logo

**Table of Contents**

UM0167 Realtek RTL8710AF/RTL8195AM MQTT Example Guide

[1 Introduction to MQTT 3](#_Toc515290988)

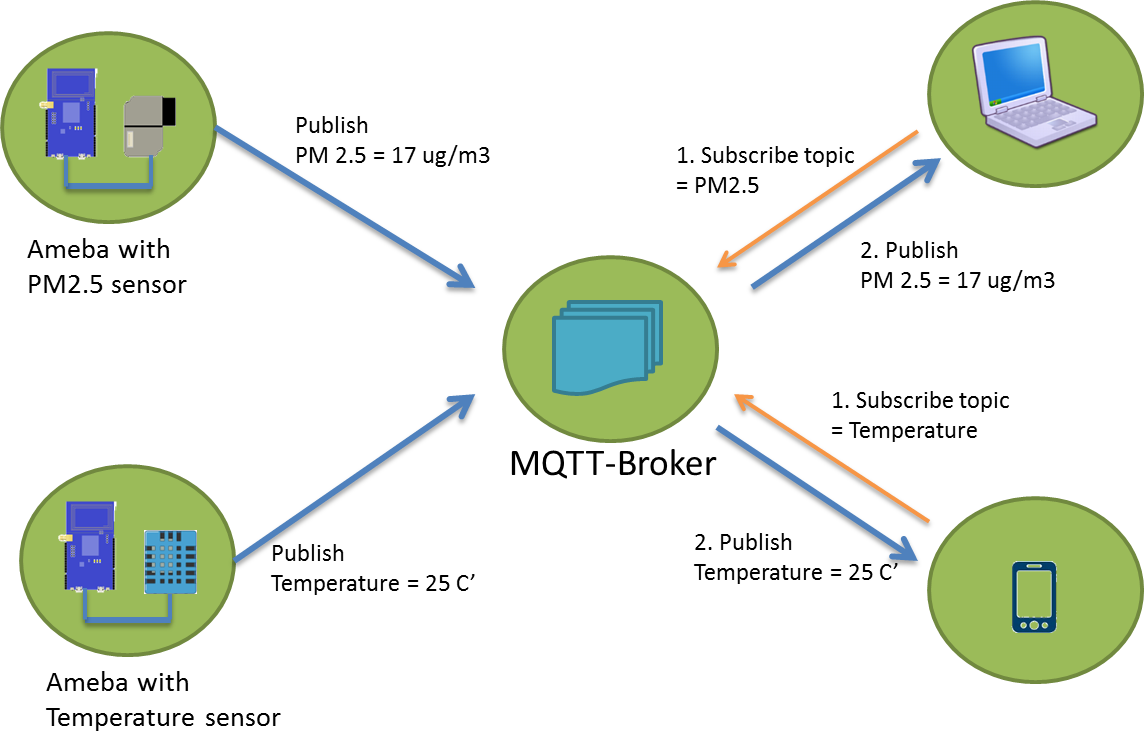
[2 Example 1: Basic MQTT connect. 4](#_Toc515290989)

[2.1 Building and Running the Basic Example 4](#_Toc515290990)

# 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.

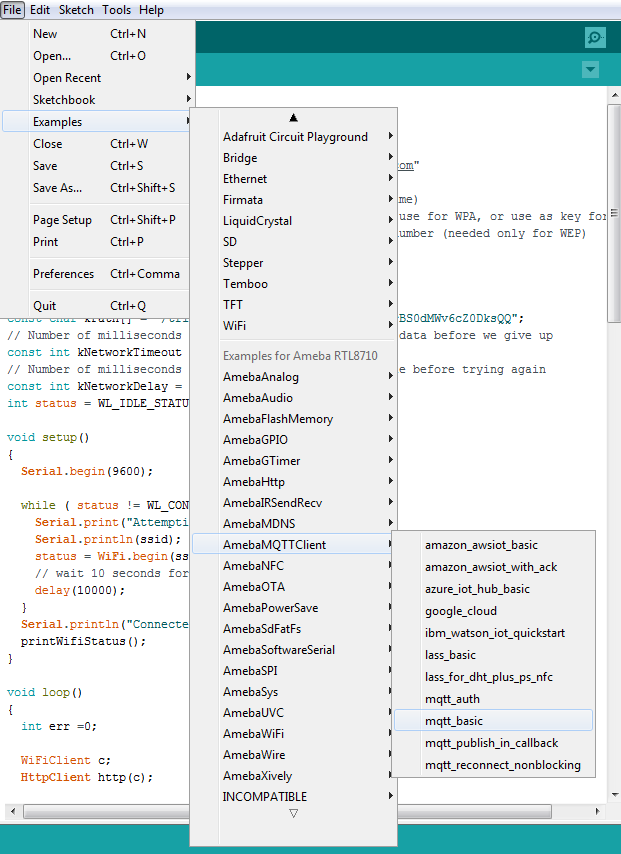
1. Simple MQTT cloud connection to subscribe and publish messages using specific topics.
2. Connect the DHT11 sensor to the board and transmitting ambient temperature and humidity values from the sensor to the MQTT broker.

# Example 1: Basic MQTT connect.

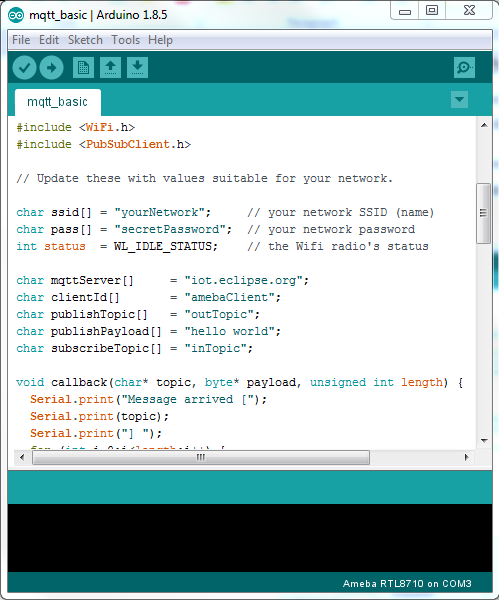
In this example we demonstrate the basic functionality of how the example can communicate to an MQTT broker and publish and subscribe to topics and send data.

## 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:

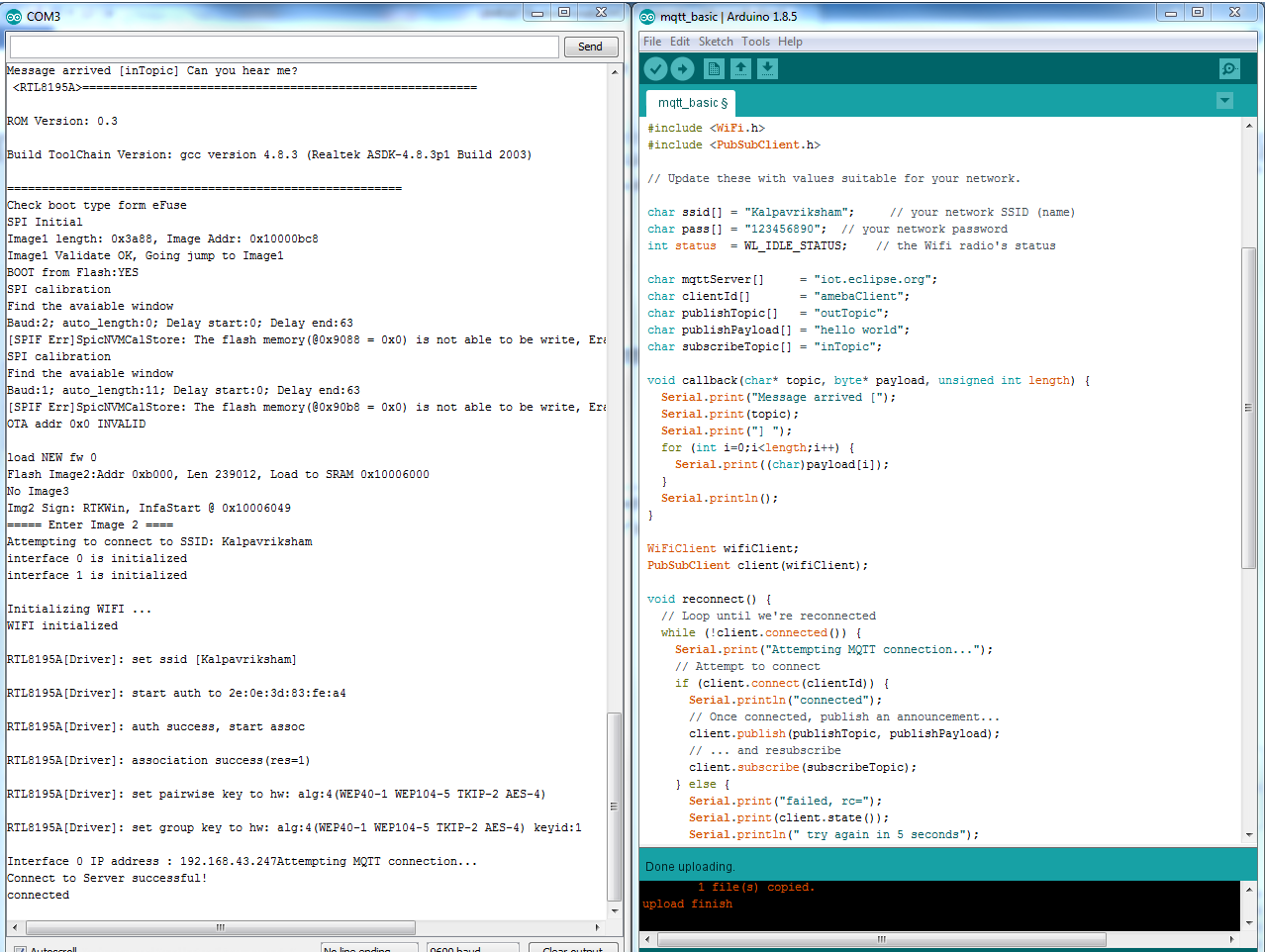


* Once the example is loaded you need to edit the Wi-Fi SSID and password in the fields provided in the example and fill in the SSID and password of the access point or the hotspot appropriately.
* It is to be noted that in the code there are certain fields for the MQTT application as shown below:
  + “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.

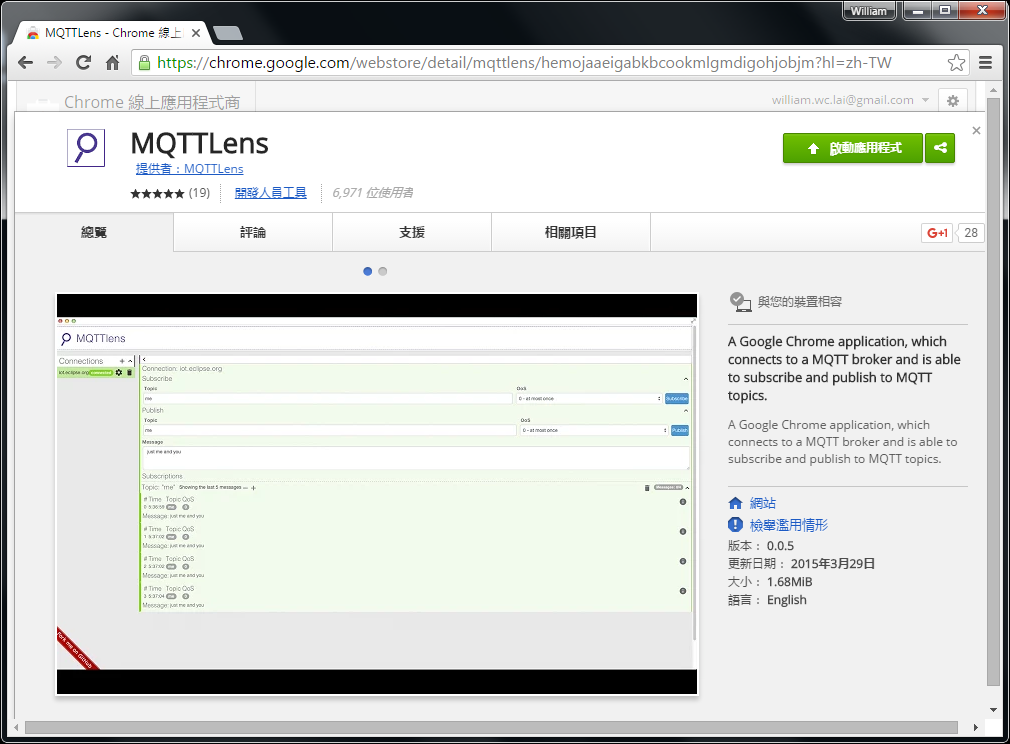


In this example we are using the standard “iot.eclipse.org” which is provided for free from eclipse for all people around the world to use. The server has both encrypted and unencrypted ports for different applications. In case you do not wish to use this server the example can be configured to use any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

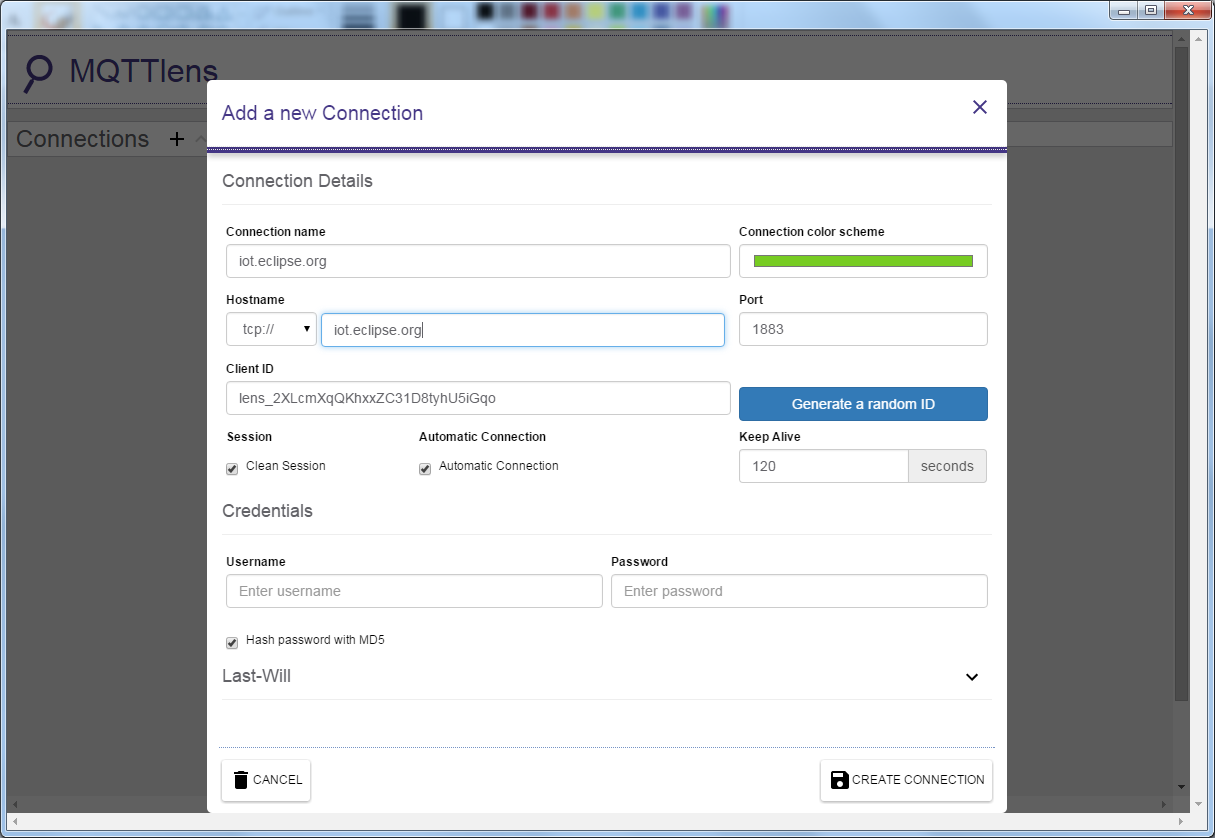
* Once the example is setup, it can be verified and flashed onto the device.
* Once the flashing is over, press the “Reset” button and use the serial monitor to see the output logs as shown below:

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.

* Once the MQTT server is connected to we need to use a 3rd party application to test the working of the same.
* In order to test this example we use the browser plugin for chrome called “MQTTLens” to demonstrate the subscribing and publishing of data as shown in the steps below.



* Install and open the MQTTLens, click "+" next to "Connection" on the left, and fill in the required information
  + Connection Name: Used to identify the connection, you can choose a name you like.
  + Hostname: The MQTT-Broker server, here we use "iot.eclipse.org"
  + Client ID: We use the default randomly generated ID
  + Then click "CREATE CONNECTION".



* Since we have not registerd the topic we want to listen to, we would not receive any messages now.
* Fill in "outTopic" in the "Topic" field, and click "Subscribe". Once done, we will receive a "hi" message from "iot.eclipse.org".
* Wait for Ameba to send next message (or you can press the reset button). Then you can see the "hello world" shows up.

