logo

**Table of Contents**

UM0167 Realtek RTL8710AF/RTL8195AM MQTT Example Guide

[1 Introduction to MQTT 3](#_Toc515397904)

[2 Example 1: Basic MQTT connect. 4](#_Toc515397905)

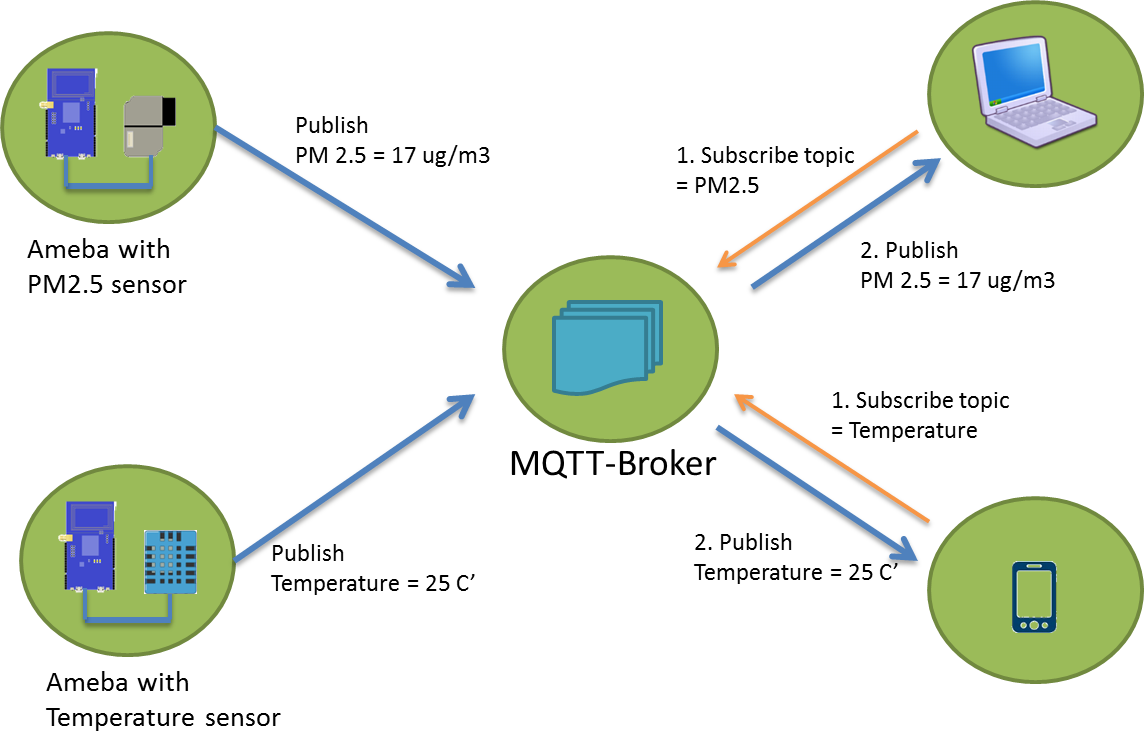
[2.1 Building and Running the Basic Example 4](#_Toc515397906)

[2.2 Example 2: DHT Sensor with MQTT 8](#_Toc515397907)

# Introduction to MQTT

MQTT (Message Queuing Telemetry Transport) is a protocol proposed by IBM and Eurotech. The introduction in [MQTT Official Website](http://mqtt.org/):  
MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport.

We can say MQTT is a protocol designed for IoT. MQTT is based on TCP/IP and transmits/receives data via publish/subscribe.



In the operation of MQTT, there are several roles:

* Publisher: Usually publishers are the devices equipped with sensors (ex. Ameba). Publishers upload the data of the sensors to MQTT-Broker, which serves as a database with MQTT service.
* Subscriber: Subscribers are referred to the devices which receive and observe messages, such as a laptop or a mobile phone.
* Topic: Topic is used to categorize the messages, for example the topic of a message can be "PM2.5" or "Temperature". Subscribers can choose messages of which topics they want to receive.

In order to demonstrate the working of MQTT we have prepared 2 examples.

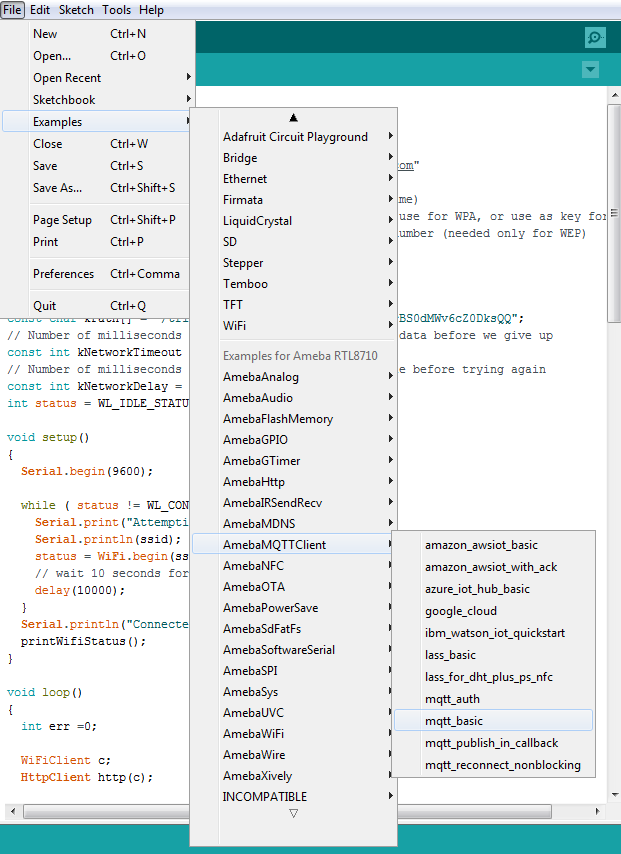
1. Simple MQTT cloud connection to subscribe and publish messages using specific topics.
2. Connect the DHT11 sensor to the board and transmitting ambient temperature and humidity values from the sensor to the MQTT broker.

# Example 1: Basic MQTT connect.

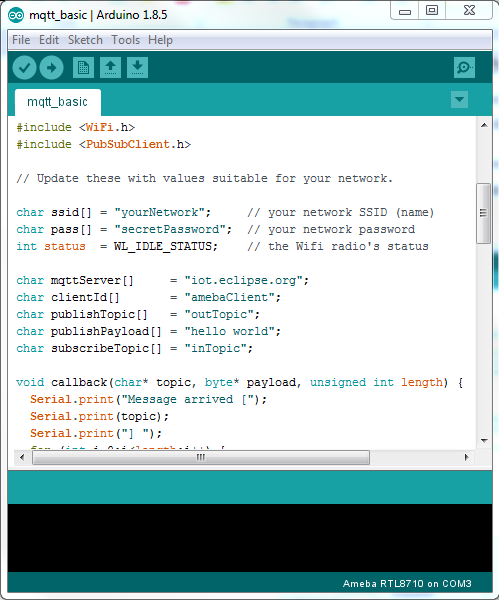
In this example we demonstrate the basic functionality of how the example can communicate to an MQTT broker and publish and subscribe to topics and send data.

## Building and Running the Basic Example

* This MQTT example can be found in the inbuilt suite of examples for the Realtek boards’ package within the Arduino IDE as shown below. We can navigate to the application specific to the Realtek board in use as shown below:

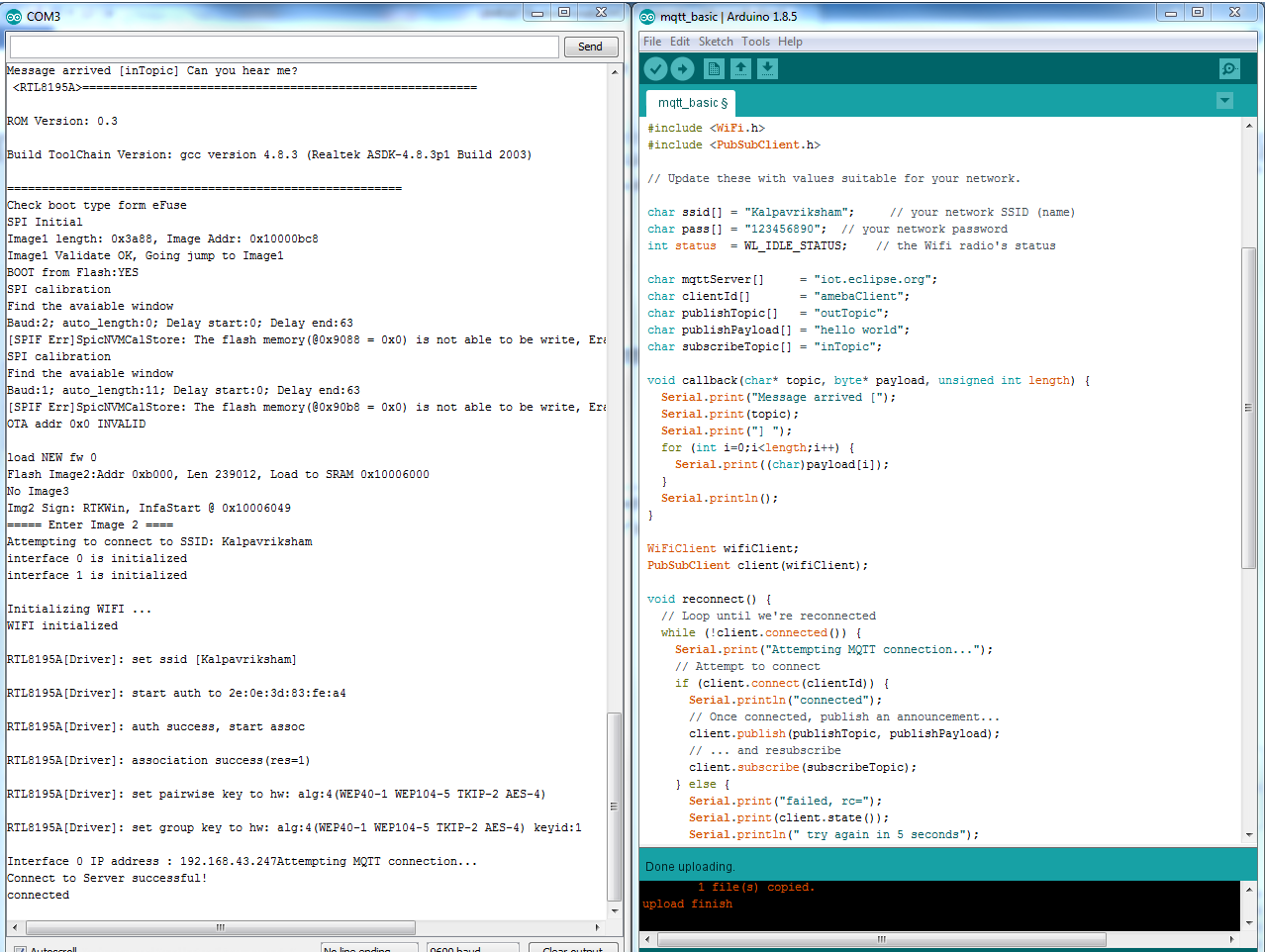


* Once the example is loaded you need to edit the Wi-Fi SSID and password in the fields provided in the example and fill in the SSID and password of the access point or the hotspot appropriately.
* It is to be noted that in the code there are certain fields for the MQTT application as shown below:
  + “mqttServer[]”: This is to specify the mqtt broker that we intend to connect to. In this case we are connecting to the free mqtt broker provided by “iot.eclipse.org”
  + “clientId[]”: This is just a unique identifier for our mqtt client which is the board. This is an optional parameter.
  + “publishTopic[]”: This is the name of the topic that is being published from the device.
  + “publishPayload[]”: This is the payload data that is sent under the publish topic to the mqtt broker. This is the data that is sent with the publish topic.
  + “subscribeTopic[]”: This is the topic to which the board subscribes to. All messages that are published to the mqtt server with this topic are routed to the device automatically if it has subscribed to that particular topic.



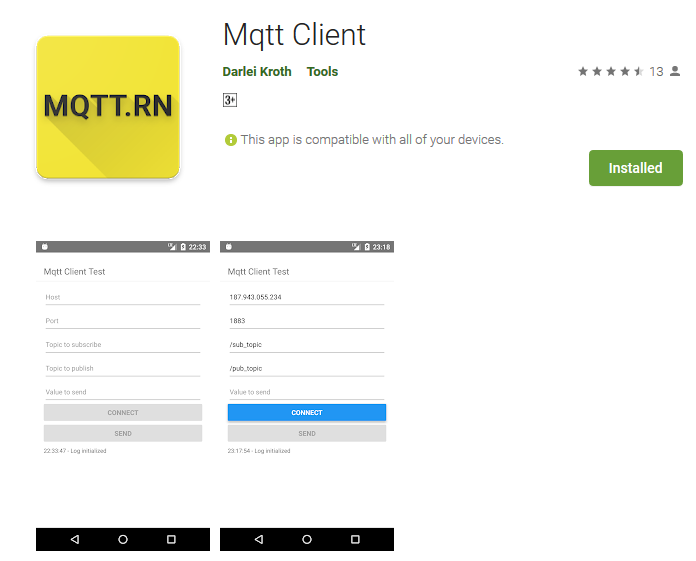
In this example we are using the standard “iot.eclipse.org” which is provided for free from eclipse for all people around the world to use. The server has both encrypted and unencrypted ports for different applications. In case you do not wish to use this server the example can be configured to use any other MQTT broker.

* Once the example is setup, it can be verified and flashed onto the device.
* Once the flashing is over, press the “Reset” button and use the serial monitor to see the output logs as shown below:

Now it is to be noted that since we are connecting to a public MQTT server there is a possibility that other people on the internet might publish messages with the same topic name as we have subscribed to and hence it is common to see stray messages appear on the console, In case you do not wish to get stray messages it is advised to use a private server.

### Testing the MQTT functionality

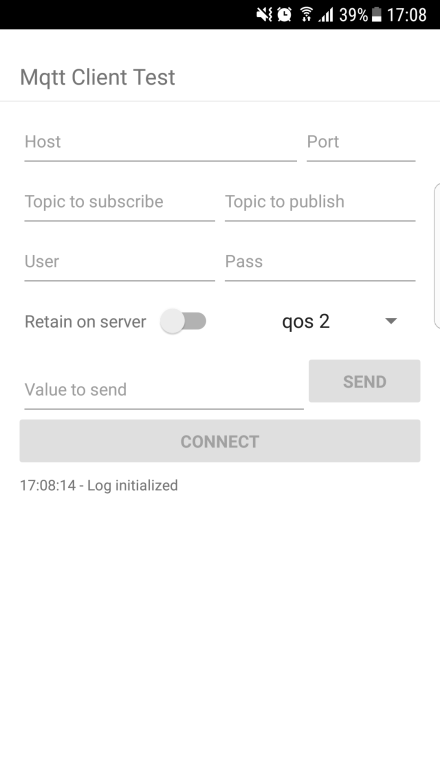
Once the MQTT example is flashed we need to test it with another MQTT client. Since MQTT is an M2M(Machine to Machine) protocol another client needs to be in place to test the subscribing and publishing of the messages. In case you have another MQTT capable device, it can be used to publish and subscribe to the topics used in this example and test. In case there is no MQTT capable device available the MQTT functionality can be tested using a mobile app, there are many such apps available for both Android as well as IOS but for simplicity we shall be using the “MqttClient” app available on the play store ash shown below:



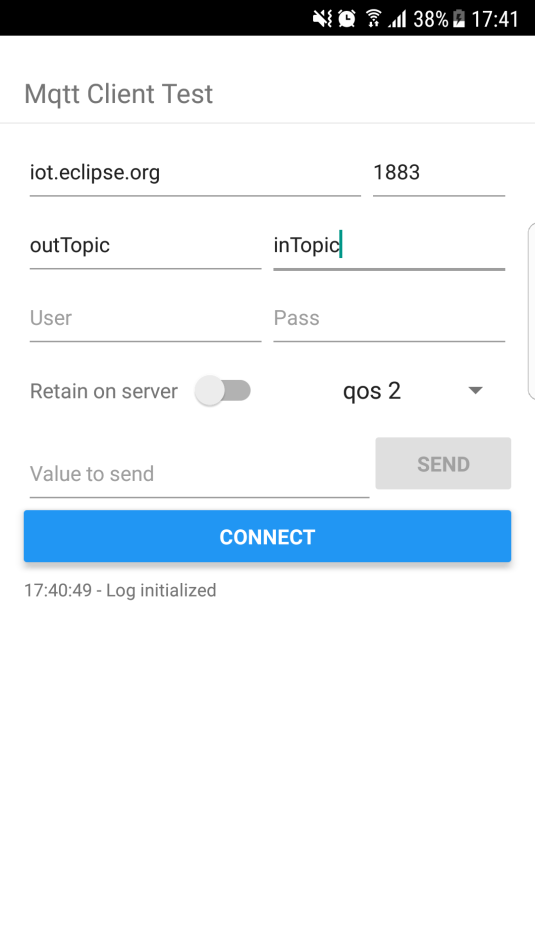
Disclaimer: It is to be noted that these mobile apps are 3rd party softwares and Realtek takes no responsibility to the availability/reliability and/or accuracy of these services. It is the user’s responsibility to find a sutiable client to test his/her application and the 3rd party softwares referred to in this document are only a reference.

In order to test the MQTT example using the MQTTClient app, the following needs to be done.

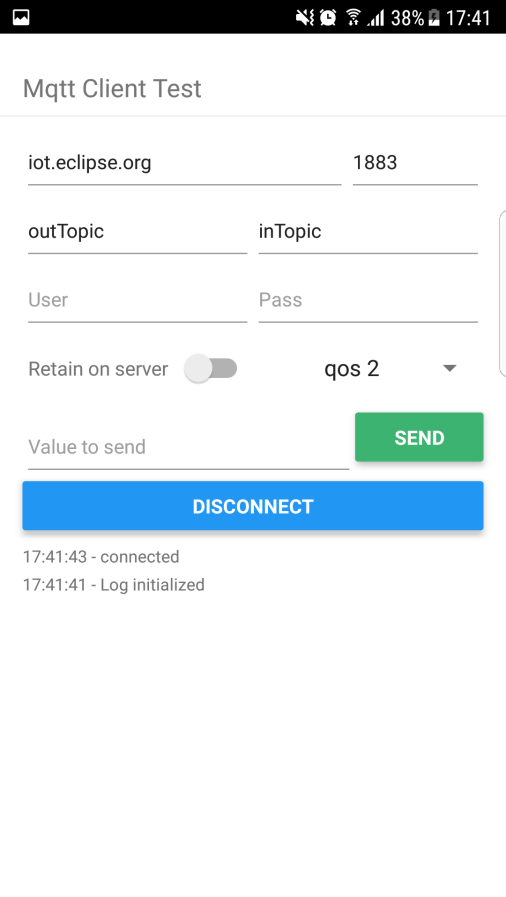
* Download and install the application from the play store in android phones.
* Once the app is downloaded, the interface looks as follows:



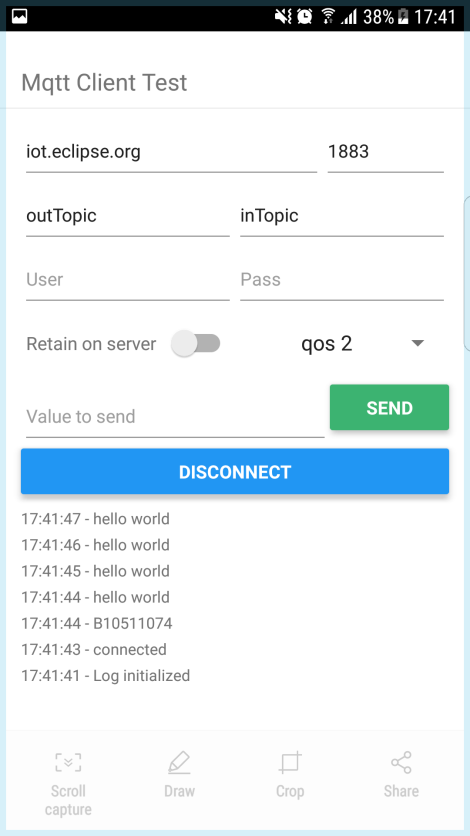
* The MQTT broker URL needs to be entered in the “Host” field and the port needs to be specified. In this case the “Topic to subscribe” needs to be the topic that is being published from the board and the “Topic to publish” needs to be the topic that the board subscribes to only then can an 2 way communication via MQTT be setup. The configuration for this example is as shown below.



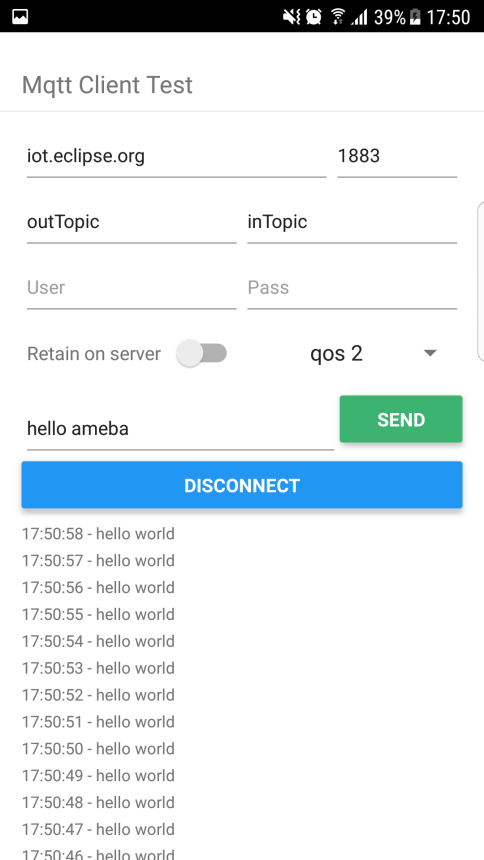
* The username and password fields can be left empty as we are using an open server to test the MQTT functionality. In case a closed server is being used, the username and password can be used.
* In case only one action either publish or subscribe is being done, the app doesn’t allow us to connect, in this scenario it is advised to just fill in a random topic for the feature that isint being used and enable the connection.
* Once all the configurations are done, press “CONNECT” to initiate a connection to the server.
* It is to be noted here that some infrastructure Wi-Fi connections and some Access Points deliberately block MQTT servers, it is essential to use an internet connection that allows you to connect to MQTT servers in order to test using these types of applications. It is advised to use mobile data in order to perform these tests.
* Once all configurations are set and the app connects to the MQTT server, the connect icon changes to “DISCONNECT” as shown below.



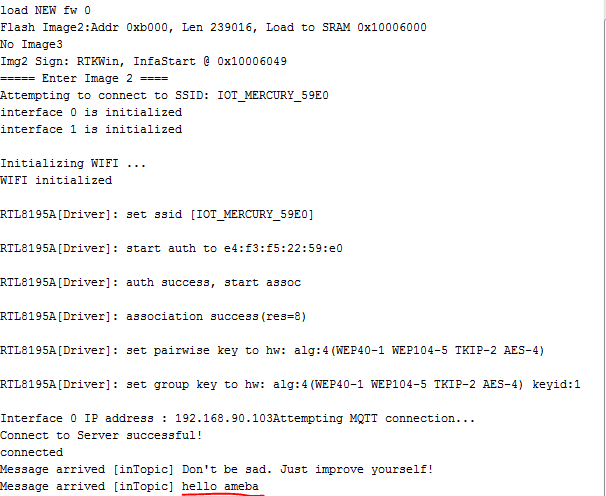
* As seen in the code in the example, the publish payload from the board is “Hello World” and hence when the board publishes the topic we are able to see “Hello World” printed on the console as shown below.



* In order to send any data to the board we can enter the text we wish to send in the “Value to send” field and press the send button as shown below.



* Once you hit send, the text you type is sent as the payload along with the topic that specified and since the same topic is being subscribed by the board it will be received and printed on the serial monitor as shown.

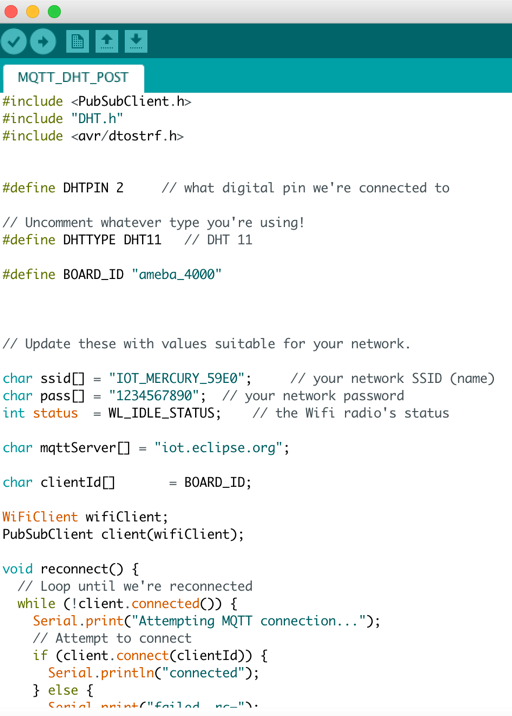


## Example 2: DHT Sensor with MQTT

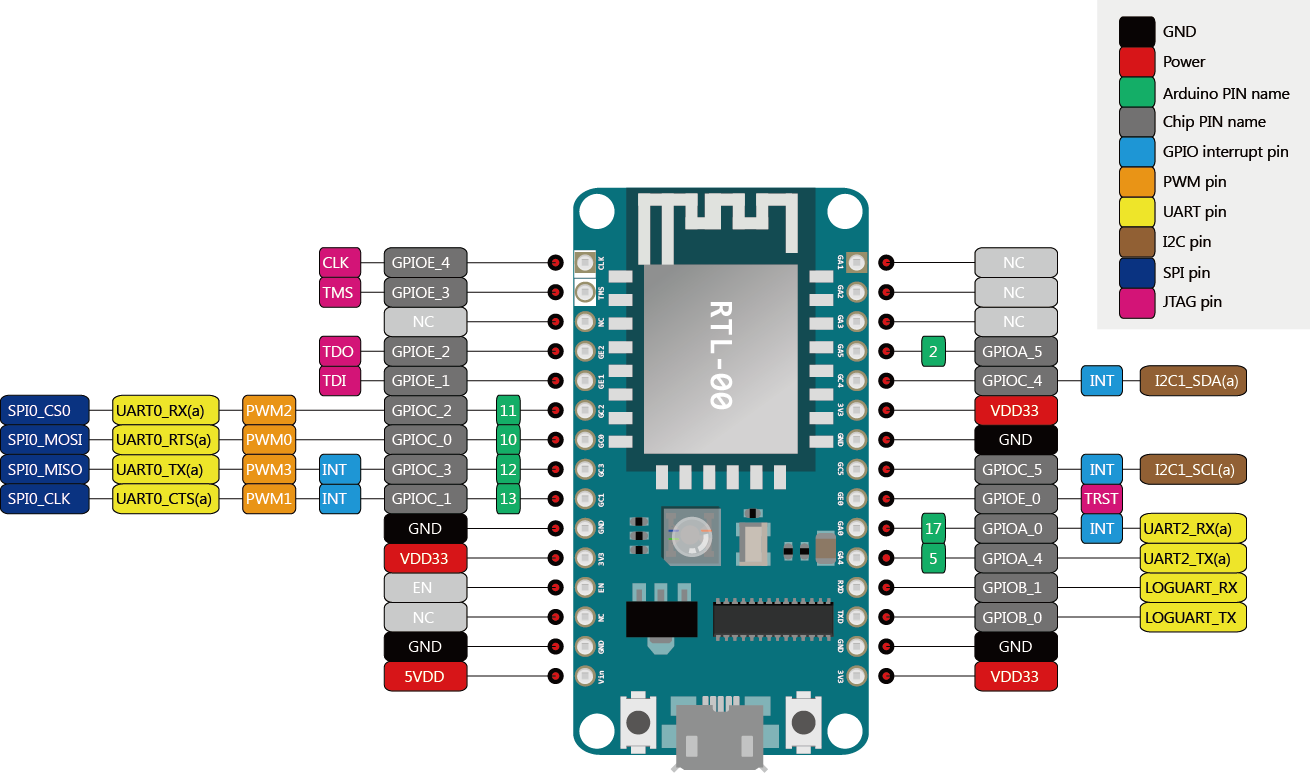
This example details how to connect the DHT sensor to the Realtek board and read the temperature and humidity data and transmit the results to an MQTT broker.

The code to run this particular example is placed inside the folder “MQTT\_DHT\_POST”. Load the example on the Arduino IDE.

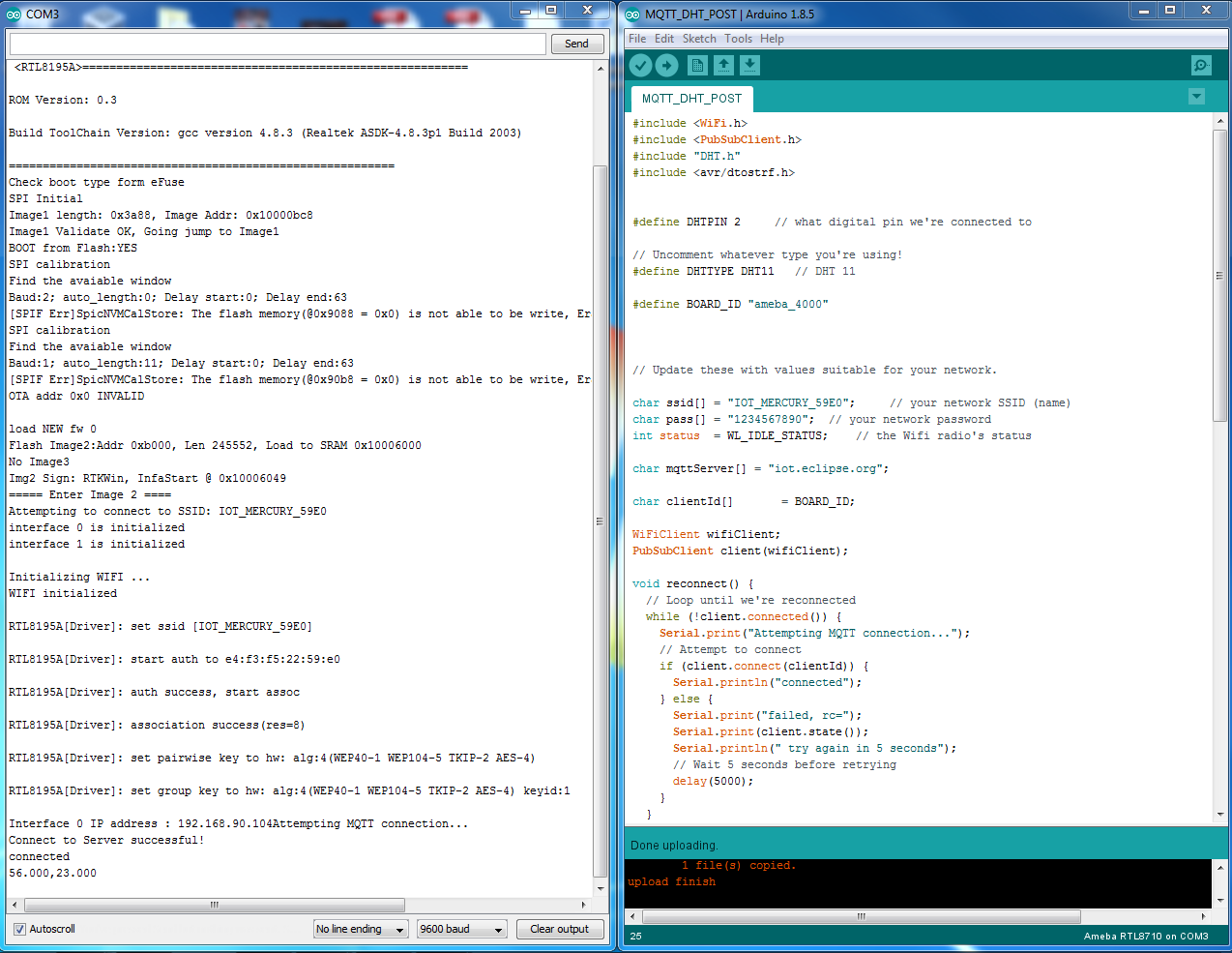
* Open the Example and edit the SSID and password to the Wi-Fi connection that you are connecting to.



* Once this is done connect the DHT sensor to the physical pins on the board.
* It is to be noted that the 3 pins of the DHT sensor should be connected to the Realtek board as detailed below:
  + VCC🡪3V3
  + GND🡪GND
  + DAT🡪GA5
* It is to be noted that inside the code we have defined the DHTPIN as 2 this is because that in the actual pinout of the RTL8710AF board the GPIOA\_5 is mapped to the Arduino pin 2 as shown in the pinout diagram below.



* In this example we are using the DHT 11 sensor and hence the DHTYPE define is set to “DHT11”
* Once the board resets it will connect to the Wi-Fi and start sending the temperature and humidity values over MQTT to the broker at an interval of 10 seconds. It can be observed in the serial monitor as shown below.



* In order to reduce the amount of data being sent over MQTT the Humdity in Percentage(%) and the Temperature in Celsius(degC) are transmitted as comma separated values. The data can be formatted as per requirement in different applications within the code.
* In order to view the MQTT results the MqttClient software can be used as specified in 2.1.1.
* It is to be noted in this example that the topic to be subscribed to is the clientID since the data is being published with the clientID as the publish topic in this case the clientID is mapped to the “BOARD\_ID” which is the string “ameba\_4000”.