Assignment -4

Student Roll Number	19L152

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:

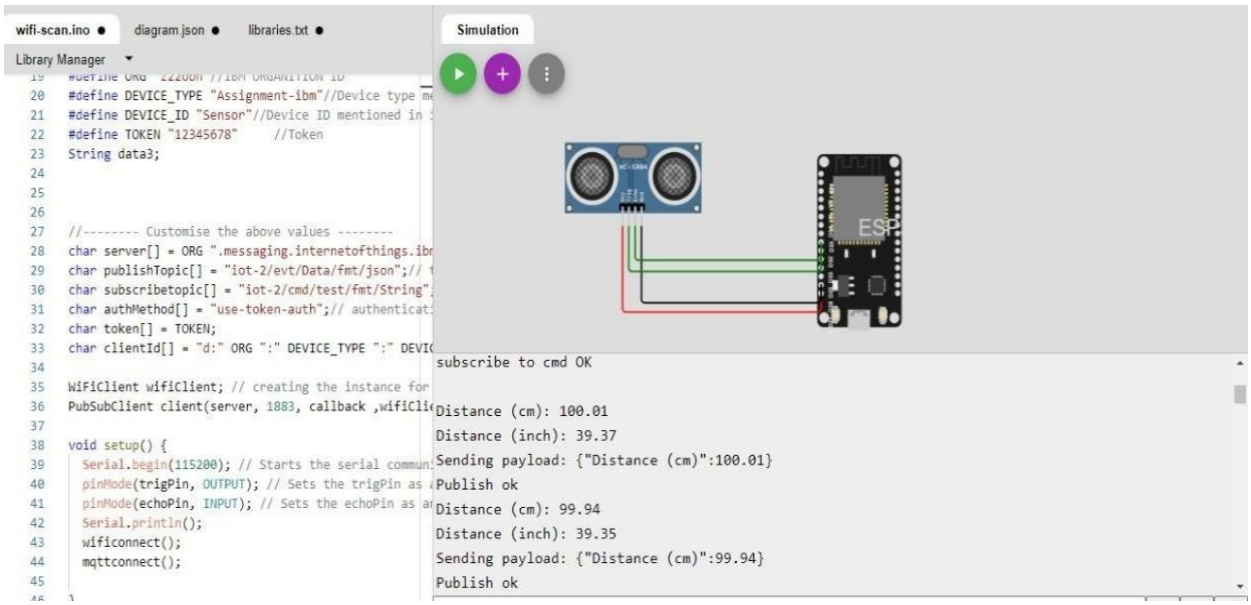
```
#include <WiFi.h>

#include

<WiFiClient.h>

#include

<PubSubClient.h>

const int trigPin = 27;

const int echoPin = 26;

 #define SOUND_SPEED

0.034

#define CM_TO_INCH

0.393701 long duration;

float distanceCm; float

distanceInch;

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



#define ORG "z22obn"

#define DEVICE_TYPE "Assignment-ibm"

#define DEVICE_ID "Sensor"

#define TOKEN "12345678"

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);
   pinMode(trigPin, OUTPUT);
   pinMode(echoPin, INPUT);
  Serial.println();
  wificonnect();
  mqttconnect();

} void
loop() {
digitalWrite(trigPin,LOW);
delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);


 duration = pulseIn(echoPin, HIGH);

distanceCm = duration * SOUND_SPEED/2;
distanceInch =distanceCm *CM_TO_INCH;
 Serial.print("Distance (cm): ");
Serial.println(distanceCm);
```

```
Serial.print("Distance (inch): ");
Serial.println(distanceInch);


   PublishData(distanceCm);
 delay(1000);   if (!
 client.loop()) {
 mqttconnect();
   } }   void PublishData(float Cm) { mqttconnect();

String payload = "{\"Distance (cm)\":";
payload += Cm;    payload += "}";
   Serial.print("Sending payload: ");
   Serial.println(payload);
       if (client.publish(publishTopic, (char*)
 payload.c_str())) {     Serial.println("Publish ok");

   } 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()
 {
   Serial.println();
   Serial.print("Connecting to ");

   WiFi.begin("Wokwi-GUEST", "", 6);(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 Output:



IBM Cloud Alert:

Event	Value	Format	Last Received
Data	{"Distance (cm)":99.98}	json	a few seconds ago
Data	{"Distance (cm)":99.96}	json	a few seconds ago
Data	{"Distance (cm)":99.98}	json	a few seconds ago
Data	{"Distance (cm)":99.98}	json	a few seconds ago
Data	{"Distance (cm)":99.98}	json	a few seconds ago

Wokwi Share Link:

https://wokwi.com/projects/305569599398609473