PHASE 3 - PUBLIC TRANSPORT OPTIMIZATION

INTRODUCTION

Optimizing public transport using Arduino is an exciting project. Simulating the behavior of hardware components like ultrasonic sensors in software can be done through virtual platforms. In this phase the system includes components such   as ESP32 Development Board, Ultrasonic Sensors (2x), LED, Blynk App, Resistors and Wires. Here are the steps to simulate code using Wokwi.

COMPONENTS

1. ESP32 Development Board : powerful microcontroller board.

2. Ultrasonic Sensors (2x): Ultrasonic sensors are used for distance measurement.

3. LED: An LED (Light Emitting Diode) is used as a visual indicator.

4.Resistors and Wires: Resistors to limit the current flowing through the LED (typically a 220-330 ohm resistor) and jumper wires to connect the components on the breadboard.

5.Blynk App: The Blynk app is required to visualize the data and control the system remotely.

PROCEDURE:

1. Software: Install Arduino IDE on your computer.

2. Board Support:

   - Open Arduino IDE.

- Add ESP32 board and Install ESP32 board package via Tools > Board > Boards Manager.

- Select "ESP32 Dev Module".

3.Upload Code:

- Copy and paste your modified code.

- Click Upload to compile and upload the code to your ESP32 board.

```
#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();
}
```

3.Hardware Setup:

- Connect ultrasonic sensors to PINs 15, 14, 13, and 12 for TRIG1, ECHO1, TRIG2, and ECHO2 respectively.

- Connect an LED's long leg to PIN 2 and short leg to ground.

4. Blynk Setup:

- Download Blynk app on your smartphone and Create a new project.

- Set device to "ESP32 Dev Board".

- Obtain Auth Token via email and replace it.

5.Testing:

- Add widgets for Virtual Pins V0, V1, and V2 to visualize the entered count, left count, and inside count respectively.

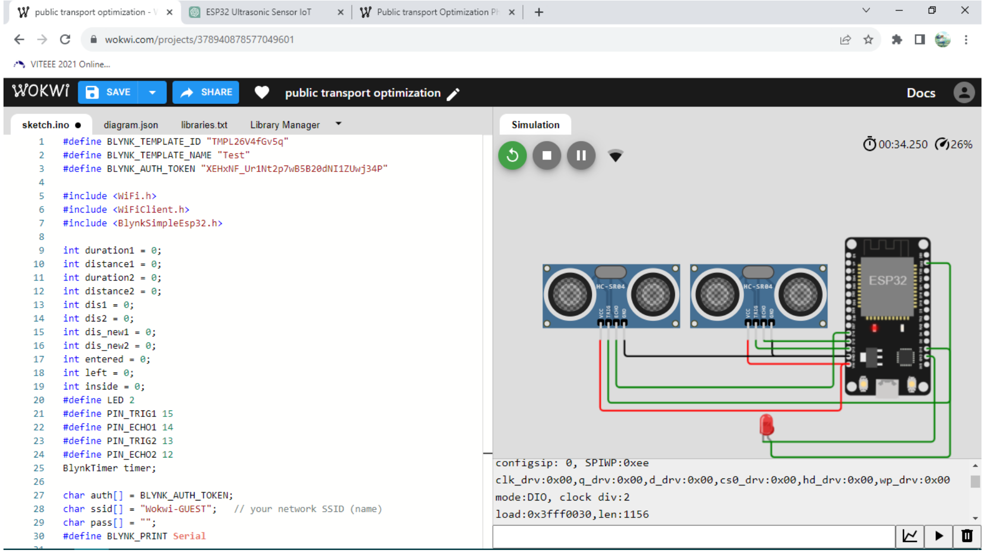
- LED on PIN 2 will indicate entry (ON) or exit (OFF).

- Check Serial Monitor (Tools > Serial Monitor) for debug messages and sensor readings.

Simulation

Code implementation: implemented the Arduino code to optimize the public transport level.

Simulation: clicked the “simulate” button in Wokwi to start the simulation.



```
#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
```

```
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
```

CONCLUSION

The presented project showcases a practical implementation of an object detection system using ESP32 microcontroller and ultrasonic sensors, facilitated by the Blynk IoT platform.

In conclusion, this project exemplifies the seamless integration of hardware, software, and IoT technologies to create an efficient object detection and counting system. By leveraging the power of the ESP32 microcontroller and Blynk platform, it exemplifies the potential of IoT solutions in real-world applications, offering a glimpse into the future of smart, responsive systems.