Elevator Control System

Ihab Mohamad
200401862

ENSE-452 Project Report
December 5, 2024

Table of Contents

Abstract

The Elevator Control System is an embedded software project designed and implemented on an STM32 microcontroller using FreeRTOS. This project focuses on simulating a fully functional elevator system controlled through a Command-Line Interface (CLI). The system supports features like floor selection, emergency stop handling, maintenance mode, and an elevator call mechanism. The software demonstrates robust multitasking, real-time updates, and user-friendly CLI feedback using ANSI escape sequences.

Introduction

Elevator systems are critical in modern infrastructure, requiring reliability, safety, and user-friendly controls. This project aims to simulate a simplified elevator system using an STM32 microcontroller with the following objectives:

- Develop a multitasking elevator system using FreeRTOS.
- Provide interactive user control via a CLI.
- Implement essential safety features, such as an emergency stop.
- Allow for maintainability

The system supports up to five floors and features dynamic CLI updates, emergency handling through an external button, and a maintenance mode for diagnostics. Users can also call the elevator to their current floor, demonstrating practical usability.

System Design

Hardware Setup

1. Microcontroller: STM32 with FreeRTOS.
2. External Components:
   - Emergency stop button connected to pin PB0.
   - UART communication for CLI interaction.

Software Components

1. **CLI Commands**:
   - open: Opens the elevator door.
   - close: Closes the elevator door.
   - [1-5]: Calls the elevator to the specified floor.
   - E: Activates the emergency stop.
   - M: Engages maintenance mode.
2. **FreeRTOS Tasks**:
   - Floor Selection Task: Handles user inputs for floor selection or elevator calls.
   - Elevator Movement Task: Moves the elevator to the requested floor.

   - Emergency Button Polling Task: Monitors the emergency stop button and manages corresponding actions.

System Features

1. **Emergency Stop**:
   - Activates via an external button.
   - Halts the elevator immediately and opens the doors.
   - Can be released by pressing the button again, but the elevator remains idle until reset. By pressing on the black button on the STM32.
   -
2. **Maintenance Mode**:
   - Activates via the CLI by entering M.
   - Displays a status of "Out of Order (Maintenance)".
   - Requires maintenance personnel to enter exit to release the elevator.
   - Upon exiting, the elevator will need to be reset by pressing on the black button on the STM32. for it becomes ready for normal operation.
3. **Elevator Call**:
   - Allows users to summon the elevator to their floor by entering the floor number.
   - The elevator updates its status dynamically and moves to the requested floor.

Implementation

Initialization

The system initializes with the following:

- FreeRTOS tasks for floor selection, elevator movement, and emergency polling.
- A queue for managing floor requests.
- UART for CLI-based communication.

Functional Workflow

1. **Startup**:
   - The system initializes and displays the elevator status.
   - Prompts users to enter commands via the CLI.
2. **Floor Selection**:
   - Users input the desired floor, and the system dynamically updates the elevator's position and state.
3. **Emergency Stop**:

    - The external button halts the elevator at the current or nearest floor.
    - Doors open automatically, and the system displays "Emergency Stop" in the CLI.
4. **Maintenance Mode**:
    - Maintenance personnel enter M to engage the mode.
    - Exiting the mode resets the elevator for normal operation.
5. **Elevator Call**:
    - Users call the elevator by entering their floor number. The system handles the movement and opens the doors upon arrival.

Results

The Elevator Control System demonstrates the following:

1. **Real-Time Performance**:
    - Smooth multitasking with FreeRTOS.
    - Accurate floor transitions and prompt CLI feedback.
2. **Safety**:
    - Reliable emergency stop mechanism with immediate door operation.
3. **Usability**:
    - Intuitive CLI commands for calling and controlling the elevator.
    - Maintenance mode for easy diagnostics and controlled reset.

Conclusion

The Elevator Control System successfully simulates a safe, reliable, and user-friendly elevator operation on an STM32 microcontroller. By integrating multitasking, emergency handling, and maintenance functionality, the project highlights practical applications of embedded systems in real-world scenarios. Future enhancements will further improve usability and scalability, making this a robust solution for elevator control in embedded systems.