DROWSY DRIVER DETECTION USING 8051

A Course Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electronics and Communication Engineering

by

Vinayak Prakash

(LTVE22EC074)

Computer Architecture and Microcontrollers

ECT 206

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

KERALA

May 2024

DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING TRIVANDRUM

2023 - 24

CERTIFICATE

This is to certify that the report entitled **DROWSY DRIVER DETECTION USING 8051** submitted by **Vinayak Prakash** (LTVE22EC074) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Electronics and Communication Engineering is a bonafide record of the course project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Joaquim Ignatious Monteiro
Assistant Professor
Dept.of ECE
College of Engineering
Trivandrum

Abstract

This document contains essential templates required to write technical reports using LaTeX. This template may be used for the preparation of B.Tech seminar reports of APJ Abdul Kalam Technological University, Kerala. Also minimum working examples to create equations, include figure, include table, table of contents symbols list and bibliographic citation in a LaTeX document are provided.

Please note that this template is provided without warranty on an AS IS basis.

JIM

Contents

Abstract **i**

List of Figures **iv**

1 Introduction **1**

1.1 Problem statement 1

1.2 Project objectives 2

2 Project Design and Testing **3**

2.1 CODING :KEILUVISION 3

2.1.1 CODE EXECUTION:keiluvision5 4

2.1.2 Generating hex file 4

2.2 BURNING CODE TO AT89S51 4

2.2.1 Arduino as ISP 4

2.2.2 Connecting Arduino with AT89S51 5

2.2.3 Uploading code to AT89S51 5

2.3 PROJECT CIRCUIT AND TESTING 6

2.3.1 Circuit Connections 7

2.3.2 Working of Code 7

3 Conclusion **9**

3.1 APPLICATIONS 9

4 Bill of Materials **11**

5 Code **12**

References **14**

List of Figures

2.1 Code debugging and execution 4
2.2 HEX FILE GENERATION 4
2.3 Burning of hexfile to AT89S51 5
2.4 Arduino as ISP 5
2.5 Connection circuit 6
2.6 IR Sensor Module 7
2.7 Buzzer 7
2.8 Circuit Diagram 8
2.9 Circuit 8

Chapter 1

Introduction

Road safety is a primary concern with driver fatigue being a major contributing factor to accidents. Drowsy driving can impair reaction time, judgment, and awareness, significantly increasing the risk of collisions. This project aims to develop a microcontroller-based drowsy driver detection system using an infrared (IR) sensor to enhance driver safety.

The 8051 microcontroller is a widely used microcontroller in embedded systems due to its simplicity, versatility, and ease of programming. By leveraging its capabilities, we can easily interface it with IR sensor module.

The major components in this project are AT89S51, a IR Sensor Module

1.1 Problem statement

Drowsy driving is a critical safety concern, contributing to a substantial number of road accidents. When drivers become fatigued, their reaction times decrease, judgment becomes impaired, and awareness of surroundings diminishes. This significantly increases the risk of collisions and potentially fatal outcomes.

Current solutions for drowsy driver detection may involve complex camera systems, physiological monitoring, or lane departure warning systems. However, these solutions can be expensive, intrusive, or have limitations in effectiveness.

There is a need for a simpler, more cost-effective, and driver-focused solution to detect drowsiness at an early stage. This system should provide an immediate and clear warning to the driver without requiring complex user interaction or additional visual

displays within the vehicle.

1.2 Project objectives

This project aims to develop a microcontroller-based drowsy driver detection system using an infrared (IR) sensor to address the problem of driver fatigue and enhance road safety. The specific objectives are:

- **Design a system that utilizes an IR sensor to detect closed eyes.** The system should be able to reliably differentiate between open and closed eyes based on the change in reflected IR light.
- **Implement an 8051 microcontroller to process sensor data and trigger alerts.** The microcontroller will analyze sensor readings and activate the alert mechanism if closed-eye detection persists for a predetermined period.
- **Develop an audible alert system using a buzzer.** The chosen alert method should be clear, immediate, and effective in grabbing the driver's attention when drowsiness is detected.
- **Maintain a cost-effective and user-friendly design.** The system should prioritize affordability and ease of use while achieving its core functionality.
- **Promote driver awareness of drowsiness.**The timely buzzer alert aims to jolt the driver back to alertness and encourage them to take appropriate actions like pulling over for a rest.

Chapter 2

Project Design and Testing

By achieving these objectives, the project aims to deliver a fully functional and user-friendly drowsy driver detector

THIS SECTION INCLUDES: **SOFTWARE IMPLEMENTATION**

*WRITING CODE IN 8051 IN KIELuVISION

*DEBUGGING CODE ON ANY ERROR

*SIMULATION

HEX FILE GENERATION

HARDWARE IMPLEMENTATION

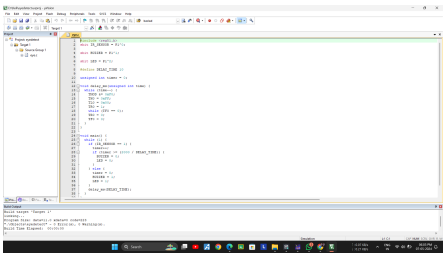
BURN HEXFILE TO 8051 MICROCONTROLLER(USING ARDUINO AS ISP) CONNECT ALL OTHER COMPONENTS TO MAKE THE CIRCUIT FOR WORKING OF PROJECT.

2.1 CODING :KEILUVISION

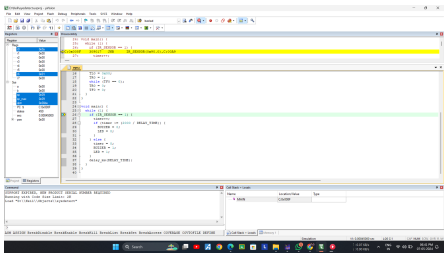
i have made the code using embedded C program,C language is generally preferred BECAUSE C language code is often more readable and easier to understand compared to assembly language code. It uses higher-level constructs, such as functions, variables, and structured control flow, which make the code more modular and maintainable. I have used 8051 keiluvision for simulation.

2.1.1 CODE EXECUTION:keiluvision5

I entered the embedded c code in keil.The code was executed and errors were rectified. Then simulation was done using keil.



(a) code execution

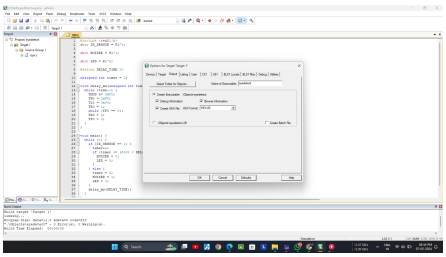


(b) Code debugging

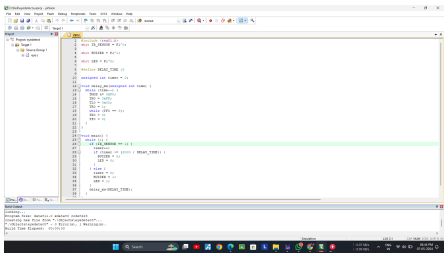
Figure 2.1: Code debugging and execution

2.1.2 Generating hex file

hex file is generated for buring code to AT89S51 microcontroller hex file is created after every error is rectified



(a) creating hex file



(b) hex file generated

Figure 2.2: HEX FILE GENERATION

2.2 BURNING CODE TO AT89S51

2.2.1 Arduino as ISP

Set up the Arduino as an ISP: Connect the Arduino board to the computer and upload the ArduinoISP sketch to the Arduino.

2.2.2 Connecting Arduino with AT89S51

Connect the AT89S51: Make the necessary connections between the Arduino and AT89S51 microcontroller. This typically involves connecting the SPI pins (MISO, MOSI, SCK) of the Arduino to the corresponding pins of the AT89S51 (P1.6, P1.5, P1.7). Additionally, connect the Reset pin (RST) of the Arduino to the Reset pin (P3.5) of the AT89S51. connect crystal oscillator,33uf capacitors to 18 and 19 pin of at89s51 connect reset circuit to the 9th pin of at89s51

2.2.3 Uploading code to AT89S51

avrdude.exe file is downloaded and stored and path of location is noted. The location

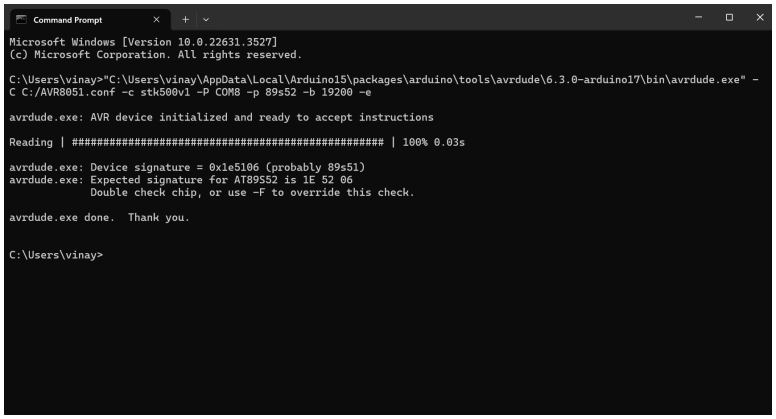


Figure 2.3: Burning of hexfile to AT89S51

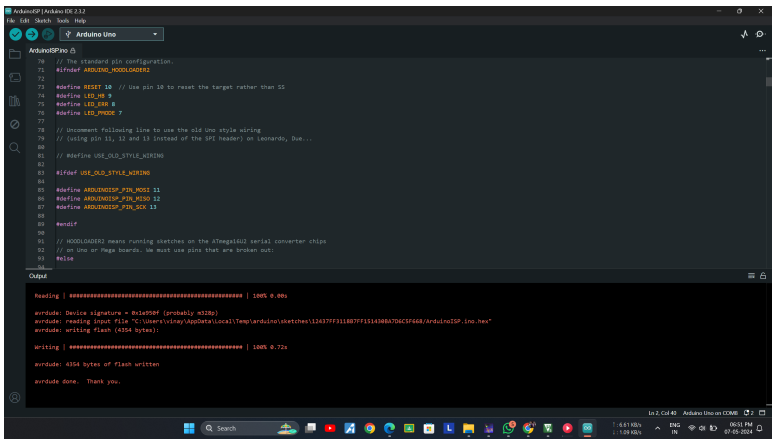


Figure 2.4: Arduino as ISP

path of avrdude,microcontroller model(AT89S51 or AT89S52) ,output port number (arduino connected port) is inserted at corresponding places in the code in command

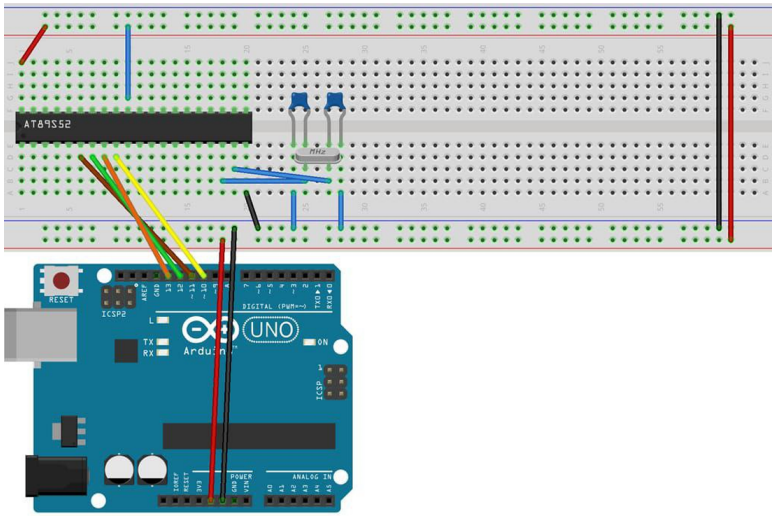


Figure 2.5: Connection circuit

prompt window(as in figure below) and then the code is made to run and the hexfile is burned to microcontroller

2.3 PROJECT CIRCUIT AND TESTING

For this project the components required are:

1. AT89S51 microcontroller
2. IR sensor Module
3. Breadboard
4. Connecting wires
5. Crystal Oscillator 11.0592 Mhz
6. Capacitors (33pf,10uf)
7. Resistor 10k
8. Buzzer

2.3.1 Circuit Connections

IR sensor: The DOUT pin of the IR sensor module is connected to the 1st pin(PORT 1.0) of the Microcontroller. VCC, GND are connected to the 5V input and to the ground respectively.

Figure 2.6: IR Sensor Module

Buzzer: The anode of the buzzer is connected to 2nd pin (PORT 1.1) of the microcontroller. Cathode connected to the GND.

Figure 2.7: Buzzer

Other connections: Vcc of the microcontroller shorted with 31th pin (EA pin). Crystall Oscillator is connected to 18th,19th pin of MC.

2.3.2 Working of Code

The IR sensor is connected to pin P1.0, and a buzzer is connected to pin P1.1. The code defines a delay time constant and initializes a timer variable. A delayms function generates delays in milliseconds using Timer 0. In the main loop, the program continuously checks the IR sensor's status. If an object is detected, the timer is incremented. If the timer reaches a certain value (2 seconds), the buzzer is turned off. If no object is detected, the timer is reset, and the buzzer is turned on. Timer 0 is used to create the delay, operating in 16-bit mode. The program waits for Timer 0

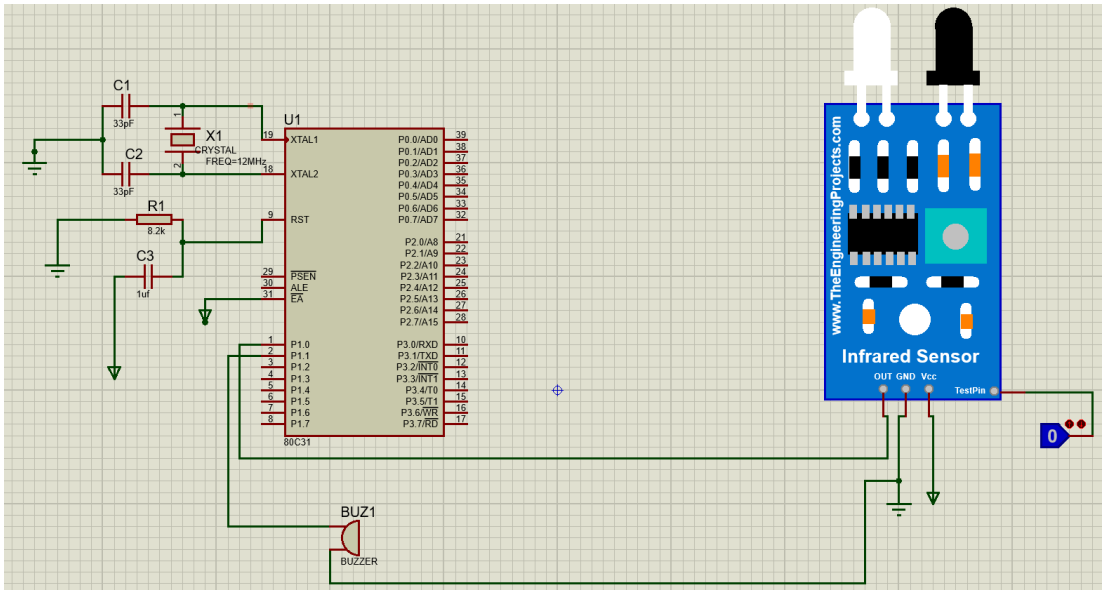


Figure 2.8: Circuit Diagram

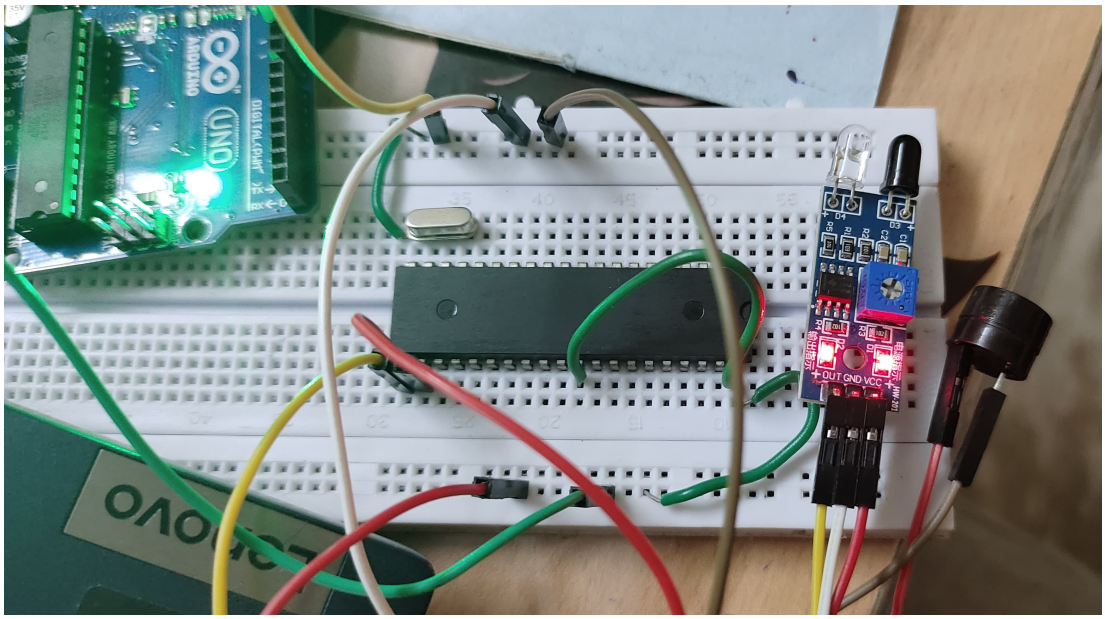


Figure 2.9: Circuit

to overflow to indicate that the delay has elapsed. Timer 0 is then stopped, and the overflow flag is cleared for the next operation.

Chapter 3

Conclusion

This project successfully developed a microcontroller-based drowsy driver detection system using an infrared (IR) sensor. The system leverages the principle that closed eyes alter the reflection of IR light, allowing the 8051 microcontroller to detect drowsiness based on sensor readings. When closed-eye detection persists for a predetermined period, the system triggers an audible alert via a buzzer. This immediate and clear notification aims to jolt the driver back to alertness and encourage them to take corrective action.

The project demonstrates the feasibility of a simple and cost-effective solution for mitigating drowsy driving risks. The system offers several advantages, including:
Affordability: The use of an IR sensor and 8051 microcontroller keeps the system cost-effective. Ease of Use: The system requires minimal user interaction and no additional visual displays within the vehicle.
Driver Focus: The immediate and unavoidable buzzer alert directly addresses the driver's state of alertness.

3.1 APPLICATIONS

- **Commercial Trucking Industry:** Enforced regulations and long haul journeys make truck drivers particularly susceptible to fatigue. This system can be integrated into commercial trucks to provide timely alerts and potentially link to fleet management systems for monitoring driver behavior.

- **Public Transportation:** Bus drivers and train operators navigate long routes and potentially monotonous schedules. Integrating this system into public transportation vehicles can enhance safety for both drivers and passengers.

- **Long-Distance Driving Scenarios:** Road trips or journeys on isolated highways can lead to driver drowsiness. This system can be a valuable addition to rental cars or personal vehicles used for long-distance travel.

- **Off-Road Applications:** Construction vehicles, agricultural equipment, and mining machinery often operate in demanding environments with long shifts. This system can be adapted for such vehicles to improve operator alertness and safety.

- **Driver Training and Monitoring Systems:** Driving schools and professional driver training programs can incorporate this system as a training tool to help drivers recognize and address drowsiness symptoms. Additionally, fleet management companies can utilize this technology for real-time monitoring and intervention when drowsiness is detected.

Chapter 4

Bill of Materials

SI.No	Item	Manufacturer	Price/Unit (Rs.)	Quantity	Cost (Rs.)
1	80S51	Atmel	150	1	150
2	Capacitor(33pF)	Keltron	1	2	2
3	Capacitor(10uf)	Keltron	10	1	10
4	Breadboard	Esel International	90	1	90
5	Buzzer	Electronic Spices	15	1	15
6	IR sensor	Generic	150	1	150
7	Oscillator(11.0592MHz)	Generic	8	1	8
8	Resistor(10k)	Elevctronic devices	2	1	2

Chapter 5

Code

```
#include <reg51.h>
sbit IR_SENSOR = P1^0;

sbit BUZZER = P1^1;

#define DELAY_TIME 10

unsigned int timer = 0;

void delay_ms(unsigned int time) {
  while (time--) {
    TMOD &= 0xF0;
    TH0 = 0xFF;
    TL0 = 0x00;
    TR0 = 1;
    while (TF0 == 0);
    TR0 = 0;
    TF0 = 0;
  }
}

void main() {
  while (1) {
    if (IR_SENSOR == 1) {
      timer++;
      if (timer >= (2000 / DELAY_TIME)) {
```

```
27              BUZZER = 0;
28
29            }
30        } else {
31            timer = 0;
32            BUZZER = 1;
33
34        }
35        delay_ms(DELAY_TIME);
36    }
37 }
```

Listing 5.1: Source code of the project

References

[1] Gemini AI, ChatGPT

[2] https://www.electronicsforu.com/technology-trends/learn-electronics/ir-led-infrared-sensor-basics

[3] https://www.tutorialspoint.com/microprocessor/microcontrollers-8051-pin-description.html