

Assignment - 4

wokwi for the ultrasonic sensor

Date	01 November 2022
Student Name	Abdul Rahman S
Student Roll No	911719104002
Maximum Marks	2 Marks

Question-3:

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.

Output :

The image shows two side-by-side screenshots. The left screenshot is from the Wokwi web simulator, displaying a code editor with an Arduino sketch for an ESP32. The code includes a WiFi module, a PubSubClient library, and logic to send an alert to the IBM Watson IoT Platform when the ultrasonic sensor detects a distance less than 100 cm. The right screenshot is from the IBM Watson IoT Platform dashboard, showing a table of recent events for the device 'Assignment4'. The events list shows the device sending an alert with a distance of 90.03 cm, 89.95 cm, and 90.15 cm.

Wokwi Code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "38552b"
6 #define DEVICE_TYPE "NodeMcu"
7 #define DEVICE_ID "Assignment4"
8 #define TOKEN "911719104002"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Abdul";
13 char topic[] = "iot-2/cmd/home/fmt/Str";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_ID";
17 PubSubClient client(server, 1883, wificlient);
18 void publishData();
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30 void setup()
31 {
32   Serial.begin(115200);
33   client.begin(ORG, clientId, token);
34 }
```

IBM Watson IoT Platform Recent Events:

Event	Value	Format	Last Received
Abdul	["Alert Distance":90.03]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago

The image shows the IBM Watson IoT Platform 'Device Drilldown' page for the device 'Assignment4'. The page displays connection information, including the device ID, type, and status. It also shows a table of recent events, which matches the data from the previous screenshot.

Device Drilldown - Assignment4

Connection Information

Basic connection information about this device.

Property	Value
Device ID	Assignment4
Device Type	NodeMcu
Date Added	Nov 8, 2022 8:51 PM
Added By	911719104002@smartinernz.com
Connection Status	Disconnected
Last Connected	Nov 8, 2022 8:58 PM
Client Address	216.246.119.62 Insecure
Duration	a few seconds
Data Transferred	2.2 KB

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
Abdul	["Alert Distance":90.03]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago
Abdul	["Alert Distance":89.95]	json	a few seconds ago

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays a table with columns: Device ID, Status, and Device Type. The first row shows 'Assignment4' with a status of 'Disconnected' and a device type of 'NodeMcu'. Below this, a 'Recent Events' tab is selected, showing a list of events. The events table has columns: Event, Value, Format, and Last Received. The events are all from 'Abdul' and contain the JSON string '{"Alert Distance":89.95}' in 'json' format, received 'a few seconds ago'. The bottom status bar shows '28°C Cloudy' and system icons.

Device ID	Status	Device Type
Assignment4	Disconnected	NodeMcu

Event	Value	Format	Last Received
Abdul	{"Alert Distance":89.95}	json	a few seconds ago
Abdul	{"Alert Distance":89.95}	json	a few seconds ago
Abdul	{"Alert Distance":89.95}	json	a few seconds ago
Abdul	{"Alert Distance":89.95}	json	a few seconds ago
Abdul	{"Alert Distance":89.95}	json	a few seconds ago

Code :

```
/*
Done By Abdul Rahman S

Assignment 4

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

*/

#include <WiFi.h>
```

```
#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "38552b"

#define DEVICE_TYPE "NodeMcu"

#define DEVICE_ID "Assignment4"

#define TOKEN "91119104002"

#define speed 0.034

#define led 14

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/Abdul/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);

    }

    data3="";

}
```