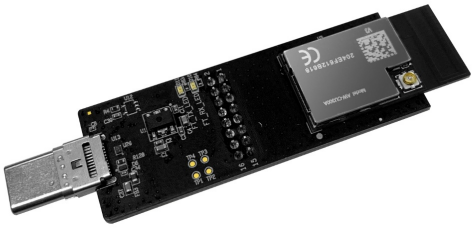
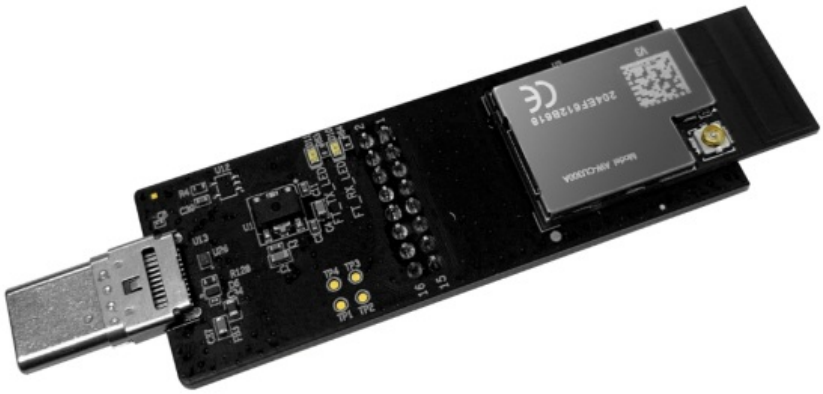


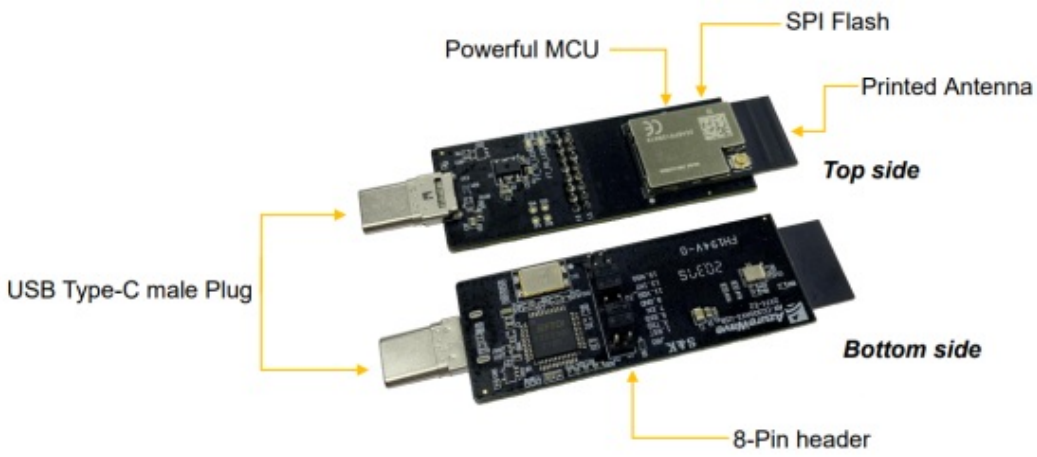
AzureWave AW-CU300AV3-USB WLAN Microcontroller Module with Type-C Dongle User Guide

AzureWave AW-CU300AV3-USB WLAN Microcontroller Module with Type-C Dongle User Guide



Contents

Hardware Specification

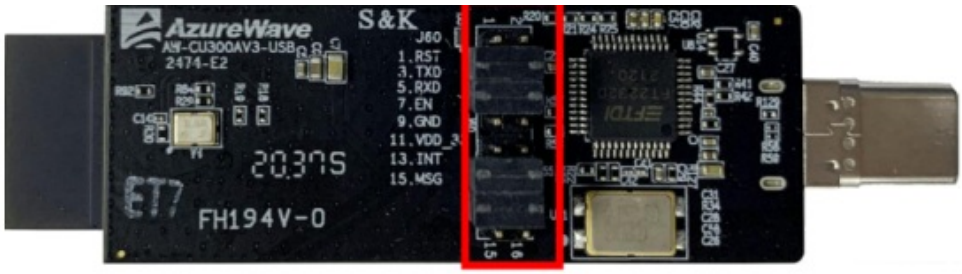


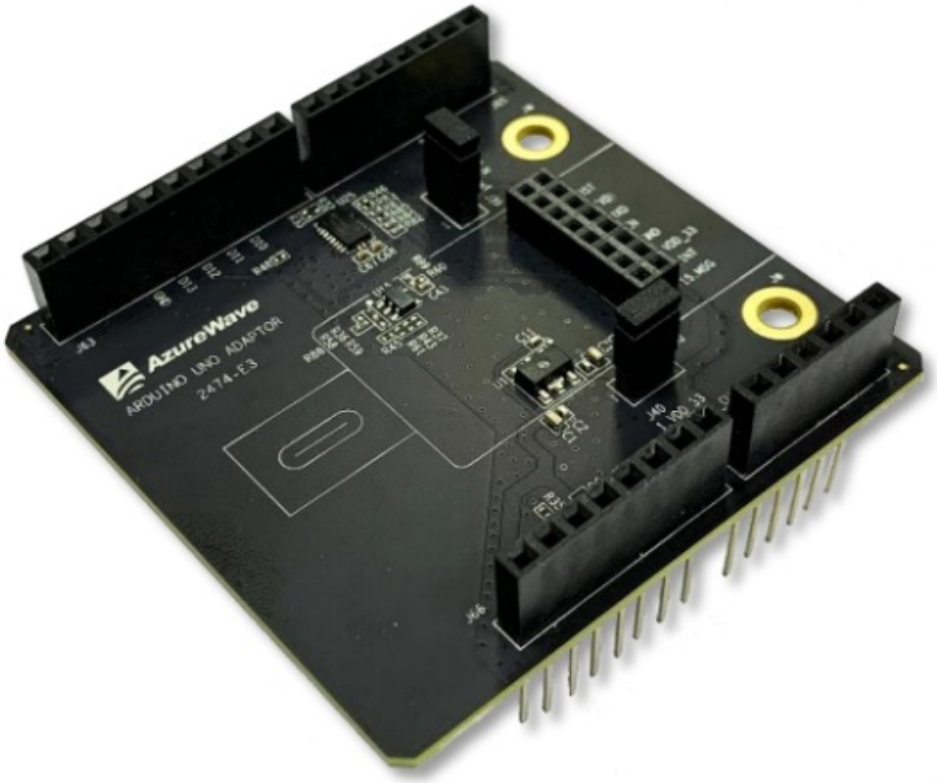
Hardware Setup

Components

- Plug the AW-CU300AV3-USB dongle into the USB type C port or through an USB type A male to USB type C female adapter (not included in the box) if the computer does not have the USB type C port or it can not recognize the dongle.

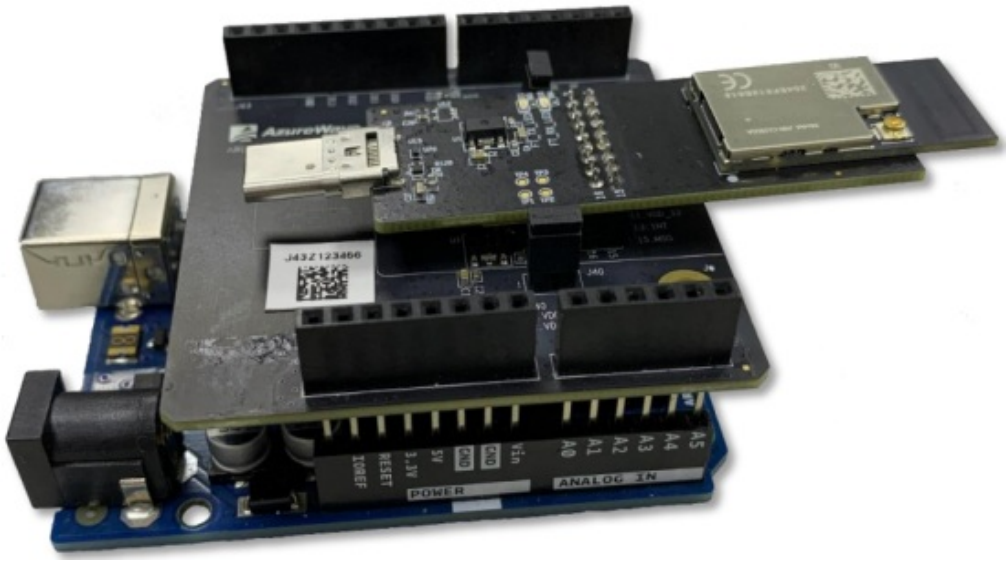
- Remove all the jumpers (2mm pitch) on the dongle before mating it with the Arduino Uno through the ARDUINO UNO ADPAPTOR. **The digital logic level & power supply input of the dongle are 3.3V** . DO NOT directly connect the dongle to the Arduino Uno.





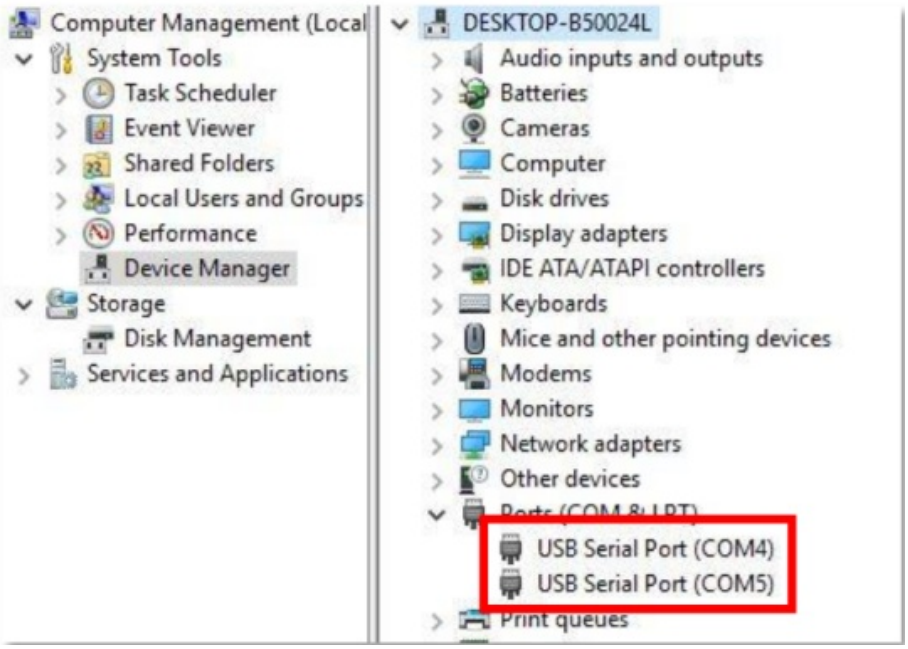
ARDUINO UNO ADPAPTOR

- AW-CU300AV3-USB connected to Arduino Uno through the ARDUINO UNO ADPAPTOR.

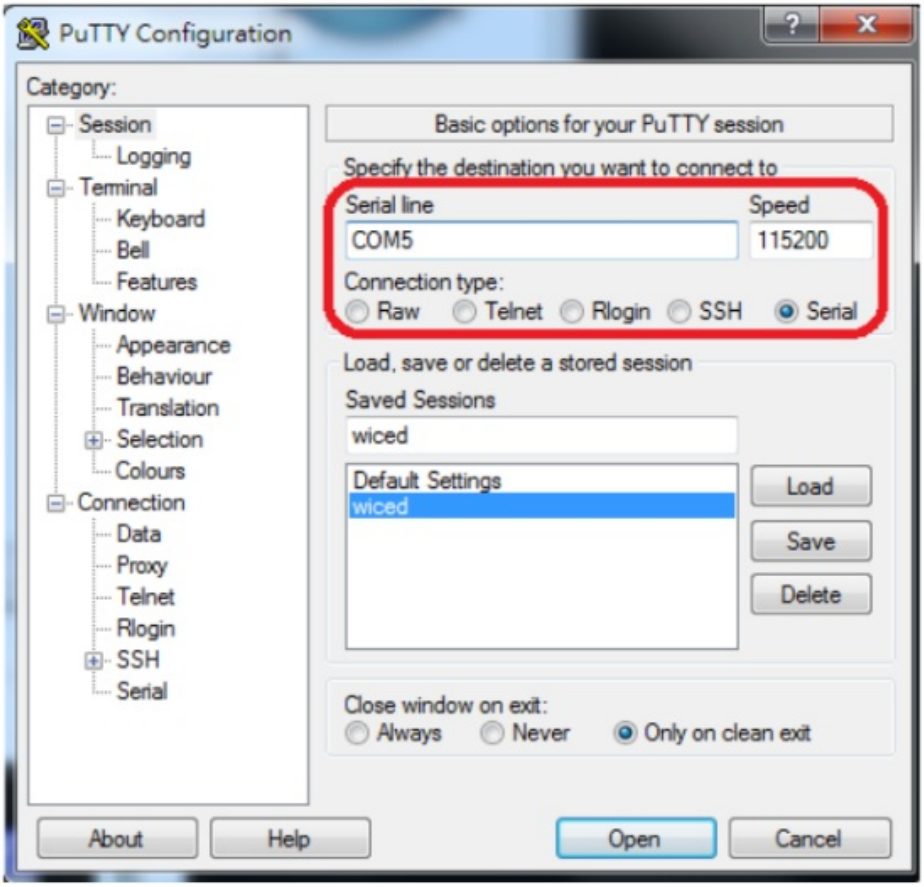


Identify the Hardware and Firmware

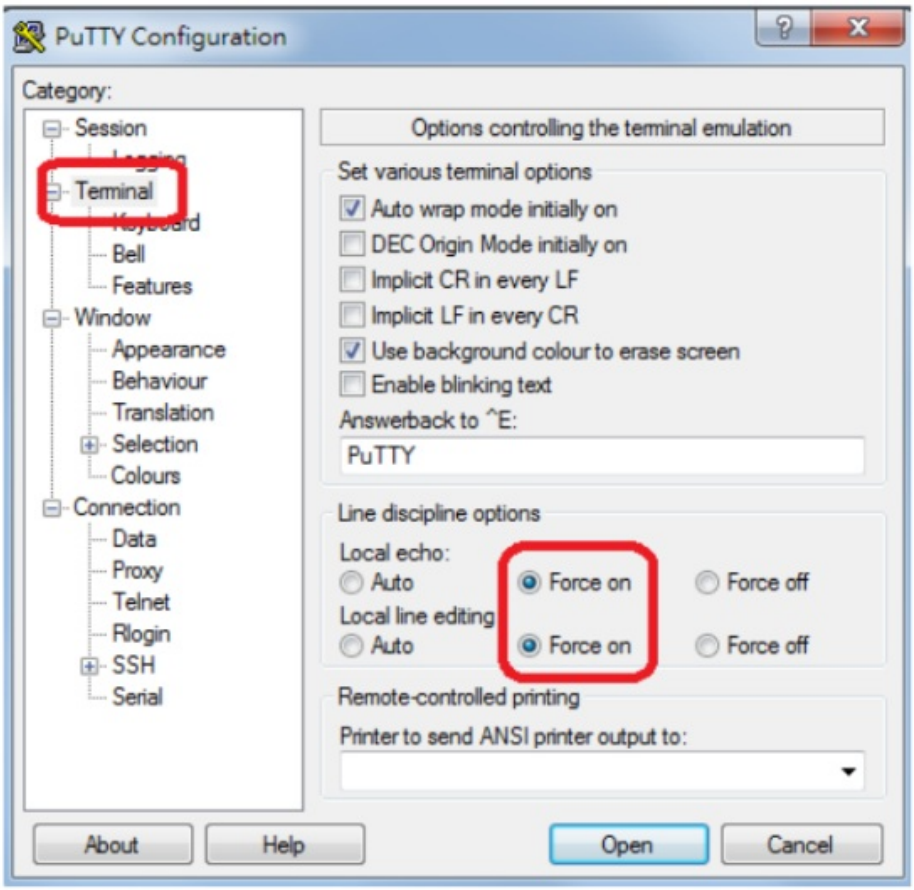
- Check if Device Manager can recognize AW-CU300AV3-USB. If not, download FTDI Driver at https://www.ftdichip.com/Drivers/VCP.htm, then send the AT commands through the second FTDI com port.



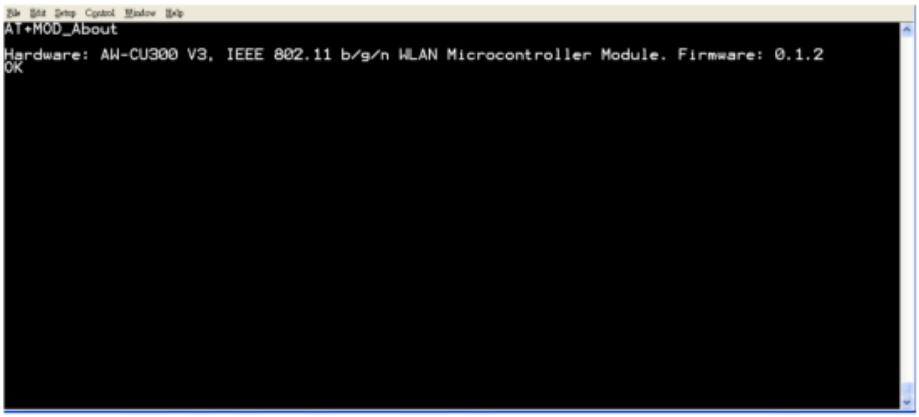
- Open a serial connection by using PuTTY or the similar terminal emulator.



- 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.



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_ Get Mode**
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_ Set AP=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, root CA certificate, client certificate, client private key.

   client ID: Thing name(Client ID).

   endpoint: AWS Iot endpoint URL.

   root CA certification: Certificate for root CA.

   client certificate: Certificate for this Thing.

   client private key: Private key for thing 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

Getting Started with AWS IoT Core

Step 0: The below link is a documents of how to setup AWS IOT console, 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 (Amazon Free RTOS Full Access, AWS IoT Full Access) 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**

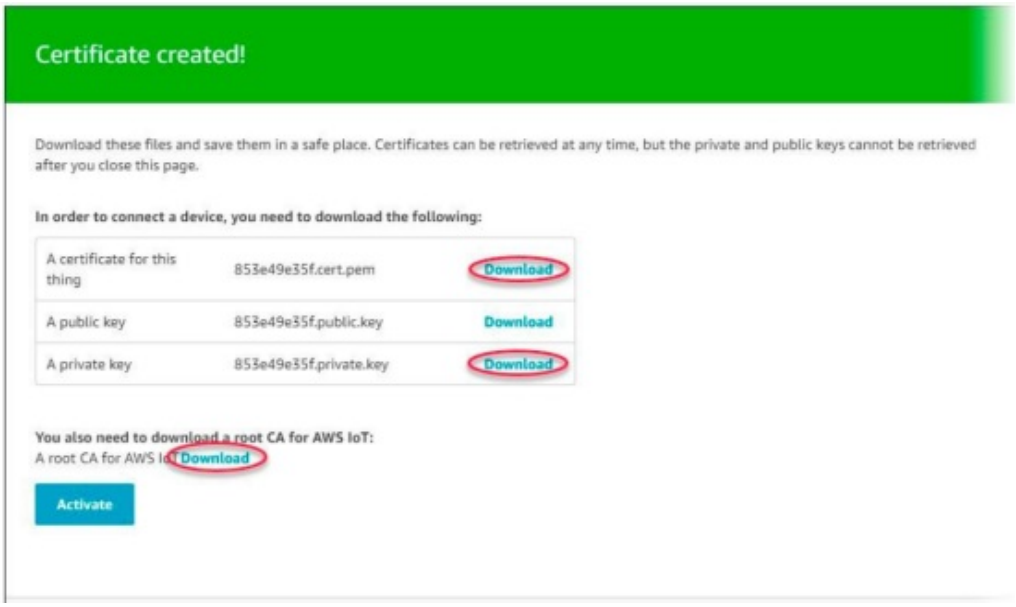
Step 3: Register a device
This step will create certificate and private key. You can use certificate, private key, thing name and endpoint as **AT+THING_ Set** command parameter. After this command executing, the four parameter will be provision to our connector. After provisioning, you can connect AWS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

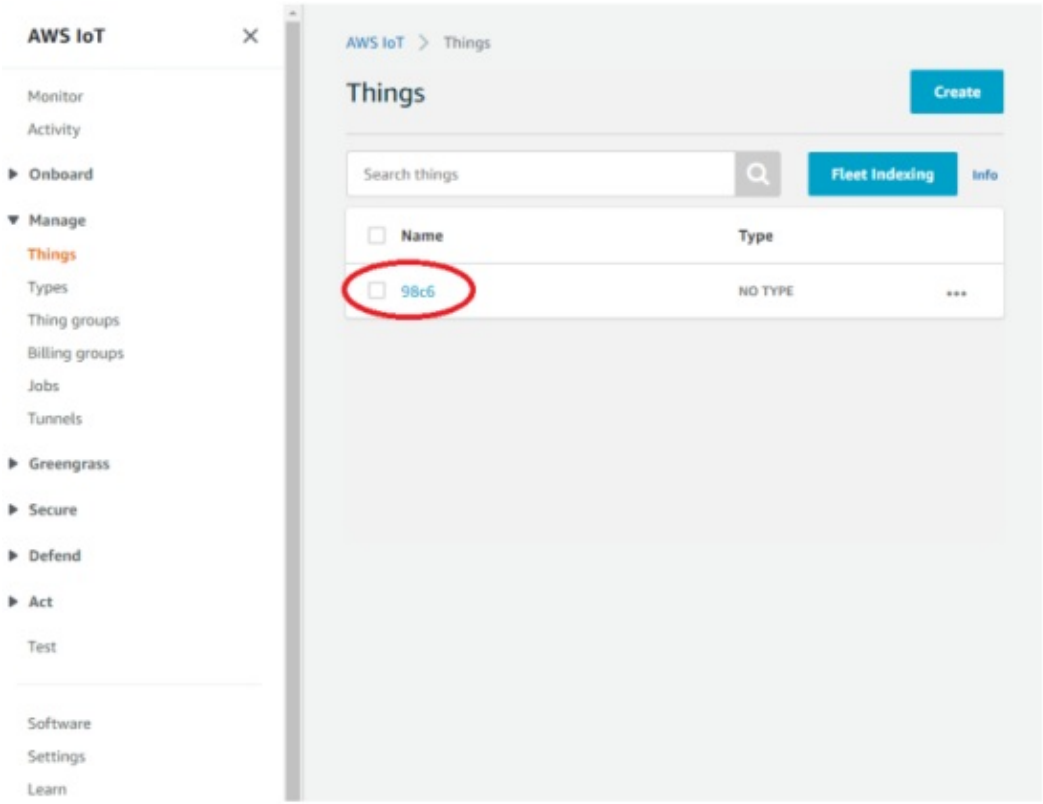
Because they will be used when send **AT+THING_ Set** command.

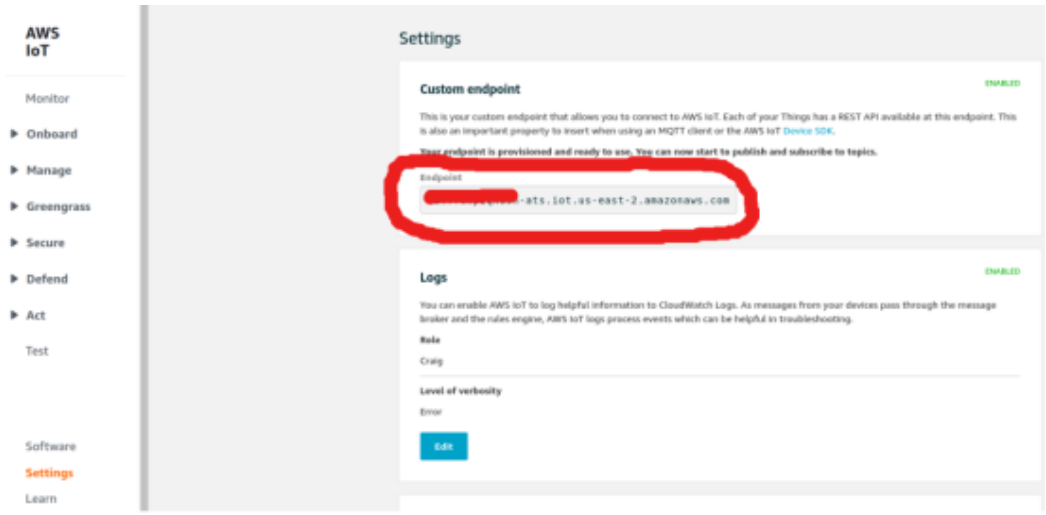


- **Thing Name and Endpoint**

  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 dashboard.





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 WPA

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, a25giitjes2mei-ats.iot.ap-northeast-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

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

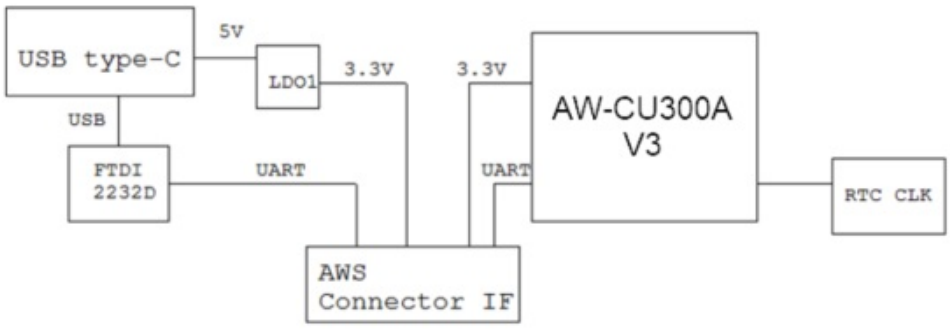
Step 9: Use the MQTT client in the AWS IoT console to monitor the messages that device sends to the AWS Cloud.

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.

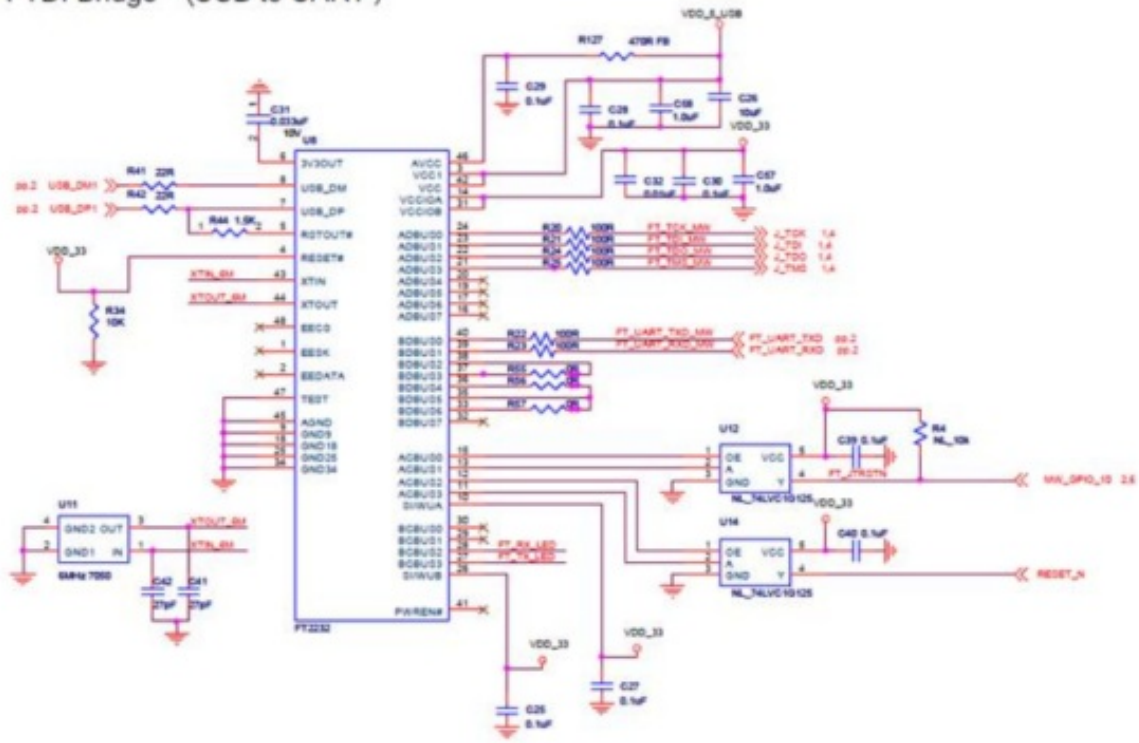
Block Diagram and Schematic

Block Diagram for AW-CU300AV3-USB

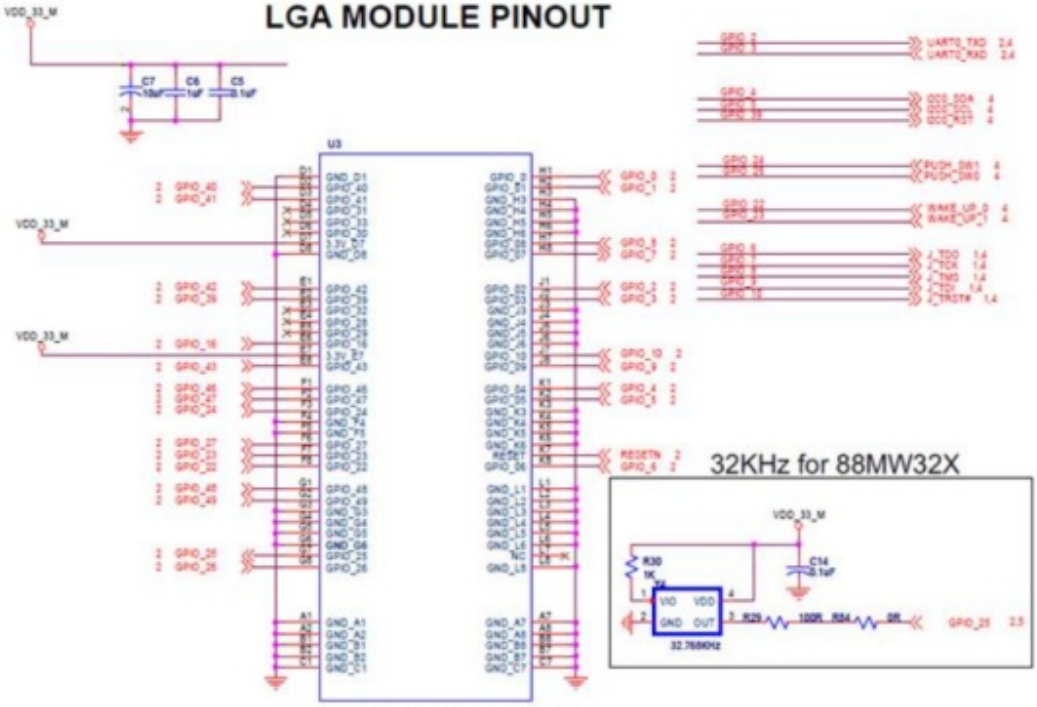


Schematics for AW-CU300AV3-USB

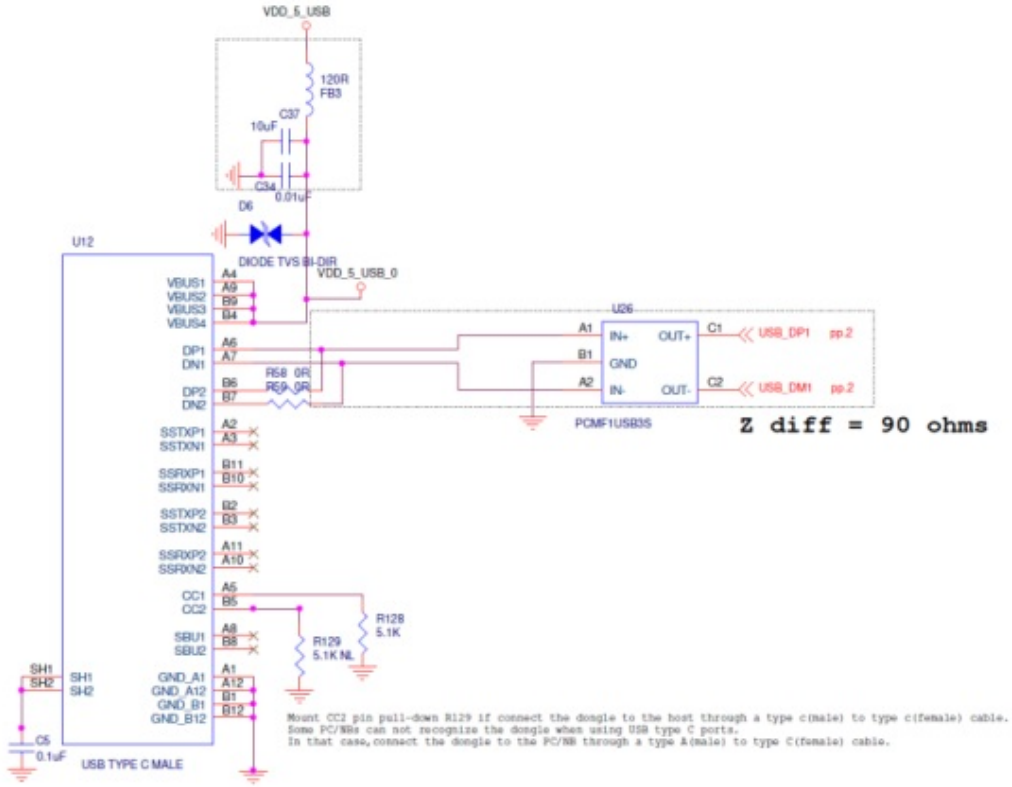
FTDI Bridge (USB to UART)

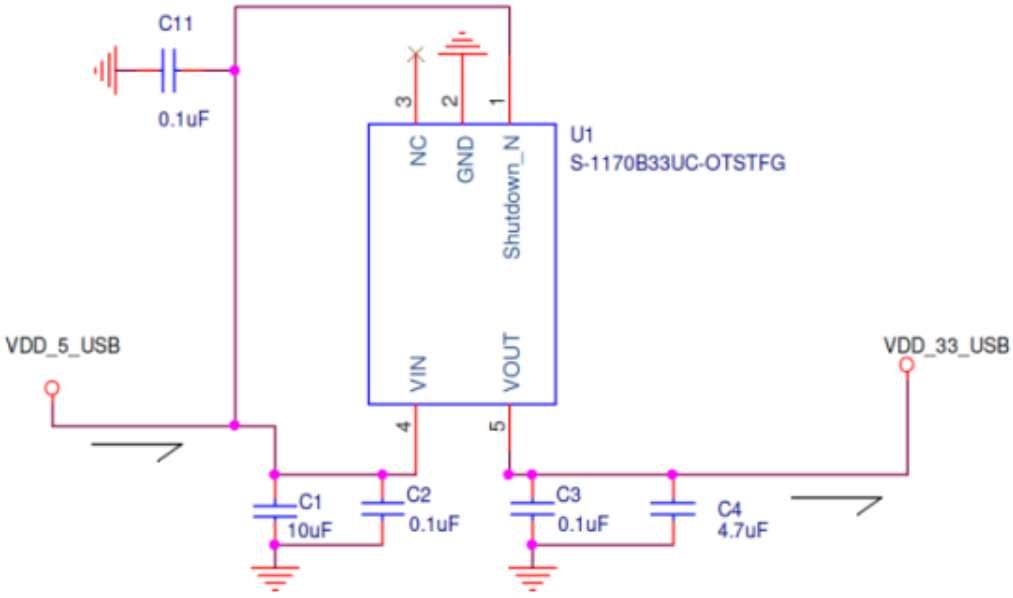


LGA MODULE PINOUT

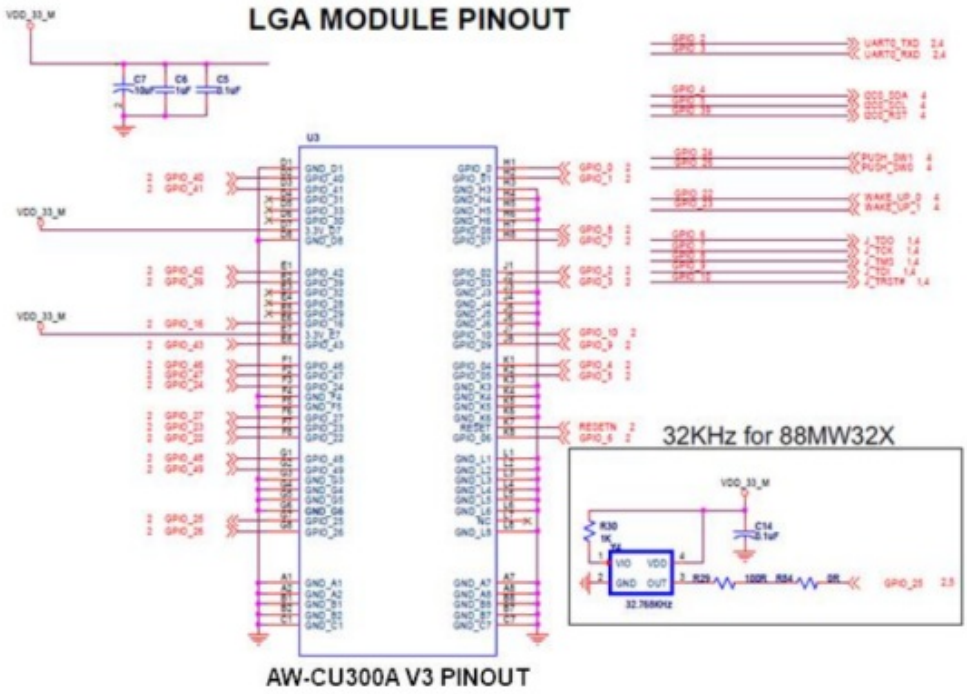


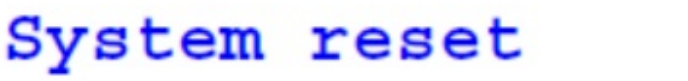
AW-CU300A V3 PINOUT

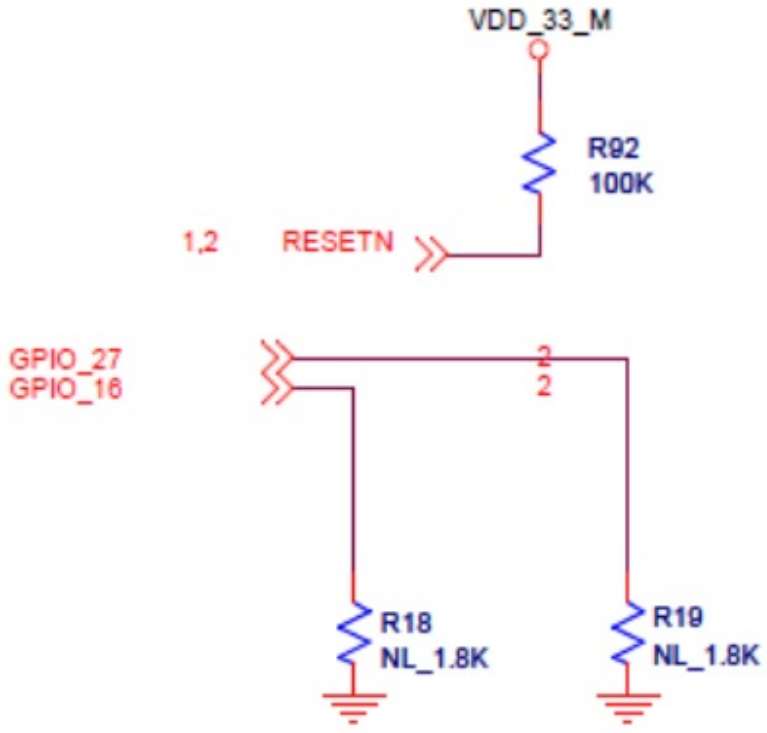

VDD_5_USB
120R
FB3
C37
10uF
C34
0.01uF
D6
DIODE TVS BI-DIR
U12
VDD_5_USB_0
VBUS1
VBUS2
VBUS3
VBUS4
DP1
DN1
DP2
DN2
R58 0R
R59 0R
U26
IN+
GND
IN-
OUT+
OUT-
C1
C2
USB_DP1
USB_DM1
pp.2
PCMF1USB3S
Z diff = 90 ohms
SSTXP1
SSTXN1
SSRXP1
SSRXN1
SSTXP2
SSTXN2
SSRXP2
SSRXN2
CC1
CC2
R128
5.1K
SBU1
SBU2
R129
5.1K NL
SH1
SH2
GND_A1
GND_A12
GND_B1
GND_B12
C5
0.1uF
USB TYPE C MALE
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.


C11
0.1uF
U1
S-1170B33UC-OTSTFG
NC
GND
Shutdown_N
VIN
VOUT
VDD_5_USB
VDD_33_USB
C1
10uF
C2
0.1uF
C3
0.1uF
C4
4.7uF

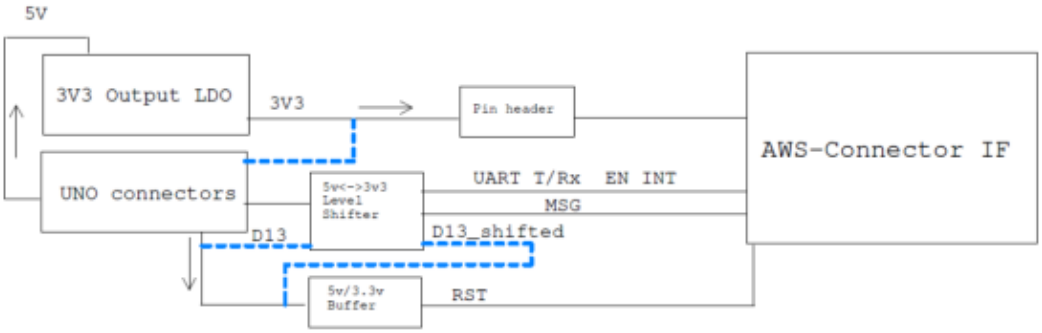
Module pinout (AW-CU300A V3)



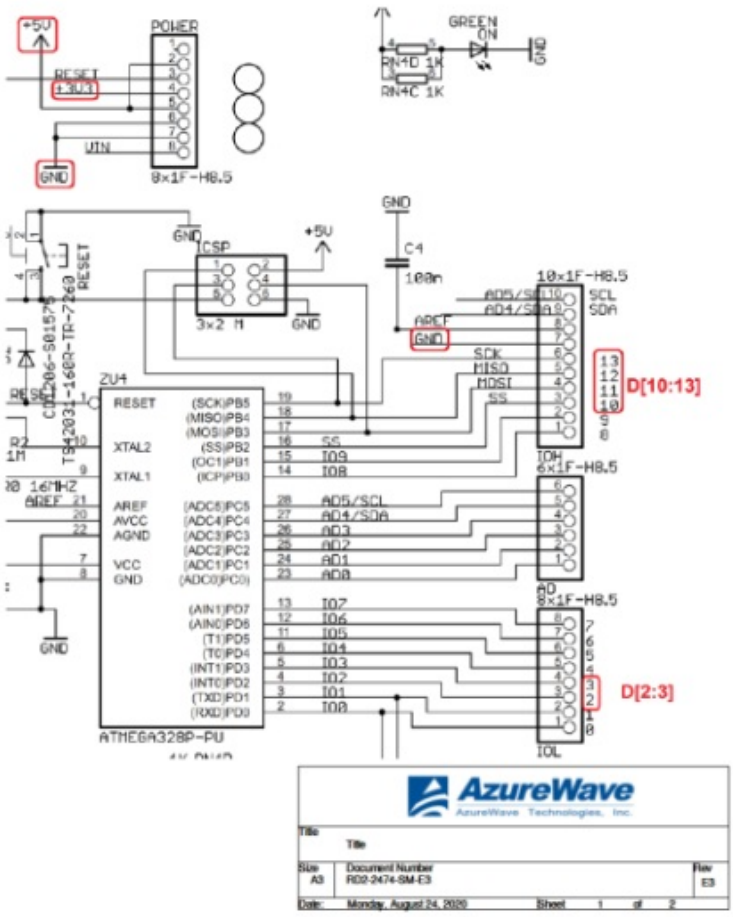




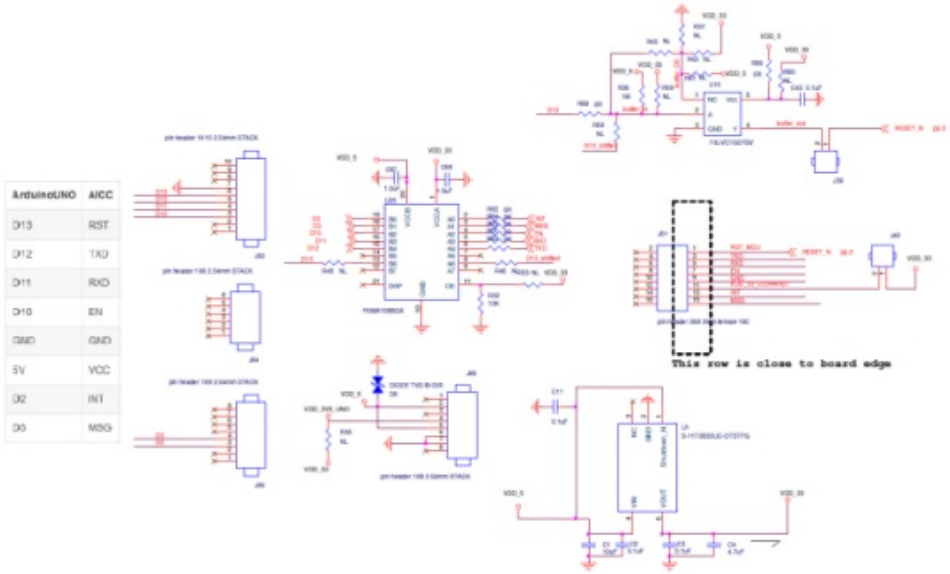
Block Diagram for Arduino Uno Adaptor



Schematics for Arduino Uno Adaptor



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



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**

Documents / Resources

AzureWave AW-CU300AV3-USB IEEE 802.11 b/g/n WLAN Microcontroller Module with Type-C Dongle Getting Started Guide Rev. 0.1	AzureWave AW-CU300AV3-USB WLAN Microcontroller Module with Type-C Dongle [pdf] User Guide AW-CU300AV3-USB, WLAN Microcontroller Module with Type-C Dongle

References

- 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