

Assignment -4

| | |
|-----------------|--|
| Assignment Date | 01 November 2022 |
| Team ID | PNT2022TMID45219 |
| Title name | Industry specific intelligent fire management system |
| Maximum Marks | 2 Marks |

Question-1:

Write code and connections in wokwi for the ultrasonic sensor.

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

Upload document with wokwi share link and images of IBM cloud

Solution :

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

WiFiClient wifiClient;

#define ORG "nhpwjc"
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "123456789"
#define speed 0.034

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

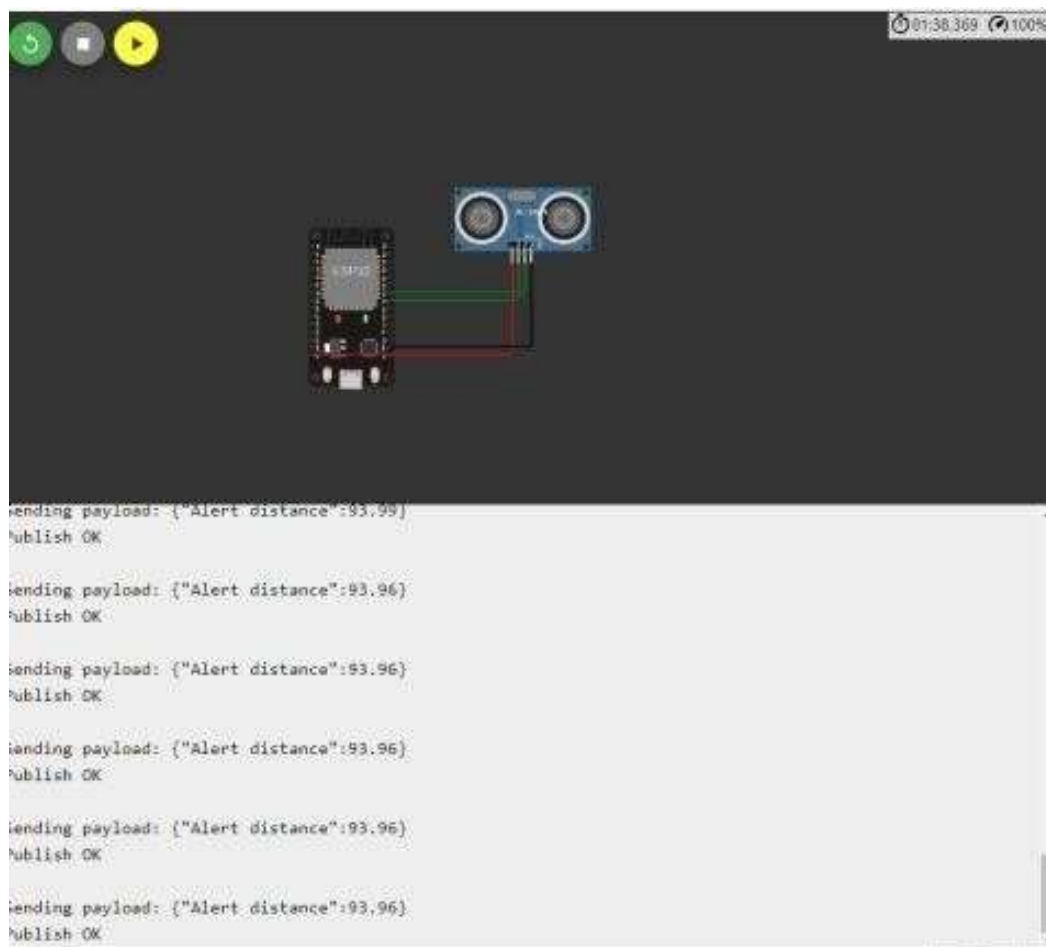
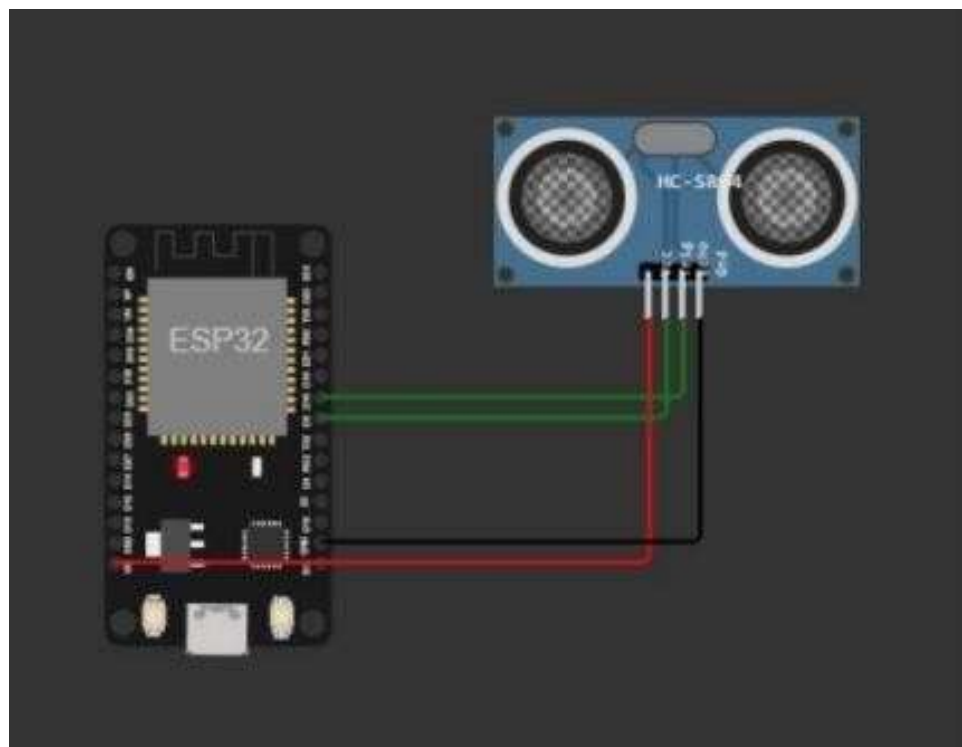
void mqttConnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting MQTT client to "); Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(1000);
        }
        initManagedDevice();
        Serial.println();
    }
}

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){
        DynamicJsonDocument doc(1024);
        String payload;
        doc["AlertDistance:"]=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");
        }
    }
}
}

```



NOTE: The above symptoms of dissonance that have been noted, it can be fully explained and resolved using different criteria. To get started, you can get help by using the 40/20/40 rule, by using 40/20.

 Springer by Nature | 80

bioRxiv preprint doi: <https://doi.org/10.1101/000000>; this version posted January 1, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



The screenshot shows the AWS IoT console interface. At the top, there's a navigation bar with tabs for 'Device ID', 'Status', 'Device Type', 'Class ID', 'Data Address', 'Descriptive Location', 'Added By', 'Device Class', and 'Provisioning Location'. Below this, a header bar displays the device ID '123456', its status 'Connected', device type 'NodeMCU', class ID 'Class', data address '01.01.0001.00.00', and the user '123456123456@amazon.com'. A 'Log' button is on the right.

Below the header, there's a section titled 'Identity' and 'Device Information'. A message states: 'This device is not yet active. It is in the process of being activated and will be active soon.' Below this message is a table of events.

| Event | Steps | Reason | Log Description |
|-------|--|--------|---------------------------|
| Start | ["Start", "Connect", "01.01.0001.00.00"] | start | A new device is starting. |
| Stop | ["Stop", "Disconnect", "01.01.0001.00.00"] | stop | A new device is stopping. |
| Start | ["Start", "Connect", "01.01.0001.00.00"] | start | A new device is starting. |
| Stop | ["Stop", "Disconnect", "01.01.0001.00.00"] | stop | A new device is stopping. |
| Start | ["Start", "Connect", "01.01.0001.00.00"] | start | A new device is starting. |
| Stop | ["Stop", "Disconnect", "01.01.0001.00.00"] | stop | A new device is stopping. |

At the bottom, there's a status bar showing '123456123456' and a 'Log' button.