Assignment -4

Assignment Date	1 November 2022
Student Name	FERSHIA G GEONA
Student Roll Number	962819106014
Maximum Marks	2 Marks

Question-1:

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

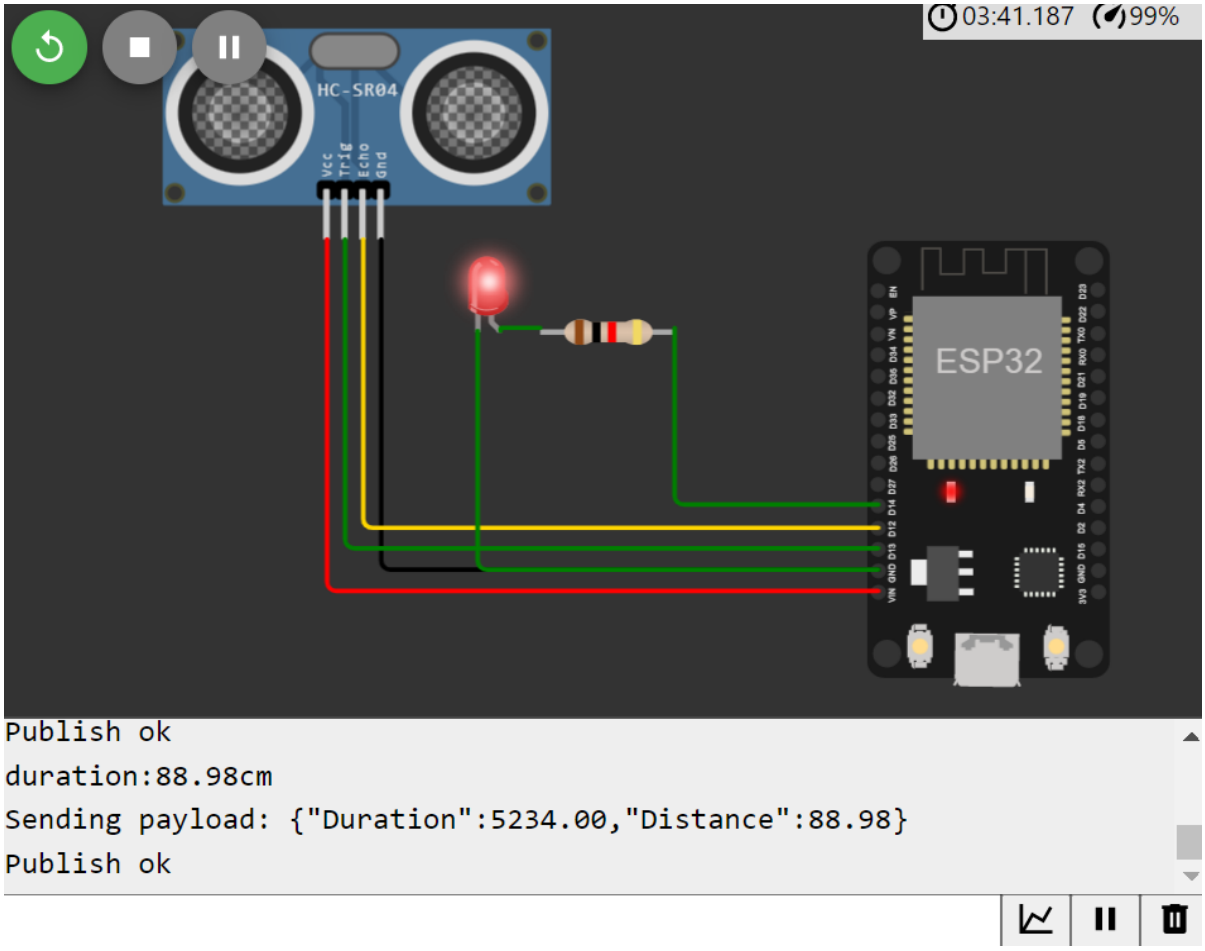
Solution:

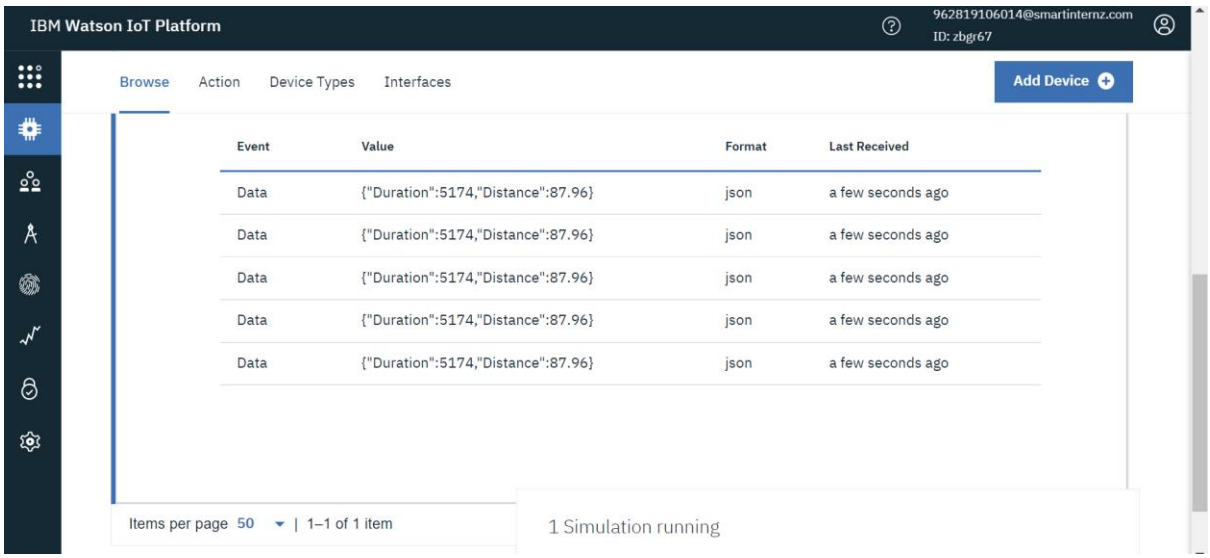
```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#define echoPin 12
#define trigPin 13
#define led1 14
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

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

#define ORG "zbgr67"//IBM ORGANITION ID
#define DEVICE_TYPE "fershidevicetype"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "fershideviceid"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "fershiageona"     //Token
String data3;
float duration,distance;

//-------- 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";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
```

```
//----------------------------------------
void setup()// configureing the ESP32
{
  pinMode(trigPin,OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT);
  pinMode(led1,OUTPUT); // Sets the echoPin as an INPUT
  Serial.begin(115200); // // Serial Communication is starting with 9600 of baudrate speed
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  distance=duration*0.034/2;
  Serial.println("duration:"+String(distance)+ "cm");
  if (distance<=100)
  {
    digitalWrite(led1,HIGH);
    PublishData(duration,distance);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
  else{
    digitalWrite(led1,LOW);
  }

  }
}


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

void PublishData(float dur, float dist)
{
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
```

```
 */
 String payload = "{\"Duration\":";
 payload += dur;
 payload += "," "\"Distance\":";
 payload += dist;
 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(led1,HIGH);

  }

  else
  {
Serial.println(data3);
digitalWrite(led1,LOW);

  }
data3="";
}
```





Wokwi link: https://wokwi.com/projects/347137640781316690