

Assignment -4

Distance Detection Using Ultrasonic Sensor

Assignment Date	27 October 2022
Student Name	M.Anuvarshitha
Student Roll Number	622119105004
Maximum Marks	2 Marks

Question-1:

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.

WOKWI LINK: <https://wokwi.com/projects/346502216516895315>

CODE:

```
#include <WiFi.h> //library for wifi #include
<PubSubClient.h> //library for MQTT

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-----credentials of IBM Accounts-----

#define ORG "f59trs" //IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonicsensor" //Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "distancedetection" //Device ID mentioned in ibmwatson IOT
Platform
#define TOKEN "AIGMGaaF01nawa1QA3" //Token
String data3;
float dist;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; //Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform
and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String"; //
cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //clientid

//
WiFiClient wifiClient; // creating the instance for wificlient
```

```

PubSubClient client(server, 1883, callback, wifiClient);
//calling the predefined client id by passing parameter likeserver id,portand
wificredential

int LED = 4; int
trig = 5; int echo =
18; void setup()
{
  Serial.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED, OUTPUT);
  delay(10); wificonnect();
  mqttconnect();
}
void loop()// Recursive Function
{

  digitalWrite(trig, LOW);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  float dur = pulseIn(echo, HIGH); float dist
= (dur * 0.0343)/2; Serial.print
("Distance in cm"); Serial.println(dist);

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

/* ..... retrieving to
Cloud ..... */

void PublishData(float dist) { mqttconnect();//function call for connecting
to ibm
/*
  creating the String in in form JSON to update the data to ibm cloud
*/
  String object;

```

```

if (dist < 100)
{
    digitalWrite(LED, HIGH);
    Serial.println("object is near"); object =
    "Near";
}
else
{
    digitalWrite(LED, LOW); Serial.println("no
    object found"); object = "No";
}

String payload = "{"distance\":"; payload +=
dist;
payload += ", " "\"object\":"; payload +=
object;
payload += "\"}";

Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok"); // if it successfully upload data on the cloud then it will
    print publish ok in Serial monitor or else it will print publish failed
} 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() //function defination for wificonnect
{
    Serial.println(); Serial.print("Connecting
    to ");

    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentialsto establish the
connection
    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 intpayloadLength)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
        //Serial.print((char)payload[i]);data3 +=
        (char)payload[i];
    }

    // Serial.println("data: "+ data3);
    // if(data3=="Near")
    // {
    // Serial.println(data3);
    // digitalWrite(LED,HIGH);

    // }

    // else
    // {
    // Serial.println(data3);

```

```
// digitalWrite(LED,LOW);
```

```
/
```

OUTPUT:

When object is not near to the ultrasonic sensor

The screenshot displays the Arduino IDE interface with a simulation running. The left pane shows the sketch code, which includes headers for WiFi and MQTT, defines for IBM IoT Platform credentials, and logic for publishing sensor data. The right pane shows a simulation of the hardware: an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor and an LED. The simulation console at the bottom right shows the following output:

```
no object found
Sending payload: {"distance":403.45,"object":"No"}
Publish ok
Distancein cm233.00
no object found
Sending payload: {"distance":233.00,"object":"No"}
Publish ok
```

Data sent to the IBM cloud device when the object is far

The screenshot displays the IBM Cloud IoT Platform console. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue header bar shows the device name 'distancedetection', its status 'Connected', type 'ultrasonicsensor', and a timestamp 'Oct 19, 2022 11:56 AM'. Below this, a tabbed interface has 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of data events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. Five rows of data are shown, all with 'Data' as the event type and JSON objects as values. The 'Last Received' column indicates 'a few seconds ago' for each entry. At the bottom, a status bar shows 'Items per page 50' and '1-1 of 1 item'. A small box in the bottom right corner states '0 Simulations running'.

Items per page 50 | 1-1 of 1 item

0 Simulations running

Event	Value	Format	Last Received
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago
Data	{"distance":235.02,"object":"No"}	json	a few seconds ago

When object is nearer to the ultrasonic sensor

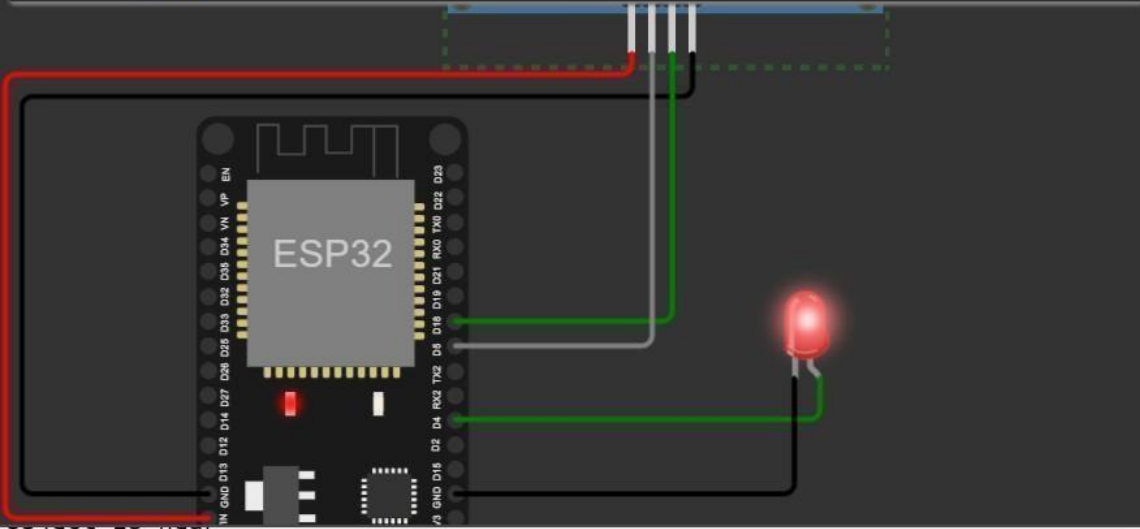
Simulation

HC-SR04

01:03.599 94%

Editing Ultrasonic Distance Sensor

Distance: 91cm



Sending payload: {"distance":91.77,"object":"Near"}

Publish ok

Distancein cm91.77

object is near

Sending payload: {"distance":91.77,"object":"Near"}

Publish ok

Distancein cm91.77

Data sent to the IBM cloud device when the object is near

The screenshot displays the IBM Cloud IoT Platform console. At the top, there are tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is located below the tabs. On the right, there is a button labeled 'Add Device' with a plus icon.

The main content area shows a list of devices. The first device is 'distancedetection', which is 'Connected' and has a device type of 'ultrasonicsensor'. It was added on 'Oct 19, 2022 11:56 AM'. A dropdown menu is visible next to the device name.

Below the device list, the 'Recent Events' tab is selected. It shows a message: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this message is a table of recent events.

Event	Value	Format	Last Received
Data	{"distance":91.77,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.75,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.77,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.79,"object":"Near"}	json	a few seconds ago
Data	{"distance":91.8,"object":"Near"}	json	a few seconds ago

At the bottom right, there is a status indicator that says '0 Simulations running'.

