**ASSIGNMENT – 4**

|  |  |
| --- | --- |
| Date | 22nd October 2022 |
| Name | P.MAHALAKSHMI |
| Project name | Hazardous area monitoring for industrial plant powered by IOT |
| Maximum marks | 2 marks |

#include<WiFi.h>

#include<PubSubClient.h>

voidcallback(char\* subscribetopic,byte\* payload, unsignedintpayloadLength);

#define ORG "arkiki"

#define DEVICE\_TYPE "IOT"

#define DEVICE\_ID "123"

#define TOKEN "12345678"

String data3;

charserver[]= ORG ".messaging.internetofthings.ibmcloud.com";

charpublishTopic[]="iot-2/evt/distance/fmt/json";

charsubscribeTopic[]="iot-2/cmd/test/fmt/String";

charauthMethod[]="use-token-auth";

chartoken[]=TOKEN;

charclientID[]="d:"ORG":"DEVICE\_TYPE":"DEVICE\_ID;

WiFiClientwifiClient;

PubSubClient client(server,1883,callback,wifiClient);

#define ECHO\_PIN 2

#define TRIG\_PIN 4

#define led 5

voidsetup() {

  // put your setup code here, to run once:

**Serial**.begin(115200);

  pinMode(led, OUTPUT);

  pinMode(TRIG\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

  wificonnect();

  mqttconnect();

}

floatreadDistanceCM() {

  digitalWrite(TRIG\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG\_PIN, LOW);

  int duration=random(1,200);

  //Serial.println(duration);

  //duration = pulseIn(ECHO\_PIN, HIGH);

  returnduration ;

  //Serial.println(duration);

}

voidloop() {

  float distance = readDistanceCM();

  //Serial.println(distance);

  boolisNearby = distance <100;

  digitalWrite(led, isNearby);

**Serial**.print("Measured distance: ");

**Serial**.println(distance);

  if(distance<100){

    PublishData2(distance);

  }else{

    PublishData1(distance);

  }

  //PublishData(distance);

  delay(1000);

  if(!client.loop()){

    mqttconnect();

  }

  //delay(2000);

}

void PublishData1(floatdist){

  mqttconnect();

  String payload= "{\"distance\":";

  payload += dist;

  payload+="}";

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

**Serial**.println(payload);

  if(client.publish(publishTopic,(char\*)payload.c\_str())){

**Serial**.println("publish ok");

  } else{

**Serial**.println("publish failed");

  }

}

void PublishData2(floatdist){

  mqttconnect();

  String payload= "{\"ALERT\":";

  payload += dist;

  payload+="}";

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

**Serial**.println(payload);

  if(client.publish(publishTopic,(char\*)payload.c\_str())){

**Serial**.println("publish ok");

  } else{

**Serial**.println("publish failed");

  }

}

voidmqttconnect(){

  if(!client.connected()){

**Serial**.print("Reconnecting to ");

**Serial**.println(server);

    while(!!!client.connect(clientID, authMethod, token)){

**Serial**.print(".");

      delay(500);

    }

    initManagedDevice();

**Serial**.println();

  }

}

voidwificonnect(){

**Serial**.println();

**Serial**.print("Connecting to");

  WiFi.begin("Wokwi-GUEST","",6);

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

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.println("WIFI CONNECTED");

**Serial**.println("IP address:");

**Serial**.println(WiFi.localIP());

}

voidinitManagedDevice(){

  if(client.subscribe(subscribeTopic)){

**Serial**.println((subscribeTopic));

**Serial**.println("subscribe to cmd ok");

  }else{

**Serial**.println("subscribe to cmd failed");

  }

}

voidcallback(char\* subscribeTopic, byte\* payload, unsignedintpayloadLength){

**Serial**.print("callback invoked for topic:");

**Serial**.println(subscribeTopic);

  for(inti=0; i<payloadLength; 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="";

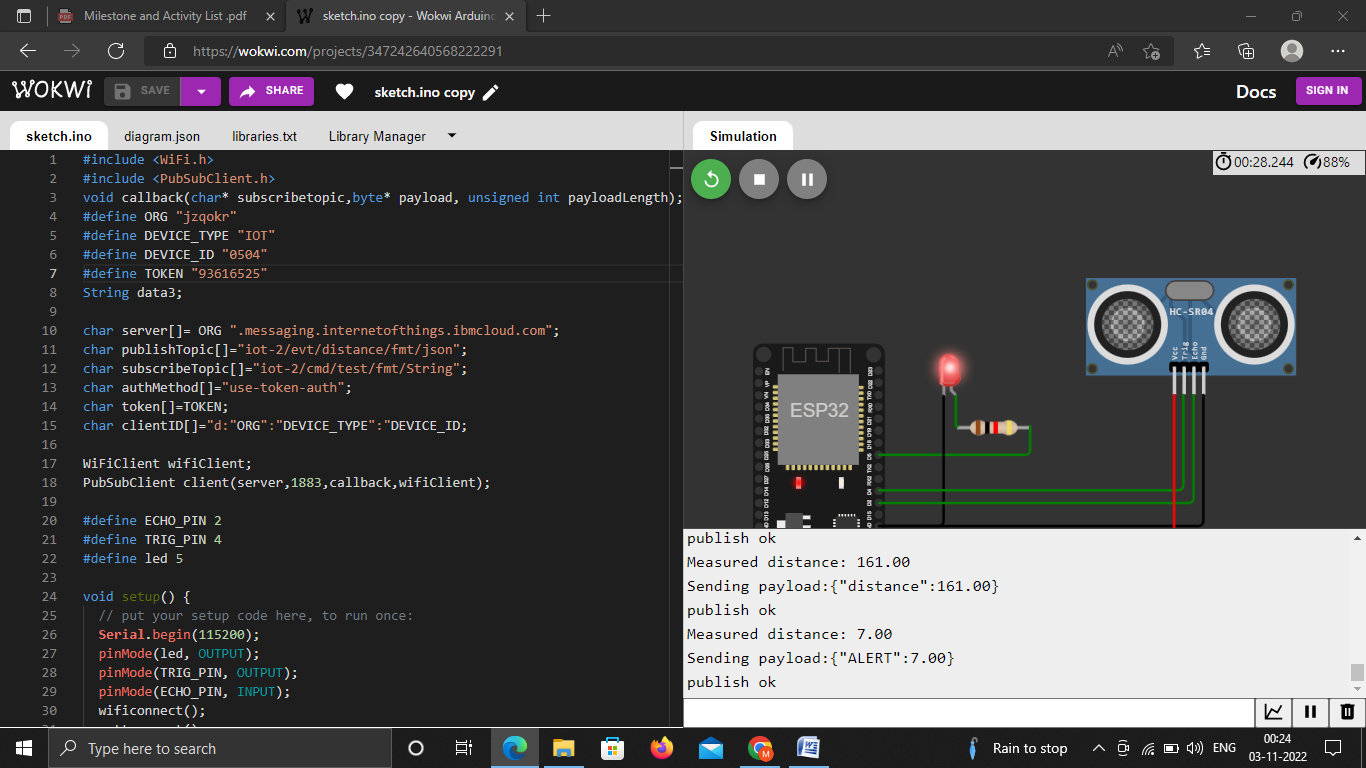
}

231a7359149a59989b7e00cafb0cada6

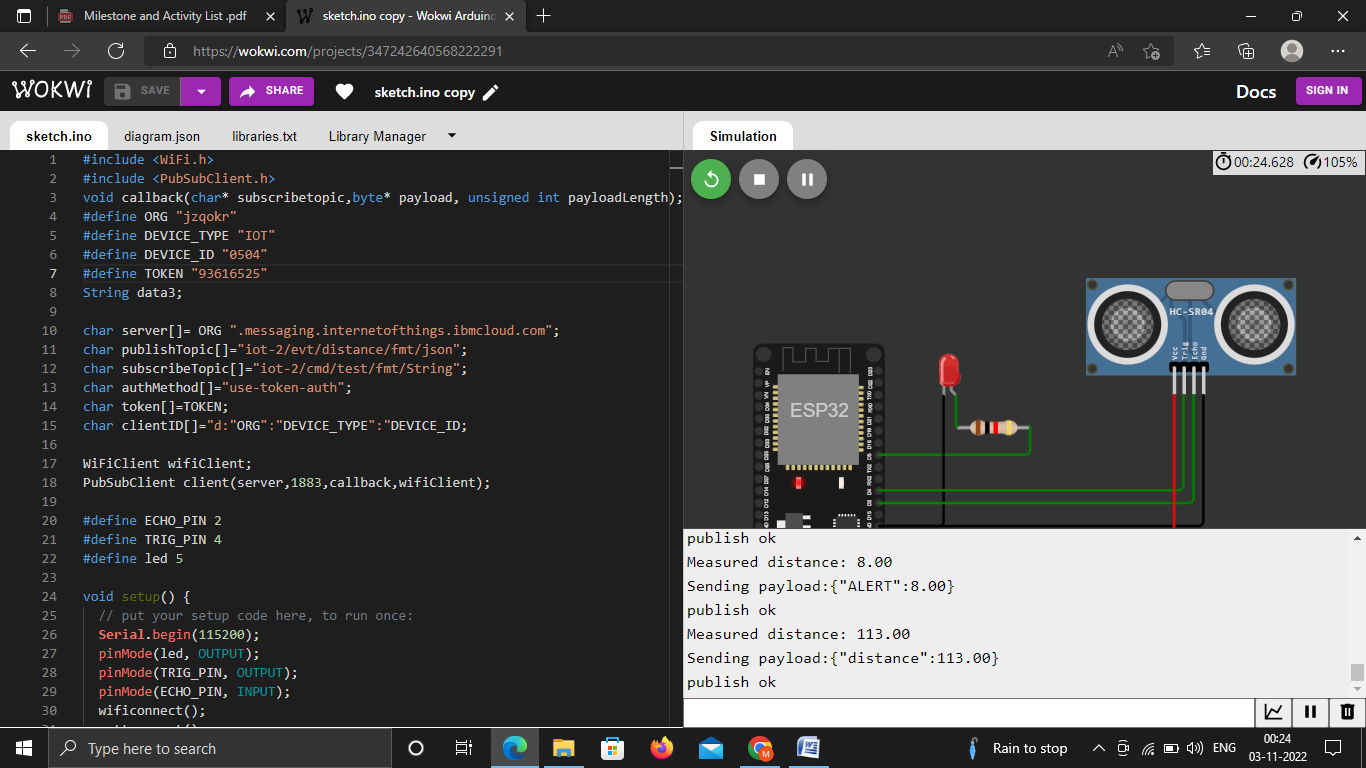
**Wokwi project link:**

https://wokwi.com/projects/347242640568222291

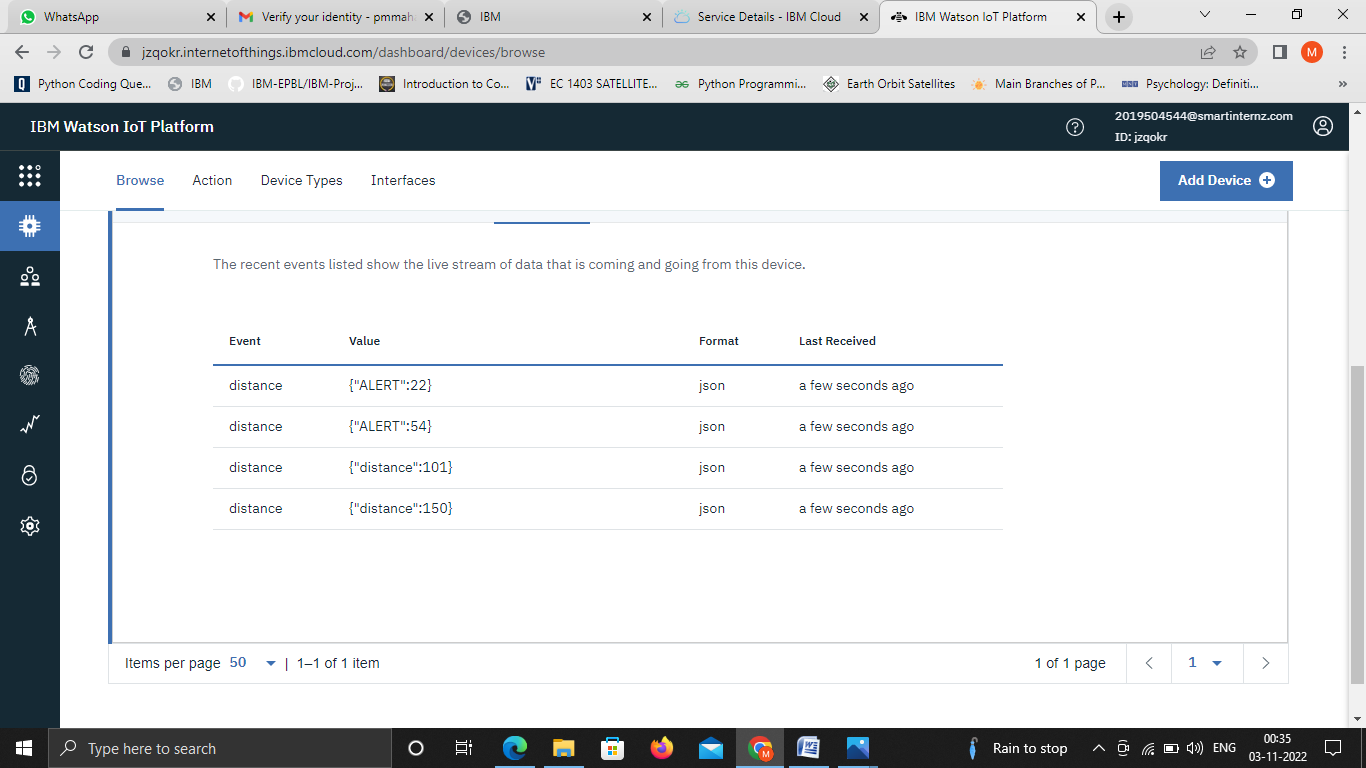
**Alert case:**

****

**Normal case:**

****

**Cloud storage:**

****