|  |  |
| --- | --- |
| **TEAM ID** | **PNT2022TMID50597** |
| **PROJECT NAME** | **Smart Waste Management System for Metropolitan Cities** |

CODE FOR DATA TRANSFER FROM SENSORS

#include <WiFi.h> //Library for WiFi #include <PubSubClient.h> //Library for MQTT #include <ArduinoJson.h> //Library for ArduinoJson

WiFiClient wifiClient;

// Credentials on IBM Account-

#define ORG "k6spbs" //IBM Organisation ID

#define DEVICE\_TYPE "MSD" //Device mentioned in IBM Watson IOT Platform #define DEVICE\_ID "12345" //Device ID mentioned on IBM Watson IOT Platform #define TOKEN "123456789" //Token

#define speed 0.034

// Customise above values

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //Server Name char publishTopic[] = "iot-2/evt/Data/fmt/json";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth"; //Authentication Method

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID; //Client id

//

PubSubClient client(server, 1883, wifiClient); void publishData();

const int trigpin=5; const int echopin=18; String command; String data="";

String lat="13.167558"; String lon="80.244510"; String name="point2"; String icon="fa-trash-o"; String color="green"; long duration;

int dist;

void setup()

{

Serial.begin(115200); pinMode(trigpin, OUTPUT); pinMode(echopin, INPUT); wifiConnect(); mqttConnect();

}

void loop() {

publishData(); delay(500);

if (!client.loop()) { mqttConnect();

}

}

// Retrieving to Cloud

void wifiConnect() {

Serial.print("Connecting to "); Serial.print("Wifi"); WiFi.begin("Wokwi-GUEST", "", 6);

while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.print(".");

}

Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());

}

void mqttConnect() {

if (!client.connected()) {

Serial.print("Reconnecting MQTT client to "); Serial.println(server); while (!client.connect(clientId, authMethod, token)) { Serial.print(".");

delay(1000);

}

initManagedDevice(); Serial.println();

}

void initManagedDevice() { if (client.subscribe(topic)) {

Serial.println(client.subscribe(topic)); Serial.println("subscribe to cmd OK");

} else {

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

}

}

// Publish Smart Bin level

void publishData()

{

digitalWrite(trigpin,LOW); digitalWrite(trigpin,HIGH); delayMicroseconds(10); digitalWrite(trigpin,LOW); duration=pulseIn(echopin,HIGH); dist=duration\*speed/2; dist=dist/4;

dist=100-dist; if(dist>80){ icon="fa-trash"; color="red";

}else{

icon="fa-trash-o"; color="green";

DynamicJsonDocument doc(1024); String payload; doc["Name"]=name; doc["Latitude"]=lat; doc["Longitude"]=lon; doc["Icon"]=icon; doc["FillPercent"]=dist; doc["Color"]=color; serializeJson(doc, payload); delay(3000);

// Print on LCD

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

if (client.publish(publishTopic, (char\*) payload.c\_str())) { Serial.println("Publish OK");

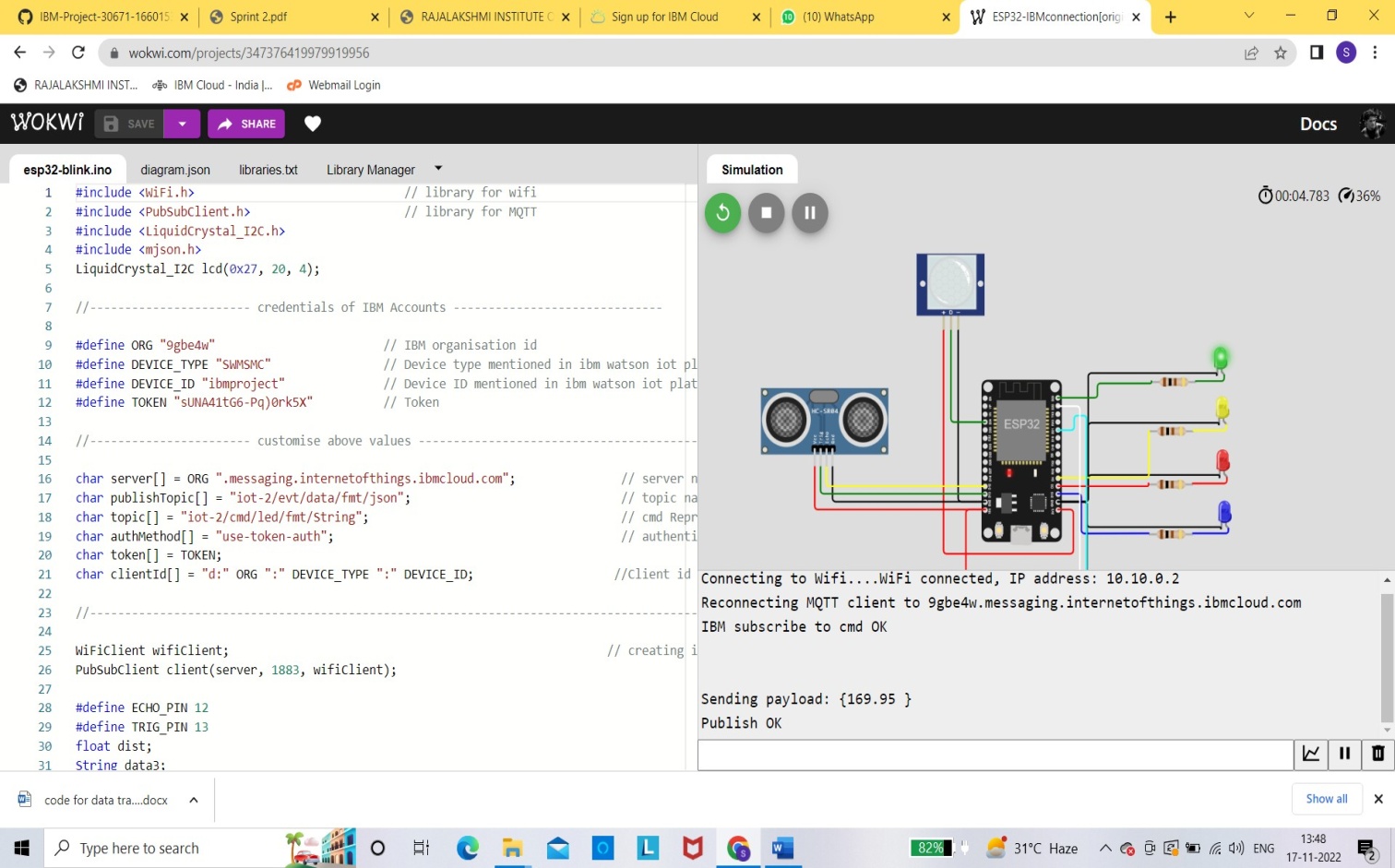
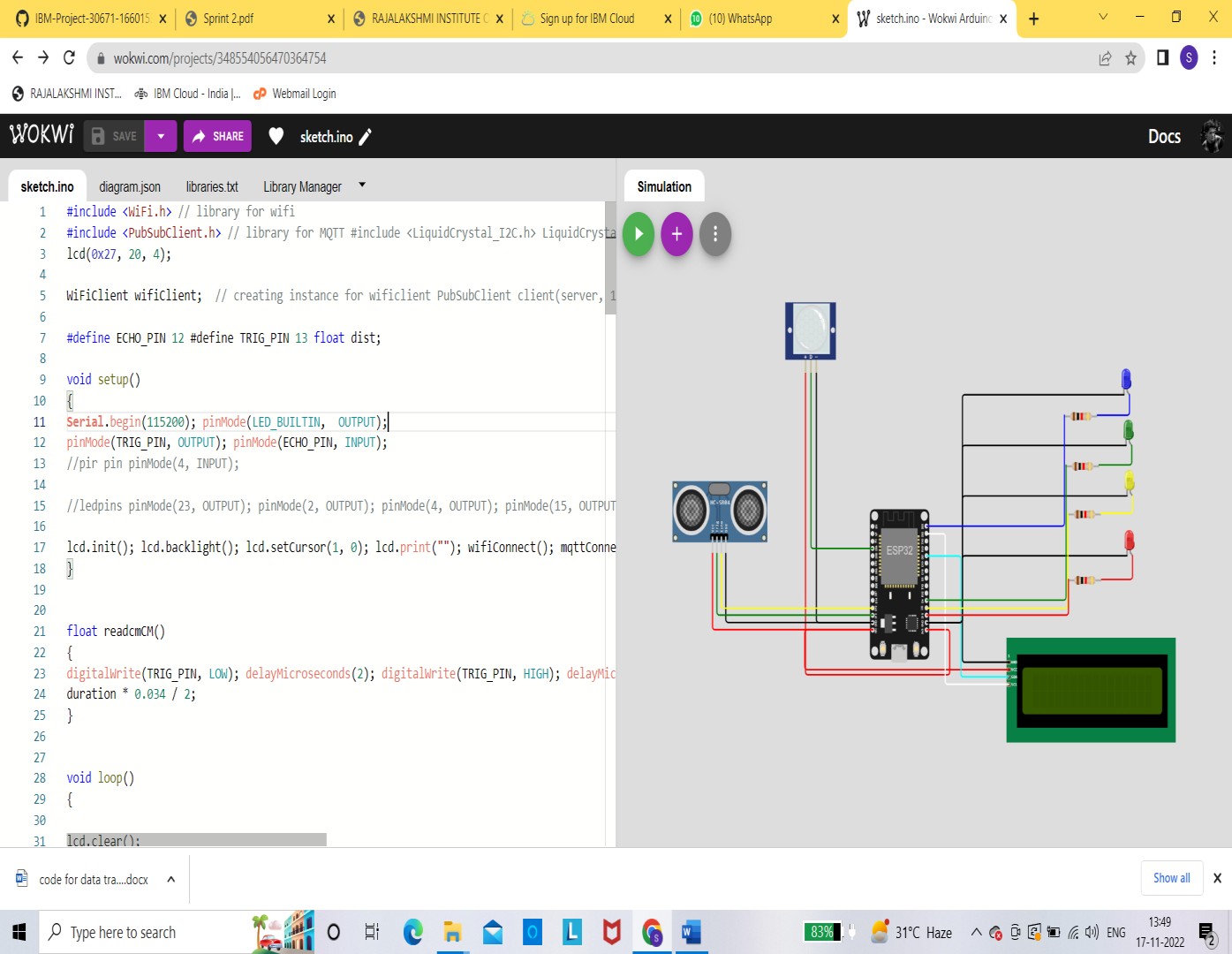
} else {

Serial.println("Publish FAILED");

}

}

// End of Program

CIRCUIT CONFIGURATION: