

## Assignment -4

### Python Programming

Assignment Date	10-Oct-2022
Student Name	MOHAMMED SAJID . F
Student Roll Number	AC19UIT055
Maximum Marks	2 Marks

#### Question-1:

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

Program:

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

WiFiClient wifiClient;

#define ORG "j12rst"
#define DEVICE_TYPE "esp32"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd_1/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="";
String lat="14.167589";
String lon="80.248510";
String name="point2";
String icon="";

long duration;
int dist;

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

```

    mqttConnect();
}

void loop() {

    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(1000);
        }
        initManagedDevice();
        Serial.println();
    }
}

void initManagedDevice() {
    if (client.subscribe(topic)) {
        Serial.println(client.subscribe(topic));
        Serial.println("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){
        dist=100-dist;
        icon="fa-trash";
    }else{
        dist=0;
        icon="fa-trash-o";
    }
}

```

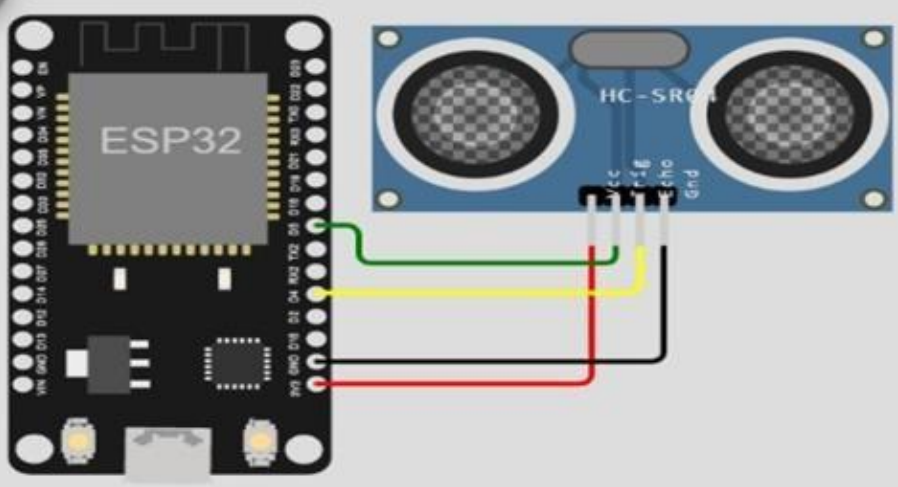
```

}
DynamicJsonDocument doc(1024);
String payload;
doc["Name"]=name;
doc["Latitude"]=lat;
doc["Longitude"]=lon;
doc["Icon"]=icon;
doc["FillPercent"]=dist;
serializeJson(doc, payload);
delay(3000);
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
} else {
    Serial.println("Publish FAILED");
}
}
}

```

## Output:

**Simulation**



Publish ok  
 Measured distance: 18.94  
 Sending payload: {"ALERT\_MESSAGE":1,"DISTANCE":18.94}  
 Publish ok  
 Measured distance: 18.94  
 Sending payload: {"ALERT\_MESSAGE":1,"DISTANCE":18.94}  
 Publish ok

The screenshot displays the IBM Watson IoT Platform interface. At the top, the breadcrumb navigation shows 'Browse' > 'Action' > 'Device Types' > 'Interfaces'. A search bar is present above the main content area. The main content area is divided into two sections. The top section is a table listing devices, with columns for Device ID, Status, Device Type, Class ID, and Date Added. The bottom section is a detailed view for a specific device, showing tabs for Identity, Device Information, Groups, Recent Events, State, and Logs. The 'Recent Events' tab is active, displaying a list of events with columns for Event, Value, Format, and Last Received.

Device ID	Status	Device Type	Class ID	Date Added
12345	Disconnected	esp32	Device	Nov 8, 2022 1:20 PM

Event	Value	Format	Last Received
Distance	{"distance":123}	json	a few seconds ago
Distance	{"distance":116}	json	a few seconds ago
Distance	{"distance":193}	json	a few seconds ago
Distance	{"distance":66}	json	a few seconds ago
Distance	{"distance":85}	json	a few seconds ago