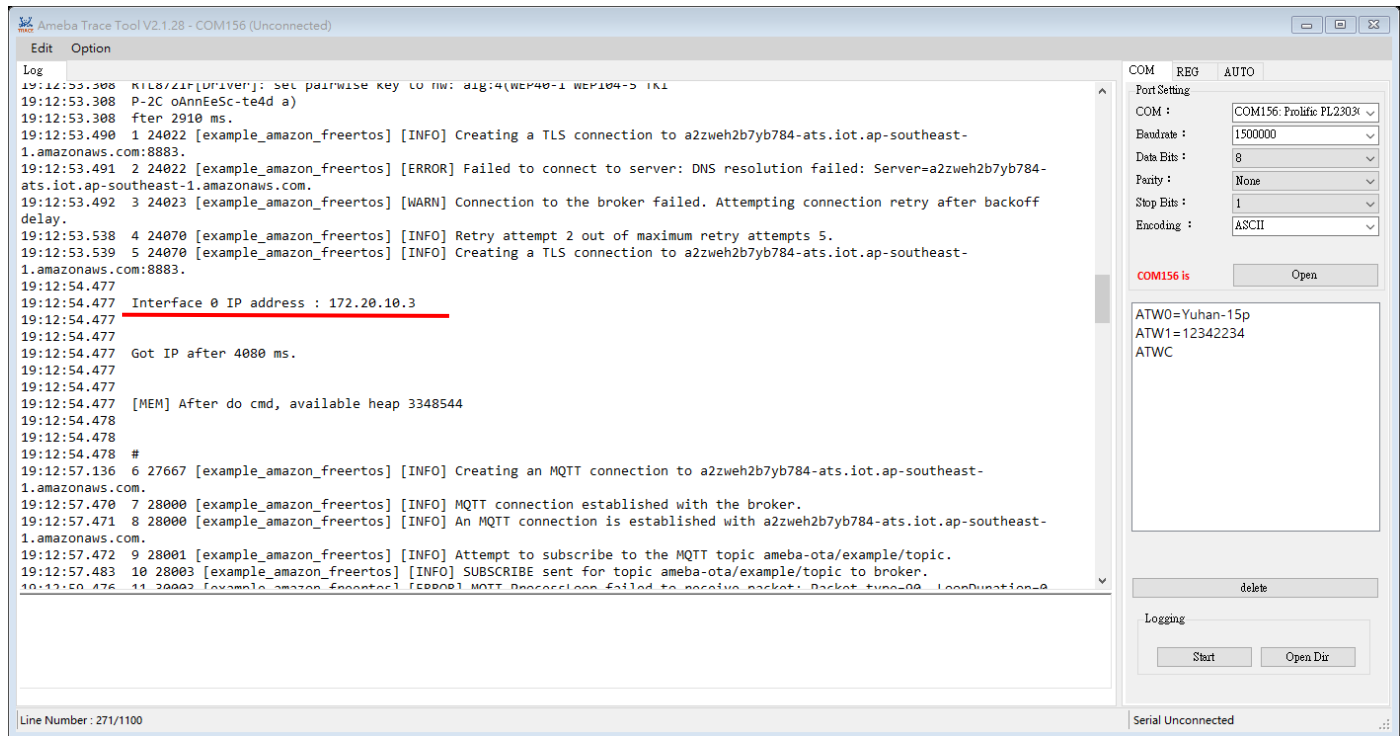
Fig 3-1 Hardware setup

3.2 Run MQTT Demo

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



Use ATcmd to connect wifi.



Ameba Trace Tool V2.1.28 - COM156 (Unconnected)

Log

```

19:12:53.308 KIL6ZIF[Driver]: set pairwise key to mw: a1g:4(WEP40-1 WEP104-3 TK1
19:12:53.308 P-2C oAnnEeSc-te4d a)
19:12:53.308 fter 2910 ms.
19:12:53.490 1 24022 [example_amazon_freertos] [INFO] Creating a TLS connection to a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com:8883.
19:12:53.491 2 24022 [example_amazon_freertos] [ERROR] Failed to connect to server: DNS resolution failed: Server=a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.
19:12:53.492 3 24023 [example_amazon_freertos] [WARN] Connection to the broker failed. Attempting connection retry after backoff delay.
19:12:53.538 4 24070 [example_amazon_freertos] [INFO] Retry attempt 2 out of maximum retry attempts 5.
19:12:53.539 5 24070 [example_amazon_freertos] [INFO] Creating a TLS connection to a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com:8883.
19:12:54.477 Interface 0 IP address : 172.20.10.3
19:12:54.477 Got IP after 4080 ms.
19:12:54.477 [MEM] After do cmd, available heap 3348544
19:12:54.478 #
19:12:57.136 6 27667 [example_amazon_freertos] [INFO] Creating an MQTT connection to a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.
19:12:57.470 7 28000 [example_amazon_freertos] [INFO] MQTT connection established with the broker.
19:12:57.471 8 28000 [example_amazon_freertos] [INFO] An MQTT connection is established with a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.
19:12:57.472 9 28001 [example_amazon_freertos] [INFO] Attempt to subscribe to the MQTT topic ameba-ota/example/topic.
19:12:57.483 10 28003 [example_amazon_freertos] [INFO] SUBSCRIBE sent for topic ameba-ota/example/topic to broker.
19:12:57.476 11 30003 [example_amazon_freertos] [ERROR] MQTT_ProcessLoop failed to receive packet: Packet type=90, LoopDuration=0, Status=MQTTSuccess
19:12:59.477 12 30004 [example_amazon_freertos] [WARN] Server rejected subscription request. Attempting to re-subscribe to topic ameba-ota/example/topic.
19:12:59.780 13 30308 [example_amazon_freertos] [INFO] Retry attempt 2 out of maximum retry attempts 5.
19:12:59.781 14 30308 [example_amazon_freertos] [INFO] Attempt to subscribe to the MQTT topic ameba-ota/example/topic.
19:12:59.781 15 30309 [example_amazon_freertos] [INFO] SUBSCRIBE sent for topic ameba-ota/example/topic to broker.
19:12:59.975 16 30503 [example_amazon_freertos] [INFO] Subscribed to the topic ameba-ota/example/topic with maximum QoS 1.
19:12:59.975 17 30503 [example_amazon_freertos] [INFO] Publish to the MQTT topic ameba-ota/example/topic.
19:12:59.976 18 30504 [example_amazon_freertos] [INFO] Attempt to receive publish message from broker.
19:13:00.080 19 30604 [example_amazon_freertos] [INFO] Subscribed to the topic ameba-ota/example/topic with maximum QoS 1.
19:13:00.191 20 30718 [example_amazon_freertos] [INFO] Ack packet deserialized with result: MQTTSuccess.
19:13:00.192 21 30719 [example_amazon_freertos] [INFO] State record updated. New state=MQTTPublishDone.
19:13:00.192 22 30719 [example_amazon_freertos] [INFO] PUBACK received for packet Id 2.
19:13:00.268 23 30796 [example_amazon_freertos] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
19:13:00.269 24 30796 [example_amazon_freertos] [INFO] State record updated. New state=MQTTPubAckSend.
19:13:00.269 25 30797 [example_amazon_freertos] [INFO] Incoming QoS : 1
19:13:00.269 Incoming Publish Topic Name: ameba-ota/example/topic matches subscribed topic.Incoming Publish Message : Hello World!
19:13:00.271 27 30799 [example_amazon_freertos] [INFO] Keeping Connection Idle...
19:13:00.272 28 31700 [example_amazon_freertos] [INFO] Publish to the MQTT topic ameba-ota/example/topic

```

Line Number: 271/1100

Serial Unconnected

COM REG AUTO

Port Setting

COM : COM156: Prolific PL2303R

Baudrate : 1500000

Data Bits : 8

Parity : None

Stop Bits : 1

Encoding : ASCII

COM156 is Open

ATW0=Yuhan-15p

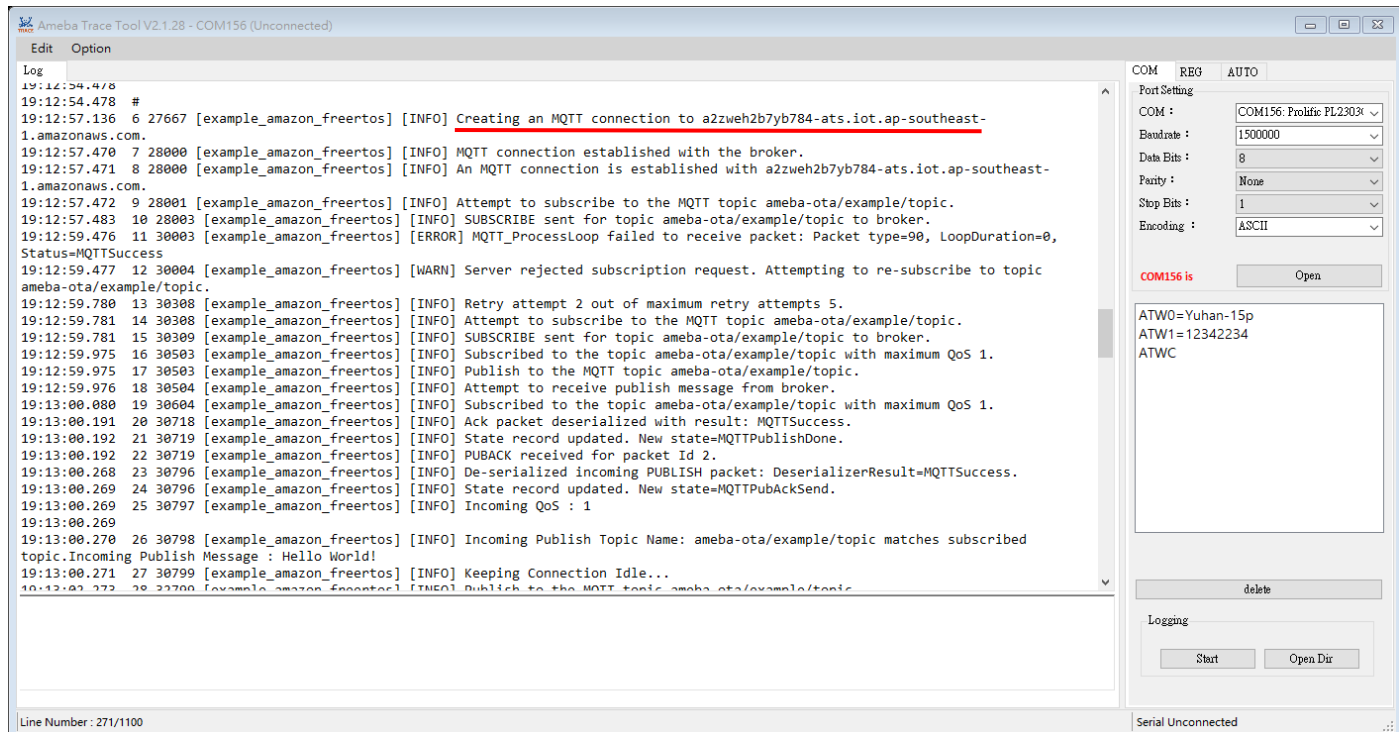
ATW1=12342234

ATWC

delete

Logging

Start Open Dir



Ameba Trace Tool V2.1.28 - COM156 (Unconnected)

Log

```

19:12:54.478 #
19:12:57.136 6 27667 [example_amazon_freertos] [INFO] Creating an MQTT connection to a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.
19:12:57.470 7 28000 [example_amazon_freertos] [INFO] MQTT connection established with the broker.
19:12:57.471 8 28000 [example_amazon_freertos] [INFO] An MQTT connection is established with a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.
19:12:57.472 9 28001 [example_amazon_freertos] [INFO] Attempt to subscribe to the MQTT topic ameba-ota/example/topic.
19:12:57.483 10 28003 [example_amazon_freertos] [INFO] SUBSCRIBE sent for topic ameba-ota/example/topic to broker.
19:12:59.476 11 30003 [example_amazon_freertos] [ERROR] MQTT_ProcessLoop failed to receive packet: Packet type=90, LoopDuration=0, Status=MQTTSuccess
19:12:59.477 12 30004 [example_amazon_freertos] [WARN] Server rejected subscription request. Attempting to re-subscribe to topic ameba-ota/example/topic.
19:12:59.780 13 30308 [example_amazon_freertos] [INFO] Retry attempt 2 out of maximum retry attempts 5.
19:12:59.781 14 30308 [example_amazon_freertos] [INFO] Attempt to subscribe to the MQTT topic ameba-ota/example/topic.
19:12:59.781 15 30309 [example_amazon_freertos] [INFO] SUBSCRIBE sent for topic ameba-ota/example/topic to broker.
19:12:59.975 16 30503 [example_amazon_freertos] [INFO] Subscribed to the topic ameba-ota/example/topic with maximum QoS 1.
19:12:59.975 17 30503 [example_amazon_freertos] [INFO] Publish to the MQTT topic ameba-ota/example/topic.
19:12:59.976 18 30504 [example_amazon_freertos] [INFO] Attempt to receive publish message from broker.
19:13:00.080 19 30604 [example_amazon_freertos] [INFO] Subscribed to the topic ameba-ota/example/topic with maximum QoS 1.
19:13:00.191 20 30718 [example_amazon_freertos] [INFO] Ack packet deserialized with result: MQTTSuccess.
19:13:00.192 21 30719 [example_amazon_freertos] [INFO] State record updated. New state=MQTTPublishDone.
19:13:00.192 22 30719 [example_amazon_freertos] [INFO] PUBACK received for packet Id 2.
19:13:00.268 23 30796 [example_amazon_freertos] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
19:13:00.269 24 30796 [example_amazon_freertos] [INFO] State record updated. New state=MQTTPubAckSend.
19:13:00.269 25 30797 [example_amazon_freertos] [INFO] Incoming QoS : 1
19:13:00.269 Incoming Publish Topic Name: ameba-ota/example/topic matches subscribed topic.Incoming Publish Message : Hello World!
19:13:00.271 27 30799 [example_amazon_freertos] [INFO] Keeping Connection Idle...
19:13:00.272 28 31700 [example_amazon_freertos] [INFO] Publish to the MQTT topic ameba-ota/example/topic

```

Line Number: 271/1100

Serial Unconnected

COM REG AUTO

Port Setting

COM : COM156: Prolific PL2303R

Baudrate : 1500000

Data Bits : 8

Parity : None

Stop Bits : 1

Encoding : ASCII

COM156 is Open

ATW0=Yuhan-15p

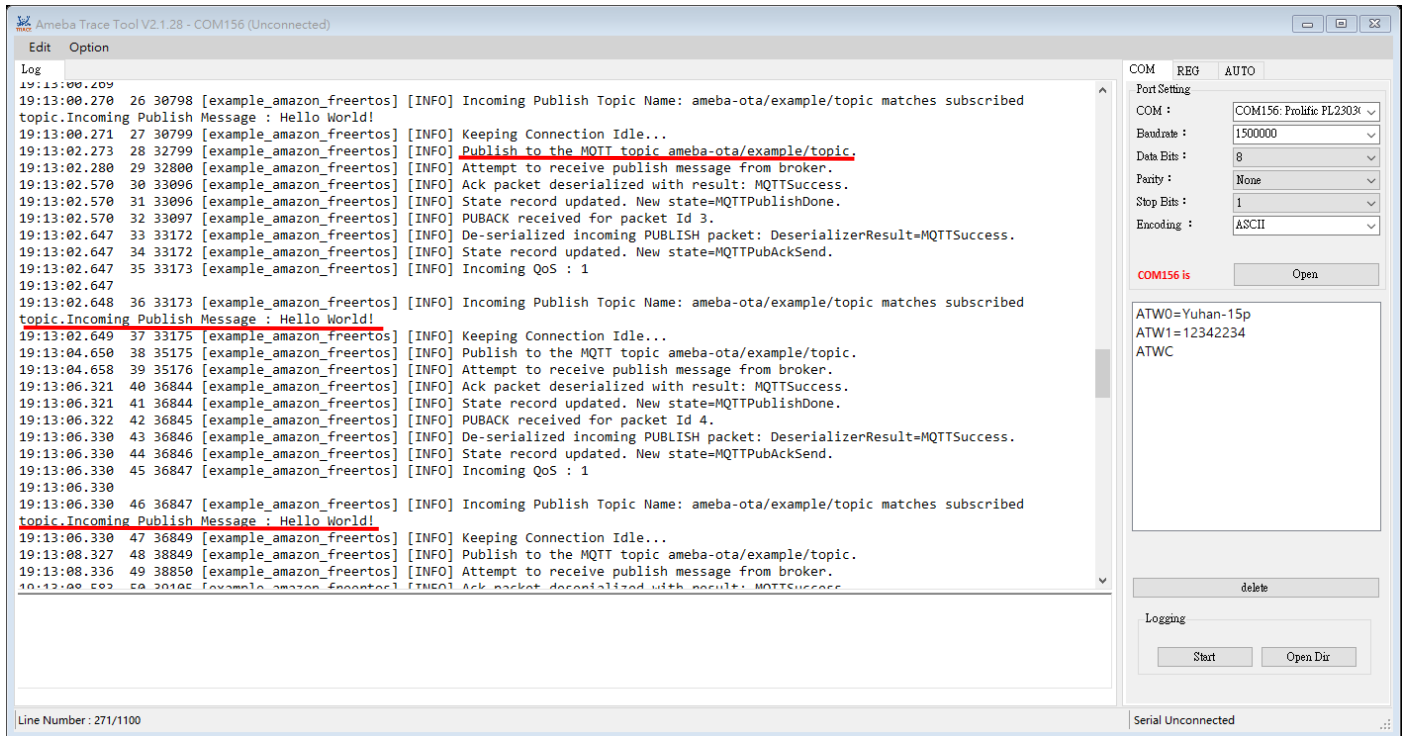
ATW1=12342234

ATWC

delete

Logging

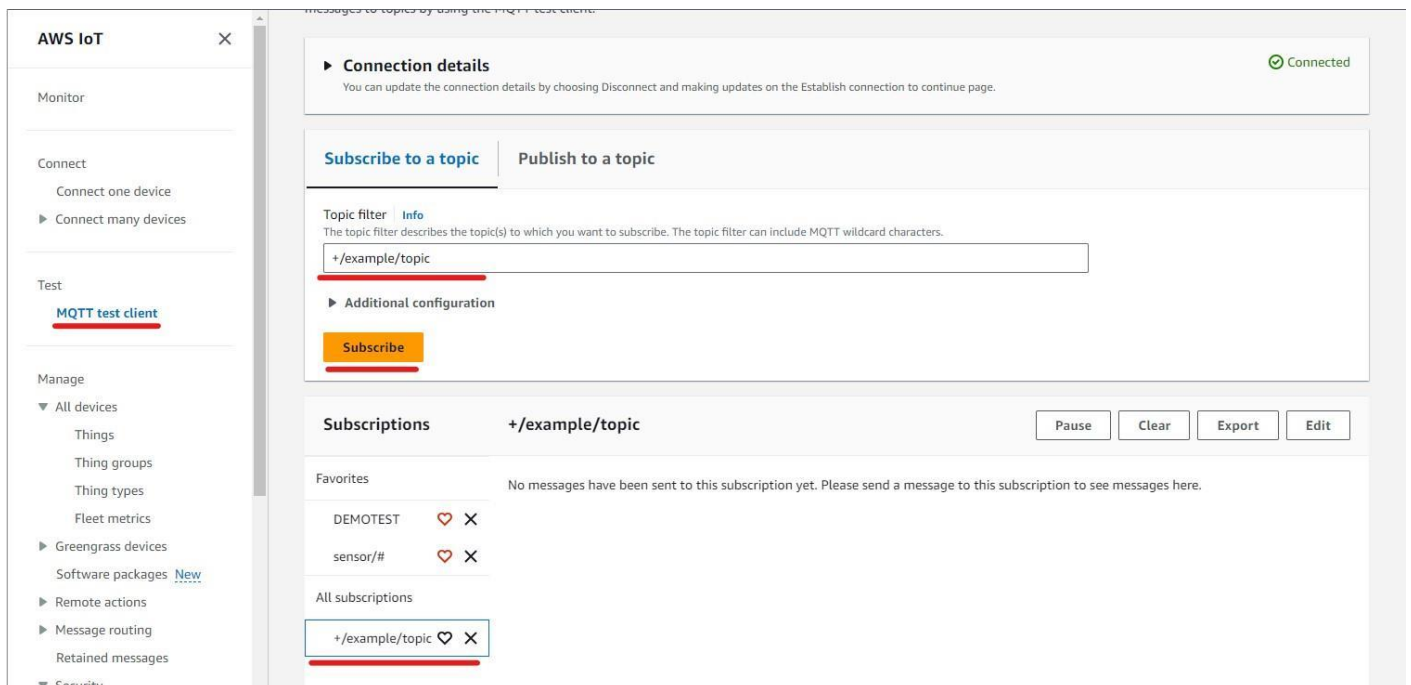
Start Open Dir



3.3 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.



AWS IoT

Monitor

Connect

Connect one device

Connect many devices

Test

MQTT test client

Manage

All devices

Greengrass devices

Software packages [New](#)

Remote actions

Message routing

Retained messages

Security

Subscriptions

+ /example/topic

Pause

Clear

Export

Edit

Favorites

sensor/#

DEMOTEST

+ /example/topic

▼ ameba-ota/example/topic

July 14, 2023, 13:16:19 (UTC+0800)

Message cannot be displayed in specified format.

Hello World!

Properties

▼ ameba-ota/example/topic

July 14, 2023, 13:16:16 (UTC+0800)

Message cannot be displayed in specified format.


```

/**
 * @brief Major version of the firmware.
 *
 * This is used in the OTA demo to set the appFirmwareVersion variable
 * that is declared in the ota_appversion32.h file in the OTA library.
 */
#ifndef APP_VERSION_MAJOR
#define APP_VERSION_MAJOR    0 /* TODO */
#endif

/**
 * @brief Minor version of the firmware.
 *
 * This is used in the OTA demo to set the appFirmwareVersion variable
 * that is declared in the ota_appversion32.h file in the OTA library.
 */
#ifndef APP_VERSION_MINOR
#define APP_VERSION_MINOR    9 /* TODO */
#endif

/**
 * @brief Build version of the firmware.
 *
 * This is used in the OTA demo to set the appFirmwareVersion variable
 * that is declared in the ota_appversion32.h file in the OTA library.
 */
#ifndef APP_VERSION_BUILD
#define APP_VERSION_BUILD    2 /* TODO */
#endif

```

Also, change manifest. Please find more detail about manifest in application note.

```

"//": "cert/app share IMG_ID/IMG_VER, rdp img is in app",
"app":
{
    "IMG_ID": "1",
    "IMG_VER_MAJOR": 1,
    "IMG_VER_MINOR": 1,
    "SEC_EPOCH": 1,

    "HASH_ALG": "sha256",

    "RSIP_IV": "213253647586a7b8"
},

```

Now, configure the `aws_demo_config.h` for OTA demo. Then, build the project and get the image file (.bin).

```
int aws_main( void )
{
    /* Create tasks that are not dependent on the Wi-Fi being initialized. */
    xLoggingTaskInitialize( mainLOGGING_TASK_STACK_SIZE,
                           tskIDLE_PRIORITY+6,
                           mainLOGGING_MESSAGE_QUEUE_LENGTH );

    CRYPTO_ConfigureThreading();

    //mqtt mutual auth demo
    //RunCoreMqttMutualAuthDemo(0, NULL, NULL, NULL, NULL);

    //http mutual auth demo
    //RunCoreHttpMutualAuthDemo(0, NULL, NULL, NULL, NULL);

    //device shadow demo
    //RunDeviceShadowDemo(0, NULL, NULL, NULL, NULL);

    //device defender demo
    //RunDeviceDefenderDemo(0, NULL, NULL, NULL, NULL);

    // ota over mqtt demo
    RunOtaCoreMqttDemo(0, NULL, NULL, NULL, NULL);

    return 0;
}
```

Please note that the newer image file must have the bigger app version number. So, now, you need two image file to perform this demo.

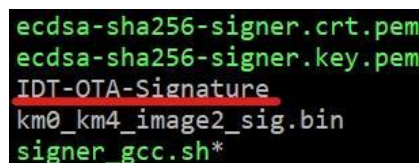
- One image with older version should be downloaded to your AmebaDplus, and wait the OTA job coming.
- Another image with newer version will be uploaded to S3 bucket. Then, create a new job for OTA.

Note: newer version image file should be signed by a private key before uploading. Next section will introduce how to sign the image.

4.3 How Custom Signed Image File is Created

The OTA binary (OTA_All.bin) will be generated when compile SDK and it locate at project/realtek_amebaDplus_va0_example/GCCRELEASE/auto_build/

The signature of OTA_All.bin can be got by shell script (component/application/amazon/amazon_ota_tools/signer_gcc.sh. The signature will need for create AWS IoT OTA jobs.



After getting the custom signed of auto_build/OTA_All.bin, you can upload it to the S3 bucket.

4.4 How to Trigger a Custom Signed OTA Job in Amazon AWS IOT Core

Go to AWS IoT Core <https://console.aws.amazon.com/iot?p=icr&cp=bn&ad=c>. Then, follow the following steps to create an AWS OTA task for AmebaDplus:

Step 1. Click on 'Create OTA update job', select your job type and then click next.

Monitor

Connect

Test

Manage

All devices

Things

Thing groups

Thing types

Fleet metrics

Greengrass devices

Software packages [New](#)

Remote actions

Jobs

Job templates

Secure tunnels

AWS IoT > Manage > Remote actions > Jobs

Jobs (6) Info

Jobs define a set of remote operations to send to and run on one or more devices that are connected to AWS IoT. If you have remote operations that are frequently performed, such as rebooting or installing new applications, use job templates to create reusable jobs.

↺

Edit

Cancel

Delete

Create job

🔍

Filter jobs

All status values

All types

<

1

>

🔊

<input type="checkbox"/>	Name	Type	Status	Created date
<input type="checkbox"/>	AFR_OTA-230714_08	Snapshot	✔ Completed	July 14, 2023, 10:40:49 (UTC+08:00)
<input type="checkbox"/>	AFR_OTA-230714_07	Snapshot	✔ Completed	July 14, 2023, 10:34:29 (UTC+08:00)
<input type="checkbox"/>	AFR_OTA-230714_06	Snapshot	✔ Completed	July 14, 2023, 10:32:42 (UTC+08:00)
<input type="checkbox"/>	AFR_OTA-230714_05	Snapshot	✔ Completed	July 14, 2023, 10:30:41 (UTC+08:00)
<input type="checkbox"/>	AFR_OTA-230714_04	Snapshot	✔ Completed	July 14, 2023, 10:23:47 (UTC+08:00)
<input type="checkbox"/>	AFR_OTA-230714_03	Snapshot	✔ Completed	July 14, 2023, 10:19:16 (UTC+08:00)

AWS IoT > Manage > Remote actions > Jobs > Create job

Create job Info

Jobs define remote operations to send to and run on devices that are connected to AWS IoT. Create a custom job or a FreeRTOS over-the-air (OTA) update job.

Job type

☐ Create custom job
Create a job to send an executable job file to one or more devices connected to AWS IoT.

☒ Create FreeRTOS OTA update job
Send a request to acquire an executable job file from one of your S3 buckets to one or more devices connected to AWS IoT.

Cancel

Next

Step 2. For Job properties, give a unique name to your OTA job, then click next.

Step 1

OTA job properties

Step 2

OTA file configuration

Step 3

OTA job configuration

OTA job properties [Info](#)

Job properties

Job name

Enter a unique name without spaces. Valid characters: a-z, A-Z, 0-9, - (hyphen), and _ (underscore)

Description - optional

► Tags - optional

Cancel

Next

Step 3. In the following page, choose your device to update and select the protocol for file transfer

Step 1

OTA job properties

Step 2

OTA file configuration

Step 3

OTA job configuration

OTA file configuration

Info

Devices

Info

This OTA update job will send your file securely over MQTT or HTTP to the FreeRTOS-based things and/or the thing groups that you choose.

Devices to update

Choose things and/or thing groups

TestDevice X

Select the protocol for file transfer

Select the protocol that your device supports.

☒ MQTT

☐ HTTP

Step 4. In the following page, choose the option '**Use my custom signed file**'.
In the signature field enter content of '**IDT-OTA-Signature**' which generated by **signer_gcc.sh**.
Choose hash algorithm as '**SHA-256**'. Choose encryption algorithm as '**ECDSA**'
In "pathname of code signing certificate", enter '/'

File

Info

Sign and choose your file

Code signing ensures that devices only run code published by trusted authors and that the code hasn't been changed or corrupted since it was signed. You have three options for code signing.

☐ Sign a new file for me.

☐ Choose a previously signed file.

☒ Use my custom signed file.

Code signing information

Enter information about your file and how it was signed so that your devices can verify its authenticity before they install it.

Signature

XX

Original hash algorithm

Choose the hash algorithm that was used to create your file signature.

SHA-256

Original encryption algorithm

Choose the encryption algorithm that was used to create your file signature.

ECDSA

Path name of code signing certificate on device

/

File

☒ Upload a new file.

☐ Select an existing file.

Step 5. Choose your custom signed firmware binary that was generated by the python script from S3 bucket.
In "Pathname of file on device", enter '/'

File to upload

Choose file

OTA_All.bin

843496 bytes

File upload location in S3

This is the location in S3 where your file will be stored.

S3 URL

s3://ameba-ota

×

View

Browse S3

Create S3 bucket

Format: s3://bucket/prefix/object.

Path name of file on device

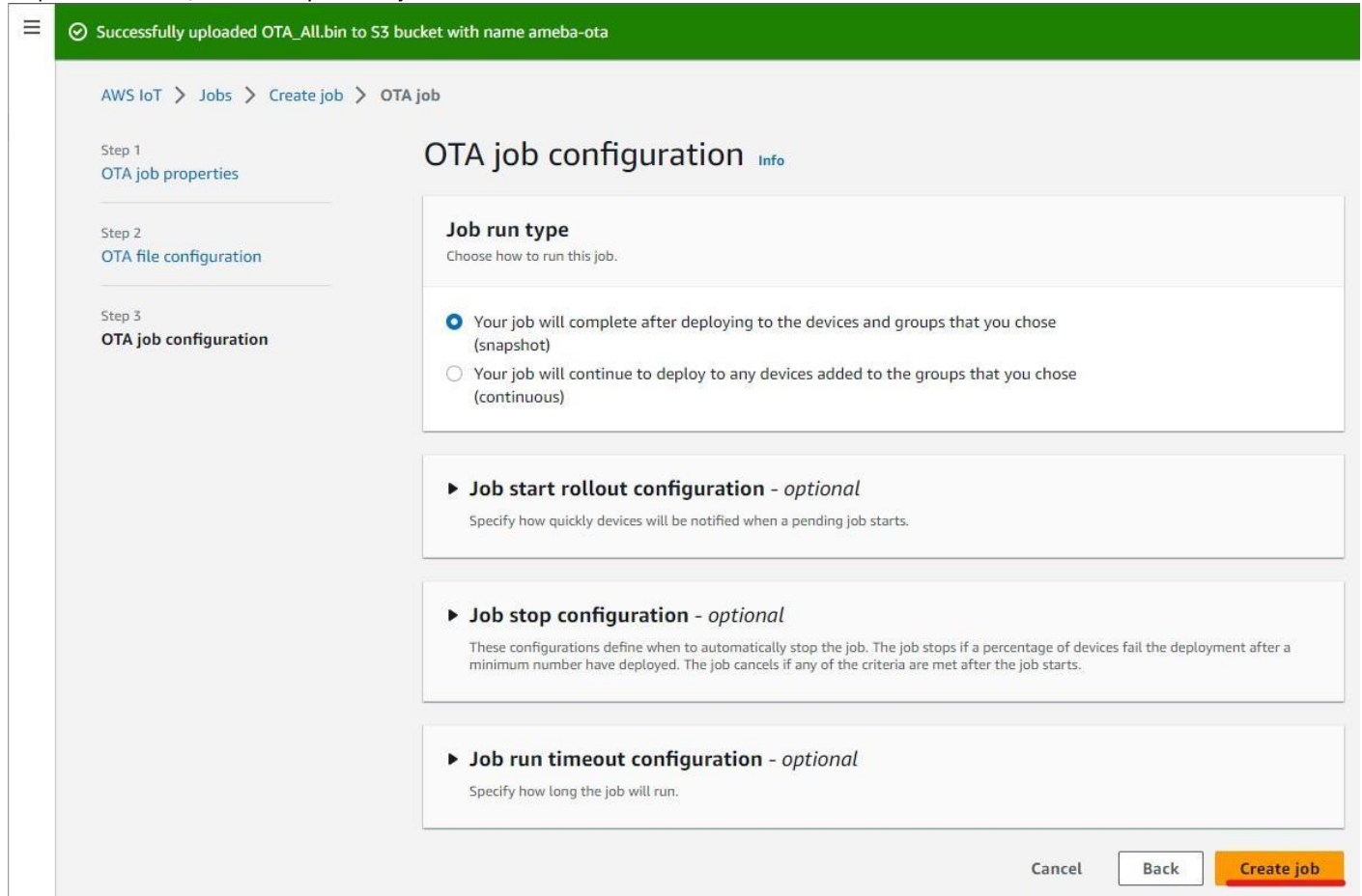
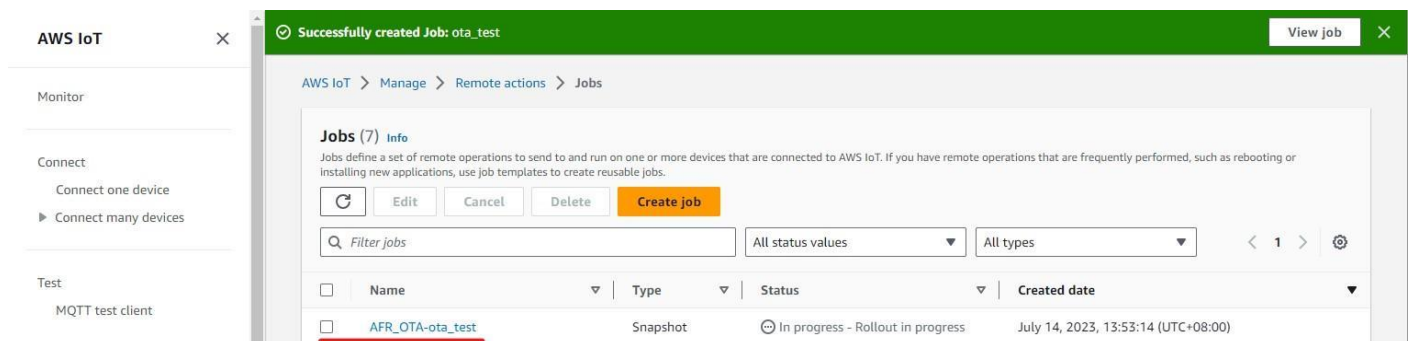
This is the name and location where the file will be stored on the FreeRTOS device.

/

Step 6. Choose the IAM role for OTA update job. (This is the same IAM role as any OTA update job)



Step 7. Click next, and create your OTA job.

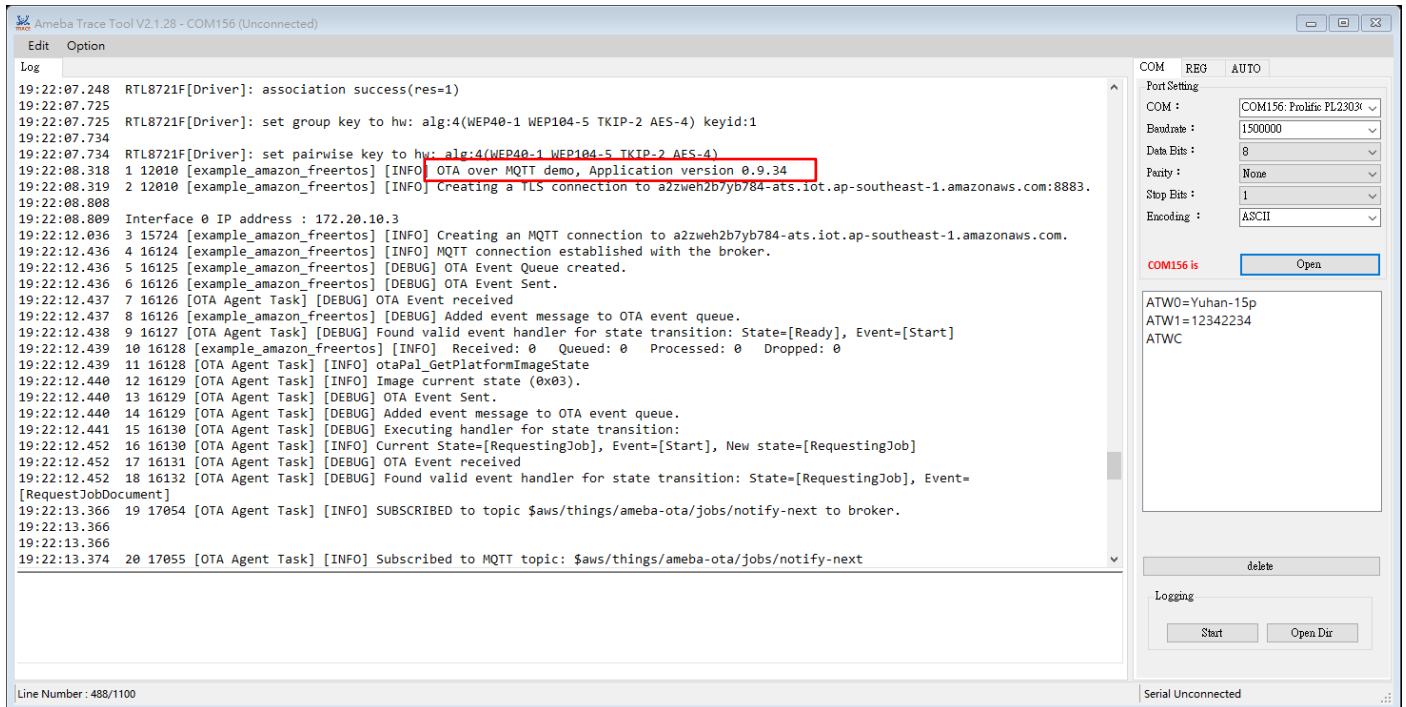



4.5 Run OTA Demo

Now we can see that the status of OTA job on AWS IoT Core is “in progress”. It means that it is waiting AmebaDplus to request the update.

Next, download the image file with older version number to AmebaDplus and then reboot the device, the application will automatically start to run the OTA demo.

In the beginning, we can check the app version (0.9.34) of this running firmware, and the OTA process by the job ID:



The screenshot shows the Ameba Trace Tool interface. The Log pane displays the following messages:

```

19:22:07.248 RTL8721F[Driver]: association success(res=1)
19:22:07.725 RTL8721F[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:1
19:22:07.734 RTL8721F[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4)
19:22:08.318 1 12010 [example_amazon_freertos] [INFO] OTA over MQTT demo, Application version 0.9.34
19:22:08.319 2 12010 [example_amazon_freertos] [INFO] Creating a TLS connection to a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com:8883.
19:22:08.808 Interface 0 IP address : 172.20.10.3
19:22:12.036 3 15724 [example_amazon_freertos] [INFO] Creating an MQTT connection to a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com.
19:22:12.436 4 16124 [example_amazon_freertos] [INFO] MQTT connection established with the broker.
19:22:12.436 5 16125 [example_amazon_freertos] [DEBUG] OTA Event Queue created.
19:22:12.436 6 16126 [example_amazon_freertos] [DEBUG] OTA Event Sent.
19:22:12.437 7 16126 [OTA Agent Task] [DEBUG] OTA Event received
19:22:12.437 8 16126 [example_amazon_freertos] [DEBUG] Added event message to OTA event queue.
19:22:12.438 9 16127 [OTA Agent Task] [DEBUG] Found valid event handler for state transition: State=[Ready], Event=[Start]
19:22:12.439 10 16128 [example_amazon_freertos] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped: 0
19:22:12.439 11 16128 [OTA Agent Task] [INFO] otaPal_GetPlatformImageState
19:22:12.440 12 16129 [OTA Agent Task] [INFO] Image current state (0x03).
19:22:12.440 13 16129 [OTA Agent Task] [DEBUG] OTA Event Sent.
19:22:12.440 14 16129 [OTA Agent Task] [DEBUG] Added event message to OTA event queue.
19:22:12.441 15 16130 [OTA Agent Task] [DEBUG] Executing handler for state transition:
19:22:12.452 16 16130 [OTA Agent Task] [INFO] Current State=[RequestingJob], Event=[Start], New state=[RequestingJob]
19:22:12.452 17 16131 [OTA Agent Task] [DEBUG] OTA Event received
19:22:12.452 18 16132 [OTA Agent Task] [DEBUG] Found valid event handler for state transition: State=[RequestingJob], Event=[RequestJobDocument]
19:22:13.366 19 17054 [OTA Agent Task] [INFO] SUBSCRIBED to topic $aws/things/ameba-ota/jobs/notify-next to broker.
19:22:13.366
19:22:13.374 20 17055 [OTA Agent Task] [INFO] Subscribed to MQTT topic: $aws/things/ameba-ota/jobs/notify-next

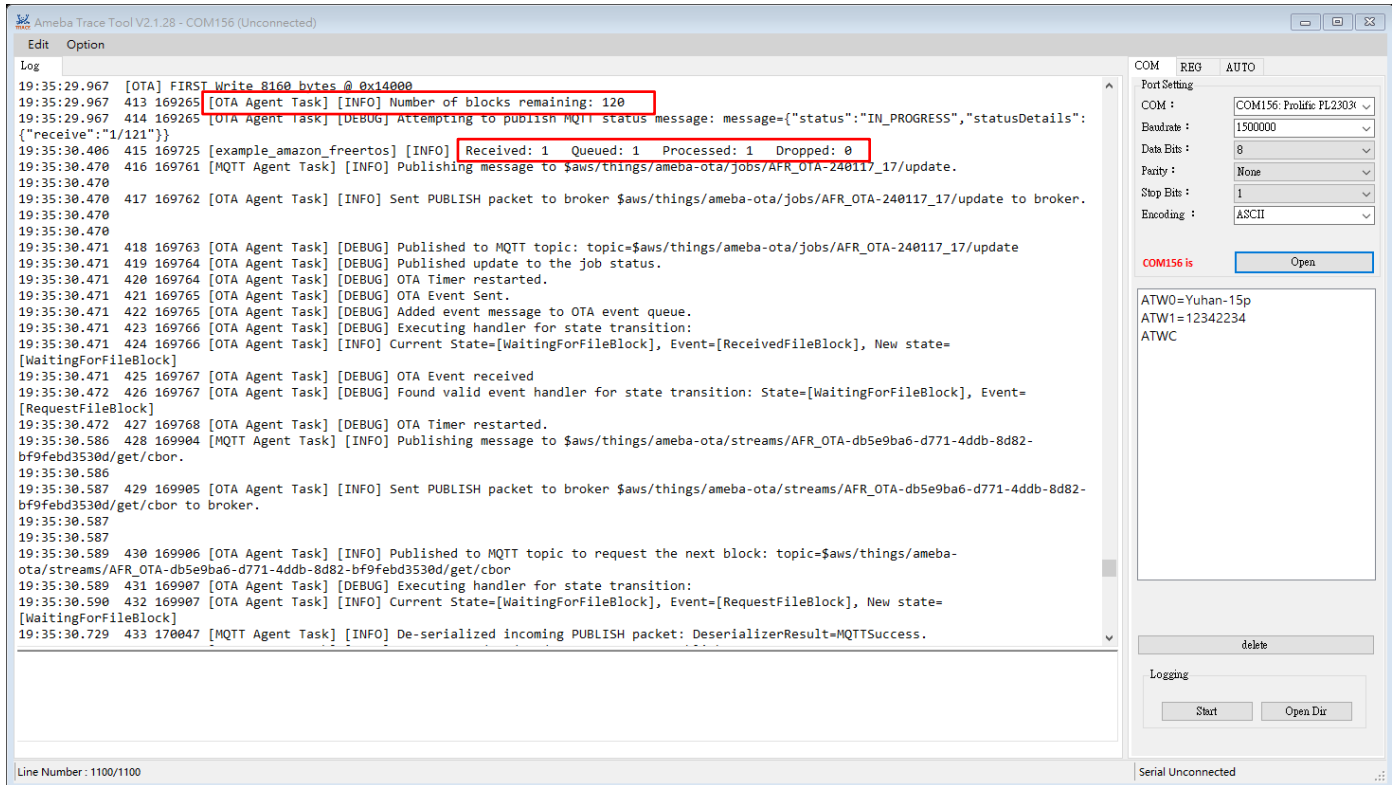
```

The Port Setting pane on the right shows the following configuration:

- COM: COM156, Prolific PL2303
- Baudrate: 1500000
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Encoding: ASCII

The ATW0=Yuhan-15p, ATW1=12342234, and ATWC are displayed below the port settings.

We can see that the OTA process start!



Log

```

19:35:29.967 [OTA] FIRST Write 8160 bytes @ 0x14000
19:35:29.967 413 169265 [OTA Agent Task] [INFO] Number of blocks remaining: 120
19:35:29.967 414 169265 [OTA Agent Task] [DEBUG] Attempting to publish MQTT status message: message={"status":"IN_PROGRESS","statusDetails":{"receive":"1/121"}}
19:35:30.406 415 169725 [example_amazon_freertos] [INFO] Received: 1 Queued: 1 Processed: 1 Dropped: 0
19:35:30.470 416 169761 [MQTT Agent Task] [INFO] Publishing message to $aws/things/ameba-ota/jobs/AFR_OTA-240117/update.
19:35:30.470 417 169762 [OTA Agent Task] [INFO] Sent PUBLISH packet to broker $aws/things/ameba-ota/jobs/AFR_OTA-240117/update to broker.
19:35:30.471 418 169763 [OTA Agent Task] [DEBUG] Published to MQTT topic: topic=$aws/things/ameba-ota/jobs/AFR_OTA-240117/update
19:35:30.471 419 169764 [OTA Agent Task] [DEBUG] Published update to the job status.
19:35:30.471 420 169764 [OTA Agent Task] [DEBUG] OTA Timer restarted.
19:35:30.471 421 169765 [OTA Agent Task] [DEBUG] OTA Event Sent.
19:35:30.471 422 169765 [OTA Agent Task] [DEBUG] Added event message to OTA event queue.
19:35:30.471 423 169766 [OTA Agent Task] [DEBUG] Executing handler for state transition:
19:35:30.471 424 169766 [OTA Agent Task] [INFO] Current State=[WaitingForFileBlock], Event=[ReceivedFileBlock], New state=[WaitingForFileBlock]
19:35:30.471 425 169767 [OTA Agent Task] [DEBUG] OTA Event received
19:35:30.472 426 169767 [OTA Agent Task] [DEBUG] Found valid event handler for state transition: State=[WaitingForFileBlock], Event=[RequestFileBlock]
19:35:30.472 427 169768 [OTA Agent Task] [DEBUG] OTA Timer restarted.
19:35:30.586 428 169984 [MQTT Agent Task] [INFO] Publishing message to $aws/things/ameba-ota/streams/AFR_OTA-db5e9ba6-d771-4ddb-8d82-bf9feb3530d/get/cbor.
19:35:30.586 429 169985 [OTA Agent Task] [INFO] Sent PUBLISH packet to broker $aws/things/ameba-ota/streams/AFR_OTA-db5e9ba6-d771-4ddb-8d82-bf9feb3530d/get/cbor to broker.
19:35:30.587 430 169986 [OTA Agent Task] [INFO] Published to MQTT topic to request the next block: topic=$aws/things/ameba-ota/streams/AFR_OTA-db5e9ba6-d771-4ddb-8d82-bf9feb3530d/get/cbor
19:35:30.589 431 169987 [OTA Agent Task] [DEBUG] Executing handler for state transition:
19:35:30.590 432 169987 [OTA Agent Task] [INFO] Current State=[WaitingForFileBlock], Event=[RequestFileBlock], New state=[WaitingForFileBlock]
19:35:30.729 433 170047 [MQTT Agent Task] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.

```

Line Number: 1100/1100

Serial Unconnected

COM REG AUTO

Port Setting

COM : COM156: Prolific PL2303R

Baudrate : 1500000

Data Bits : 8

Parity : None

Stop Bits : 1

Encoding : ASCII

COM156 is Open

ATW0=Yuhan-15p

ATW1=12342234

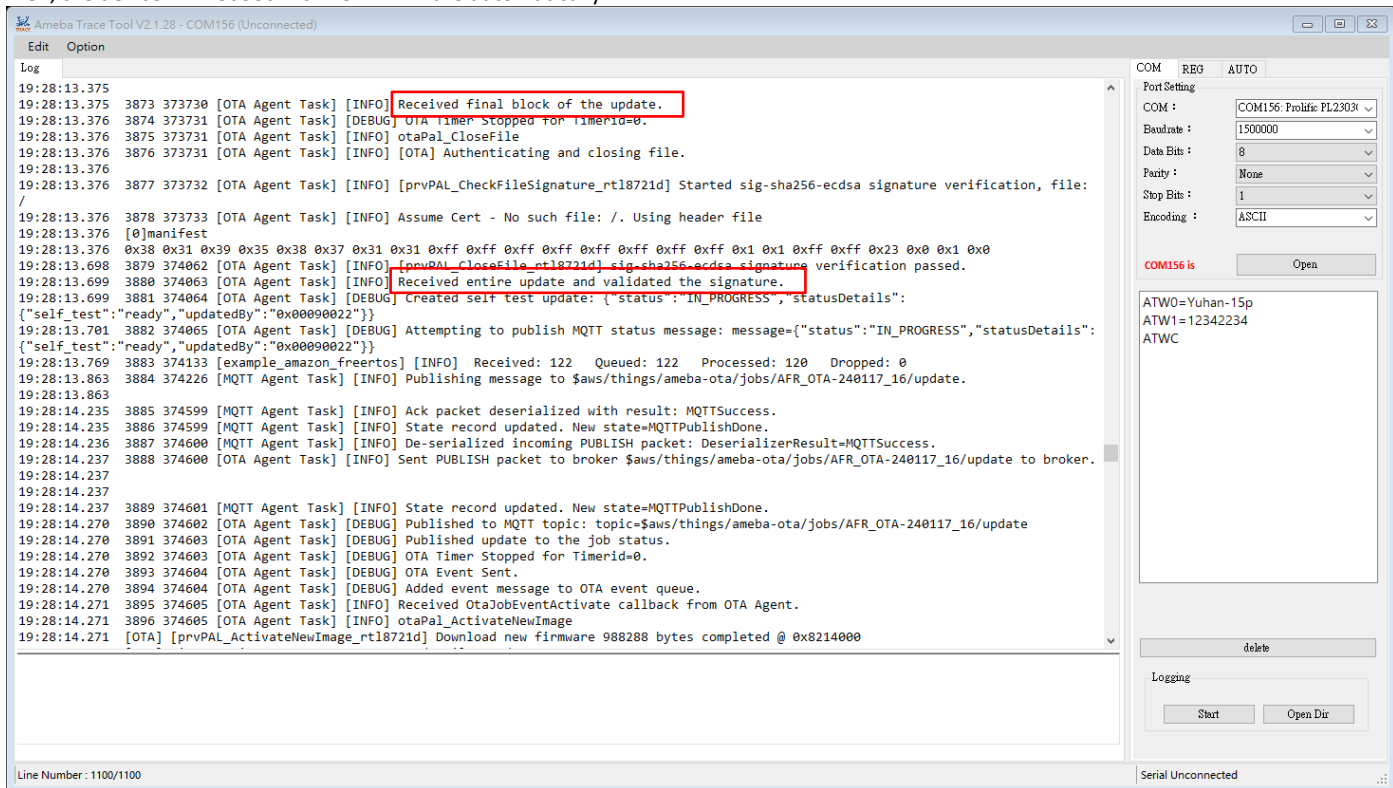
ATWC

delete

Logging

Start Open Dir

After receiving the final block, the signature will be checked if valid or not. If signature is valid, the OTA process is successful! Then, the device will reboot with new firmware automatically.



Log

```

19:28:13.375 3873 373730 [OTA Agent Task] [INFO] Received final block of the update.
19:28:13.376 3874 373731 [OTA Agent Task] [DEBUG] OTA timer Stopped for TimerId=0.
19:28:13.376 3875 373731 [OTA Agent Task] [INFO] otaPal_CloseFile
19:28:13.376 3876 373731 [OTA Agent Task] [INFO] [OTA] Authenticating and closing file.
19:28:13.376 3877 373732 [OTA Agent Task] [INFO] [prvPAL_CheckFileSignature_rt18721d] Started sig-sha256-ecdsa signature verification, file: /
19:28:13.376 3878 373733 [OTA Agent Task] [INFO] Assume Cert - No such file: /. Using header file
19:28:13.376 3879 373733 [OTA Agent Task] [INFO] [prvPAL_CheckFileSignature_rt18721d] sig-sha256-ecdsa signature verification passed.
19:28:13.698 3880 374063 [OTA Agent Task] [INFO] Received entire update and validated the signature.
19:28:13.699 3881 374064 [OTA Agent Task] [DEBUG] Created self test update: {"status":"IN_PROGRESS","statusDetails":{"self_test":"ready","updatedBy":"0x00000022"}}
19:28:13.701 3882 374065 [OTA Agent Task] [DEBUG] Attempting to publish MQTT status message: message={"status":"IN_PROGRESS","statusDetails":{"self_test":"ready","updatedBy":"0x00000022"}}
19:28:13.769 3883 374133 [example_amazon_freertos] [INFO] Received: 122 Queued: 122 Processed: 120 Dropped: 0
19:28:13.863 3884 374226 [MQTT Agent Task] [INFO] Publishing message to $aws/things/ameba-ota/jobs/AFR_OTA-240117/update.
19:28:14.235 3885 374599 [MQTT Agent Task] [INFO] Ack packet deserialized with result: MQTTSuccess.
19:28:14.235 3886 374599 [MQTT Agent Task] [INFO] State record updated. New state=MQTTPublishDone.
19:28:14.236 3887 374600 [MQTT Agent Task] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
19:28:14.237 3888 374600 [OTA Agent Task] [INFO] Sent PUBLISH packet to broker $aws/things/ameba-ota/jobs/AFR_OTA-240117/update to broker.
19:28:14.237 3889 374601 [MQTT Agent Task] [INFO] State record updated. New state=MQTTPublishDone.
19:28:14.270 3890 374602 [OTA Agent Task] [DEBUG] Published to MQTT topic: topic=$aws/things/ameba-ota/jobs/AFR_OTA-240117/update
19:28:14.270 3891 374603 [OTA Agent Task] [DEBUG] Published update to the job status.
19:28:14.270 3892 374603 [OTA Agent Task] [DEBUG] OTA Timer Stopped for TimerId=0.
19:28:14.270 3893 374604 [OTA Agent Task] [DEBUG] OTA Event Sent.
19:28:14.270 3894 374604 [OTA Agent Task] [DEBUG] Added event message to OTA event queue.
19:28:14.271 3895 374605 [OTA Agent Task] [INFO] Received OtaJobEventActivate callback from OTA Agent.
19:28:14.271 3896 374605 [OTA Agent Task] [INFO] otaPal_ActivateNewImage
19:28:14.271 [OTA] [prvPAL_ActivateNewImage_rt18721d] Download new firmware 988288 bytes completed @ 0x8214000

```

Line Number: 1100/1100

Serial Unconnected

COM REG AUTO

Port Setting

COM : COM156: Prolific PL2303R

Baudrate : 1500000

Data Bits : 8

Parity : None

Stop Bits : 1

Encoding : ASCII

COM156 is Open

ATW0=Yuhan-15p

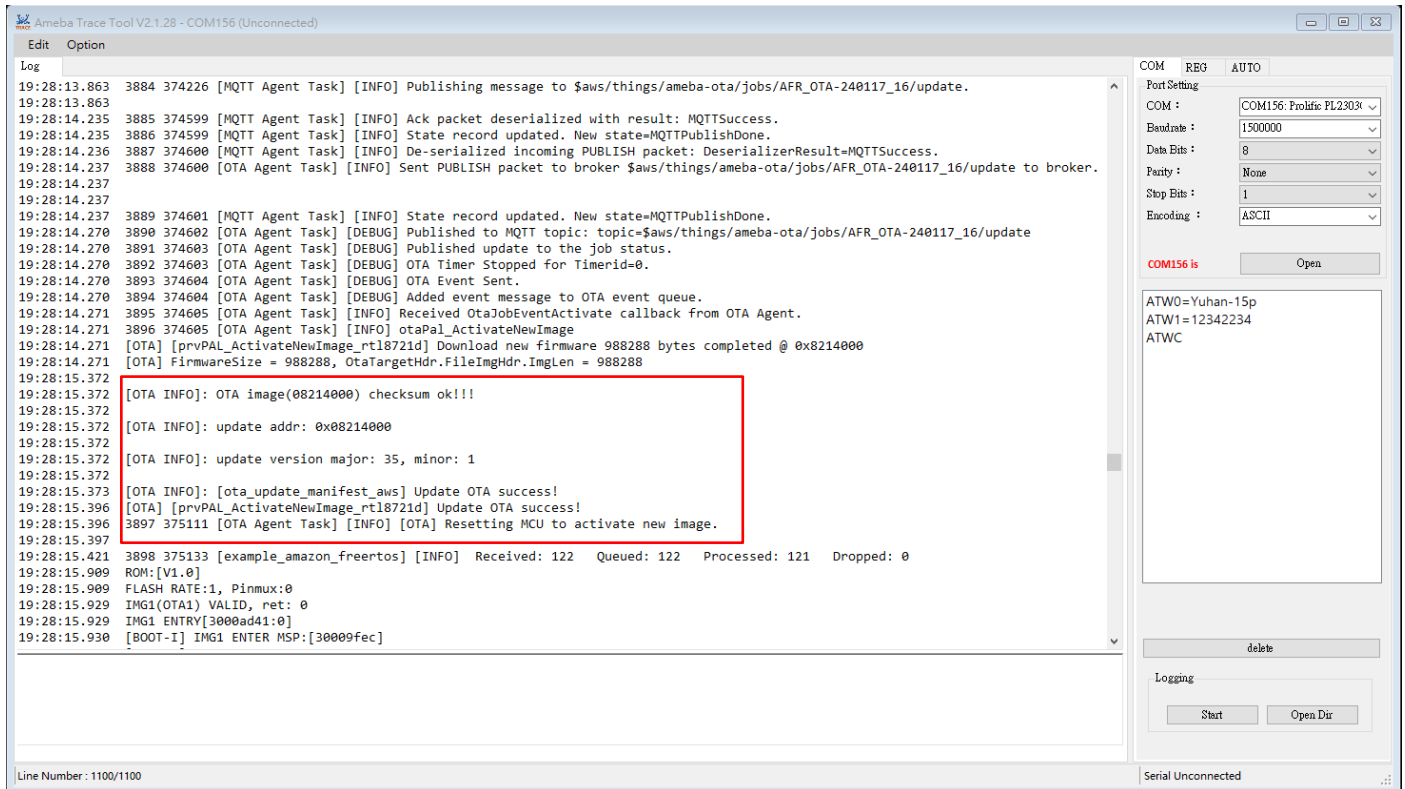
ATW1=12342234

ATWC

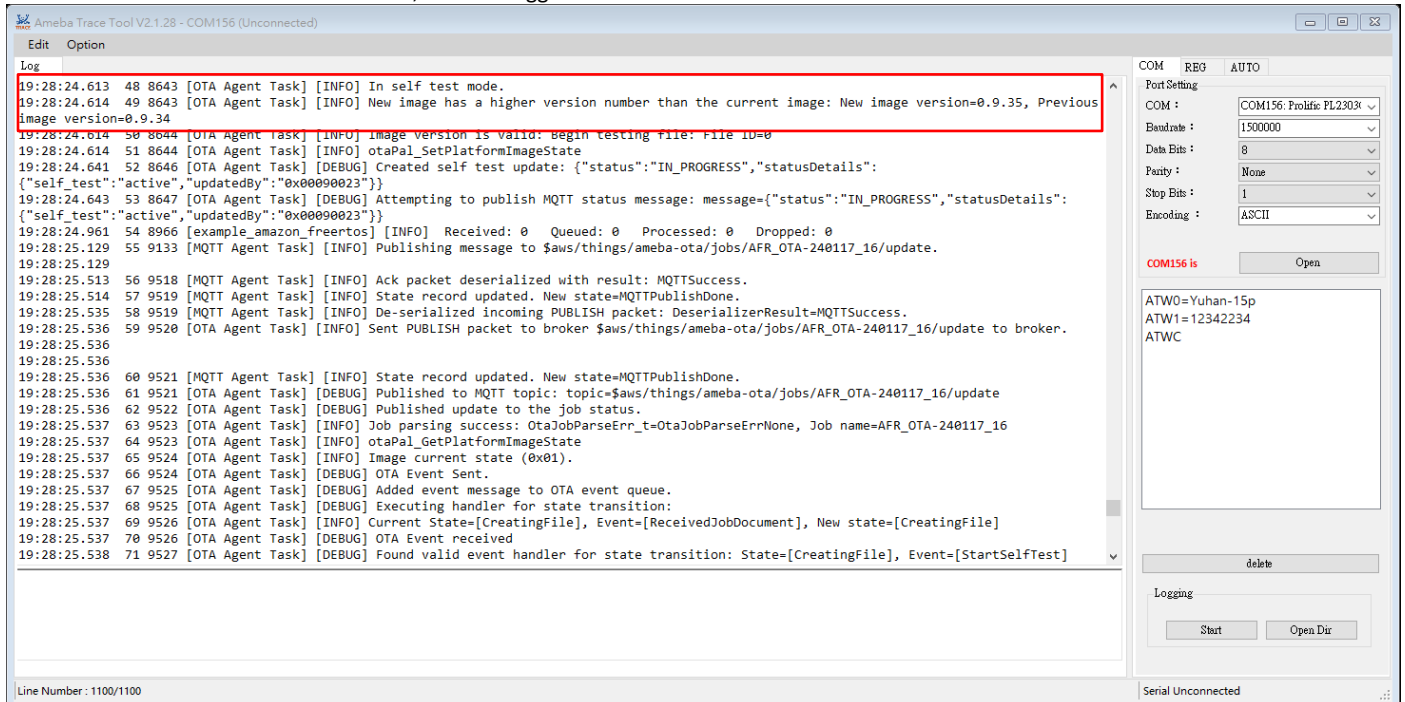
delete

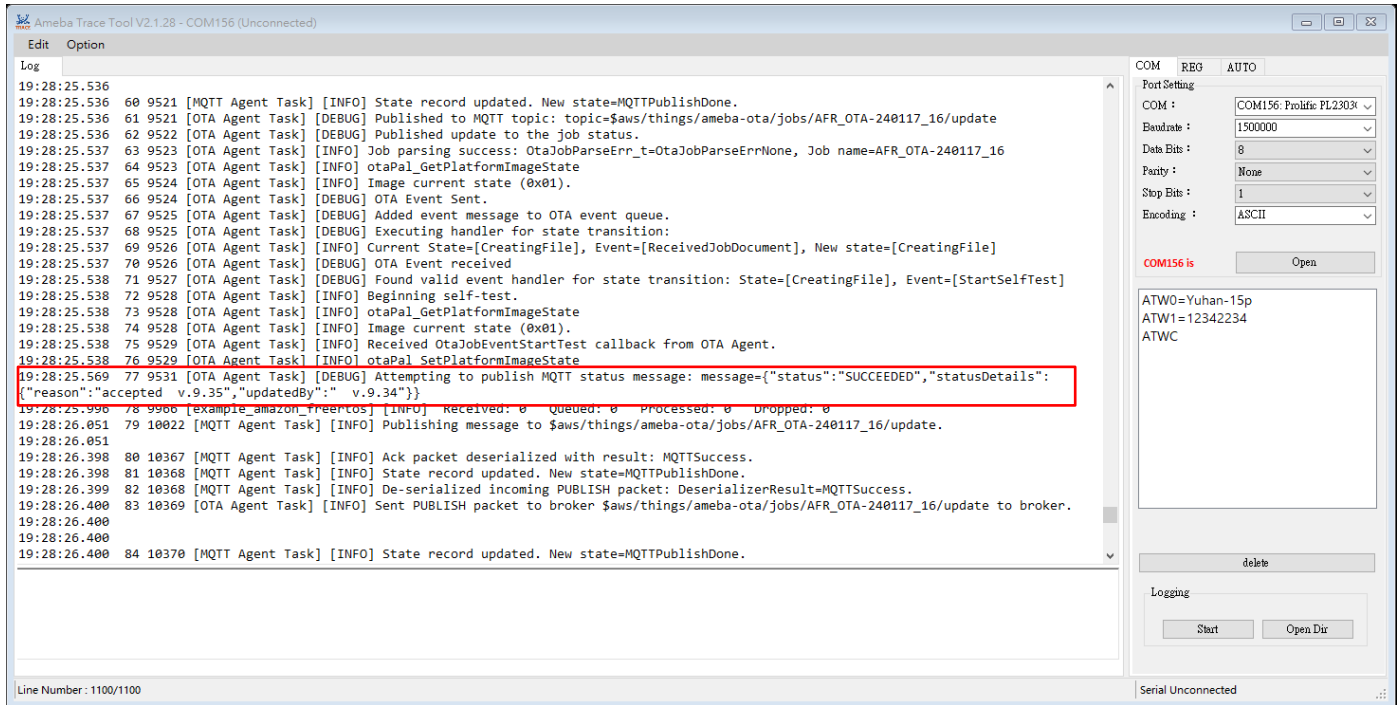
Logging

Start Open Dir



After booting with newer image, the device will start a self-test mode to check the app version is newer than before. We can see that the version now is 0.9.35, which is bigger than old one 0.9.34.





Ameba Trace Tool V2.1.28 - COM156 (Unconnected)

Log

```

19:28:25.536
19:28:25.536 60 9521 [MQTT Agent Task] [INFO] State record updated. New state=MQTTPublishDone.
19:28:25.536 61 9521 [OTA Agent Task] [DEBUG] Published to MQTT topic: topic=$aws/things/ameba-ota/jobs/AFR_OTA-240117_16/update
19:28:25.536 62 9522 [OTA Agent Task] [DEBUG] Published update to the job status.
19:28:25.537 63 9523 [OTA Agent Task] [INFO] Job parsing success: OtaJobParseErr_t=OtaJobParseErrNone, Job name=AFR_OTA-240117_16
19:28:25.537 64 9523 [OTA Agent Task] [INFO] otaPal_GetPlatformImageState
19:28:25.537 65 9524 [OTA Agent Task] [INFO] Image current state (0x01).
19:28:25.537 66 9524 [OTA Agent Task] [DEBUG] OTA Event Sent.
19:28:25.537 67 9525 [OTA Agent Task] [DEBUG] Added event message to OTA event queue.
19:28:25.537 68 9525 [OTA Agent Task] [DEBUG] Executing handler for state transition:
19:28:25.537 69 9526 [OTA Agent Task] [INFO] Current State=[CreatingFile], Event=[ReceivedJobDocument], New state=[CreatingFile]
19:28:25.537 70 9526 [OTA Agent Task] [DEBUG] OTA Event received
19:28:25.538 71 9527 [OTA Agent Task] [DEBUG] Found valid event handler for state transition: State=[CreatingFile], Event=[StartSelfTest]
19:28:25.538 72 9528 [OTA Agent Task] [INFO] Beginning self-test.
19:28:25.538 73 9528 [OTA Agent Task] [INFO] otaPal_GetPlatformImageState
19:28:25.538 74 9528 [OTA Agent Task] [INFO] Image current state (0x01).
19:28:25.538 75 9529 [OTA Agent Task] [INFO] Received OtaJobEventStartTest callback from OTA Agent.
19:28:25.538 76 9529 [OTA Agent Task] [INFO] otaPal_SetPlatformImageState
19:28:25.569 77 9531 [OTA Agent Task] [DEBUG] Attempting to publish MQTT status message: message={\"status\": \"SUCCEEDED\", \"statusDetails\": {\"reason\": \"accepted v.9.35\", \"updatedBy\": \"v.9.34\"}}
19:28:25.996 78 9966 [example_amazon_freertos] [INFO] Received: 0 Queued: 0 Processed: 0 Dropped: 0
19:28:26.051 79 10022 [MQTT Agent Task] [INFO] Publishing message to $aws/things/ameba-ota/jobs/AFR_OTA-240117_16/update.
19:28:26.051
19:28:26.398 80 10367 [MQTT Agent Task] [INFO] Ack packet deserialized with result: MQTTSuccess.
19:28:26.398 81 10368 [MQTT Agent Task] [INFO] State record updated. New state=MQTTPublishDone.
19:28:26.399 82 10368 [MQTT Agent Task] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
19:28:26.400 83 10369 [OTA Agent Task] [INFO] Sent PUBLISH packet to broker $aws/things/ameba-ota/jobs/AFR_OTA-240117_16/update to broker.
19:28:26.400
19:28:26.400
19:28:26.400 84 10370 [MQTT Agent Task] [INFO] State record updated. New state=MQTTPublishDone.

```

COM REG AUTO

Port Setting

COM : COM156: Prolific PL2303x

Baudrate : 1500000

Data Bits : 8

Parity : None

Stop Bits : 1

Encoding : ASCII

COM156 is Open

ATW0=Yuhan-15p
ATW1=12342234
ATWC

delete

Logging

Start Open Dir

Line Number : 1100/1100

Serial Unconnected

In the final, the log imply that the OTA status is changed to “SUCCEEDED” !

5 Troubleshooting

5.1 ERROR: Invalid Key

Please check **WIFI_SSID** and **WIFI_PASSWORD** in in component/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 ...
```

5.2 Failed to establish new MQTT connection

Please check **clientcredentialMQTT_BROKER_ENDPOINT** in component/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-----
```

5.3 TLS_Connect fail

Please check **keyCLIENT_CERTIFICATE_PEM** and **keyCLIENT_PRIVATE_KEY_PEM** in component/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][NET][20102] 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-----
```

6 Revision

1.0	Initial version
1.1	Update picture
1.1.1	Fix typo
2.1	Update for FreeRTOS-LTS-202210.xx