

ASSIGNMENT 4

Date	6 nov 2022
Team ID	PNT2022TMID26079
Project Name	Gas Leakage Monitoring and Alerting System
Name	RUSHENDHAR PS

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 ibmcloud.

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "9djwz2"//IBM ORGANITION ID
#define DEVICE_TYPE "sanjay"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234567"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/event_1/fmt/json";
char topic[] = "iot-2/cmd/led/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=5;
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();
  }
}
```

```
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");
}

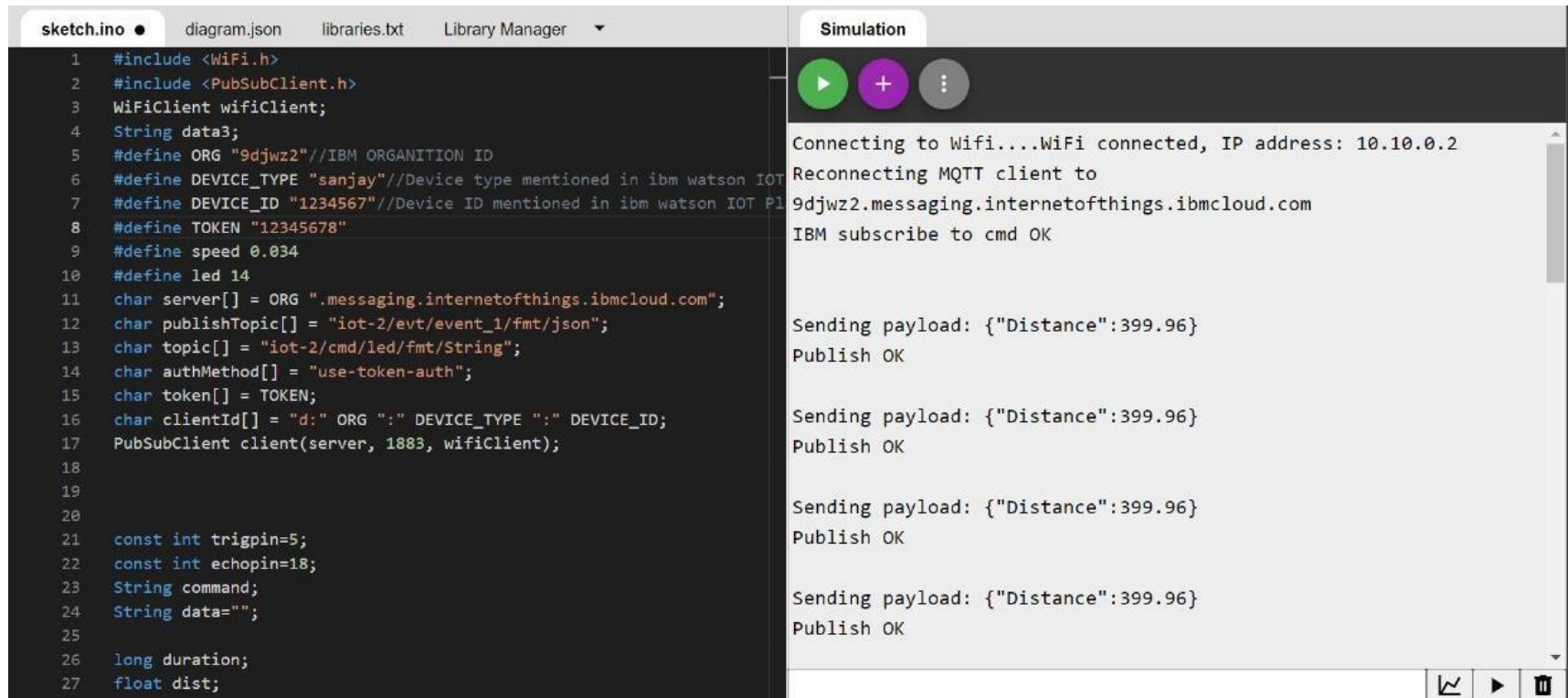
}

}
```

Wokwi link:

<https://wokwi.com/projects/347956062652990036>

output:



The screenshot displays the Wokwi IDE interface. On the left, the 'sketch.ino' file is open, showing a C++ program that simulates an IoT device. The code includes headers for `WiFi.h` and `PubSubClient.h`, defines an organization ID, device type, device ID, and token, and sets up a MQTT client to connect to an IBM Watson IoT Platform endpoint. It also defines a server, topics, and authentication details. The main logic involves sending a payload of `{"Distance": 399.96}` to a specific topic. On the right, the 'Simulation' panel shows the execution log. The log indicates a successful WiFi connection at IP 10.10.0.2, MQTT client reconnection, and successful subscription to the command topic. It then shows three sequential messages being sent to the topic `iot-2/cmd/led/fmt/String`, each with the payload `{"Distance": 399.96}`, and each published successfully.

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  WiFiClient wifiClient;
4  String data3;
5  #define ORG "9djwz2"//IBM ORGANITION ID
6  #define DEVICE_TYPE "sanjay"//Device type mentioned in ibm watson IOT
7  #define DEVICE_ID "1234567"//Device ID mentioned in ibm watson IOT Pl
8  #define TOKEN "12345678"
9  #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/event_1/fmt/json";
13 char topic[] = "iot-2/cmd/led/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
```

Simulation

Connecting to Wifi...WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
9djwz2.messaging.internetofthings.ibmcloud.com
IBM subscribe to cmd OK

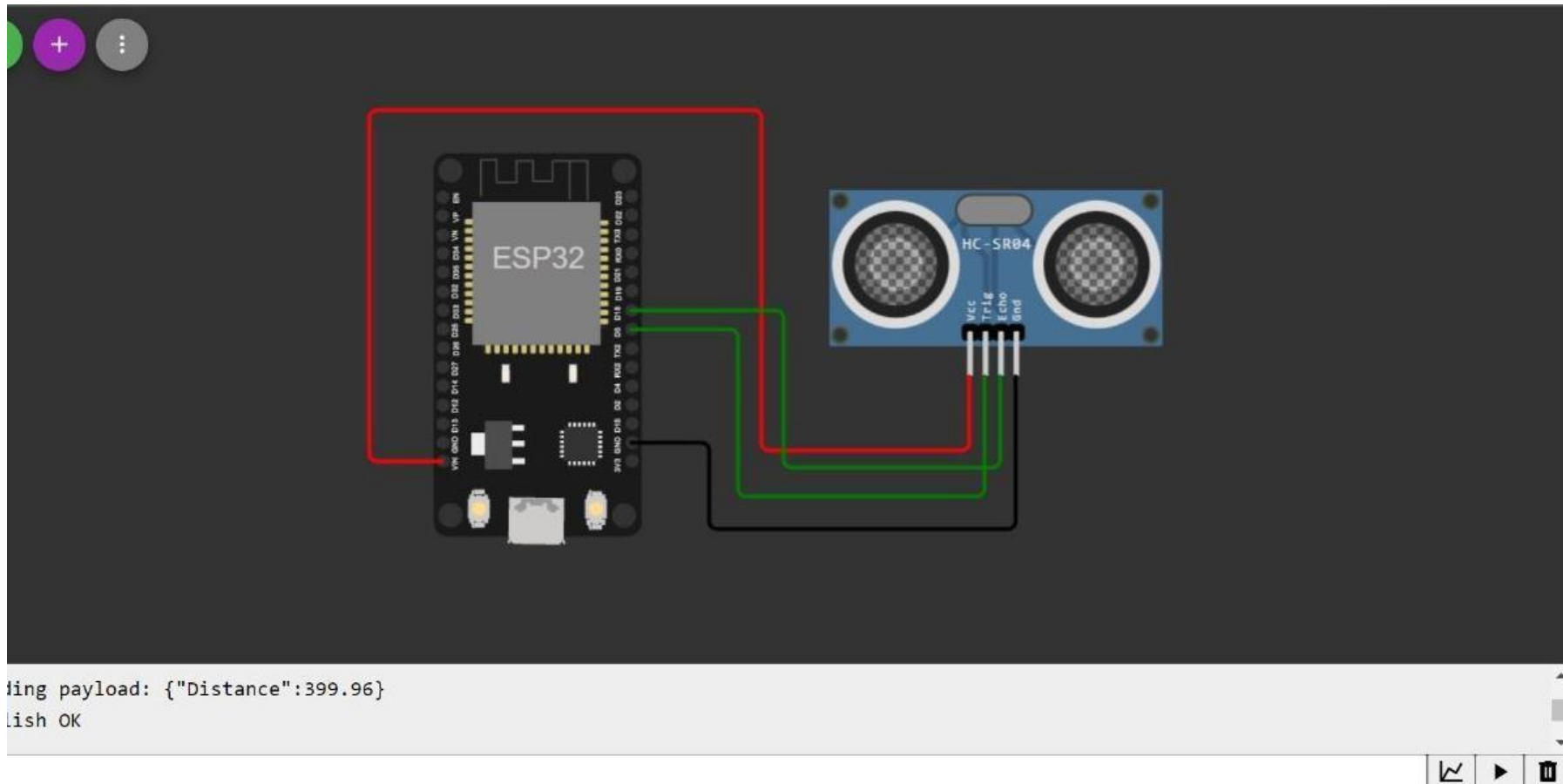
Sending payload: {"Distance":399.96}
Publish OK

Sending payload: {"Distance":399.96}
Publish OK

Sending payload: {"Distance":399.96}
Publish OK

Sending payload: {"Distance":399.96}
Publish OK

WOKWI:



IBM CLOUD:



1234567

Connected

sanjay

Device

Nov 10, 2022 9:14 PM

→ ...

Identity

Device Information

Recent Events

State

Logs

×

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

Event	Value	Format	Last Received
event_1	{"Distance":399.98}	json	a few seconds ago
event_1	{"Distance":399.96}	json	a few seconds ago
event_1	{"Distance":399.96}	json	a few seconds ago
event_1	{"Distance":399.94}	json	a few seconds ago
event_1	{"Distance":399.96}	json	a few seconds ago

1 Simulation running