KON435E – INDUSTRIAL DATA COMMUNICATIONS

FINAL PROJECT - FALL 2021

**TEAM 8

STUDENTS

NAME SURNAME : MEHMET ALİ ARSLAN 040170402

ERAY AYBEK 040170413

İSA SARAÇOĞLU 040180553

MEHMET SALİM KIZILTUĞ 040170401

ELSHAİMAA MOHAMED ABDELKADER 040170922

DUE DATE : 30.01.2022

COURSE ID & NAME : **KON 435E INDUSTRIAL DATA COMMUNICATIONS**

COURSE COOR. : **DOÇ. DR. ALİ FUAT ERGENÇ**

- Below, you can see general communication of our system:

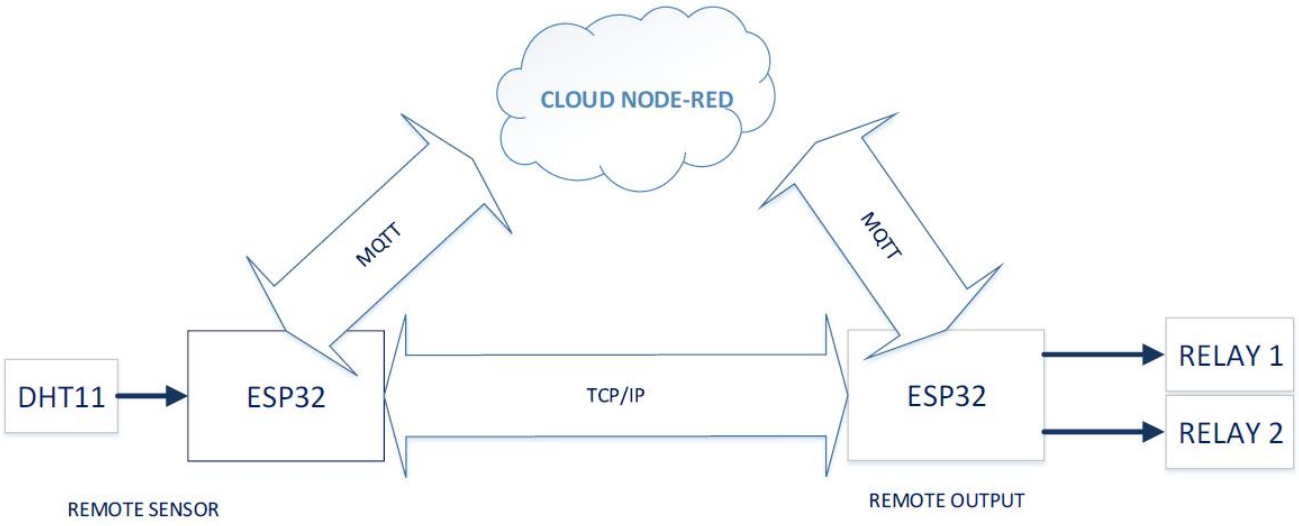


Figure 1. General Communication of the homework

1) Explanation and Comparison of MQTT, TCP/IP communications.

- Before coding the communication system, lets talk about MQTT and TCP/IP communication protocols and their differences from each other.

MQTT: MQTT (Message Queuing Telemetry Transport) protocol is a machine-to-machine (M2M) message-based protocol widely used on the Internet. It has been adopted in the Internet of Things (IoT) ecosystem with its light weight and low resource consumption. Almost all IoT cloud platforms support MQTT protocol to send and receive data from smart objects. This protocol establishes a TCP/IP connection in a publication-subscriber structure, as opposed to HTTP, which is based on a request-response structure. It works on Linux, Windows, Android, iOS, MacOS operating systems where TCP/IP protocol can be written.

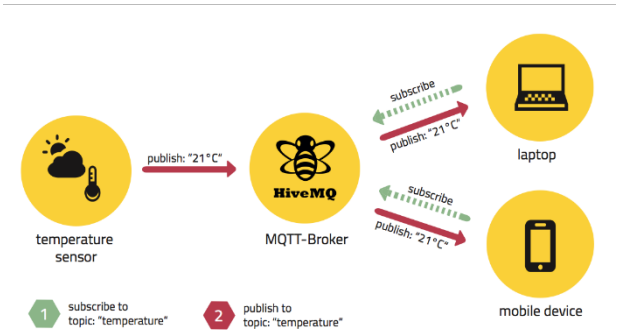


Figure 2. MQTT general structure

The MQTT protocol will separate a client (publisher) broadcasting a message in publisher-subscriber structure to other clients receiving the message (subscribers). Also, MQTT is asynchronous protocol, which means it doesn't block the client while waiting for the message. Unlike the HTTP protocol, it is essentially a concurrent protocol. Another feature of the MQTT protocol is that it does not require the client (subscriber) and publisher to be connected at the same time.

TCP/IP: TCP/IP, which stands for Transmission Control Protocol/Internet Protocol, is a package that contains the basic protocols of the internet. It was formed by the combination of many protocols. The TCP part specifies the important points in data transfer, while the IP part specifies the transport path. The protocol structure is divided into 5 as Application Layer, Transport Layer, Internet Layer, Network Access Layer and Physical Layer. Transmission Control Protocol accepts data from a data stream, divides it into chunks, and adds a TCP header creating a TCP segment The TCP segment is then encapsulated into an Internet Protocol (datagram, and exchanged with peers). Both TCP and MQTT protocol, we send string or char. to other device.

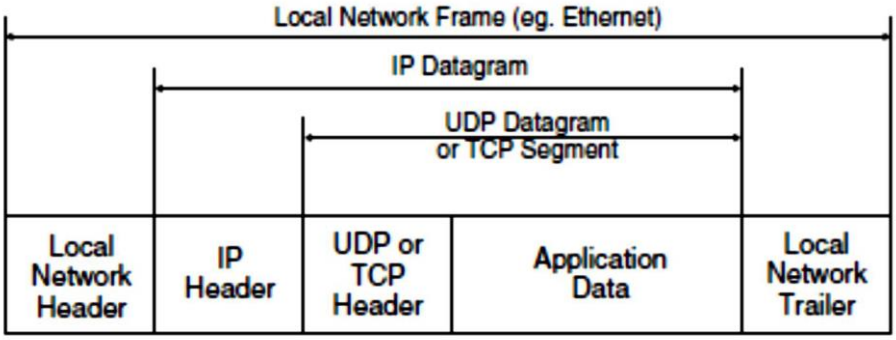


Figure 3. TCP/IP general structure

MQTT	TCP/IP
It works on publish/subscribe model.	It works on request/response model.
It has less complexity.	It is more complex.
It runs over Transmission Control Protocol.	It runs over Transmission Control Protocol (TCP) and can also adapted to User Datagram Protocol.
This protocol's design is Data centric.	This protocol's design is Document centric.
The message size generated is less as it uses binary format.	The message size generated is more as it uses ASCII format.
It provides data security with SSL/TLS.	It does not provide security but Https is built for that.
Less power consumption for Industry.	More power consumption for industry
Easy to use and faster communication.	Harder to use and slower communication.

1.1) Lets Realize MQTT Communication between our esp8266's.

- Lets look at our circuit diagram as follows.

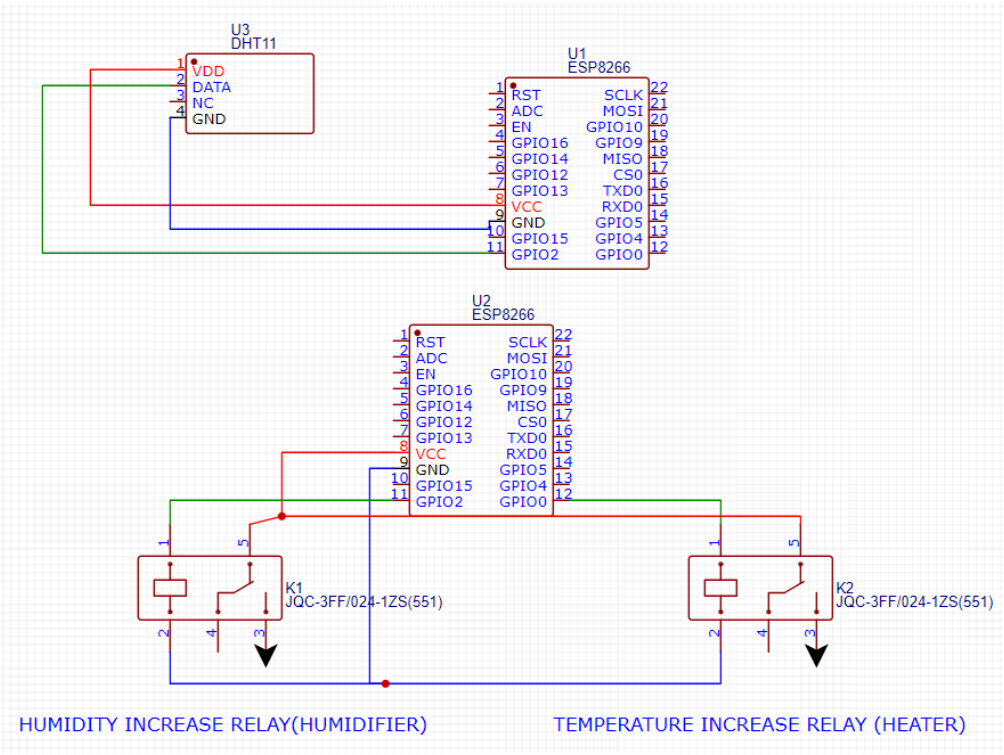


Figure 4. Circuit Diagram (easyeda)

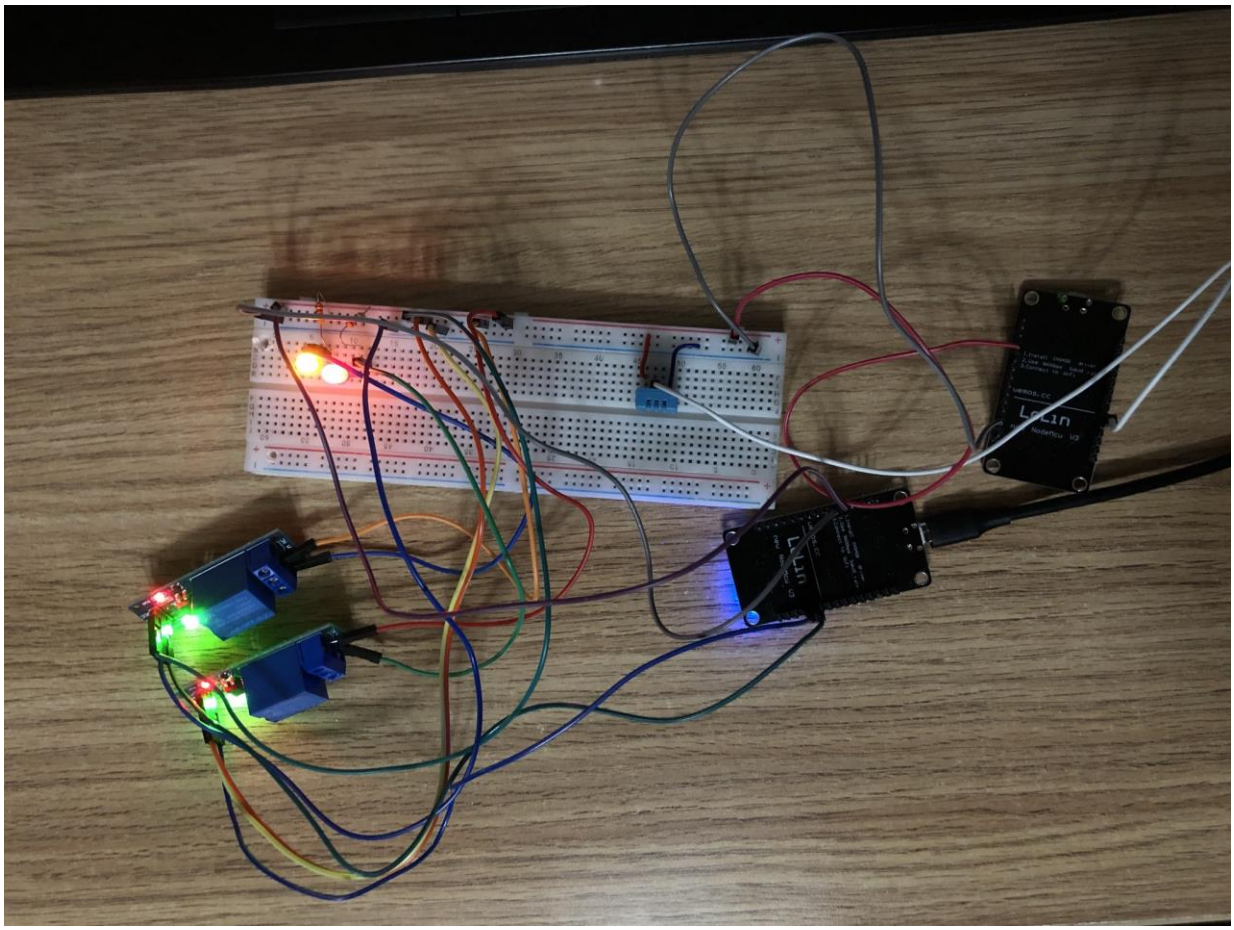
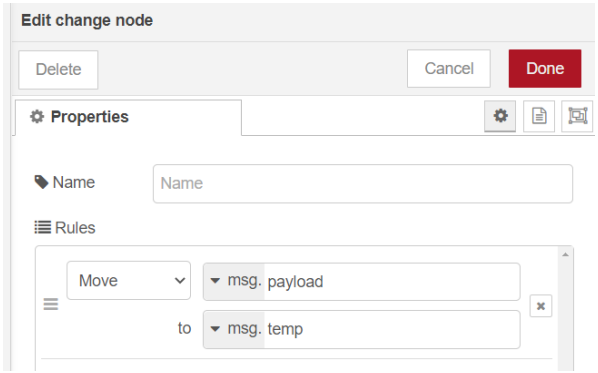
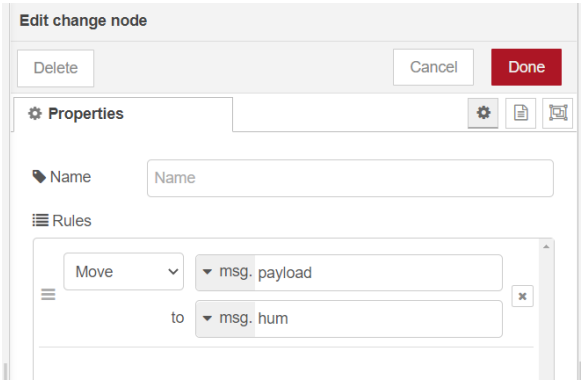
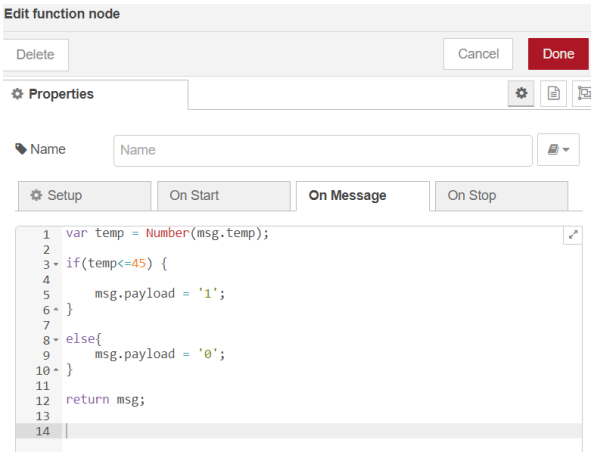


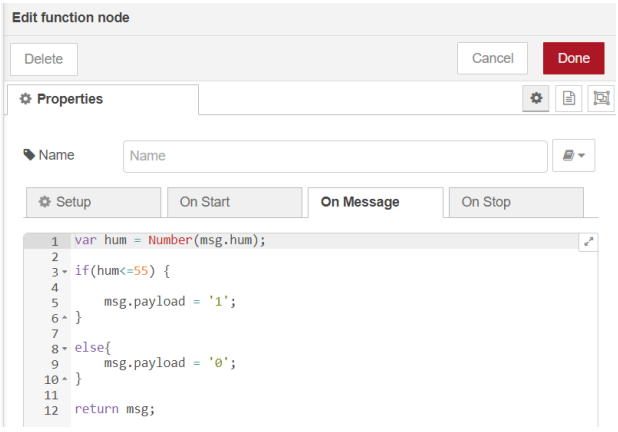
Figure 5. Circuit Diagram (real)

- **Main idea**: Upper device 1 measures the temperature and humidity data from dht11 sensor and sends it via MQTT protocol to our broker. In our flow on broker, we get the data and moved to msg.temp and msg.hum variables. Then we use our functions as master to controlling relays' variables. **If temperature is less than 45, sending "1" to device2 else sending "0". If humidity is less than 55, sending "1" to device2 else sending "0".** You can see that below:

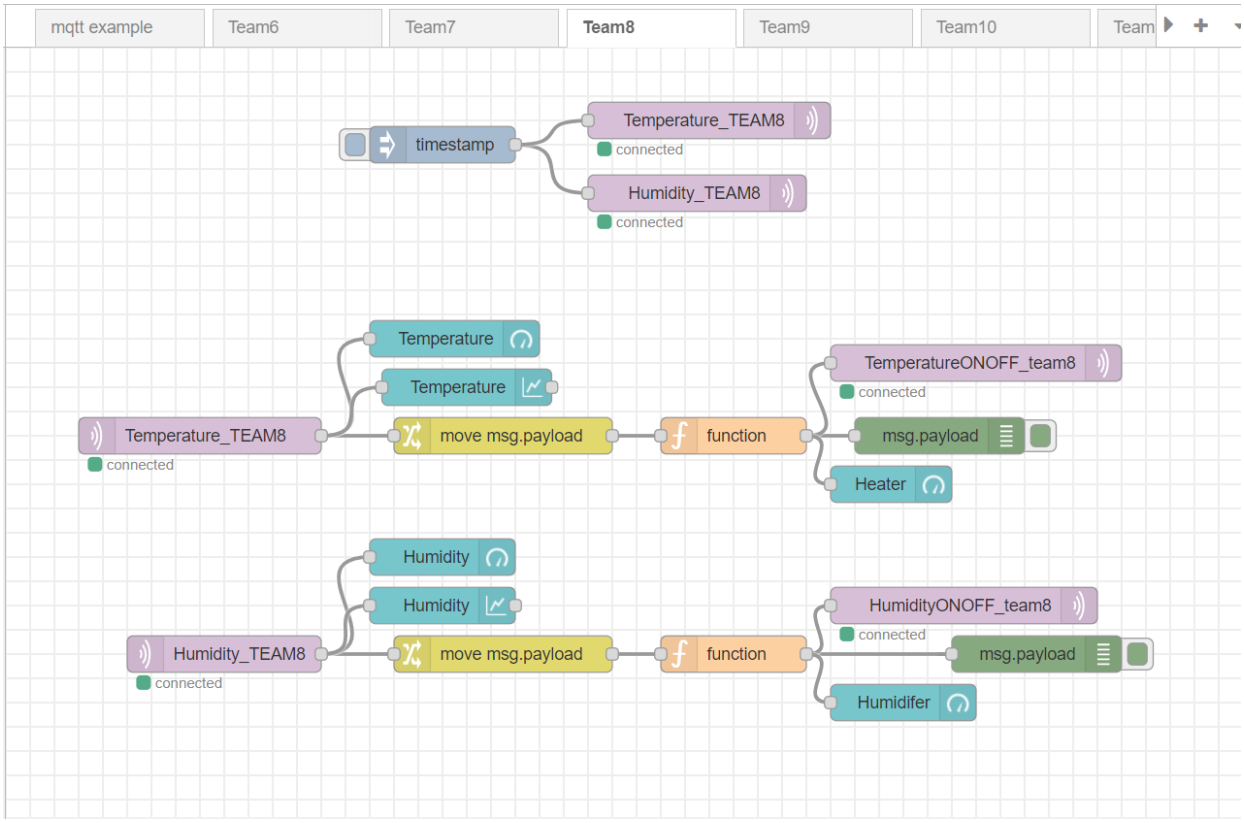








- Lets look at our flow on Node-red as follows:



1.2) CODES FOR MQTT COMMUNICATION (explanations are with the code as a comment.)

DEVICE 1 CODE (MEASURES THE TEMP. AND HUM. DATA with DHT11):

```
#include <DHT.h>
#include "ESP8266WiFi.h"
#include "PubSubClient.h"

const char* ssid = "Arslan 2.4 GHz"; //Entering WiFi SSID
const char* password = [redacted]; //Entering Password of WiFi

#define mqtt_server "160.75.154.101"  //We are team 8, so we need to connect with 101 ending.
#define mqtt_temp "Temperature_TEAM8"   //Our first topic for publishing
#define mqtt_humidity "Humidity_TEAM8"  //Our second topic for publishing

WiFiClient espClient;            //Connection to mqrr server.
PubSubClient client(espClient);

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

float h,t;


void WifiandMqttConnection()                    //Our wifi connection function for connect our wifi.
{
 WiFi.begin(ssid,password);                     //Connectiong to wifi
 while (WiFi.status() != WL_CONNECTED){delay(500); Serial.println("* Trying to connect...");}
 Serial.println("");
 Serial.println("WiFi connection Successful!!");         //Prints this when connection is OK.
 Serial.print("The IP Address of ESP8266 Module is: ");
 Serial.println(WiFi.localIP()); // Print the IP address
 WiFi.mode(WIFI_STA);
}
void ReadTemp_Hum_print()   //For reading temperature, humidity and printing them to serial port screen.
{
  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.println(("*********************"));
  Serial.print((" Humidity: "));
  Serial.println(h);
  Serial.print((" Temperature: "));
  Serial.println(t);
}

void reconnect()               //Reconnect algorithm if lose connection.
{
  int counter = 0;
  while(!client.connected())
  {
    if(counter==5)
    {
      ESP.restart();
      }
    counter+=1;
    Serial.println("Attempting MQTT connection...");
    if(client.connect("Team8_device1"))
    {
      Serial.println("connected!");
    }
    else
    {
      Serial.print("failed:");
      Serial.print(client.state());
      delay(1000);        //wait to retry for 1 seconds.
    }
    }
}
```

```
void setup() {
 Serial.begin(115200);
 WifiandMqttConnection();  //Calling for wifi connection.
 dht.begin();
 Serial.println("Connected to the WiFi network");
 client.setServer(mqtt_server, 1884);                //connecting to a mqtt broker
 client.connect("Team8_device1","iturockwell","963258741");   //Entering broker id and password
}

void loop() {
  ReadTemp_Hum_print();   //Calling for printing temperature and humidity to serial port screen.
  client.publish(mqtt_temp, String(t).c_str(),true);         //Publishing our data to mqtt broker with these codes.
  client.publish(mqtt_humidity, String(h).c_str(),true);
  reconnect();
  if(WiFi.status() != WL_CONNECTED){ Serial.println((" Wifi connecion loss "));}
  if(client.connected() == 1){Serial.println((" *MQTT CONNECTED* "));}
  delay(500);

}
```

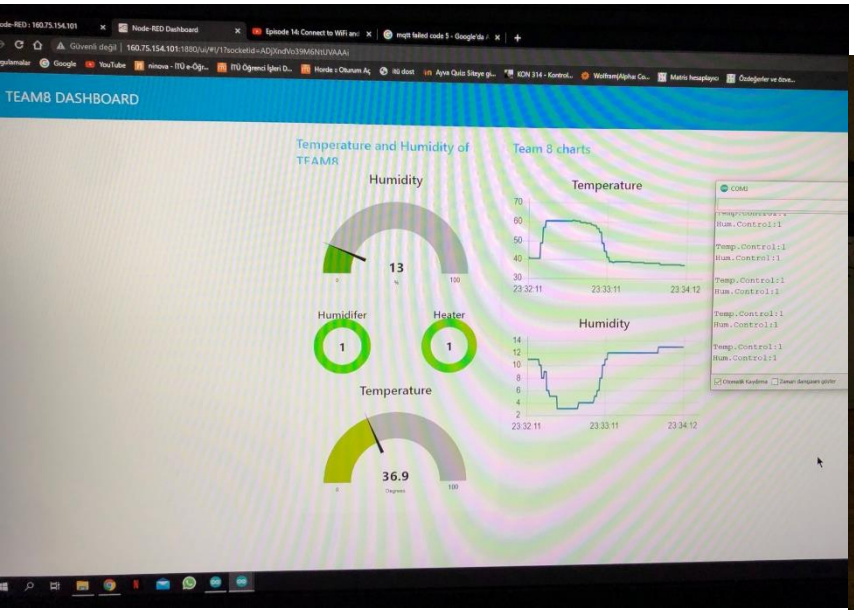
DEVICE 2 CODE (GETTING THE ON OFF DATA FOR BOTH HUMIDIFIER AND HEATER, Controlling Relays): (RED LED IS HEATER, YELLOW LED IS HUMIDIFIER.)

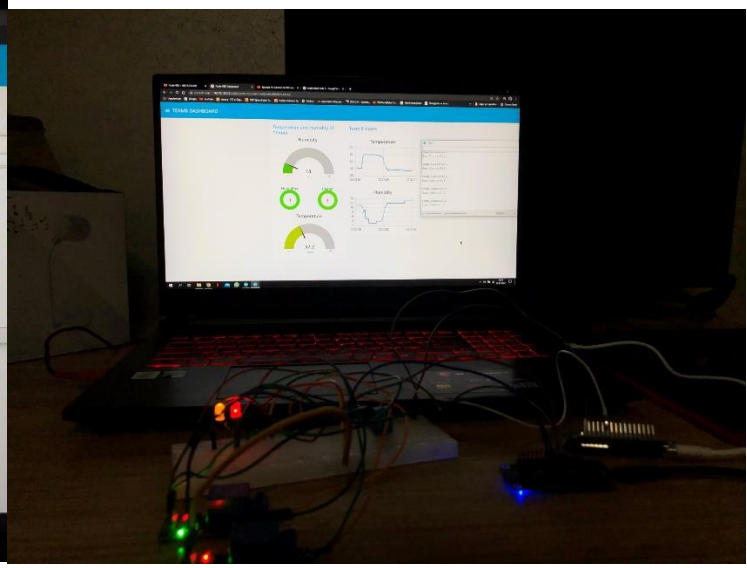
- We are getting the datas from broker with mqttCallback function. Deciding on or off situations of relays in this function too.

```
#include "ESP8266WiFi.h"
#include "PubSubClient.h"

String TempONOFF, HumONOFF;             // will get datas to this buffers

const char* ssid = "Arslan 2.4 GHz";    //Entering WiFi SSID
const char* password = [redacted]; //Entering WiFi Password

#define mqtt_server "160.75.154.101"  //We are team 8, so we need to connect with 101 ending.
#define mqtt_temp "TemperatureONOFF_team8"    // Getting topics
#define mqtt_humidity "HumidityONOFF_team8"   // Getting topics

#define Temprelay_REDled_pin 0          //Relay1 pin (Heater)
#define Humrelay_YELLOWled_pin 2        //Relay1 pin (Humidifier)

WiFiClient espClient;            //Connection to mqtt server.
PubSubClient client(espClient);



void WifiandMqttConnection()                    //Our wifi connection function for connect our wifi.
{
 WiFi.begin(ssid,password);
 while (WiFi.status() != WL_CONNECTED){delay(500); Serial.println("* Trying to connect...");}
 Serial.println("");
 Serial.println("WiFi connection Successful!!");
 Serial.print("The IP Address of ESP8266 Module is: ");
 Serial.println(WiFi.localIP());// Print the IP address
 WiFi.mode(WIFI_STA);
}
void mqttCallback(char* topic, byte* message, unsigned int length)  //This function is for getting data from mqtt broker (subscribed data).
{
 String messagetemp;
 for (int i=0; i<length;i++)
 {
   messagetemp += (char)message[i];
 }
 Serial.println();
 if(String(topic) =="TemperatureONOFF_team8")TempONOFF = messagetemp; Serial.print("Temp.Control:");Serial.println(TempONOFF); // giving the recieved value to output
 if(String(topic) =="HumidityONOFF_team8")HumONOFF = messagetemp; Serial.print("Hum.Control:");Serial.println(HumONOFF);
 if(TempONOFF == "1") digitalWrite(Temprelay_REDled_pin,LOW); //If "1" recieved, turning on red led with LOW signal (because its relay.)
 if(TempONOFF == "0") digitalWrite(Temprelay_REDled_pin,HIGH);
 if(HumONOFF == "1") digitalWrite(Humrelay_YELLOWled_pin,LOW);  //If "1" recieved, turning on yellow led with LOW signal (because relay.)
 if(HumONOFF == "0") digitalWrite(Humrelay_YELLOWled_pin,HIGH);
 //delay(100);
}
```

```
void reconnect()          //Reconnect algorithm if lose connection.
{
  int counter = 0;
  while(!client.connected())
  {
    if(counter==5)
    {
      ESP.restart();
      }
    counter+=1;
    Serial.println("Attempting MQTT connection...");
    if(client.connect("Team8_device2"))
    {
      Serial.println("connected!");
    }
    else
    {
      Serial.print("failed:");
      Serial.print(client.state());
      delay(1000);
    }
    }
}
void setup()
{
 pinMode(Temprelay_REDled_pin, OUTPUT);
 pinMode(Humrelay_YELLOWled_pin, OUTPUT);
 Serial.begin(115200);
 WifiandMqttConnection();  //Calling for wifi connection.
 Serial.println("Connected to the WiFi network");
 client.setServer(mqtt_server, 1884); //connecting to a mqtt broker
 client.setCallback(mqttCallback);
 client.connect("Team8_device2","iturockwell","963258741");
 if(client.connected() == 1){Serial.println((" *MQTT CONNECTED* "));}
}
void loop()
{
 reconnect();
 client.loop();
 client.subscribe(mqtt_temp);     //Subscribing the data topics for getting the data.
 client.subscribe(mqtt_humidity);
 delay(100);
 }
```

- Example Application screenshots for MQTT:





1.3) Lets Realize TCP/IP Communication between our esp8266's (same circuit, objective).

- Circiut diagram is the same. Both TCP and MQTT protocol, we send string or char. to other device.
- **Main idea**: **Device 1(server)** measures the temperature and humidity data from dht11 sensor and with if-else, decides the relays to be on/off. Then sends the on or off data via TCP protocol to our **Device 2(client)** with WiFi. Then we use the recieved data to controlling relays' variables. **If temperature is less than 45, sending "Temp1" to device2 else sending "Temp0". If humidity is less than 55, sending "Hum1" to device2 else sending "Hum0".** You can see codes below:

DEVICE 1 CODE (MEASURES THE TEMP. AND HUM. DATA with DHT11, SEND ON/OFF DATA. SERVER):

```
#include <DHT.h>
#include <SPI.h>
#include <ESP8266WiFi.h>

char ssid[] = "Arslan 2.4 GHz";               // SSID of our WiFi
char pass[] = [redacted];                     // password of our WiFi

#define DHTPIN 2    //DHT Configurations
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

float h,t;

WiFiServer server(80);            //This is server.

IPAddress ip(192, 168, 0, 80);            // IP address of the server
IPAddress gateway(192,168,0,1);           // gateway of our network
IPAddress subnet(255,255,255,0);          // subnet mask of our network

void ReadTemp_Hum_print()    //For reading temperature, humidity and printing them to serial port screen.
{
  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.println(("**********************"));
  Serial.print((" Humidity: "));
  Serial.println(h);
  Serial.print((" Temperature: "));
  Serial.println(t);
}
void setup() {
  Serial.begin(115200);                   // for debugging
  dht.begin();
  WiFi.config(ip, gateway, subnet);       // forces to use the fix IP
  WiFi.begin(ssid, pass);                 // connecting to our WiFi router
  while (WiFi.status() != WL_CONNECTED){Serial.println("*Trying to connect...");delay(500);}
  if(WiFi.status() == WL_CONNECTED){Serial.println("WiFi CONNECTED !!");}
  server.begin();                         // starts the server
  Serial.println("Connected to the WiFi network");
}

void loop () {
  ReadTemp_Hum_print();   //Calling for printing temperature and humidity to serial port screen.
  WiFiClient client = server.available();           //Server is available
  if (client) {            //If client is online.
    if (client.connected()) {   //if connecting with client.
      Serial.println("...");  //for debugging.
      client.flush();
      if(t<=45){client.println("Temp1\r");} else{client.println("Temp0\r");} // sends the on/off data of heater to the client.
      if(h<=55){client.println("Hum1\r");} else{client.println("Hum0\r");} // sends the on/off data of humidifier to the client.
    }
    client.stop();                // tarminates the connection with the client
  }
  delay(500);
}
```

DEVICE 2 CODE (GETTING THE ON OFF DATA FOR BOTH HUMIDIFIER AND HEATER, Controlling Relays.CLIENT):

```
#include <SPI.h>
#include <ESP8266WiFi.h>
#define Temprelay_REDled 0          //Relay1 pin (Heater)
#define Humrelay_YELLOWled 2        //Relay2 pin (Humidifier)

const char* ssid = "Arslan 2.4 GHz";           // SSID of our home WiFi
const char* pass = [redacted];                 // password of our home WiFi
unsigned long askTimer = 0;
IPAddress server(192,168,0,80);       // the fix IP address of the server
WiFiClient client;                    // this is client.

void setup()
{
  pinMode(Temprelay_REDled, OUTPUT);
  pinMode(Humrelay_YELLOWled, OUTPUT);
  Serial.begin(115200);               // only for debugging
  //At the beginnig, closeing leds with HIGH (for relays its inverse logic --> high for close leds, low for open leds.)
  digitalWrite(Temprelay_REDled,HIGH);
  digitalWrite(Humrelay_YELLOWled,HIGH);
  WiFi.begin(ssid, pass);             // connects to the WiFi router
  while (WiFi.status() != WL_CONNECTED) {Serial.println("Trying to connect...");delay(500);}
  if(WiFi.status() == WL_CONNECTED){Serial.println("WiFi CONNECTED !!");}
}
void loop ()
{
  client.connect(server, 80);   // Connection to the server
  Serial.println(".");
  String data = client.readStringUntil('\r');   // receives the answer from the sever
  Serial.println("from server: " + data);
  //If on command recieved, turning on red led with LOW signal (because relay.)
  if(data == "Temp1"){digitalWrite(Temprelay_REDled,LOW);} else{digitalWrite(Temprelay_REDled,HIGH);}
  //If on command recieved, turning on yellow led with LOW signal (because relay.)
  if(data == "Hum1"){digitalWrite(Humrelay_YELLOWled,LOW);} else{digitalWrite(Humrelay_YELLOWled,HIGH);}
  client.flush();
}
```

THE END OF THE REPORT. THANK YOU.

TEAM 8