AzureWave AW-CU427-USB IoT Connectivity Evaluation Board for AWS IoT Core User Guide

AzureWave Technologies, Inc.

Contents

AW-CU427-USB

IoT Connectivity Evaluation Board for AWS IoT Core

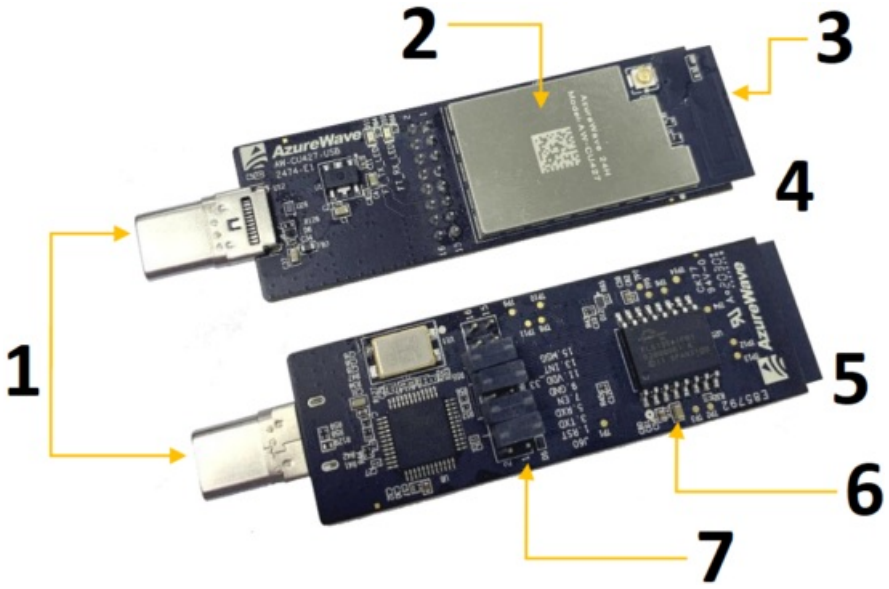
Getting Started Guide

Rev. 0.1

Revision History

Version	Revision Date	Description	Initials	Approved
0.1	2021/01/14	Initial version	Steven Jian Jackson Boon Josh Lin	Chihhao Liao Patrick Lin

1. Overview of AW-CU427-USB

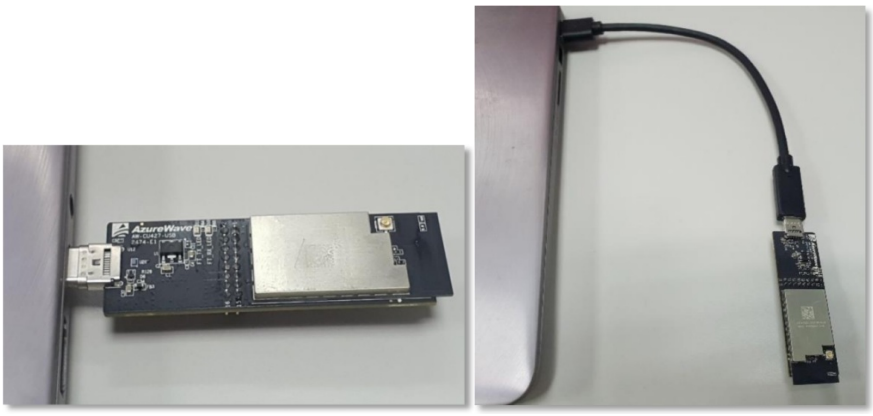


1. USB Type-C male Plug
2. Powerful MCU
3. Printed Antenna
4. Top side
5. Bottom side
6. SPI Flash
7. 8-Pin header

2. Hardware Setup

2.1 Components

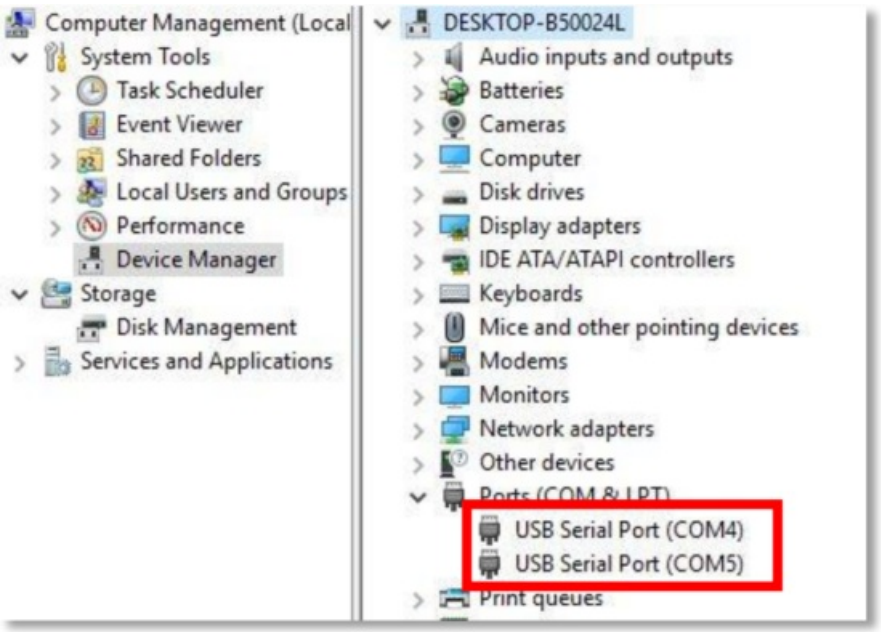
- Plug the AW-CU427-USB dongle into the USB type C port or through an USB type A male to USB type C female cable (not included in the box) if the computer does not have the USB type C port or it can not recognize the dongle



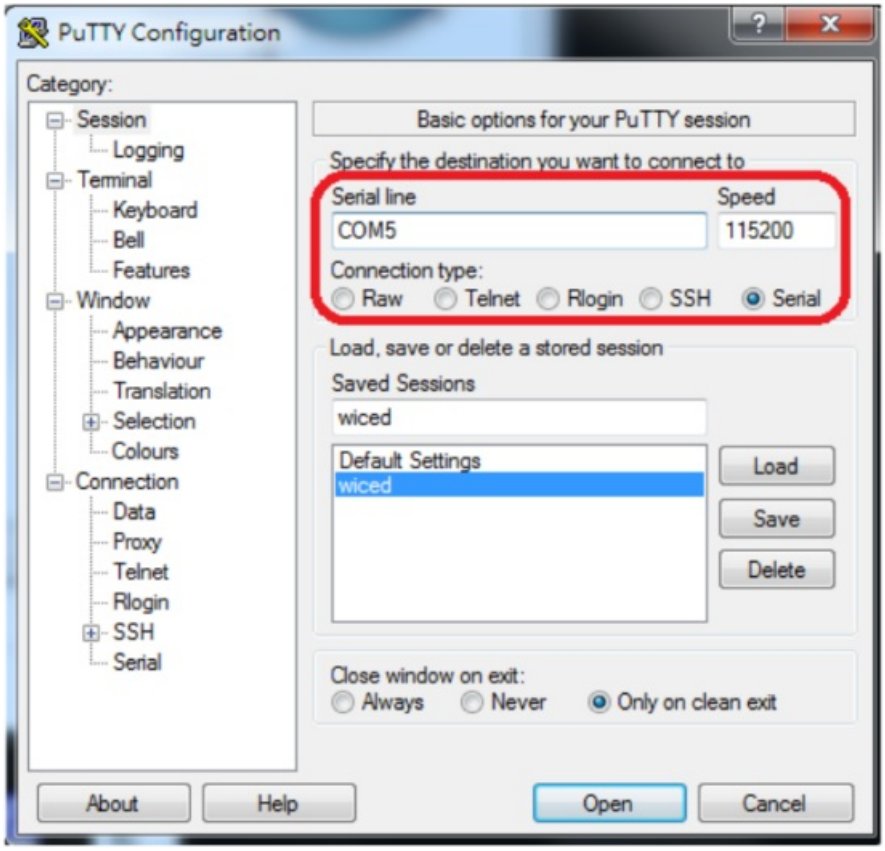
2.2 Identify the Hardware and Firmware

- The simple way to identify the hardware and firmware of AW-CU427 is under Windows. You also can identify it under Linux.

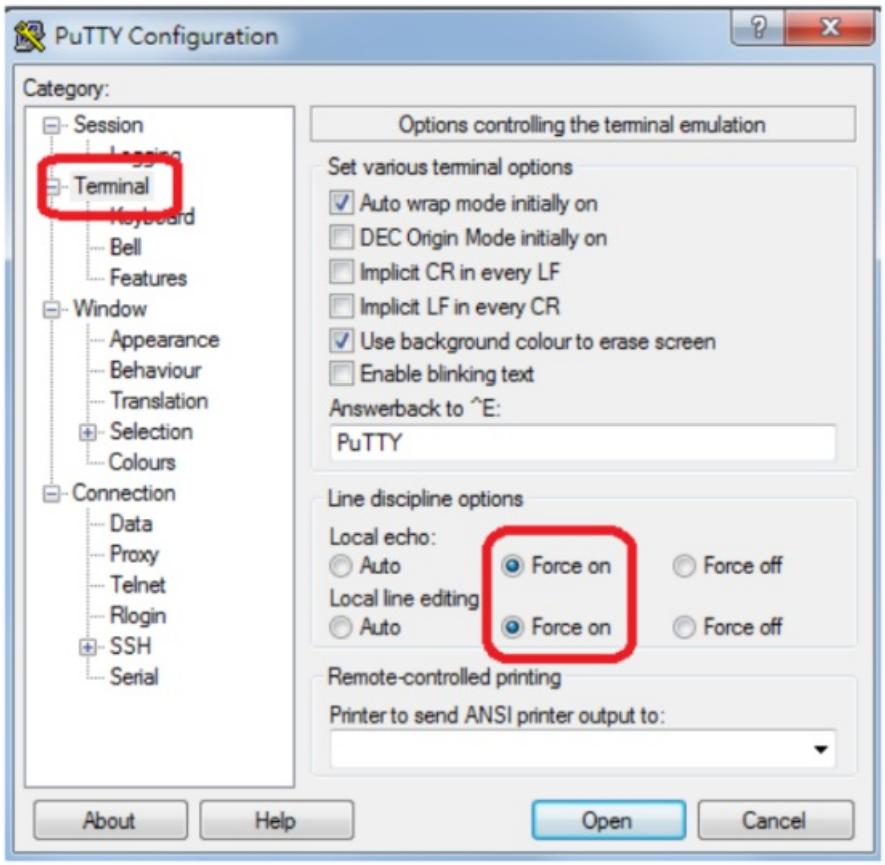
  Check if Device Manager can recognize AW-CU427-USB. If not, download FTDI Driver at https://www.ftdichip.com/Drivers/VCP.htm, then send the AT commands through the second FTDI com port



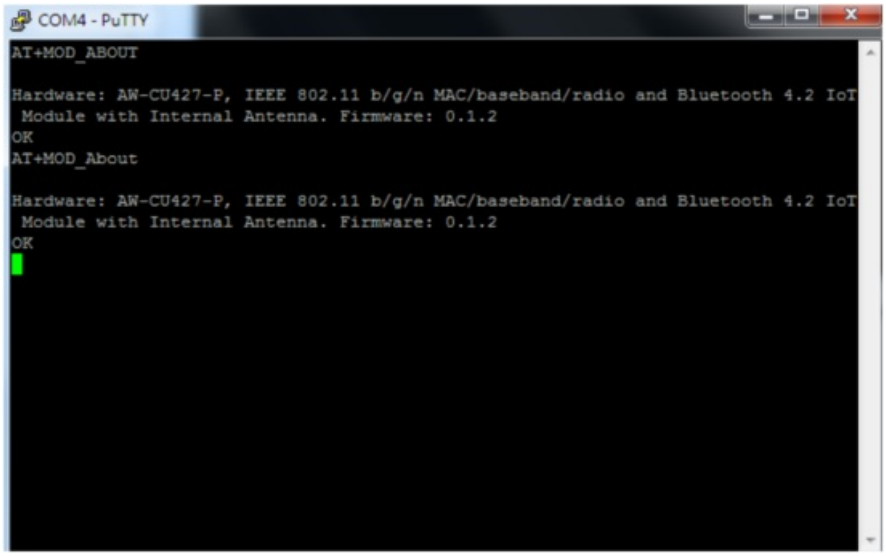
- Open a serial connection by using PuTTY or the similar terminal emulator



- To force the terminal show characters you type and send the command only after you hit the “Enter” key, set Local echo & Local line editing to “Force on”



- Type **AT+MOD_About** and hit “Enter” key to identify the dongle hardware and firmware



2.3 Antenna Installation Guide

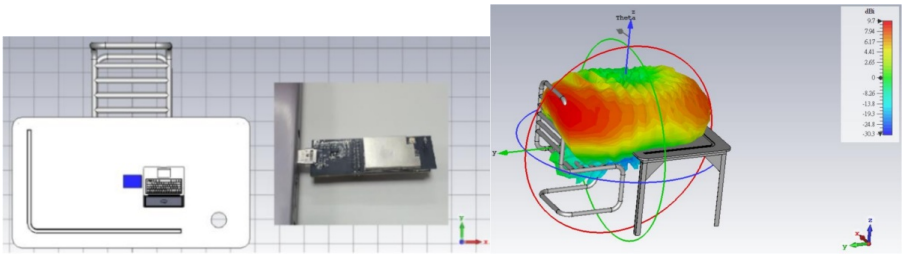
The AW-CU427-USB includes an integrated omnidirectional antenna. Hence, the dongle can be seen as an antenna.

Objects close to the antenna (the dongle) become new sources for radiation from far field perspective. This can be a problem if those objects are electrically large. Recommended minimum clearance is 20mm for conductive materials and 10mm for non-conductive materials. Failing to meet the clearance requirements will detune the antenna and the radiation pattern may not be omnidirectional.

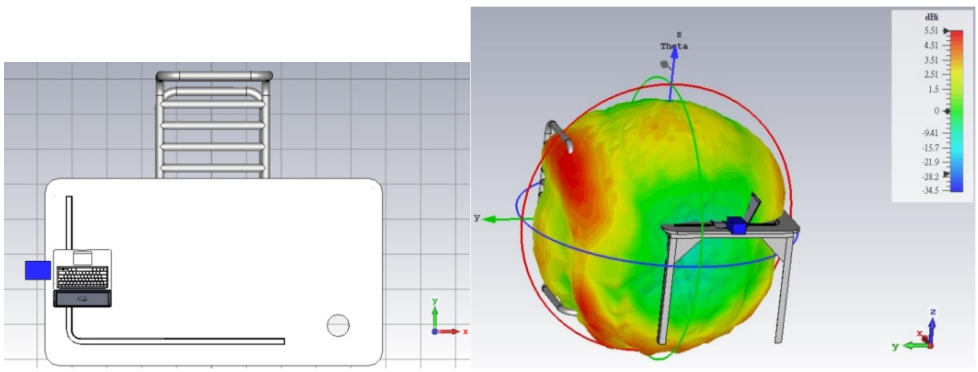
Use AT command **AT+WIFI_On** and **AT+WIFI_Scan** to check if the RSSI (Received Signal Strength Indication) is good (>-65dBm) or not. If not, adjust the setup until the RSSI >-65dBm.

The examples below are antenna radiation patterns with different setups:

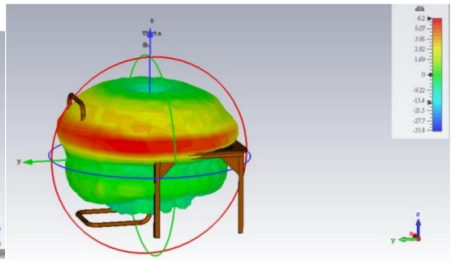
- Steel Desk & Chair + Horizontal Position



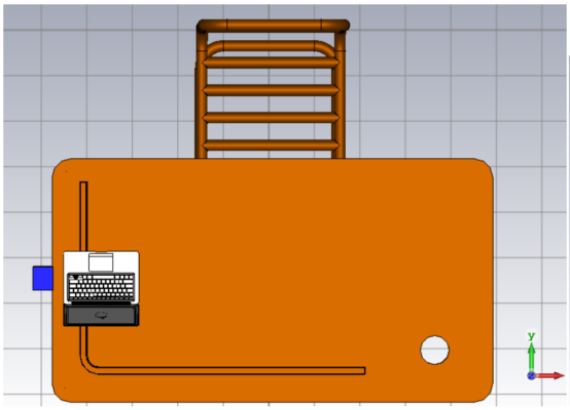
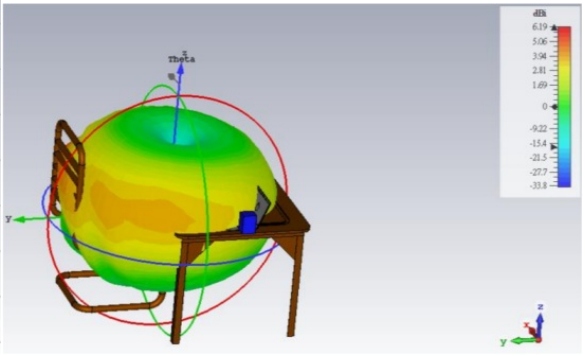
- Steel Desk & Chair + Horizontal Position + Close to Desk Edge



- Wood Desk & Chair + Vertical Position

- Wood Desk & Chair + Vertical Position + Close to Desk Edge

3. AWS Command Example

Please find the command details in **AWS CONNECTOR AT Command Set.**

Below are commands for the demo:

1. Turn on Wi-Fi module: **AT+WIFI_On**
2. Retrieve the Wi-Fi AP | Station Mode: **AT+WIFI_GetMode**
3. Perform a Wi-Fi network scan: **AT+WIFI_Scan**
4. Set and store the Wi-Fi AP information when AW-CU427-USB in Station Mode:
   AT+WIFI_SetAP=SSID,password, security type
   SSID: SSID of AP (case sensitive)
   password: password for AP (case sensitive)
   security type: OPEN | WEP | WPA | WPA2
5. Connect to the AP: **AT+WIFI_Connect**
6. Define and store Thing-specific configuration:
   AT+THING_Set=client ID,endpoint,client certificate,client private key
   client ID: Thing name(Client ID)
   endpoint: AWS IoT endpoint URL
   client certificate: Certificate for this Thing
   client private key: Private key for this Thing
7. Connect the client to MQTT broker: **AT+MQTT_Connect**
8. Subscribe to and save MQTT topic: **AT+MQTT_Subscribe**=<topic>,<qos>
9. Publish to MQTT topic:

AT+MQTT_Publish=topic,message,qos

topic: Topic to publish to

message: Message to publish

qos: 0 | 1

3.1 Getting Started with AWS IoT Core

Step 0: The below link is a documents of how to setup AWS IOT, you can refer to it for full AWS IOT knowledge.

https://docs.aws.amazon.com/iot/latest/developerguide/iot-gs.html

But, if you want to setup AzureWave AWS Connector, you would just refer to the following steps.

Step 1: Create AWS Account, Create an IAM user.
Please refer to the below link to setup AWS Account and IAM user.

https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html

If you have created an IAM user, please refer to the following setting to connect these two policies (AmazonFreeRTOSFullAccess, AWSIoTFullAccess) to your IAM.

https://docs.aws.amazon.com/freertos/latest/userguide/freertos-account-and-permissions.html

Step 2: Create a thing.
A thing represents a specific device or instance that can communicate with AWS IOT. Please refer to the following link to create a thing.

https://docs.aws.amazon.com/iot/latest/developerguide/create-aws-thing.html

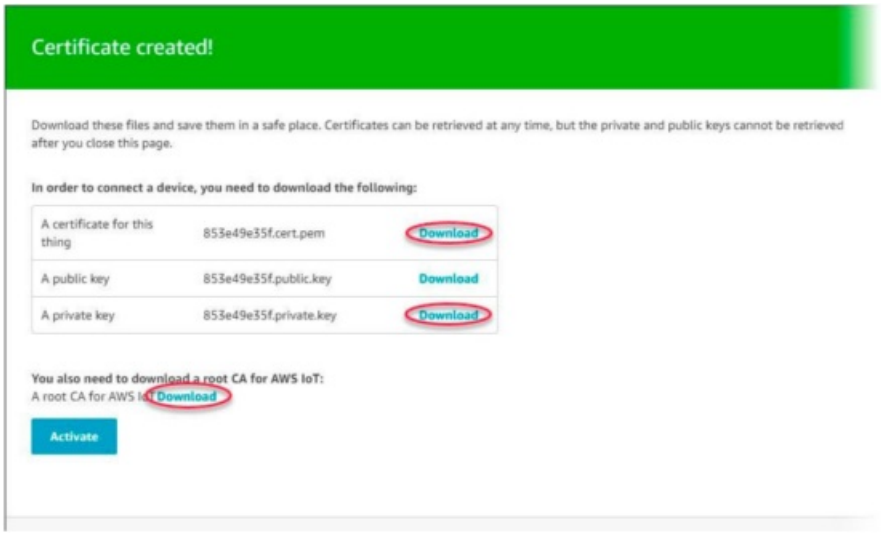
Step 3: Register a device
This step will create certificate and private key. You can use certificate, private key, thing name and endpoint as **AT+THING_Set** command parameter. After this command executing, the four parameter will be provision to our connector. After provisioning, you can connect to AWS IOT with MQTT or SHADOW operation.
Please refer to the steps at the following link.

https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html

After finishing the steps, please notice the following two actions:

- Download certificate and private key

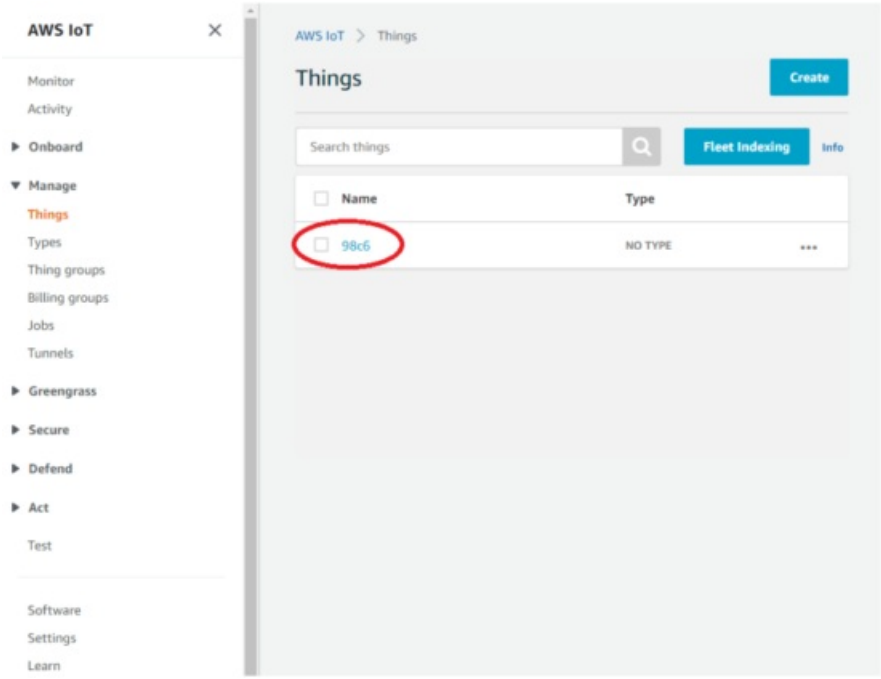
In Create and activate a device certificate chapter, please download and keep the certificate and private key. Because they will be used when sending the **AT+THING_Set** command.

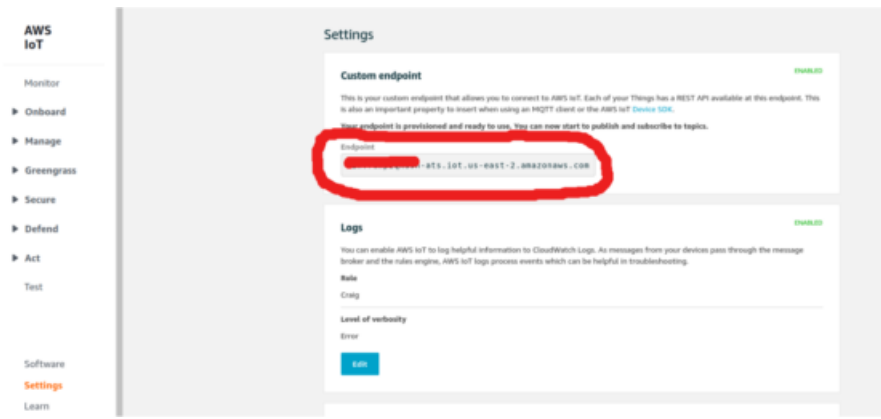


- Thing Name and Endpoint

These two data will also be used for **AT+THING_Set** command.

You can find out thing name in Manage > Things submenu, and endpoint in settings of AWS IOT Console at console.aws.amazon/iot.





3.2 Publish and Monitor MQTT message on the cloud

Step 0: Make sure the AP under test is connected to the internet using other Wi-Fi enabled devices. Assume the SSID, password and security type of the AP is MySSID, MyPassword, wpa2.

Step 1: Connect the AW-CU427-USB to the system (refer to 2. Hardware Setup) and turn off the wireless devices near the AW-CU427-USB (except for the device under test). Turn on the Wi-Fi module of the AW-CU427-USB using **AT+WIFI_On**

Step 2: Check if AW-CU427-USB is in station mode using **AT+WIFI_GetMode**

Step 3: Set and store information of the AP using **AT+WIFI_SetAP= MySSID,MyPassword,wpa2**

AT+WIFI_SetAP=SSID,password,security type
SSID: SSID of AP (case sensitive)
password: password for AP (case sensitive)
security type: OPEN | WEP | WPA | WPA2

Step 4: Connect to the AP: **AT+WIFI_Connect**

Step 5: Define and store Thing-specific configuration using **AT+THING_Set** command
AT+THING_Set=client ID,endpoint,client certificate,client private key
client ID: Thing name(Client ID)
endpoint: AWS IoT endpoint URL
client certificate: Certificate for this Thing (downloaded in 3.1 step 3)
client private key: Private key for this Thing (downloaded in 3.1 step 3)

You should create command as format below:

AT+THING_Set=98c6, a3qjEXAMPLEffp-ats.iot.us-east-1.amazonaws.com, ––BEGIN CERTIFICATE––\n…base64 data…\n––END CERTIFICATE––\n, ––BEGIN RSA PRIVATE KEY––\n…base64 data…\n––END RSA PRIVATE KEY––\n

Note: a3qjEXAMPLEffp is just an example endpoint, your endpoint URL should replace it. Please follow the instructions to find the endpoint.

- Navigate to the AWS IoT console
- Choose Settings in the navigation pane
- The endpoint can be found under Custom endpoint

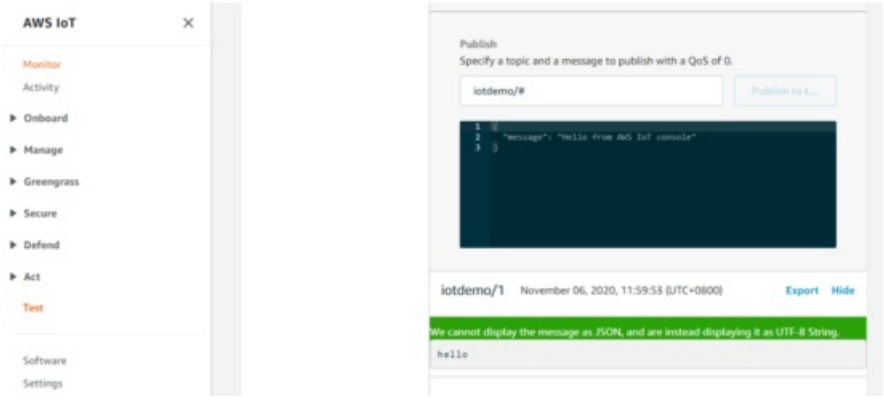
Step 6: Connect the client to MQTT broker: **AT+MQTT_Connect.**
Step 7: Subscribe to and save MQTT topic using **AT+MQTT_Subscribe=iotdemo/1,0**
Step 8: Publish to MQTT topic using **AT+MQTT_Publish=iotdemo/1, hello, 0**
AT+MQTT_Publish=topic,message,qos
topic: Topic to publish to
message: Message to publish
qos: 0 | 1

Step 9: Use the MQTT client in the AWS IoT console to monitor the messages that device sends to the AWS Cloud.

Sign in to the AWS IoT console.
https://console.aws.amazon.com/iotv2/
In the navigation pane, choose **Test** to open the MQTT client.

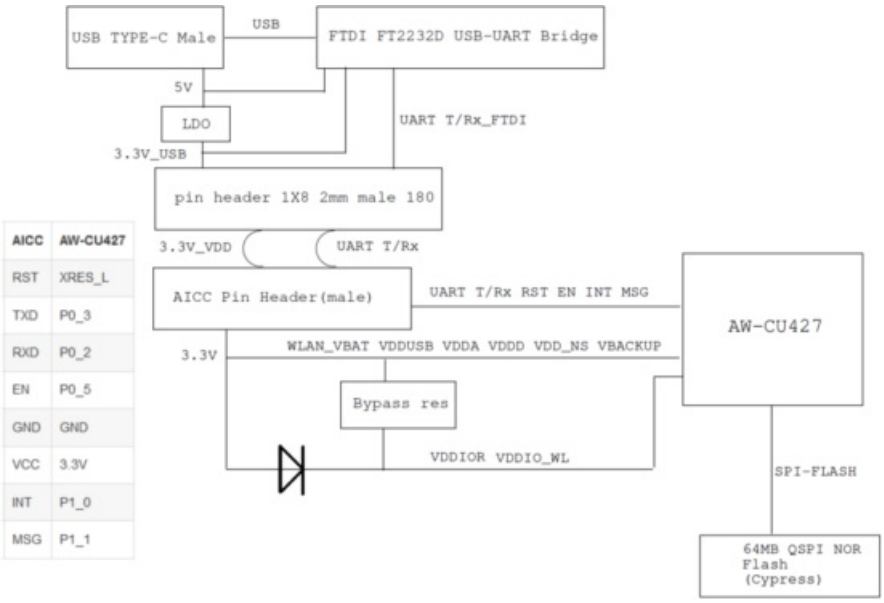
In **Subscription topic**, enter **iotdemo/#**, and then choose **Subscribe to topic**.

You should see the message send from device as like below.



4. Block Diagram and Schematic

4.1 Block Diagram for AW-CU427-USB

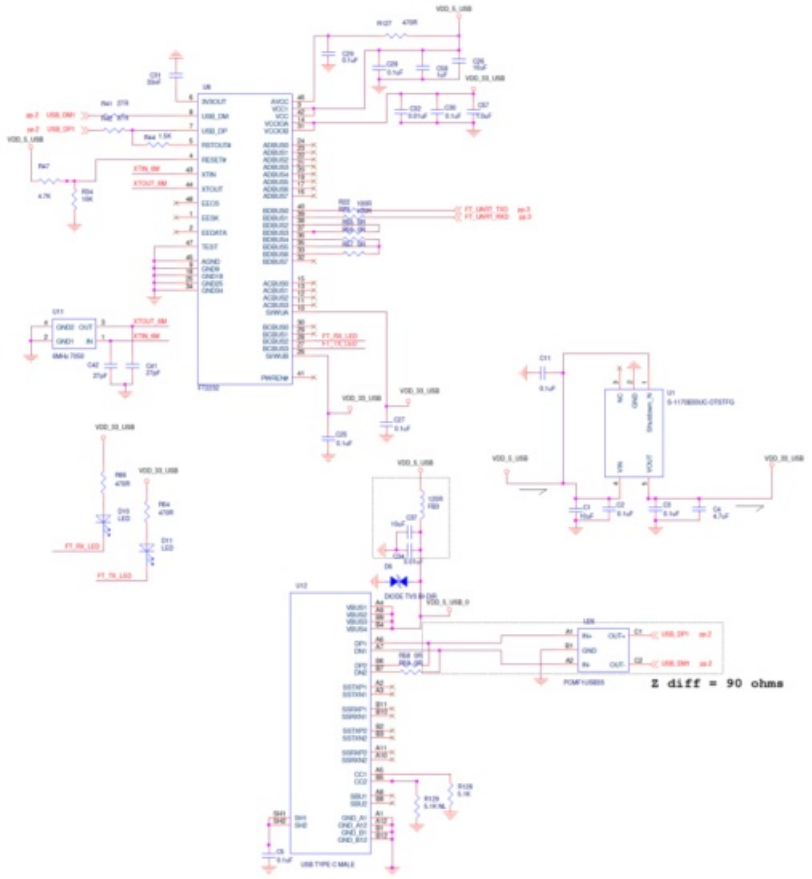


AICC	AW-CU427
RST	XRES_L
TXD	P0_3
RXD	P0_2
EN	P0_5
GND	GND
VCC	3.3V
INT	P1_0
MSG	P1_1

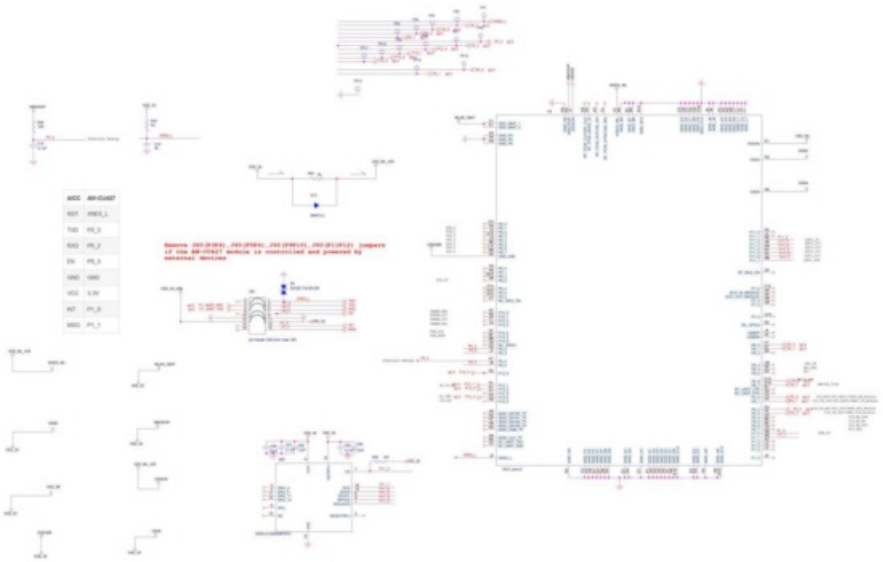
Pin Name (J60) is also shown on the bottom of PCBA

4.2 Schematics for AW-CU427-USB

FTDI Bridge (USB to UART/JTAG)



Mount CC2 pin pull-down R129 if connect the dongle to the host through a type c(male) to type c(female) cable. Some PC/NBs can not recognize the dongle when using USB type C ports. In that case, connect the dongle to the PC/NB through a type A(male) to type C(female) cable.



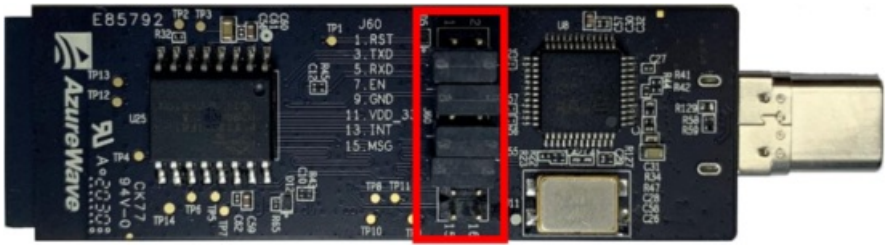
5. Additional Information

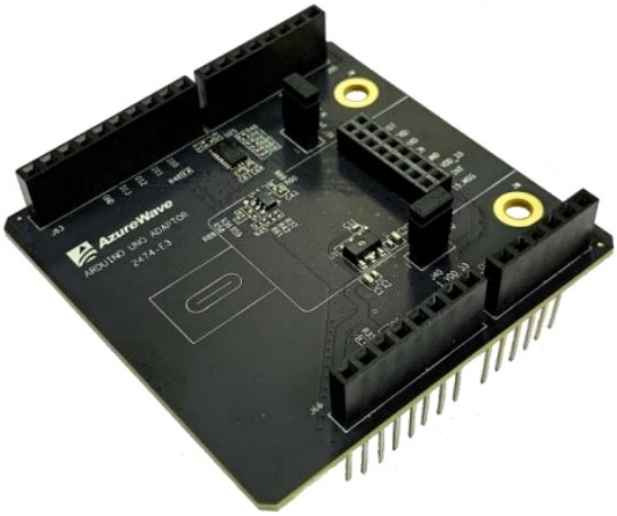
The AW-CU427-USB is designed for **Engineering Evaluation Purpose** Only as opposed to an end product. Thus, it may not comply with directives such as WEEE, CE and FCC. The dongle is not completely enclosed with a shielding case. User must take whatever measures to prevent ESD and minimize interference generated by the dongle. **For preventing any damage on this dongle, please plug it into USB female port horizontally.**

Appendix

AW-CU427-USB work with Arduino Uno

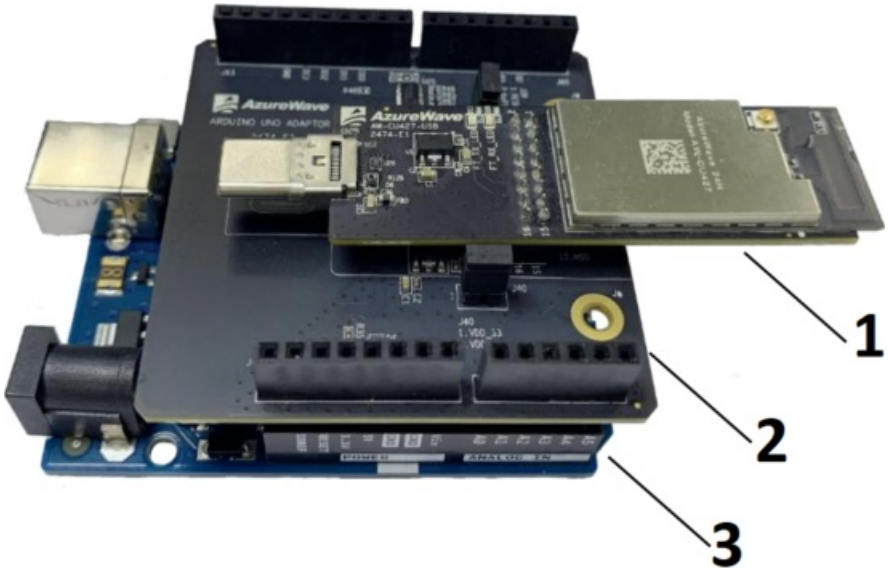
Remove all the jumpers (2mm pitch) on the dongle before mating it with the Arduino Uno through the ARDUINO UNO ADAPTOR. **The digital logic level & power supply input of the dongle are 3.3V** . **DO NOT directly connect the dongle to the Arduino Uno.**





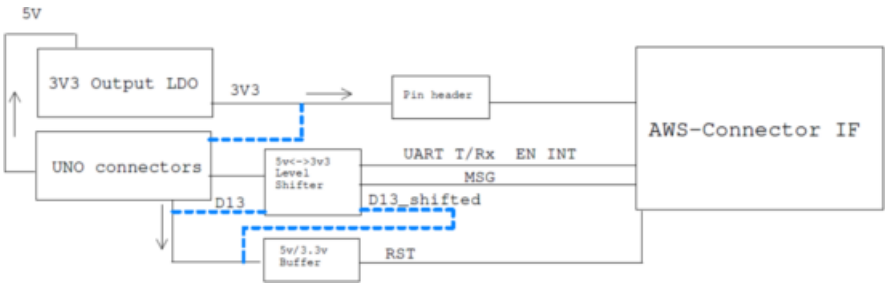
ARDUINO UNO ADAPTOR

- AW-CU427-USB connected to Arduino Uno through the ARDUINO UNO ADAPTOR

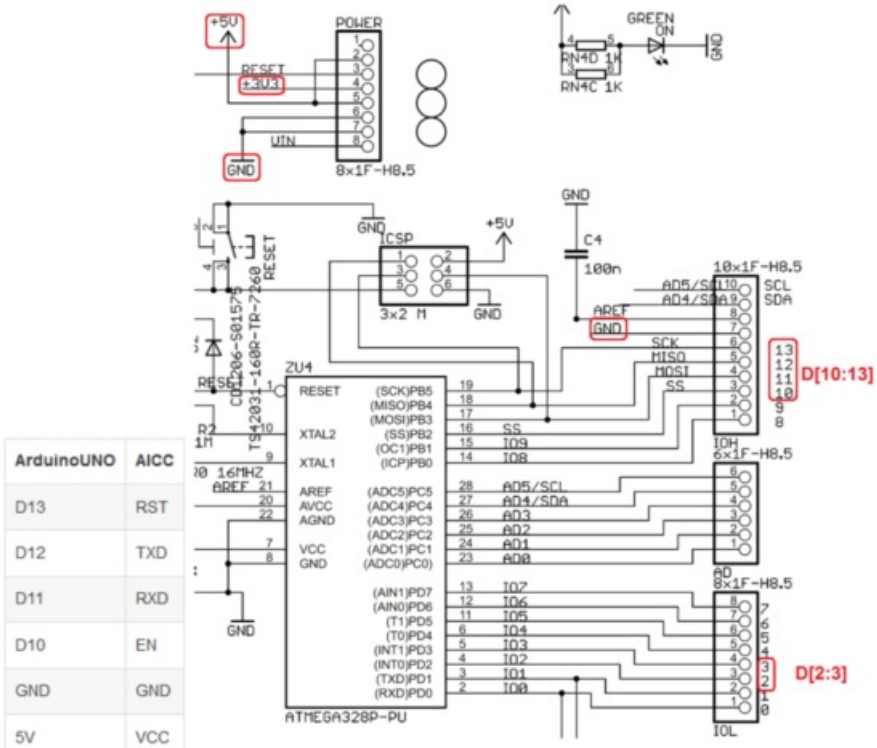


1. AW-CU300AV3-USB
2. Arduino Uno Adaptor
3. Arduino Uno

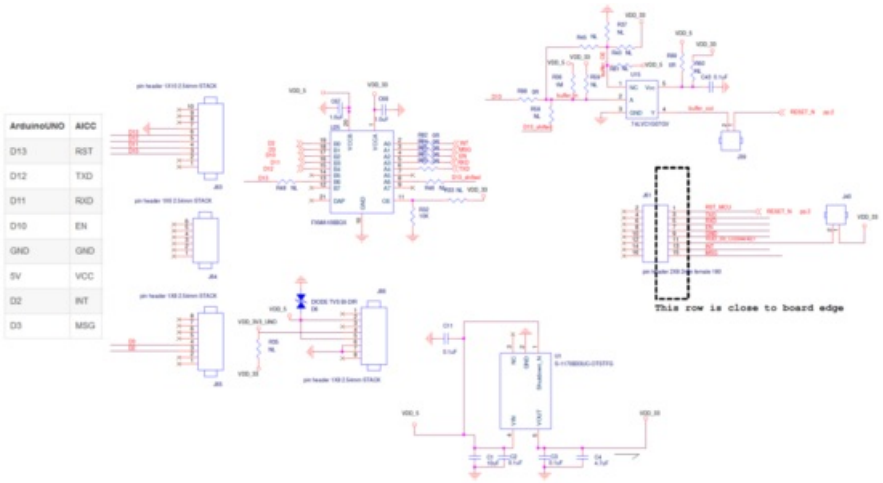
- Block Diagram for Arduino Uno Adaptor



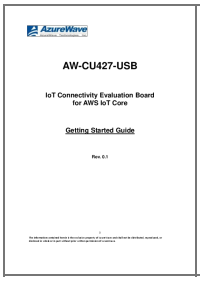
- Schematics for Arduino Uno Adaptor



ArduinoUNO	AICC
D13	RST
D12	TXD
D11	RXD
D10	EN
GND	GND
5V	VCC
D2	INT
D3	MSG



Documents / Resources

	AzureWave AW-CU427-USB IoT Connectivity Evaluation Board for AWS IoT Core [pdf] User Guide AW-CU427-USB, IoT Connectivity Evaluation Board for AWS IoT Core

References

- a3qjEXAMPLEffp-ats.iot.us-east-1.amazonaws.com
- console.aws.amazon.com/iotv2/
- First steps - FreeRTOS First steps - FreeRTOS
- How to manage things with the registry - AWS IoT Core How to manage things with the registry - AWS IoT Core
- Getting started with AWS IoT Core - AWS IoT Core
- How to manage things with the registry - AWS IoT Core How to manage things with the registry - AWS IoT Core
- Set up your AWS account - AWS IoT Core
- VCP Drivers - FTDI