A PROJECT REPORT

FOR

MICROCONTROLLER AND APPLICATIONS (ECE 3003)

SMART DRIVING SYSTEM

BY

SRIVATHSAN V K K	**18BEC0341**
ABHINAV KUMAR	**18BEC2017**
PARVESH KUMAR	**18BEC0722**

NAME OF THE FACULTY: PADMINI T N

SLOT- L13+L14

FALL 2020-21

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ABSTRACT:

Drinking driving is responsible for a high proportion of traffic accidents. It has been known that alcohol use impairs driving skills and increases accident risk. It has been found that while driving under the influence of alcohol, the risk of having an accident causing injury or death increases exponentially. In Europe, drinking driving is thought to be responsible for 10,000 deaths each year. Alcohol-impaired driving accidents contribute to approximately 31% of all traffic fatalities in the USA. In China, Li et al. revealed that about 34.1% of road accidents were alcohol related.

Drunk driving has a high probability to lead to serious accidents. Even with a small amount of alcohol assumption, drivers are twice likely to be involved in traffic accidents than sober drivers. Therefore, many countries have been working on solutions to drunk driving for a long period of time, including publicity and education and tough drunk-driving laws.

So we have developed a SMART DRIVING SYSTEM as a project where the system finds whether the driver has consumed alcohol or not and stops the car when necessary.

INTRODUCTION:

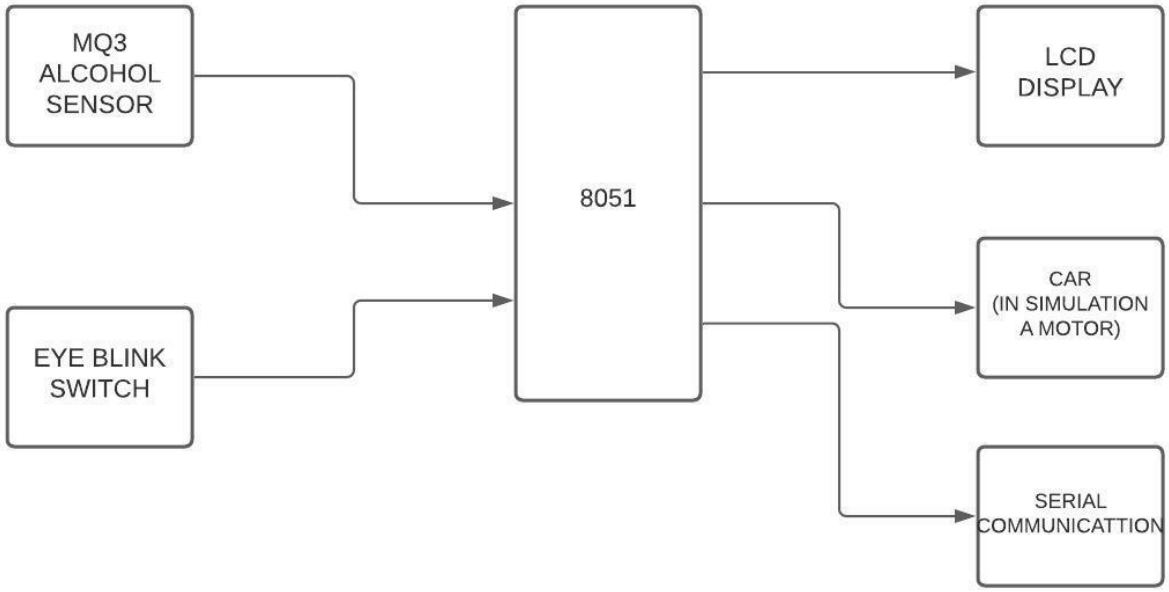
We have developed a smart driving system that will detect the alcohol content using MQ135 alcohol gas sensor and give an alert when the alcohol content is detected in the system and also stops the system when the given limit of alcohol content is increased.

But, during the COVID crisis people use sanitizers regularly from getting infected and sanitizers contain alcohol content, so when a passenger or driver uses sanitizer in the car, then our project will stop the car and alerts which is inappropriate taking considerations of today's situation.

So, we have included an eye-blink sensor (a switch in simulation), which along with the MQ3 gas sensor helps in finding out the drunk and normal person, We have attached this eye-blink because a drunk person can feel drowsy after the consumption, So it will help us to easily identify.

When both the sensors are high, then we will print the message 'DRUNK' in the LCD and send the alert to the respected number. Otherwise we will print the message 'NOT DRUNK' in the LCD.

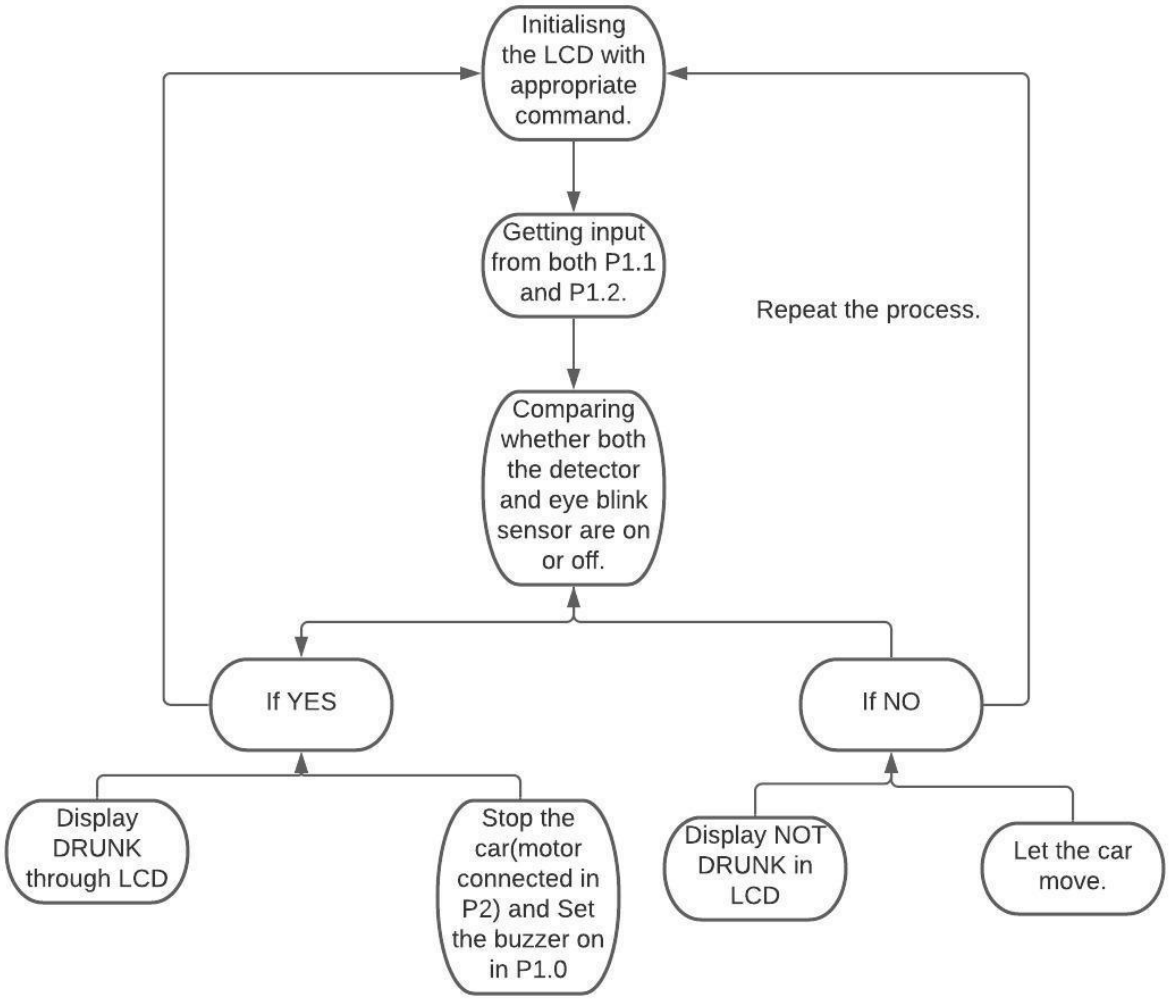
BLOCK DIAGRAM:



OVERVIEW OF COMPONENTS.

1. MQ 3 SENSOR-Alcohol sensor is a Sno2 with a lower conductivity of clean air. An alcohol sensor detects the attentiveness of alcohol gas in the air and an analog voltage is an output reading. The gas sensor senses the gases like ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulfide and smoke.
2. 8051 MICROCONTROLLER
3. EYE-BLINK SENSOR(we have used a switch for simulation)
4. LCD DISPLAY.
5. L293 DRIVER
6. ADC-0804
7. BUZZER
8. OPEN CV PYTHON
9. KEIL MICRO VISION
10. PROTEUS

FLOWCHART OF THE CODE WRITTEN:



8051 CODE:

```
ORG 0000H
MOV P1,#00H
MOV P2,#05H; MAKING THE MOTOR TO RUN FORWARD
SETB P1.2; MQ3 SENSOR INPUT.SO INITALISING THE PARTICULAR PIN
ACALL DELAY;CALLING DELAY BECAUSE MICROCONTROLLER IS VERY FAST COMPARED TO LCD
MOV A,#38H;INTIALISING LCD TO 2 LINES AND 5x7 MATRIX, SENDING THE COMMAND TO A REGISTER AND THEN TO LCD VIA PORT P0
ACALL LCD; SUBROUTINE FOR SENDING THE COMMAND REGISTERING THE LCD
MOV A,#0EH;MAKING THE LCD ON AND CURSOR TO BLINK
ACALL LCD;SIMILAR SUBROUTINE TO SEND THE COMMAND TO LCD VIA PORT P0
MOV A,#01H;CLEARING THE LCD DISPLAY SCREEN
ACALL LCD;SUBROUTINE
MOV A,#82H;FORCING CURSOR TO BLINK IN THE FIRST LINE
ACALL LCD;SIMILAR SUBROUTINE
MOV A,#00H
AGAIN:MOV C,P1.1;Sending the eye blink sensor input to Carry flag
MOV ACC.0,C;sending the carry flag data to the LSB of A register
MOV C,P1.2;Similarly sending the MQ3 Sensor input to Carry flag.
ANL C,ACC.0;Now using AND operation to check whether both the eyeblink sensor and MQ3 gas sensor input are 1 or not
MOV ACC.0,C;Now moving that to LSB of A register
CJNE A,#01H,WAIT;Checking whether LSB of A register is 1 which implies that both the sensor detected alcohol and drowsiness. Or jump to WAIT LABEL
ACALL DELAY;If yes comes down
ACALL TRANS;Subroutine for transmitting Message to concerned members via Serial communication
ACALL DELAY
MOV DPTR,#STRING;Storing DRUNK Message in ROM Location
BACK:SETB P2.0;Next four lines implies to stop the motor which is connected
SETB P2.1
SETB P2.2
SETB P2.3
SETB P1.0;Switching on the buzzer
MOV A,#00H
MOVC A, @A+DPTR; Moving each character of 'DRUNK' from DPTR to the A register
JZ EXIT; To check whether all the characters are displayed. If it then it will jump to EXIT label
ACALL SHOW;Displaying word by word of the 'DRUNK' message in the LCD via the SHOW subroutine
INC DPTR;Incrementing DPRT
```

```
SJMP BACK;Continuing the process
EXIT:MOV A,#01H; Clearing the LCD screen and then displaying otherwise the characters will overlap
ACALL LCD
ACALL DELAY
SJMP AGAIN
WAIT:CJNE A,#00H,AGAIN;If both the sensors didnt detect or either one only detected then the LSB of A register must contain 00h.
MOV DPTR, #STRING1;Storing NOT DRUNK Message in ROM Location
BACK1:SETB P2.0;Next four lines implies to power the motor or let the car run
CLR P2.1
SETB P2.2
CLR P2.3
CLR P1.0;Clearing the buzzer
MOV A,#00H
MOVC A, @A+DPTR;Similar process done like displaying DRUNK
JZ EXIT1
ACALL SHOW
INC DPTR
SJMP BACK1
EXIT1:MOV A,#01H
ACALL LCD
ACALL DELAY
ACALL DELAY
SJMP AGAIN


LCD:ACALL DELAY;Command registering the LCD
MOV P0,A;P0 is connected to the data pins of LCD ,data is transmitted through it from A
CLR P2.4;making the LCD in write mode
CLR P2.5;Making the LCD in command registering mode
SETB P2.6;Sending a high to low pulse in enable pin of LCD
CLR P2.6
RET

SHOW:ACALL DELAY;Displaying data in LCD
MOV P0,A
SETB P2.4
CLR P2.5
SETB P2.6;Making the LCD in data registering mode
CLR P2.6
RET
```

```
TRANS:ACALL DELAY;Transmitting message to the concerned member if alcohol is detected
MOV DPTR,#STRING2;String 2 contains message that is to be sent.
MOV TMOD,#20H;Setting timer 1 in mode 2 for serial communication
MOV SCON,#50H; Setting Serial communication register to mode 2 with REN
MOV TH1,#-3;Setting baud rate to 9600
SETB TR1
MOV R2,#47;Length of the string that is to be sent
AGAIN1:CLR A
MOVC A,@A+DPTR
MOV SBUF,A
HERE1:JNB TI,HERE1;Monitoring the Transmit Flag
CLR TI
INC DPTR
DJNZ R2,AGAIN1


DELAY:MOV R0,#0FFH;Delay subroutine
ONCE:MOV R1,#0FFH
TWICE:DJNZ R1,TWICE
DJNZ R0,ONCE
RET



STRING:DB 'DRUNK',00H;DPTR LOCATIONS OF ALL THE STRING
STRING1:DB 'NOT DRUNK',00H;00H implies that it has completed the whole particular string
otherwise it will print all the three strings
STRING2:DB 'ALCOHOL DETECTED, MESSAGE SENT TO +919442130430'.
```

OPEN CV DROWSINESS DETECTION CODE IN PYTHON:

```
import cv2
import numpy as np

# Load the Haar cascade files for face and eye
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

# Check if the face cascade file has been loaded correctly
if face_cascade.empty():
        raise IOError('Unable to load the face cascade classifier xml file')

# Check if the eye cascade file has been loaded correctly
if eye_cascade.empty():
        raise IOError('Unable to load the eye cascade classifier xml file')

# Initialize the video capture object
cap = cv2.VideoCapture(0)
#
# Define the scaling factor
ds_factor = 0.5

# Iterate until the user hits the 'Esc' key
while True:
    # Capture the current frame
    _, frame = cap.read()

    # Resize the frame
    frame = cv2.resize(frame, None, fx=ds_factor, fy=ds_factor, interpolation=cv2.INTER_AREA)

    # Convert to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Run the face detector on the grayscale image
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    # For each face that's detected, run the eye detector
    for (x,y,w,h) in faces:
        # Draw a rectangle around the face
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 3)

        # Extract the grayscale face ROI
        roi_gray = gray[y:y+h, x:x+w]

        # Extract the color face ROI
```

```
        roi_color = frame[y:y+h, x:x+w]

        # Run the eye detector on the grayscale ROI
        eyes = eye_cascade.detectMultiScale(roi_gray)

        # Draw circles around the eyes
        for (x_eye,y_eye,w_eye,h_eye) in eyes:
            center = (int(x_eye + 0.5*w_eye), int(y_eye + 0.5*h_eye))
            radius = int(0.3 * (w_eye + h_eye))
            color = (0, 255, 0)
            thickness = 3
            cv2.circle(roi_color, center, radius, color, thickness)

    # Display the output
    cv2.imshow('both', frame)

    # Check if the user hit the 'Esc' key
    c = cv2.waitKey(1)
    if c == 27:
        break

# Release the video capture object
cap.release()

# Close all the windows
cv2.destroyAllWindows()
```

MAM YOU HAVE TOLD US TO USE MATLAB FOR FINDING WHETHER THE DRIVER IS DRUNK OR NOT AND LINK IT IN THE KEIL SOFTWARE BUT WE WERE NOT ABLE TO DO MAM.

BUT WE HAVE DID IT IN PYTHON BUT WERE NOT ABLE TO LINK. SO WE HAVE CONTINUED WITH THE SWITCH ONLY FOR THE EYE BLINK PART.

CODE REFERENCES:

This entire 8051 code was written by **Srivathsan on his own not downloaded from internet**
Proteus simulation and report making was done by **ABHINAV AND PARVESH**

LCD Interfacing part and Serial Communication part of the project was referred from

The_8051_Microcontroller_and_Embedded_System textbook by MAZIDI.

YOUTUBE LECTURE VIDEO FOR LCD INTERFACING.

LINK OF THE VIDEO

https://www.youtube.com/watch?v=LtevacTk7Ww

Just the displaying part of the LCD was referred from the video.

The use of switch for eye blink sensor was inspired from another youtube video.

LINK OF THE VIDEO:

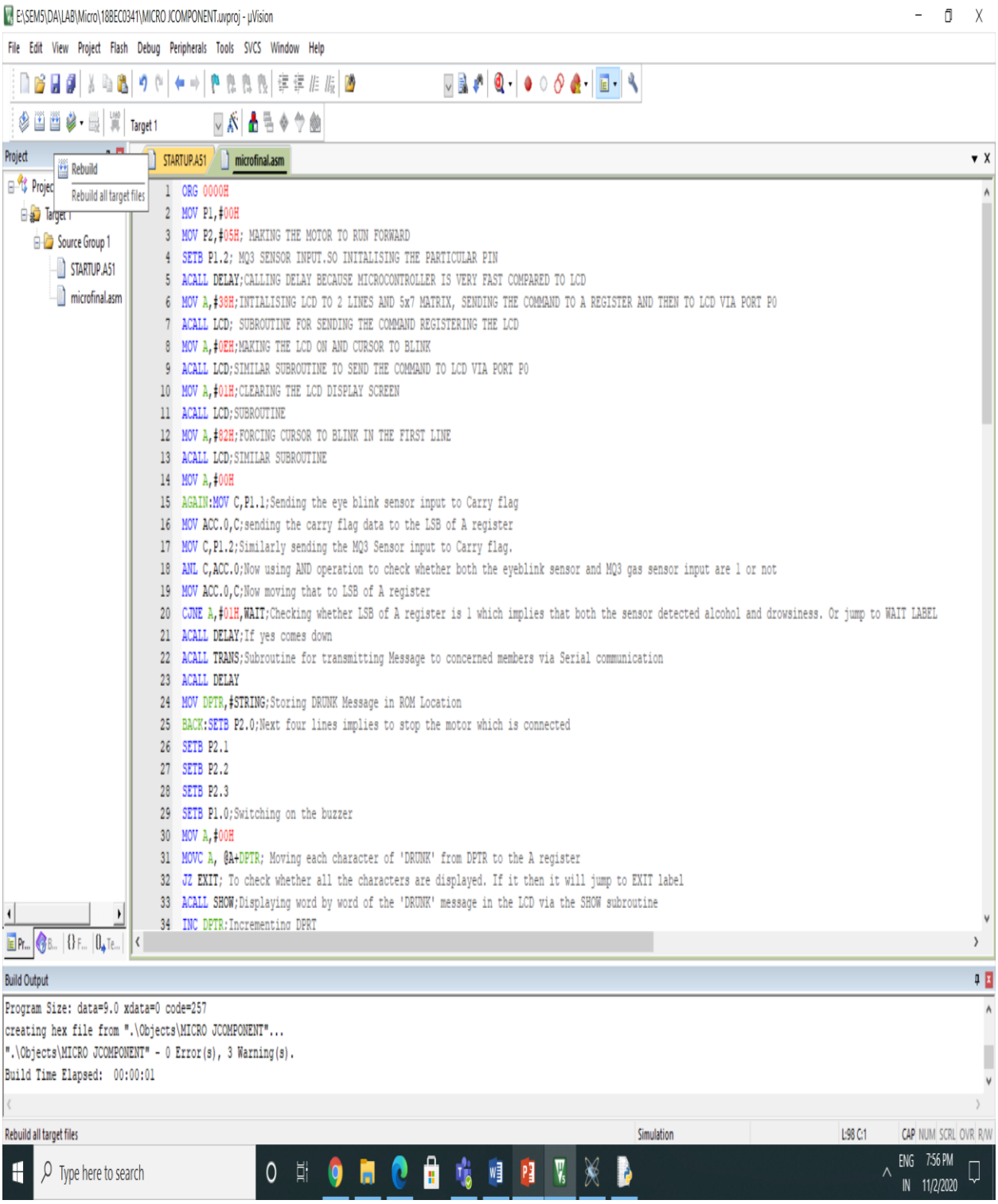
https://www.youtube.com/watch?v=aYhIWBodkfE

The code was not copied from the video as it was written in C language.

The OPEN CV Python code was referred from our theory class of Artificial Intelligence using Python (ECE-4031) Class this semester.

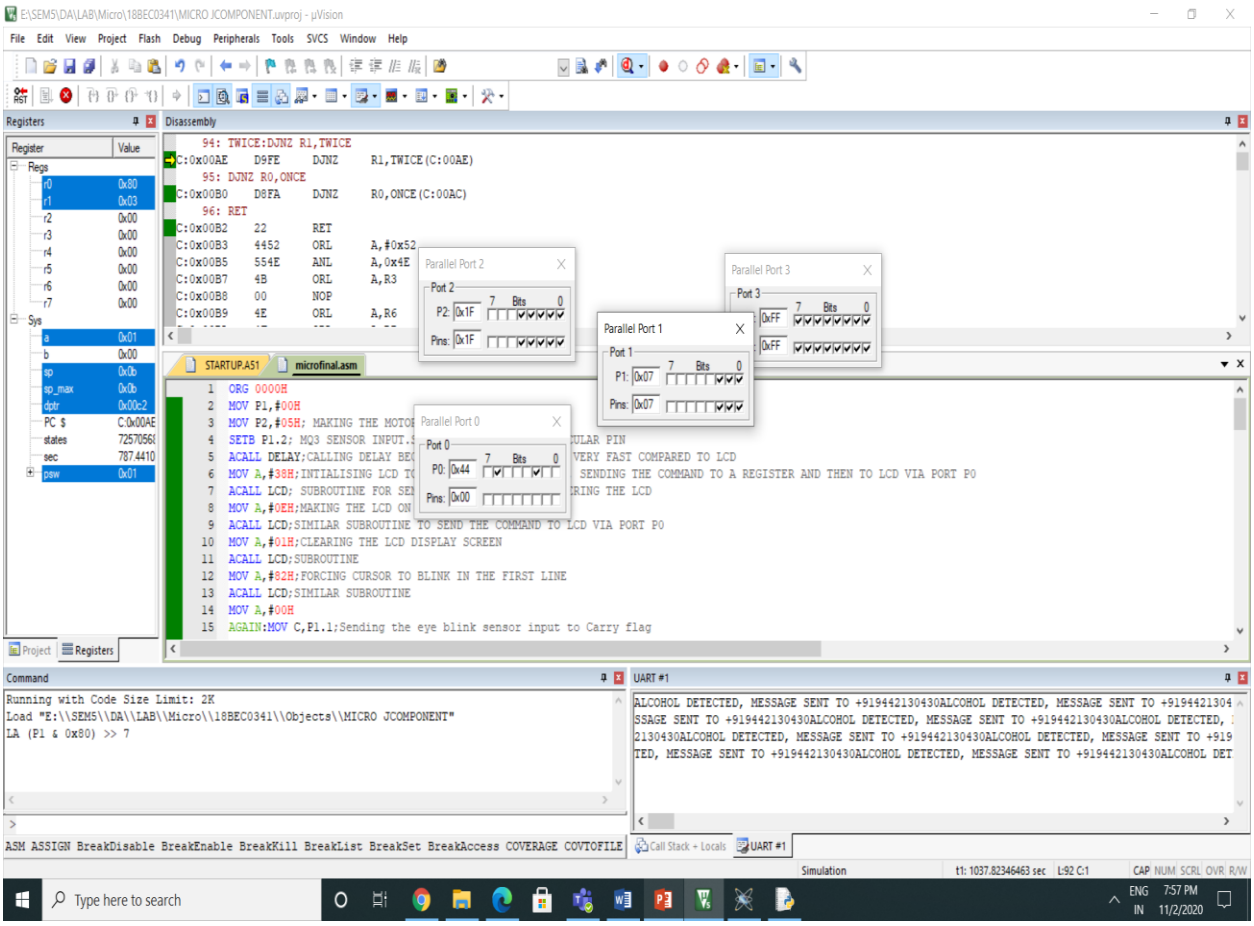
RESULTS:

CODE IN KEIL:



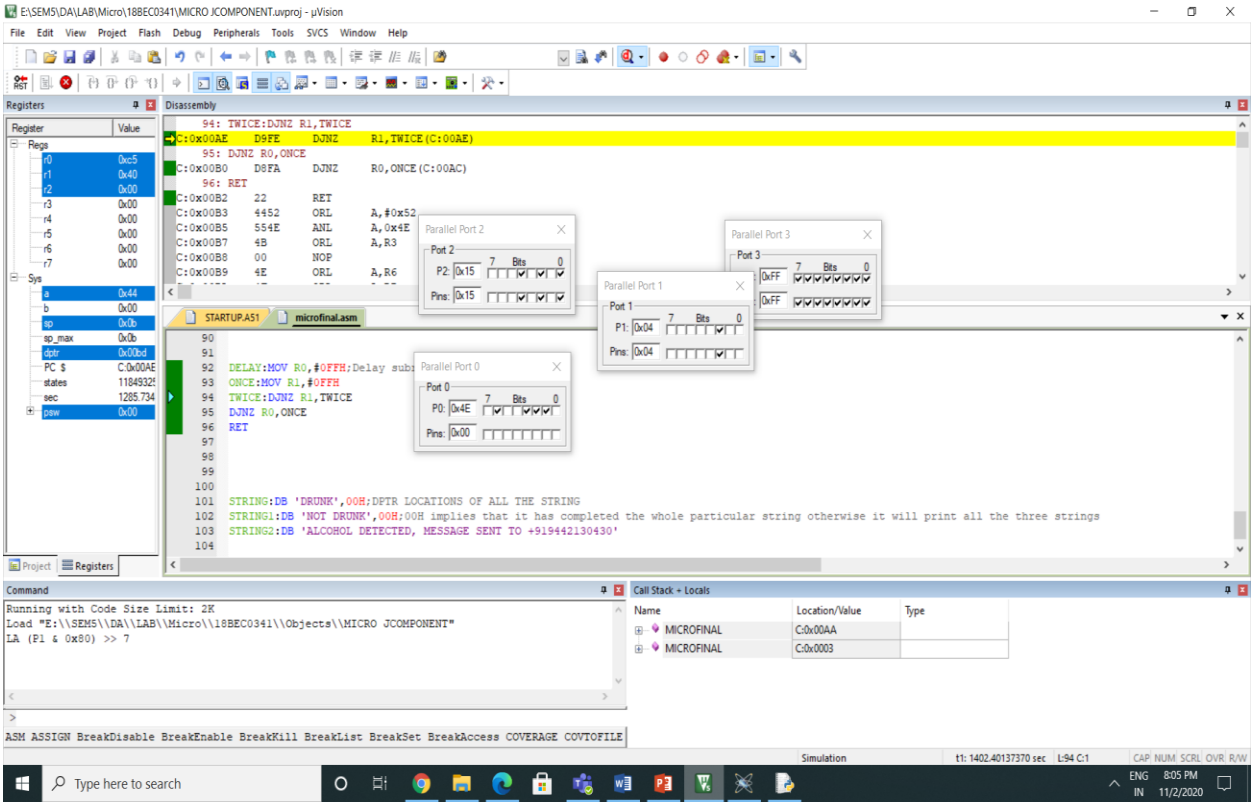
KEIL SIMULATION:

WHEN DRUNK:



We can clearly see the message is transmitted in the serial window.
The motor is off in P2 port and the buzzer is ON in the P1.0 port.

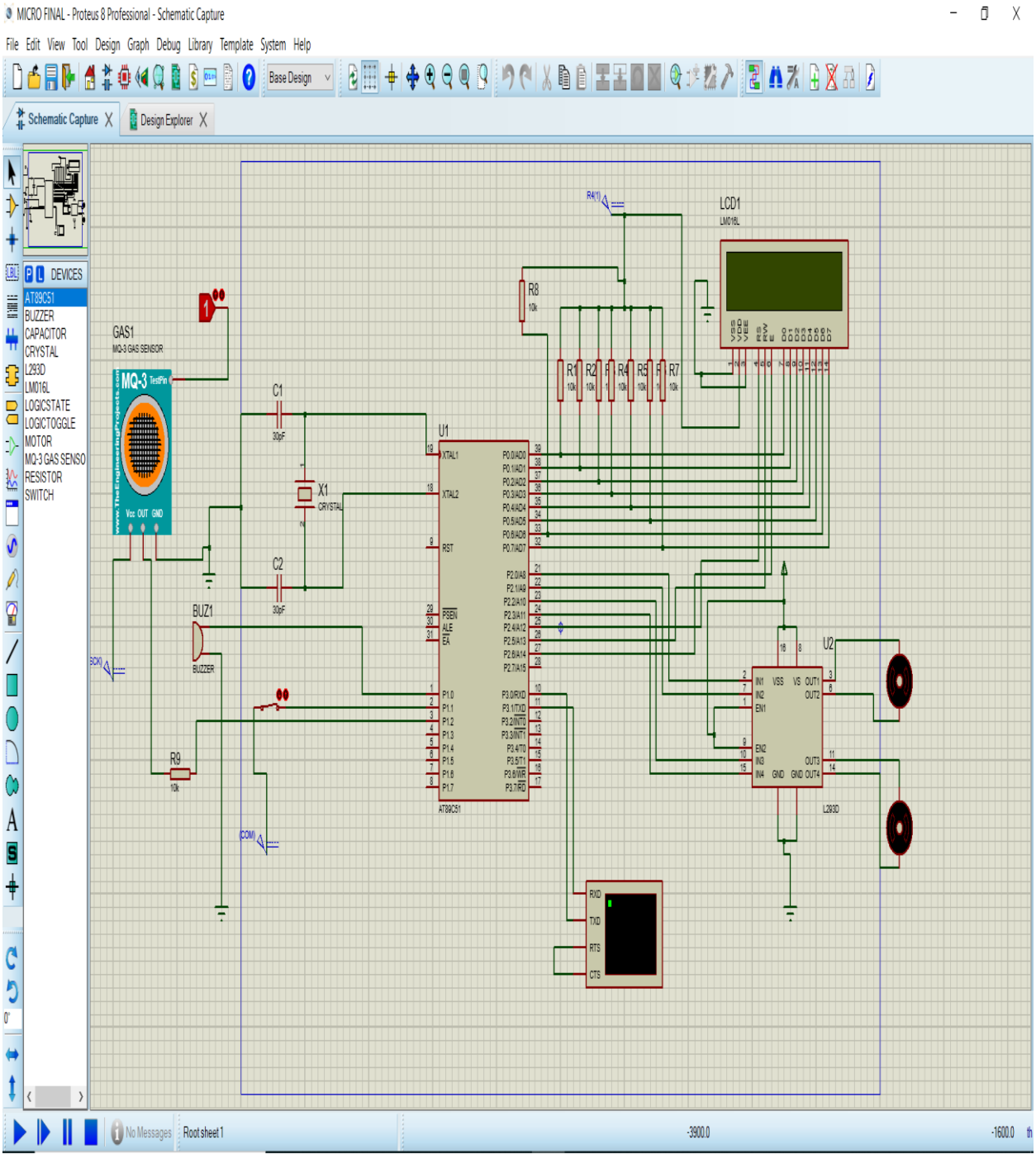
WHEN NOT DRUNK:



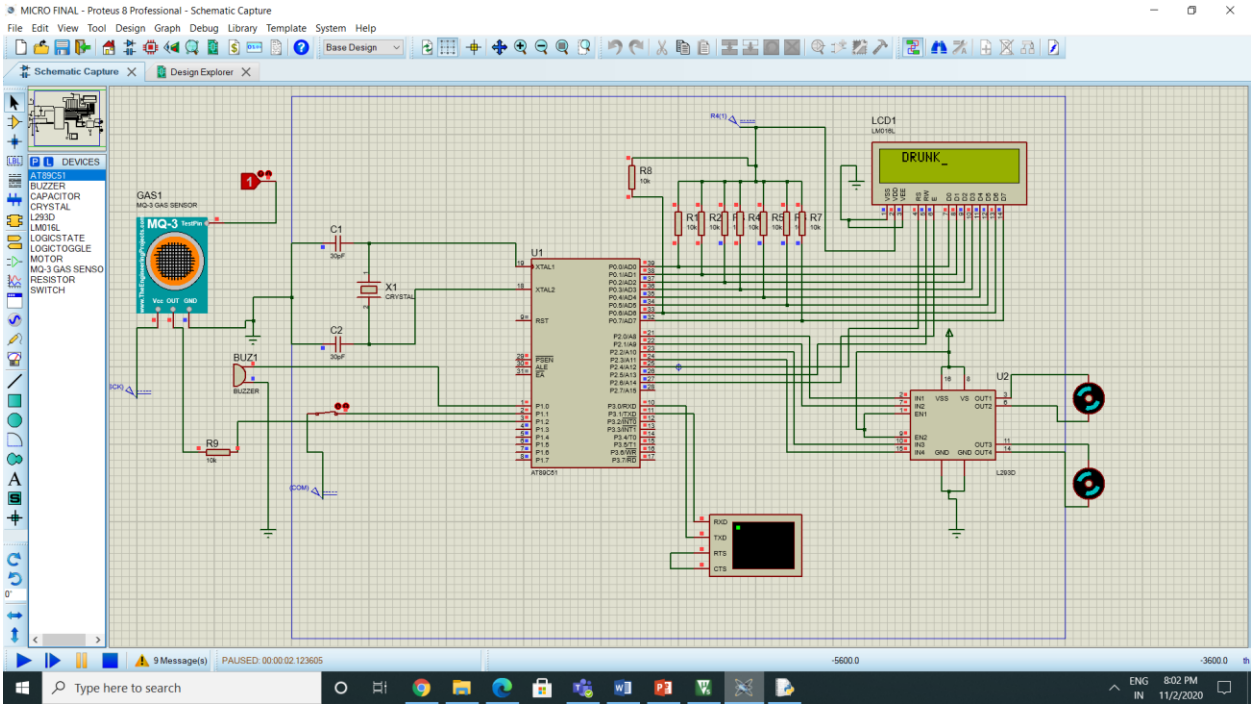
We can clearly see the motor is running ie. 05H and the buzzer is not turned ON.

PROTEUS SIMULATION:

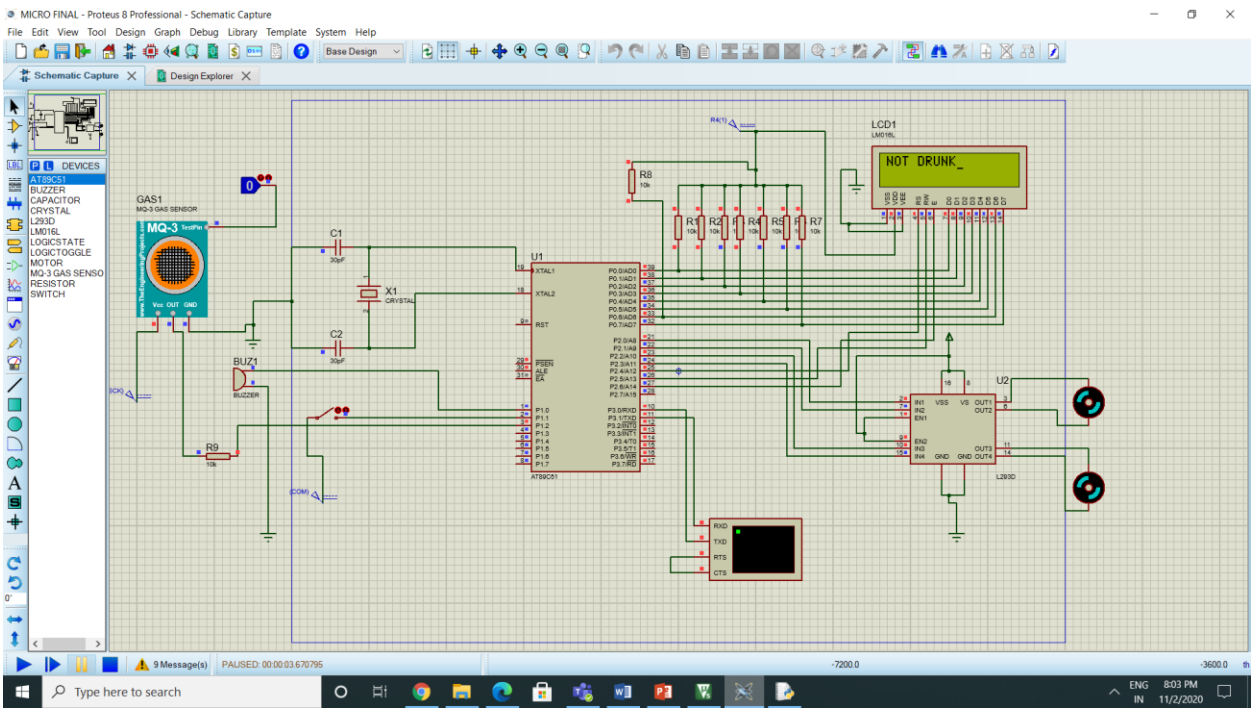
CIRCUIT:



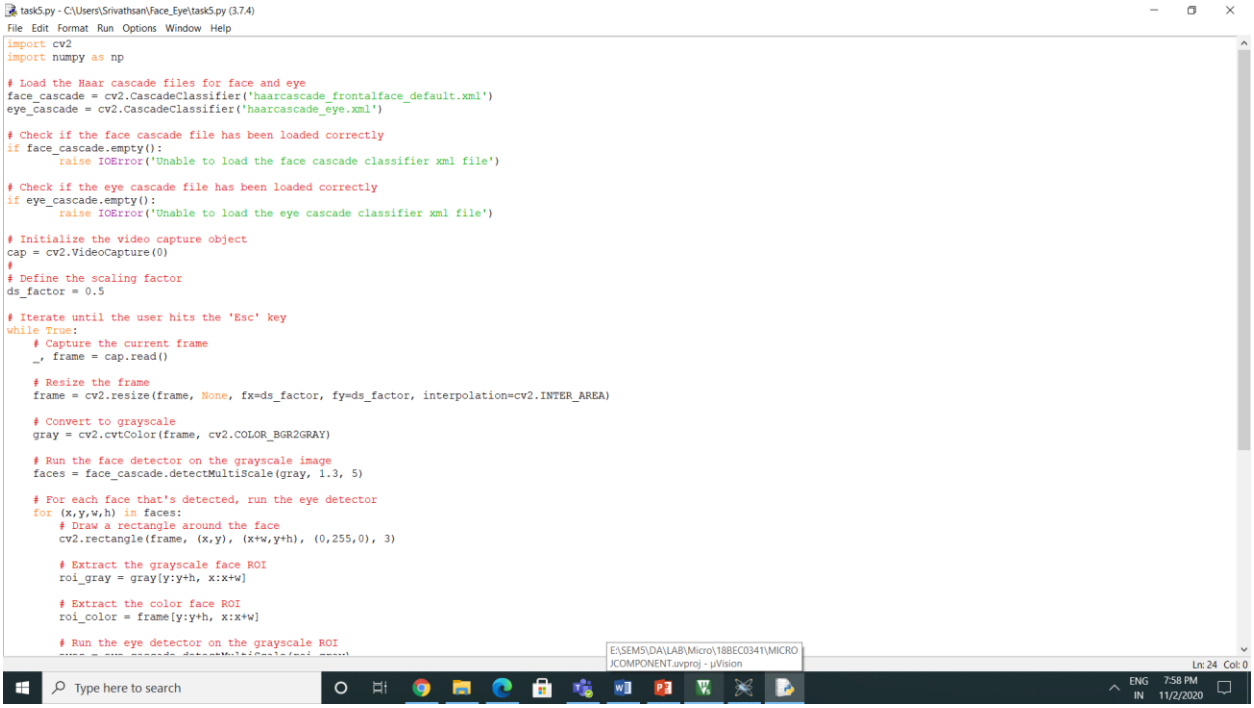
WHEN DRUNK:



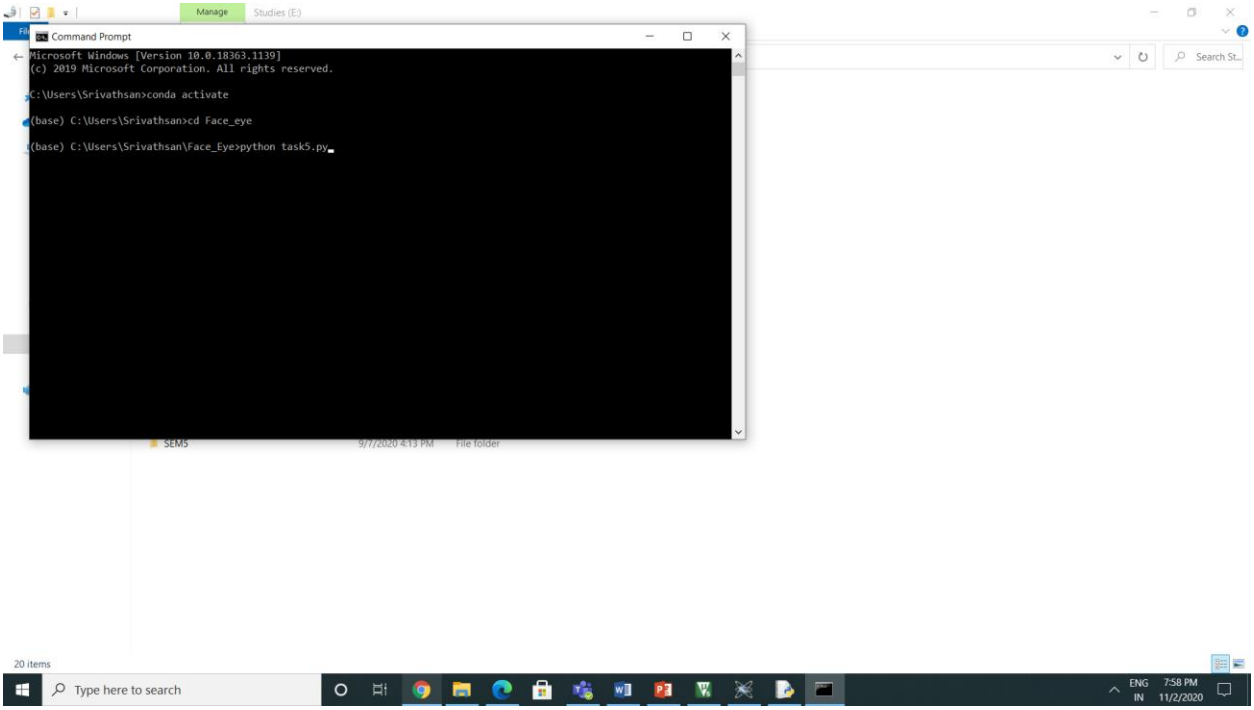
WHEN NOT DRUNK:



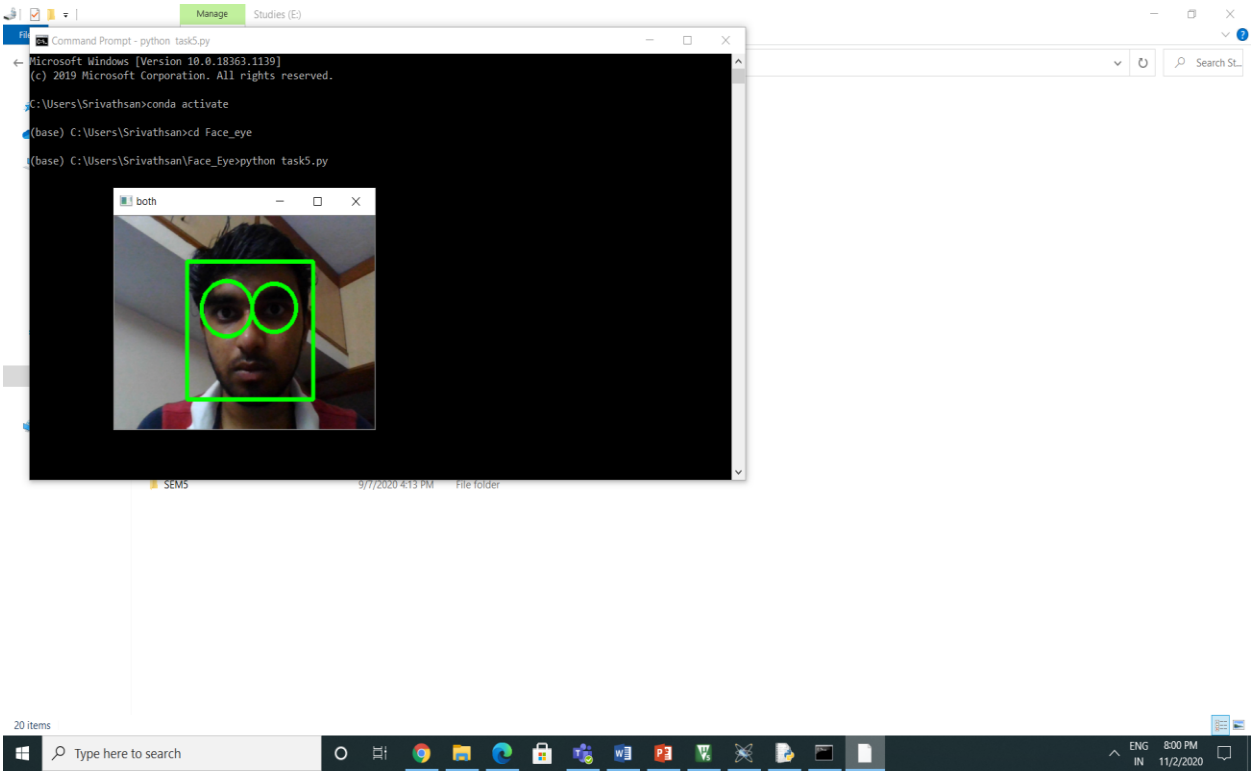
OPEN CV PYTHON CODE:



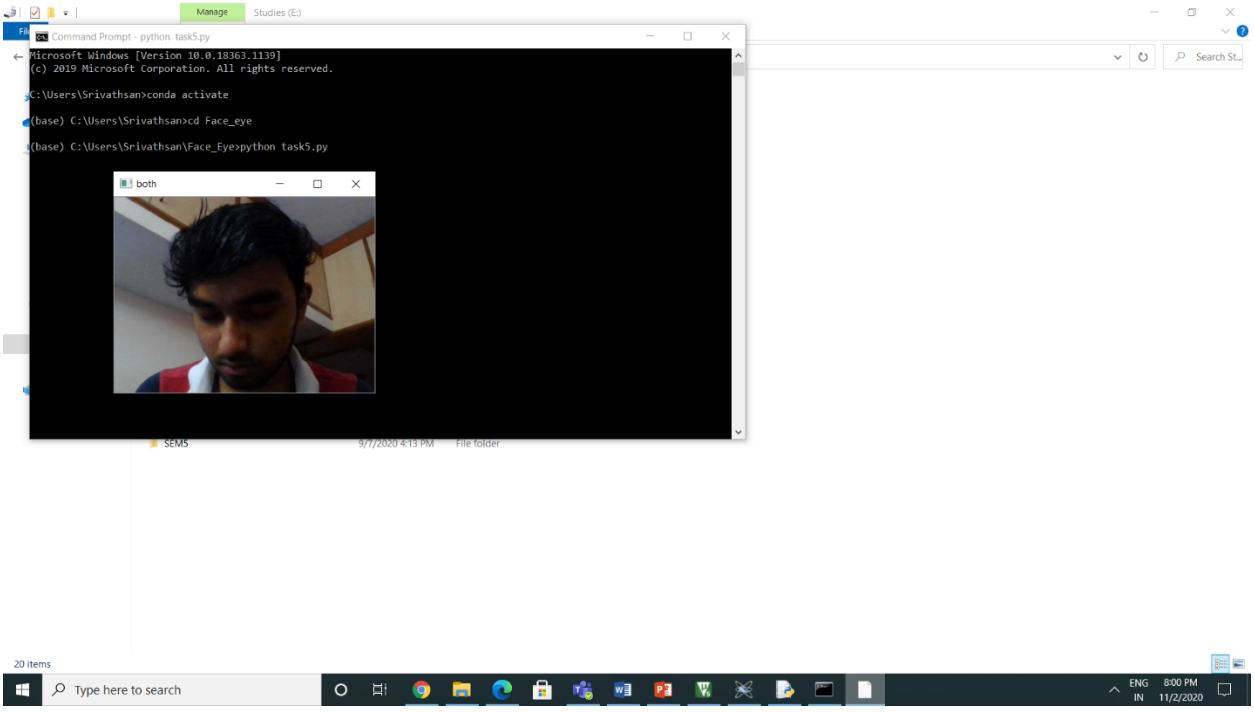
RUNNING IN THE COMMAND PROMPT



WHEN NOT DRUNK:



WHEN DRUNK OR DROWSY:



CONCLUSION:

We have created the SMART DRIVING SYSTEM to an extent that we can make with the help of KEIL and PROTEUS software and MQ3 Alcohol Sensor.
But we were not able to link the eye blink part for the COVID situation in the KEIL so instead we have used a switch, but we have done it separately using Open CV in python.
With the available requirements we will try to link both in the future.

REFERENCES:

- 8051- MAZIDI TEXTBOOK
- LCD INTERFACING AND DISPLAYING-HONGKONG VOCATIONAL CLASS OF EEE COURSE AND YOUTUBE VIDEO.
- PROTEUS SOFTWARE TUTORIAL
- MQ 3 SENSOR PACKAGE FOR PROTEUS
- ECE-4031 AIWP COURSE.
- Study of the Effects of Alcohol on Drivers and Driving Performance on Straight Road by XIAOHUA ZHAOS