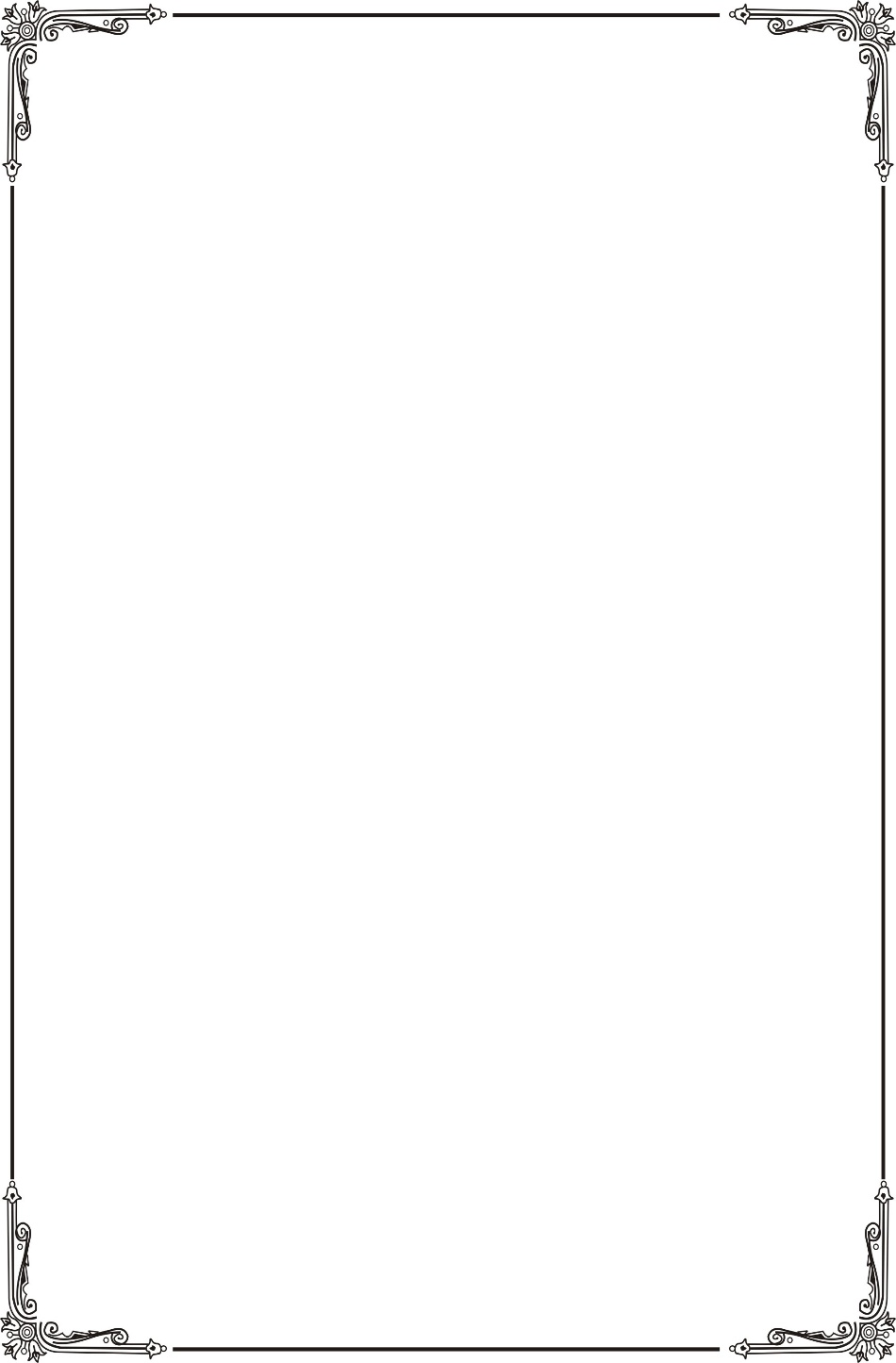
****

**CUỘC THI SÁNG TẠO THANH THIẾU NIÊN**

**NHI ĐỒNG QUỐC GIA**

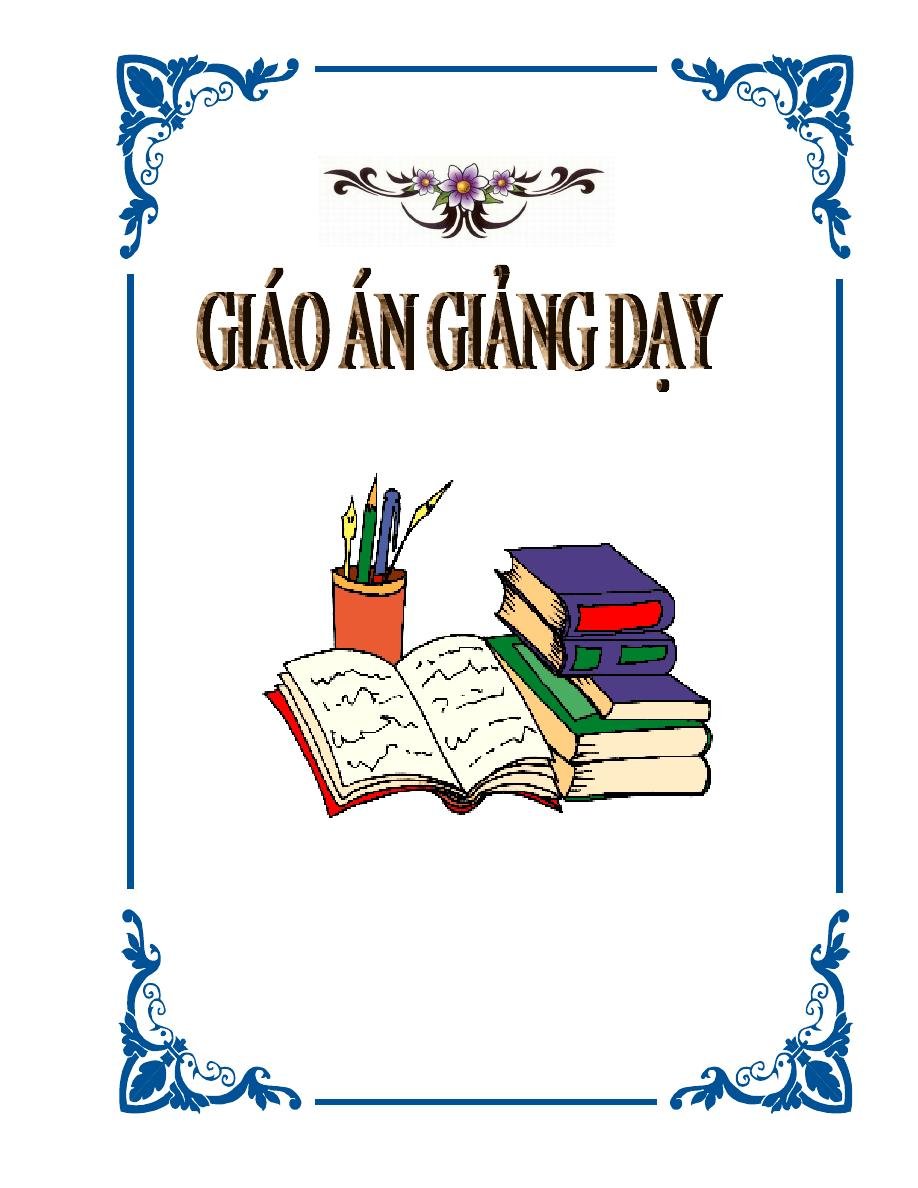
**Năm học 2023 - 2024**

**---------oOo--------**

**Tên dự án dự thi**

**“HỆ THỐNG CẢI TIẾN XE**

**THÔNG MINH DÙNG OPENCV”**

****

**Lĩnh vực dự thi: PHẦN MỀM TIN HỌC**

**BẢN THUYẾT MINH**

**I. Thông tin chung**

***1.Tên mô hình, sản phẩm dự thi***:

**HỆ THỐNG CẢI TIẾN XE THÔNG MINH DÙNG OPENCV**

***2. Lĩnh vực dự thi***:

Phần mềm tin học

***3. Tên tác giả:***

**3.1.**Bùi Đoàn Trung Hiếu - Lớp: 11A1 Trường THPT Trần Văn Ơn.

**3.2.** Nguyễn Dương Trọng An - Lớp: 10A1 Trường THPT Trần Văn Ơn.

**II. Thuyết minh giải pháp**

***1. Ý tưởng tạo ra mô hình, sản phẩm***

Hiện nay, các mẫu xe gắn máy đời mới đều được trang bị các đèn rất sáng và không thể tắt đèn (nếu không được thay đổi công tắc). Những người sử dụng xe gắn máy hiện nay còn có ý thức kém về việc bật tắt cos – pha của xe. Khi bật pha, đèn xe sẽ chiếu rất sáng có thể gây chói cho người đi đối diện, gây ra nhiều vụ tai nạn thương tiếc. Thêm vào nữa khi gặp tai nạn giao thông thì xe cũng sẽ không có cảnh báo hay báo động về cho người thân của người lái xe biết được.

Hiểu được vấn đề ấy và với mục đích nâng cao hình ảnh văn hóa tốt đẹp khi tham gia giao thông đường bộ và giảm những vụ tai nạn giao thông đường bộ xuống mức thấp nhất nhằm bảo vệ sức khỏe tốt nhất cho người tham gia giao thông đường bộ an toàn. Chính vì vậy, em đã có ý tưởng sáng tạo ra sản phẩm “**HỆ THỐNG CẢI TIẾN XE THÔNG MINH DÙNG OPENCV”** nhằm giải quyết những vấn đề trên.

***2. Trình bày tính mới, tính sáng tạo của sản phẩm***

Là nền tảng của hệ thống bật tắt cos – pha thông minh trên các dòng xe ô tô, xe gắn máy nhưng được cải tiến giảm giá thành và thêm các cảm biến để hỗ trợ người điều khiển xe.

Quy trình điều khiển đơn giản, trực quan, sinh động, dễ làm quen.

Việc ứng dụng OpenCV phân tích hình ảnh từ Camera nhận biết hướng xe chạy sẽ giảm thiểu được những vụ tai nạn giao thông không cần thiết và bảo vệ được người điều khiển xe.

Cảnh báo khi xe bị ngã hoặc người điều khiển gặp tai nạn để hạn chế những thiệt hại huy hiểm.

Tiết kiệm tuổi thọ acquy xe khi tự động bật đèn khi trời tối và sáng đèn khi trời sáng.

Nâng cao thương hiệu sản phẩm Việt Nam ở trong nước và quốc tế.

***3. Các nguyên vật liệu làm ra mô hình, sản phẩm:***

Hệ thống gồm các thiết bị trong mô hình & thực tế như sau:

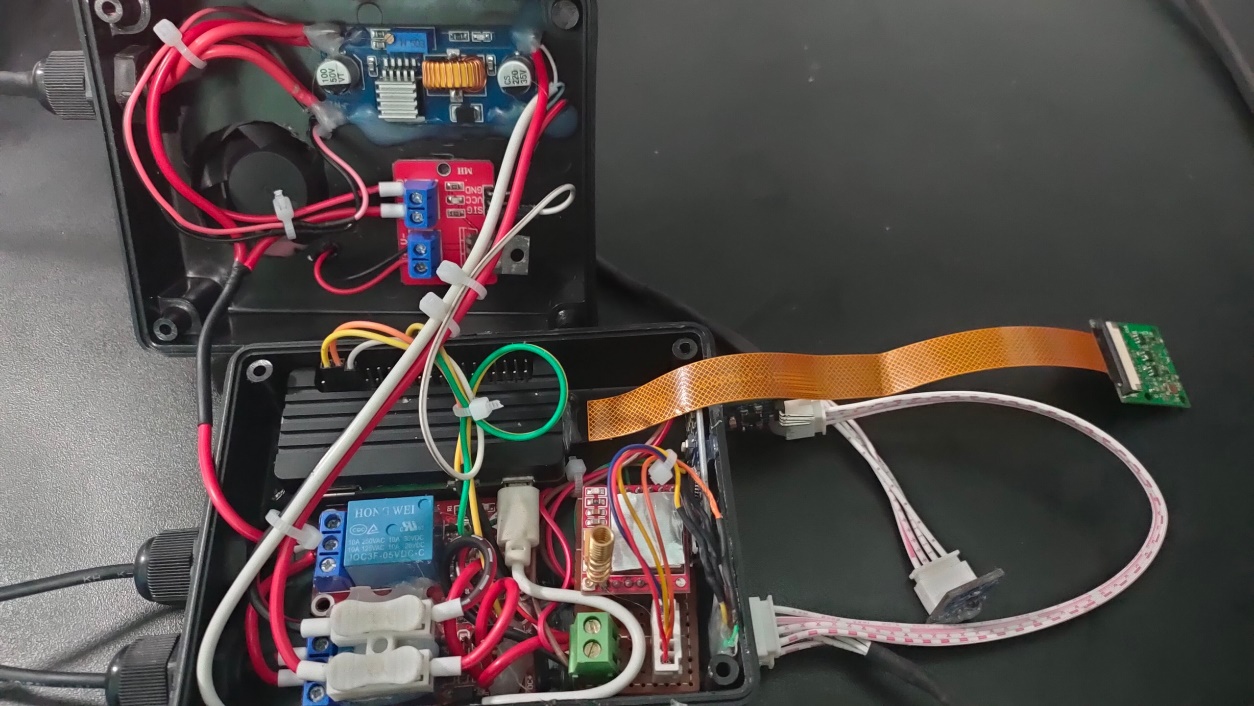
|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên thiết bị** | **Số lượng**  **(cái)** | **Hình ảnh** |
| 1 | Raspberry pi 2 zero | 1 | Raspberry Pi Zero 2 W (Wireless) - Raspberry Pi Việt Nam |
| 2 | RPI Camera V1.3 | 1 | Raspberry Pi Camera Board v1.3 (5MP, 1080p) — Pi Supply |
| 3 | MPU6050 | 1 |  |
| 4 | Module irf520 | 1 | Mạch Điều Khiển Công Suất Dùng Mosfet IRF520 - Linh Kiện 3M | Linh Kiện  Điện Tử 3M |
| 5 | Cảm biến cường độ ánh sáng GY-30 | 1 |  |
| 6 | Relay 5V | 2 | Module Relay 1 Kênh 5V-220VAC10A ( Cách Ly Quang ) |
| 7 | Module giảm áp DC XL4015 | 1 |  |
| 8 | SIM800L | 1 |  |
| 9 | Arduino Nano | 1 |  |
| 10 | Hộp | 1 |  |
| 11 | Ốc siết PG7 | 1 |  |

***4. Cách lắp ráp, lắp đặt sản phẩm***

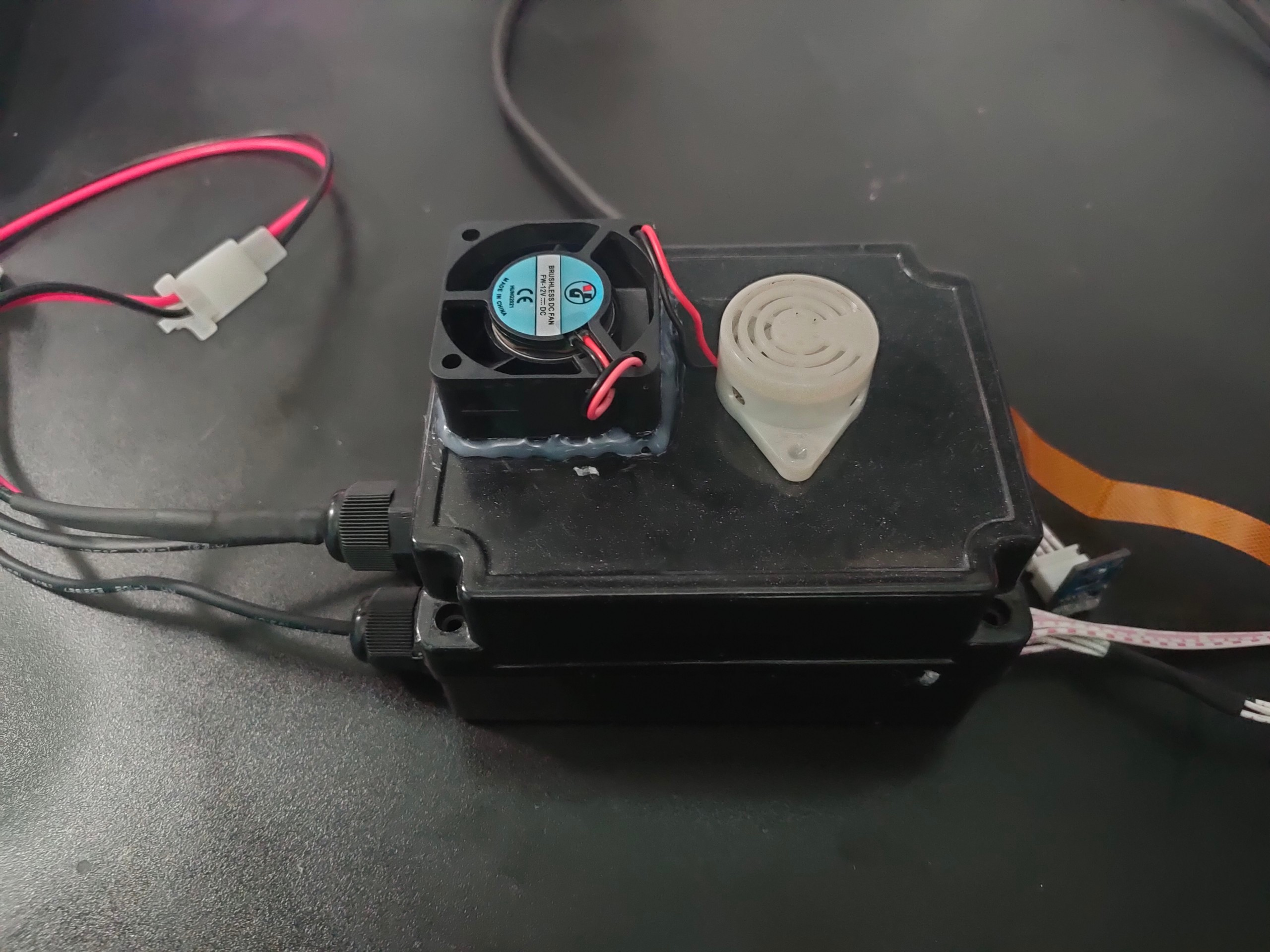
***4.1. Sơ đồ phần cứng***

|  |
| --- |
| **SƠ ĐỒ PHẦN CỨNG** |
|  |

*Hình 1 . Hình ảnh lắp ráp sản phẩm*

**

*Hình 2. Hình ảnh bên trong sản phẩm khi hoàn thiện*

**

*Hình 3. Hình ảnh sản phẩm khi hoàn thiện*

***4.2. Code lập trình Python ( Pi 2 zero W )***

import smbus

import time

from thread import Thread

y = 0

import cv2

from picamera.array import PiRGBArray

from picamera import PiCamera

import RPi.GPIO as GPIO

*#Nạp các thư viện cần thiết*

*#Khai báo Module cảm biến ánh sáng BH1750*

DEVICE = 0x23

POWER\_DOWN = 0x00

POWER\_ON = 0x01

RESET = 0x07

*#Khai báo Module góc nghiên*

PWR\_MGMT\_1 = 0x6B

SMPLRT\_DIV = 0x19

CONFIG = 0x1A

GYRO\_CONFIG = 0x1B

INT\_ENABLE = 0x38

ACCEL\_YOUT = 0x3D

bus = smbus.SMBus(1)

Device\_Address = 0x68

CONTINUOUS\_LOW\_RES\_MODE = 0x13

CONTINUOUS\_HIGH\_RES\_MODE\_1 = 0x10

CONTINUOUS\_HIGH\_RES\_MODE\_2 = 0x11

ONE\_TIME\_HIGH\_RES\_MODE\_1 = 0x20

ONE\_TIME\_HIGH\_RES\_MODE\_2 = 0x21

ONE\_TIME\_LOW\_RES\_MODE = 0x23

bus = smbus.SMBus(1) # Rev 2 Pi uses 1

*#Khai báo chân i2c của cảm biến ánh sáng và góc nghiên*

#khởi động camera

camera = PiCamera()

camera.resolution = (640, 480)

rawCapture = PiRGBArray(camera, size=(640, 480))

time.sleep(0.1)

vehiclexml = cv2.CascadeClassifier('vehicle.xml') *#Nạp module nhận diện phương tiện với OpenCV*

time.sleep(1)

def convertToNumber(data): #chuyển đổi dữ liệu số

result=(data[1] + (256 \* data[0])) / 1.2

return (result)

def readLight(addr=DEVICE*): #lấy dữ liệu i2c từ module cảm biến ánh sáng*

data = bus.read\_i2c\_block\_data(addr,ONE\_TIME\_HIGH\_RES\_MODE\_1)

return convertToNumber(data)

def MPU\_Init():

bus.write\_byte\_data(Device\_Address, SMPLRT\_DIV, 7)

bus.write\_byte\_data(Device\_Address, PWR\_MGMT\_1, 1)

bus.write\_byte\_data(Device\_Address, CONFIG, 0)

bus.write\_byte\_data(Device\_Address, GYRO\_CONFIG, 24)

bus.write\_byte\_data(Device\_Address, INT\_ENABLE, 1)

def read\_raw\_data(addr):

high = bus.read\_byte\_data(Device\_Address, addr)

low = bus.read\_byte\_data(Device\_Address, addr+1)

value = ((high << 8) | low)

if(value > 32768):

value = value - 65536

return value

MPU\_Init() #Lấy góc nghiên của cảm biến góc nghiên MPU6050

def detection(frame): *#Hàm phát hiện phương tiện*

global y

vehicles = vehiclexml.detectMultiScale(frame, 1.15, 4)

#lấy module nhận diện phương tiện

y = 0

for (x, y, w, h) in vehicles:

y = 1

cv2.rectangle(frame, (x, y), (x+w, y+h), color=(15, 254, 0), thickness=2)

# vẽ ra khung chủ nhật bao qua các phương tiện

cv2.putText(frame, 'Vehicle Detected!', (x+w, y+h+20), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (152, 255, 25), thickness=2)

return frame

GPIO.setmode(GPIO.BCM)

# Hàm bật pha

def pha():

GPIO.setup(5,GPIO.OUT)

GPIO.setup(6,GPIO.OUT)

time.sleep(5)

GPIO.output(5,GPIO.HIGH)

GPIO.output(6,GPIO.LOW)

#Hàm bật cos

def cos():

GPIO.setup(5,GPIO.OUT)

GPIO.setup(6,GPIO.OUT)

GPIO.output(5,GPIO.LOW)

GPIO.output(6,GPIO.HIGH)

*#Hàm xử lí chính MPU*

def main():

while True:

GPIO.setup(4, GPIO.OUT)

GPIO.setwarnings(False)

#Chuyển đổi dữ liệu sang góc

acc\_y = read\_raw\_data(ACCEL\_YOUT)

Ay = acc\_y/16384.0

in\_min = 1 ; in\_max = -1; out\_min = 0; out\_max = 90

value = (Ay - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min

value = int(value)

if (value > 80 ) or ( value < 10):

GPIO.output(4, GPIO.HIGH)

else:

GPIO.output(4, GPIO.LOW)

time.sleep(1)

def bh(): *#Hàm Cảm biến ánh sáng*

while True:

global y

print(y)

GPIO.setwarnings(False)

lightLevel=readLight()

lux =int(lightLevel)

*# Chuyển đổi dữ liệu vào biến lux để dễ dàng xử lí*

if (int(lux) < 40 ) and ( y == 1 ):

cos()

elif ( lux < 40 ) and ( y == 0) :

pha()

print("Light Level : ",lux)

time.sleep(1)

*# hàm camera*

def cam():

for frame in camera.capture\_continuous(rawCapture, format="bgr", use\_video\_port=True):

image = frame.array

vehicle\_frame = detection(image)

cv2.imshow('Vehicle Detection', vehicle\_frame)

rawCapture.truncate(0)

cv2.waitKey(2)

*# xử lí đa luồng*

thread1 = Thread(target = cam)

thread2 = Thread(target = bh)

thread2.start()

thread1.start()

***4.3. Code lập trình C++ ( Arduino Nano )***

#include <SoftwareSerial.h>

SoftwareSerial mySerial(3, 2);

int x;

void setup()

{

Serial.begin(9600);

mySerial.begin(9600);

Serial.println("Initializing...");

delay(1000);

mySerial.println("AT");

updateSerial();

mySerial.println("AT+CMGF=1"); // Configuring TEXT mode

updateSerial();

updateSerial();

mySerial.write(26);

pinMode(4,INPUT);

}

void loop()

{

x = digitalRead(4);

if ( x == HIGH ) {

sms();

call();

}

}

void sms(){

mySerial.println("AT+CMGS=\"+840385774777\"");

updateSerial();

mySerial.print("Cảnh báo va chạm");

}

void call(){

mySerial.println("ATD+ +840385774777;");

updateSerial();

delay(20000);

mySerial.println("ATH");

}

void updateSerial()

{

delay(500);

while (Serial.available())

{

mySerial.write(Serial.read());

}

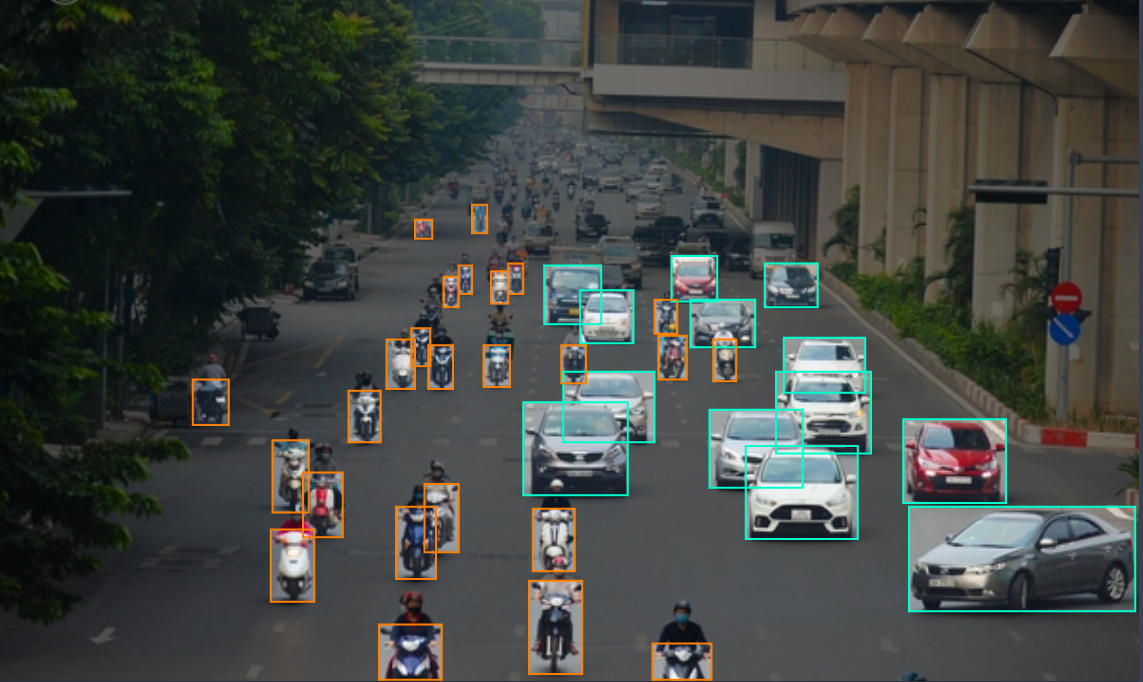
while (mySerial.available())

{

Serial.write(mySerial.read());t

}

}



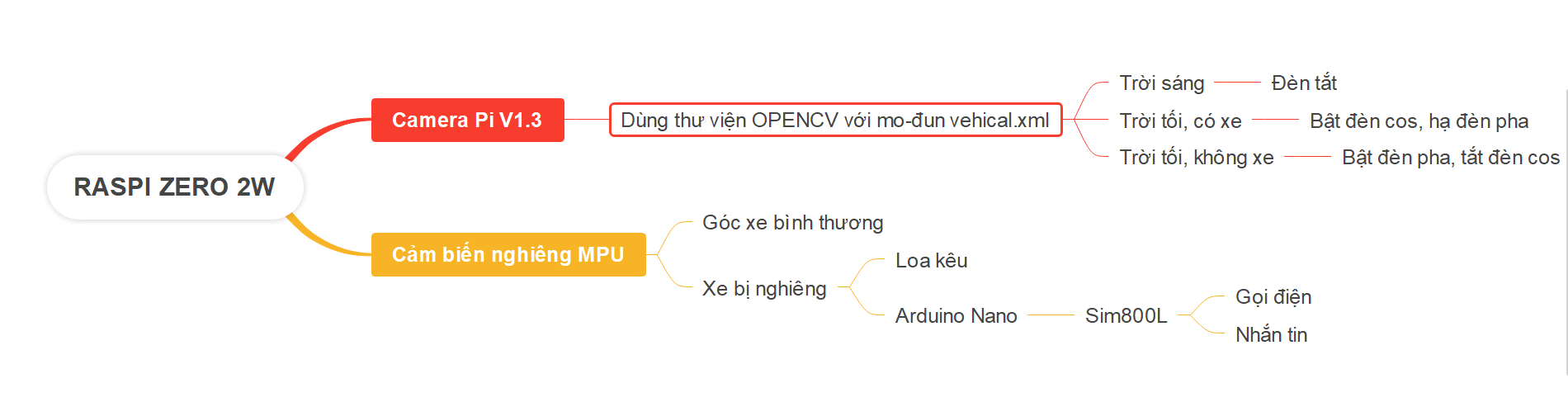
Hình 4. Ảnh demo của sản phẩm

***5. Nguyên tắc hoạt động, vận hành của các mô hình, sản phẩm dự thi***

**Nguyên tắc hoạt động:**

Camera sẽ thu thập hình ảnh, từ đó bo Pi 2 Zero W sẽ xử lí hình ảnh thông module ‘vehical.xml’ để nhận diện các phương tiện có trên đường . Module cường độ ánh sáng GY – 30 sẽ phát hiện cường độ ánh sáng mạnh hay yếu sau đó truyền dữ liệu đến bo Pi 2 Zero W. Khi cường độ ánh sáng yếu ( tức là ban đêm ) thì xe sẽ bật đèn pha. Nếu phát hiện được phương tiện trong khung hình thì sẽ hạ pha xuống cos để không gây chói với người điều khiển xe ngược chiều.

Module cảm biến gia tốc GY – 521 sẽ theo dõi 3 trục xyz của xe, nếu có sự thay đổi đột ngột (ví dụ): trong 1 – 5s nếu trục x thay đổi từ -> thì hệ thống đã được lập trình sẽ biết được là xe đã bị ngã và phát ra tín hiệu cảnh báo. Đồng thời sẽ gửi tin nhắn và điện thoại đến người thân qua module SIM được điều khiển bằng Arduino Nano.



**Các bước vận hành:**

**Bước 1:** Tiến hành cấp nguồn và bật nút khởi động hệ thống.

**Bước 2:** Đợi khoảng 10s để thiết bị ổn định.

**Bước 3:** Tiến hành điều khiển thiết bị thực hiện nhiệm vụ.

***6. Khả năng áp dụng của sản phẩm***

***a)Ưu điểm:***

Giá cả của sản phẩm dễ dàng đến tay người dùng vì nó rất cần thiết cho điều khiển xe.

Độ bền của sản phẩm cao, sử dụng có vi điều khiển tốt.

Do sử dụng chung dòng điện DC với ắc quy nên sản phẩm đảm bảo tính an toàn.

Công suất của sản phẩm thấp nên không gây ảnh hưởng đến ắc quy.

Tích hợp được nhiều module để mở rộng tính năng của sản phẩm.

Dễ dàng sử dụng, lắp đặt đối với mọi người tiêu dùng.

Hệ thống nhận diện đảm bảo tương đối phát hiện các phương tiện trên đường.

Sản phẩm có khả năng điều chỉnh cos – pha dựa trên các yếu tố khác nhau. Giúp cải thiện tầm nhìn và tăng tính an toàn của người lái.

**b) Nhược điểm:**

Số lượng khung hình trên giây của camera còn khá thấp.

***7. Hiệu quả đạt được của sản phẩm***

Sản phẩm đã được thực nghiệm và đạt hiệu quả cao khi phát hiện và nhận diện đúng các phương tiện trên đường. Từ đó cho ra khả năng bật tắt đèn cos – pha đúng lúc.

Sử dụng các vi mạch công nghệ cao, có thể chỉnh sửa, nâng cấp phần mềm làm việc dễ dàng mà không cần thao tác thay đổi phần cứng.

Tuổi thọ của các vi mạch cao nên hạn chế hư hại trong quá trình vận hành.

Ngoài ra sản phẩm còn đem lại mức độ tự động hóa cao, phù hợp cho các dây chuyền sản xuất lớn, mức độ tinh vi cao.

Giảm thiểu được các vụ tai nạn giao thông đường bộ xảy ra, bảo vệ được tính mạng của người lái xe.

Cảnh báo cho người lái xe và người thân biết được khi người lái xe gặp nguy hiểm.

Ít tiêu thụ điện năng của ắc quy, đảm bảo an toàn điện, tránh cháy nổ.

***QR của sản phẩm (video demo, thuyết minh sản phẩm)***

