

#### Assignment -4

Student Name	Jaffer Sathick Hm
Student Roll Number	714019106035

#### 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:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "uqjend"//IBM ORGANITION ID
#define DEVICE_TYPE "Device1"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "714019106035"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "714019106035" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "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()

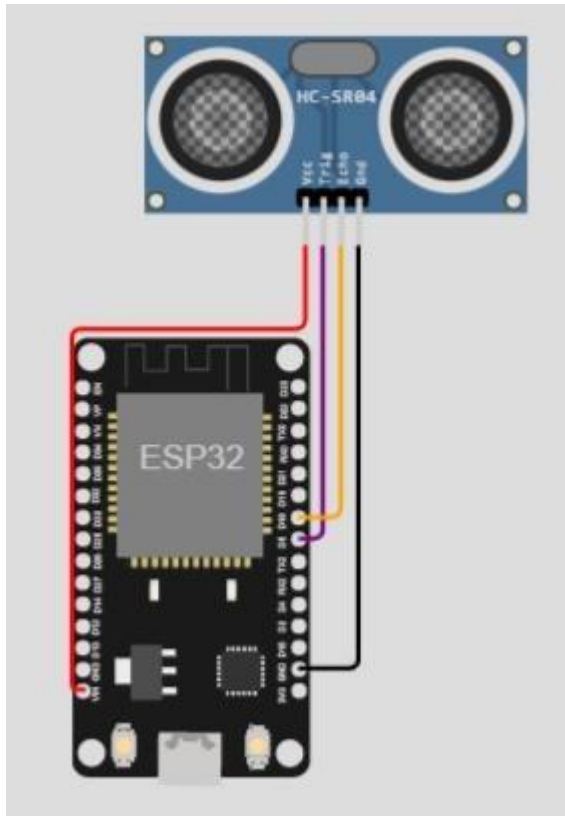
```

```

{
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="";
}

```

Connection given:



Link for the simulation:

<https://wokwi.com/projects/347321279937774164>

Simulation Output:

WOKWI

sketch.ino

diagram.json

libraries.txt

Library Manager

```

4  payloadlength);
5  //-----credentials of IOT Accounts-----
6  #define OMC "uqjend//IBM COGNITION ID"
7  #define DEVICE_TYPE "Device1"//Device type mentioned in the Watson IOT Platform
8  #define DEVICE_ID "714019106035"//Device ID mentioned in the Watson IOT Platform
9  #define TOKEN "714019106035" //Token
10 String data;
11 char server[] = OMC ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/ev/Data/fmt/json";
13 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" OMC ":" DEVICE_TYPE ":" DEVICE_ID;
17 WiFiClient wifiClient;
18 PubSubClient client(server, 1883, callback, wifiClient);
19 const int trigPin = 5;
20 const int echoPin = 18;
21 #define SOUND_SPEED 0.034
22 long duration;
23 float distance;
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wifiConnect();
29   mqttConnect();
30 }
31 void loop() {
32   {
33     digitalWrite(trigPin, LOW);
34     delayMicroseconds(2);
35     digitalWrite(trigPin, HIGH);
36     delayMicroseconds(10);
37     digitalWrite(trigPin, LOW);
38     duration = pulseIn(echoPin, HIGH);
39     distance = duration * SOUND_SPEED/2;
40     Serial.print("Distance (cm): ");
41     Serial.println(distance);
42     if(distance<100)
43     {
44       Serial.println("ALERT!!");
45       delay(1000);
46       PublishData(distance);
47       delay(1000);
48     if (!client.loop()) {

```

Simulation

00:34.531 100%

Editing Ultrasonic Distance Sensor

Distance: 90cm

IBM Watson IoT Platform

Browse Action Device Types Interfaces

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By	Device Class
714019106035	Disconnected	Device1	Device	Nov 3, 2022 2:05 PM		714019106035@smartinternz.com	

Identity Device Information Recent Events State Logs

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

Event	Value	Format	Last Received
event_1	{}	json	a few seconds ago
event_1	{}	json	a few seconds ago
event_1	{}	json	a few seconds ago
event_1	{}	json	a few seconds ago
event_1	{}	json	a few seconds ago

1 Simulation running