Assignment-4

ASSIGNMENT 4

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 cm send "alert" to IBM cloud and display in device recent events.

Upload document with wokwi share link and images of IBM cloud.

CODE

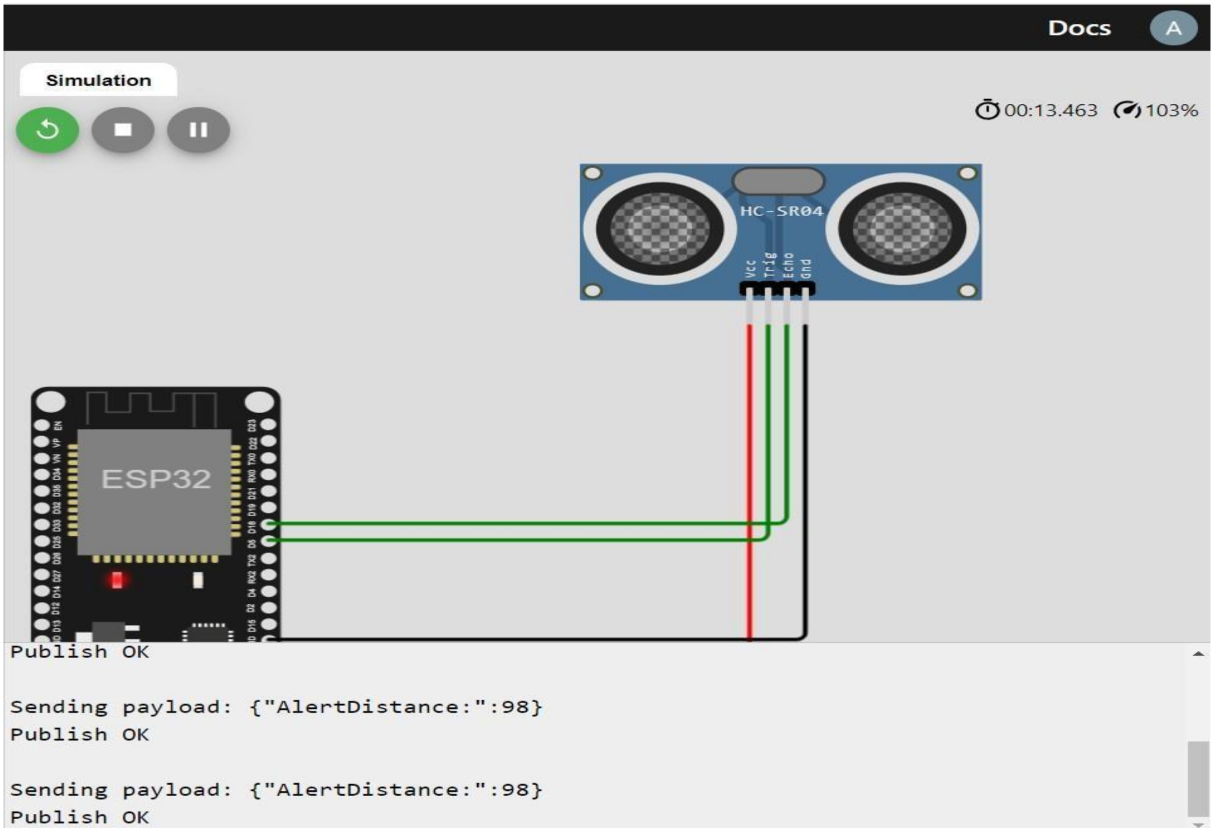
```
#include <WiFi.h>

#include
<PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;

#define ORG "mz6rat"

#define DEVICE_TYPE "arduino"
#define DEVICE_ID "54321"
#define TOKEN "26072002"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();

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

```
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();
  }
}

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");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH)
  ; delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH)
  ; dist=duration*speed/2;

  if(dist<100){
    DynamicJsonDocument
    doc(1024); String payload;
    doc["AlertDistance:"]=dist;
    serializeJson(doc, payload);
    delay(3000);
    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");
    }
  }
}
```

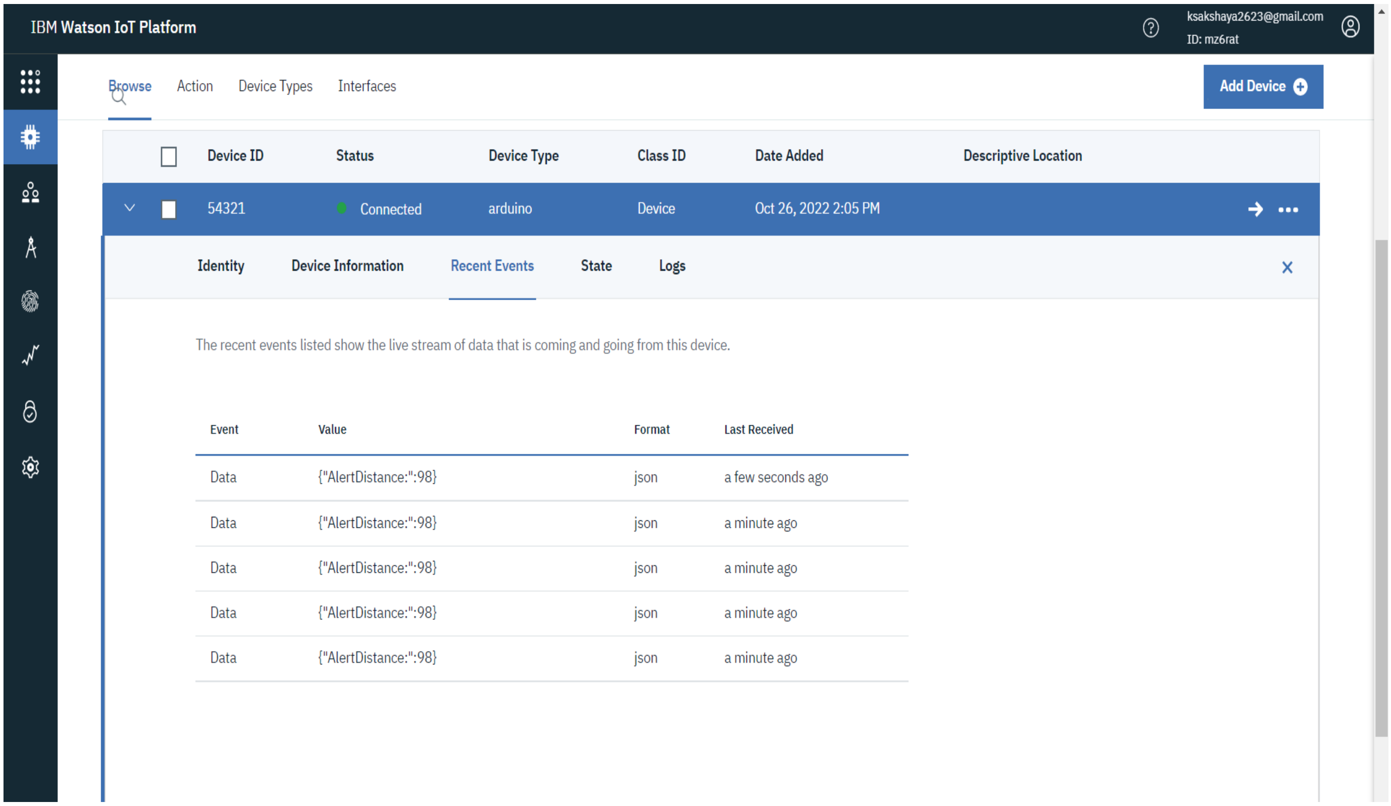
WOKWI LINK:

https://wokwi.com/projects/346571307833360979

OUTPUT:



```
Publish OK

Sending payload: {"AlertDistance:":98}
Publish OK

Sending payload: {"AlertDistance:":98}
Publish OK
```



Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
54321	● Connected	arduino	Device	Oct 26, 2022 2:05 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"AlertDistance:":98}	json	a few seconds ago
Data	{"AlertDistance:":98}	json	a minute ago
Data	{"AlertDistance:":98}	json	a minute ago
Data	{"AlertDistance:":98}	json	a minute ago
Data	{"AlertDistance:":98}	json	a minute ago