CSE411: Real-Time and Embedded Systems Design

Presented to Dr. Sherif Hammad, Eng. Ahmed Hassabou,

& Eng. Ayman Baharia

May 17, 2025

Team (34)

Name	**ID**
Omar Ahmed Ahmed AbdAlAal	**18P2998**
Ahmed Mohamed Elmahdy	**2100723**
Mohamed Ayman	**19P9201**
Abanoub Sami Samir	**21P0356**
Ezzeldine Ahmed Ali	**21P0078**

Abstract

This report documents the design, implementation, and functionality of an embedded control system for an automotive safety system. The system uses three sensor inputs: a mechanical end switch, an analog sensor (via ADC) to measure speed, and an ultrasonic sensor to measure distance. The door state is determined by a combination of these inputs according to defined override rules.

Contents

1 Code Repository

`https://github.com/omar4a/RTOS`

2 Demo Video

`https://drive.google.com/file/d/1m01r5FwGq2cBsoCEca7Sovbl9MRNLC8M/view?usp=sharing`

3 Introduction

The purpose of this embedded system is to implement some automotive safety mechanisms:

1. **Automatic Door Lock:** Toggles when the car exceeds a threshold speed.
2. **Rear Parking Assistance:** Alerts the driver if the car is dangerously close to an object.
3. **Automatic Door Unlock:** Toggles when the car's ignition is stopped.

The system uses FreeRTOS for task scheduling and uses interrupt service routines to handle different events. Peripheral initializations, NVIC configuration, and various tasks are implemented in C.

4 System Architecture

4.1 Hardware Components

- A TM4C microcontroller.
- A mechanical end switch.
- An analog sensor (potentiometer).
- A DIP switch.
- An ultrasonic sensor.
- A buzzer and an RGB LED for audio/visual feedback.
- An LCD display.

4.2 Software Components

The software is structured around several FreeRTOS tasks:

`vADCTask`: Triggers ADC conversions, waits for conversion completion, and updates a global speed variable and LCD with the new speed measurement.

`vDoorLockTask:` Periodically polls the mechanical end switch, the ADC-based speed value, and the DIP switch to compute the desired door state. The task then updates the LCD display accordingly.

`vUltrasonicTask:` Reads the distance measured by the ultrasonic sensor and updates the LCD.

`vBuzzerTask:` Adjusts the beep frequency and RGB LED color based on ultrasonic distance measurements.

`vTransmissionTask:` Receives and processes mode updates via a queue.

Additionally, an interrupt service routine `ADC0SS0_Handler` is used to process ADC conversion results asynchronously.

5 Conclusion

This document presented a thorough explanation of a simulated automotive safety system implemented on a TM4C microcontroller with the FreeRTOS. The system determines door status using three inputs: a mechanical end switch, an ADC-based speed sensor that forces the door to close when speed exceeds a given threshold, and a DIP switch that triggers a door-open override if turned off. In addition, other tasks manage ultrasonic sensor readings, buzzer output, LED control, and transmission mode updates.

6 Future Work

Future enhancements could include:

- Implementing hardware debouncing or more sophisticated software filtering for improved sensor readings.
- Converting some of the polling-based mechanisms into interrupt-driven methods to reduce latency.
- Logging system events for debugging purposes.