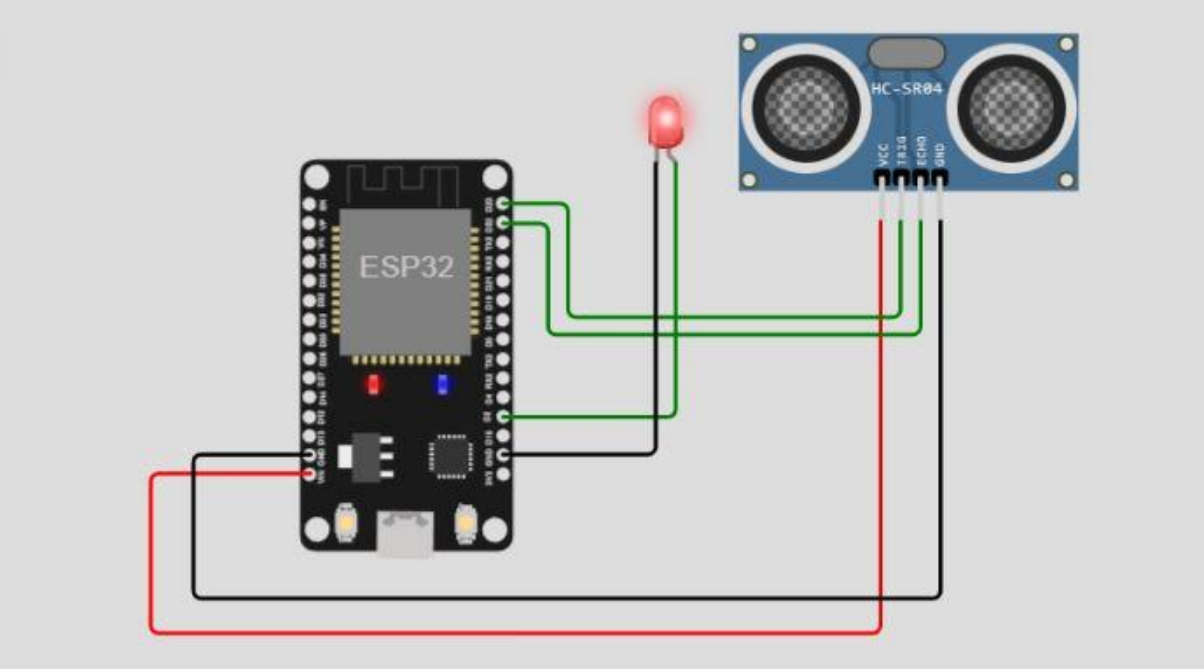
Assignment 3

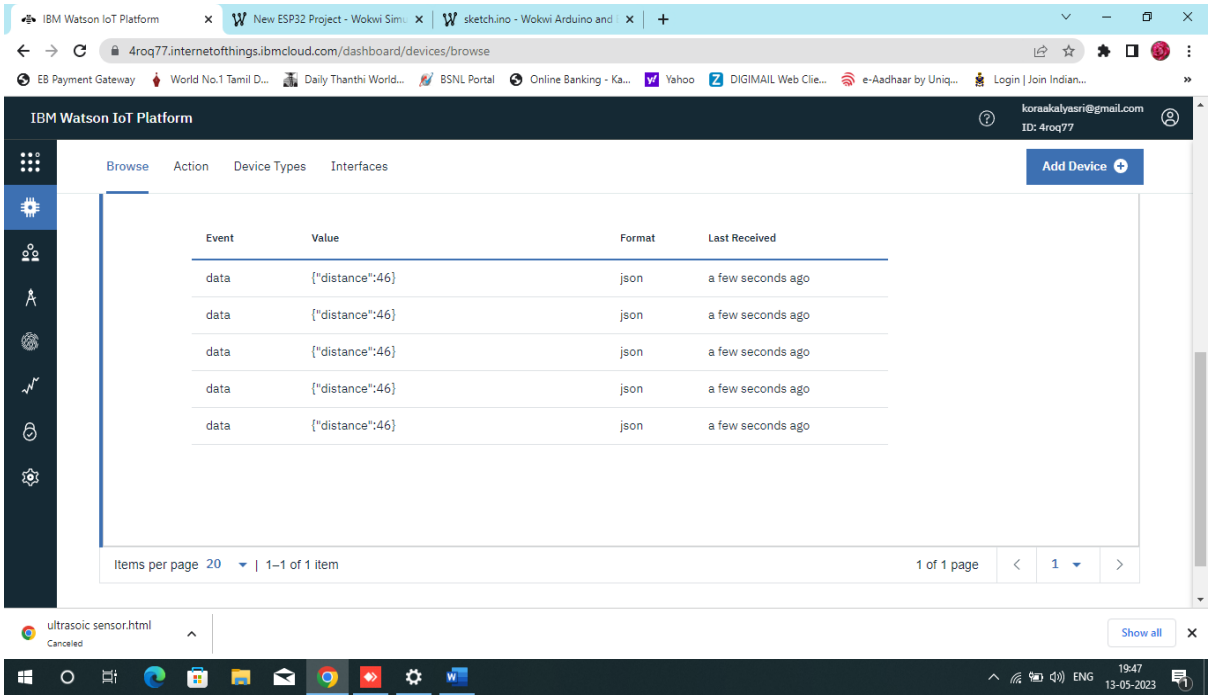
Name: Akalyasri D

Reg No: 711620106001

https://wokwi.com/projects/364410485265755137




```
Sending payload: {"distance":46.00}
Publish ok
Distance: 46.00
Sending payload: {"distance":46.00}
Publish ok
Distance: 46.00
Sending payload: {"distance":46.00}
Publish ok
Distance: 46.00
Sending payload: {"distance":46.00}
Publish ok
Distance: 46.00
Sending payload: {"distance":46.00}
Publish ok
Distance: 46.00
Sending payload: {"distance":46.00}
Publish ok
Distance: 46.00
Sending payload: {"distance":46.00}
Publish ok
```



Sketch. Ino :

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "Ultrasonic.h"

#define TRIG_PIN 23
#define ECHO_PIN 22
#define LED_PIN 2

Ultrasonic ultrasonic(TRIG_PIN, ECHO_PIN);

// IBM Watson IoT Platform credentials
#define ORG "4roq77"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "1234"
#define TOKEN "12345678"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);

void setup() {
```

```
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  wificonnect();
  mqttconnect();
}

void loop() {
  float distance = ultrasonic.read();
  Serial.print("Distance: ");
  Serial.println(distance);

  if (distance < 100) {
    String payload = "{\"distance\":";
    payload += distance;
    payload += "}";
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Publish ok");
      digitalWrite(LED_PIN, HIGH);
      delay(500);
      digitalWrite(LED_PIN, LOW);
    } else {
      Serial.println("Publish failed");
    }
  }

  if (!client.loop()) {
    mqttconnect();
  }
  delay(500);
}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Connecting to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    Serial.println("connected");
  }
}

void wificonnect() {
  Serial.print("Connecting to WiFi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
```

```
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Diagram.json:

```
 "version": 1,
  "author": "Akalyasri Durairaj",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0,
"attrs": {} },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -61.43,
      "left": 218.83,
      "attrs": { "distance": "51" }
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -37.87,
      "left": 162.73,
      "attrs": { "color": "red" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "esp:VIN", "ultrasonic1:VCC", "red", [ "h-79.8", "v79.51", "h-6.67" ] ],
    [ "esp:GND.2", "ultrasonic1:GND", "black", [ "h-58.47", "v71.68",
"h382.67" ] ],
    [ "esp:D23", "ultrasonic1:TRIG", "green", [ "h32.56", "v56.68", "h162.67"
] ],
    [ "esp:D22", "ultrasonic1:ECHO", "green", [ "h22.56", "v55.34", "h186.67"
] ],
    [ "led1:A", "esp:D2", "green", [ "v0" ] ],
    [ "led1:C", "esp:GND.1", "black", [ "v0" ] ]
  ],
  "dependencies": {}
}
```