**ASSIGNMENT 4**

|  |  |
| --- | --- |
| **TEAM ID** | **PNT2022TMID20364** |

Question 1:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

**Wokwi Project Link:** <https://wokwi.com/projects/348100832051331666>

CODE:

#include<WiFi.h>

#include<PubSubClient.h>

#define TRIGGER 2

#define ECHO 15

#definesound\_speed0.034

int distance;

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

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

#define ORG "0va7j8"

#define DEVICE\_TYPE "ultasonic"

#define DEVICE\_ID "123456"

#define TOKEN "123456789"

String data3;

//-------- Customise the above values --------

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

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

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

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

chartoken[] = TOKEN;

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

//-----------------------------------------

WiFiClientwifiClient;

PubSubClientclient(server, 1883, callback ,wifiClient);

voidsetup()

{

**Serial**.begin(115200);

  pinMode(TRIGGER, OUTPUT);

  pinMode(ECHO, INPUT);

  delay(10);

**Serial**.println();

  wificonnect();

  mqttconnect();

}

voidloop()

{

  digitalWrite(TRIGGER, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIGGER, LOW);

  int duration=pulseIn(ECHO,HIGH);

  distance=(duration\*sound\_speed)/2;

**Serial**.print(distance);

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

  if(distance<100){

  PublishData(distance);

  }

  delay(1000);

  if(!client.loop()) {

    mqttconnect();

  }

}

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

voidPublishData(int d) {

  mqttconnect();

  String payload = "{\"message\":\"alert\"";

  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 client 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++) {

    //Serial.print((char)payload[i]);

    data3 += (char)payload[i];

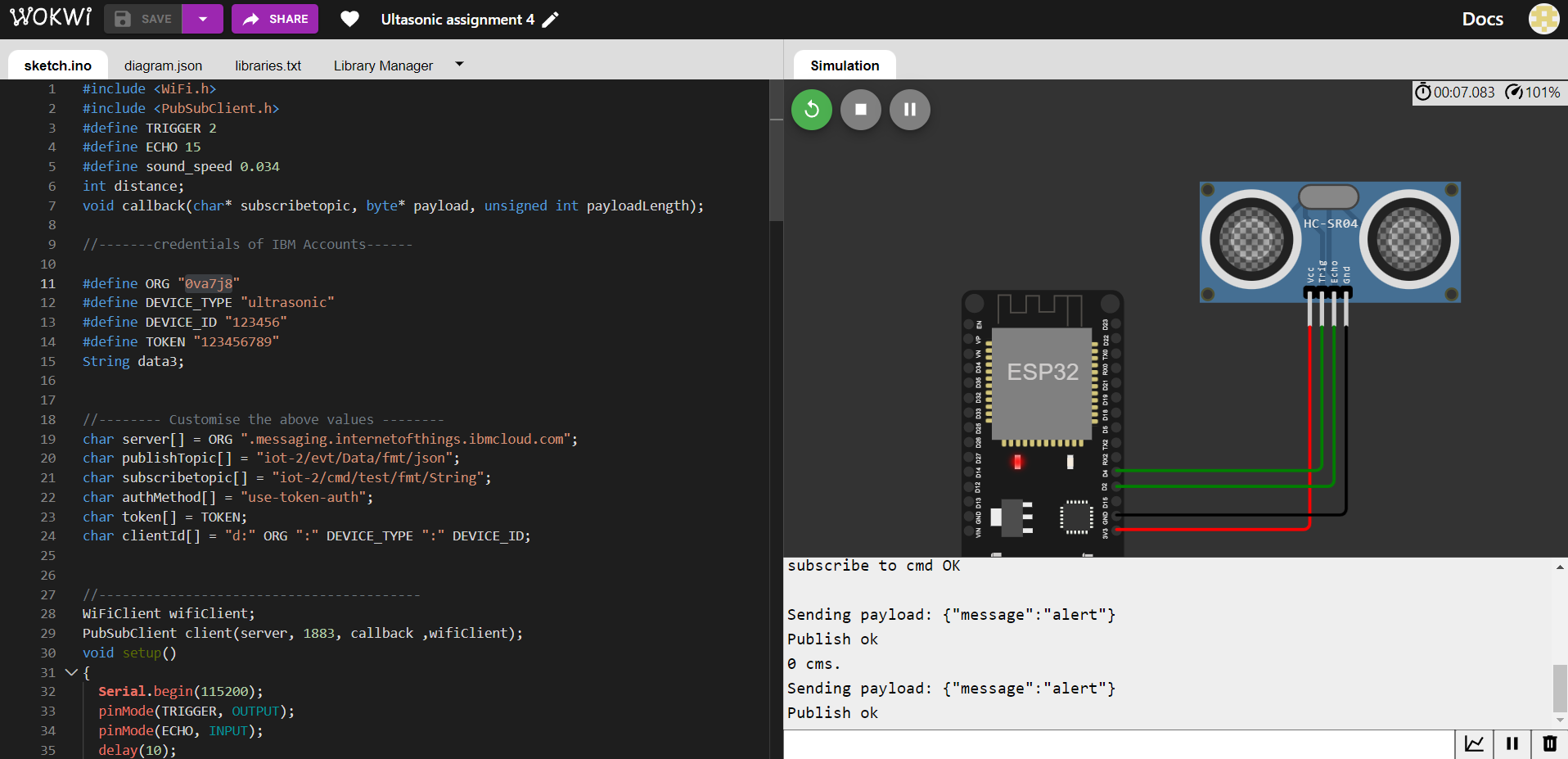
  }

**Serial**.println("data: "+ data3);

data3="";

}

**Wokwi Platform Coding and Circuit Design**



**IBM IoT Platform Device Recent Events**