**AMERICAN INTERNATIONALA close up of a sign

Description automatically generated**

**UNIVERSITY-BANGLADESH**

**Project Title: Vehicle Safety System**

Course Title: Advanced Operating System Section: [A]

Course Teacher: Md. Arifur Rahman Semester: Fall 2021-2022

Date of Submission: 12-12-2021

Group Member Details:

|  |  |  |
| --- | --- | --- |
| NO. | Name | ID |
| 1 | Fahad Mollah | 18-36702-1 |
| 2 | Fahim Hossain | 18-36954-1 |
| 3 | Ahnaf Sayed | 18-36920-1 |
| 4 | Md Sifat Rahman | 18-36705-1 |

# Table of Contents

|  |  |
| --- | --- |
| **Contents** | **Page No.** |
| Project Idea and Problem Statement | 03 |
| Component List | 03 |
| Project Implementation Description | 03 |
| Circuit Diagram | 06 |
| Implemented Code | 07 |
| Reference | 11 |

# Project Idea and Problem Statement

## Idea: Vehicle Safety System

We will create a vehicle safety automation system, when the driver enters and starts the car breath analyzer will check if he is drunk or not then using BP analysis meter and oximeter by checking driver's pulse and oxygen saturation. If the system finds any problem in data it will not start the vehicle and if the system doesn't find any problem in data from the breath analyzer, BP sensor and pulse oximeter it will start the vehicle. There will be a SOS message system if any accident occurs, ambulance and police immediately to place where accident occurred using GPS tracking. Also, there will an eye blink analyzer to detect if he is alive, if there is no eye blinking for 5 or 7 second it will stop the vehicle immediately.

# Component List

|  |  |  |
| --- | --- | --- |
| Component | Amount | Price |
| Arduino Mega | 1 | 1150 |
| L298N Motor Driver | 1 | 150 |
| Gear Motor | 1 | 200 |
| Buck Converter | 1 | 60 |
| Battery | 1 | 1200 |
| Wires | 1 | 60 |
| MQ 3 Alcohol sensor | 1 | 100 |
| Pulse Sensor | 1 | 300 |
| GPS Module | 1 | 850 |
| GSM Module | 1 | 800 |
| Display | 1 | 300 |

# Project Implementation Description

At first the alcohol sensor will be used for taking persons breath and pulse sensor will collect the bpm and then if any abnormal bpm or alcohol analog value is detected the car will not start. SMS Will be sent. For the device we have use 11.1-volt battery and buck converter for convert 11 volts to 5 volts. As all of the sensor requires 5v. For the eye blinking we have used the open CV library which allows multiple libraries to detect several faces and from that faces we detect eyes. Then the trained data can detect close eyes and open eyes. So, counting time we can detect the eyes are close or open. Time set for the whole detected and that’s detects the blinking.

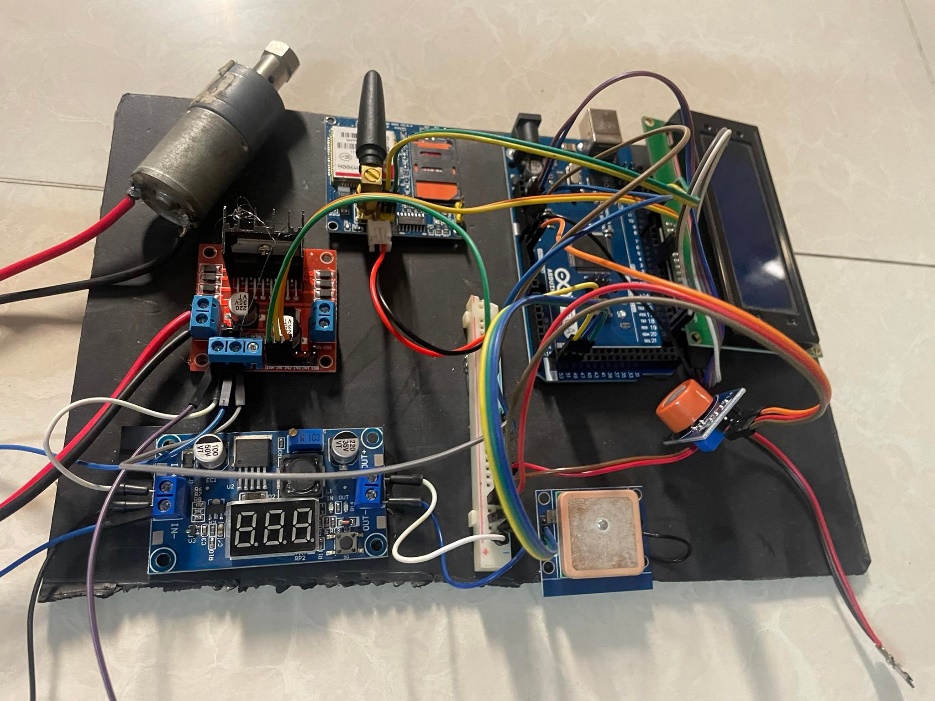


Figure: Device Implementation

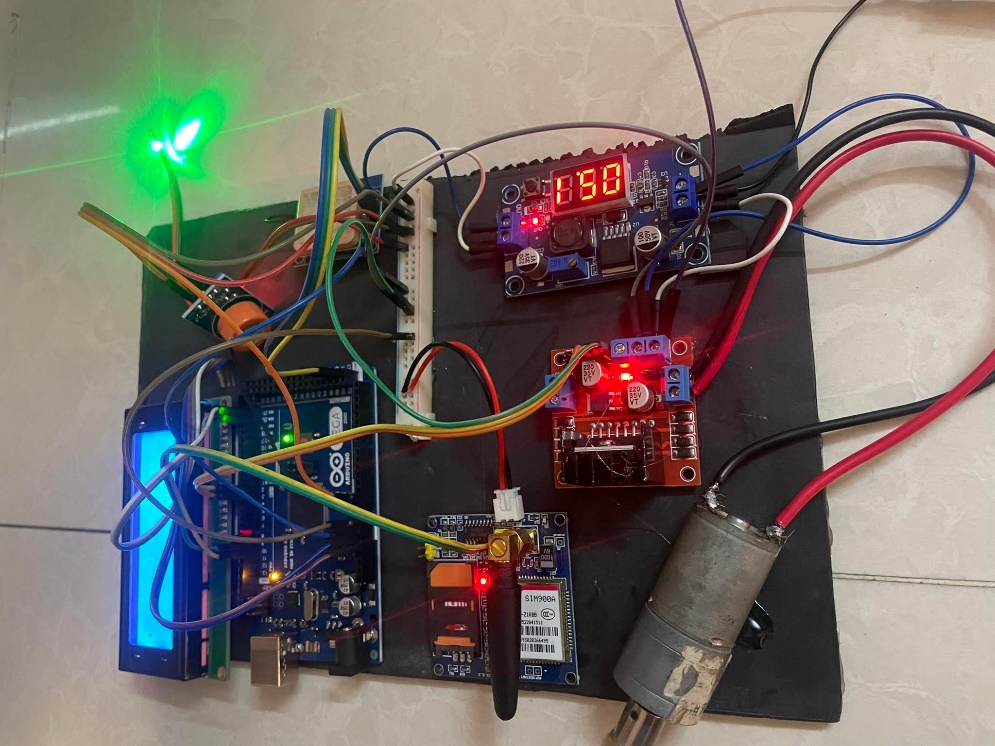


Figure: Device Implementation

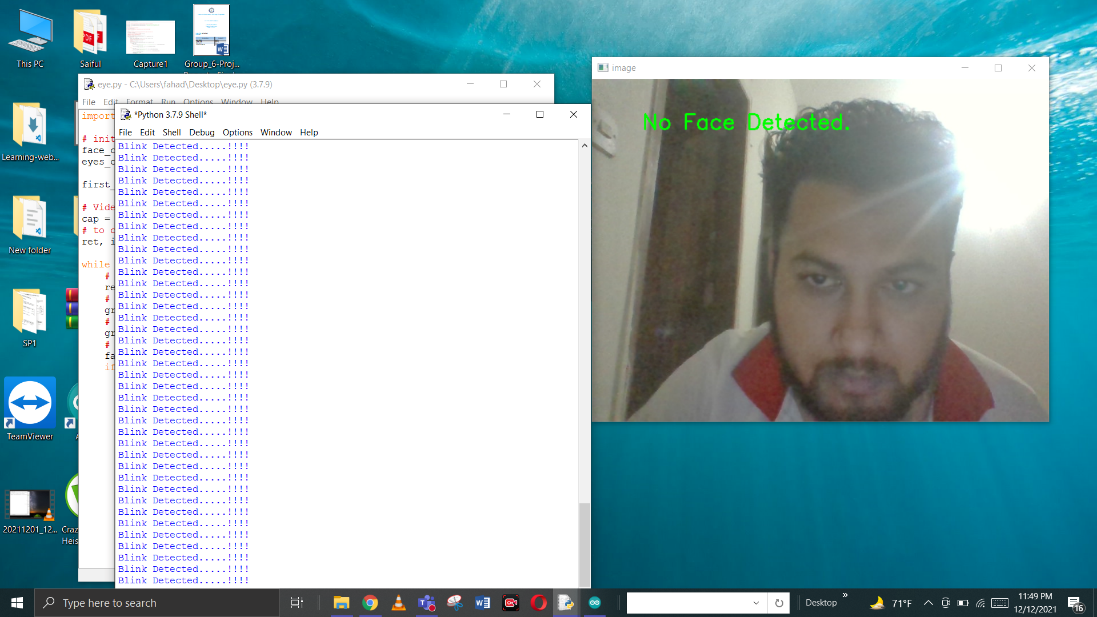


Figure: open Cv face detection

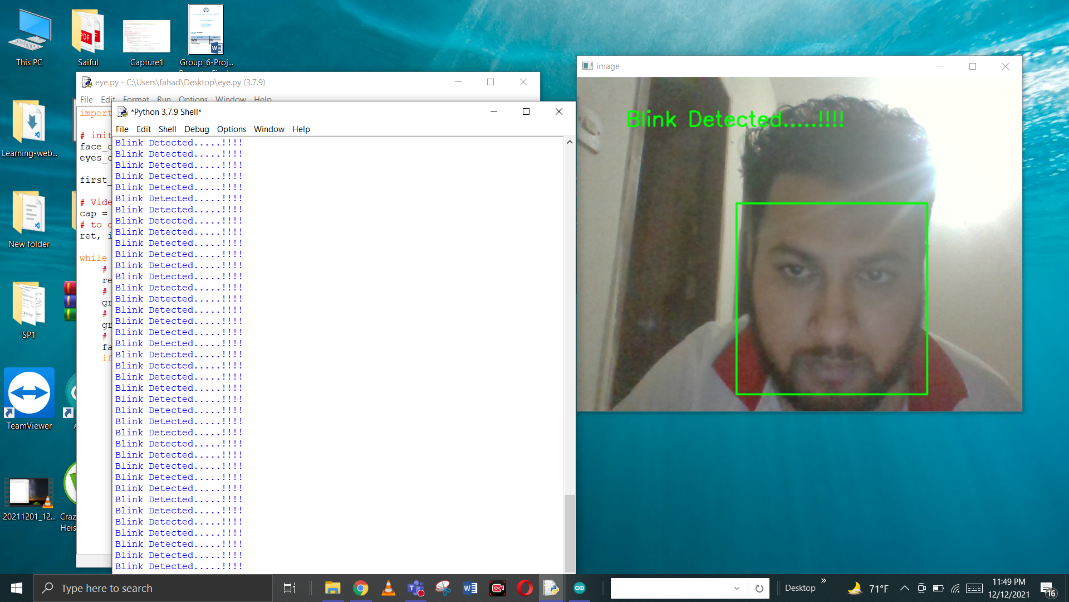


Figure: open Cv face detection

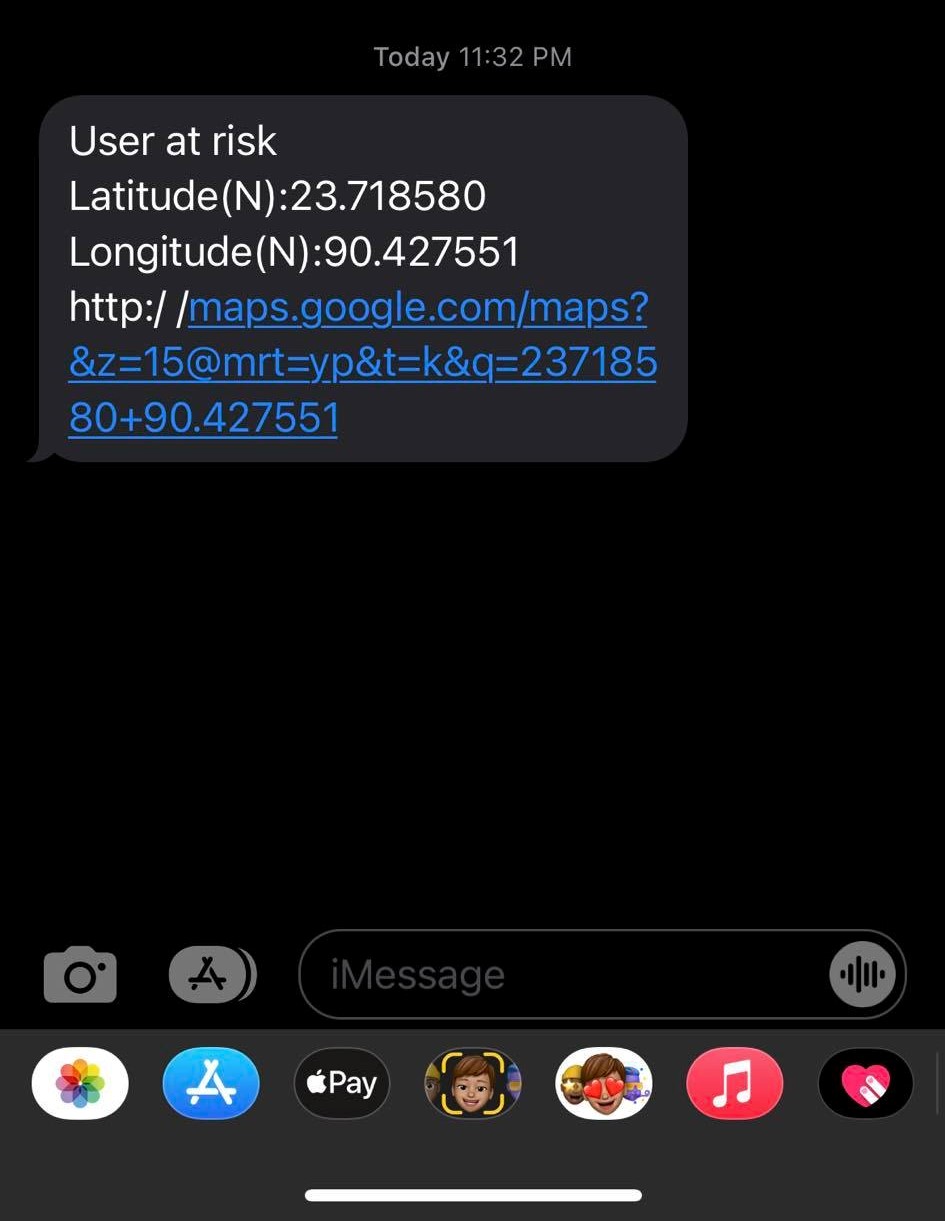


Figure: SMS from the GSM Module

# Diagram

## **Circuit Diagram**

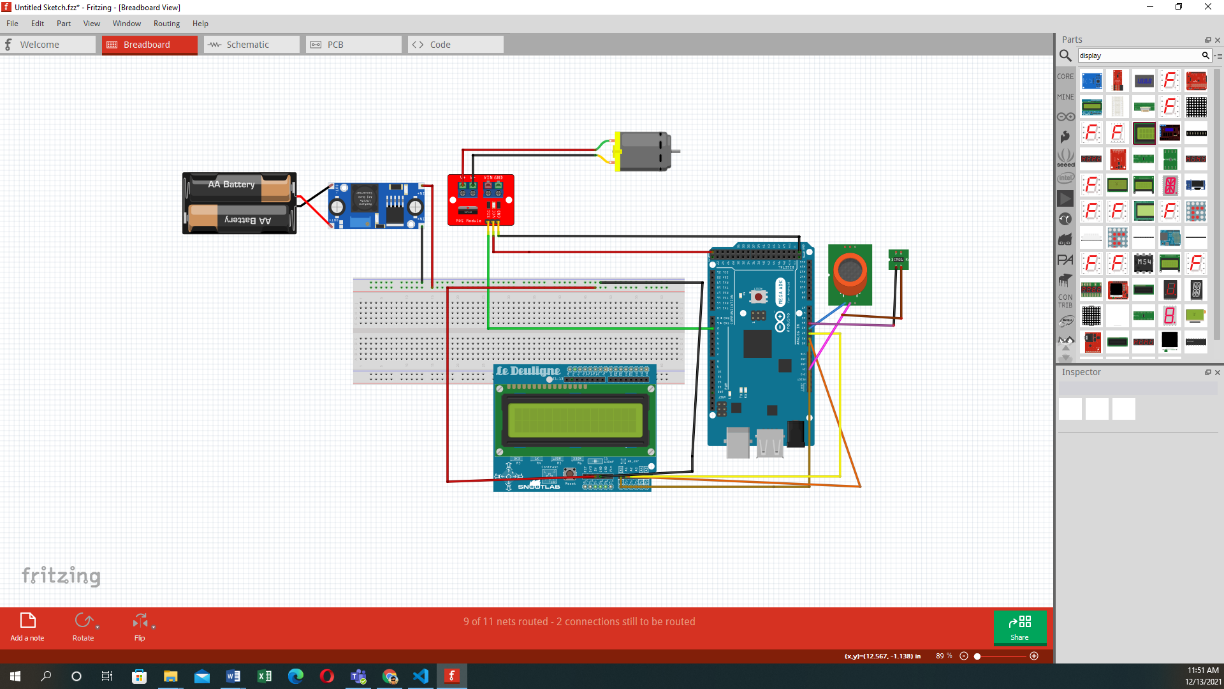


Figure: Circuit Diagram

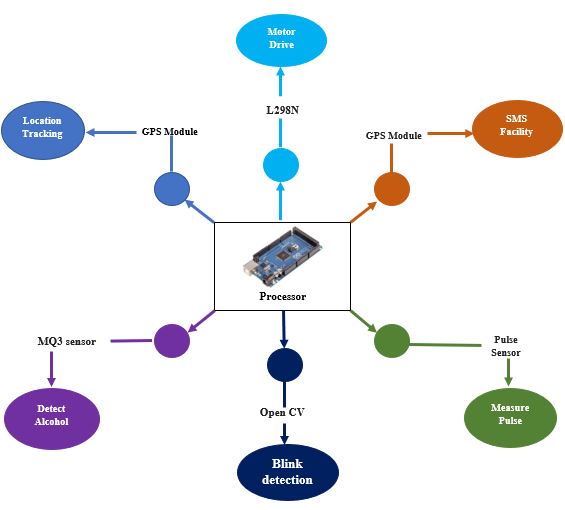


Figure: Diagram of the project

# Result Analysis

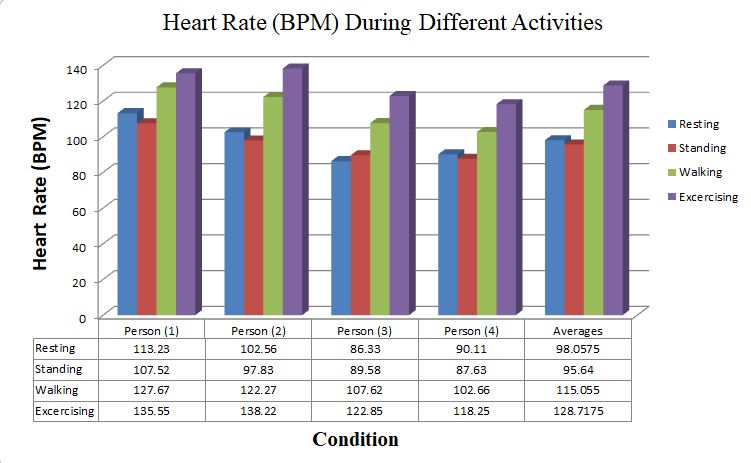


Figure: Heart BPM in different Activities to measure the abnormality

# Implemented Code

#define USE\_ARDUINO\_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.  
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library.    
#include <LiquidCrystal\_I2C.h>  
  
#include <SoftwareSerial.h> //library for the gsm module  
SoftwareSerial mySerial(9, 10);  
char msg;  
char call;  
  
LiquidCrystal\_I2C lcd(0x27,20,4);  //adress of the display  
  
//  Variables  
const int PulseWire = 0;       // PulseSensor PURPLE WIRE connected to ANALOG PIN 0         // The on-board Arduino LED, close to PIN 13.  
int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore.  
#define analog\_pin 1  
float sensor\_value;  //variable to store sensor value  
  
#define notDrunk 500   // Define max value that we consider not Drunk  
#define Drunk 440   // Define min value that we consider drunk  
  
int motor1pin1 = 2;  // motor driver pin1  
int motor1pin2 = 3;  //Motor driver pin 2  
  
  
                                 
PulseSensorPlayground pulseSensor;    
  
  
void setup() {    
  
  Serial.begin(115200);      
    mySerial.begin(9600);// For Serial Monitor  
  Serial.println("MQ3 warming up!");  
  delay(6000);  
  pinMode(motor1pin1, OUTPUT);  
  pinMode(motor1pin2, OUTPUT);  
  lcd.init();                      // initialize the lcd  
  // Print a message to the LCD.  
  lcd.backlight();  
  // Configure the PulseSensor object, by assigning our variables to it.  
  pulseSensor.analogInput(PulseWire);          //auto-magically blink Arduino's LED with heartbeat.  
  pulseSensor.setThreshold(Threshold);    
  
  // Double-check the "pulseSensor" object was created and "began" seeing a signal.  
   if (pulseSensor.begin()) {  
    Serial.println("We created a pulseSensor Object !");  //This prints one time at Arduino power-up,  or on Arduino reset.    
  }  
}  
  
  
  
void loop() {  
  
 int myBPM = pulseSensor.getBeatsPerMinute();  // Calls function on our pulseSensor object that returns BPM as an "int".  
                                               // "myBPM" hold this BPM value now.  
  
if (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if "a beat happened".  
 Serial.println("♥  A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".  
 Serial.print("BPM: ");                        // Print phrase "BPM: "  
 Serial.println(myBPM);    
  lcd.setCursor(3,0);  
  lcd.print("BPM:");  
  lcd.setCursor(2,1);  
  lcd.print(myBPM);// Print the value inside of myBPM.  
}  
  
  delay(20);      
   
sensor\_value = analogRead(analog\_pin);  
  
  Serial.print("Sensor Value: ");  
  Serial.print(sensor\_value);  
   
  if (sensor\_value < notDrunk) {  
    Serial.println("  |  Status: Not Drunk");  
    lcd.setCursor(0,2);  
  lcd.print("Status: Not Drunk");  
     digitalWrite(motor1pin1, HIGH);  
  digitalWrite(motor1pin2, LOW);  
  }  else {  
    Serial.println("  |  Status: DRUNK");  
     lcd.setCursor(0,2);  
  lcd.print("Status: DRUNK");  
   
     digitalWrite(motor1pin1, LOW);  
  digitalWrite(motor1pin2, LOW);  
  }  
   
  delay(2000);    
   
   
   
   
   
   
   
  // considered best practice in a simple sketch.  
  
}  
void SendMessage()  
{  
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode  
  delay(1000);  // Delay of 1000 milli seconds or 1 second  
  mySerial.println("AT+CMGS=\"+8801767670530\"\r"); // Replace x with mobile number  
  delay(1000);  
  mySerial.println("User at risk Latitude(N):23.718580 Longitude(N):90.427551

http://maps.google.com/maps?&z=15@mrt=yp&t=k&q=23718580+90.427551 ");// The SMS text you want to send  
  delay(100);  
   mySerial.println((char)26);// ASCII code of CTRL+Z  
  delay(1000);  
}

**Eye blinking code**

import cv2  
  
# initialize face and eye cascade xml of opencv library to detect face and eyes  
face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade\_frontalface\_default.xml")  
eyes\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade\_eye\_tree\_eyeglasses.xml")  
  
first\_read = True  
  
# Video Capturing by opening web-cam  
cap = cv2.VideoCapture(0)  
# to check for first instance of capturing it will return True and image  
ret, image = cap.read()  
  
while ret:  
    # this will keep the web-cam running and capturing the image for every loop  
    ret, image = cap.read()  
    # Convert the recorded image to grayscale  
    gray\_scale = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
    # Applying filters to remove impurities  
    gray\_scale = cv2.bilateralFilter(gray\_scale, 5, 1, 1)  
    # to detect face and eye  
    faces = face\_cascade.detectMultiScale(gray\_scale, 1.3, 5, minSize=(200, 200))  
    if len(faces) > 0:  
        for (x, y, w, h) in faces:  
            image = cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)  
            # eye\_face var will be i/p to eye classifier  
            eye\_face = gray\_scale[y:y + h, x:x + w]  
            # image  
            eye\_face\_clr = image[y:y + h, x:x + w]  
            # get the eyes  
            eyes = eyes\_cascade.detectMultiScale(eye\_face, 1.3, 5, minSize=(50, 50))  
            if len(eyes) >= 2:  
                if first\_read:  
                    cv2.putText(image, "Eye's detected, press s to check blink", (70, 70), cv2.FONT\_HERSHEY\_SIMPLEX,  
                                1, (0, 255, 0), 2)  
                else:  
                    cv2.putText(image, "Eye's Open", (70, 70), cv2.FONT\_HERSHEY\_SIMPLEX,  
                                1, (255, 255, 255), 2)  
            else:  
                if first\_read:  
                    cv2.putText(image, "No Eye's detected", (70, 70), cv2.FONT\_HERSHEY\_SIMPLEX,  
                                1, (255, 255, 255), 2)  
                else:  
                    cv2.putText(image, "Blink Detected.....!!!!", (70, 70), cv2.FONT\_HERSHEY\_SIMPLEX,  
                                1, (0, 255, 0), 2)  
                    cv2.imshow('image',image)  
                    cv2.waitKey(1)  
                    print("Blink Detected.....!!!!")  
                     
    else:  
        cv2.putText(image, "No Face Detected.", (70, 70), cv2.FONT\_HERSHEY\_SIMPLEX,  
                    1, (0, 255, 0), 2)  
    cv2.imshow('image', image)  
    a = cv2.waitKey(1)  
    # press q to Quit and S to start  
    # ord(ch) returns the ascii of ch  
    if a == ord('q'):  
        break  
    elif a == ord('s'):  
        first\_read = False  
# release the web-cam  
cap.release()  
# close the window  
cv2.destroyAllWindows()

# Reference

* <https://ijireeice.com/wp-content/uploads/2019/05/IJIREEICE.2019.7418.pdf>

**Smart Bus System Using IOT**

* <https://www.nabto.com/esp8266-for-iot-complete-guide/>

**ESP8266 for IoT: A Complete Guide**

* <https://create.arduino.cc/projecthub/ruchir1674/how-to-interface-gps-module-neo-6m-with-arduino-8f90ad>

**How to Interface GPS Module (NEO-6m) with Arduino**

* <https://www.seeedstudio.com/blog/2019/11/06/arduino-gps-modules-which-one-to-use-guide-and-comparisons/>

**Arduino GPS Modules – Which one to use? Comparison and Arduino Tutorial**

* <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>

**What is RFID? How It Works? Interface RC522 RFID Module with Arduino**

* <https://components101.com/articles/buck-converter-basics-working-design-and-operation>

[**Buck Converter: Basics, Working, Design and Operation**](https://components101.com/articles/buck-converter-basics-working-design-and-operation)

* <https://www.saftbatteries.com/energizing-iot/types-batteries-iot-devices>

**Which types of batteries for your IoT devices?**

* <https://www.cablinginstall.com/standards/network-protocols/article/16465159/cabling-infrastructure-for-the-internet-of-things>

**Cabling infrastructure for the Internet of Things**

* <https://www.techshopbd.com/>

**TechShopBD**