

## Assignment -4

Assignment Date	26 October 2022
Team ID	PNT2022TMID45857
Project Name	Project - IoT based safety gadget for Child Safety Monitoring and Notification
Name	Vinitha C V

### QUESTION:

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. Upload document with wokwi share link and images of ibm cloud.

### CODE:

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

WiFiClient wifiClient;
#define ORG "fvdupc"

#define DEVICE_TYPE "abcd"
#define DEVICE_ID "rasp"
#define TOKEN "12345678" #define
speed 0.034
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char publishTopic[]
= "iot-2/evt/abcd_1/fmt/json"; char topic[] = "iot-
2/cmd/home/fmt/String"; char authMethod[] = "use-tokenauth";
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="";
String lat="14.167589";
String lon="80.248510";
String name="point2";
String
icon=""; long
duration; int
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(1000);
        }
        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){ dist=100-
    dist; icon="fatrash";

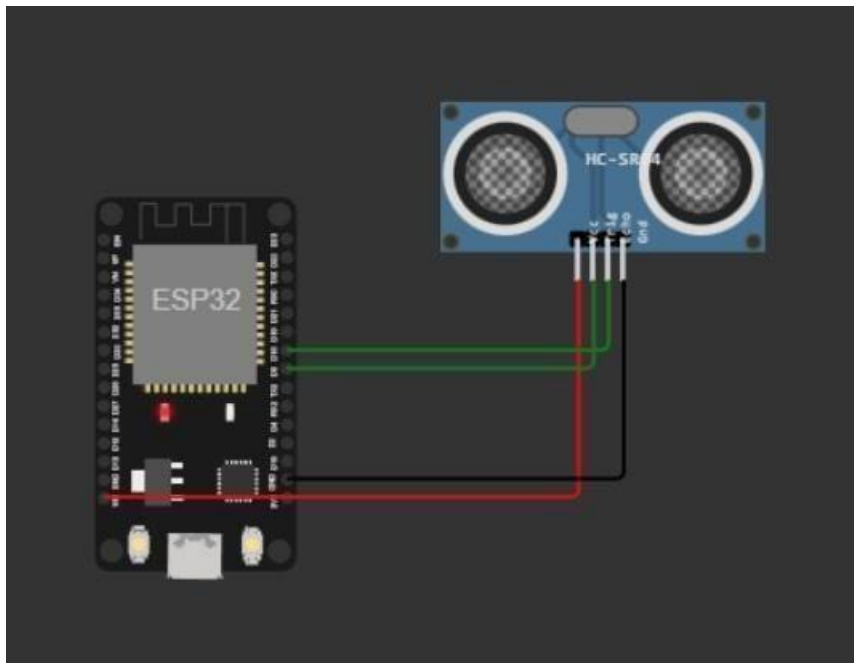
```

```

}else{ dist=0;
    icon="fa-trasho";
}
DynamicJsonDocument doc(1024);
String payload;
doc["Name"]=name;
doc["Latitude"]=lat;
doc["Longitude"]=lon;
doc["Icon"]=icon;
doc["FillPercent"]=dist;
serializeJson(doc, payload);
delay(3000);
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");
}
}
}

```

## CONNECTIONS:



## OUTPUT:

WOKWI

SAVE

SHARE

Docs

sketch.ino

diagram.json

libraries.txt

Library Manager

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

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

WiFiClient wificlient;

#define ORG "fvdupc"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "rasp"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/abcd_1/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="";
String lat="14.167589";
String lon="80.248510";
String name="point2";
String icon="";

long duration;
int dist;

void setup()
{

```

Simulation

00:06.799

Restart the simulation



The diagram shows an ESP32 microcontroller board connected to an HC-SR04 ultrasonic sensor. The sensor's VCC pin is connected to the ESP32's 5V pin, and its GND pin is connected to the ESP32's GND pin. The sensor's Trig pin is connected to the ESP32's pin 5, and its Echo pin is connected to the ESP32's pin 18.

subscribe to cmd OK

Sending payload:  
{ "Name": "point2", "Latitude": "14.167589", "Longitude": "80.248510", "Icon": "fa-trash-o", "FillPercent": 0 }

Publish OK