

ASSIGNMENT-4

- 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

Submitted by

JEFFRINA

CODE

```
•  
#include <WiFi.h>  
//#include <PubSubClient.h>  
void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);  
#define ORG "46d2e1"//IBM ORGANITION ID  
#define DEVICE_TYPE "87654321"//Device type mentioned in ibm watson IOT  
#define DEVICE_ID "12345678"//Device ID mentioned in ibm watson IOT  
#define TOKEN "123123123" //Token String data3;  
char server[] = ORG ".messaging.internetofthings.ibmcloud.c om";  
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;  
const int ledpin= 12;
```

```
#define SOUND_SPEED 0.034
long duration;
float distance; void setup() {
Serial.begin(115200); pinMode(trigPin, OUTPUT);
pinMode(ledpin, 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)
```

```
{
  digitalWrite(ledpin, HIGH);
  delay(1000);
  Serial.println("ALERT!!"); delay(1000); PublishData(distance); delay(1000);
  /*if (!client.loop()) { mqttconnect();
  }
  */}
else
{
  digitalWrite(ledpin, LOW);
  delay(1000);
}
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);
}
```

OUTPUT

•
•

The screenshot displays the Wokwi IoT simulator interface. At the top, the browser address bar shows the project URL: wokwi.com/projects/347659683499082322. The project title is "esp32-dht22.ino copy". The simulation is running, as indicated by the "Simulation" tab and the "03:31.613 78%" timer.

The circuit diagram shows an ESP32 microcontroller connected to an ultrasonic distance sensor and an LED. The sensor's output is connected to the ESP32's GPIO pins, which are also connected to the LED. The sensor's distance is displayed as 61cm in the "Editing Ultrasonic Distance Sensor" window.

The console output shows the following sequence of events:

```
1 Distance (cm): 317.97
1 Distance (cm): 317.97
1 Distance (cm): 317.97
2 Distance (cm): 317.97
2 Distance (cm): 94.94
2 ALERT!!
2 Sending payload: {"Distance":94.94,"ALERT!!":"Distance less than 100cms"}
```

The bottom of the screen shows the Windows taskbar with the system clock at 16:16 on 07-11-2022, and the weather as 30°C Cloudy.

Simulation

03:23.764 98%

Editing Ultrasonic Distance Sensor

Distance: 318cm

```
1 Distance (cm): 317.97
1 Distance (cm): 317.97
2 Distance (cm): 317.97
2 Distance (cm): 317.97
2 Distance (cm): 317.97
2 Distance (cm): 317.97
2 Distance (cm): 317.97
```

wokwi

link:

<https://wokwi.com/projects/34765968349908232>

2