System Analysis and Design Document

Project name: *Elevator project*

Author:

      Jerene, Gedewon G.D.
      Swinkels, Wouter
      Gramoun,Merna M.A.G.H
Document date/version: *24-05-24 V1.0*

Introduction

In this project, we are going to program the FHICT Bot with the goal of applying ES/SD learning outcomes related to C/C++ programming, low-level embedded control, timing, interrupts, and feedback-control systems. Optional IoT communication and creativity are encouraged to extend the project. The project will be structured in phases, and the final delivery will include comprehensive documentation, structured code, and performance demonstrations.

Components

- STM32 Nucleo Board
- 3D printed elevator
- ESP8266 (D1 mini)

Design

Phase 1: Control the servo

Requirements:

- FR1: Elevator can move to a desired position
- FR2: The desired position corresponds to the available elevator floors
- FR3: The desired floor can be controlled via a serial command

Design Approach:

- Develop initial control algorithms for moving the elevator to specific floor positions.
- Implement a simple serial command interface to input desired floor positions.

Phase 2: Smooth Elevator Operation with Feedback Control

Requirements:

- FR4: The elevator uses PID control to move smoothly from one floor to another.
- FR5: Elevator should not exceed a specified speed.
- FR6: Actual position and error are printed via the serial interface.

Design Approach:

- Integrate PID control for smooth floor transitions.
- Implement speed control mechanisms.

● Update the serial command interface to display real-time position and error metrics.

Implementation

Phase 2: Smooth Elevator Operation with Feedback Control

Steps:

● Implement the PID control algorithm for smooth motion
● Integrate speed control settings and test to ensure compliance with specified limits
● Update the serial interface to provide real-time feedback on position and error.

Optional functionality implemented

IOT Web service dashboard

Steps:

1. Setup ESP8
   a. Connect to wifi
   b. Connect to websocket hosted on the wifi via node-red
   c. Send received data from dashboard to Elevator(Nucelo) via serial command
2. Setup Nucelo
   a. Ensure Nucelo sends real-time data back to ESP based on the level of the elevator and speed
3. Setup Node-REd
   a. Create dashboard from the dashboard extension in node-red
   b. Create websocket server to listen on
   c. Connect the flow from the dashboard to the websocket
   d. Create functions to handle messages received and sent to/from the Nucelo to be compatible with the dashboards interface

Dashboard

elevator

LEVEL 2

LEVEL 3

LEVEL 4

SPEED
speed:44

ELEVATOR LEVEL

LEVEL 1

LEVEL 5