

PROJECT DEVELOPMENT DELIVERY OF SPRINT-1

Date	30 October 2022
Team ID	PNT2022TMID38131
Project Name	Project - SmartFarmer -IOT Enabled smart farming application
Team Leader	Manoj Kumar.S
Team Member	Surya.G
Team Member	Vignesh.E
Team Member	Surendar.K

SPRINT-1

Simulation : To Create the Simulation by connecting the sensors by using the Arduino and connect with the code.

ABOUT SPRINT 1:

This simulation used to connect IBM Watson IOT platform devices to wokwi to monitor Temperature and Humidity and control light on and off connections.

SIMULATION IMAGE:

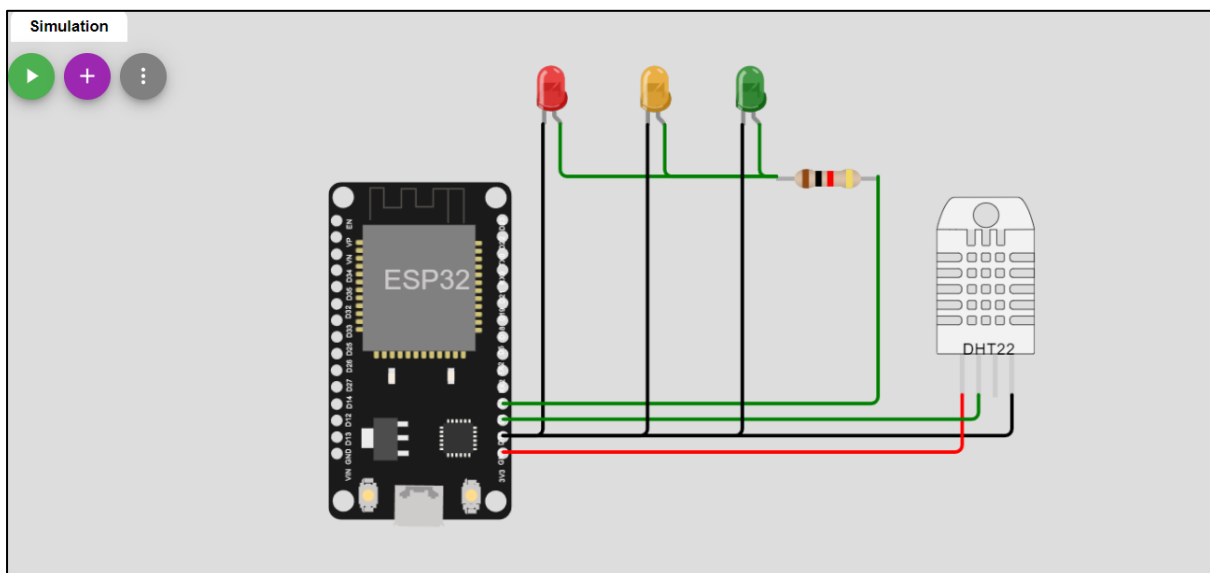


FIG 1:IMAGE OF SIMULATION

CODE:

```
#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQTT

#include "DHT.h"// Library for dht11

#define DHTPIN 15      // what pin we're connected to

#define DHTTYPE DHT22  // define type of sensor DHT 11

#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin
and type of dht connected

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

//-----credentials of IBM Accounts-----

#define ORG "fxm0tp"//IBM ORGANIZATION ID
#define DEVICE_TYPE "SMART-IOT"//Device type mentioned in ibm watson
IOT Platform
#define DEVICE_ID "MSVS-SMART-IOT"//Device ID mentioned in ibm
watson IOT Platform
#define TOKEN "R&-B5p)HZ!e3_@Lt6B"      //Token
String data3;
float h, t;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";//
Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and
type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";//
cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
```

```

PubSubClient client(server, 1883, callback ,wifiClient); //calling
the predefined client id by passing parameter like server id,portand
wificredential
void setup()// configureing the ESP32
{
    Serial.begin(115200);
    dht.begin();
    pinMode(LED,OUTPUT);
    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop()// Recursive Function
{

    h = dht.readHumidity();
    t = dht.readTemperature();
    Serial.print("temperature:");
    Serial.println(t);
    Serial.print("humidity:");
    Serial.println(h);

    PublishData(t, h);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

/*.....retrieving to
Cloud.....*/

void PublishData(float temp, float humid) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm
    cloud
    */
    String payload = "{\"Temperature\":\"";
    payload += temp;
    payload += "," " \"Humidity\":\"";
    payload += humid;
    payload += "}";
}

```

```

Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on
the cloud then it will print publish ok in Serial monitor or else it
will print publish failed
} else {
    Serial.println("Publish failed");
}
}

void mqttconnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!!!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }

        initManagedDevice();
        Serial.println();
    }
}

void wificonnect() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void initManagedDevice() {
    if (client.subscribe(subscribetopic)) {
        Serial.println((subscribetopic));
        Serial.println("subscribe to cmd OK");
    } else {

```

```

        Serial.println("subscribe to cmd FAILED");
    }
}

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
        //Serial.print((char)payload[i]);
        data3 += (char)payload[i];
    }

    Serial.println("data: "+ data3);
    if(data3=="lighton")
    {
        Serial.println(data3);
        digitalWrite(LED,HIGH);

    }

    else
    {
        Serial.println(data3);
        digitalWrite(LED,LOW);

    }
    data3="";

}

```

LIBRARIES:

```

# Wokwi Library List
# See https://docs.wokwi.com/guides/libraries

# Automatically added based on includes:
DHT sensor library

PubSubClient

```

ACESSING TEMPERATURE AND HUMIDITY BY MIT APP

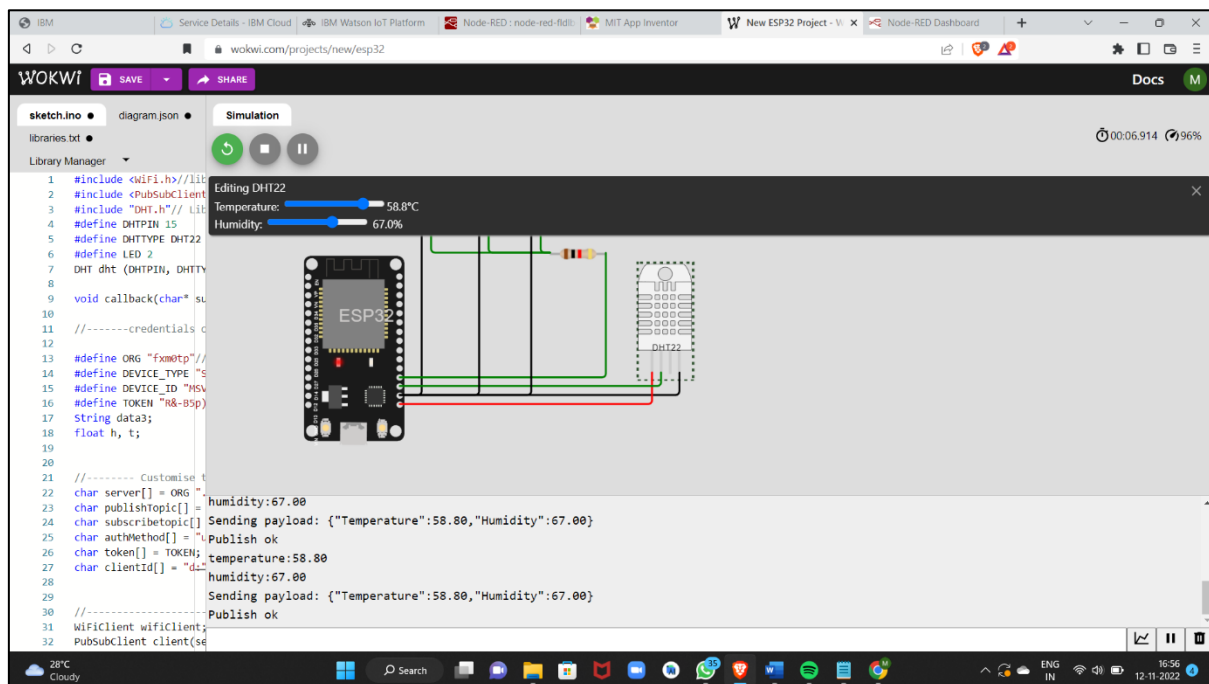


FIG 2: STARTING SIMULATION

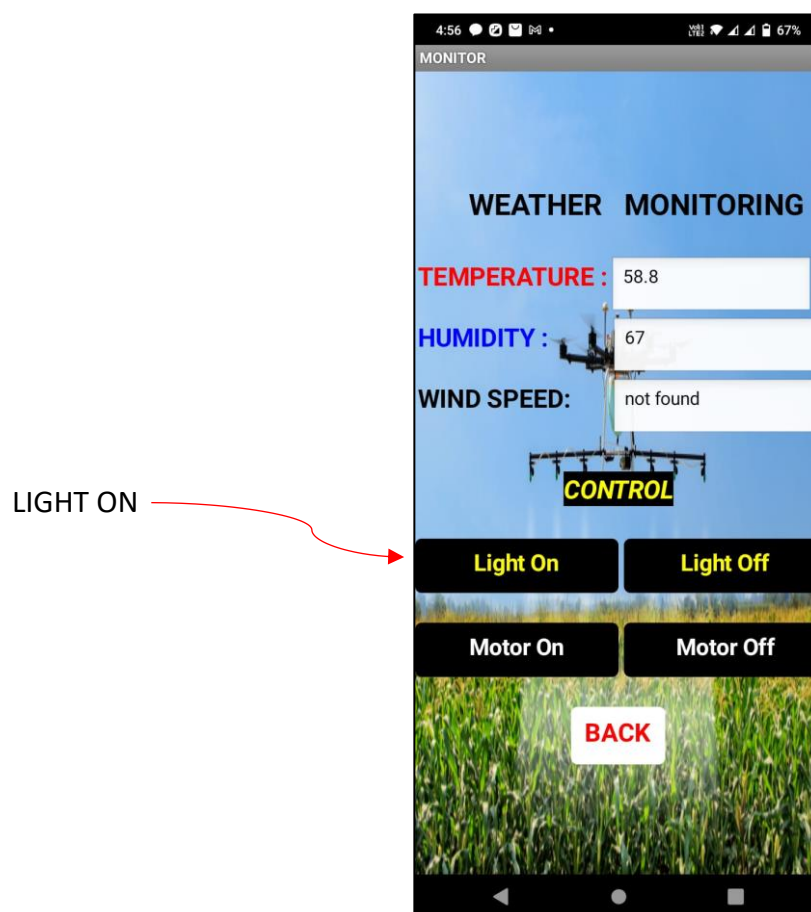


FIG 3: TOUCHING LIGHT ON BUTTON IN MIT APP

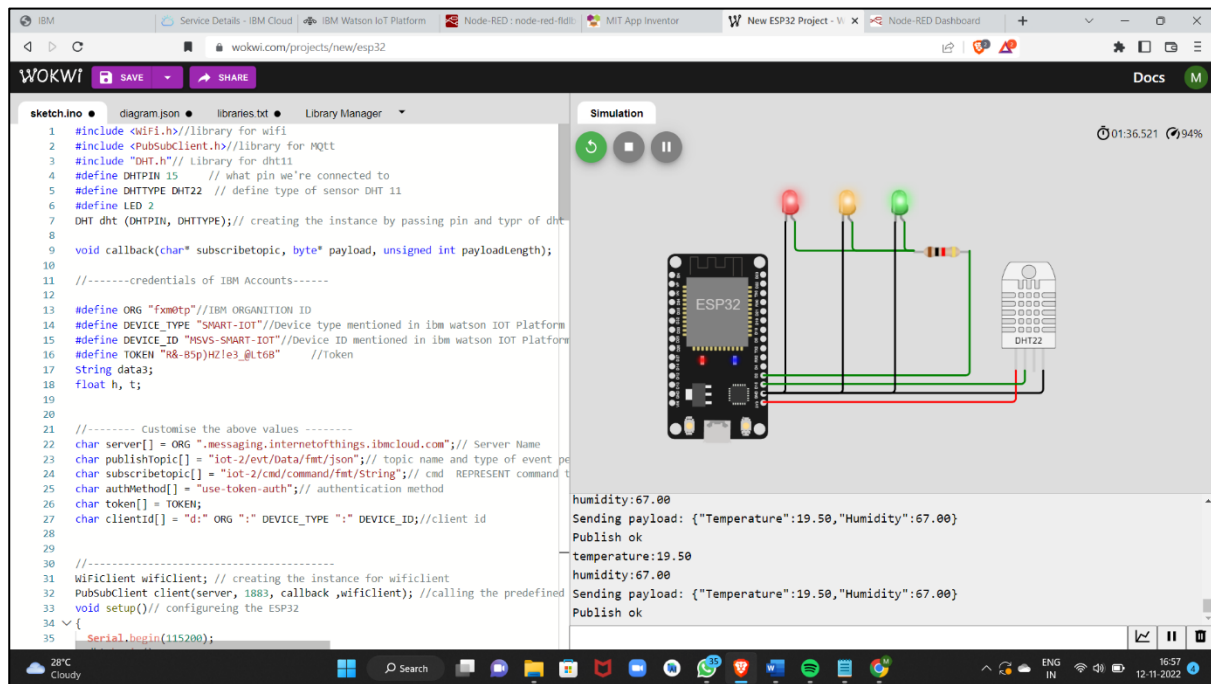
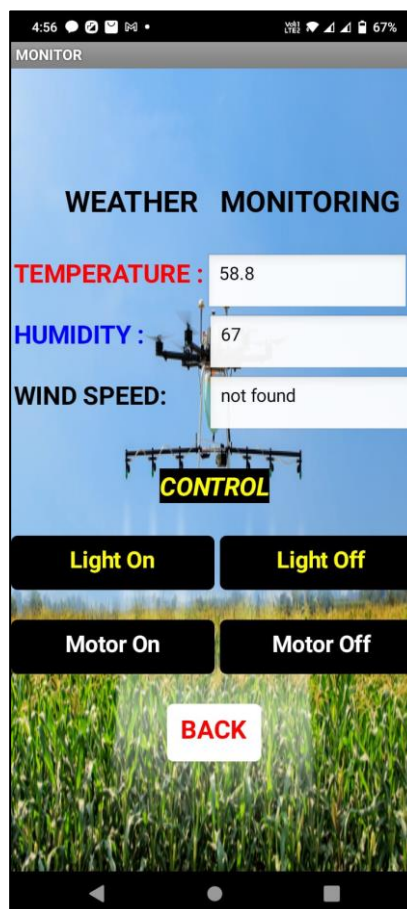


FIG 4: ON CLICKING LIGHT ON BUTTON IN MIT APP THEN LIGHT WILL TURN ON



LIGHT OFF

FIG 5 : TOUCHING LIGHT OFF BUTTON IN MIT APP

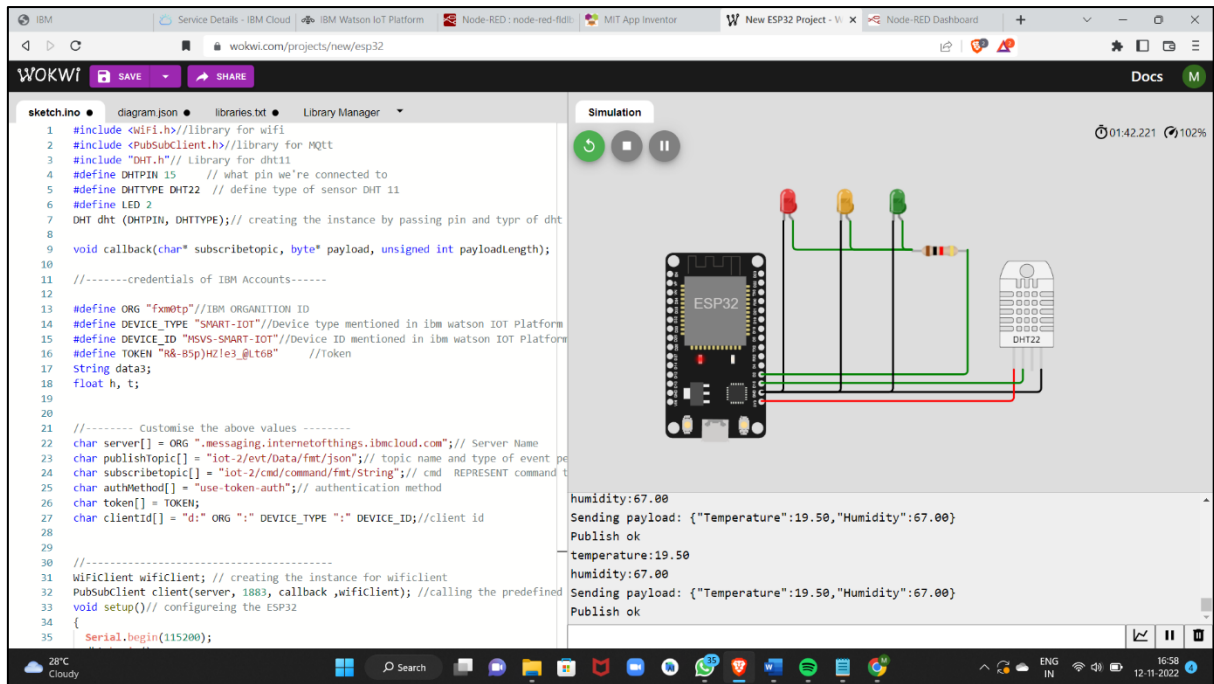


FIG 6: ON CLICKING LIGHT OFF BUTON IN MIT APP THEN LIGHT WILL TURN OFF

ACCESSING TEMPERATURE AND HUMIDITY BY NODE RED WEB UI

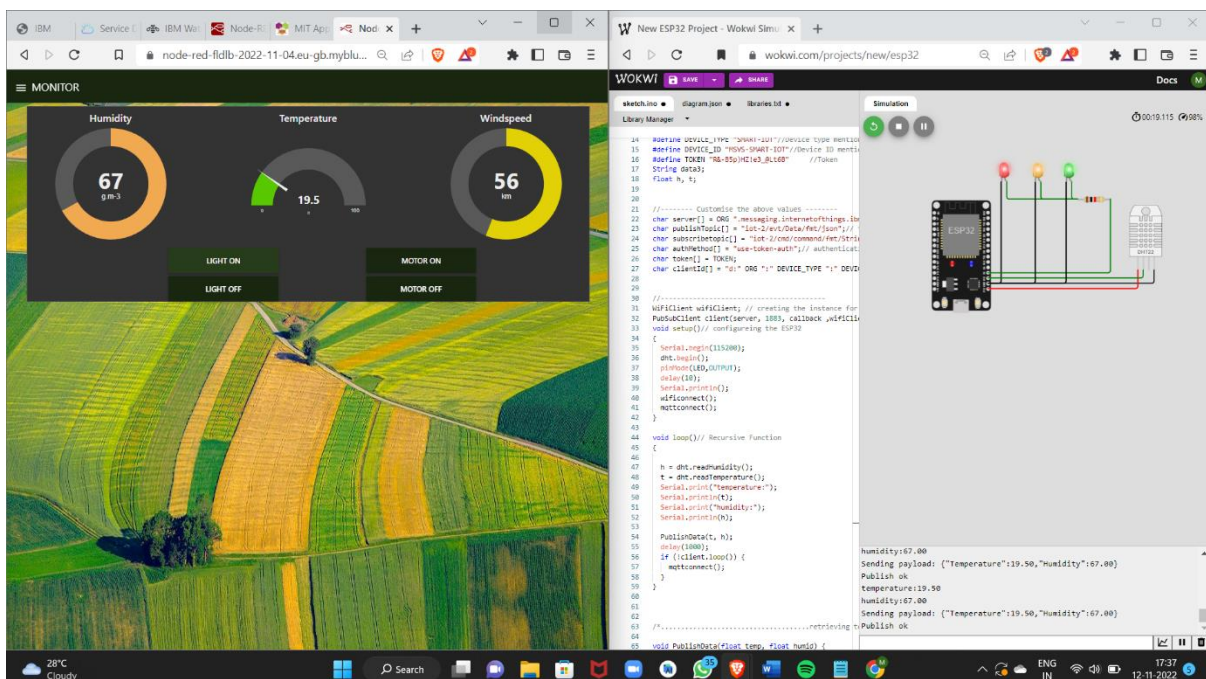


FIG 7: ON CLICKING LIGHT ON BUTON IN NODE RED WEB UI THEN LIGHT WILL TURN ON

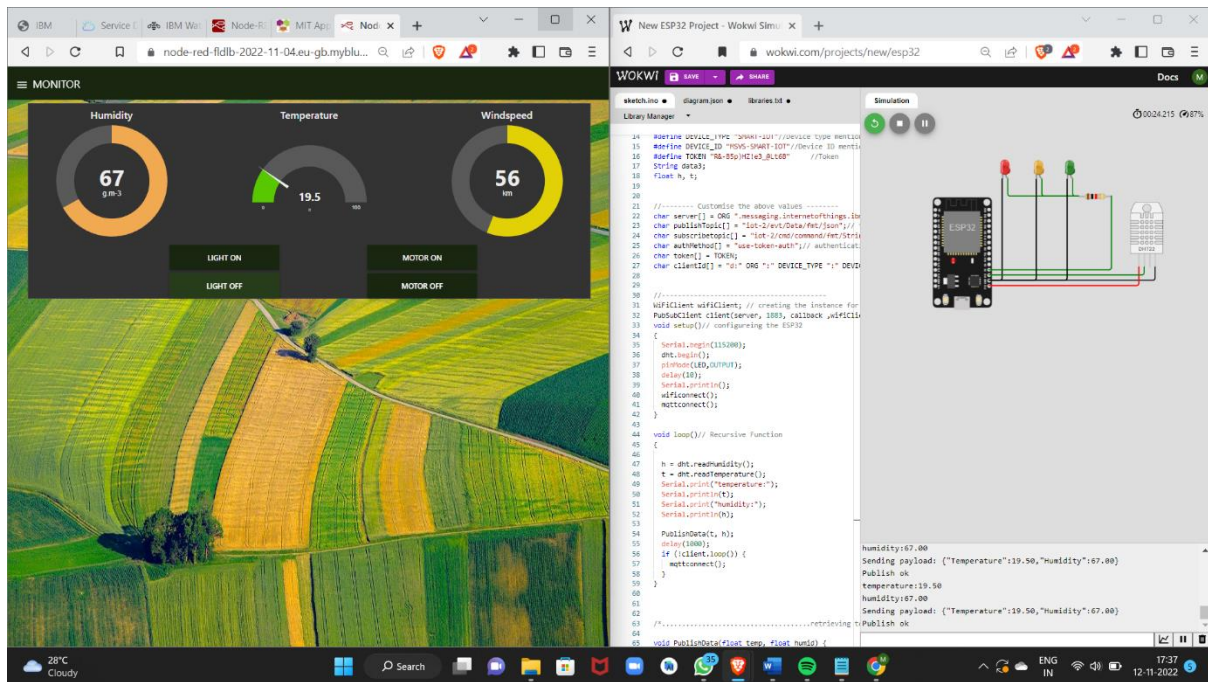


FIG 7: ON CLICKING LIGHT OFF BUTON IN NODE RED WEB UI THEN LIGHT WILL TURN OFF

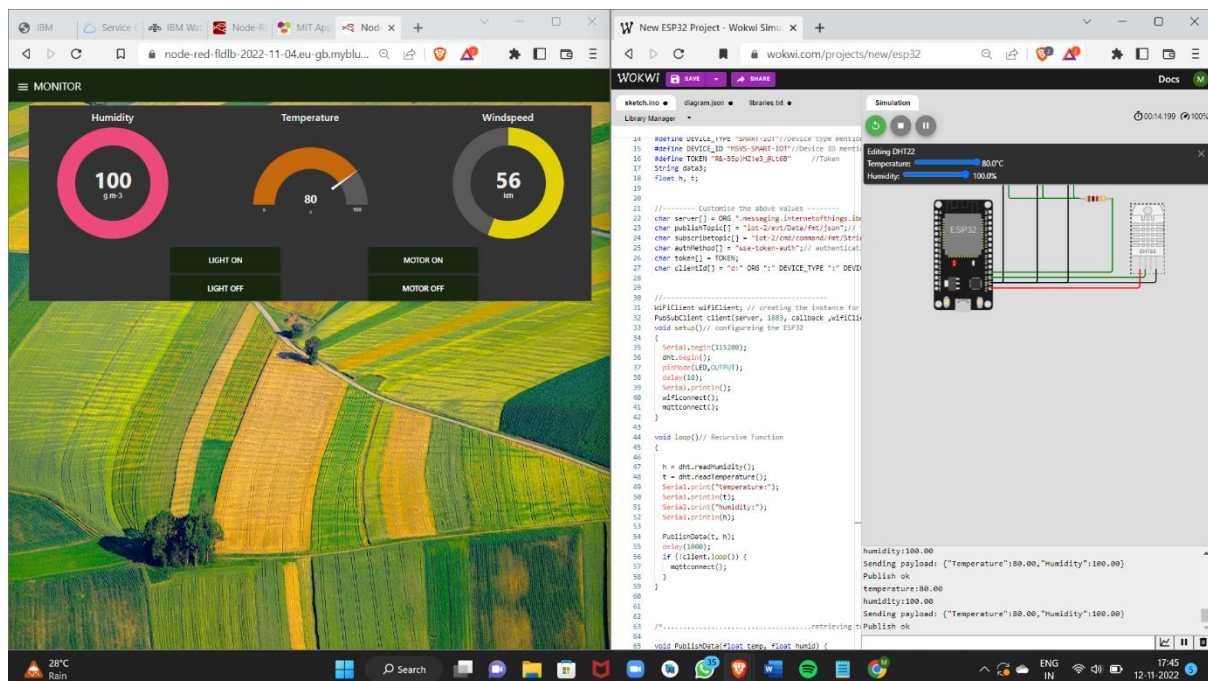


FIG 8:CHANGING TEMPERATURE AND HUMIDITY IN WOKWI TO WEB UI