ASSIGNMENT-4

Assignment Date	24 Oct 2022
Team ID	PNT2022TMID16510
Student Name	Karthik
Project Name	Smart Farmer-IoT Enabled Smart Farming Application

Question:

1.Write Code and connections in wokwi for ultrasonic sensor. whatever distance is less than 100 cms send "Alert" to ibm cloud and display in device recent events
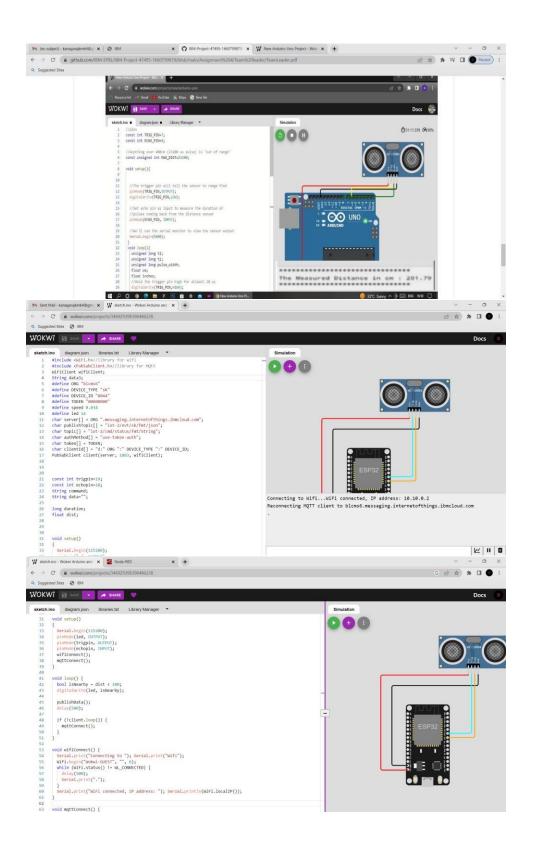
Code:

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

#include <PubSubClient.h>//library for MQtt

WiFiClient wifiClient;

String data3;

#define ORG "blcms6"

#define DEVICE_TYPE "sk"

#define DEVICE_ID "0044"

#define TOKEN "00000000"

#define speed 0.034 #define led 14 char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char publishTopic[]
= "iot-2/evt/sk/fmt/json"; char topic[] = "iot-
2/cmd/status/fmt/String";   char authMethod[] = "use-token-
auth"; char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

PubSubClient client(server, 1883, wifiClient);
```

```cpp
const int trigpin=19; const
int echopin=18; String
command;
String data="";



long duration; float
dist;



void setup()
{
  Serial.begin(115200);
pinMode(led, OUTPUT);
pinMode(trigpin, OUTPUT);
pinMode(echopin, INPUT);
wifiConnect();   mqttConnect();
}



void loop() {   bool isNearby
= dist < 100;
digitalWrite(led, isNearby);
publishData();   delay(500);


  if (!client.loop()) {
mqttConnect();
```

```arduino
  }
}


void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
WiFi.begin("Wokwi-GUEST", "", 6);   while
(WiFi.status() != WL_CONNECTED) {     delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}


void mqttConnect() {   if
(!client.connected()) {
    Serial.print("Reconnecting MQTT client to "); Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");       delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}


void initManagedDevice() {   if
(client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  } else {
```

```
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;   if(dist<100){
    String payload = "{\"Alert Distance\":";
payload += dist;     payload += "}";


    Serial.print("\n");
    Serial.print("Sending payload: ");    Serial.println(payload);     if
(client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Publish OK");
  }
      }
if(dist>100){
    String payload = "{\"Distance\":";
payload += dist;     payload += "}";


    Serial.print("\n");
    Serial.print("Sending payload: ");
Serial.println(payload);     if(client.publish(publishTopic,
(char*) payload.c_str())) {
      Serial.println("Publish OK");
```

```
  }else {
    Serial.println("Publish FAILED");
  }


}
}
```

```arduino
//pins
const int TRIG_PIN=7;
const int ECHO_PIN=8;

//Anything over 40Hcm (23200 us pulse) is "out of range"
const unsigned int MAX_DIST=23200;

void setup(){

  //The trigger pin will tell the sensor to range find
  pinMode(TRIG_PIN,OUTPUT);
  digitalWrite(TRIG_PIN,LOW);

  //Set echo pin as input to measure the duration of
  //pulses coming back from the distance sensor
  pinMode(ECHO_PIN, INPUT);

  //We'll use the serial monitor to view the sensor output
  Serial.begin(9600);
}
void loop(){
  unsigned long t1;
  unsigned long t2;
  unsigned long pulse_width;
  float cm;
  float inches;
  //Hold the trigger pin high for atleast 10 us
  digitalWrite(TRIG_PIN,HIGH);
```

The Measured Distance in cm : 201.79


```arduino
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
WiFiClient wificlient;
String data3;
#define ORG "blcms6"
#define DEVICE_TYPE "sk"
#define DEVICE_ID "0044"
#define TOKEN "00000000"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/sk/fmt/json";
char topic[] = "iot-2/cmd/status/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);


const int trigpin=19;
const int echopin=18;
String command;
String data="";

long duration;
float dist;


void setup()
{
  Serial.begin(115200);
```

Connecting to Wifi...WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to blcms6.messaging.internetofthings.ibmcloud.com
.


```arduino
void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {
  bool isNearby = dist < 100;
  digitalWrite(led, isNearby);

  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}

void mqttConnect() {
```

WOKWI · SAVE · SHARE · ♥ Docs

sketch.ino | diagram.json | libraries.txt | Library Manager ▼

Simulation

```
53  void wifiConnect() {
54    Serial.print("Connecting to "); Serial.print("WiFi");
55    WiFi.begin("Wokwi-GUEST", "", 6);
56    while (WiFi.status() != WL_CONNECTED) {
57      delay(500);
58      Serial.print(".");
59    }
60    Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
61  }
62
63  void mqttConnect() {
64    if (!client.connected()) {
65      Serial.print("Reconnecting MQTT client to "); Serial.println(server);
66      while (!client.connect(clientId, authMethod, token)) {
67        Serial.print(".");
68        delay(500);
69      }
70      initManagedDevice();
71      Serial.println();
72    }
73  }
74
75  void initManagedDevice() {
76    if (client.subscribe(topic)) {
77      // Serial.println(client.subscribe(topic));
78      Serial.println("IBM subscribe to cmd OK");
79    } else {
80      Serial.println("subscribe to cmd FAILED");
81    }
82  }
83  void publishData()
84  {
85    digitalWrite(trigpin,LOW);
```

WOKWI · SAVE · SHARE · ♥ Docs

sketch.ino | diagram.json | libraries.txt | Library Manager ▼

Simulation

```
89    duration=pulseIn(echopin,HIGH);
90    dist=duration*speed/2;
91    if(dist<100){
92      String payload = "{\"Alert Distance\":";
93      payload += dist;
94      payload += "}";
95
96      Serial.print("\n");
97      Serial.print("Sending payload: ");
98      Serial.println(payload);
99      if (client.publish(publishTopic, (char*) payload.c_str())) {
100       Serial.println("Publish OK");
101     }
102
103   }
104   if(dist>100){
105     String payload = "{\"Distance\":";
106     payload += dist;
107     payload += "}";
108
109     Serial.print("\n");
110     Serial.print("Sending payload: ");
111     Serial.println(payload);
112     if(client.publish(publishTopic, (char*) payload.c_str())) {
113       Serial.println("Publish OK");
114     }else {
115       Serial.println("Publish FAILED");
116     }
117
118   }
119 }
120
```