

Assignment-4

Date	24 October 2022
Name	Naveen Kumar K
Roll Number	620119106058
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/348202729479340627>

Code:

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

WiFiClient wifiClient;

#define ORG "dhhnmy"
#define DEVICE_TYPE "naveen"
#define DEVICE_ID "ibm"
#define TOKEN "Nanee_Depp_3112"
#define speed 0.034

char server[] = ORG".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/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);
}
}

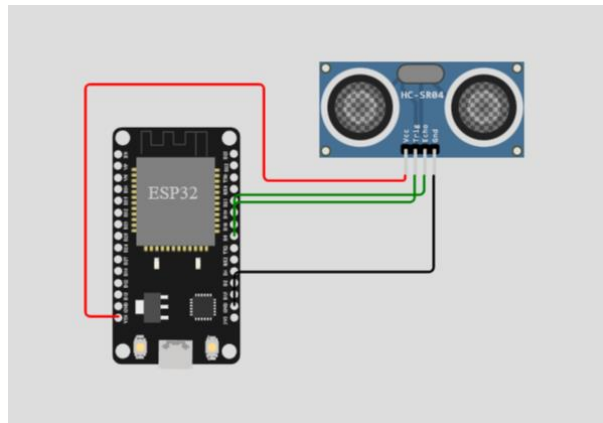
```

```

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

```

Diagram:



Wokwi Output:

The screenshot displays the Wokwi IDE interface. On the left, the sketch editor shows a C++ program that includes `WiFi.h` and `PubSubClient.h`. It defines an MQTT server, topic, and token, and implements a `publishData()` function that publishes distance data to an MQTT broker. The main loop calls `publishData()` every 500ms. On the right, the simulation window shows the circuit diagram of the ESP32 and HC-SR04 sensor. Below the diagram, the console output shows the following messages:

```

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

```

IBM cloud output:

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A user profile for 'naneedepp@gmail.com' with ID 'dhhnmy' is visible in the top right. The main content area shows a list of devices, with one device named 'ibm' selected. The device status is 'Connected', and it was added on 'Nov 13, 2022 2:16 PM'. Below the device list, a 'Recent Events' tab is active, showing a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are all 'status1' with a value of '["Alert distance":99.98]' or '["Alert distance":99.99]' in 'json' format, received 'a few seconds ago'. At the bottom, it indicates '0 Simulations running'.

Event	Value	Format	Last Received
status1	["Alert distance":99.98]	json	a few seconds ago
status1	["Alert distance":99.99]	json	a few seconds ago
status1	["Alert distance":99.99]	json	a few seconds ago
status1	["Alert distance":99.98]	json	a few seconds ago
status1	["Alert distance":99.98]	json	a few seconds ago

0 Simulations running