

#### ASSIGNMENT-4

Assignment Date	29 october 2022
Student Name	AJAY K
Student Roll Number	711619104701
Maximum Marks	2 Marks

#### QUESTION:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events. Upload document with wokwi share link and images of ibm cloud.

#### SOLUTION:

##### PROGRAM:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "bjj5tn"
#define DEVICE_TYPE "FIRE"
#define DEVICE_ID "WATER"
#define TOKEN "16431653"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/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 = "{\"Normal Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Publish OK");
        }
    }
    if(dist>101 && dist<111){
        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);
        }else {
            Serial.println("Publish FAILED");
        }
    }
}

```

```

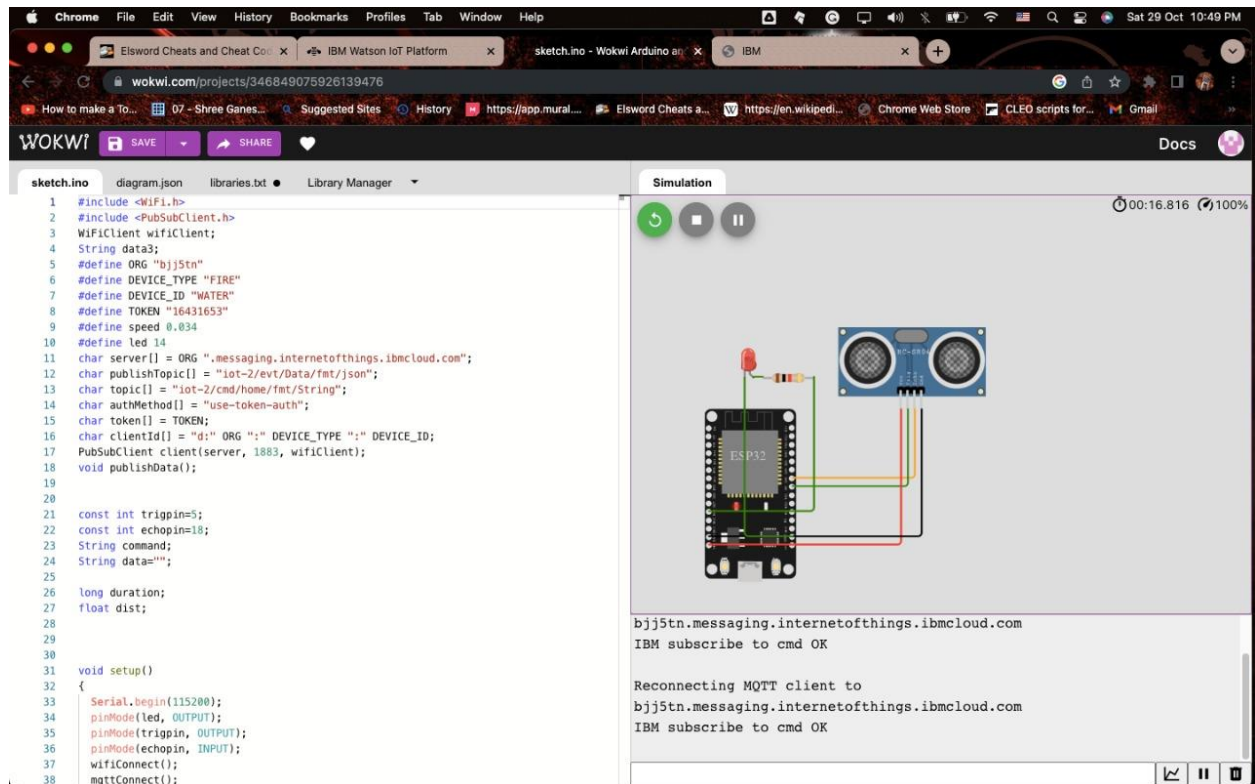
}
void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
  Serial.print("callback invoked for topic:");
  Serial.println(subscribeTopic);
  for(int i=0; i<payloadLength; i++){
    dist += (char)payload[i];
  }
  Serial.println("data:"+ data3);
  if(data3=="lighton"){
    Serial.println(data3);
    digitalWrite(led,HIGH);
  }
  data3="";
}
}

```

## SIMULATION SCREENSHOTS:

The screenshot displays the IBM Watson IoT Platform interface in a web browser. The main dashboard shows a list of devices under the 'Browse' tab. Three devices are listed: ABCD\_1 (Disconnected), AJAY (Disconnected), and WATER (Connected). The 'WATER' device is selected, and its details are shown in a modal window. The modal has tabs for Identity, Device Information, Recent Events, State, and Logs. The 'Recent Events' tab is active, showing a table of events. The table has columns for Event, Value, Format, and Last Received. The events listed are all of type 'Data' with a value of '{"Normal Distance":60.94}' in json format, received a few seconds ago. At the bottom of the modal, it states '0 Simulations running'.

Event	Value	Format	Last Received
Data	{"Normal Distance":60.94}	json	a few seconds ago
Data	{"Normal Distance":60.94}	json	a few seconds ago
Data	{"Normal Distance":60.94}	json	a few seconds ago
Data	{"Normal Distance":60.94}	json	a few seconds ago
Data	{"Normal Distance":60.94}	json	a few seconds ago



**WOKWI LINK:**

<https://wokwi.com/projects/346849075926139476>