

Quick Start Guide

Security Starter Kit with STM32WB55 and OPTIGA™ Trust M

Date: June 15, 2021 | Version 1.2



The Solutions People



CONTENTS

1	INTRODUCTION	3
1.1	Purpose of the Document	3
1.2	Prerequisite	3
2	INSTALLATION STEPS	4
2.1	Hardware setup	4
2.2	Software Setup	5
2.2.1	Setup Serial console on your PC	5
2.2.2	AWS Account creation and Arrow Cloud Connect tool configuration	7
2.2.3	Setup for Mobile App	7
3	DEMO SETUP	11
3.1	FreeRTOS MQTT Demo configuration	11
3.2	Demo Inference	18

DEFINITION, ACRONYMS AND ABBREVIATIONS

Definition/Acronym/Abbreviation	Description
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
OTA	Over-the-Air
SSK	Security Starter Kit
STM	STMicroelectronics
APP	Mobile Application
MQTT	An open OASIS and ISO standard (ISO/IEC 20922) lightweight, publish-subscribe network protocol that transports messages between devices

1 INTRODUCTION

1.1 Purpose of the Document

The Quick Start Guide for Security Starter Kit with STM32WB55 and OPTIGA™ Trust M will showcase FreeRTOS and MQTT running on the STM32WB55 using the OPTIGA™ TrustM (shield2go) with secure communications to the cloud via a Mobile Application (SSK app). This demo also covers the use of AWS Cognito for Mutual authentication & provisioning between AWS IoT Core and the STM32WB55.

1.2 Prerequisite

This guide presumes that the below Hardware and software are available to enable demonstration of the FreeRTOS running on the STM32WB55 and OPTIGA™ Trust M secure element:

- SSK kit
 - Infineon Shield2Go Cloud Security OPTIGA™ Trust M
 - ST Micro STM32WB55 evaluation kit (P-NUCLEO-WB55)
 - Custom cable connecting OPTIGA™ Trust M with STM32WB55 P-Nucleo kit
 - Micro USB cable (required for power and communication with the Host PC)
- HOST PC - Linux or Windows system
- Android or IOS mobile phone (with internet connectivity) – Not included



Figure 1: Architecture

2 INSTALLATION STEPS

2.1 Hardware setup

The Security Starter Kit with STM32WB55 and OPTIGA™ Trust M has been shipped pre-configured with the correct jumper settings.

If this is not a new kit right out of the box, please refer to the [STM32WB55-SSK Developers Guide.pdf](#) Section 2.1 for the proper jumper settings:

<https://www.arrow.com/en/products/stm32wb55-ssk/arrow-development-tools>

Connect the Micro USB cable to the PC & the STM32WB55 board USB connector (shown below) to supply power and serial console connectivity:

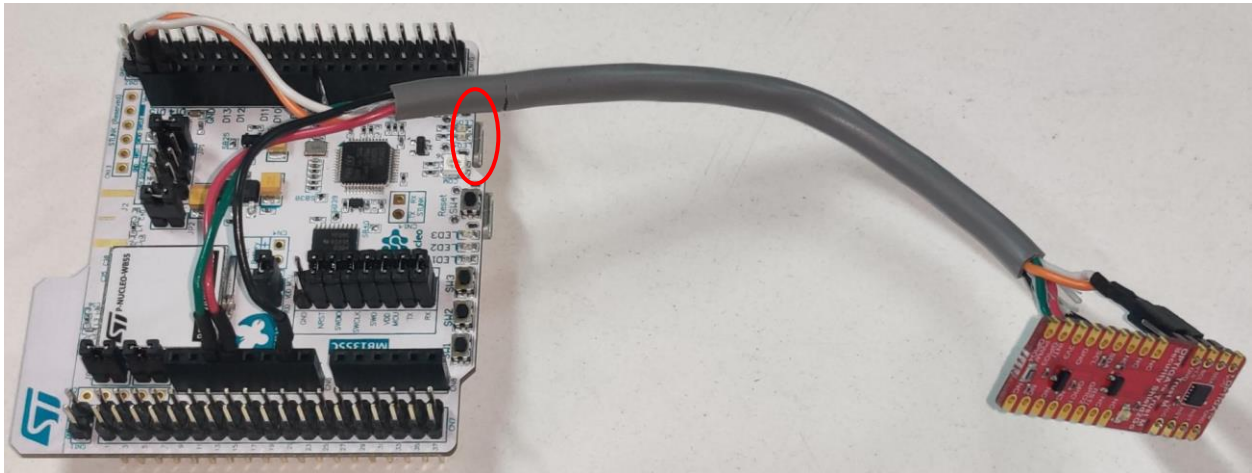


Figure 2: Hardware Setup

Connection between OPTIGA™ Trust M with STM32WB55 board with the cable as per the below table:

Connections No	OPTIGA™ TrustM	STM32WB55
1(Red)	VCC	CN6.4 (+3v3)
2(Black)	GND	CN6.6 (GND)
3(Green)	RST	CN6.3 (NRST)
4(White)	SCL	CN5.10(D15)
5(Orange)	SDA	CN5.9(D14)

2.2 Software Setup

2.2.1 Setup Serial console on your PC.

For Linux:

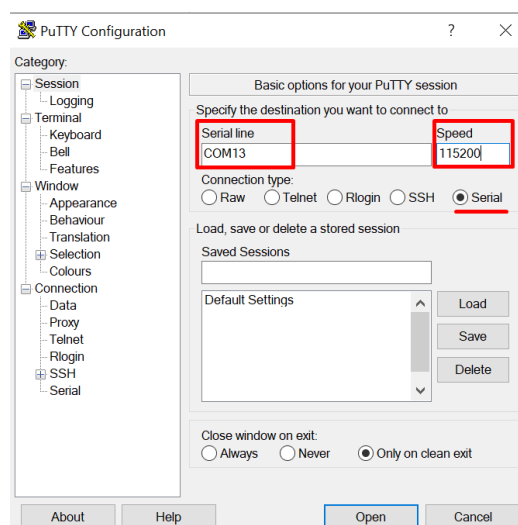
1. Ensure **minicom** is installed in Linux PC
2. [Use Minicom for the Linux PC.](#)
3. Open minicom serial setup window using below command
einfochips@dell:~\$ sudo minicom -s
4. Set baud rate and other setting as per below
 - Baud rate = 115200
 - Data = 8 bits
 - Parity = none
 - Stop = 1 bit
 - Flow control = none
 - Serial device **/dev/ttyACM0**
 - **save setup as dfl**

For Windows:

1. [Use Putty for the windows PC.](#)
2. Open Device Manager and check your COM Port as shown below.



3. Launch the Putty application, select the connection type: **Serial**. Set the COM Port: **COM13** (or whichever port was assigned in step 2) and Speed: **115200** as shown below and click on "Open" button.



Power ON the STM32WB55 Board.

Power ON the board. The serial console will start showing logs as seen below. Initial SBOOT logs indicate the Secure Bootloader and then the Application.

```
=====
= (C) COPYRIGHT 2017 STMicroelectronics =
= Secure Boot and Secure Firmware Update =
=====

= [SBOOT] SECURE ENGINE INITIALIZATION SUCCESSFUL
= [SBOOT] STATE: CHECK STATUS ON RESET
INFO: A Reboot has been triggered by a Hardware reset!
Consecutive Boot on error counter = 0
INFO: Last execution detected error was:No error. Success.
= [SBOOT] STATE: CHECK NEW FIRMWARE TO DOWNLOAD
= [SBOOT] STATE: CHECK USER FW STATUS
A valid FW is installed in the active slot - version: 1
= [SBOOT] STATE: VERIFY USER FW SIGNATURE
= [SBOOT] STATE: EXECUTE USER FIRMWARE0 7 [iot_thread] [INFO ][DEMO][lu] -----STARTING DEMO-----

1 17 [iot_thread] [INFO ][INIT][lu] SDK successfully initialized.
[optiga example] : Initializing OPTIGA for example demonstration...

[optiga util] : optiga_util_open_application
[optiga shell] : Initializing OPTIGA completed...

[optiga shell] : Begin pairing of host and OPTIGA...
[optiga example] : example_pair_host_and_optiga_using_pre_shared_secret
[optiga util] : optiga_util_read_metadata
[optiga crypt] : optiga_crypt_random
[optiga util] : optiga_util_write_data
[optiga util] : optiga_util_write_metadata
[optiga example] : Passed
```

NOTE: If this is not a new kit right out of the box and the user does not see the SBOOT logs as shown above, please refer to [STM32WB55-SSK Developers Guide.pdf](#) Section 2.2 Software Setup for further instructions.

2.2.2 AWS Account creation and Arrow Cloud Connect tool configuration

The items mentioned below are specific to enabling AWS Cloud Services with the Security Starter Kit and only need to be completed once. The output from these configuration steps can be reused to connect other Security Starter Kits to AWS Cloud Services. **These steps must be completed prior to running the included demo.**

1. It is presumed that the user has an AWS Management Console account, required to complete the steps listed below. If not, you will need to create an account: <https://aws.amazon.com/console/>
2. Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
 - The user will need to configure a unique EC2 instance, which will provide a unique URL and login credentials tied to your AWS account, for the Arrow Cloud Connect Tool. The EC2 configuration instructions are outlined in the [SSK Cloud Connect Quick Start Guide.pdf](https://www.arrow.com/en/products/stm32wb55-ssk/arrow-development-tools), <https://www.arrow.com/en/products/stm32wb55-ssk/arrow-development-tools>

2.2.3 Setup for Mobile App

Mobile App contains following feature:

- Scan and connect to nearby BLE devices running FreeRTOS
- Act as a proxy to transmit MQTT messages between a STM32WB55 and the AWS IoT cloud
- Mutual Authentication between STM32WB55 and Mobile App

Prerequisite:

- Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers.

Please follow the below steps to create an AWS Cognito user. This user will be needed to login into the Mobile App.

1. If you have created the Amazon Cognito user pool in AWS console then jump to step 5, otherwise continue to step 2
2. To create an Amazon Cognito user pool in AWS Console:
 - In AWS console, Open the Amazon Cognito console (Service), and choose **Manage User Pools**.
 - Choose **Create a user pool**.
 - Give the user pool a name, and then choose **Review defaults**.
 - From the navigation pane, choose **App clients**, and then choose **Add an app client**.
 - Enter a name for the app client, and then choose **Create app client**.
 - From the navigation pane, choose **Review**, and then choose **Create pool**.
 - **Make a note of the pool ID** that appears on the **General Settings** page of your user pool.

- From the navigation pane, choose **App clients**, and then choose **Show details**.
 - **Make a note** of the app client ID and app client secret.
3. To create an Amazon Cognito identity pool in AWS Console:
- In AWS Console, Open the Amazon Cognito console (Service), and choose **Manage Identity Pools**.
 - Choose **Create new identity pool**.
 - Enter a name for your identity pool.
 - Expand **Authentication providers**, choose the **Cognito** tab, and then enter your user pool ID and app client ID.
 - Choose **Create Pool**.
 - Expand **View Details**, and **make a note of the two IAM role names**. Choose **Allow** to create the IAM roles for authenticated and unauthenticated identities to access Amazon Cognito.
 - Choose **Edit identity pool**. **Make a note** of the identity pool ID. It should be of the form: **us-west-2:12345678-1234-1234-1234-123456789012**
4. To create and attach an IAM policy to the authenticated identity:
- In the AWS console, Open the IAM console, and from the navigation pane, choose **Roles**.
 - Find and choose your authenticated identity's role, choose **Attach policies**, and then choose **Add inline policy**.
 - Choose the **JSON** tab and paste the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:AttachPolicy",
        "iot:AttachPrincipalPolicy",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive",
        "iot:GetThingShadow",
        "iot:UpdateThingShadow",
        "iot>DeleteThingShadow"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```


- Choose **Review policy**, enter a name for the policy, and then choose **Create policy**.

Note : Kindly double check IAM policy as any misconfiguration will lead to issues while login to AWS cognito on mobile app

5. Create New User in Cognito User pool:

This can be done either from the Android application (login screen) or from the AWS console. If a confirmation is needed, this can be done from the AWS Cognito console (in Users and groups).

- Login AWS console - <https://aws.amazon.com/console/>
- AWS Console >> Cognito Service >> Manage User pool >>**
 - Select your newly created user pool from step #3
- General Settings >> Users and groups >> Create user**

The screenshot shows the AWS Cognito console for a user pool named 'stm32wb55_pool'. The 'Users' tab is active, showing a list of users. The table below represents the data visible in the console:

Username	Enabled	Account status	Email verified	Phone number verified
alpesh	Enabled	CONFIRMED	true	false
darshak	Enabled	CONFIRMED	true	true

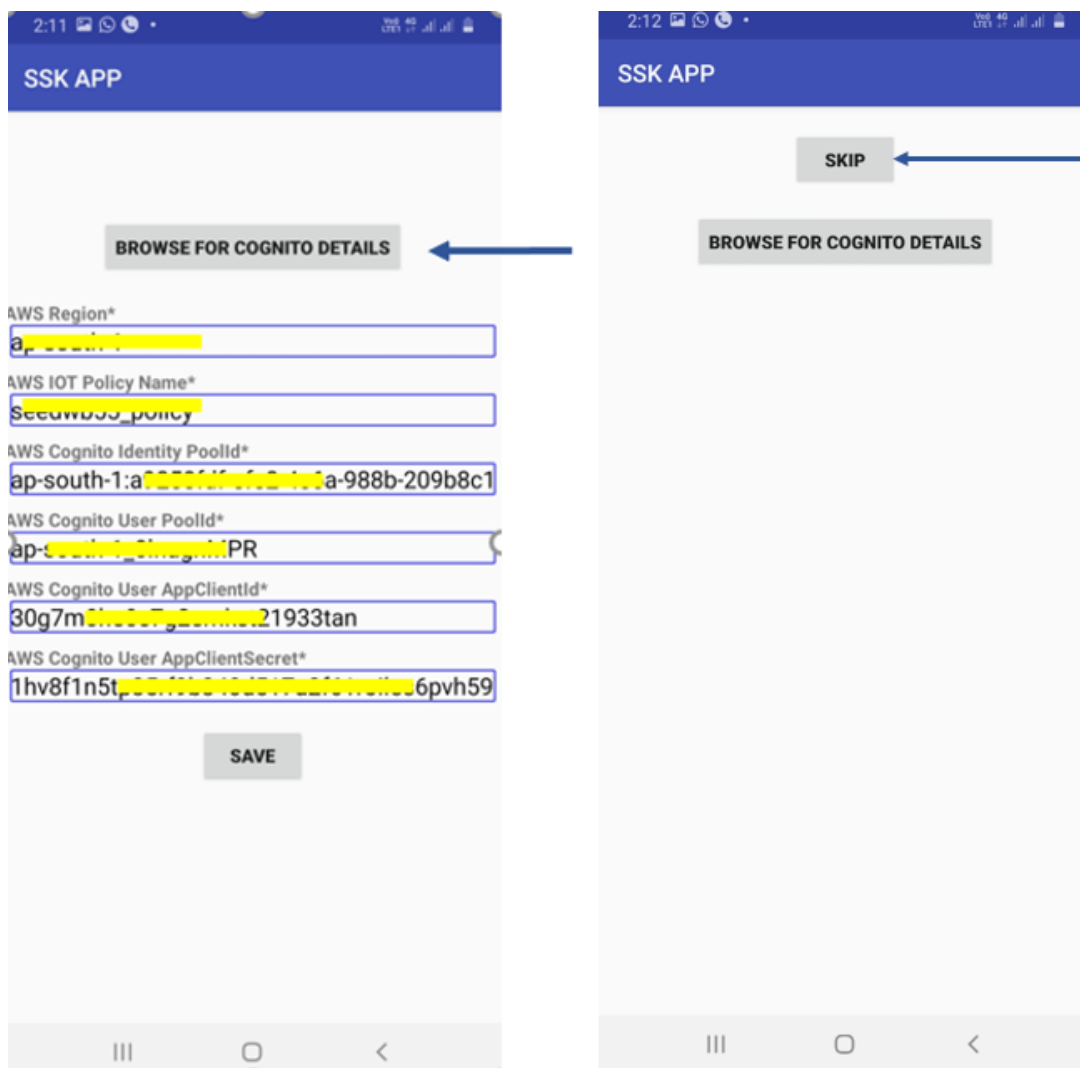
- Make a note of Username and Password**, which is used for login into the Mobile APP.
6. Create a new text file with the name 'AWS_Config.txt' and enter the following parameters in the file:

```
AWS_REGION#<Enter-Region>
AWS_IOT_POLICY#STM32WB55-policy
AWS_COGNITO_IDENTITY_POOLID#<Enter-Identity-Pool-ID>
AWS_COGNITO_USER_POOLID#<Enter-User-Pool-ID>
AWS_COGNITO_USER_APPCLIENTID#<Enter-User-App-ID>
AWS_COGNITO_USER_APPCLIENTSECRET#<Enter-User-App-SECRET>
```

7. Install the Mobile APK and run it. The APK can be found at the link:
<https://www.arrow.com/en/products/stm32wb55-ssk/arrow-development-tools>
8. Follow for FIRST-TIME SETUP ONLY: (otherwise press “SKIP”)
 Click on “Browse for the Cognito details” and upload the **AWS_Config.txt** file created in step#6, verify the configurations, and then Press “Save” button on screen as shown - [First Time configuration screen](#)

Note: If the Mobile App crashes after clicking “save”, please ensure the information in the AWS_Config.txt file was entered exactly as shown

[First Time configuration screen](#)
 Press Skip or reconfigure if required



3 DEMO SETUP

3.1 FreeRTOS MQTT Demo configuration

[**Note:** It is presumed that user has successfully created their own AWS account and completed the creation of an AWS EC2 instance for the Cloud Connect Tool, as outlined in Section 2.2.2.]

1. Enter the URL that was provided during the configuration of the AWS EC2 instance and outlined in the Security Starter Kit Cloud Connect Quick Start Guide
2. Login to the SSK Cloud Connect tool with your AWS Active Credentials, which were created when configuring your AWS EC2 instance.

SSK Security Starter Kit

Home Contact

SSK Security Starter Kit

UserName:
Enter email

Password:
Password

Login

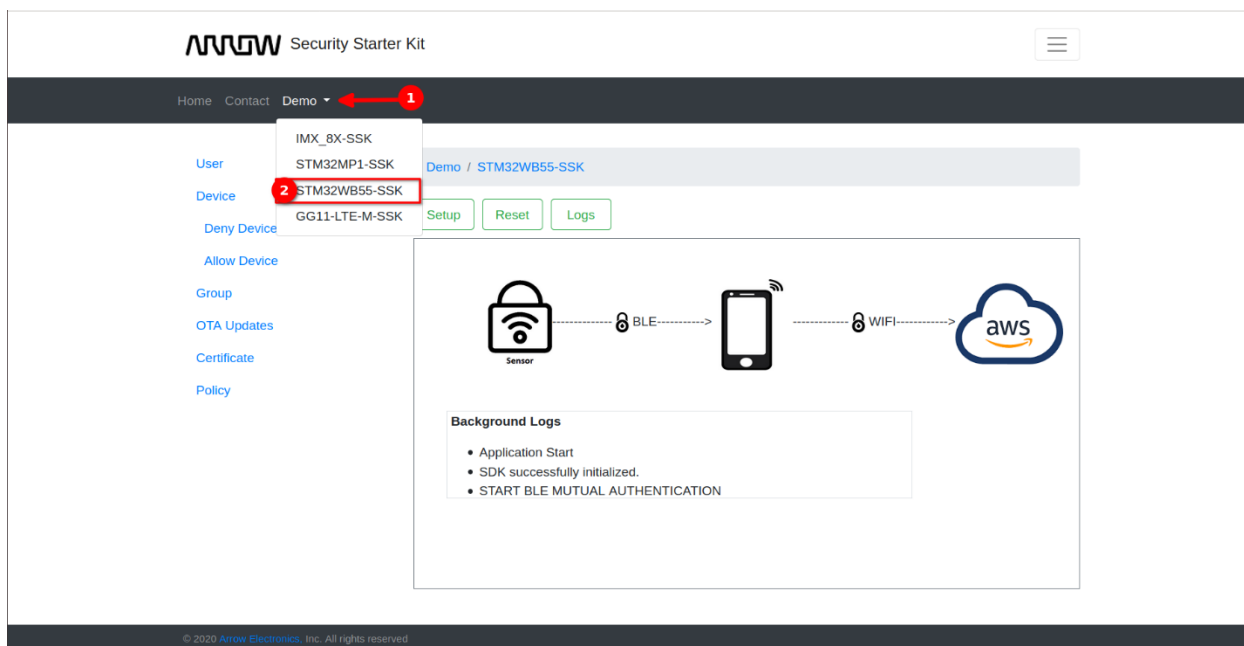
© 2020 Arrow Electronics, Inc. All rights reserved

1. STM32WB55 Quick Demo Setup

1. Collect STM32WB55 MAC Address from serial console by pressing the reset button on the board.

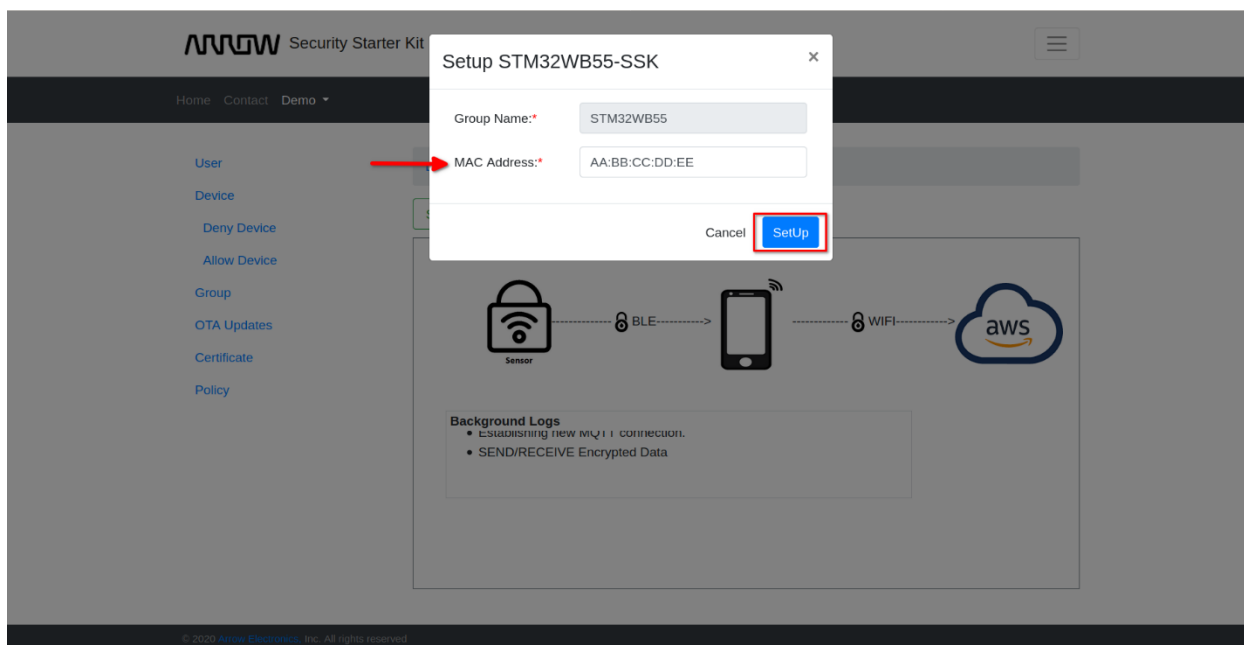
```
===== EUI64 Address =====  
[EUI64_Addr]:: 80:e1:26:01:00:07:e8:27  
===== MAC Address =====  
[MAC_Addr]:: 80:e1:26:07:e8:27  
=====
```

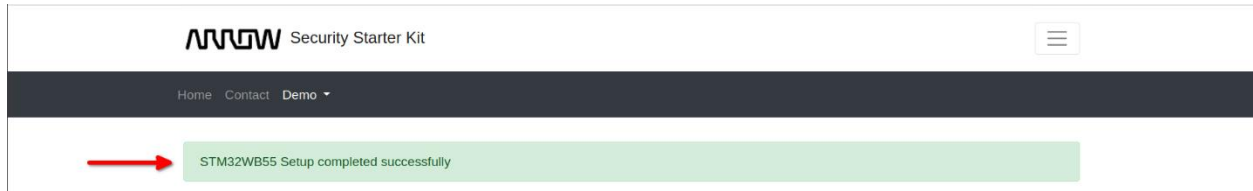
2. Open the STM32WB55 demo page. Go to [SSK Cloud Connect >> Demo >> STM32WB55-SSK](#).



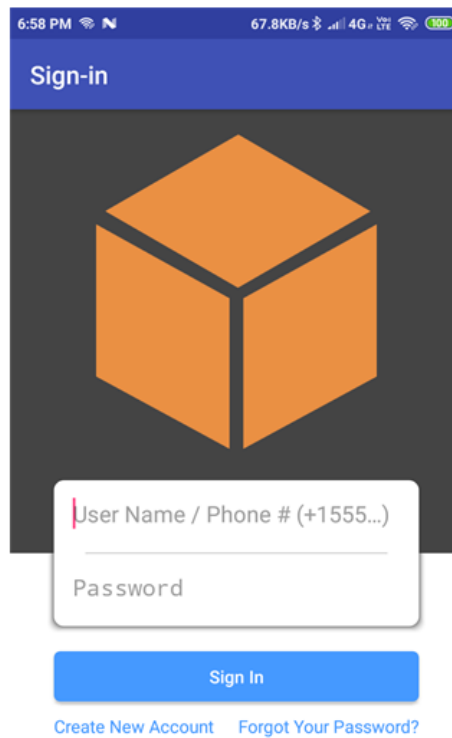
3. Click on the “Setup” button.
4. Enter your MAC address and click the “Setup” button.

Note: If “Setup Failed” error is seen, press the Reset button [SW4] and follow steps 3 & 4 again.



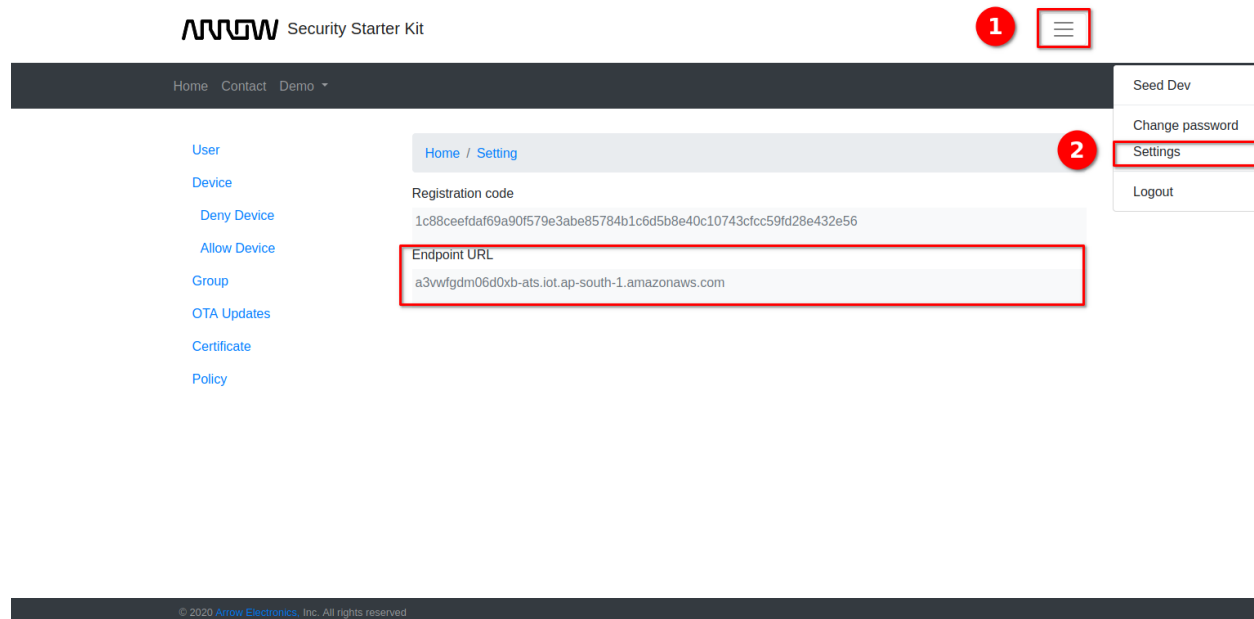


2. Sign into the App using the Cognito user credentials as mentioned above section 2.2.3 'Setup for Mobile App' step #5.



3. Configure Endpoint URL into the device

1. Copy Endpoint URL from [SSK Cloud Connect >> Settings](#)



Press the SW2 button and then Reset the board by pressing the “Reset” [SW4] button as shown in figure 2. It will ask the user to enter an endpoint URL in the Serial console screen as below:

```
===== Please Enter Your Endpoint URL =====
e.g- xxxxxxxxxxxx-ats.iot.xx-xxx-x.amazonaws.com

Entered Endpoint URL=a3vwfgdm06d0xb-ats.iot.ap-south-1.amazonaws.com
=====
```

Paste the Endpoint URL here and press the “Enter” key on keyboard. (**Note:** Default Echo is OFF, so you cannot see your Endpoint URL value.)

So, to verify that the correct endpoint URL was entered, on the serial console it will display the URL as shown above (with yellow redaction).

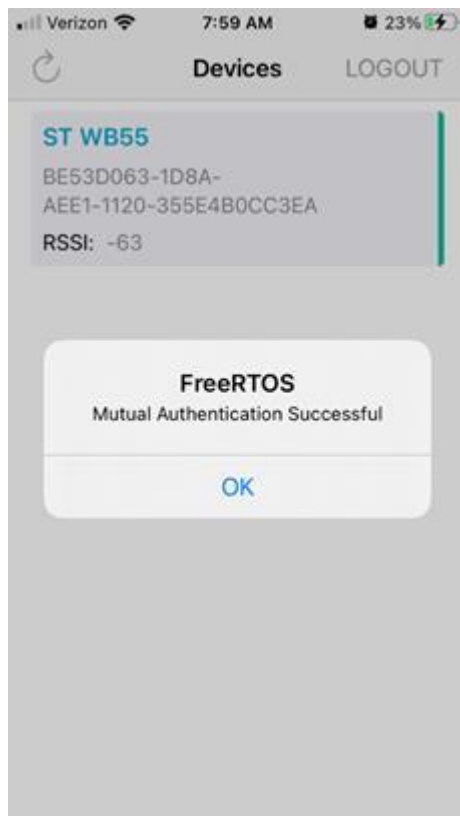
4. Scan and Pairing the STM32WB55 with Mobile.

1. For Android based Mobile App users, Click on the SCAN FOR BLE DEVICES button. Connect and Pair with STM32WB55 device. On the serial console to Press “Y” if passkey matches with the mobile pair key.



2. For Apple based Mobile App users, Click on the SCAN FOR BLE DEVICES button. Connect and Pair with STM32WB55 device. On the serial console to Press "Y" if passkey matches with the mobile pair key.

Note: watch closely for "mutual authentication successful" message and to quickly press OKAY shown below.



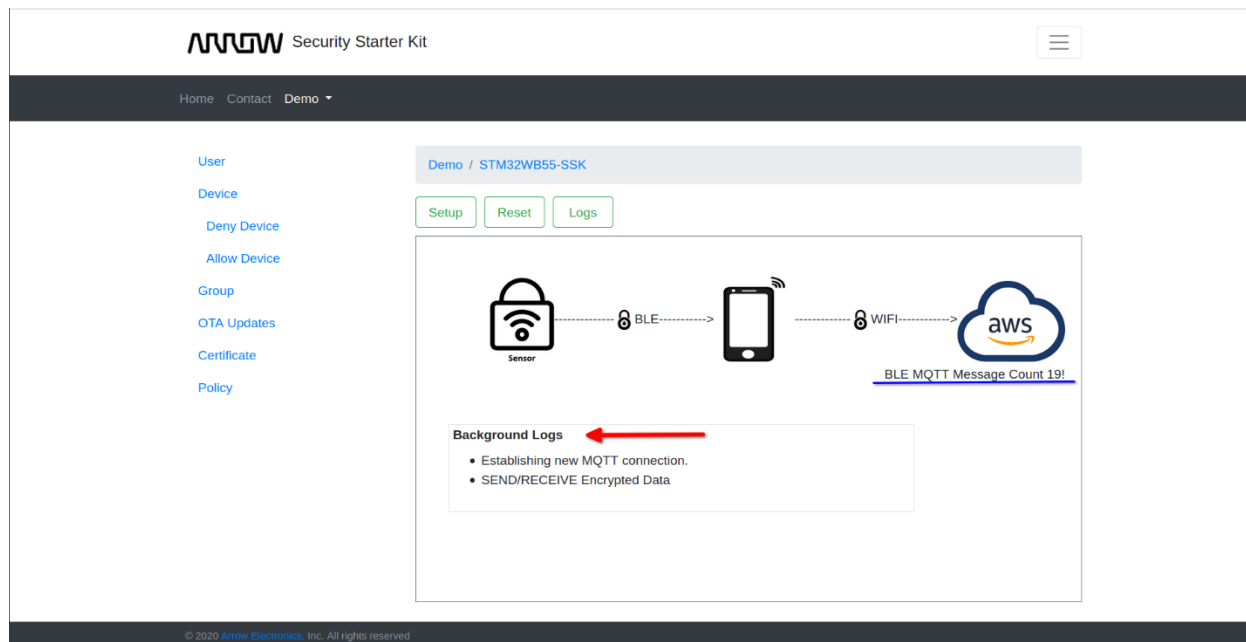
SSK STM32WB55 - MQTT Demo

Note: Kindly note to keep WIFI active (no airplane mode) as this will hamper AWS Communication.

This Demo shows how FreeRTOS based BLE device can securely communicate with AWS IoT Core Services using a Mobile app as a proxy. Only Authenticated devices can communicate with the Mobile app and the AWS cloud, which is achieved by implementing the **mutual authentication service** with help of OPTIGA™ Trust M Chip.

Note: Kindly note if Mutual Authentication fails, please verify the H/w Connection provided [Figure 2: Hardware Setup](#)

- ❖ On SSK Cloud Connect, one can check the demo
[SSK Cloud Connect >> Demo >> STM32WB55-SSK](#)



- MQTT Message will display on the screen i.e. 'BLE MQTT Message Count 2!'

```

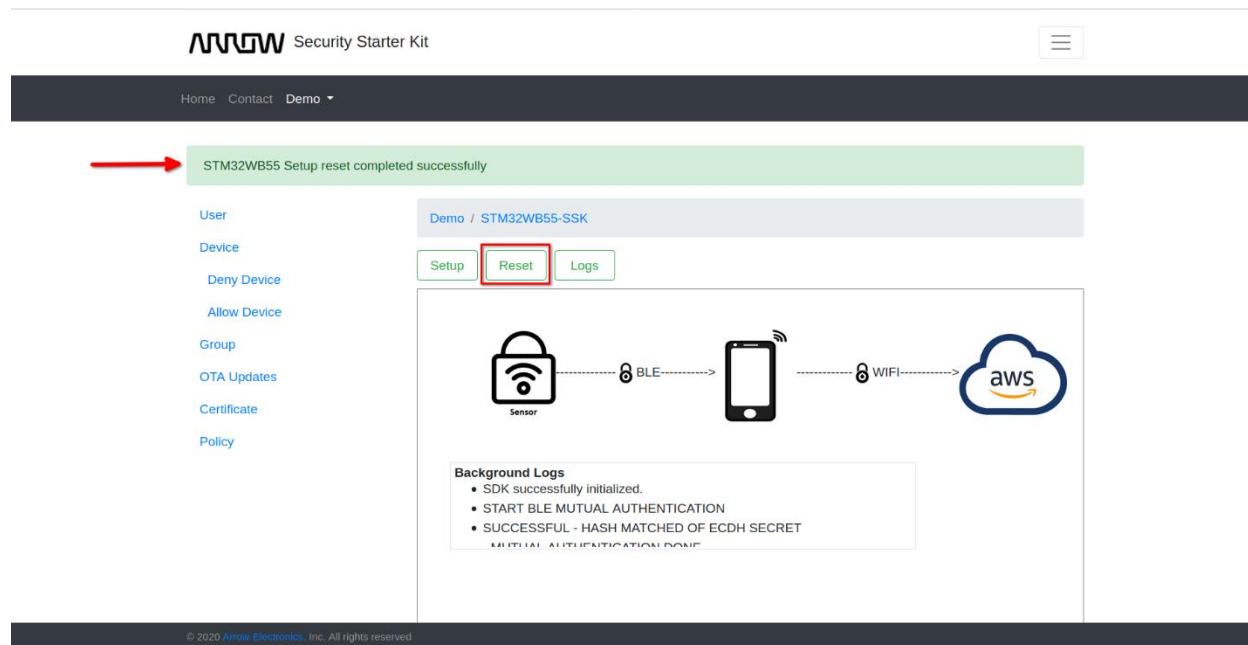
40 16521 [iot_thread] Incoming PUBLISH received:
Subscription topic filter: iotdemo/topic/2
Publish topic name: iotdemo/topic/2
Publish retain flag: 0
Pub141 16536 [iot_thread] (MQTT connection 0x200116b8) MQTT PUBLISH operation queued.
42 16545 [iot_thread] Acknowledgment message for PUBLISH 1 will beYooXogoo;/Refoooooo#####ooc#=#3EoUooAooEoo#sDv,ooN oooOIoo#t hao?o(
43 16557 [iot_thread] [SECURITY][SEND] CIPHER TEXT = oVocJWoo:Q#o
44 16571 [iot_thread] 2 publishes received.
45 16575 [iot_thread] Publishing messages 2 to 3.
46 16580 [iot_thread] Plain Text=BLE MQTT Message Count 2!
47 16591 [iot_thread] [SECURITY][SEND] CIPHER TEXT = oVocJW_oo-o|oUo'ooPooo\oo(ooGooooonSooJooT+MW
48 16601 [iot_thread] (MQTT connection 0x200116b8) MQTT PUBLISH operation queued.
49 16609 [iot_thread] Plain Text=BLE MQTT Message Count 3!
Zodooo*cSo)ooow7dooo! [SECURITY][SEND] CIPHER TEXT = oVocJW_oo-o|oUo'ooPooNooQoo

```

- On the SSK Cloud Connect dashboard, MQTT Messages are received in the AWS Cloud securely using hardware security chip - OPTIGA™ Trust M.
- Background Logs – Displays the Secure bootloader, BLE Application and OPTIGA™ Trust M message on the dashboard as a continuous scroll.

- ◆ If a user wants to reset the Setup

- Please follow: [SSK Cloud Connect >> Demo >> STM32WB55-SSK >> Press “Reset” button](#)
- Reboot the STM32WB55 board by pressing the “Reset” Button.



3.2 Demo Inference

The Security Starter kit with STM32WB55 and OPTIGA™ Trust M Demo provides examples of the below listed functionalities:

1. **AWS Cognito** – It provides simple user identity and data synchronization service that helps securely manage and synchronize App data for your users across their mobile devices
2. **Mutual Authentication** – The STM32WB55 device performs mutual authentication with the Mobile APP using OPTIGA™ Trust M before starting any communication. Only an authenticated device can communicate with the AWS IoT Core Services via Mobile app.
3. **Secure Communication** – Secure communications between AWS IoT Core Services and the STM32WB55 are ensured by storing the session credentials in the OPTIGA™ Trust M.
4. **FreeRTOS** – Used open-source FreeRTOS for the driver and Application development. This includes securely connecting your small, low-power devices to AWS cloud services like [AWS IoT Core](#)
5. **Secure Boot** – Enabled secure bootloader features on STM32WB55 Board.

Note: For more details about all above functionalities, please refer the following documents, located here: <https://www.arrow.com/en/products/stm32wb55-ssk/arrow-development-tools>

- [STM32WB55-SSK Developers Guide.pdf](#)
- [Security Starter Kit Cloud Quick Start Guide.pdf](#)
- [Security Starter Kit Cloud Connect Installation & Setup Guide.pdf](#)
- [Security Starter Kit Cloud Connect Users Guide.pdf](#)