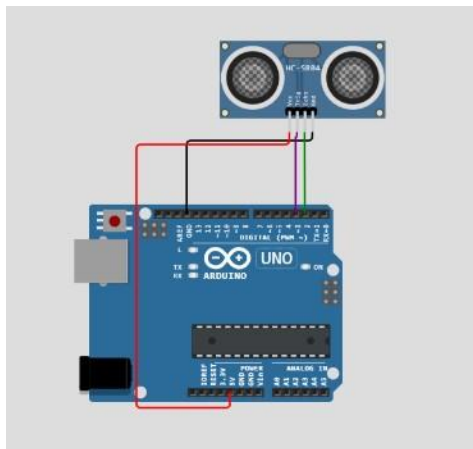


Assignment 4

Name	Kanisha R
Date	07 November 2022
Team ID	PNT2022TMID12941
Title	Signs with smart connectivity for better road safety

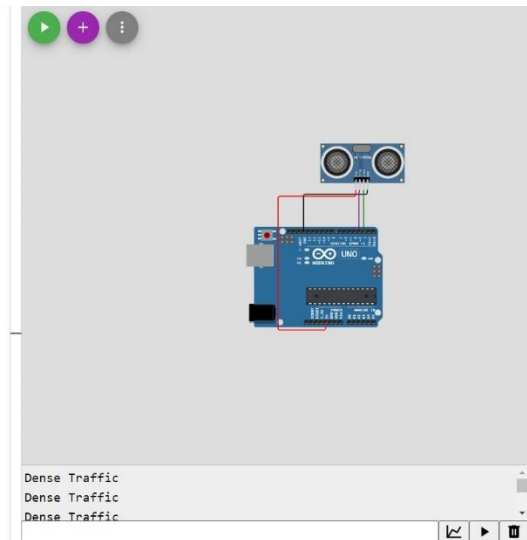
Simulation in wowki website:

Simulation circuit:

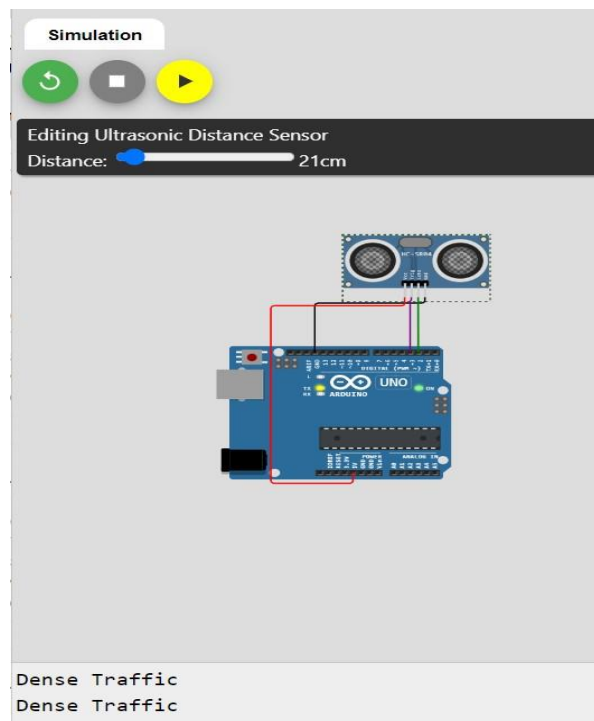
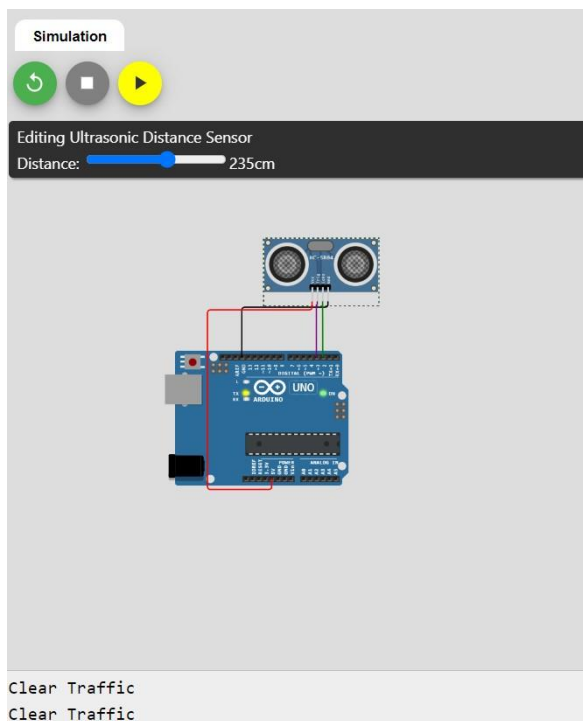


Simulation window:

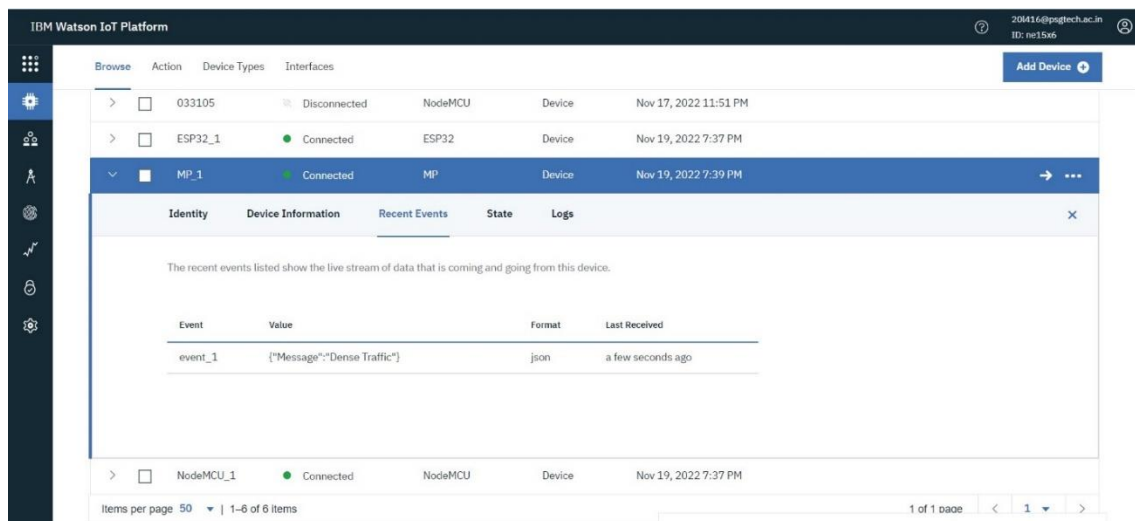
```
8  #include <WiFi.h>
9  #include <PubSubClient.h>
10 void callback(char* topic, byte* payload, unsigned int payloadLength);
11 //IBM credentials//
12 #define ORG "ne15x6"
13 #define DEVICE_TYPE "Mp"
14 #define DEVICE_ID "0331"
15 #define TOKEN "03082000"
16 String data3;
17 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
18 char publishTopic[] = "iot-2/evt/Data/fmt/json";
19 char subscribTopic[] = "iot-2/cmd/test/fmt/String";
20 char authMethod[] = "use-token-auth";
21 char token[] = TOKEN;
22 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
23 WiFiClient wificlient;
24 PubSubClient client(server, 1883, callback, wificlient);
25 //Main code//
26 #define ECHO_PIN 2
27 #define TRIG_PIN 3
28
29 void setup() {
30   Serial.begin(115200);
31   pinMode(TRIG_PIN, OUTPUT);
32   pinMode(ECHO_PIN, INPUT);
33 }
34
35 float readDistanceCM() {
36   digitalWrite(TRIG_PIN, LOW);
37   delayMicroseconds(2);
38   digitalWrite(TRIG_PIN, HIGH);
39   delayMicroseconds(10);
40   digitalWrite(TRIG_PIN, LOW);
41   int duration = pulseIn(ECHO_PIN, HIGH);
42   return duration * 0.034 / 2;
```



Output at wowki simulation tool:



Interfacing with IBM Watson IOT Platform:



C++ Code for wowki simulation:

```
#include <WiFi.h>
#include<PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
//IBM credentials//
#define ORG "ne15x6"
```

```

#define DEVICE_TYPE "MP"
#define DEVICE_ID "0331"
#define TOKEN "03082000"
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);
//Main code//
#define ECHO_PIN 2
#define TRIG_PIN 3

void setup() {
    Serial.begin(115200);
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
}

float readDistanceCM() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    int duration = pulseIn(ECHO_PIN, HIGH);
    return duration * 0.034 / 2;
}

void loop()
{
    float distance = readDistanceCM();
    if(distance<25)
    {
        Serial.println("Dense Traffic");
    }
    else
    {
        Serial.println("Clear Traffic");
    }
    delay(100);
}

```