

## Assignment 4

Date	03 November 2022
Name	Ashwath R
Team ID	PNT2022TMID43379
Project Name	River Water Quality Monitoring and Control System
Maximum Marks	4 Marks

### Program:

```
#include <WiFi.h>
#include<PubSubClient.h>

void callback(char* subscribetopic,byte* payload,unsigned int
payloadLength);
#define ORG "rj0qwb"
#define DEVICE_TYPE "RivWatQuality"
#define DEVICE_ID "RivWatQuality"
#define TOKEN "UFT_PB+dHA3k)0_pA7"
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701

const int trigPin = 5;
const int echoPin = 18;

long duration;
float distanceCm;
float distanceInch;
String data;

char server[]=ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/RivWatQuality/fmt/json";
char subscribeTopic[] = "iot-2/cmd/home/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);

void setup() {
    Serial.begin(115200); // Starts the serial communication
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    wificonnect();
    mqttconnect();
}

void loop() {
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in
microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance
    distanceCm = duration * SOUND_SPEED/2;

    // Convert to inches
    distanceInch = distanceCm * CM_TO_INCH;

    // Prints the distance in the Serial Monitor
    Serial.print("Distance (cm): ");
    Serial.println(distanceCm);

    delay(1000);

    PublishData(distanceCm);
    delay(1000);
    if(!client.loop())
    {
        mqttconnect();
    }
}

```

```

}

void PublishData(float distanceCm)
{

    mqttconnect ();
    String payload;

    if(distanceCm<100.0)
    {
        payload = "{\"Alert\":\"";
        payload += distanceCm;
        payload += "\"}";
    }
    else
    {
        payload = "{\"distanceCm\":\"";
        payload += distanceCm;
        payload += "\"}";
    }

    Serial.print("Sending payload: ");
    Serial.print(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++)
    {
        data +=(char)payload[i];
    }
    Serial.println("data: "+data);
    if(data>(char)100)
    {
        Serial.println("Alert!");
    }*/
}

```

## Wokwi Platform:

**WOKWI** SAVE SHARE Assignment 4 Docs

sketch.ino diagram.json Ultrasonic.h libraries.txt Ultrasonic.cpp Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4
5 void callback(char* subscribetopic,byte* payload,unsigned int payloadLength);
6 #define ORG "rj0qwb"
7 #define DEVICE_TYPE "RivWatQuality"
8 #define DEVICE_ID "RivWatQuality"
9 #define TOKEN "Urt_PB+dhA3k0_pA7"
10 #define SOUND_SPEED 0.034
11 #define CM_TO_INCH 0.393701
12
13 const int trigPin = 5;
14 const int echoPin = 18;
15
16 long duration;
17 float distanceCm;
18 float distanceInch;
19 String data;
20
21 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
22 char publishTopic[] = "iot-2/evt/RivWatQuality/fmt/json";
23 char subscribetopic[] = "iot-2/cmd/home/fmt/String";
24 char authMethod[] = "use-token-auth";
25 char token[] = TOKEN;
26 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
27
28 WiFiClient wifiClient;
29 PubSubClient client(server,1883,callback,wifiClient);
30
31 void setup() {
32     Serial.begin(115200); // Starts the serial communication
33     pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

```

**Simulation** 00:08.855 99%

Distance (cm): 400.01  
 Sending payload: {"distanceCm":400.01}Publish ok  
 Distance (cm): 399.94  
 Sending payload: {"distanceCm":399.94}Publish ok  
 Distance (cm): 399.94  
 Sending payload: {"distanceCm":399.94}Publish ok

# IBM Watson Platform Cloud

IBM Watson IoT Platform

19nc127@pimgtech.ac.in  
ID: r0Qweb

← Back

Device Drilldown - RivWatQuality

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Recent Events

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

Event	Value	Format	Last Received
RivWatQuality	{"distanceCm":399.96}	json	a few seconds ago
RivWatQuality	{"distanceCm":399.92}	json	a few seconds ago

State

This table shows a list of data points that are reported by this device.

Showing Raw Data

No Interfaces Available

Property	Value	Type	Event	Last Received
distanceCm	399.96	Number	RivWatQuality	a few seconds ago

0 Simulations running