LED Bike Vest

A Project Report

FALL SEM 2022-23

ECS1001

(Engineering Clinics - Arduino using Embedded C - Project)

Submitted by:

NIRDESH SINGH 20BCE7062

VARIGONDA ANJANI GAYATHRI 20BCI7003

MANYA ARORA 20BCE7441

SAKETI BHAVANI 20BES7073

DEVARAPU KUSUMA 20BEC7074

ALAGAPPAN SP 20BCE7211

Under the guidance of-

Prof. Hari Kishan Kondaveeti,

Deputy Director, Engineering Clinics

VIT-AP UNIVERSITY,

AMARAVATI,

ANDHRA PRADESH, INDIA,

2022

Abstract

Nighttime bicycle riding is risky. Because bicycles lack indicators and lighting, making them less safe than trucks, cars, and motorbikes, bicyclists are particularly vulnerable. At night, motorists frequently miss bicyclists because they are too busy scanning the road for other vehicles. Therefore, accidents become more likely to happen to bikers.

Making yourself as conspicuous as you can when riding a bicycle is the answer. Cycling may be made more visible by wearing brightly coloured vests like those used by road construction workers.

We developed the "LED Bike Vest," an open-source Arduino turn signal bike safety vest, to address this problem. A tool called the Bike Suit is intended to increase a cyclist's visibility, especially at night. improving a cyclist's intent and interaction with other road users, such as vehicles and pedestrians.

Safety might be significantly improved with Light Vest, especially for those who ride at night.

Index

List of Figures

Introduction

LED Bike Vests are a staple of the cyclers. The main concept of the project is to find a safety solution for cyclers who ride on the streets to be distinguished by car and truck drivers. The open-source Arduino Turn Signal LED Bike Vest is designed to be a wearable technology.

It is designed specifically for individuals who practice cycling as a sport or use it as a method of transportation during the late hours at night when it is dark.

Background

Based on the prevailing science, there is a solution to every hassle that we're going through in our everyday lives.

LED Bike Vest is a light strip controlled by a microcontroller safely applied to a wearable fabric, such as a vest. From this point, we can build on this idea, making it weatherproof, lightweight, portable, and even communicate with other devices.

Problem Definition

Riding bicycles is dangerous at night. Bicyclists are especially at risk because bicycles lack indicators and headlights. The drivers in cars often don't see you at night— they are only looking for other cars. For that reason, there should be a safety solution with a technology system for the user to use to make them safer and practice their sport easily.

The solution is to make yourself as visible as possible when you're on a bike. To combat this issue, we are working on a wearable gadget called "LED Bike Vest", an open-source Arduino turn signal bike safety vest.

The Bike Suit is a device designed to improve a cyclist's visibility, particularly at night. It will help increase a cyclist's purpose and communication with other drivers and pedestrians on the street.

Objectives

The aim of this project as the name suggests that the vest is electrically designed for cyclers on the road to reduce serious and critical injures and accidents.

During the execution of the project of the Smart LED Bike Vest, we have had come up with detailed and transparent aims and objectives.

These include the following:

1. To implement a wearable jacket that has LED colors used for turning signals.
2. To alarm the cycler of the bike physically through a signal if an object is detected near the cycler or in a blind spot that is hard to identify.
3. Installing the LED that automatically turns on when there is not enough light.
4. The Bike Remote has four customizable push buttons. When a button is pressed, a value is sent to the Light Vest's Arduino, triggering the LED strip/ WS2812B to light up a specific way based on the particular value received.

Methodology/Procedure

For the project, we used the following components: -

- 2 x Arduino Nano
- MPU6050 (Accelerometer)
- LED Strip (WS2812B)
- Jumper Wires
- 2 x Bluetooth HC-05
- Rechargeable Battery, 9 V
- 2 x Slider Switch
- Capacitors
- Breadboard

Our project is basically divided into two parts:

1. Handle Circuit
2. Vest Circuit

The figure of Handle Circuit is attached below for reference:

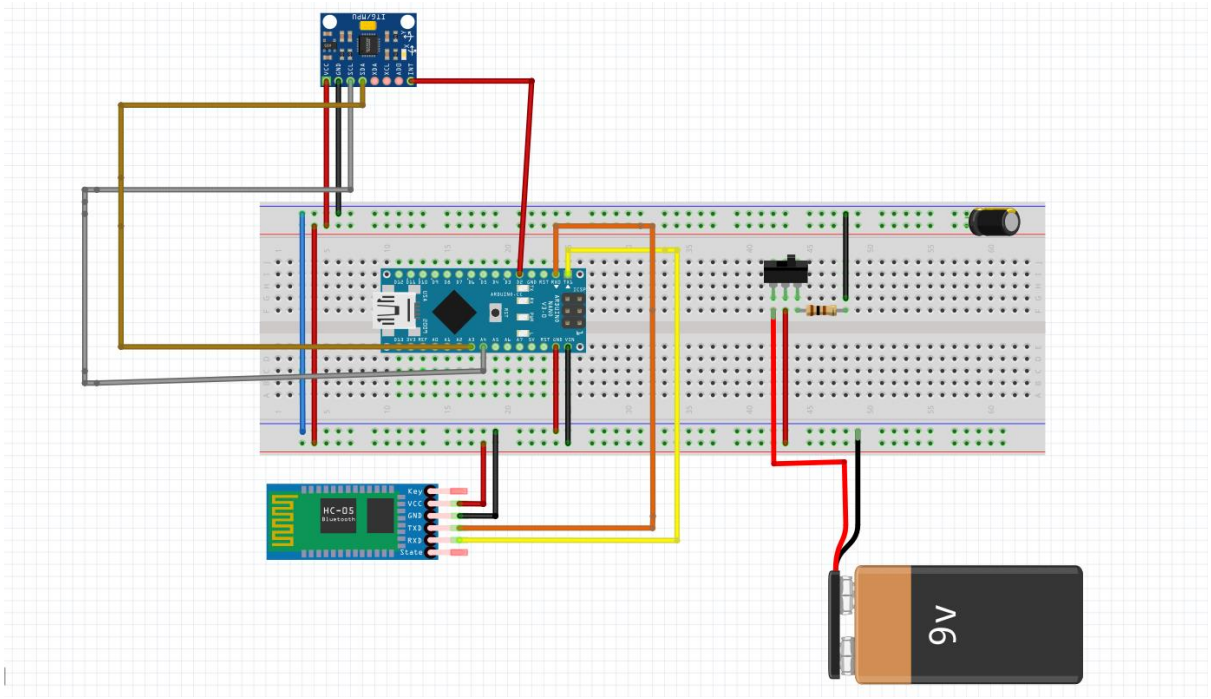

Figure 1: Handle Circuit Diagram

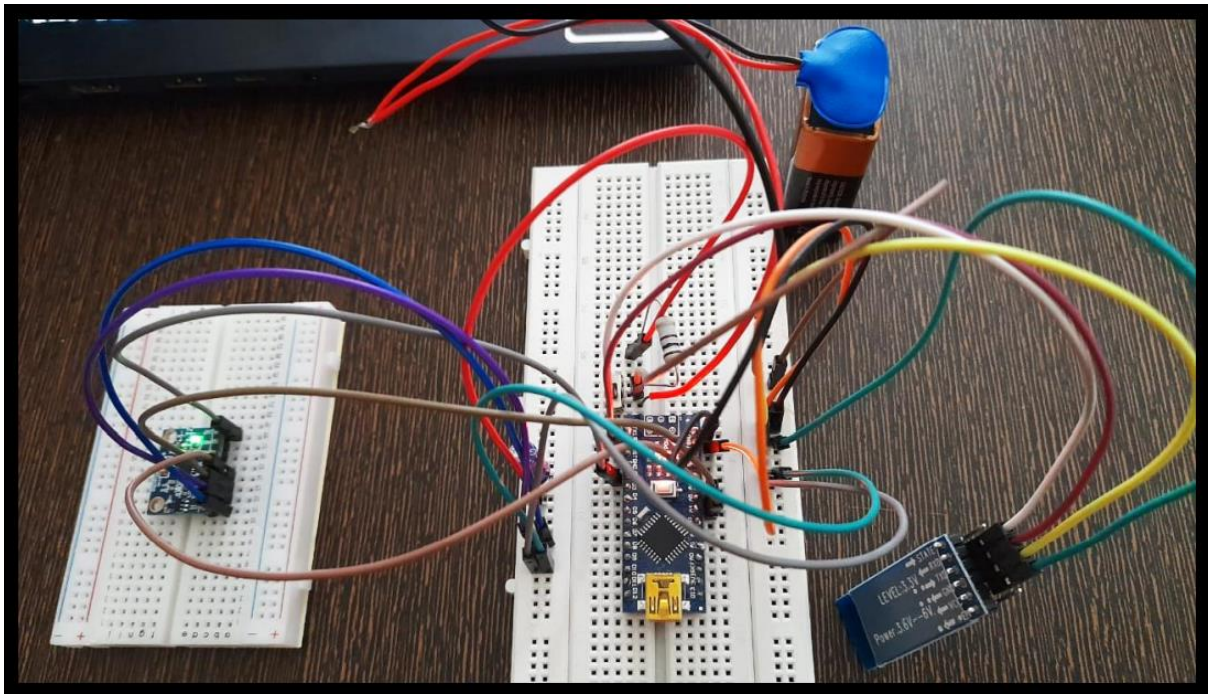

Figure 2: Handle Circuit

The handle circuit make use of the Nano Arduino, a HC-05 Bluetooth module and a Gyroscope (MPU6050 - Accelerometer). The circuit is powered through a 9V battery. Using the slider switch we can turn on/off the handle circuit.

The figure of Vest Circuit is attached below for reference:

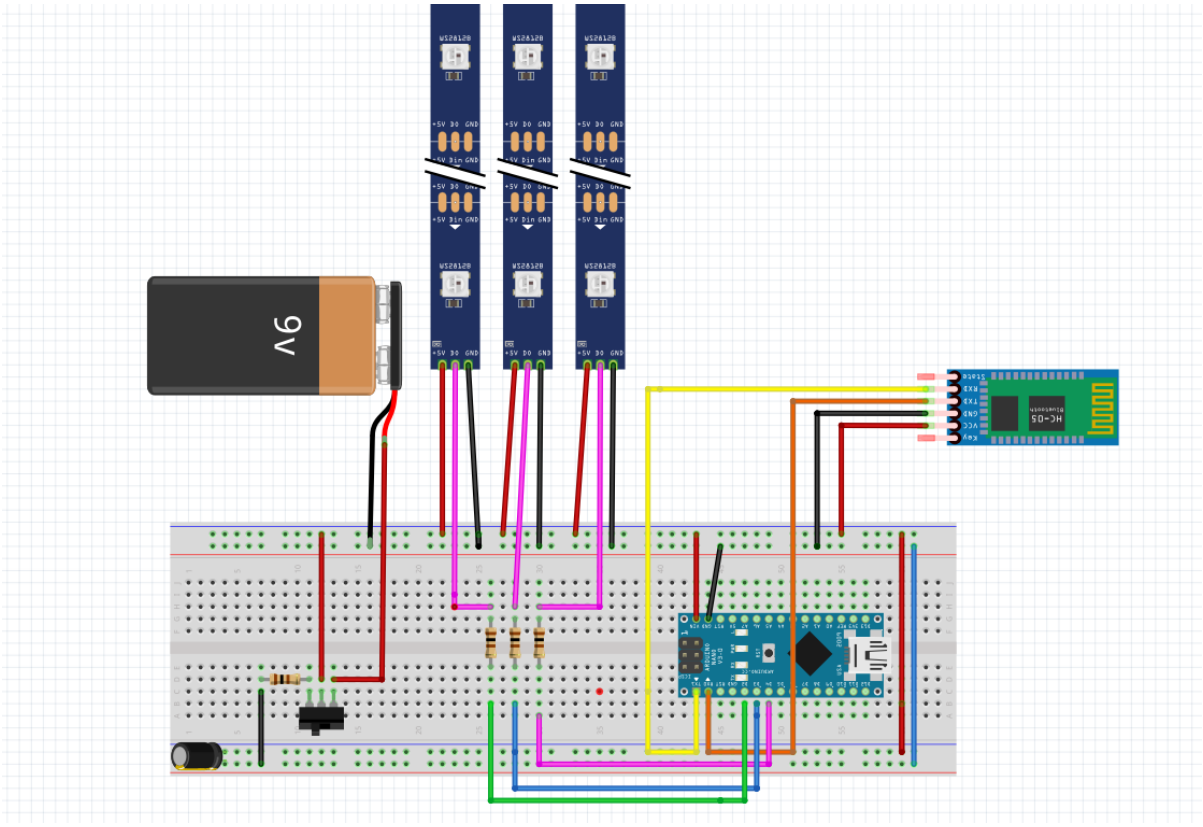

Figure 3: Vest Circuit Diagram

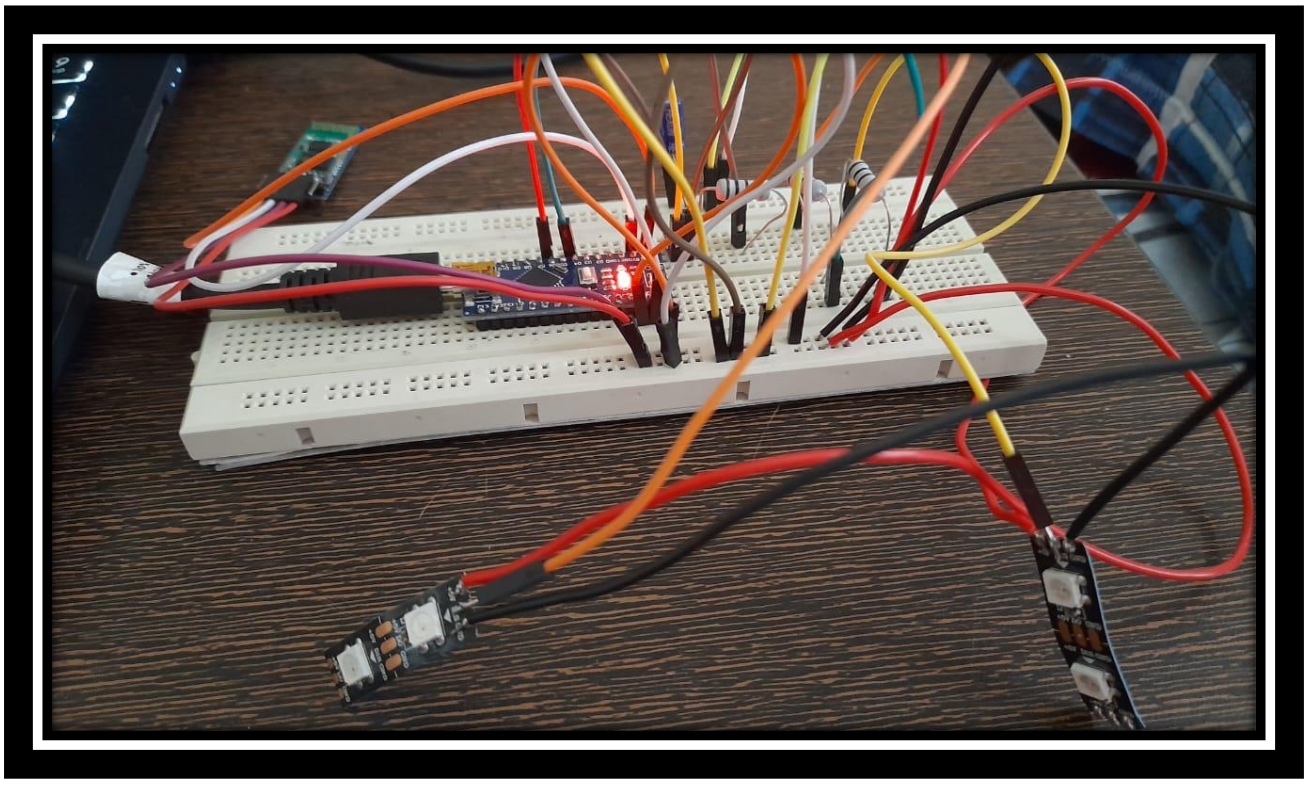

Figure 4: Vest Circuit

The vest circuit make use of the Nano Arduino, a HC-05 Bluetooth module and two Led Strips (WS2812B). The circuit is powered through a 9V battery. Using the slider switch we can turn on/off the handle circuit.

We have put both the Bluetooth modules in Master and Slave configuration. The Bluetooth module of handle circuit acts the master while the Bluetooth module of the vest circuit acts a slave. This is done to ensure the security of the product. If we don't ensure the master and slave configuration of the Bluetooth modules, then anyone can connect to the circuit using their smartphones' Bluetooth. However, putting the Bluetooth modules in master and slave configuration ensures that no one can connect his/her Bluetooth to the circuits Bluetooth modules. This ensures proper functioning of the both the circuits and makes the product secure.

The working of the project is described below:

1. Handle circuits is attached to the handle of the bicycle.
2. Vest circuit is attached to the vest of the cyclist.
3. When the person wants to go anywhere using the cycle at any time of the day, he has to turn on the handle as well as the vest circuits using the slider switches that are provided in the circuits.
4. When the cyclist has to turn, he will turn the cycle's handle right/left. As soon he turns the handle, the Gyroscope present in the handle circuits detects the change in direction of motion and it sends the data to the Nano Arduino which after processing feds 'L' or 'R' to the Master HC-05 Bluetooth Module. It sends these signals to Slave HC-05 Bluetooth Module present in vest circuit.
5. The slave Bluetooth in vest circuit receives 'L' or 'R' signals from the master Bluetooth module. The Bluetooth module feeds these signals to Nano Arduino of the vest circuit which after processing, lights up the right or the left LED Strips (if the cyclist turned right or left respectively) by sending the data through the data lines to the LED Strips.
6. This ensures that the cyclist's visibility increases at any time of the day (especially at night). So, any mishap can be avoided using this.
7. Finally, the cyclist can turn off both the circuits using the slider switches when he reaches home/destination.

Result and Discussion

Action: When cyclist turn the handlebar right.

Microcontroller: The Gyroscope detects the relative change in orientation to the Right. The value 'R' (representing Right) is sent to the LED Vest microcontroller via Bluetooth.

Result: The Right LED strip would light up.

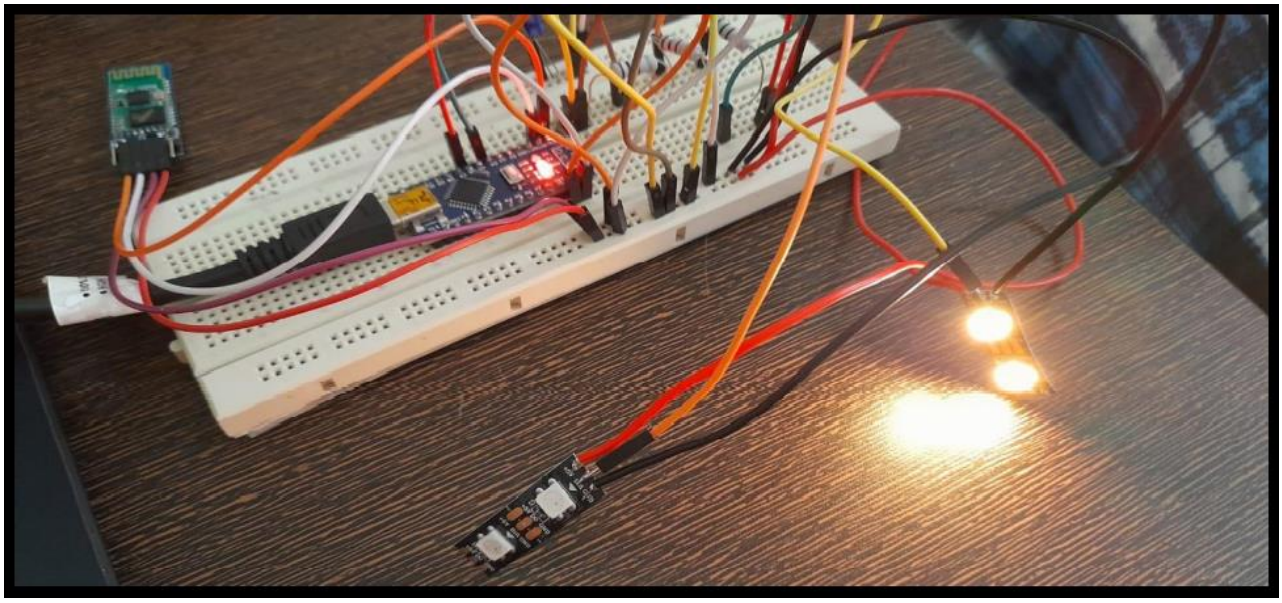

Figure 5: Right Blinking of the Vest Circuit

Action: When cyclist turn the handlebar left.

Microcontroller: The Gyroscope detects the relative change in orientation to the Left. The value ‘L’ (representing Left) is sent to the LED Vest microcontroller via Bluetooth.

Result: The Left LED strip would light up.

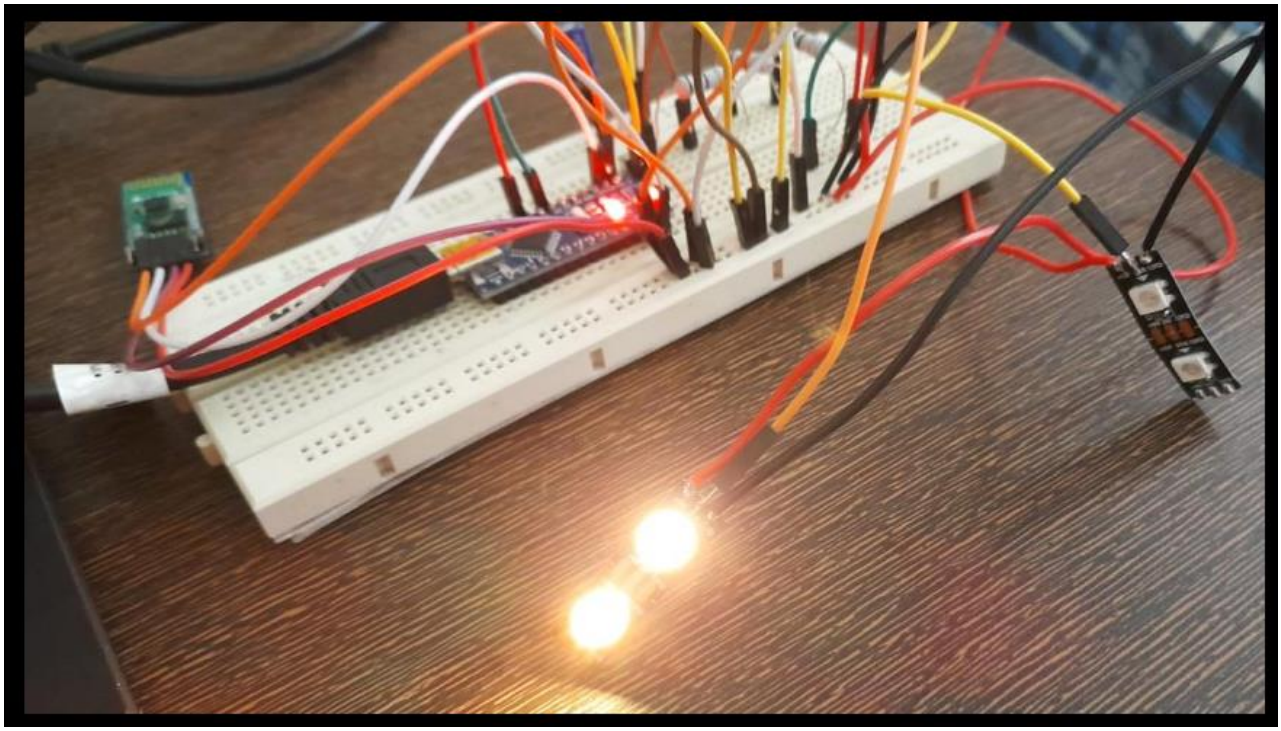

Figure 6: Left Blinking of the Vest Circuit

The working of the circuits can be seen by clicking on the following link:

https://youtu.be/gZhSDD2KSlw

Conclusion and Future Scope

Thus, the LED Bike Vest is a creative product for increasing a cyclist's visibility at night. The biker can communicate with other drivers and pedestrians on the road thanks to the controllable LED lights on the vest.

However, there are certain significant aspects that may be improved, such as wearable technology, smaller handle circuit, and modern LED strip layouts. By building a PCB to link the circuit's numerous components, it may be further improved. This facilitates component assembly on both sides and minimizes circuit space requirements.

We can also add other functionalities such as hazardous blinking during an emergency which will further increase the productivity of LED Bike Vest.

References

[1] https://smartbuilds.io/lightvest-arduino-led-bike-vest-tutorial/

[2] https://www.instructables.com/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/

Appendix

handle.ino

```
#include<Wire.h>

const int MPU_addr = 0x68;
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

int minVal = 265;
int maxVal = 402;

double x;
double y;
```

```
double z;

bool bool_caliberate = false;
int response_time = 400;

bool offset_pos = false;

void setup() {

  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true); Wire.begin();
  Serial.begin(9600);
  delay(1000);

}

void loop() {

  MPU_debug();
  GetMpuValue1(MPU_addr);

  if (offset_pos == false) {
    if ( GyZ > 15000) {
      Serial.println("L");
      offset_pos = true;
      delay(response_time);
    }
  }

  if (offset_pos == true) {
    if ( GyZ > 15000) {
      Serial.println("L");
      offset_pos = false;
      delay(response_time);
    }
  }

  if (offset_pos == true) {
    if ( GyZ < -15000) {
      Serial.println("R");
      delay(response_time);
      offset_pos = false;

    }
  }
```

```
  if (offset_pos == false) {
    if ( GyZ < -15000) {
      Serial.println("R");
      delay(response_time);
      offset_pos = true;

    }
  }

}

void GetMpuValue1(const int MPU) {

  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 14, true);

  AcX = Wire.read() << 8 | Wire.read();
  AcY = Wire.read() << 8 |  Wire.read();
  AcZ = Wire.read() << 8 | Wire.read();

  Tmp = Wire.read() << 8 | Wire.read();

  int xAng = map(AcX, minVal, maxVal, -90, 90);
  int yAng = map(AcY, minVal, maxVal, -90, 90);
  int zAng = map(AcZ, minVal, maxVal, -90, 90);

  GyX = Wire.read() << 8 | Wire.read();
  GyY = Wire.read() << 8 | Wire.read();
  GyZ = Wire.read() << 8 | Wire.read();
  x = RAD_TO_DEG * (atan2(-yAng, -zAng) + PI) + 4;
  y = RAD_TO_DEG * (atan2(-xAng, -zAng) + PI);
  z = RAD_TO_DEG * (atan2(-yAng, -xAng) + PI);

}

void MPU_debug() {

}
```

vest.ino

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
```

```
#endif

#define LED_LEFT    4
#define LED_RIGHT   6

#define LED_COUNT 4 // Define the number of LEDs in the strip

char state = 0;
int light_delay = 50;

Adafruit_NeoPixel strip_left(LED_COUNT, LED_LEFT, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel strip_right(LED_COUNT, LED_RIGHT, NEO_GRB + NEO_KHZ800);

void setup() {

#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
#endif

  strip_left.begin();
  strip_left.show();
  strip_left.setBrightness(150);

  strip_right.begin();
  strip_right.show();
  strip_right.setBrightness(150);
  Serial.begin(9600);
  delay(1000);
}

void loop() {

  if (Serial.available() > 0) {


    state = Serial.read();
    Serial.print(state);

    if (state == 'L') {
      leftBlink();
      delay(light_delay);

    }

    if (state == 'R') {
      rightBlink();
      delay(light_delay);

    }
```

```
  }

}


void colorWipeLeft(uint32_t color, int wait) {
  for (int i = 0; i < strip_left.numPixels(); i++) {
    strip_left.setPixelColor(i, color);
    strip_left.show();
    delay(wait);
    strip_left.clear();
  }
}


void rightBlink() {

  for (int i = 0; i < 4; i++) {

    colorBlinkersRight(strip_right.Color(255,   100,   0), 50); //Yellow
    delay(400);
    colorBlinkersRight(strip_right.Color(0,   0,   0), 25); //Yellow
    delay(300);

  }
}

void colorBlinkersRight(uint32_t c, int wait) {

  for(int i=0;i<2;i++) {
    strip_right.setPixelColor(i, c);
  }

  strip_right.show();
  delay(wait);
  strip_right.clear();
  delay(wait);

}

void hazardBlink() {

  for (int i = 0; i < 4; i++) {

    colorBlinkersLeft(strip_left.Color(255,   100,   0), 50); //Yellow
    colorBlinkersRight(strip_right.Color(255,   100,   0), 50); //Yellow

    delay(400);
```

```
    colorBlinkersLeft(strip_left.Color(0,   0,   0), 25); //Yellow
    colorBlinkersRight(strip_right.Color(0,   0,   0), 50); //Yellow

    delay(300);

  }
}


void leftBlink() {

  for (int i = 0; i < 4; i++) {

    colorBlinkersLeft(strip_left.Color(255,   100,   0), 50); //Yellow
    delay(400);
    colorBlinkersLeft(strip_left.Color(0,   0,   0), 25); //Yellow
    delay(300);

  }
}

void colorBlinkersLeft(uint32_t c, int wait) {

  for(int i=0;i<2;i++) {
    strip_left.setPixelColor(i, c);
  }


  strip_left.show();
  delay(wait);
  strip_left.clear();
  delay(wait);

}

void colorWipeRight(uint32_t color, int wait) {
  for (int i = 0; i < strip_right.numPixels(); i++) {
    strip_right.setPixelColor(i, color);
    strip_right.show();
    delay(wait);
    strip_right.clear();
  }
}

void rainbow_left(int wait) {

  for (long firstPixelHue = 0; firstPixelHue < 5 * 65536; firstPixelHue += 256) {
    for (int i = 0; i < strip_left.numPixels(); i++) {
```

```
    int pixelHue = firstPixelHue + (i * 65536L / strip_left.numPixels());

    strip_left.setPixelColor(i, strip_left.gamma32(strip_left.ColorHSV(pixelHue)));
   }
   strip_left.show();
   delay(wait);  // Pause for a moment
  }
}

void rainbow_right(int wait) {

  for (long firstPixelHue = 0; firstPixelHue < 5 * 65536; firstPixelHue += 256) {
   for (int i = 0; i < strip_right.numPixels(); i++) {

    int pixelHue = firstPixelHue + (i * 65536L / strip_right.numPixels());

    strip_right.setPixelColor(i, strip_right.gamma32(strip_right.ColorHSV(pixelHue)));
   }
   strip_right.show();
   delay(wait);  // Pause for a moment
  }
}

void clearLights() {
  strip_left.clear();
  strip_right.clear();
}
```