# PUBLIC TRANSPORT OPTIMIZATION

**Wokwi URL**: https://wokwi.com/projects/379023477884978177

**Pyhton code**

sketch.ino#define BLYNK\_TEMPLATE\_ID "TMPL26V4fGv5q"

#define BLYNK\_TEMPLATE\_NAME "Test"

#define BLYNK\_AUTH\_TOKEN "XEHxNF\_Ur1Nt2p7wB5B20dNI1ZUwj34P"

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

int duration1 = 0;

int distance1 = 0;

int duration2 = 0;

int distance2 = 0;

int dis1 = 0;

int dis2 = 0;

int dis\_new1 = 0;

int dis\_new2 = 0;

int entered = 0;

int left = 0;

int inside = 0;

#define LED 2

#define PIN\_TRIG1 15

#define PIN\_ECHO1 14

#define PIN\_TRIG2 13

#define PIN\_ECHO2 12

BlynkTimer timer;

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Wokwi-GUEST";   // your network SSID (name)

char pass[] = "";

#define BLYNK\_PRINT **Serial**

long get\_distance1() {

  // Start a new measurement:

  digitalWrite(PIN\_TRIG1, HIGH);

  delayMicroseconds(10);

  digitalWrite(PIN\_TRIG1, LOW);

  // Read the result:

  duration1 = pulseIn(PIN\_ECHO1, HIGH);

  distance1 = duration1 / 58;

  return distance1;

}

long get\_distance2() {

  // Start a new measurement:

  digitalWrite(PIN\_TRIG2, HIGH);

  delayMicroseconds(10);

  digitalWrite(PIN\_TRIG2, LOW);

  // Read the result:

  duration2 = pulseIn(PIN\_ECHO2, HIGH);

  distance2 = duration2 / 58;

  return distance2;

}

void myTimer() {

**Serial**.println("100");

  dis\_new1 = get\_distance1();

  dis\_new2 = get\_distance2();

  if (dis1 != dis\_new1 || dis2 != dis\_new2){

**Serial**.println("200");

    if (dis1 < dis2){

**Serial**.println("Enter loop");

      entered = entered + 1;

      inside = inside + 1;

      digitalWrite(LED, HIGH);

      Blynk.virtualWrite(V0, entered);

      Blynk.virtualWrite(V2, inside);

      dis1 = dis\_new1;

      delay(1000);

      digitalWrite(LED, LOW);

    }

    if (dis1 > dis2){

**Serial**.println("Leave loop");

      left = left + 1;

      inside = inside - 1;

      Blynk.virtualWrite(V1, left);

      Blynk.virtualWrite(V2, inside);

      dis2 = dis\_new2;

      delay(1000);

    }

  }

}

 void setup() {

**Serial**.begin(115200);

  pinMode(LED, OUTPUT);

  pinMode(PIN\_TRIG1, OUTPUT);

  pinMode(PIN\_ECHO1, INPUT);

  pinMode(PIN\_TRIG2, OUTPUT);

  pinMode(PIN\_ECHO2, INPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 8080);

  timer.setInterval(1000L, myTimer);

}

void loop() {

  Blynk.run();

  timer.run();

}

Diagram.json{

  "version": 1,

  "author": "Godwin",

  "editor": "wokwi",

  "parts": [

    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -24.1, "left": 484.6, "attrs": {} },

    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -27.3, "left": -32.9, "attrs": {} },

    { "type": "wokwi-hc-sr04", "id": "ultrasonic2", "top": -27.3, "left": 178.3, "attrs": {} },

    {

      "type": "wokwi-led",

      "id": "led1",

      "top": 178.8,

      "left": 349.4,

      "attrs": { "color": "red" }

    }

  ],

  "connections": [

    [ "esp:TX0", "$serialMonitor:RX", "", [] ],

    [ "esp:RX0", "$serialMonitor:TX", "", [] ],

    [ "esp:GND.1", "led1:C", "black", [ "h47.7", "v143.9", "h-268.8", "v-19.2" ] ],

    [ "led1:A", "esp:D22", "green", [ "v19.2", "h297.6", "v-240", "h-28.8" ] ],

    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v67.2", "h451.2" ] ],

    [ "ultrasonic2:VCC", "esp:VIN", "red", [ "v67.2", "h240" ] ],

    [ "ultrasonic2:GND", "esp:GND.2", "black", [ "v57.6", "h210" ] ],

    [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v57.6", "h421.2" ] ],

    [ "ultrasonic1:TRIG", "esp:D15", "green", [ "v124.8", "h594.8", "v-76.8" ] ],

    [ "ultrasonic1:ECHO", "esp:D14", "green", [ "v86.4", "h412", "v-57.6", "h-19.2" ] ],

    [ "ultrasonic2:TRIG", "esp:D13", "green", [ "v0" ] ],

    [ "ultrasonic2:ECHO", "esp:D12", "green", [ "v38.4", "h220" ] ]

  ],

  "dependencies": {}

}

Libraries.txt