Assignment -4

Student Name	Jawaharprasad S
Student Roll Number	714019106036

Question:

Write code and connection 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.

Coding:

Sketch.ino

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "9tm5df"//IBM ORGANITION ID
#define DEVICE_TYPE "device1"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "714019106036"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "714019106036" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
```

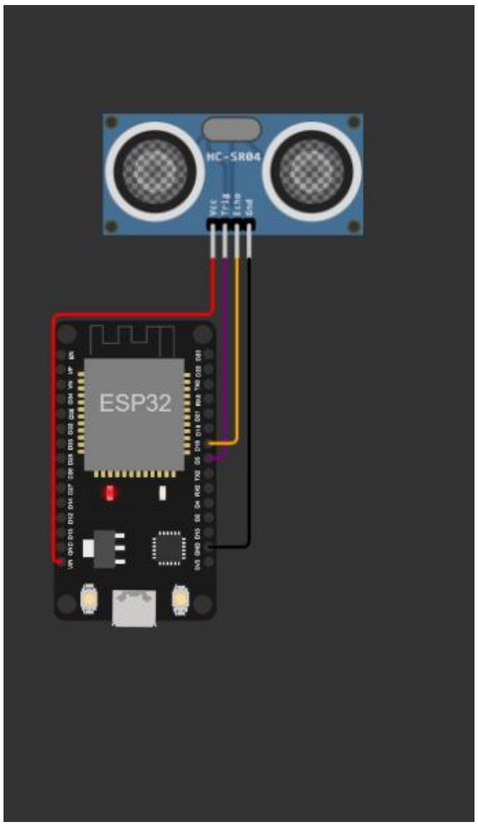
```
distance = duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(float dist) {
mqttconnect();
String payload = "{\"Distance\":";
payload += dist;
payload += ",\"ALERT!!\":""\"Distance less than 100cms\"";
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 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()
{
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());
}
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);
data3="";
}
```

Diagram.json:

```
{
  "version": 1,
  "author": "jawaharrprasad S",
  "editor": "ibmssfr",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 13.19, "left": -
141.14, "attrs": {} },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -119.46, "left": -
109.51, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v36.21", "h-103.59", "v168.16" ]
],
    [ "ultrasonic1:TRIG", "esp:D5", "purple", [ "v0" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "orange", [ "v0" ] ],
    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ]
  ]
}
```

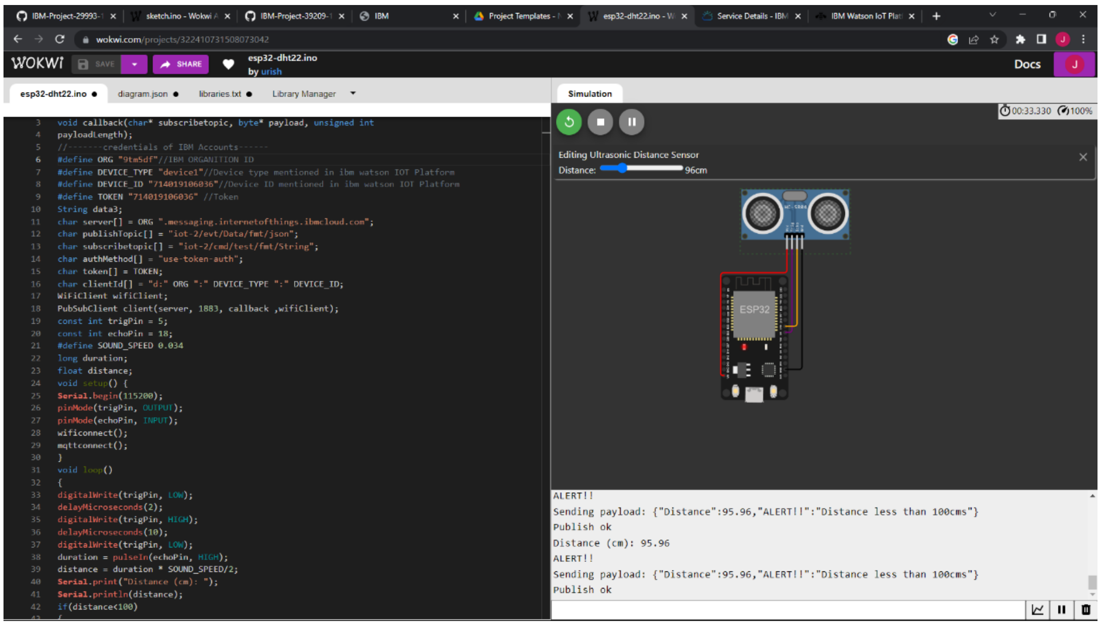
Connection given:



Link for the simulation:

https://wokwi.com/projects/322410731508073042

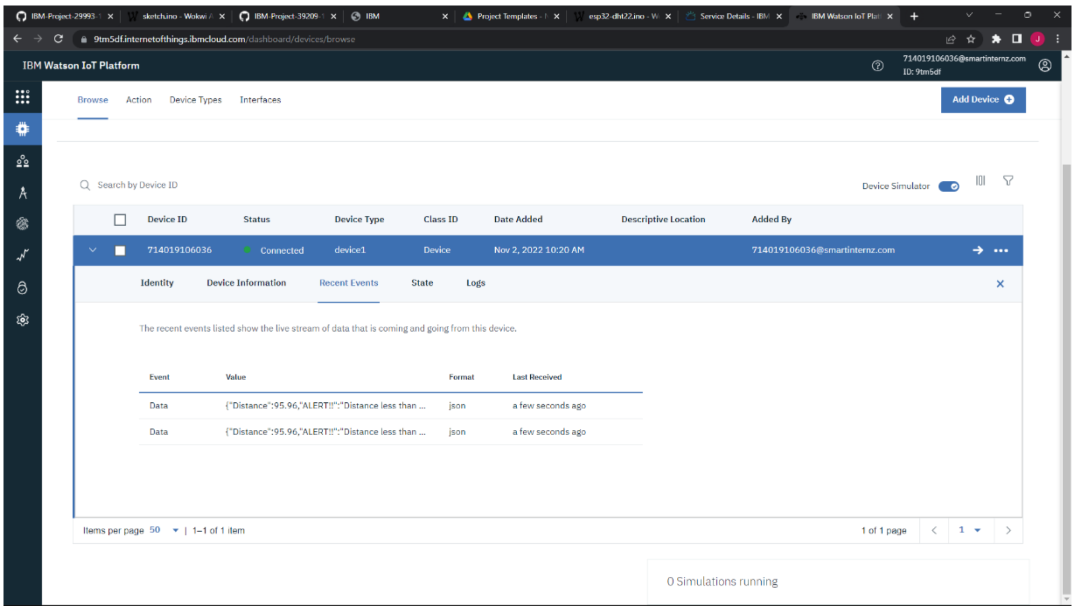
Simulation Output:



```
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "9tm5df"//IBM ORGANITION ID
#define DEVICE_TYPE "device1"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "714019106036"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "714019106036" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
```

```
ALERT!!
Sending payload: {"Distance":95.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 95.96
ALERT!!
Sending payload: {"Distance":95.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
```

IBM cloud Output:



	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
	714019106036	Connected	device1	Device	Nov 2, 2022 10:20 AM		714019106036@smartinternz.com

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

Event	Value	Format	Last Received
Data	{"Distance":95.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":95.96,"ALERT!!":"Distance less than ...	json	a few seconds ago