

Assignment-4

Date	24 October 2022
Name	Vetri Vignesh A
Roll Number	620119106103
Team ID	PNT2022TMID30898
Project Name	Project – Personal Assistance for seniors who are self-reliant

Question :

Write code and connections in wokwi for ultrasonic sensors. That 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.

Wokwi:

<https://wokwi.com/projects/348217160152121939>

Code:

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

WiFiClient wifiClient;

#define ORG "sphfkr"
#define DEVICE_TYPE "vetri"
#define DEVICE_ID "iot_ibm"
#define TOKEN "VickyVetri@04"
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/distance/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: ");

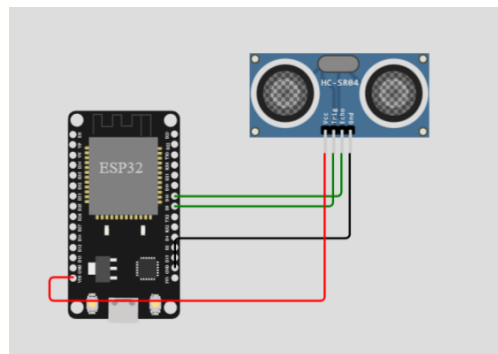
```

```

Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str()))
{ Serial.println("Publish OK");
} else {
Serial.println("Publish FAILED");
}
}
}
}

```

Diagram:



Wokwi Output:

Wokwi interface showing the simulation of the ESP32 and HC-SR04 sensor setup. The left pane displays the sketch code, and the right pane shows the simulation results.

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 WiFiClient wificlient;
5
6 #define ORG "sphfkr"
7 #define DEVICE_TYPE "vetri"
8 #define DEVICE_ID "iot_ibm"
9 #define TOKEN "Vickyvetri@04"
10 #define speed 0.034
11
12 char server[] = ORG".messaging.internetofthings.ibmcloud.com";
13 char publishTopic[] = "iot-2/evt/distance/fmt/json";
14 char topic[] = "iot-2/cmd/home/fmt/string";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 PubSubClient client(server, 1883, wificlient);
19 void publishData();
20 const int trigpin=4;
21 const int echopin=5;
22 String command;
23 String data="";
24 long duration;
25 float dist;
26 void setup()
27 {
28   Serial.begin(115200);
29   pinMode(trigpin, OUTPUT);
30   pinMode(echopin, INPUT);
31   wifiConnect();
32   mqttConnect();
33 }
34 void loop() {
35   publishData();
36   delay(500);
37   if (!client.loop()) {
38     mqttConnect();

```

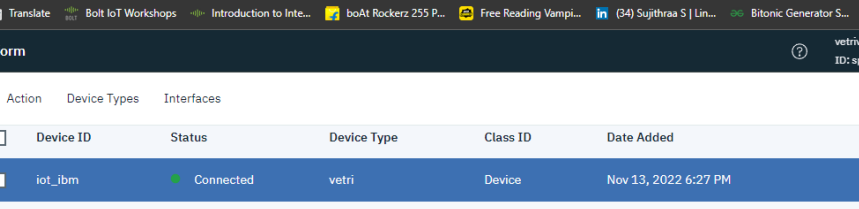
Simulation output log:

```

Publish OK
Sending payload: {"Alert distance":99.98}
Publish OK
Sending payload: {"Alert distance":99.98}
Publish OK

```

IBM cloud output:



The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays a table of devices, with one device 'vetri' selected. Below the device table, the 'Recent Events' tab is active, showing a table of events with columns for Event, Value, Format, and Last Received. The events listed are 'distance' with a value of '{"Alert distance":99.98}' in json format, received a few seconds ago. A status bar at the bottom indicates '0 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added
iot_ibm	Connected	vetri	Device	Nov 13, 2022 6:27 PM

Event	Value	Format	Last Received
distance	{"Alert distance":99.98}	json	a few seconds ago
distance	{"Alert distance":99.98}	json	a few seconds ago
distance	{"Alert distance":99.98}	json	a few seconds ago
distance	{"Alert distance":99.98}	json	a few seconds ago
distance	{"Alert distance":99.98}	json	a few seconds ago

0 Simulations running