

ASSIGNMENT 4

Assignment Date	29-10-2022
Student Name	SURENDHAR S
Team ID	PNT2022TMID17383
Project Name	Smart Farmer – IoT Smart farming Application
Maximum marks	2 Marks

Write a code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100cms send “Alert” to IBM cloud and display in the device recent events. Upload document with wokwi share link and images of cloud.

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient; String
data3;
#define ORG "s8ov1q"
#define DEVICE_TYPE "gayathri"
#define DEVICE_ID "gayathri123"
#define TOKEN "1234"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/Gayathri/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; float
dist;
```

```

void setup()
{
    Serial.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect(); mqttConnect();
}

void loop() {
    bool isNearby = dist < 100; digitalWrite(led,
    isNearby);

    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(500);
        }

        initManagedDevice();

        Serial.println();
    }
}

void initManagedDevice() {
    if (client.subscribe(topic)) {
        // Serial.println(client.subscribe(topic));
        Serial.println("IBM 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){
        String payload = "{\"Alert Distance\":\""; payload
        += dist;
        payload += "}";

        Serial.print("\n"); Serial.print("Sending
        payload: "); Serial.println(payload);
        if(client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Warning crosses 110cm -- it automatically of the loop");
            digitalWrite(led,HIGH);}

    }

    if(dist>101 && dist<111){
        String payload = "{\"Normal Distance\":\""; payload
        += dist;
        payload += "}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);

    }

}

```

```

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++){
        dist += (char)payload[i];
    }
    Serial.println("data:" + data3);
    if(data3=="lighton"){
        Serial.println(data3);
        digitalWrite(led,HIGH);
    }
}

```

Identity	Device Information	Recent Events	State	Logs
NalaiyaThiran	Disconnected	NalaiyaThiran	Device	Oct 28, 2022 9:46 AM
The recent events listed show the live stream of data that is coming and going from this device.				
Event	Value	Format	Last Received	
Data	{"Alert":"ON","Distance_cm":67.95}	json	a few seconds ago	
Data	{"Alert":"ON","Distance_cm":67.95}	json	a few seconds ago	
Data	{"Alert":"ON","Distance_cm":67.95}	json	a few seconds ago	
Data	{"Alert":"ON","Distance_cm":67.97}	json	a few seconds ago	
Data	{"Alert":"ON","Distance_cm":67.95}	json	a few seconds ago	

```

}
data3="";
}

```

7:09

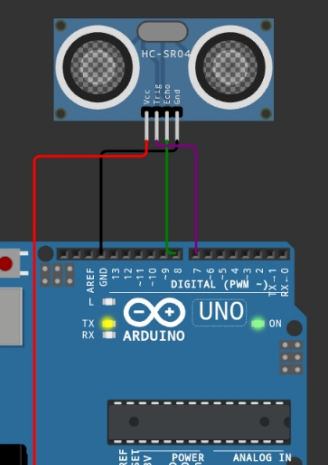
wokwi.com/projects/34

WOKWI

Simulation

Code

00:00.866 23%



182.55 cm	71.54 in
182.55 cm	71.54 in
182.55 cm	71.54 in
182.62 cm	71.57 in
182.55 cm	71.54 in
182.55 cm	71.54 in
182.55 cm	71.54 in

7:10

0.19 KB/S

Vo LTE

4G

75

wokwi.com/projects/34

WOKWI

Simulation

Code

hc-sr04.ino

diagram.json

Library Manager

```

35 digitalWrite(TRIG_PIN, LOW);
36
37 //Set Echo pin as input to measure
38 //pulses coming back from the sensor
39 pinMode(ECHO_PIN, INPUT);
40
41 // We'll use the serial monitor to
42 Serial.begin(9600);
43 }
44
45 void loop() {
46
47     unsigned long t1;
48     unsigned long t2;
49     unsigned long pulse_width;
50     float cm;
51     float inches;
52
53     // Hold the trigger pin high for 10 microseconds
54     digitalWrite(TRIG_PIN, HIGH);
55     delayMicroseconds(10);
56     digitalWrite(TRIG_PIN, LOW);
57
58     // Wait for pulse on echo pin
59     while ( digitalRead(ECHO_PIN) == LOW ) {}
60
61     // Measure how long the echo pin was high
62     // Note: the micros() counter will start at 0
63     t1 = micros();
64     while ( digitalRead(ECHO_PIN) == HIGH ) {}
65     t2 = micros();
66     pulse_width = t2 - t1;
67

```