

IBM ASSIGNMENT – 4

Assignment Date	08 NOVEMBER 2022
Student Name	Jerusha Jaisal J L
Student Roll Number	710019106019
Team ID	PNT2022TMID42272

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.

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "2gttaf"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP-32"//Device type mentioned in ibm
watson IOT Platform
#define DEVICE_ID "DHT-22"//Device ID mentioned in ibm watson
IOT Platform
#define TOKEN "123456789" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
```

```

Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
Serial.print("Distance (cm): ");
Serial.println(distance);
if(distance<100)
{
Serial.println("ALERT!!");
delay(1000);
PublishData(distance);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(float dist) {
mqttconnect();
String payload = "{\"Distance\": ";
payload += dist;
payload += ", \"ALERT!!\": \"\" \"Distance less than 100cms\"";
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);

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

```

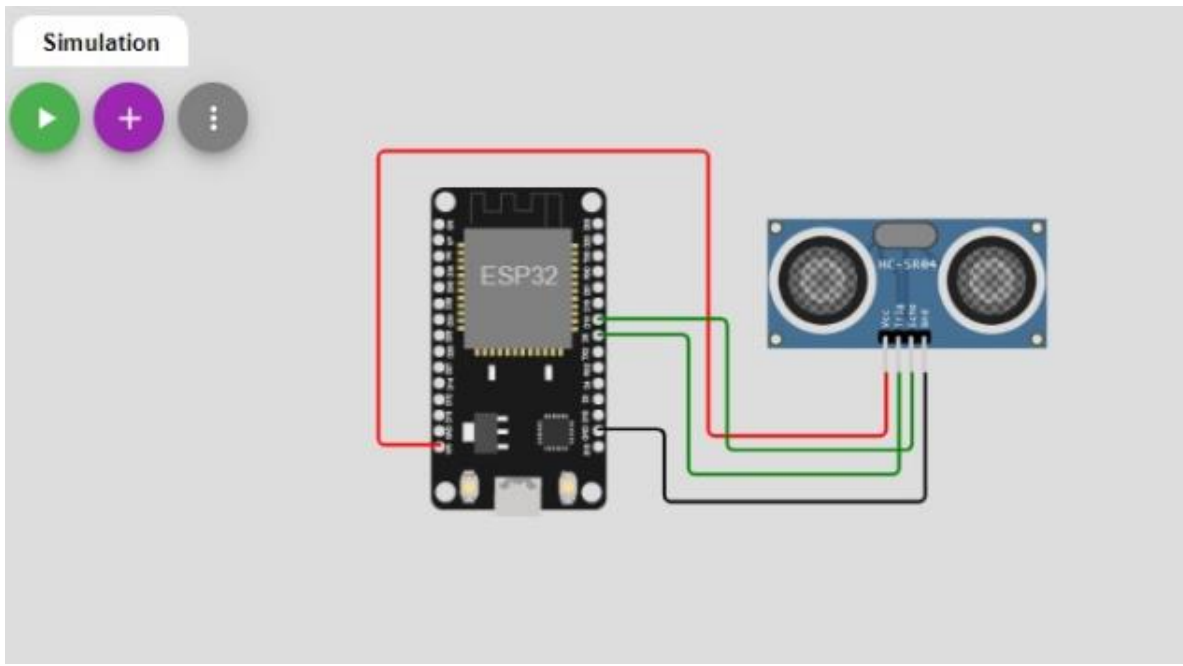
```

void mqttconnect() {
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
if (client.subscribe(subscribetopic)) {
Serial.println((subscribetopic));
Serial.println("subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (int i = 0; i < payloadLength; i++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
}

```

```
data3="";  
}
```

SCHEMATIC/CIRCUIT DIAGRAM:



WOKWI Output:

IBM Watson IoT Platform

710019106019@smartinternz.com
ID: 2gttaf

Browse Action Device Types Interfaces Add Device

Device ID	Status	Device Type	Class ID	Date Added
DHT-22	Connected	ESP-32	Device	8 Nov 2022 22:24

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	{"randomNumber":36}	json	a few seconds ago
event_1	{"randomNumber":15}	json	a few seconds ago
event_1	{"randomNumber":4}	json	a few seconds ago
Data	{"Distance":12.99,"ALERT!!":"Distance less than ...	json	a few seconds ago
event_1	{"randomNumber":32}		

1 Simulation running

Type here to search

22:36 08-11-2022

IBM CLOUD OUTPUT:

The screenshot displays a Wokwi project titled "esp32-dht22.ino copy". The code on the left is as follows:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribetopic, byte* payload, unsigned int
4 payloadLength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "2gttaf"//IBM ORGANITION ID
7 #define DEVICE_TYPE "ESP-32"//Device type mentioned in ibm watson IOT Platform
8 #define DEVICE_ID "DHT-22"//Device ID mentioned in ibm watson IOT Platform
9 #define TOKEN "123456789" //Token
10 String data3;
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Data/fmt/json";
13 char subscribetopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 WiFiClient wifiClient;
18 PubSubClient client(server, 1883, callback ,wifiClient);
19 const int trigPin = 5;
20 const int echoPin = 18;
21 #define SOUND_SPEED 0.034
22 long duration;
23 float distance;
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wifiConnect();
29   mqttConnect();
30 }
```

The simulation window on the right shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The terminal output displays the following messages:

```
ALERT!!
Sending payload: {"Distance":36.97,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 36.97
ALERT!!
Sending payload: {"Distance":36.97,"ALERT!!":"Distance less than 100cms"}
Publish ok
```

WOKWI LINK:

<https://wokwi.com/projects/347779108828086867>