INSTITUTE OF ENGINEERING AND MANAGEMENT

PROJECT REPORT

ON

BASIC ELECTRICAL ENGINEERING (ESC 101)

SECTION A

TOPIC: Anti Sleep Driving Goggles!

Name	Enrolment No.	Class Roll No.
ABATANSHA DASGUPTA	12022002002098	2
ADARSH KUMAR	12022002002012	4
ADRIJA CHAUDHARY	12022002002091	6
AGNIK SARKAR	12022002002061	7
AMAN AGARWAL	12022002002113	9
AMBARISH SENGUPTA	12022002002011	10
AMITABHA GANGULY	12022002002078	11
AMRIT RAJ THAKUR	12022002002062	12

CONTENT

ACKNOWLEDGEMENT

We would like to express our thanks of gratitude to our professors who gave us the golden opportunity to do this wonderful project of Basic Electrical Engineering on “Anti Sleep Driving Google”. Who also helped us in completing our project. We came to know about so many new things. We are really thankful to them. Secondly, we would also like to thanks our parents and friends who helped us a lot in finalising this project within the limited time frame.

INTRODUCTION

The best foundation for a successful project starts with a comprehensive construction estimate prepared by a group of professional estimators. For this project we came across one of the most common reasons for car accidents that is sleeping while driving. This is very dangerous as it may lead to serious injuries and even death in some vital circumstances. For the solution of this problem, we have designed a High-Tech google which will let the driver know whenever he is about to sleep. This will allow the driver not to sleep and escape from various disastrous results which may have occurred otherwise.

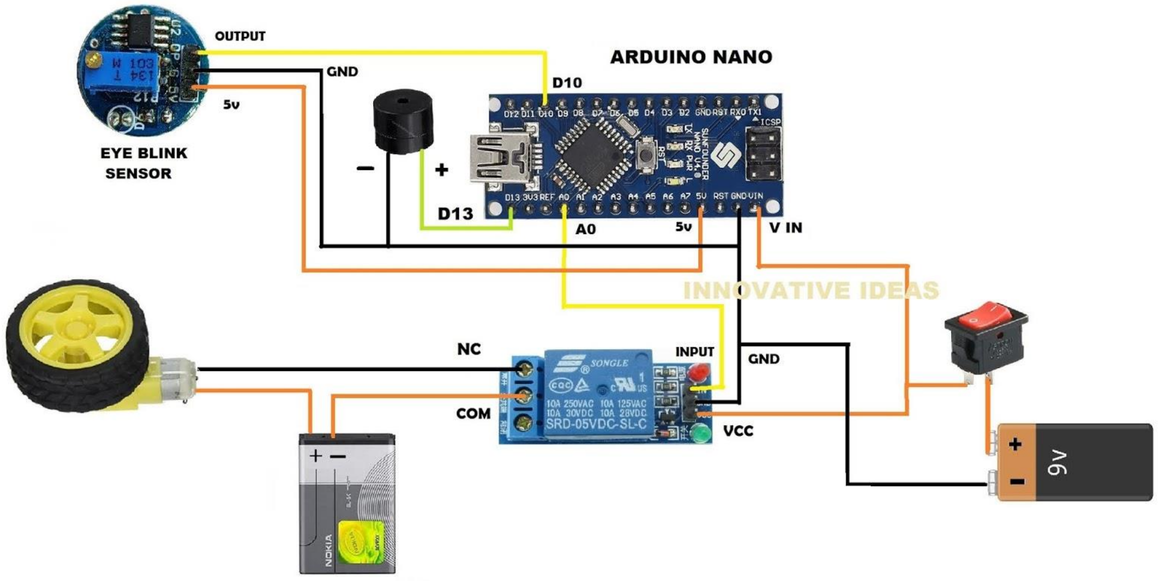
Abstract

Feeling sleepy while driving could cause hazardous traffic accident. However, when driving alone on highway or driving over a long period of time, drivers are inclined to feel bored and sleepy, or even fall asleep. Nowadays most of the products of driver anti-sleep detection sold in the market are simply earphone making intermittent noises, which is quite annoying and inefficient. As such, there is a high demand for cheap and efficient driver sleep detection. Therefore, we came up with an idea and successfully developed a sleepy detection and alarming system, which could effectively meet this demand.

BASIC THEORY BEHIND THIS PROJECT

In modern-times, owing to hectic schedules it becomes very difficult to remain active all the time. Imagine a situation where a person is driving home from work, dead tired after facing all the challenges of the day. His hands are on the wheel and foot on the pedal but suddenly he starts feeling drowsy, his eyes start shutting and his vision blurs and before he knows it, he’s asleep. Falling asleep on the wheel can lead to serious consequences, there may be accidents and people may even lose their lives. This situation is much more common then we notice and hence, it is very important to counter this problem. So to address this issue, we have come up with a Driver Anti-sleep Device. This system alerts the user if he/she falls asleep at the wheel thereby, avoiding accidents and saving lives. This system is useful especially for people who travel long distances and people who are driving late at night. The circuit is built around Schmitt trigger, timer IC, transistor, a relay and a logic gate. Around half an hour after the reset of timer IC, transistors rive the buzzer to sound an intermediate beep. If timer IC is not reset at that time, around one minute later the output of gate conducts. Due to this the clock stops counting further and relay energizes to deactivate the load. This state changes only reset switch is pressed. As a result of pressing the reset switch a next timer is set which will trigger the same events after half an hour.

CIRCUIT DIAGRAM



DETAILS OF HARDWARE COMPONENTS USED

SparkFun Arduino Pro Mini 328 - 5V/16MHz

Buzzer

Battery, 9 V

Brown Dog Gadgets Solar Cockroach Vibrating Disc Motor
General Purpose Transistor NPN

Resistor 4.75k ohm

APPLICATION

These glasses alert the driver whenever he is getting into sleep while driving the vehicle. Since sleeping on wheels is dangerous sometimes it may convert into fettle accidents can leads to death so to prevent such consequences of accident, we can use this gadget to alert the driver when he feels drowsiness.

CONCLUSION

As for the software part, we fulfilled our goal successfully. The detection algorithm could not only work effectively and accurately at daytime, but also at night. The Eye portion extraction is smooth and in real time with no delays on the computer. In addition, there is a bonus function in the software part – detection with glasses. For the Beagleboard, we achieved two major difficulties. First, we were not able to power up the board with any commercial chargers initially, including the ones for Iphone, for Samsung, or the USB charger on car. But later we added DXPOWER battery to power our board and used the power supply we designed to charge the battery to solve the problem. Second, we experienced a few difficulties while installing the OpenCV library on Beagleboard, but were able to solve it by changing flags in make files to the one corresponding to ARM board architecture. The power supply unit basically completes all its design requirements. By adding the extra USB battery stage, the problem of powering the entire microcontroller and alarming system has been solved. Moreover, the alarming system works as we supposed. The voltage ripple of the power supply unit can be mitigated by applying more resilient capacitor components. It is apparent that the overall project success is not derived from one team member’s mind but the keen coloration within our group. Each part is indispensable and every team member made the great dedication on the completion of this design project.

The pace is intense, the learning, immense. Ethnical consideration: To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment. By using our Driver Sleep Detection and Alarming System, customers would be warned when his/her physical condition is not good enough for driving and thus prevents dangerous behaviours from happening. It is consistent with the safety and welfare of the public. To improve the understanding of technology; its appropriate application, and potential consequences; By using open CV and related libraries, we try to develop and improve algorithm for eye closeness detecting. We then apply this technology to our application in order to help drivers achieve a better and safer driving condition. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others; We consult Professor and TAs for review advices and improve, seek online resources to help correcting errors, and properly cite the contributions of other people. To avoid injuring others, their property, reputation, or employment by false or malicious action; We design our product using qualified components and follow proper safety rules, avoid wrong actions happening to other people.

REFERENCES

[1] “Innovative Ideas”

https://www.youtube.com/watch?v=KvXQ-GPyfc4

https://innovativeideasyoutube.blogspot.com/2021/06/anti-sleep-alarm-for-drivers.html

[2] Shubham Shinganapure’s post at hackster.io

https://www.hackster.io/shubhamsuresh

[3] Shubham Suresh’s article

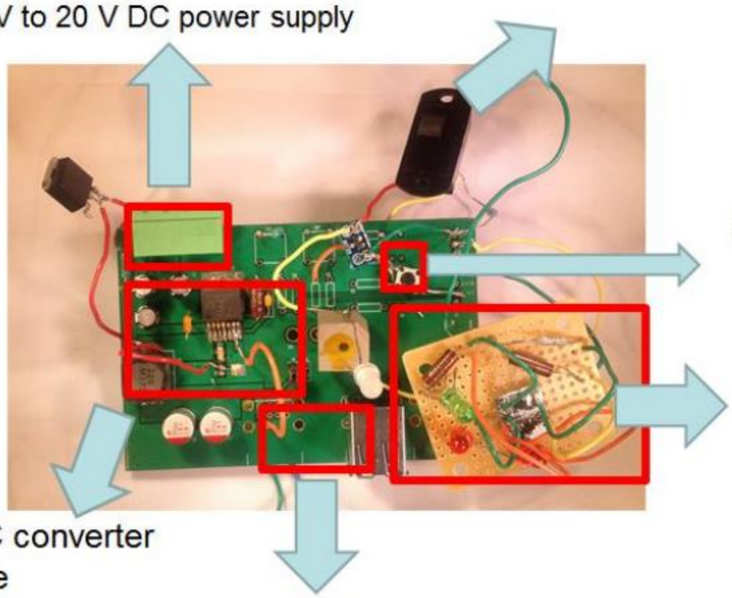
https://www.instructables.com/Anti-Sleep-Glasses/

[4] Google Patents

https://patents.google.com/patent/EP2729922A1/en

APPENDIX

- Overall system PCB view



Warning System slide switch

8V to 20 V DC power supply

Push button

Warning module LED+NMOS

DC-DC converter module

USB power ports (6 in total)

CODE

```
#define Relay 13
#define buzzer A0
static const int sensorPin = 10;                  // sensor input pin
int SensorStatePrevious = LOW;                    // previousstate of the sensor

unsigned long minSensorDuration = 3000; // Time we wait before  the sensor active as long
unsigned long minSensorDuration2 = 6000;
unsigned long SensorLongMillis;              // Time in ms when the sensor was active
bool SensorStateLongTime = false;                // True if it is a long active

const int intervalSensor = 50;                   // Time between two readings sensor state
unsigned long previousSensorMillis;              // Timestamp of the latest reading

unsigned long SensorOutDuration;                 // Time the sensor is active in ms

//// GENERAL ////

unsigned long currentMillis;         // Variabele to store the number of milleseconds since the Arduino
has started

void setup() {
  Serial.begin(9600);               // Initialise the serial monitor

  pinMode(sensorPin, INPUT);         // set sensorPin as input
  Serial.println("Press button");
  pinMode(Relay,OUTPUT);
  pinMode(buzzer,OUTPUT);
}
```

```
// Function for reading the sensor state
void readSensorState() {

  // If the difference in time between the previous reading is larger than intervalsensor
  if(currentMillis - previousSensorMillis > intervalSensor) {

    // Read the digital value of the sensor (LOW/HIGH)
    int SensorState = digitalRead(sensorPin);

    // If the button has been active AND
    // If the sensor wasn't activated before AND
    // IF there was not already a measurement running to determine how long the sensor has been
activated
    if (SensorState == LOW && SensorStatePrevious == HIGH && !SensorStateLongTime) {
     SensorLongMillis = currentMillis;
       SensorStatePrevious = LOW;

      Serial.println("Button pressed");
    }

    // Calculate how long the sensor has been activated
   SensorOutDuration = currentMillis - SensorLongMillis;

    // If the button is active AND
    // If there is no measurement running to determine how long the sensor is active AND
    // If the time the sensor has been activated is larger or equal to the time needed for a long active
    if (SensorState == LOW && !SensorStateLongTime && SensorOutDuration >= minSensorDuration)
{
      SensorStateLongTime = true;
      digitalWrite(Relay,HIGH);
```

```
        Serial.println("Button long pressed");

}

   if (SensorState == LOW && SensorStateLongTime && SensorOutDuration >= minSensorDuration2)
{

    SensorStateLongTime = true;

     digitalWrite(buzzer,HIGH);

     delay(1000);

     Serial.println("Button long pressed");

   }


   // If the sensor is released AND

   // If the sensor was activated before

   if (SensorState == HIGH && SensorStatePrevious == LOW) {

     SensorStatePrevious = HIGH;

     SensorStateLongTime = false;

     digitalWrite(Relay,LOW);

     digitalWrite(buzzer,LOW);

     Serial.println("Button released");



   }


   // store the current timestamp in previousSensorMillis

  previousSensorMillis = currentMillis;


 }


}
```

```
  void loop() {
currentMillis = millis();    // store the current time
  readSensorState();           // read the sensor state
}
```