Elevator Controller

Antoine Karam, Ghady Youssef

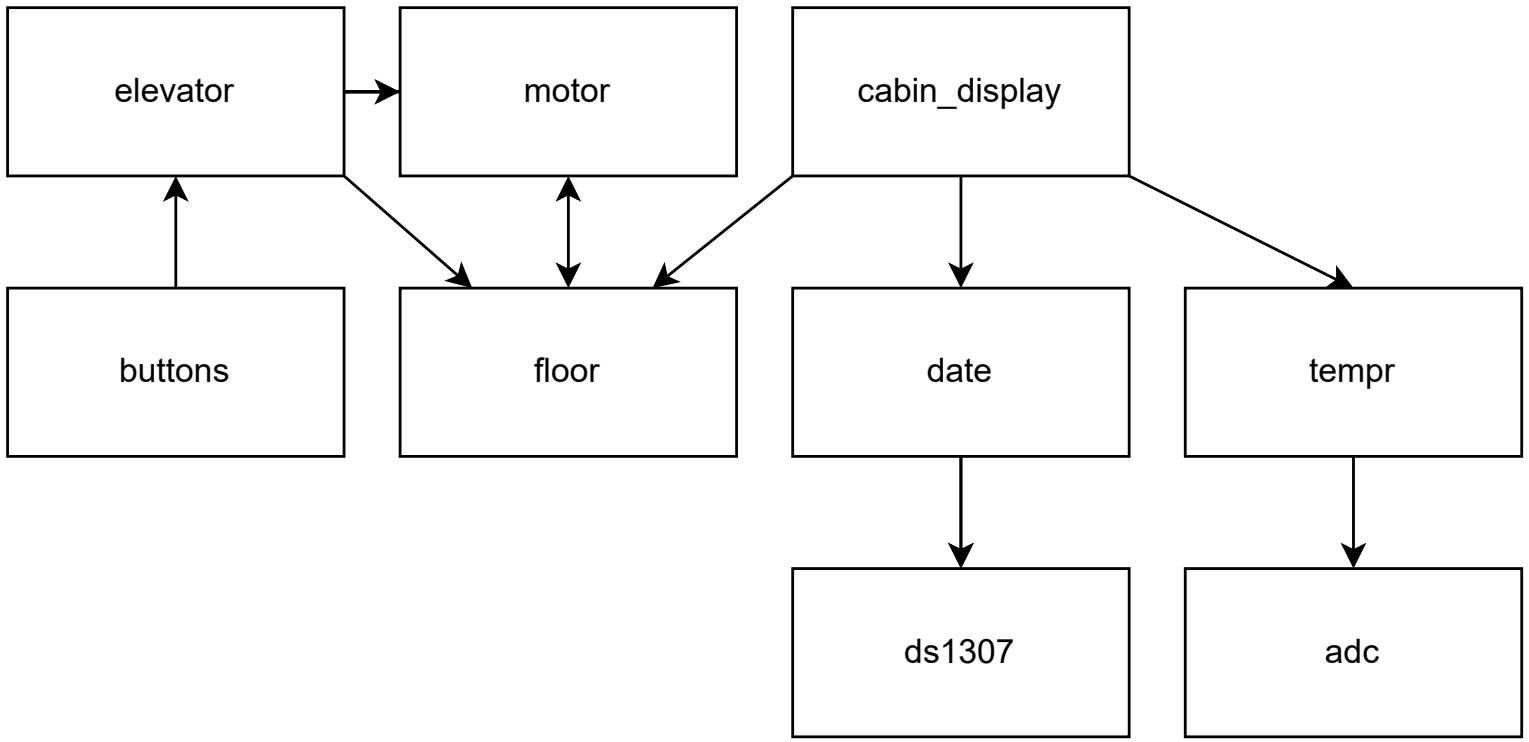
Saint Joseph University of Beirut

January 10, 2025

Introduction

- Overview of the project
- Objectives and goals
- Design considerations

Firmware Architecture



Drivers

- **Temperature: LM35**
  - Measures temperature
  - Connected to ADC
  - Two functions for temperature retrieval
- **Real-Time Clock: DS1307**
  - Retrieves time and date
  - Utilizes I^2C communication
  - API provides functions to access date and time

Display

- **Floor Display**
  - Shows current floor
  - Monitors the switches on each stop to update global state
- **Cabin Display**
  - Displays time, date, and temperature
  - Cycles every 10 seconds

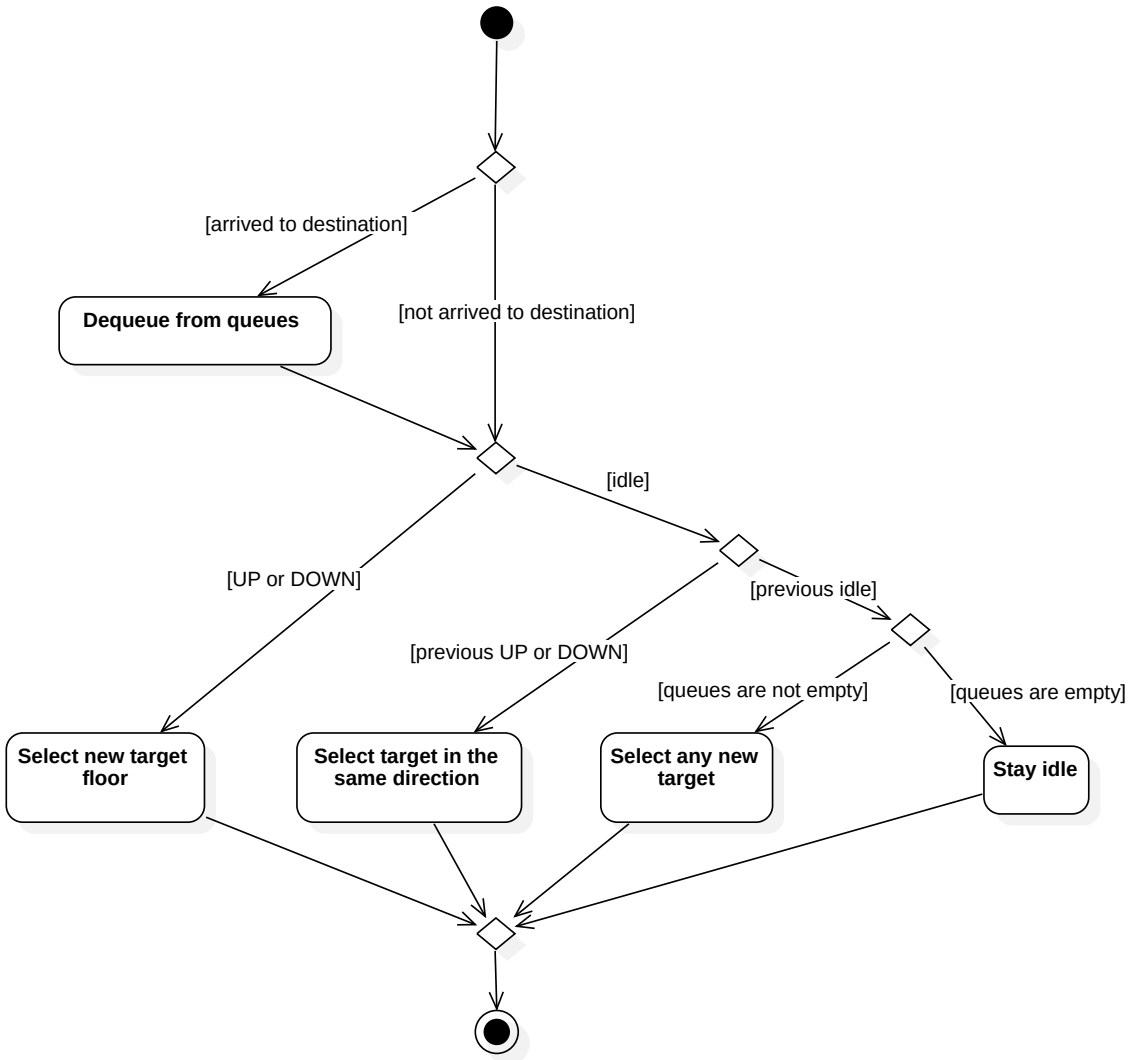
Motor

- Moving the cabin between floors
- Critical for scheduler and floor display
- Prevent abrupt direction changes

Scheduler

- Determine next floor for the elevator
- Avoid abrupt direction changes for safety
- Efficient handling of floor requests
- Queue management: Separate queues for up/down requests

Scheduler


[arrived to destination]
[not arrived to destination]
Dequeue from queues
[idle]
[UP or DOWN]
[previous idle]
[previous UP or DOWN]
[queues are not empty]
[queues are empty]
Select new target floor
Select target in the same direction
Select any new target
Stay idle

Development Workflow

- **Version Control**
  - Git and GitHub for collaboration
  - Task management using GitHub Issues
  - Branching strategy for feature development
- **Continuous Integration**
  - `clang-format` and `cppcheck` for code quality
  - GitHub Actions for CI pipelines
  - Live documentation available

Documentation with doxygen

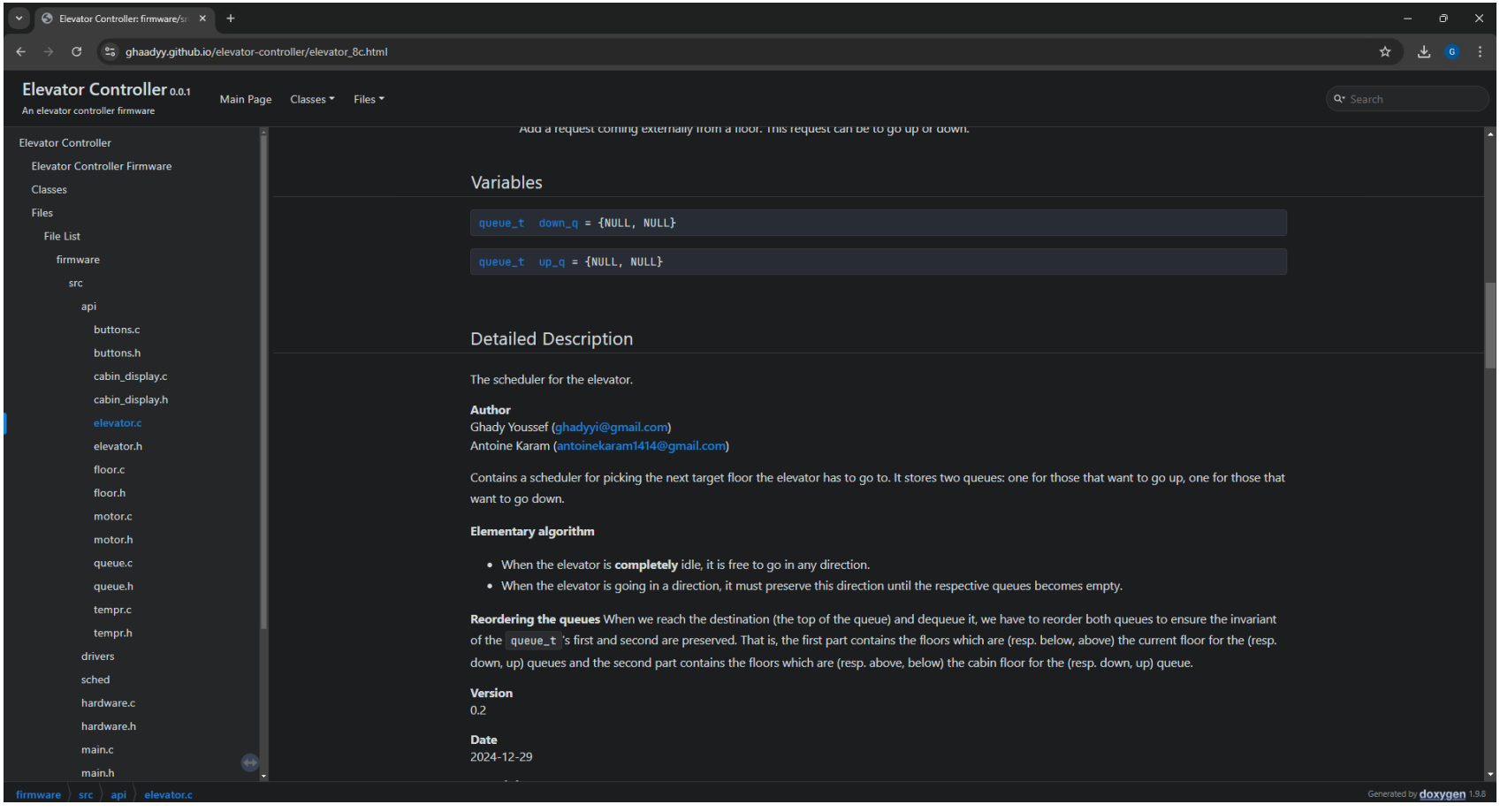


Figure: Available at https://ghaadyy.github.io/elevator-controller/

Challenges

- Designing efficient elevator scheduling algorithm
- Debugging logical errors

Future Work

- Error detection and handling for power failures and motor issues