

Assignment 4

Name	Jagadeeswaran K
Team ID	PNT2022TMID38320
Project Name	Smart Waste Management System For Metropolitan Cities

Question:

Write a 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

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "2xip4t"
#define DEVICE_TYPE "Waran"
#define DEVICE_ID "Assignment"
#define TOKEN "87654321"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/waran/fmt/json"; char
topic[] = "iot-2/cmd/status/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=19;
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");
}
}

```

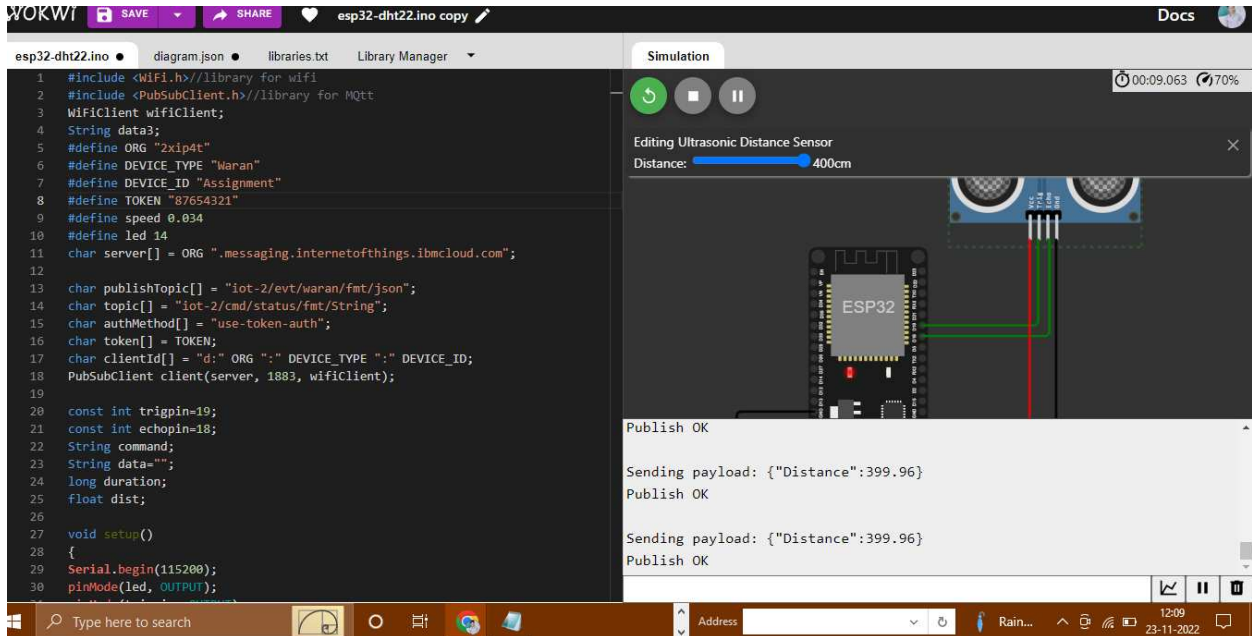
```

}
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");
}
}
}
}

```

Output: 1. When distance greater than 100 cm



IBM RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform dashboard. The user is logged in as 'waranjagadesh6@gmail.com' with ID '2xip4t'. The dashboard displays the 'Recent Events' for a device named 'Waran'. The events are listed in a table with columns: Event, Value, Format, and Last Received.

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

2. When distance less than 100 cm

The screenshot shows the Wokwi IDE interface. The left pane displays the code for 'esp32-dht22.ino'. The right pane shows a simulation of the hardware, including an ESP32 microcontroller and an HC-SR04 ultrasonic sensor. The console output shows the device sending a payload when the distance is less than 100 cm.

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "2xip4t"
6 #define DEVICE_TYPE "Waran"
7 #define DEVICE_ID "Assignment"
8 #define TOKEN "87654321"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12
13 char publishTopic[] = "iot-2/evt/waran/fmt/json";
14 char topic[] = "iot-2/cmd/status/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
20 const int trigpin=19;
21 const int echopin=18;
22 String command;
23 String data="";
24 long duration;
25 float dist;
26
27 void setup()
28 {
29   Serial.begin(115200);
30   pinMode(led, OUTPUT);

```

Simulation console output:

```

Publish OK
Sending payload: {"Alert Distance":89.95}
Publish OK
Sending payload: {"Alert Distance":89.95}
Publish OK

```

IBM RECENT EVENTS:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various system functions. The main content area shows a device named 'Waran' with a status of 'Connected'. Below this, a tabbed interface allows viewing 'Identity', 'Device Information', 'Recent Events' (selected), 'State', and 'Logs'. The 'Recent Events' tab displays a table of live data streams.

IBM Watson IoT Platform

waran jagadesh6@gmail.com
ID: 2xip4t

Browse Action Device Types Interfaces

Add Device +

Assignment Connected Waran Device Nov 22, 2022 8:46 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
waran	{"Alert Distance":89.95}	json	a few seconds ago
waran	{"Alert Distance":89.95}	json	a few seconds ago
waran	{"Alert Distance":89.95}	json	a few seconds ago
waran	{"Alert Distance":89.95}	json	a few seconds ago
waran	{"Alert Distance":89.96}	json	a few seconds ago

Type here to search

Address Rain... 12:09 23-11-2022