MQTT Guide
Prof. Ruth Rubio <erubio@utec.edu.pe>
IOT UTEC 2022-2

1. Download and install the Mosquitto MQTT broker:

https://mosquitto.org/download/

Windows: Use the installer.
MacOS: brew install mosquito
Linux: build and compile from repository: https://github.com/eclipse/mosquitto

2. Run the mosquitto broker

```
$ mosquitto -v
```

```
1664490965: mosquitto version 2.0.15 starting
1664490965: Using default config.
1664490965: Starting in local only mode. Connections will only be possible from clients running on this machine.
1664490965: Create a configuration file which defines a listener to allow remote access.
1664490965: For more details see https://mosquitto.org/documentation/authentication-methods/
1664490965: Opening ipv4 listen socket on port 1883.
1664490965: Opening ipv6 listen socket on port 1883.
1664490965: mosquitto version 2.0.15 running
^C1664490969: mosquitto version 2.0.15 terminating
```

End with CTRL+C (SIGINT)

3. Edit the config file:

MacOS: `$ vim /usr/local/etc/mosquitto/mosquitto.conf`
Win: C:\installation_path\mosquitto\mosquitto.conf

```
# Config file for mosquitto
#
# See mosquitto.conf(5) for more information.
#
# Default values are shown, uncomment to change.
#
# Use the # character to indicate a comment, but only if it is the
# very first character on the line.

# =================================================================
# General configuration
# =================================================================

# Use per listener security settings.
#
# It is recommended this option be set before any other options.
#
# If this option is set to true, then all authentication and access control
# options are controlled on a per listener basis. The following options are
# affected:
#
# acl_file
```

Add the following two lines uncommented (can be placed at the beginning of the file):

```
listener 1883
allow_anonymous true
```

These lines allow communication from outside the broker (publishers).

4. Execute the mosquitto broker using the config file.
   $ mosquitto -c mosquitto.conf -v

```
jorge@sican:~ $ /usr/local/opt/mosquitto/sbin/mosquitto -v -c /usr/local/etc/mosquitto/mosquitto.conf &
[1] 3666
1664493113: mosquitto version 2.0.15 starting
1664493113: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
1664493113: Opening ipv6 listen socket on port 1883.
1664493113: Opening ipv4 listen socket on port 1883.
1664493113: mosquitto version 2.0.15 running
```

5. Print received messages within a specific topic (subscription):

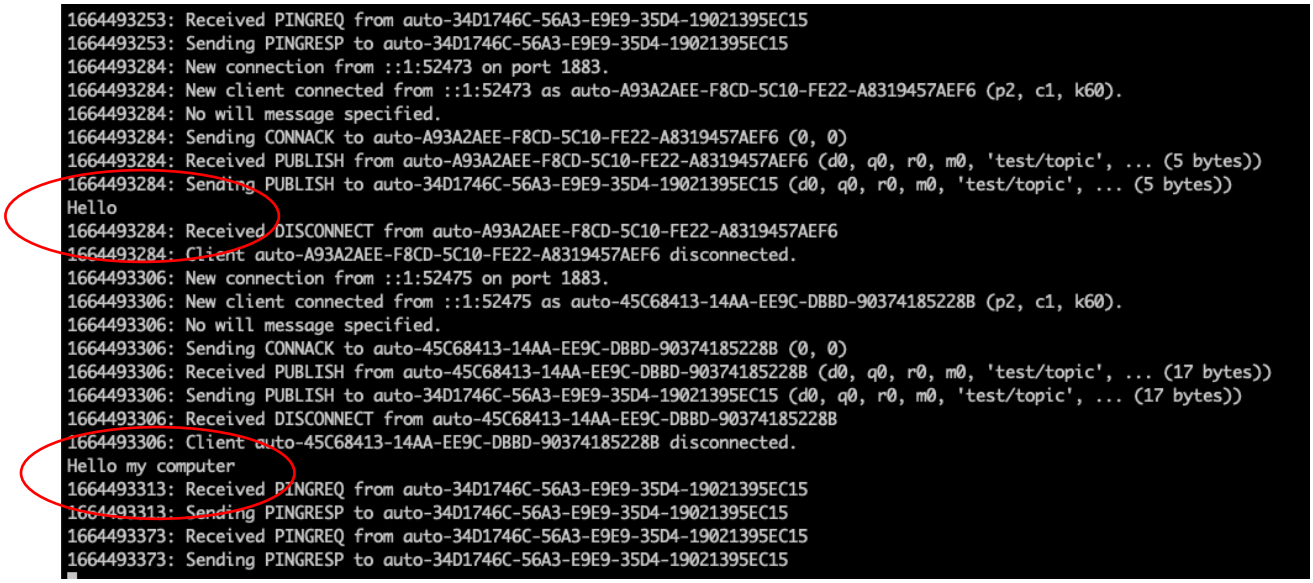
   $ mosquitto_sub -h localhost -p 1883 -t test/topic

```
jorge@sican:~ $ mosquitto_sub -h localhost -p 1883 -t test/topic
1664493134: New connection from ::1:52461 on port 1883.
1664493134: New client connected from ::1:52461 as auto-34D1746C-56A3-E9E9-35D4-19021395EC15 (p2, c1, k60).
1664493134: No will message specified.
1664493134: Sending CONNACK to auto-34D1746C-56A3-E9E9-35D4-19021395EC15 (0, 0)
1664493134: Received SUBSCRIBE from auto-34D1746C-56A3-E9E9-35D4-19021395EC15
1664493134:     test/topic (QoS 0)
1664493134: auto-34D1746C-56A3-E9E9-35D4-19021395EC15 0 test/topic
1664493134: Sending SUBACK to auto-34D1746C-56A3-E9E9-35D4-19021395EC15
```

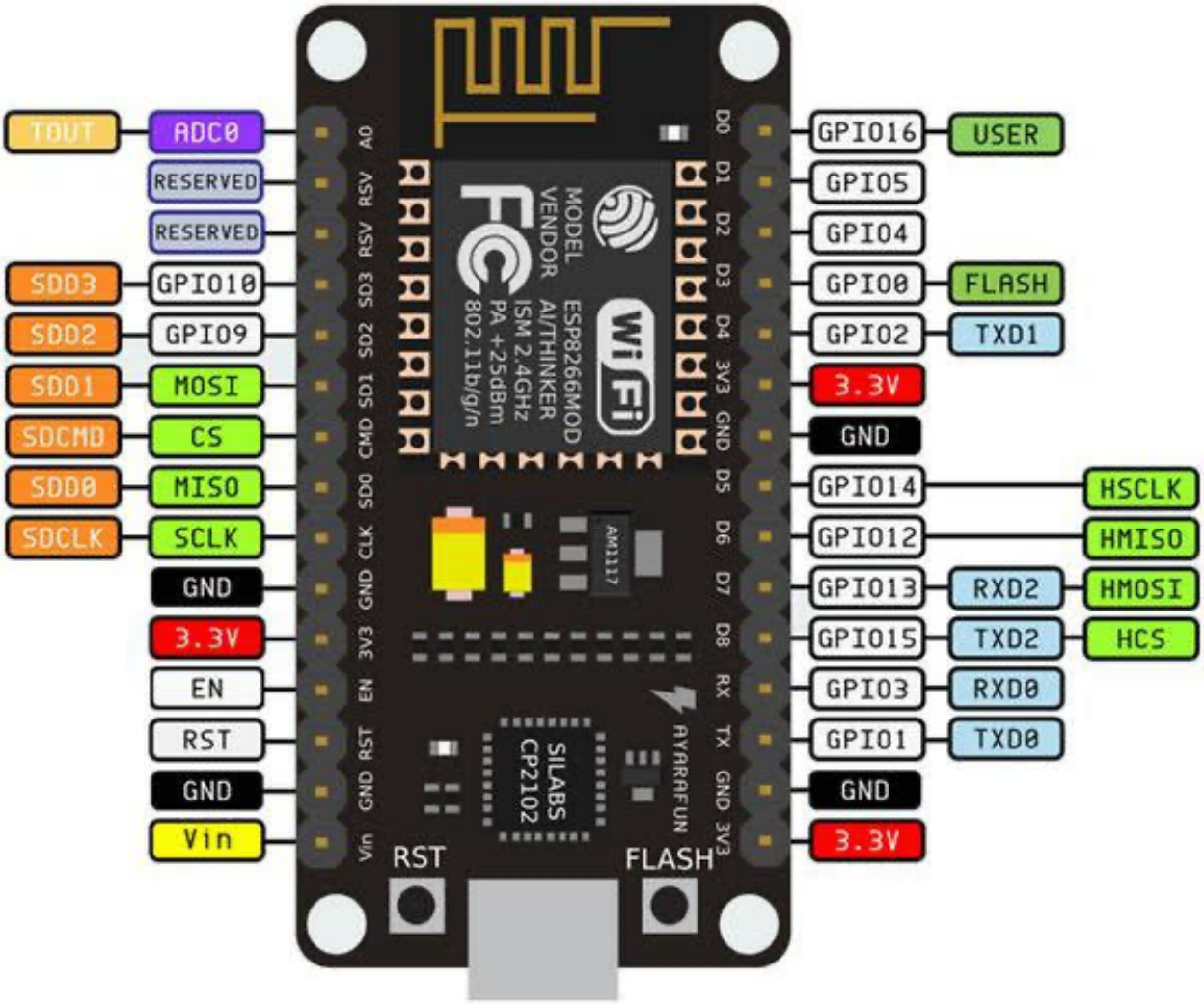
6. Open a new terminal (same environment) CTRL+T, then send a message to a specific topic (publish):
   $ mosquitto_pub -h localhost -p 1883 -t test/topic -m “Hello fom my pc!”

```
⌥⌘1   mosquitto_sub (mosquitto_sub)   ⌘1   ~ (-zsh)
Last login: Thu Sep 29 17:34:27 on ttys001
jorge@sican:~ $ mosquitto_pub -h localhost -p 1883 -t test/topic -m "Hello"
jorge@sican:~ $ mosquitto_pub -h localhost -p 1883 -t test/topic -m "Hello my computer"
jorge@sican:~ $
```

7. Observe on the other terminal:



8. Now we want to use the ESP8266 Arduino module to send messages to the broker:

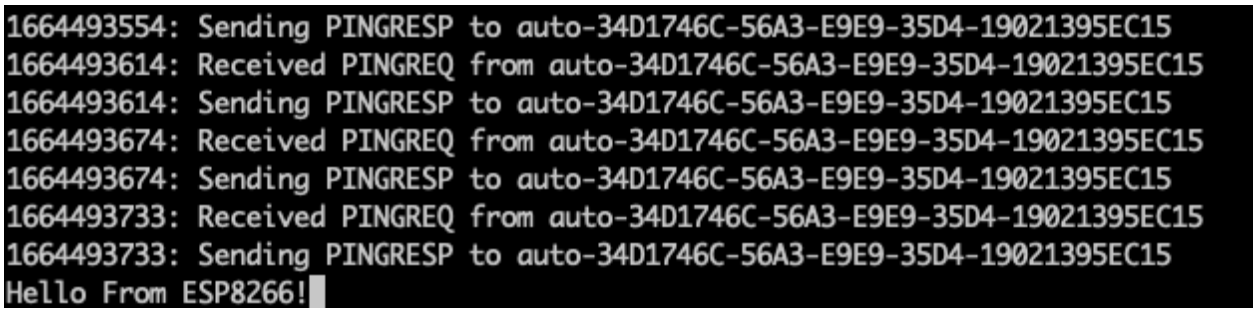


In your Arduino IDE:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>// WiFi
const char *ssid = ""; // Enter your WiFi name
const char *password = "";  // Enter WiFi password// MQTT
Broker
const char *mqtt_broker = ""; // Enter your WiFi or
Ethernet IP
const char *topic = "test/topic";
const int mqtt_port = 1883;WiFiClient espClient;
PubSubClient client(espClient);void setup() {
 // Set software serial baud to 115200;
 Serial.begin(115200);

 // connecting to a WiFi network
```

```
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("Connecting to WiFi..");
 }

 Serial.println("Connected to the WiFi network");

 //connecting to a mqtt broker
 client.setServer(mqtt_broker, mqtt_port);
 client.setCallback(callback);

 while (!client.connected()) {
 String client_id = "esp8266-client-";
 client_id += String(WiFi.macAddress());

 Serial.printf("The client %s connects to mosquitto mqtt
broker\n", client_id.c_str());

 if (client.connect(client_id.c_str())) {
  Serial.println("Public emqx mqtt broker connected");
 } else {
  Serial.print("failed with state ");
  Serial.print(client.state());
  delay(2000);
 }
}

 // publish and subscribe
 client.publish(topic, "Hello From ESP8266!");
 client.subscribe(topic);
}void callback(char *topic, byte *payload, unsigned int
length) {
 Serial.print("Message arrived in topic: ");
 Serial.println(topic);
 Serial.print("Message:");

 for (int i = 0; i < length; i++) {
  Serial.print((char) payload[i]);
 }

 Serial.println();
 Serial.println(" - - - - - - - - - - - -");
}void loop() {
 client.loop();
}
```

9. Expected output:



10. **OBS: First install**

    ARDUINO ADDON ESP8266
    https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide

    https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/installing-the-esp8266-arduino-addon
    Arduino core for ESP8266 WiFi chip
    https://github.com/esp8266/Arduino

    Arduino MQTT library
    https://github.com/knolleary/pubsubclient

    How to connect the board to Arduino?

11. **Challenge:** complete the MQTT message interchange!