

Assignment - 4
Wokwi & IBM Cloud

Assignment Date	04 October 2022
Student Name	R. S. R. RAMESH
Student Roll Number	910619106042
Maximum Marks	2 Marks

Question-1:

Write code and connections in wowki for the ultrasonic sensor. Whenever the distance is less than 100cms sent "alert" to IBM cloud and display in device recent events.

Solution:

CODE:

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

WiFiClient wifiClient;

#define ORG "kr9fjo"
#define DEVICE_TYPE "TestDeviceType"
#define DEVICE_ID "12345"
#define TOKEN "VJsSC148dk1dCN3UqS"
#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");
}

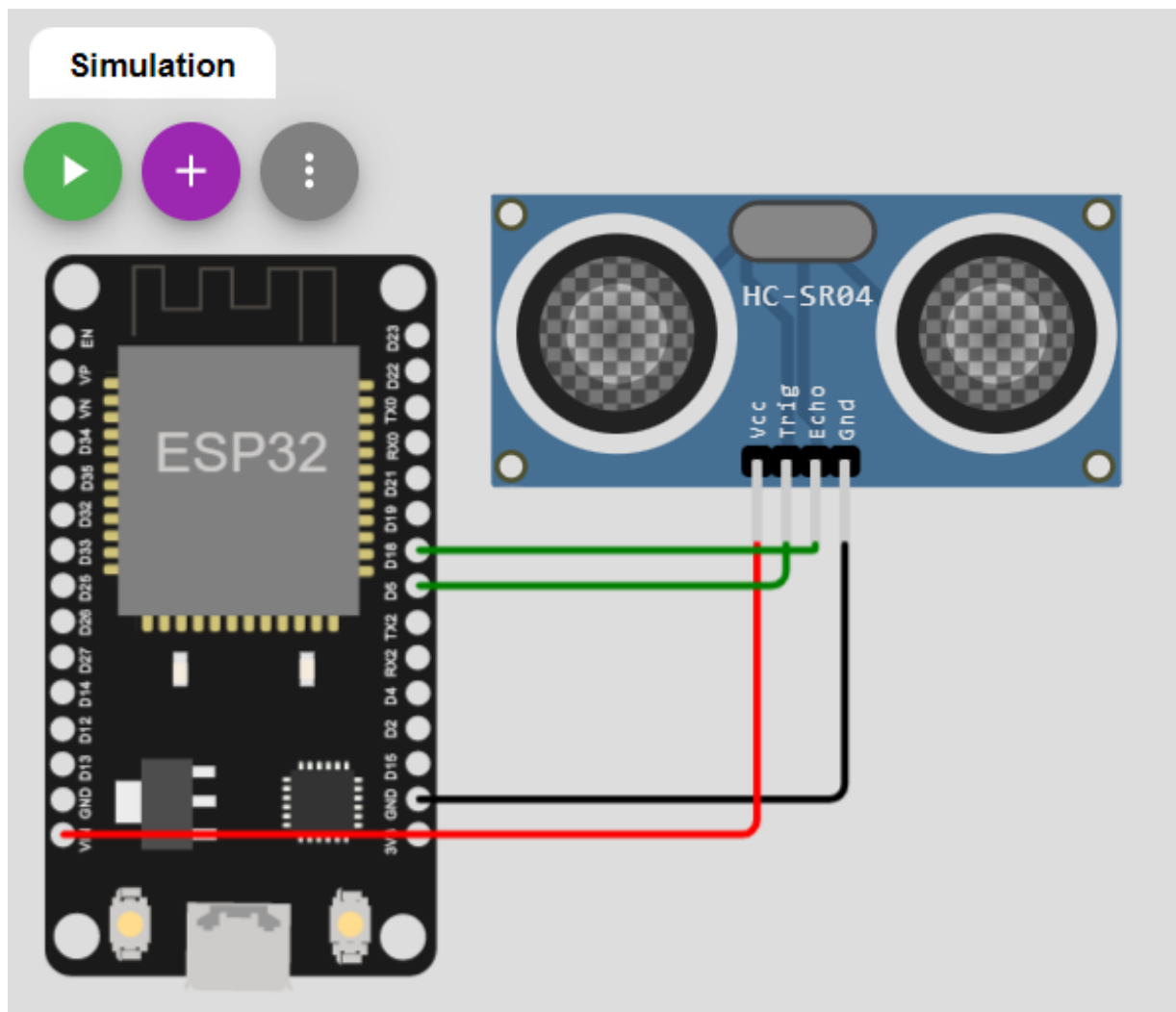
```

```

Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
} else {
    Serial.println("Publish FAILED");
}
}
}

```

Connections:



Output:

The screenshot shows the Wokwi IDE interface. On the left, the sketch code is displayed, which includes headers for WiFi, PubSubClient, and Arduino. It defines a device type, ID, token, and speed. The code sets up a server, publishes data to a topic, and defines a trigpin and echopin. The simulation on the right shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The sensor's output is being sent to an IoT cloud, as indicated by the "Sending payload" messages in the console.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 WiFiClient wifiClient;
6
7 #define ORG "kr9fjo"
8 #define DEVICE_TYPE "TestDeviceType"
9 #define DEVICE_ID "12345"
10 #define TOKEN "Vj5SC148k1dCH3Uq5"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/abcd_1/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/string";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19 PubSubClient client(server, 1883, wifiClient);
20 void publishData();
21
22 const int trigpin=5;
23 const int echopin=18;
24 String command;
25 String data="";
26 String lat="13.253432";
27 String lon="75.007215";
28 String name="Main_Point";
29 String icon="";
30
31 long duration;
32 int dist;
```

Sending payload:
{ "Name": "Main_Point", "Latitude": "13.253432", "Longitude": "75.007215", "Icon": "fa-trash-o", "FillPercent": 0 }
Publish OK

Sending payload:
{ "Name": "Main_Point", "Latitude": "13.253432", "Longitude": "75.007215", "Icon": "fa-trash-o", "FillPercent": 0 }
Publish OK

Output (IBM Cloud):

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes "Browse", "Action", "Device Types", and "Interfaces". A table lists devices, with one device (ID: 12345) selected. The "Recent Events" tab is active, showing a stream of data from the device. The events are listed in a table with columns for Event, Value, Format, and Last Received.

Event	Value	Format	Last Received
Ultra_Sonic	{"Distance":29}	json	a few seconds ago
Ultra_Sonic	{"Distance":92}	json	a few seconds ago
Ultra_Sonic	{"Distance":86}	json	a few seconds ago
Ultra_Sonic	{"Distance":62}	json	a few seconds ago
Ultra_Sonic	{"Distance":10}	json	a few seconds ago

1 Simulation running