

Assignment- 4

SmartFarmer- IoT Enabled Smart Farming Application

StudentName	Kalaivani.B
StudentRollNumber	611719106009

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

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

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

#define ORG "az10eu"//IBM ORGANITION ID
#define DEVICE_TYPE "UltraSonicSensor"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //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 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 wifi client
PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client
id by passing parameter like server id, port and wifi credential

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 = "Alert: Person Detected";
    }
    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 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 (inti = 0; i<payloadLength; i++) {
  //Serial.print((char)payload[i]);
  data3 += (char)payload[i];
}

data3="";

```

WokwiLink:

<https://wokwi.com/projects/347204840687927891>

OutputandSimulation:

The screenshot displays the Wokwi web IDE interface. On the left, the 'sketch.ino' file is open, showing a C++ program that uses the MQTT library to connect to the IBM Cloud IoT Platform. The program reads distance data from an HC-SR04 ultrasonic sensor and publishes it as a JSON payload. It also controls an LED based on whether an object is detected within a 100cm range.

The right side of the interface shows a 'Simulation' window with a visual representation of the ESP32, the HC-SR04 sensor, and the LED. Below the simulation, a serial output log shows the following messages:

```

object is near
Sending payload: {"distance":64.54,"object":"Alert: Person Detected"}
Publish ok
Distancein cm64.54
object is near
Sending payload: {"distance":64.54,"object":"Alert: Person Detected"}
Publish ok

```

IBM cloud:

Device Information:

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 of devices. The first device, ID 1234, is highlighted. Its details are shown in a modal window under the 'Device Information' tab.

Identity	Device Information	Recent Events	State	Logs
Device ID	1234			
Device Type	UltraSonicSensor			
Date Added	Nov 8, 2022 11:11 AM			
Added By	611719106030@smartinternz.com			
Connection Status	Disconnected Last Connected: Nov 8, 2022 11:18 AM Client Address: 216.246.119.62 Insecure Duration: 3 minutes Data Transferred: 9.0 KB			

DeviceRecentEvents:

Whenever the distance is less than 100cm send an "alert" to the IBM cloud and display in the device recent events.

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 of devices. The first device, ID 1234, is highlighted. Its details are shown in a modal window under the 'Recent Events' tab.

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"distance":64.59,"object":"","Alert: Person Detect..."}	json	a few seconds ago
Data	{"distance":64.54,"object":"","Alert: Person Detect..."}	json	a few seconds ago
Data	{"distance":64.54,"object":"","Alert: Person Detect..."}	json	a few seconds ago
Data	{"distance":64.54,"object":"","Alert: Person Detect..."}	json	a few seconds ago
Data	{"distance":64.54,"object":"","Alert: Person Detect..."}	json	a few seconds ago