



QUICK START GUIDE

# VIA VAB-3000

AWS IoT Core



## **Copyright**

Copyright © 2023 VIA Technologies Incorporated. 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, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of VIA Technologies, Incorporated.

## **Trademarks**

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

## **Disclaimer**

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information (including use or connection of extra device/equipment/add-on card) in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right to make changes to the products described in this manual at any time without prior notice.



## Revision History

Version	Date	Remarks
1.00	05/12/2023	Initial release

## Table of Contents

1.	Introduction.....	1
1.1	VIA VAB-3000 Board .....	1
1.2	Directing Data from a VIA VAB-3000 device to AWS IoT Core.....	2
2.	Connecting to AWS IoT Core.....	3
2.1	Introduction .....	3
2.2	Set up your AWS Account and Permissions .....	4
2.3	Create Resources in AWS IoT .....	4
2.4	Image Installation.....	11
2.5	Connecting the VIA VAB-3000 device to AWS IoT Core .....	12
2.5.1	Connecting the Debug Console.....	12
2.5.2	Connecting with the Android 12.0 EVK.....	13
2.5.3	Connecting with the Yocto 3.1 EVK.....	16
2.6	Debugging .....	18
2.7	Troubleshooting.....	18

# 1. Introduction

This document provides instructions on how to set up AWS IoT Core to connect with a VIA VAB-3000 device.

## 1.1 VIA VAB-3000 Board

Powered by the high performance and energy-efficient quad-core MediaTek Genio 350 SoC, the VIA VAB-3000 Edge AI board delivers blistering edge processing and multimedia performance, while combining advanced AI technology for display, object recognition, and voice, with rich wireless and I/O connectivity features. Besides quad-core ARM Cortex-A53 processors, the MediaTek Genio 350 platform also features an integrated AI processor for deep learning, neural network acceleration, and computer vision applications, including facial recognition, object identification, and OCR.

Based on the 3.5" SBC 14.6cm x 10.2cm form factor, the VIA VAB-3000 board offers a wealth of network connectivity options, including an integrated SIM card slot for 4G LTE, dual-band 802.11ac Wi-Fi, a 10/100Mbps Ethernet port, and Bluetooth 5.0. The board also features a USB 2.0 port, a Micro USB 2.0 OTG port, as well as HDMI, MIPI DSI display, MIPI CSI camera, and touch panel support with multi-function pins for I<sup>2</sup>C, SPI and GPIO connectivity. On-board storage is provided by 16GB eMMC 5.1 flash memory.

The VIA VAB-3000 board's multimedia capabilities include high-speed graphics processing, hardware-accelerated H.265/H.264 Full HD video decoding, and support for MIPI/HDMI/touch panel displays and cameras - making it the ideal solution for an unlimited array of home, commercial, industrial, and educational applications.



Refer to the datasheet and user manual on the [VIA VAB-3000](#) product page for detailed information.

## 1.2 Directing Data from a VIA VAB-3000 device to AWS IoT Core

To direct data from VIA VAB-3000 devices to a cloud implementation, the AWS IoT Core Web Service must be set up and configured to receive data coming from VIA VAB-3000 devices.

Follow the steps below to set up your AWS services in order to connect a VIA VAB-3000 device to an AWS backend:

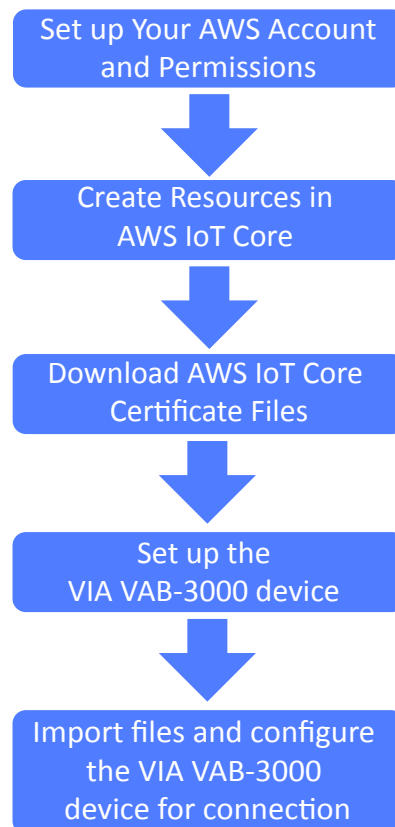
- Follow the steps listed in [Section 2.3](#) to register an AWS IoT Thing and to acquire the AWS IoT certificate files for the VIA VAB-3000 devices.
  - Acquire the certificate (\*-certificate.pem) and the private key (\*-private.pem.key) files associated with the IoT thing created, as described in [Step 7 of Section 2.3](#).
  - Acquire the Device Data Endpoint described in [Step 9 of Section 2.3](#).

## 2. Connecting to AWS IoT Core

### 2.1 Introduction

This section provides instructions on how to establish a connection between VIA VAB-3000 devices and AWS IoT Core, including how to add and manage VIA VAB-3000 devices as an IoT Core device on AWS, set up the device, and configure the VIA IoT application.

To register an AWS IoT thing and acquire the AWS IoT certificate files for the VIA VAB-3000 device, follow the steps below:



## 2.2 Set up your AWS Account and Permissions

Refer to instructions in the following sections on webpage <https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html> to set up an AWS Account:

- [Sign up for an AWS account](#)
- [Create a user and grant permissions](#)
- [Open the AWS IoT console](#)



### Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

☐ Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)



Pay special attention to the Notes on the AWS account setup webpage.

## 2.3 Create Resources in AWS IoT

Refer to instructions in the following sections on webpage <https://docs.aws.amazon.com/iot/latest/developerguide/create-iot-resources.html> to provision resources for your VIA VAB-3000 device:

- [Create an AWS IoT policy](#)
- [Create a thing object](#)

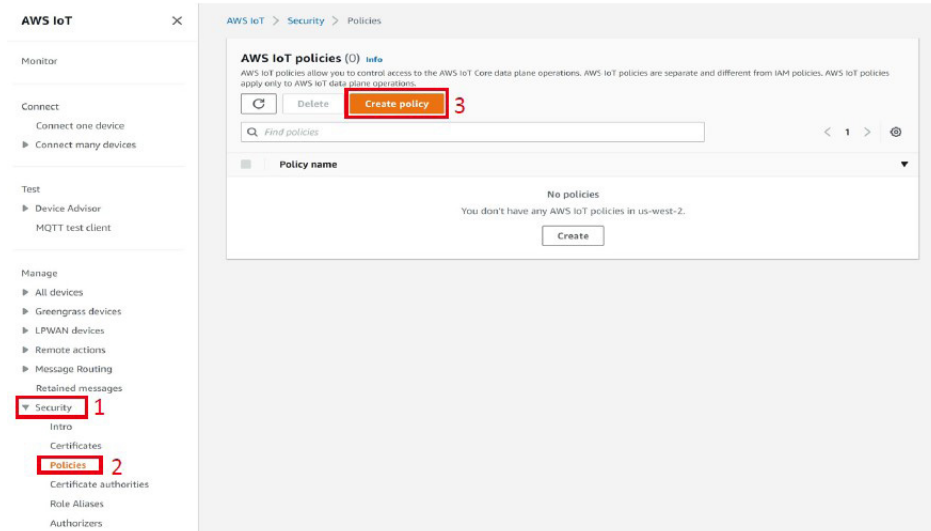
Pay special attention to the Notes on the AWS IoT resource creation webpage.



The following instructions demonstrate how to create an AWS IoT Policy and a thing object, and acquiring the required AWS IoT certificate files and Device Data Endpoint for the VIA VAB-3000 device.

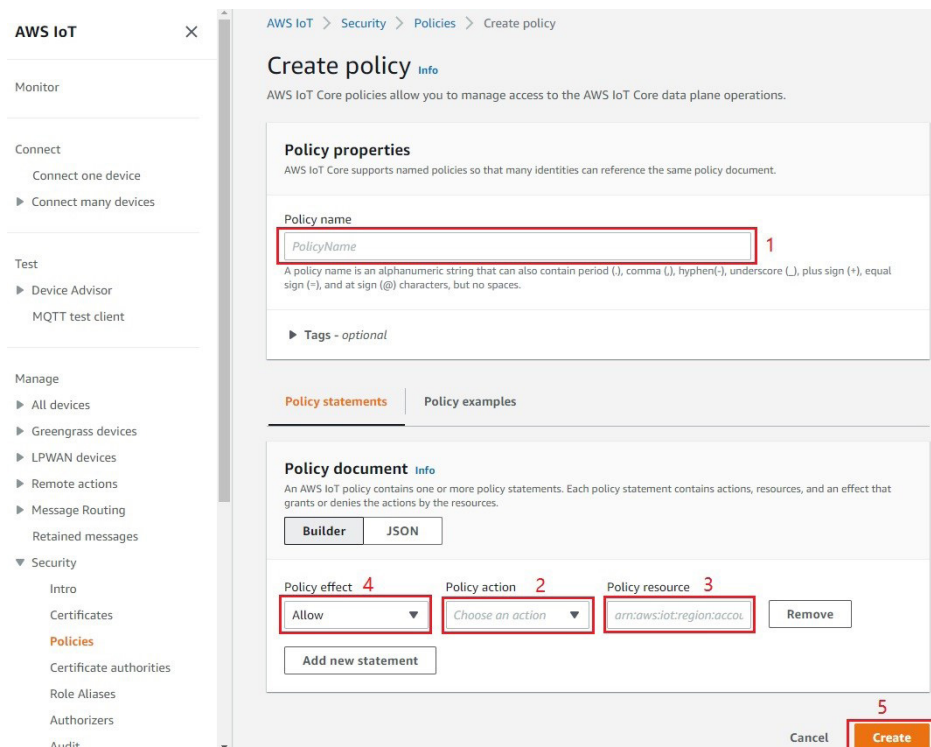
## Step 1

Click on 'Security/Policies' in the left panel and then click on the 'Create policy' button in the right panel.

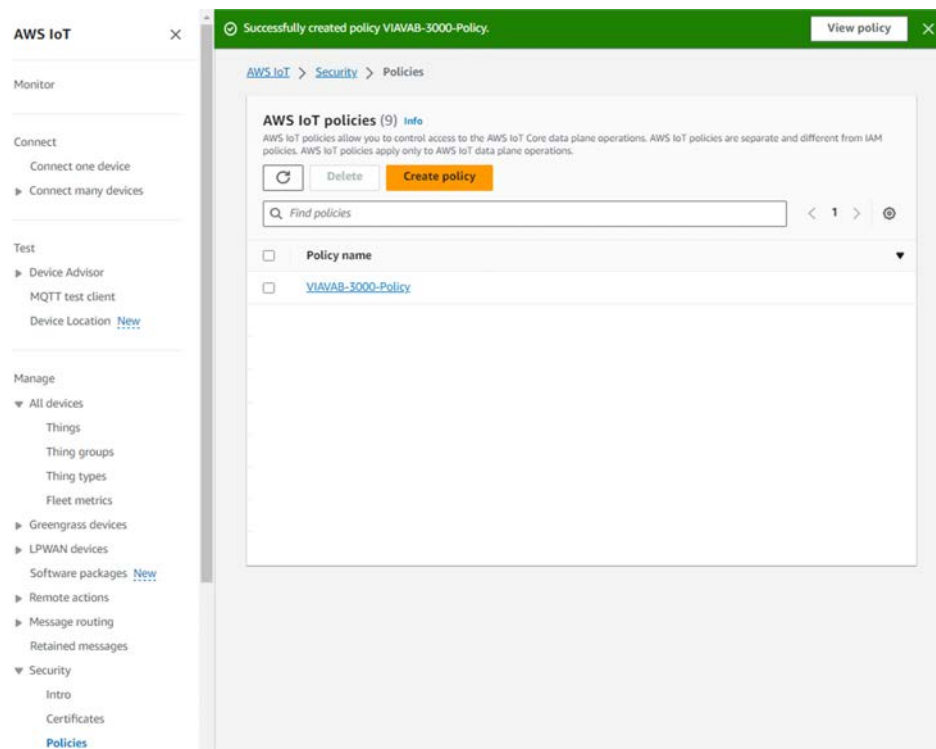


## Step 2

In the "Create policy" form, fill in the "Policy name" in the "Policy properties" section. In the "Policy document" section, select the "Policy effect" and "Policy action", and fill in the "Policy resource" name. Then, click the 'Create' button to submit.

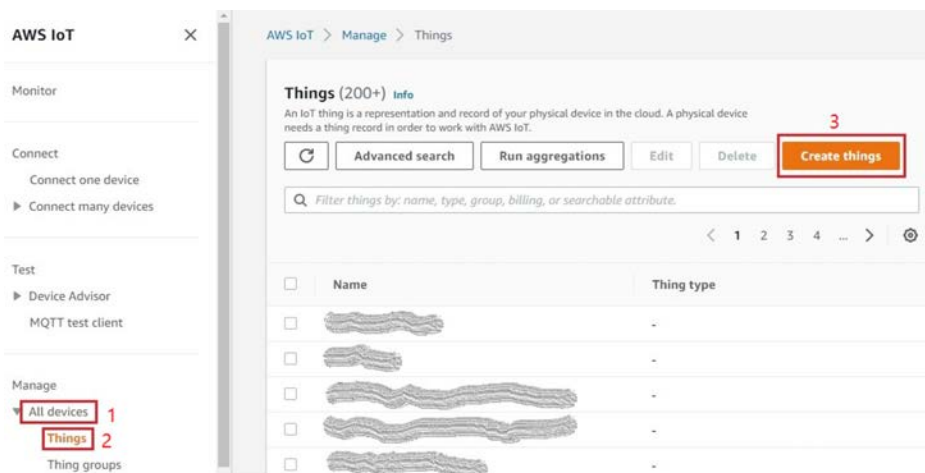


After clicking the 'Create' button, the new policy's name will be listed on the right panel.



### Step 3

To create a new 'IoT Thing' resource with the new policy and to create certificate files for this resource, click on 'All devices/Things' in the left panel and then click on the 'Create things' button in the right panel.



Next, select 'Create single thing' and click on "Next".

AWS IoT > Manage > Things > Create things

## Create things Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

### Number of things to create

☒ **Create single thing**  
 Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.

☐ **Create many things**  
 Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

Cancel **Next**

### Step 4

Under 'Thing properties' in the right panel, enter the 'Thing Name'. Under 'Device Shadow', select 'Unnamed shadow (classic)' and click on the 'Next' button to continue.

AWS IoT > Manage > Things > Create things > Create single thing

Step 1  
**Specify thing properties**

Step 2 - optional  
Configure device certificate

Step 3 - optional  
Attach policies to certificate

## Specify thing properties Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

### Thing properties Info

Thing name

VIAVAB-3000

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

### Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

- ▶ Thing type - optional
- ▶ Searchable thing attributes - optional
- ▶ Thing groups - optional
- ▶ Billing group - optional
- ▶ Packages and versions - optional

### Device Shadow Info

Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

- ☐ No shadow
- ☐ Named shadow  
Create multiple shadows with different names to manage access to properties, and logically group your devices' properties.
- ☒ **Unnamed shadow (classic)**  
A thing can have only one unnamed shadow.
- ▶ Edit shadow statement - optional

Cancel **Next**

## Step 5

Under 'Device certificate', select 'Auto-generate a new certificate (recommended)' and click on the 'Next' button to configure a device certificate for your VIA VAB-3000 device.

## Step 6

Click on the new policy's name and click on the 'Create Thing' button to attach policies to the device certificate for your VIA VAB-3000 device and to complete creating a new 'IoT Thing'.

**Step 7**

Download the device certificate (CertFile), the private key file (PvkFile) and the 2048-bit Amazon Root CA 1 certificate (CAFile) to a local folder and click on 'Done' to complete the certificate creation process for your VIA VAB-3000 device.

Download certificates and keys

×

Download certificate and key files to install on your device so that it can connect to AWS.

### Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate  
ec66a2571e3...te.pem.crt

Deactivate certificate

1  
Download

### Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

2  
Public key file  
ec66a2571e3f0e8afc98b35...ef6d8a1-public.pem.key  
Download

3  
Private key file  
ec66a2571e3f0e8afc98b35...f6d8a1-private.pem.key  
Download

### Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint  
RSA 2048 bit key: Amazon Root CA 1  
4  
Download

Amazon trust services endpoint  
ECC 256 bit key: Amazon Root CA 3  
Download

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

5  
Done

## Step 8

Click on the created IoT thing's name under 'Things' to view your VIA VAB-3000 device's Thing and Device Shadow details, including the MQTT topic prefix.



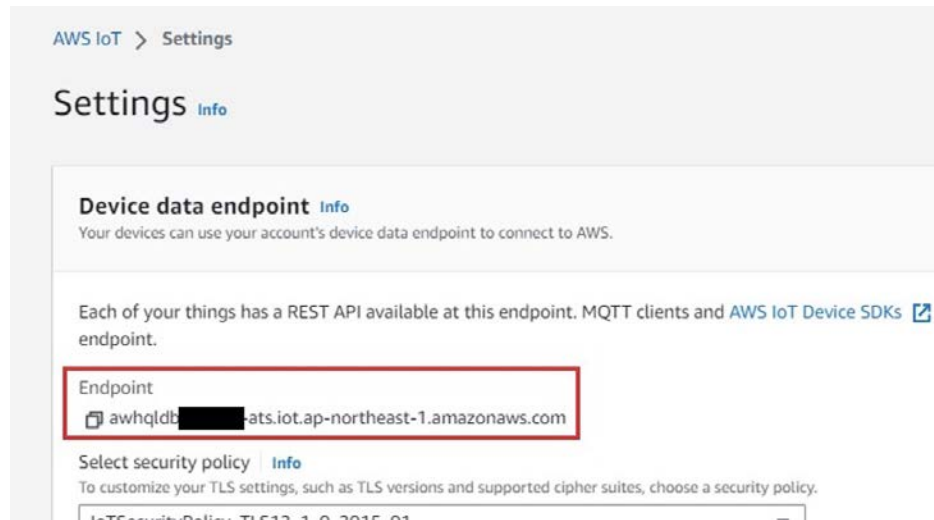
### Note:

The MQTT topic prefix is for binding your VIA VAB-3000 device with AWS IoT Core Web services. The VIA VAB-3000 device will use the specified MQTT topic to interact with AWS IoT Core Web services. Some [MQTT topics listed on the AWS webpage](#) are reserved.

The screenshot displays the AWS IoT console interface. On the left, the navigation menu shows 'Monitor', 'Connect', 'Test', and 'Manage' sections. The 'Manage' section is expanded, showing 'All devices' and 'Things'. The main content area shows a confirmation message: 'You successfully created thing VIAVAB-3000.' and 'You successfully created certificate ec66a2571e3f0e8afc98b352b1579191fee48089fba74d7bba15b7c1ef6d8a1.' Below this, the 'Things (12) info' section is visible, with a search bar and a table listing the created thing, 'VIAVAB-3000'. The bottom section shows the 'VIAVAB-3000' details page, including 'Thing details' (Name: VIAVAB-3000, ARN: araws:iot:ap-northeast-1:620203865716:thing/VIAVAB-3000) and 'Device Shadows (1) info'. The 'Device Shadows' section shows a table with one entry: 'Classic Shadow' with the MQTT topic prefix '\$aws/things/VIAVAB-3000/shadow'.

## Step 9

In the left "AWS-IoT" panel, select "Settings". The "Endpoint" is displayed in the "Device data endpoint" section.



## 2.4 Image Installation

The VIA VAB-3000 BSP supports Android 12.0 and Yocto 3.1 operating systems. To enable hardware functionality, download either the VIA VAB-3000 Android 12.0 or Yocto 3.1 EVK package, and refer to the corresponding EVK Quick Start Guide included in the package to flash the device.



### VIA VAB-3000 BSP

The VIA VAB-3000 BSP supports Yocto 3.1 and Android 12.0. VIA VAB-3000 Yocto 3.1 and Android 12.0 EVK packages to enable the hardware functionality can be downloaded here. A full set of software customization services that speed up time to market and minimize development costs is also available.

↓ Android 12.0 EVK	[972.8 MB]	v2.0.0	2023-07-06
↓ Yocto 3.1 EVK	[236.4 MB]	v3.1.0	2023-06-09

**Important:** By downloading these software packages, you indicate your acceptance of the [VIA Software License agreement](#). If you do not agree with any of the terms and conditions, do not continue to download the software.

## 2.5 Connecting the VIA VAB-3000 device to AWS IoT Core

### 2.5.1 Connecting the Debug Console

The first step is to connect the VIA VAB-3000 device to the debug console for setting up a connection between the VIA VAB-3000 device and AWS IoT Core. The optional VIA VAB-3000 development kit and a serial port communication program such as PuTTY or Tera Term are required for the setup.

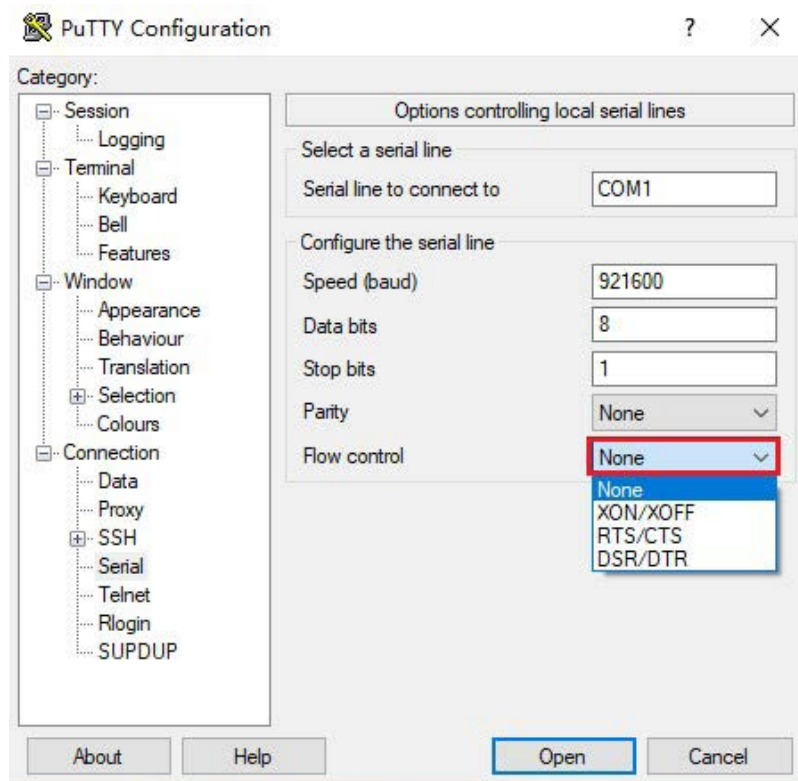
Follow the steps below to connect the VIA VAB-3000 device to the debug console:

#### Step 1

Refer to Appendix section A.2 of the VIA VAB-3000 User Manual for instructions on setting the power jumpers and installing the development kit.

#### Step 2

Use PuTTY or Tera Term to connect and set the Baud Rate to "921600".



#### Step 3

Power on the VIA VAB-3000 device to initiate the boot process.

#### Step 4

When the boot process has completed, log in to the debug console. The default account "username: root / password: root" can be used for login if the Yocto 3.1 EVK will be used to connect to AWS IoT Core.

#### Step 5

Refer to the section 2.5.2 (if using the Android 12.0 EVK) or 2.5.3 (if using the Yocto 3.1 EVK) to connect the VIA VAB-3000 device to your AWS IoT Core Web service.



## 2.5.2 Connecting with the Android 12.0 EVK

The VIA VAB-3000 Android 12.0 EVK includes an "IoTDeviceSDKSample" application which establishes a connection between the VIA VAB-3000 device and the AWS IoT Core Web service.

This section guides developers on how to enable and run the "IoTDeviceSDKSample" application.

### Step 1

Edit the "iotcoreConfig.json" settings file and modify the following values based on the AWS IoT Core setup:

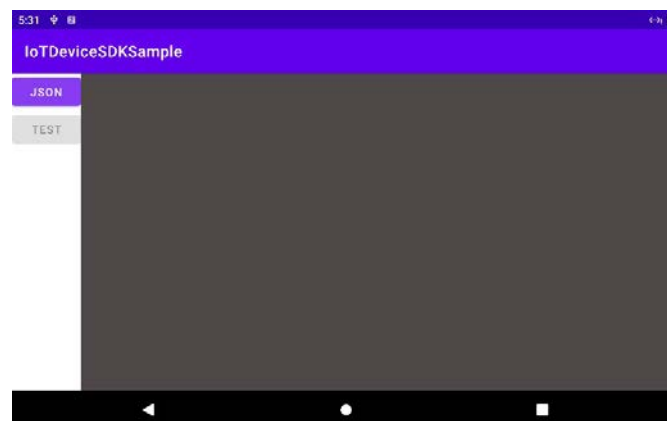
- **thingName:** Insert the defined thing's name.
- **ca:** Input the content text of the thing's Amazon Root CA certificate.
- **cert:** Input the content text of the thing's Certificate file.
- **key:** Input the content text of the thing's Private Key file.
- **ReqRetry:** Number of retries for establishing the connection between the device and the IoT Core server when the connection is disrupted or not successful.
- **topic:** The MQTT topic.
- **endpoint:** Input the domain name from the Device Data Endpoint obtained in [Step 9 of Section 2.3](#).

Save the settings file to a USB flash drive and plug it to the device's USB port.

```
{
  "thingName": "VIAVAB-3000",
  "ca": "-----BEGIN CERTIFICATE-----MIIDQTCCAimgAWIBAgITBmyfz5m/jA054
  "cert": "-----BEGIN CERTIFICATE-----MIIDWjCCAkKgAWIBAgIVANMO7BsnK8z0
  "key": "-----BEGIN RSA PRIVATE KEY-----MIIEpAIBAAKCAQEAYuWp8RiCy8f5
  "ReqRetry": "10",
  "topic": "$aws/things/VIAVAB-3000/shadow/update",
  "endpoint": "ar[REDACTED]-ats.iot.ap-northeast-1.amazonaws.com"
}
```

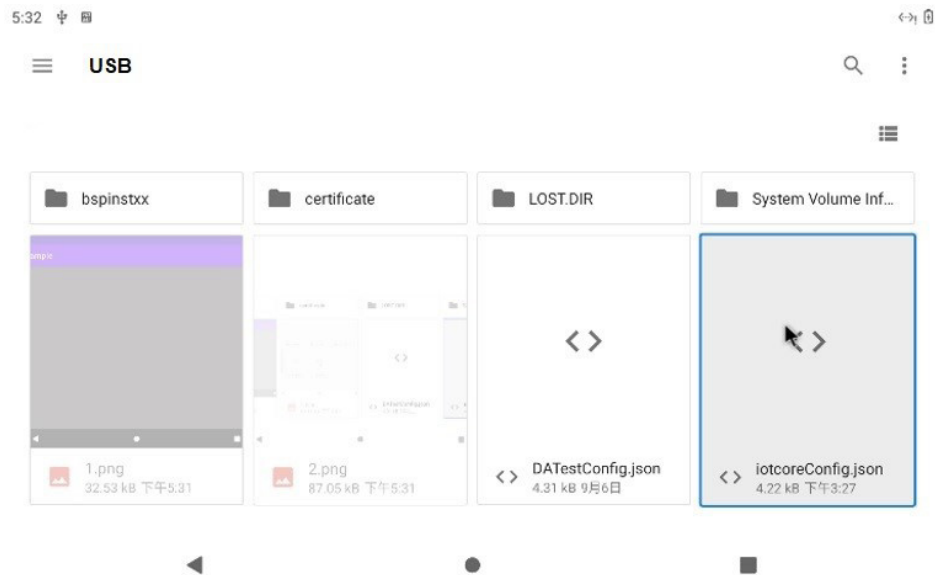
### Step 2

Run the "IoTDeviceSDKSample" application as shown below.

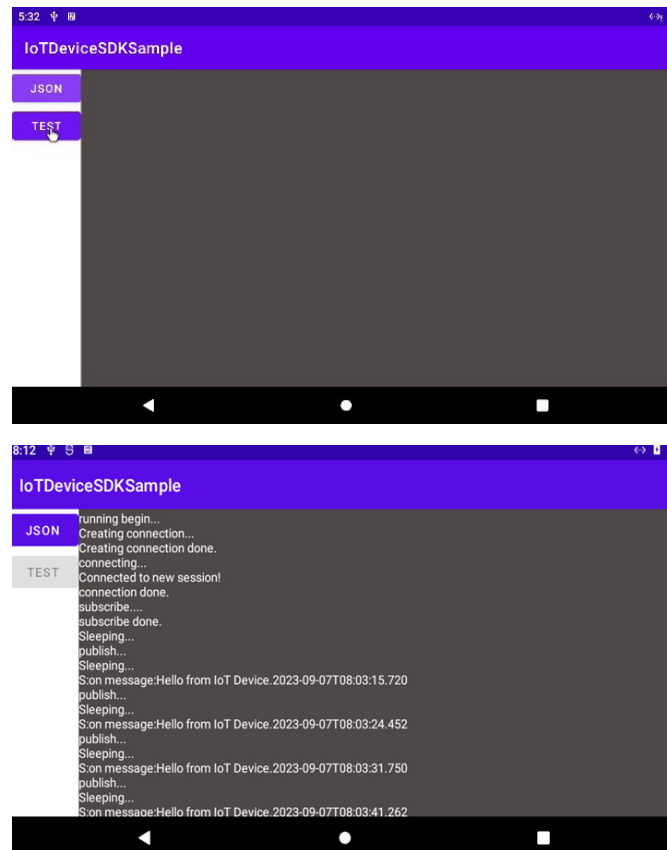


### Step 3

Click the "JSON" button to select the "iotcoreConfig.json" created in Step 1 from USB flash drive.



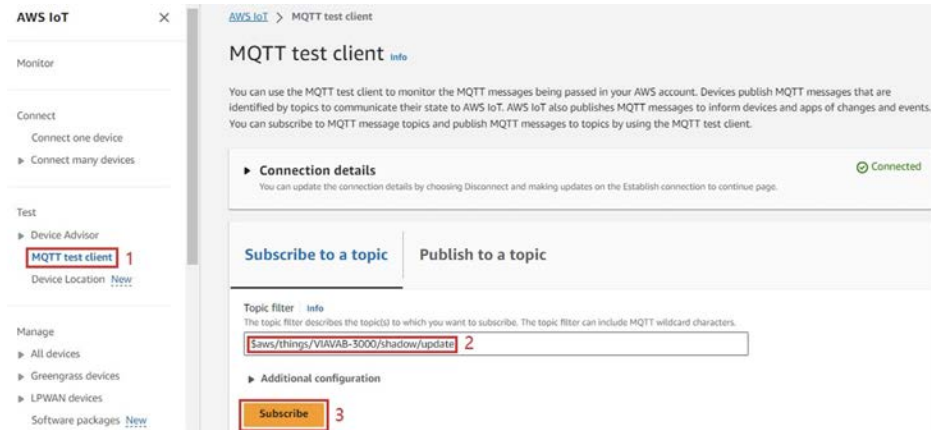
Click the "TEST" button to run the MQTT subscription publishing process as shown below.



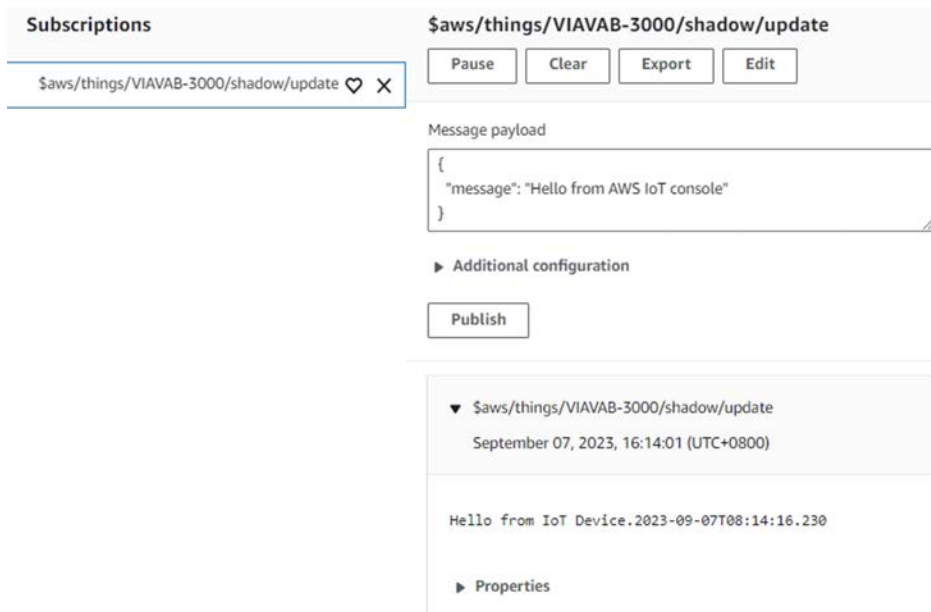
#### Step 4

Verify the "IoTSDKSample" application using the AWS IoT Core Web service console. Click on the 'MQTT test client' tab.

To subscribe to the interaction topic with the MQTT topic prefix obtained in [step 8 of Section 2.3](#) "\$aws/things/ThingName/shadow/update", enter a topic filter name and click the 'Subscribe' button in the 'Subscribe to a topic' tab in the right panel. The value of "thingName" must be the same as the value of "iotcoreConfig.json" settings file.



The last updated message will be shown in the AWS IoT MQTT test client.



## 2.5.3 Connecting with the Yocto 3.1 EVK

The VIA VAB-3000 Yocto 3.1 EVK includes an "IoTSDKSample" application which establishes a connection between the physical sensors/actuators on the VIA VAB-3000 device and the AWS IoT Core Web service.

This section guides developers on how to enable and run the "IoTSDKSample" application.

### Step 1

Copy the device certificate (CertFile), private key file (PvkFile) and the Amazon Root CA certificate (CAFile) created in [Step 7 of Section 2.3](#) to "/data/aws/certs".

Open the "/data/aws/iot.json" settings file and modify the following values based on the AWS IoT Core setup:

- **Endpoint:** Input the Device Data Endpoint obtained in [Step 9 of Section 2.3](#) as the Endpoint in line 1.
- **CAFile:** Input the name of the Amazon Root CA certificate in line 4.
- **CertFile:** Input the name of the Certificate file in line 5.
- **PvkFile:** Input the name of the Private Key file in line 6.
- **ThingName:** Insert the defined "ThingName" in line 7.
- **ClientID:** Insert the defined "ClientID" in line 8. This value can be randomly generated but must be unique.

Save the changes made to the settings file.

```
{
  "Endpoint" : "ar[REDACTED]-ats.iot.ap-northeast-1.amazonaws.com",
  "Certs" : {
    "FolderFullName" : "/data/aws/certs",
    "CAFile" : "AmazonRootCA1.pem",
    "CertFile" : "ec66a2571e3f0e8afc98b352b1579191feea48089fba74d7bba15b7clef6d8a1-certificate.pem.crt",
    "PvkFile" : "ec66a2571e3f0e8afc98b352b1579191feea48089fba74d7bba15b7clef6d8a1-private.pem.key"
  },
  "ThingName" : "VIAVAB-3000",
  "ClientID" : "python-sdk-client-id-001"
}
```

### Step 2

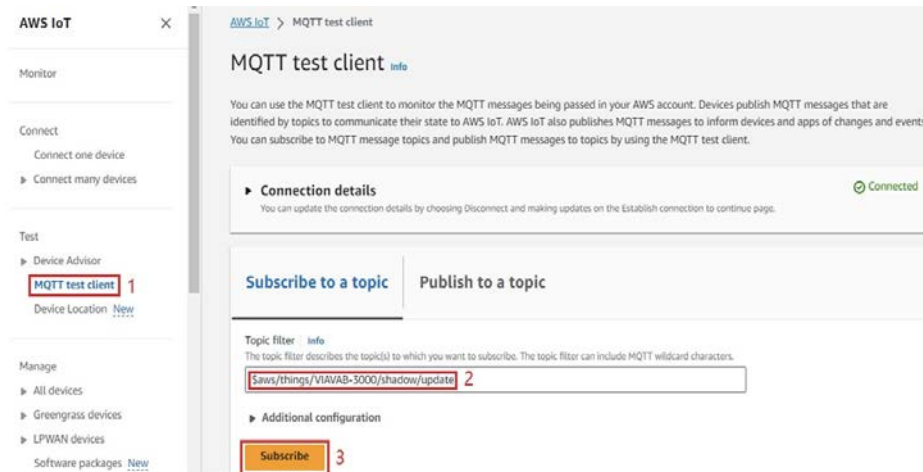
Run the "IoTSDKSample" application as shown below:

```
root@aiv8365mlv1-aiot-emmc:~# IoTSDKSample
```

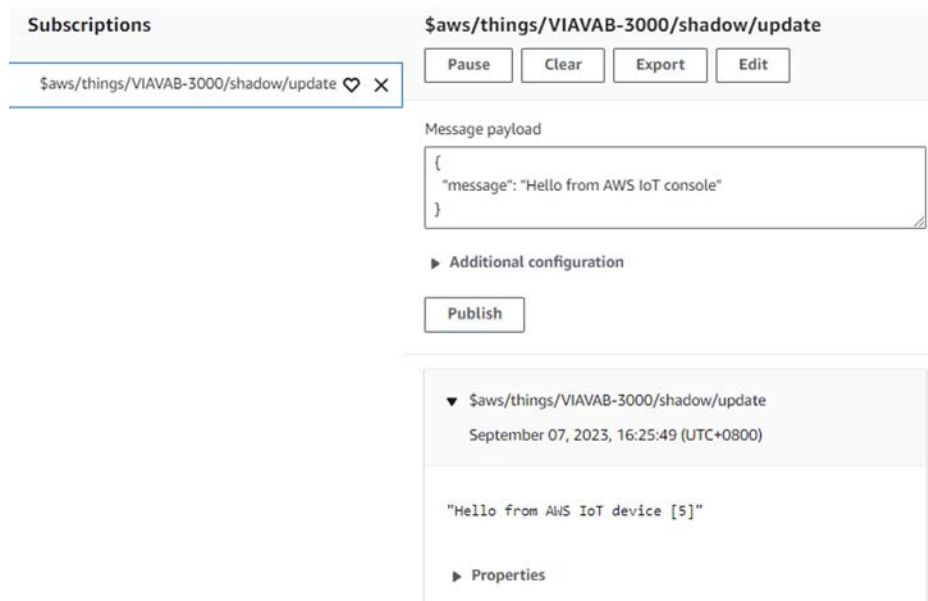
### Step 3

Verify the "IoTSDKSample" application using the AWS IoT Core Web service console. Click on the 'MQTT test client' tab.

To subscribe to the interaction topic with the MQTT topic prefix obtained in [step 8 of Section 2.3](#) "\$aws/things/ThingName/shadow/update", enter a topic filter name and click the 'Subscribe' button in the 'Subscribe to a topic' tab in the right panel. The value of "ThingName" must be the same as the value in the "iot.json" settings file described in step 1.



The last updated message will be shown in the AWS IoT MQTT test client.



Open a console (e.g. Putty) and configure as described in [Section 2.5.1](#). The boot up messages will be presented with a command line interface as well as debug output. Use busybox commands "in /bin" to set up and debug the VIA VAB-3000 device.

### For Android

```
$ ls /bin
```

## For Yocto

[illegible]

Check the table below for troubleshooting common AWS IoT Core connection issues that may arise during development:

Issue	Solution
Device does not connect to the Internet	<p>Confirm that the SIM card is inserted properly.</p> <p>Confirm that the SIM card is activated by the service provider.</p> <p>Verify the APN settings.</p>
Device does not connect to AWS	<p>Confirm that the date and time is synchronized.</p> <p>Verify the AWS endpoint in the "iot.json" (Yocto 3.1 EVK) or "iot-coreConfig.json" (Android 12.0) settings file.</p> <p>Verify that appropriate certificates and keys are loaded on the VIA VAB-3000 device.</p> <p>Verify the IoT Core thing and policies set on AWS IoT Core.</p>



#### **Taiwan Headquarters**

1F, 531 Zhong-zheng Road,  
Xindian Dist., New Taipei City 231  
Taiwan

Tel: 886-2-2218-5452  
Fax: 886-2-2218-9860  
Email: [embedded@via.com.tw](mailto:embedded@via.com.tw)



#### **USA**

940 Mission Court  
Fremont, CA 94539,  
USA

Tel: 1-510-687-4688  
Fax: 1-510-687-4654  
Email: [embedded@viatech.com](mailto:embedded@viatech.com)



#### **Japan**

3-15-7 Ebisu MT Bldg. 6F,  
Higashi, Shibuya-ku  
Tokyo 150-0011  
Japan

Tel: 81-3-5466-1637  
Fax: 81-3-5466-1638  
Email: [embedded@viatech.co.jp](mailto:embedded@viatech.co.jp)



#### **China**

Tsinghua Science Park Bldg. 7  
No. 1 Zongguancun East Road,  
Haidian Dist., Beijing, 100084  
China

Tel: 86-10-59852288  
Fax: 86-10-59852299  
Email: [embedded@viatech.com.cn](mailto:embedded@viatech.com.cn)



#### **Europe**

Email: [embedded@via-tech.eu](mailto:embedded@via-tech.eu)