

**V.S.B. Engineering College**  
**Department of Computer Science and Engineering**  
**Assignment-4**

**NAME: DHASARATH KP**

**Assignment :**

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

**Solution:**

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

#define ORG "za7x6f"//IBM ORGANITION ID

#define DEVICE_TYPE "rj46"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "raj46"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "R0Q4uhcOCCD0hnom)K"

String data3; float dist;

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;//client id

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback ,wifiClient);
```

```

int LED = 4; int
trig = 5;
int echo = 18; void
setup()
{
  Serial.begin(115200);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT); pinMode(LED,
  OUTPUT);
  delay(10); wificonnect(); mqttconnect();
}
void loop()
{
  digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  float dur = pulseIn(echo,HIGH);
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distancein cm");
  Serial.println(dist);
  PublishData(dist);
  delay(1000);
  if(!client.loop()) {
    mqttconnect();
  }
}
void PublishData(float dist) { mqttconnect();
  String object;
  if(dist <100)
  {

```

```

    digitalWrite(LED,HIGH);
Serial.println("object is near");    object
= "Near";
}
else
{
    digitalWrite(LED,LOW);
Serial.println("no object found");    object
= "No";
}

String payload = "{\"distance\":\"";
payload += dist;  payload += ","
"\"object\":\"\""; payload += object;
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];

}

data3="";

}

```

Reference:

<https://wokwi.com/projects/347322163482591827>

