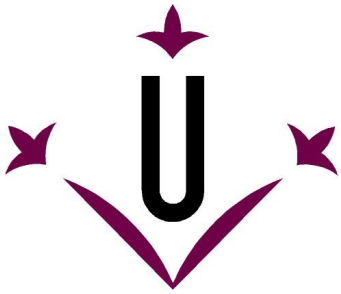

University of Lleida

Master's Degree in Informatics Engineering

Higher Polythecnic School

MQTT Broker

Ubiquitous Computing and Embedded Systems

Francesc Contreras

Albert Pérez

Marc Visa

November 10, 2021

Table of contents

List of Figures

1 Introduction

The purpose of this document is to explain the MQTT Broker development for the second sprint of the project.

We shall introduce all the steps and software we have used when developing it and showing images as a real example for detailed explanation.

2 Environment

Firstly, these are the steps for installing the libraries and setting up the Arduino IDE to get started for the MQTT Broker development:

1. Install ESP8266 Board from Boards Manager from Arduino IDE.
2. Install the following libraries from the Manager Libraries of Arduino IDE:
   (a) Adafruit MCP23017
   (b) Adafruit Unified Sensor by Adafruit.
3. Install the following libraries from GitHub: (These libraries should be added to the Arduino/libraries folder):
   (a) uMQTTBroker
4. Select NodeMCU v1.0 as compilation environment.
5. Connect the NodeMCU v1.0 along with the micro-USB adapter to your machine.
6. Code the program for NodeMCU v1.0.

2.1 Material required

Subsequently, you will find all the required components for the Data Producer 2, and a short description for those uncommon ones.

- **NodeMCU**: open source firmware for which open source prototyping board designs are available. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. Initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module.[6][7] Later, support for the ESP32 32-bit MCU was added.
- **Breadboard**: A board, having a matrix of small holes to which components may be attached without solder.
- **MCP23017**: Component that measures the accelerometer sensor, it allows us to use I2C or SPI. It is a port expander that gives you virtually identical PORTS compared to standard micro-controllers e.g. Arduino or PIC devices and it even includes interrupts. It gives you an extra 16 I/O pins using an I2C interface as well as comprehensive interrupt control.
- **Wires**
- **LED**

3 Development

As regards the development, this is based on two parts, the first one related to work directly with the electronic components, and the second by programming the arduino sketch.

The following sections shows you how the electronic schema is in real, as well as how finally we got the visualization of the acceleration.

Moreover, there is the code shown below, but also you may consult the GitHub repository of the project, the one is specified subsequently. So each led represents the number of subscribers that are connected to the broker. The two tables that are following are the MCP23017 GPIO mapping and the NodeMCU and MCP23017 pins mapping, respectively.

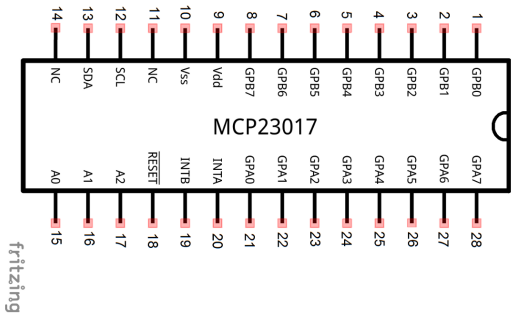

Figure 1: MCP23017 Schema

Pin Name	Physical PIN	Adafruit PIN	Function
GPB0	1	8	Bidirectional I/O pin.
GPB1	2	9	Bidirectional I/O pin.
GPB2	3	10	Bidirectional I/O pin.
GPB3	4	11	Bidirectional I/O pin.
GPB4	5	12	Bidirectional I/O pin.
GPB5	6	13	Bidirectional I/O pin.
GPB6	7	14	Bidirectional I/O pin.
GPB7	8	15	Bidirectional I/O pin.
VDD	9	NA	Power
VSS	10	NA	Ground
NC	11	NA	Not connected
SCL	12	NA	I Serial clock input
SDA	13	NA	I/O Serial data I/O
NC	14	NA	Not connected
A0	15	NA	Hardware address pin. Must be externally biased.
A1	16	NA	Hardware address pin. Must be externally biased.
A2	17	NA	Hardware address pin. Must be externally biased.
RESET	18	NA	Hardware reset. Must be externally biased.
INTB	19	NA	Interrupt output for PORTB.
INTA	20	NA	Interrupt output for PORTA.
GPA0	21	0	Bidirectional I/O pin.
GPA1	22	1	Bidirectional I/O pin.
GPA2	23	2	Bidirectional I/O pin.
GPA3	24	3	Bidirectional I/O pin.
GPA4	25	4	Bidirectional I/O pin.
GPA5	26	5	Bidirectional I/O pin.
GPA6	27	6	Bidirectional I/O pin.
GPA7	28	7	Bidirectional I/O pin.

Table 1: MCP23017 pin descriptions

Adafruit PIN	**Pin Name**	**Physical Pin**
0	GPA0	21
1	GPA1	22
2	GPA2	23
3	GPA3	24
4	GPA4	25
5	GPA5	26
6	GPA6	27
7	GPA7	28
8	GPB0	1
9	GPB1	2
10	GPB2	3
11	GPB3	4
12	GPB4	5
13	GPB5	6
14	GPB6	7
15	GPB7	8

Table 2: Adafruit MCP23017 pin mapping

MCU Pin	MCU Pin Name	MCP23017 Pin	MCP23017 Pin Name
17	D1	12	SCL
18	D2	13	SDA
21	3V3	18	Reset
21	3V3	9	VDD
22	GND	10	VSS
22	GND	15	A0
22	GND	16	A1
22	GND	17	A2

Table 3: Our schema

3.1 Hardware

3.1.1 Diagram schema

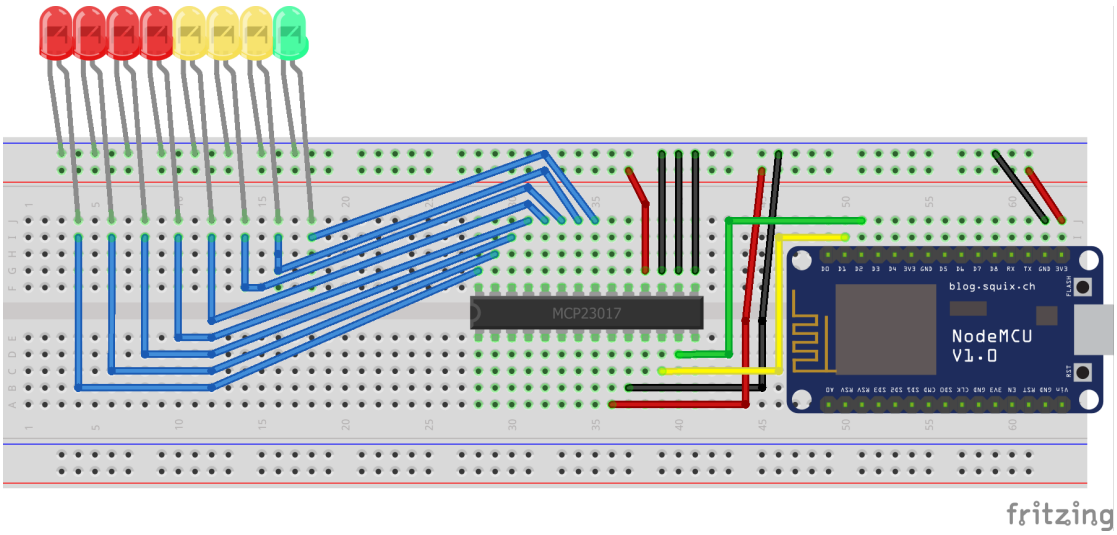

Figure 2: MQTT Broker Diagram

3.2 Testing

For the testing process of the development, it has been used the MQTT Explorer in order to check if there were real connection with the broker and also check if the Data Producers sent the data correctly. Additionally, as it can be observed in the figure, when testing if a client was connected it is coded as the green Led is lighted if one client was connected. If two clients were connected, the green one and the first yellow one were lighted and so in succession.

3.2.1 Data Producer 1

In this picture, we can see the number of connections that our MQTT Broker has, along with the retrieved data from temperature and humidity sensor

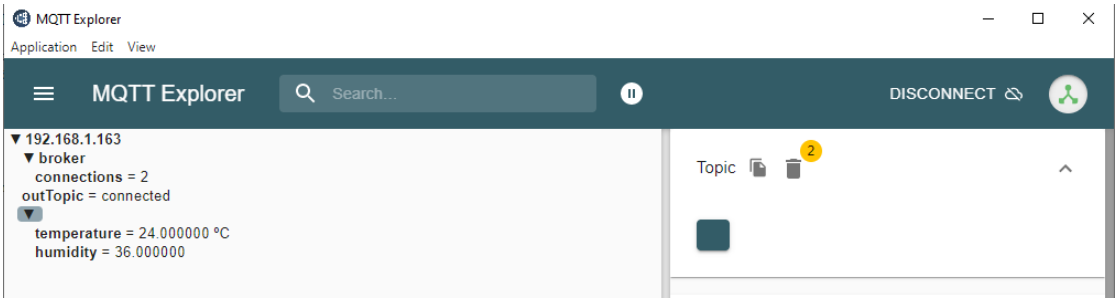

Figure 3: MQTT Explorer for Data Producer 1

For this second picture, we can see in the right side of the Explorer some information regarding the topics, values of the MQTT Broker. In this case, we have two topics, the humidity topic and the temperature topic. As for the values we can observe a history of the temperature and humidity sensor values, which the broker has retrieved

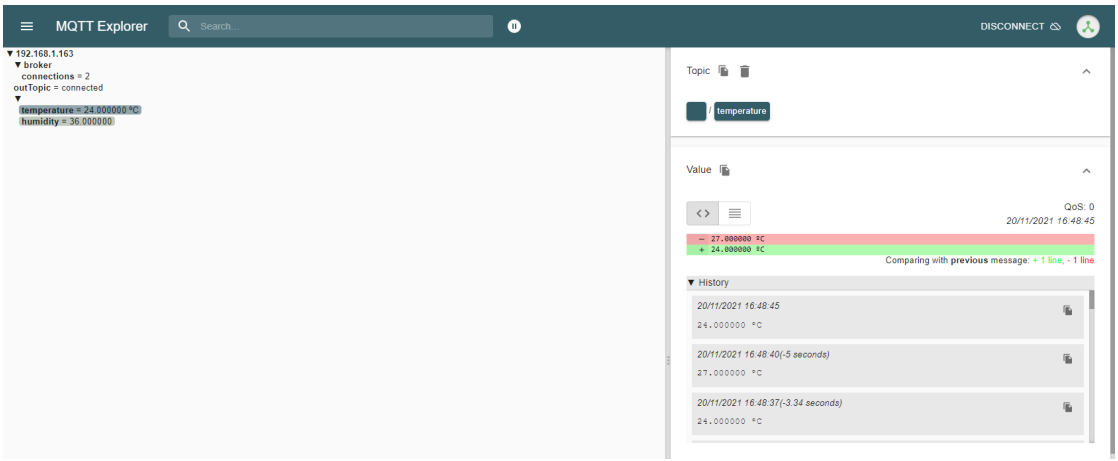

Figure 4: MQTT Explorer for Data Producer 1 History and values

Finally, in this picture we can observe that the Explorer shows us some chart that renders the high and low peaks, which represents the data retrieving period as a time-series.

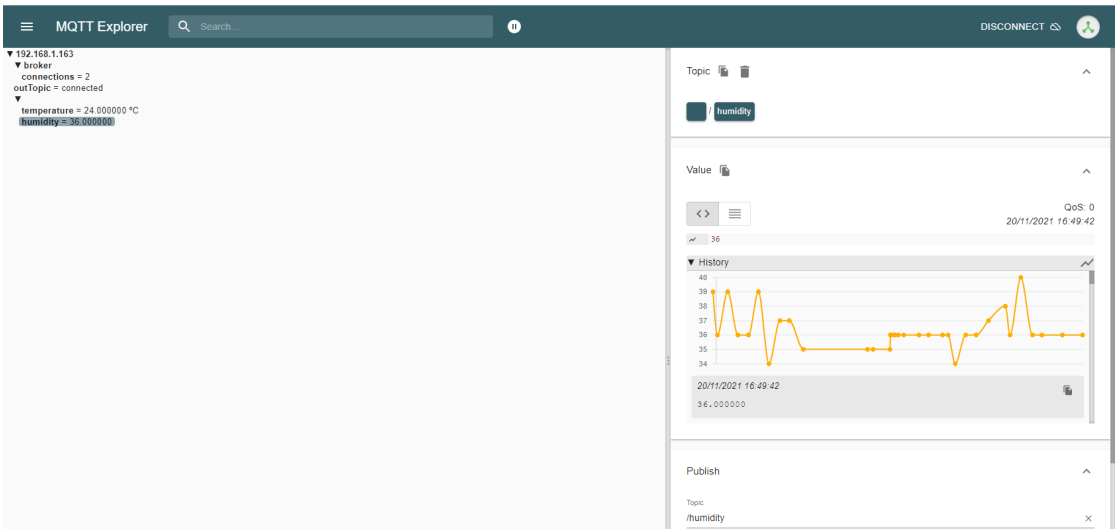

Figure 5: MQTT Explorer History chart

3.2.2 Data Producer 2

For the Data Producer 2 in the MQTT Broker, it is defined the acceleration topic, which the accelerometer publishes the data and the Broker retrieves it.

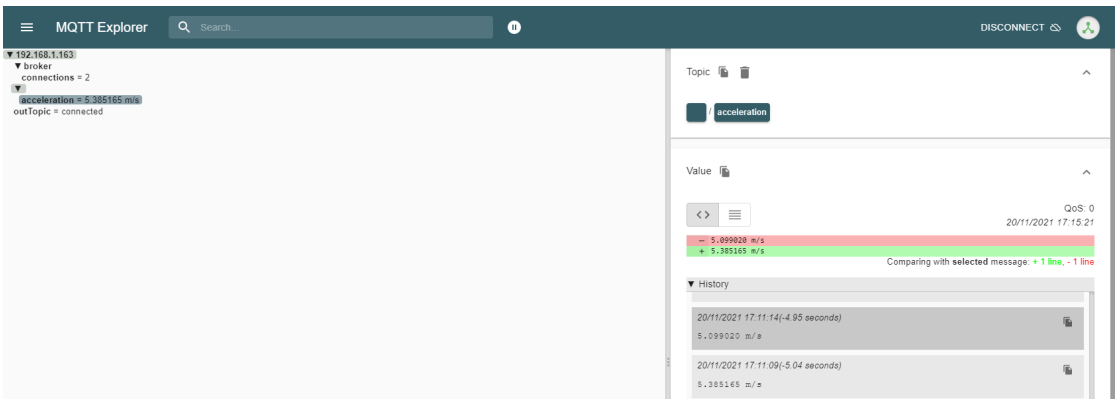

Figure 6: MQTT Explorer for Data Producer 2

3.2.3 All

In this final picture, we can observe both Data Producers that are publishing data in the three mentioned topics previously (humidity, temperature, acceleration) and the MQTT is retrieving all the data by the topics and shows the values of the respective topics on the left part of the screen of the MQTT Explorer.

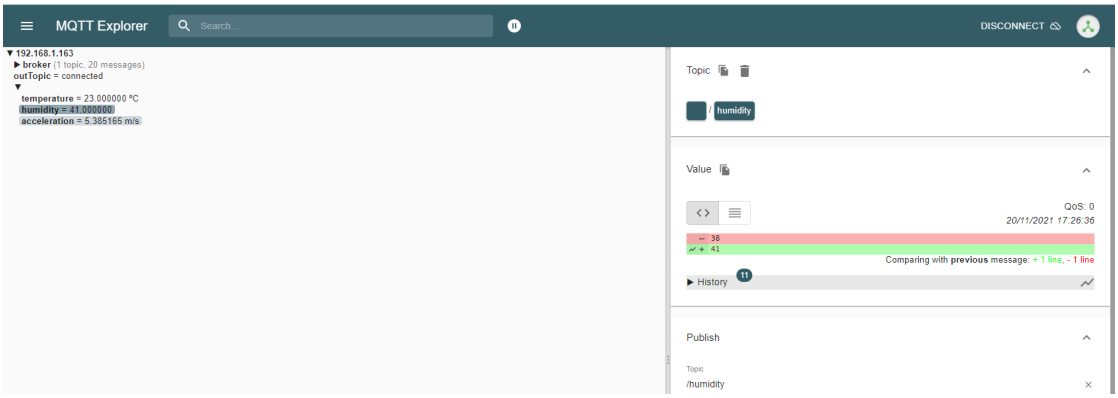

Figure 7: MQTT Explorer for both Data Producers

3.3 Software

```
// Import required libraries
#include <Arduino.h>
#include <ESP8266WiFi.h>

// MCP23017 Lib
#include <Wire.h>
#include <Adafruit_MCP23017.h>

// MQTT Broker
#include "uMQTTBroker.h"
uMQTTBroker myBroker;

// Replace with your network credentials
const char* ssid = "-";
const char* password = "-";

// Instance of MCP23017 library
Adafruit_MCP23017 mcp;

int counter = 0;
void setup(){

  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }

  // Print ESP8266 Local IP Address
  Serial.println(WiFi.localIP());

  //MCP
  mcp.begin(); //0x20

  // Init LED (RED = [7,4], Yellow = [3,1], Green=[0])
  for(int i=0; i<8;i++){
     mcp.pinMode(i, OUTPUT);
  }

  //Serial.println("Starting MQTT broker");
  myBroker.init();
  myBroker.subscribe("#"); // all topics
  myBroker.subscribe("/temperature");
  myBroker.subscribe("/humidity");
  myBroker.subscribe("/acceleration");
}
```

```
void loop(){

  delay(5000); // 5 seconds

  myBroker.publish("broker/connections", (String)myBroker.getClientCount());

 // MQTT Broker LED BAR

  Serial.println((String)"Current Clients: " + myBroker.getClientCount());

  // Check if we have clients
  if(myBroker.getClientCount() > 0){
      for(int i=0; i < myBroker.getClientCount();i++){
      if(i<8){
        mcp.digitalWrite(i, HIGH);
      }
     }
  }else{
    for(int i = 0; i < 8 ;i++){
      mcp.digitalWrite(i, LOW);
    }
  }
}
```

4 Code repository

Wind Turbine Generator Github repository

References

[1] MCP23017 Tutorial

[2] MCP23017 Diagram