

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITIAN CITIES

ASSIGNMENT-4

TEAM ID : PNT2022TMID04391
NAME : ROHITH VIGNESH
ROLL NO : 737819CSR165

Write code and connections in wokwi for ultrasonic sensors. 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;
#define ORG "jant3i"
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
#define speed 0.034
char server[] = ORG".messaging.internetofthings.ibmcloud.com"; char
publishTopic[] = "iot-2/evt/raspberrypi_1/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;

void setup()
```



```

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

void loop() {
  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("subscribe to cmd OK");
}
else {
Serial.println("subscribe to cmd FAILED");
}
}

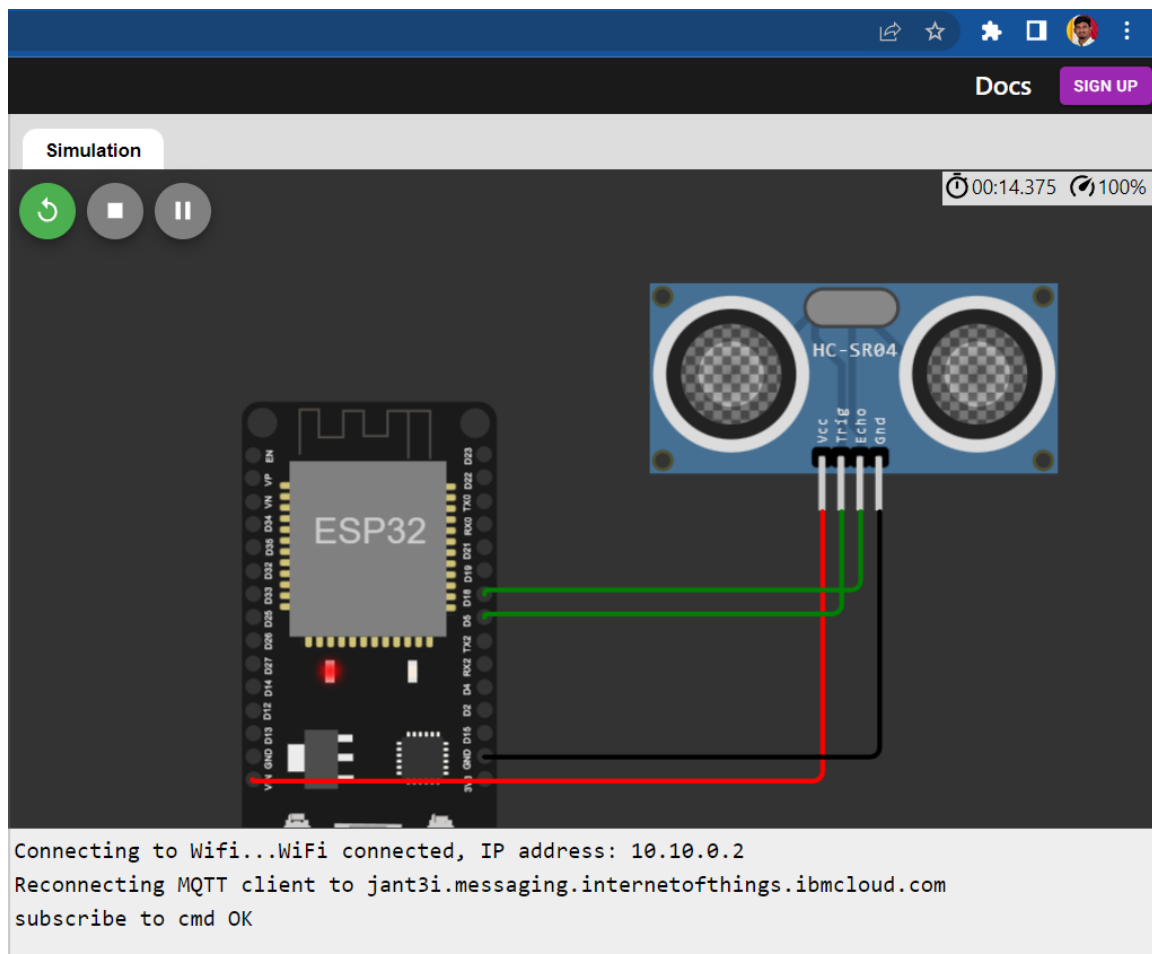
void publishData()
{
digitalWrite(trigpin,LOW);
digitalWrite(trigpin,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin,LOW);
duration=pulseIn(echopin,HIGH);
dist=duration*speed/2;
if(dist<100){
String payload = "{\\"Alert distance\\":\\";
payload += dist;
payload += "\\"}";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str()))
{ Serial.println("Publish OK");
} else {
Serial.println("Publish FAILED");
}
}

}

```



CONNECTIONS:



WOKWI LINK:

<https://wokwi.com/projects/347022808948146772>

OUTPUT:

The screenshot shows the Wokwi web interface with a project titled "ESP32". The left pane displays the sketch code, and the right pane shows the simulation and the serial terminal output.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 #define ORG "jant3i"
5 #define DEVICE_TYPE "raspberrypi"
6 #define DEVICE_ID "12345"
7 #define TOKEN "12345678"
8 #define speed 0.034
9 char server[] = ORG".messaging.internetofthings.ibmcloud.com"; char
10 publishTopic[] = "iot-2/evt/raspberrypi_1/fmt/json";
11 char topic[] = "iot-2/cmd/home/fmt/string";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 PubSubClient client(server, 1883, wifiClient);
17 void publishData();
18 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24
25 void setup()
26 {
27   Serial.begin(115200);
28   pinMode(trigpin, OUTPUT);
29   pinMode(echopin, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33 void loop() {
34   publishData();
35   delay(500);
36 }
```

Connecting to Wifi...Wifi connected, IP address: 10.10.0.2
Reconnecting MQTT client to jant3i.messaging.internetofthings.ibmcloud.com
subscribe to cmd OK
Reconnecting MQTT client to jant3i.messaging.internetofthings.ibmcloud.com
subscribe to cmd OK

The screenshot shows the IBM Watson IoT Platform dashboard. The top section displays a list of devices, and the bottom section shows the details for the 'raspberrypi_1' device, including its status, device type, and recent events.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	raspberrypi	Device	Oct 28, 2022 2:47 PM	
raspberrypi_1	Disconnected	raspberrypi	Device	Oct 29, 2022 3:26 PM	

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	("Alert distance":53)	json	a few seconds ago
event_1	("Alert distance":93)	json	a few seconds ago
event_1	("Alert distance":90)	json	a minute ago
event_1	("Alert distance":63)	json	a minute ago
event_1	("Alert distance":4)	json	2 minutes ago

1 Simulation running



Edit with WPS Office

