

## ASSIGNMENT 4

<b>Team ID</b>	PNT2022TMID53630
<b>Project name</b>	Smart Farmer - IoT Enabled Smart Farming Application

### QUESTION:

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. 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 "w79t1i" //IBM ORGANITION ID
#define DEVICE_TYPE "Newdevice" //Device type mentioned in ibm watson
IOT Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "ibm12345678" //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 subscribtopic[] = "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];
    }
    data3="";
}

```

# WOKWI:

WOKWI interface showing a sketch and a simulation.

**Sketch (sketch.ino):**

```
117 Serial.println();
118 }
119 }
120 void wificonnect() //function definition for wificonnect
121 {
122   Serial.println();
123   Serial.print("Connecting to ");
124
125   WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
126   while (WiFi.status() != WL_CONNECTED) {
127     delay(500);
128     Serial.print(".");
129   }
130   Serial.println("");
131   Serial.println("WiFi connected");
132   Serial.println("IP address: ");
133   Serial.println(WiFi.localIP());
134 }
135
136 void initManagedDevice() {
137   if (client.subscribe(subscribetopic)) {
138     Serial.println((subscribetopic));
139     Serial.println("subscribe to cmd OK");
140   } else {
141     Serial.println("subscribe to cmd FAILED");
142   }
143 }
144
145 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
146 {
147   Serial.print("callback invoked for topic: ");
148   Serial.println(subscribetopic);
149   for (int i = 0; i < payloadLength; i++) {
150     // ...
151   }
152 }
```

**Simulation:**

The simulation shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The sensor is emitting a red laser beam. The console output shows the following:

```
Sending payload: {"distance":75.94,{"message":"Alert"}
Publish ok
distance75.96cm
```

## DISTANCE IS LESS THAN 100 cms:

WOKWI interface showing a sketch and a simulation.

**Sketch (sketch.ino):**

```
117 Serial.println();
118 }
119 }
120 void wificonnect() //function definition for wificonnect
121 {
122   Serial.println();
123   Serial.print("Connecting to ");
124
125   WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the connection
126   while (WiFi.status() != WL_CONNECTED) {
127     delay(500);
128     Serial.print(".");
129   }
130   Serial.println("");
131   Serial.println("WiFi connected");
132   Serial.println("IP address: ");
133   Serial.println(WiFi.localIP());
134 }
135
136 void initManagedDevice() {
137   if (client.subscribe(subscribetopic)) {
138     Serial.println((subscribetopic));
139     Serial.println("subscribe to cmd OK");
140   } else {
141     Serial.println("subscribe to cmd FAILED");
142   }
143 }
144
145 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
146 {
147   Serial.print("callback invoked for topic: ");
148   Serial.println(subscribetopic);
149   for (int i = 0; i < payloadLength; i++) {
150     // ...
151   }
152 }
```

**Simulation:**

The simulation shows the same setup as the first image. A dialog box titled "Editing Ultrasonic Distance Sensor" is open, showing a distance of 76cm. The console output shows the following:

```
distance75.94cm
Sending payload: {"distance":75.94,{"message":"Alert"}
Publish ok
distance75.94cm
Sending payload: {"distance":75.94,{"message":"Alert"}
Publish ok
distance75.94cm
```

## DISTANCE IS GREATER THAN 100 cms:

```
117 }
118 Serial.println();
119 }
120 void wificonnect() //function definition for wificonnect
121 {
122   Serial.println();
123   Serial.print("Connecting to ");
124
125   WiFi.begin("wokwi-GUEST", "", 0); //passing the wifi credentials to establish the connect
126   while (WiFi.status() != WL_CONNECTED) {
127     delay(500);
128     Serial.print(".");
129   }
130   Serial.println("");
131   Serial.println("WiFi connected");
132   Serial.println("IP address: ");
133   Serial.println(WiFi.localIP());
134 }
135
136 void initManagedDevice() {
137   if (client.subscribe(subscribetopic)) {
138     Serial.println(subscribetopic);
139     Serial.println("subscribe to cmd OK");
140   } else {
141     Serial.println("subscribe to cmd FAILED");
142   }
143 }
144
145 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
146 {
147   Serial.print("callback invoked for topic: ");
148   Serial.println(subscribetopic);
149   for (int i = 0; i < payloadLength; i++) {
150     //
151   }
152 }
```

Simulation

distance247.95cm  
Sending payload: {"distance":247.95},{"message":"No problem"}  
Publish ok  
distance247.93cm  
Sending payload: {"distance":247.93},{"message":"No problem"}  
Publish ok  
distance247.95cm

## DEVICE RECENT EVENTS IN IBM WATSON:

IBM Watson IoT Platform

2019ec0165@svce.ac.in  
ID: w79tUj

4567 Disconnected NodeMCU Device 0 Simulations running

Event	Value	Format	Last Received
Data	{"d":{"distance":177.62,"message":"No problem"}}	json	a few seconds ago
Data	{"d":{"distance":90.68,"message":"Alert"}}	json	a few seconds ago
Data	{"d":{"distance":203.18,"message":"No problem"}}	json	a few seconds ago
Data	{"d":{"distance":43.06,"message":"Alert"}}	json	a few seconds ago
Data	{"d":{"distance":11.9,"message":"Alert"}}	json	a few seconds ago

## WOKWI LINK:

<https://wokwi.com/projects/347131248185967187>