

Sprint – 2 (NOTIFICATION)

Date	8 NOVEMBER 2022
Team ID	PNT2022TMID30936
Project Name	IoT Based Safety Gadget for Child Safety Monitoring and Notification.

In sprint 2 As a user, I should be able to notify my parent and guardian in emergency situations

Code:

```
#include <WiFi.h>

#include <PubSubClient.h>

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

#define ORG "jk50ow"

#define DEVICE_TYPE "python"

#define DEVICE_ID "py1"

#define TOKEN "-lY3WeW2q!fQFywml0"

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++) {
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
}

```

URL :

<https://wokwi.com/projects/348105244660466260>

ULTRASONIC SENSOR OUTPUT :

The screenshot shows the Wokwi IDE interface. On the left, the sketch.ino file contains the following code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int
4   payloadLength);
5 #define ORG "jk50ow"
6 #define DEVICE_TYPE "python"
7 #define DEVICE_ID "py1"
8 #define TOKEN "-lY3Wb2q!fQFyvmIo"
9 String data3;
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 WiFiClient wifiClient;
17 PubSubClient client(server, 1883, callback, wifiClient);
18 const int trigPin = 5;
19 const int echoPin = 18;
20 #define SOUND_SPEED 0.034
21 long duration;
22 float distance;
23 void setup() {
24   Serial.begin(115200);
25   pinMode(trigPin, OUTPUT);
26   pinMode(echoPin, INPUT);
27   wifiConnect();
28   mqttConnect();
29 }
30 void loop()
```

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

```
ALERT!!
Sending payload: {"Distance":99.98,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 99.98
ALERT!!
Sending payload: {"Distance":99.98,"ALERT!!":"Distance less than 100cms"}
Publish ok
```

IBM CLOUD STATUS :

The screenshot shows the IBM Watson IoT Platform dashboard. The device 'py1' is listed as 'Connected' with a status of 'python'. The 'Recent Events' tab is selected, showing a table of events:

Event	Value	Format	Last Received
Data	{"Distance":99.98,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":99.98,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":99.94,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":99.98,"ALERT!!":"Distance less than ...	json	a few seconds ago

At the bottom of the dashboard, a status message indicates '1 Simulation running'.

