

Using MQTT on Arduino with ESP ATCMD and RTL872xD dev board

This guide demonstrates how to connect to an MQTT server on an Arduino board without WiFi capability, by using the ESP AT command set to control the RTL872xD dev board to connect to WiFi and communicate over the internet.

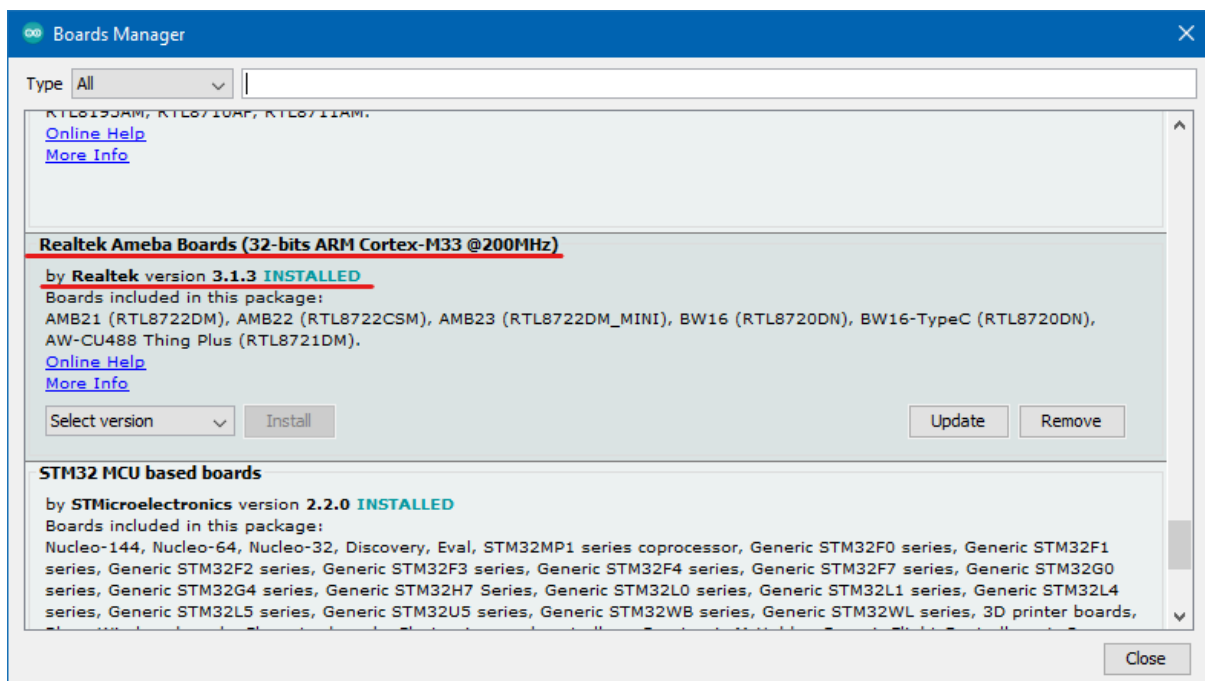
Parts required

- AmebaD [AMB23 / AMB21 / AMB22 / BW16] x 1
- Arduino Due x 1

Preparation

For the RTL872xD board, you will need to compile and upload an Arduino sketch that will accept and process the ESP AT commands.

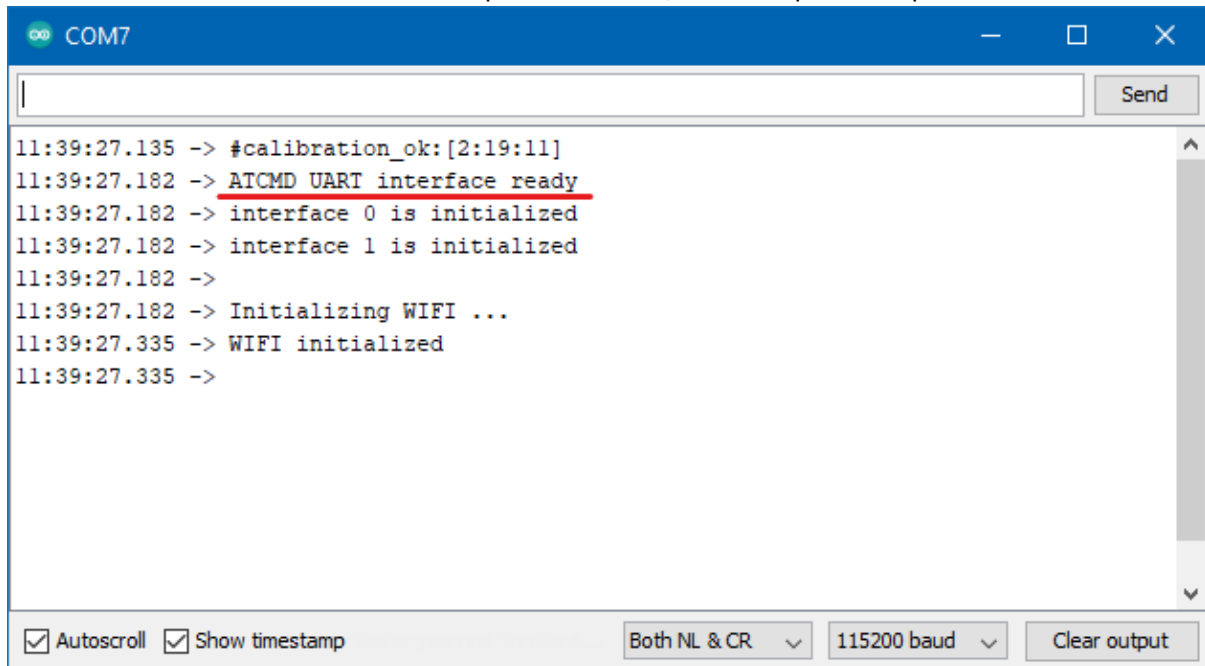
First, in Arduino -> Tools -> Board -> Boards Manager, ensure that the board support package for Realtek Ameba Boards is installed.



Download the Ameba_ATCMD_ESP sketch from

https://github.com/ambiot/ambd_arduino/tree/dev/Ameba_misc and open it in Arduino IDE.

Ensure that the correct board and COM port is selected, then compile and upload.

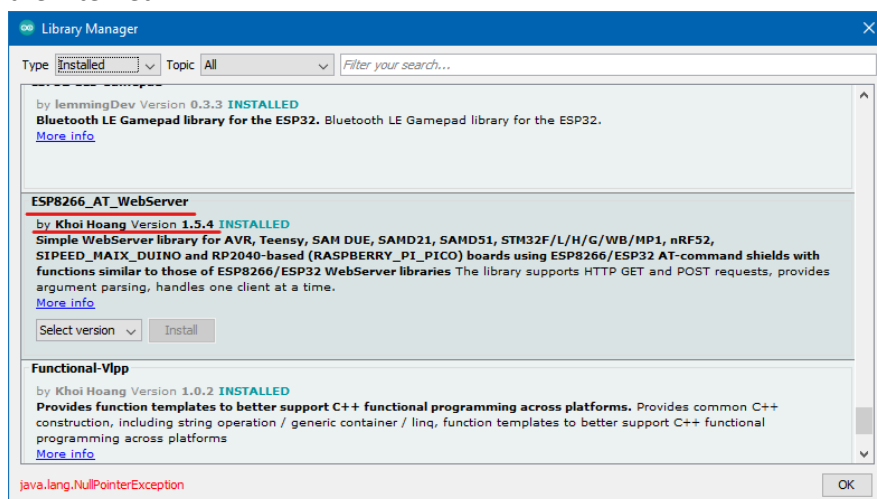


After uploading, open the serial terminal and you should see the message “ATCMD UART interface ready”. This indicates that the board is ready to accept AT commands. The default UART port uses TX pin PB_1 and RX pin PB_2, with a baud rate of 115200 bps, 8 data bits, 1 stop bit, no parity bit, no flow control.

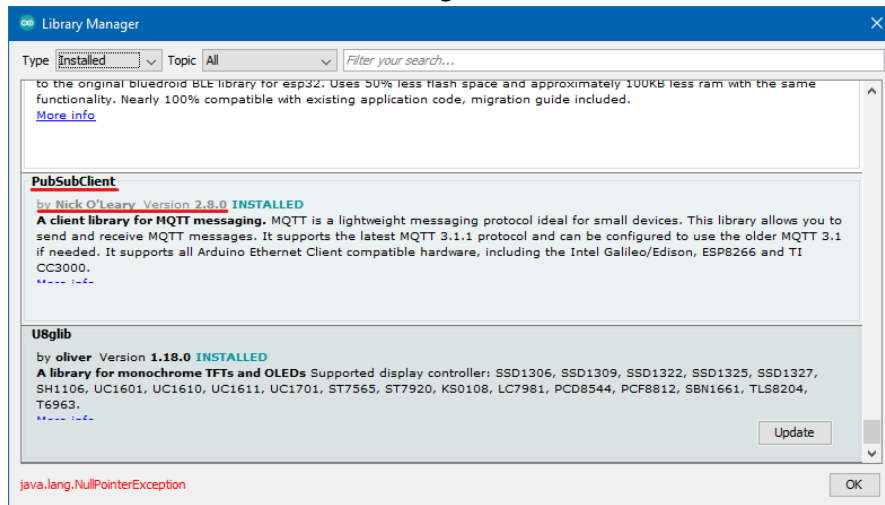
To test the ATCMD interface, you may choose to connect a USB-UART adapter to the TX and RX pins, then open a serial terminal and type in AT commands. Refer to https://espressif-docs.readthedocs-hosted.com/projects/esp-at/en/release-v2.2.0.0_esp32/AT_Command_Set/index.html for the command syntax. Note that not all ESP AT commands are supported on AmebaD boards.

For the Arduino Due board, you will need to install these two libraries from the Arduino IDE Library Manager:

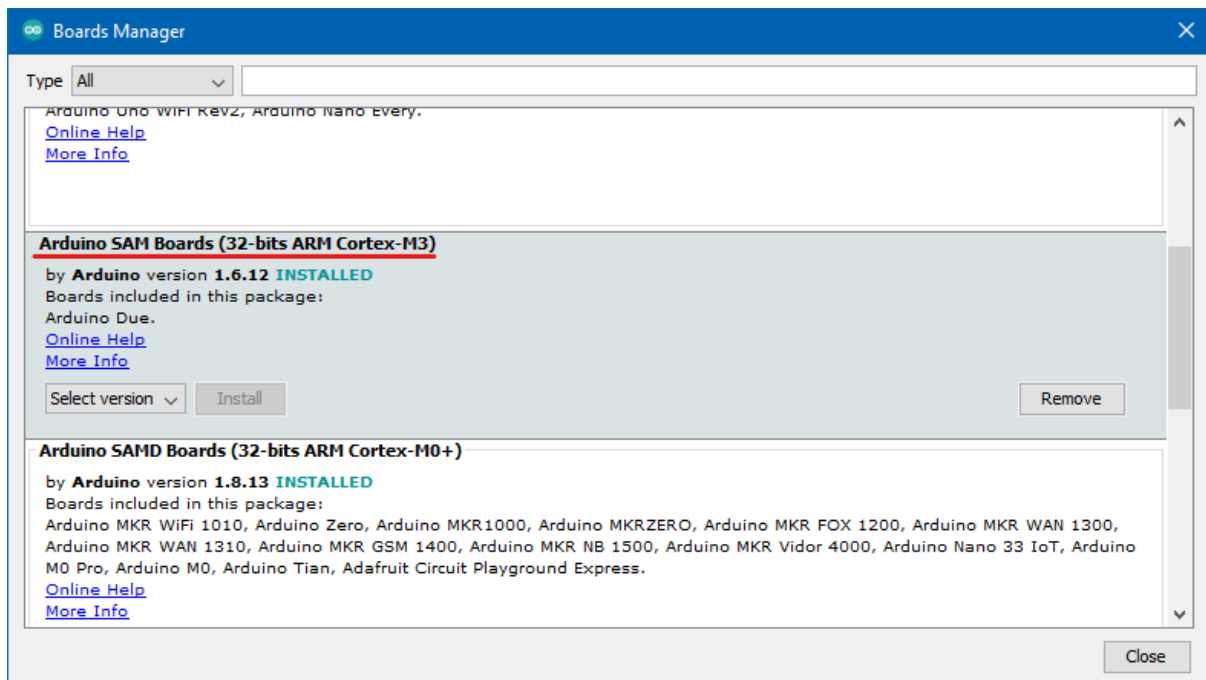
- ESP8266_AT_WebServer by Khoi Hoang
 - This provides a WiFi class and Client class that can use AT commands to connect to the internet.



- PubSubClient by Nick O'Leary
 - This connects to a MQTT server using a Client class.



in Arduino -> Tools -> Board -> Boards Manager, ensure that the board support package for Arduino Due is installed.

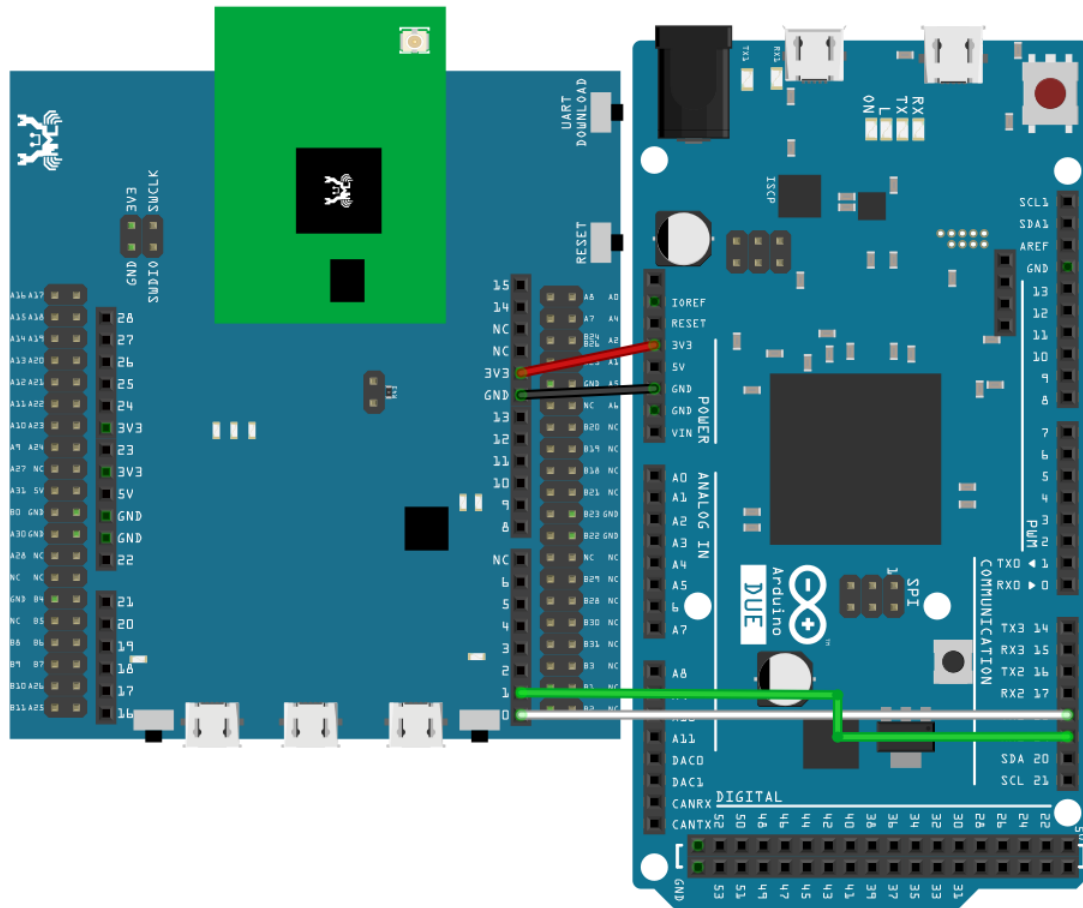


Download the Arduino_MQTT_ATCMD sketch from https://github.com/ambiot/ambd_arduino/tree/dev/Ameba_misc and open it in Arduino IDE.

In the code, modify "ssid" and "pass" to connect to your WiFi network.

Ensure that the correct board and COM port is selected, then compile and upload.

Connect the UART ports of the AmebaD board and Arduino Due board together as shown.



Reset both boards, select the COM port for Arduino Due and open the Serial Monitor, you should see the following messages if it manages to connect to the MQTT server successfully.

```

COM6
13:54:54.242 -> [ESP_AT] Using ESP32-AT Command
13:54:58.702 -> WiFi shield init done
13:54:58.702 -> Connecting to SSID: xiaomi_test
13:55:13.895 -> Attempting MQTT connection...connected
  
```

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

On your mobile phone, you can use a MQTT client app to send and receive MQTT messages.

- Android: <https://play.google.com/store/apps/details?id=in.dc297.mqttclpro>
- iOS: <https://apps.apple.com/us/app/mqtttool/id1085976398>

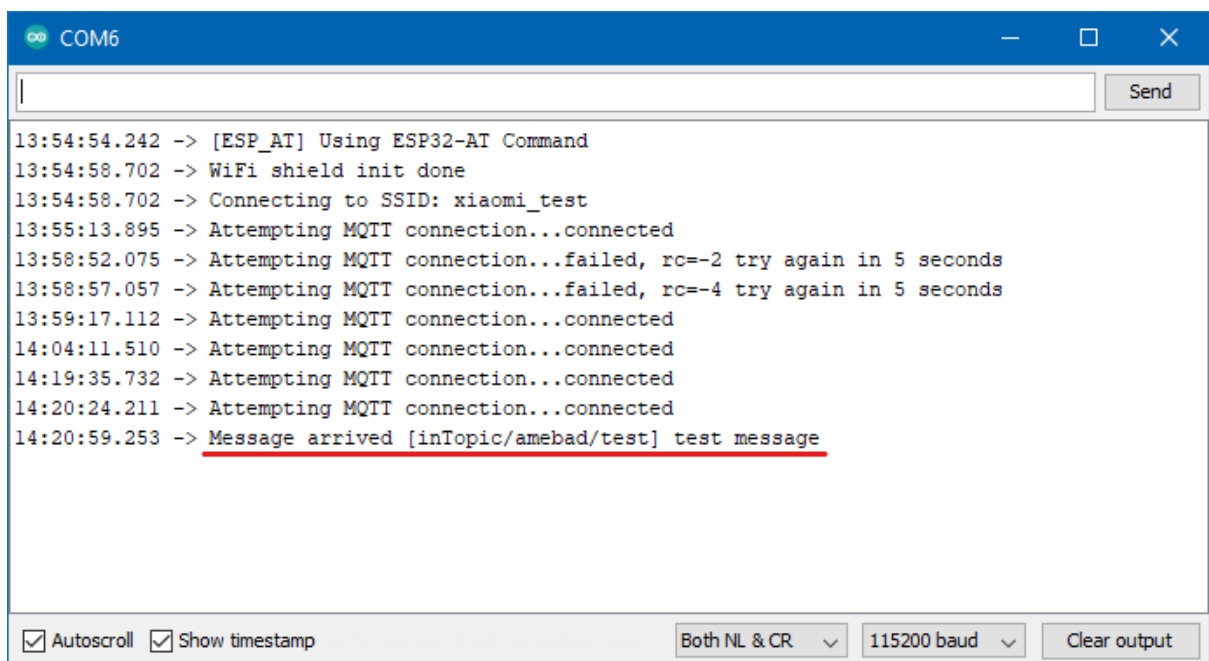
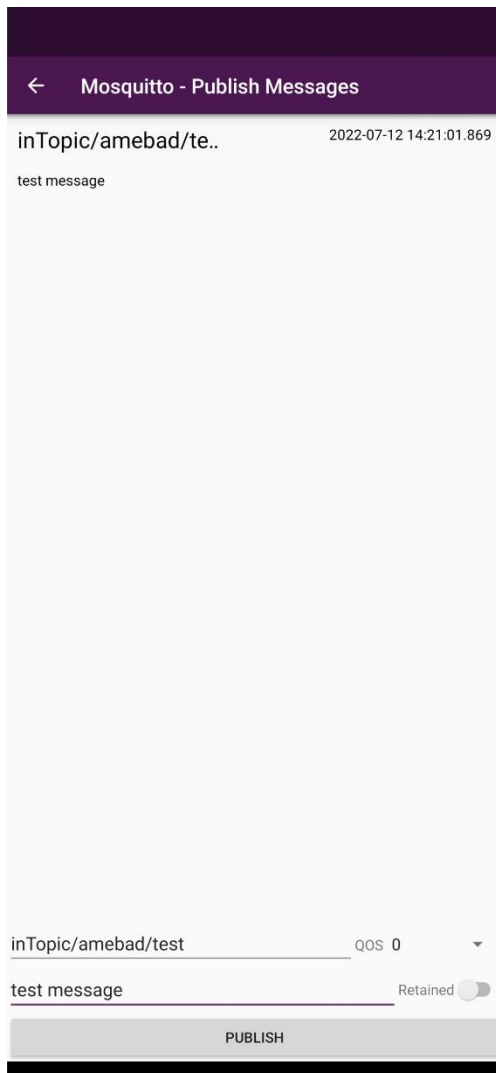
In the MQTT client app, connect to the public Mosquitto MQTT server with the following settings:

- Host: test.mosquitto.org
- Port: 1883

Subscribe to the topic “outTopic/amebad/test” and you will receive a message every time the Arduino Due reconnects to the MQTT server.



Publish a message to the topic “inTopic/amebad/test” and you will see it received and printed out in the Serial Terminal.



References

- Documentation and syntax for ESP AT commands can be found at https://espressif-docs.readthedocs-hosted.com/projects/esp-at/en/release-v2.2.0.0/esp32/AT_Command_Set/index.html