

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

```
#include "Ultrasonic.h"
```

```
#include <PubSubClient.h>
```

```
#include <WiFiClient.h>
```

```
#include <WiFi.h>
```

```
#define LED 19
```

```
#define ORG "9pdt70"
```

```
#define DEVICE_TYPE "abcd"
```

```
#define DEVICE_ID "12345"
```

```
#define TOKEN "123456789"
```

```
String data3;
```

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

```
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/command/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
```

```
Ultrasonic ultrasonic(5, 18);
```

```
int distance;

void setup() {
  Serial.begin(9600);
  wificonnect();
  mqttconnect();
}

void loop() {
  // Pass INC as a parameter to get the distance in inches
  //client.loop();

  distance = ultrasonic.read(CM);

  Serial.print("Distance in CM: ");
  Serial.println(distance);
  if(distance>100){
    digitalWrite(LED,HIGH);
    PublishData(distance);
  }

  delay(1000);
}

void PublishData(int distance){
  mqttconnect();

  String payload = "{\"distance\":\"";
  payload += distance;
  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());
}

```

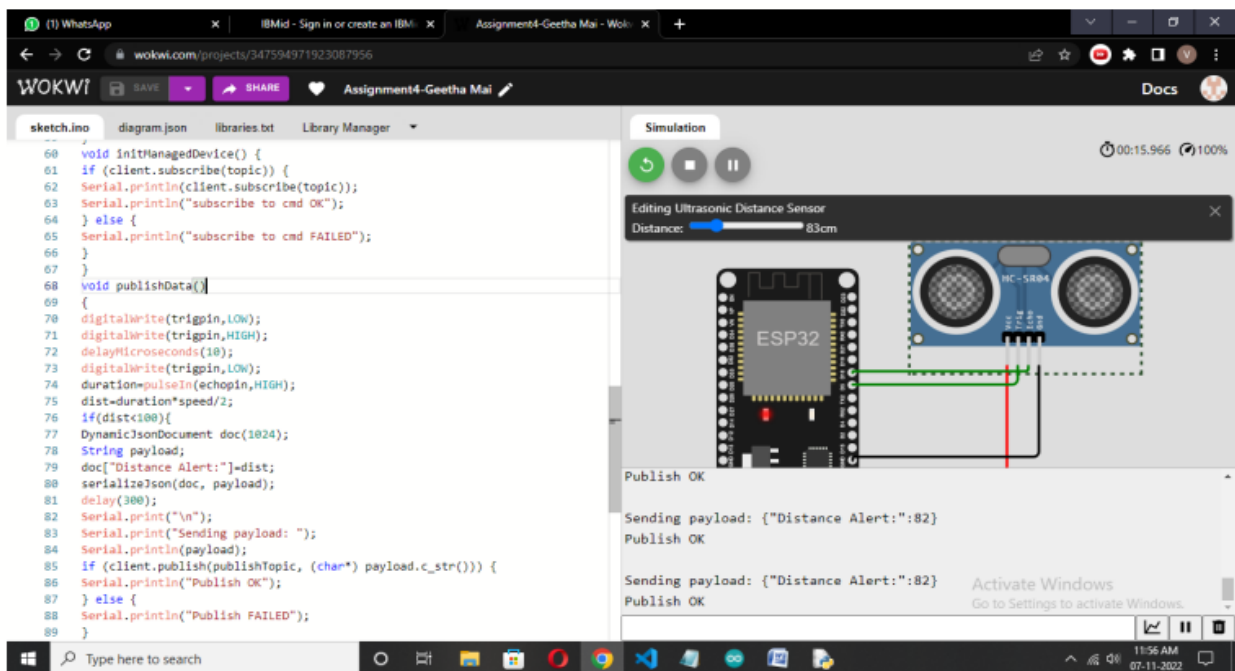
```

void 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="";
}

```

Wowki document linl: <https://wokwi.com/projects/348294436555326035>

OUTPUT:



CLOUD OUTPUT:

