Automobile Control System Using ATmega32 Microcontroller

This project was developed as part of the **Computer and Microprocessor Structure Lab**, focusing on the practical application of concepts taught in the course. The aim was to design and implement a functional automobile control system using the ATmega32 microcontroller. This project allowed students to explore real-world applications of microprocessor architecture, embedded systems programming, and hardware-software integration.

The automobile control system combines dynamic motion control, real-time monitoring of environmental parameters, and safety mechanisms into a cohesive system. By integrating peripherals such as motors, sensors, an LCD, and a buzzer, the microcontroller serves as the central unit for managing and synchronizing various functionalities. Advanced techniques such as Pulse Width Modulation (PWM), Analog-to-Digital Conversion (ADC), and interrupt-driven programming were employed to achieve precise control and responsiveness.

Project Overview

The system was designed to simulate the control of a small-scale vehicle with capabilities for forward and backward motion, turning, and speed adjustment. It uses four DC motors connected via an L298 motor driver, with the ATmega32 microcontroller managing their operation. PWM signals are used to dynamically adjust motor speeds, ensuring smooth acceleration and deceleration.

Key features include a temperature monitoring system using an LM35 sensor, which stops the vehicle if the temperature exceeds a critical threshold (90°C), and a fuel monitoring mechanism simulated via an analog input. Fuel levels are displayed using LEDs, with a warning LED lighting up when levels drop below a certain point.

The system also includes a real-time clock implemented with internal timers, displaying the current time on an LCD along with other critical information like vehicle status and temperature. A hand brake mechanism, activated via an external interrupt, provides immediate halting of the vehicle, enhancing the system's safety measures.

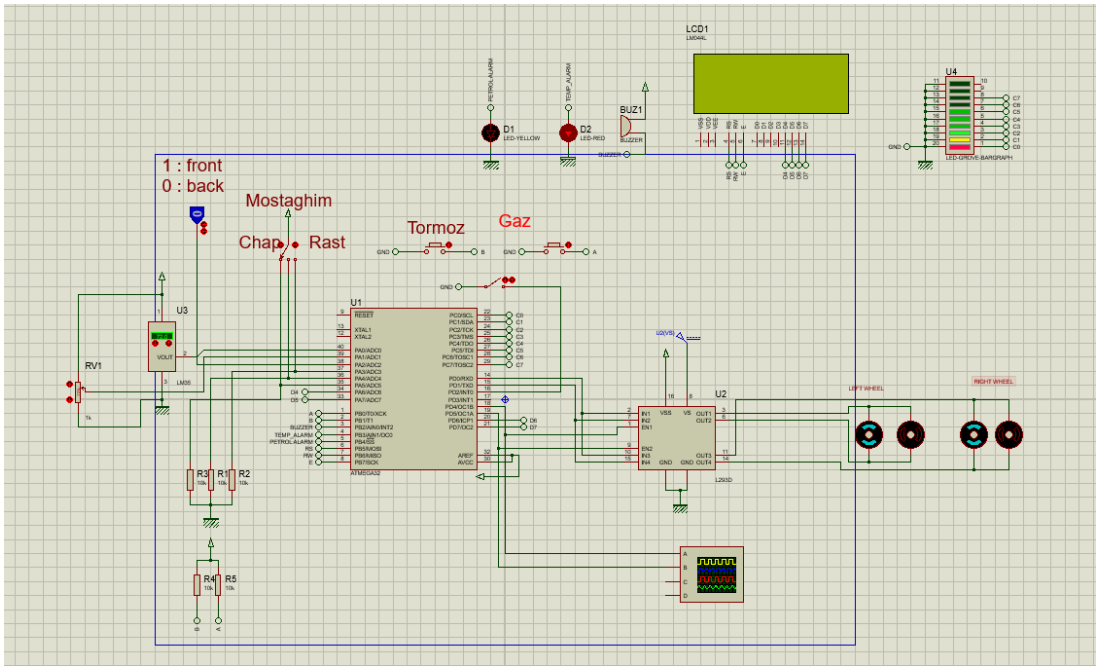
Hardware Components

The project utilized the following hardware:

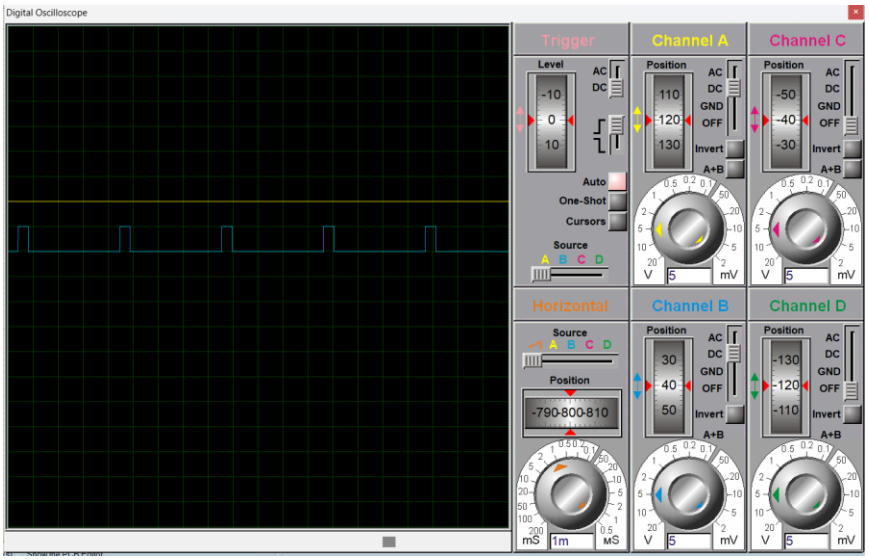
1. **ATmega32 Microcontroller:** The central unit controlling all operations.
2. **DC Motors and L298 Driver:** For movement in forward, backward, left, and right directions.
3. **Temperature Sensor (LM35):** For real-time temperature monitoring.
4. **LCD (16x2):** To display time, temperature, and vehicle status.

5. **LEDs and Buzzer:** For visual and auditory alerts, such as fuel warnings or high-temperature notifications.

The figure below illustrates the Proteus simulation of the system, highlighting how all components interact:



The following image illustrates the PWM signals on the oscilloscope during operation. The square waveforms represent the duty cycles that dictate motor speed, with higher duty cycles corresponding to increased speed.



Software Implementation

The software was written in C using AVR-GCC. The program logic included:

1. **Motion Control:** Functions like front() and back() adjust motor directions, while PWM signals regulate speed.
2. **Real-Time Clock:** Timer interrupts update the time every second, displaying it on the LCD.
3. **Fuel Monitoring:** The benzin() function reads analog values from a simulated fuel sensor, processes them, and updates the LEDs.
4. **Temperature Control:** The temp() function converts sensor readings into temperature values in °C and triggers safety mechanisms when thresholds are crossed.
5. **Interrupt Handling:**

   An external interrupt (INT0) toggles the hand brake.

   Timer interrupts manage periodic tasks like clock updates and sensor monitoring.

System Workflow

The control logic operates in a continuous loop, dynamically adjusting the vehicle's motion based on input signals:

- Forward and backward motion, as well as turning, are managed by modifying PWM outputs for the motors.
- If the temperature exceeds 70°C, a warning buzzer and LED are activated. At temperatures above 90°C, the motors are stopped.
- Fuel levels are displayed using LEDs, with a warning light indicating low levels.
- The hand brake, triggered by an external interrupt, immediately halts the vehicle by setting PWM outputs to zero.

The LCD provides real-time feedback, displaying the current time, vehicle direction, speed, and temperature. It also shows alerts for critical events, such as low fuel or high temperatures.

Results

The automobile control system functioned as intended, showcasing the following results:

1. Precise motion control, with smooth speed adjustments using PWM.
2. Reliable monitoring and alerting systems for temperature and fuel levels.

3. Seamless interaction between hardware and software components, ensuring real-time responsiveness.
4. Robust implementation of safety mechanisms, such as the hand brake and over-temperature halt.

The project demonstrated the significance of interrupt-driven programming in handling real-time tasks efficiently. By effectively integrating multiple sensors and outputs, the system highlighted the potential of microcontroller-based designs in creating intelligent and interactive applications.

Conclusion

This project exemplifies the practical application of microcontroller and embedded systems concepts in designing a functional and safe automobile control system. By integrating features like motion control, real-time monitoring, and safety mechanisms, the project provided an excellent learning experience in system design, programming, and hardware interfacing.

Future improvements could include adding wireless control for remote operation, GPS tracking for navigation, and advanced sensors for enhanced automation. This project serves as a solid foundation for exploring more complex applications in embedded systems and real-time control.