

Assignment - 4
Python Programming

Assignment Date	06 November 2022
Student Name	Mr. KAVIN P
Student Roll Number	621319106038
Maximum Marks	2 Marks

Question :

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.

CODE :

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
const int trigPin = 5;
const int echoPin = 18;
//define sound speed in cm/uS
#define Speed 0.034
#define cm_to_inch 0.393701
long duration;
float distance;
float distanceInch;

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----

#define ORG "625xj1"//IBM ORGANITION ID
#define DEVICE_TYPE "ultrasonic"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "ultrasonicsensor"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "jvBiD0m(uhKxiLZz+-" //Token
String data3;

//----- 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;//client id

```

```

WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient);

```

```

void setup() {
    Serial.begin(115200); // Starts the serial communication
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    Serial.println();
    wificonnect();
    mqttconnect();
}

```

```

void loop() {
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance
    distance = duration * Speed/2;

    // Convert to inches
    distanceInch = distance * cm_to_inch;

    // Prints the distance in the Serial Monitor
    Serial.print("Distance : ");
    Serial.println(distance);

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

```

```

}

void PublishData(float centimeter) {
  mqttconnect();//function call for connecting to ibm
  /*
    creating the String in in form JSON to update the data to ibm cloud
  */
  String payload = "{\"Distance in Centimeter\":\"";
  payload += centimeter;
  payload += "\"}";

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

  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully 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... ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to 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 int payloadLength)
{
    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];
    }
}

```

WOKWI

SAVE

SHARE

sketch.ino

Docs

sketch.ino

diagram.json

libraries.txt

Library Manager

```

1  #include <WiFi.h>
2  #include <WiFiClient.h>
3  #include <PubSubClient.h>
4  const int trigPin = 5;
5  const int echoPin = 18;
6  //define sound speed in cm/uS
7  #define Speed 0.034
8  #define cm_to_inch 0.393701
9  long duration;
10 float distance;
11 float distanceInch;
12
13
14 void callback(char* subscribetopic, byte* payload, unsigned int payloadLen) {
15 //-----credentials of IBM Accounts-----
16
17 #define ORG "625xj1"//IBM ORGANITION ID
18 #define DEVICE_TYPE "ultrasonic"//Device type mentioned in ibm watson IOT
19 #define DEVICE_ID "ultrasonicsensor"//Device ID mentioned in ibm watson IOT
20 #define TOKEN "jvBiD0m(uhKxILZz+-" //Token
21 String data3;
22
23
24
25 //----- Customise the above values -----
26 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
27 char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of
28 char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT com
29 char authMethod[] = "use-token-auth";// authentication method
30 char token[] = TOKEN;
31

```

Simulation

00:27.093 46%

Publish ok

Distance : 399.94

Sending payload: {"Distance in Centimeter":399.94}

Publish ok

Distance : 399.96

Sending payload: {"Distance in Centimeter":399.96}

Publish ok

WOKWI

SAVE

SHARE

sketch.ino

Docs

sketch.ino

diagram.json

libraries.txt

Library Manager

```

64 // Prints the distance in the Serial Monitor
65 Serial.print("Distance : ");
66 Serial.println(distance);
67
68 PublishData(distance);
69 delay(1000);
70 if (!client.loop()) {
71   mqttconnect();
72 }
73
74
75 void PublishData(float centimeter) {
76   mqttconnect();//function call for connecting to ibm
77   /*
78   | creating the String in in form JSON to update the data to ibm cloud
79   */
80   String payload = "{"Distance in Centimeter\":";
81   payload += centimeter;
82   payload += "}";
83
84
85   Serial.print("Sending payload: ");
86   Serial.println(payload);
87
88
89   if (client.publish(publishTopic, (char*) payload.c_str())) {
90     Serial.println("Publish ok");// if it successfully upload data on the
91   } else {
92     Serial.println("Publish failed");
93   }
94

```

Simulation

01:19.115 66%

Publish ok

Distance : 399.96

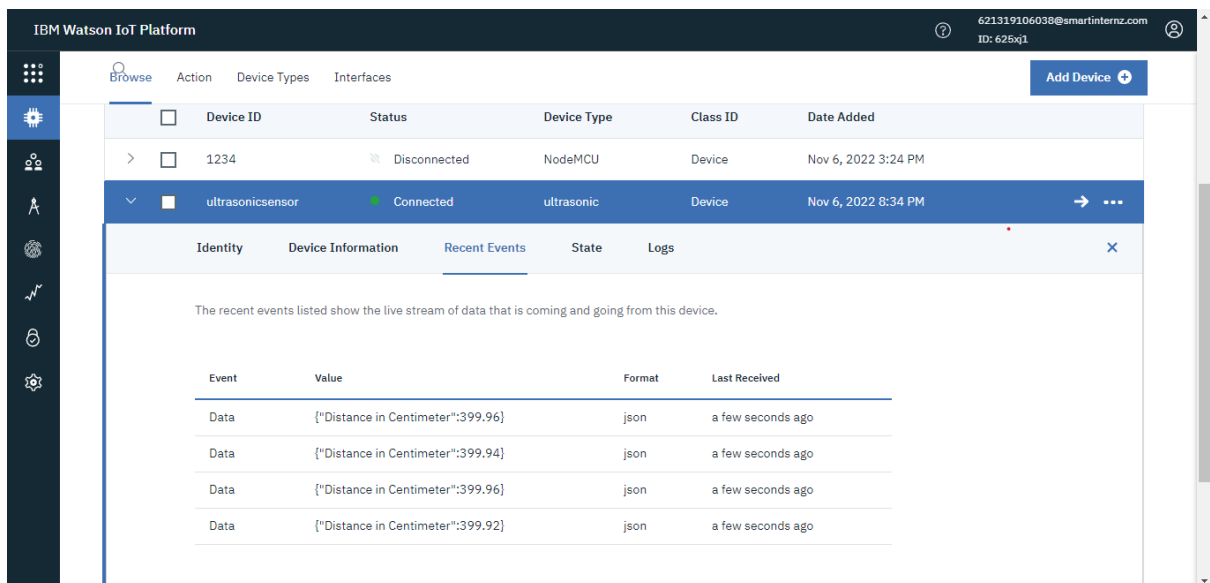
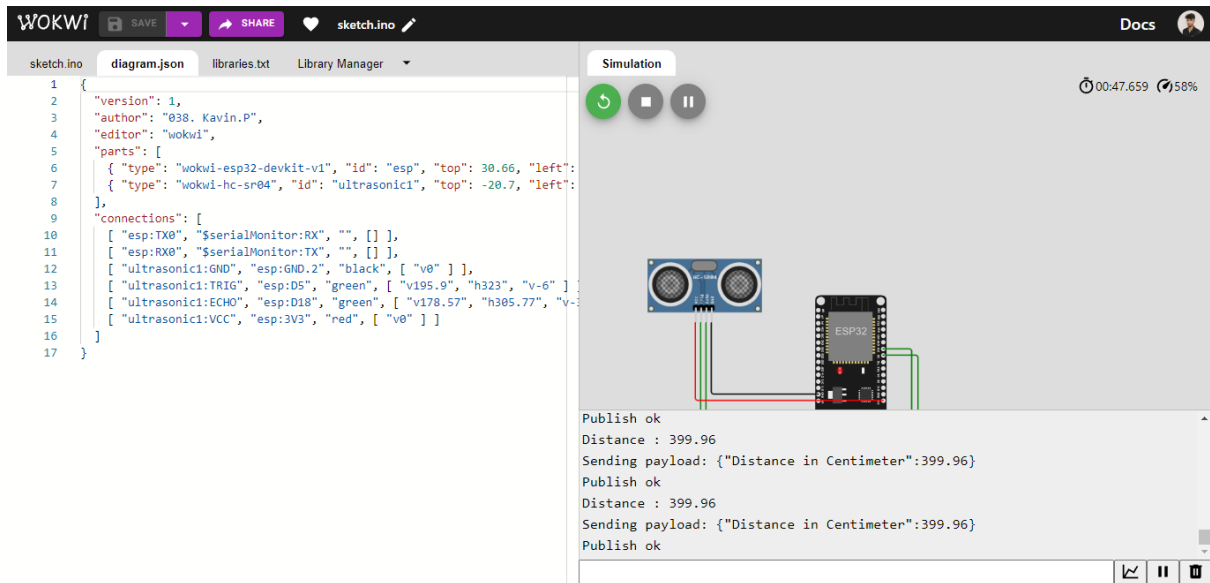
Sending payload: {"Distance in Centimeter":399.96}

Publish ok

Distance : 399.96

Sending payload: {"Distance in Centimeter":399.96}

Publish ok



Wokwi Share Link :

<https://wokwi.com/projects/347587554296463954>