

ASSIGNMENT – 4

Team ID	PNT2022TMID53651
Project name	Project - Industry Specific Intelligent Fire Management system

QUESTION:

Write a 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:

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 "jysy98" //IBM ORGANITION ID
#define DEVICE_TYPE "Sivasubramanian" //Device type mentioned in ibm
watson IOT Platform
#define DEVICE_ID "1710" //Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "Siva@17102001" //Token

// #define ORG "jysy98" //IBM ORGANITION ID
// #define DEVICE_TYPE "Sivasubramanian" //Device type mentioned in ibm
watson IOT Platform
// #define DEVICE_ID "1710" //Device ID mentioned in ibm watson IOT
Platform
// #define TOKEN "Siva@17102001" //Token
String data3;
```

```

float distance;
#define sound_speed 0.034
int trigpin=18;
int echopin=19;
int led=5;
int LED=9;
long duration;
String message;

//----- 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); //calling the predefined
client id by passing parameter like server id,portand wificredential
void setup()// configuring the ESP32
{
    Serial.begin(115200);
    pinMode(trigpin,OUTPUT);
    pinMode(echopin,INPUT);
    pinMode(led,OUTPUT);
    delay(10);
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop()// Recursive Function
{
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delay(1000);

```

```

    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    distance=duration*sound_speed/2;

    Serial.println("distance"+String(distance)+"cm");

    if(distance<100)
    {message="Alert";
    digitalWrite(led,HIGH);}
    else
    {message="No problem";
    digitalWrite(led,LOW);}
    delay(1000);
    PublishData(distance,message);
    if (!client.loop()) {
        mqttconnect();
    }
}

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

void PublishData(float d,String a) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"distance\":";
    payload += d;
    payload += "}";
    payload += ", \"{"message\":";
    payload += a;
    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 to ");

  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];
}

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

}

else
{
Serial.println(data3);
digitalWrite(LED,LOW);

}
data3="";}

```

WOKWI:

The screenshot displays the WOKWI simulation interface. On the left, the 'sketch.ino' file is open, showing the following code:

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4
5 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
6
7 // -----credentials of IBM Accounts-----
8 #define ORG "0jjsl2" //IBM ORGANITION ID
9 #define DEVICE_TYPE "b11m3edevicetype" //Device type mentioned in ibm watson IOT Platform
10 #define DEVICE_ID "b11m3edeviceid" //Device ID mentioned in ibm watson IOT Platform
11 #define TOKEN "_zLY3G7os50?M5puUo" //Token
12
13 // #define ORG "jysy98" //IBM ORGANITION ID
14 // #define DEVICE_TYPE "Sivasubramanian" //Device type mentioned in ibm watson IOT Platform
15 // #define DEVICE_ID "1710" //Device ID mentioned in ibm watson IOT Platform
16 // #define TOKEN "siva@17102001" //Token
17
18 String data3;
19 float distance;
20 #define sound_speed 0.034
21 int trigpin=18;
22 int echopin=19;
23 int led=5;
24 long duration;
25 String message;
26
27
28 //----- Customise the above values -----
29 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
30 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and for

```

On the right, the 'Simulation' tab shows a 3D model of the hardware. An ESP32 microcontroller is connected to an HC-SR04 ultrasonic sensor. The sensor's VCC pin is connected to the ESP32's 5V pin, GND to GND, and the Trig pin to pin 18. The Echo pin is connected to pin 19. A red LED is connected to pin 5 (anode) and GND (cathode).

DISTANCE IS LESS THAN 100 cms:

WOKWI SAVE SHARE Sivasubramanian Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4
5 void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
6
7 // -----credentials of IBM Accounts-----
8 #define ORG "0jjsl2" //IBM ORGANITION ID
9 #define DEVICE_TYPE "b11m3edevicetype" //Device type mentioned in ibm watson IOT Platform
10 #define DEVICE_ID "b11m3edevicid" //Device ID mentioned in ibm watson IOT Platform
11 #define TOKEN "_z1Y3G7Os5O?M5puuo" //Token
12
13 // #define ORG "jysy98" //IBM ORGANITION ID
14 // #define DEVICE_TYPE "Sivasubramanian" //Device type mentioned in ibm watson IOT Platform
15 // #define DEVICE_ID "1710" //Device ID mentioned in ibm watson IOT Platform
16 // #define TOKEN "Siva@17102001" //Token
17
18 String data3;
19 float distance;
20 #define sound_speed 0.034
21 int trigpin=18;
22 int echopin=19;
23 int led=5;
24 int LED=9;
25 long duration;
26 String message;
27
28 //----- Customise the above values -----
29 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
30 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and for
31 char subscribtopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command type AND COMMAND
32 char authMethod[] = "use-token-auth"; // authentication method
```

Simulation 01:02.576

Editing Ultrasonic Distance Sensor
Distance: 60cm

distance59.94cm
Sending payload: {"distance":59.94},{"message":Alert}
Publish ok
distance59.94cm
Sending payload: {"distance":59.94},{"message":Alert}
Publish ok

DISTANCE IS GREATER THAN 100 cms:

WOKWI SAVE SHARE Sivasubramanian Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4
5 void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
6
7 // -----credentials of IBM Accounts-----
8 #define ORG "0jjsl2" //IBM ORGANITION ID
9 #define DEVICE_TYPE "b11m3edevicetype" //Device type mentioned in ibm watson IOT Platform
10 #define DEVICE_ID "b11m3edevicid" //Device ID mentioned in ibm watson IOT Platform
11 #define TOKEN "_z1Y3G7Os5O?M5puuo" //Token
12
13 // #define ORG "jysy98" //IBM ORGANITION ID
14 // #define DEVICE_TYPE "Sivasubramanian" //Device type mentioned in ibm watson IOT Platform
15 // #define DEVICE_ID "1710" //Device ID mentioned in ibm watson IOT Platform
16 // #define TOKEN "Siva@17102001" //Token
17
18 String data3;
19 float distance;
20 #define sound_speed 0.034
21 int trigpin=18;
22 int echopin=19;
23 int led=5;
24 int LED=9;
25 long duration;
26 String message;
27
28 //----- Customise the above values -----
29 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
30 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform and for
31 char subscribtopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command type AND COMMAND
32 char authMethod[] = "use-token-auth"; // authentication method
```

Simulation 01:11.042

Editing Ultrasonic Distance Sensor
Distance: 128cm

distance59.94cm
Sending payload: {"distance":59.94},{"message":Alert}
Publish ok
distance127.94cm
Sending payload: {"distance":127.94},{"message":No problem}
Publish ok

DEVICE RECENT EVENTS IN IBM WATSON:

The screenshot displays the IBM Watson IoT Platform interface. At the top, the header shows 'IBM Watson IoT Platform' and a user profile with email '2019ec0062@svce.ac.in' and ID 'jyzy98'. The main navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The 'Device Simulator' toggle is turned off. The main content area shows a list of devices. The selected device has ID '1710', status 'Connected', and type 'Sivasubramanian'. Below the device list, the 'Recent Events' tab is active, showing a table of events.

Event	Value	Format	Last Received
Data	{"d":{"distance":342.3,"message":"No problem"}}	json	a few seconds ago
Data	{"d":{"distance":363.8,"message":"No problem"}}	json	a few seconds ago
Data	{"d":{"distance":87.89,"message":"Alert"}}	json	a few seconds ago
Data	{"d":{"distance":367.01,"message":"No problem"}}	json	a few seconds ago
Data	{"d":{"distance":110.52,"message":"No problem"}}	json	a few seconds ago

WOKWI LINK:

<https://wokwi.com/projects/347660572061336147>