



# Project 6: Week 1

## Status Report:

[Engineering Project VI \(EECE73125\)](#)

### Project Status Report – Week #1 2022-05-16

**Project Name:** Unfortunate Series of Elevator Events

**Team Members:** Chris Le, Robyn Porter, Tori Le

**Project Overall Status:** GREEN

**Status Overview:** Completed all Tasks – Overall Group on same page and understands material and project.

Activity Title 05/16/2022	Status (Last Week) G/Y/R	Status (This Week)		Plan (Next Week)
		Plan (as planned last week for this week)	Actual Green/Yellow/Red	
View Project Documentation	N/A	COMPLETED BY ALL: 1. Project Information and First Briefing 2. Instructional Plan 3. Project Charter 4. Documentation for Raspberry Pi Server 5. Documentation for CAN Protocol 6. Review Rubric	GREEN	COMPLETED BY ALL: 1. Review Rubric
Test Plan	N/A	COMPLETED BY TORI: 1. Project Plan / Gantt Chart on Excel	GREEN	N/A
Communication	N/A	COMPLETED BY CHRIS AND ROBYN: 1. Set up Remote Connection to Raspberry Pi and IP Camera and provide evidence (tests/video to confirm)	GREEN	1. Confirm client/server communication via both HTTP and SSG with server (Raspberry Pi) - ROBYN 2. STM32 Programming and wiring - CHRIS 3. Demonstrate (video) ability to read messages on the CAN Bus - TORI
Economics & Project Management	N/A	COMPLETED BY ALL: 1. Complete Weekly Status Report 2. Complete Logbook Entry	GREEN	COMPLETED BY ALL: 1. Update Weekly Status Report 2. Update Logbook Entry
Design	N/A	COMPLETED BY ALL: 1. Design of control page GUI with controls and indicators	GREEN	COMPLETED BY ALL: 1. Update of control page GUI with controls and indicators

#### Guidelines:

- Overall Status: Green is the best, Red is the worst, and Yellow is in between. You may add + or – suffix to indicate the status more accurately.
- Planned completion date and name of responsible team member: indicate if these are unchanged or changed/new.
- You have to compare your status in three ways:
  1. Status of this week versus the status in last week
  2. Status of this week versus the plan for this week
  3. Planned completion date and responsible person (this week) versus the planned completion date and responsible person (last week)

# Project Plan

Phase	Task	Week												
		1	2	3	4	5	6	7	8	9	10	11	12	13
	1. Project plan													
	Complete and submit a Project Plan	ALL												
	2. Set up Raspberry Pi as Server and confirmed communication	Chris												
	Set up Raspberry Pi as a server	Chris												
	Confirm communication between client computer and server (HTTP/SSH)		Tori											
	3. STM32 Programming via HAL													
	Explain common CAN messaging protocol (see document)	Chris												
1	Implement the CAN messaging protocol by properly setting the filters (code demo) on STM32	Chris												
	Implement Wiring on STM32, push buttons/LED indicators (possible COVID restrictions)	Chris/Robyn												
	request, floor		Tori											
	request, door open/close, location of elevator car		Tori/Robyn											
	Develop and implement a test plan to confirm functionality of software/hardware													
	3. Controls / indicators (virtual instead of buttons - COVID restrictions)													
	Controls (up/down) built in HTML (possibly add CSS if students work ahead)	Tori												
	Elevator Car Controls (up/down/floor number) built in HTML (possibly CSS if students work ahead)	Chris												
	4. Finite State Machine (recommended)													
	Develop Finite State Machine Logic for Elevator Operation	Robyn												
	1. User Interface / Client Program													
	Fully working and completed front end Graphical User Interface /Webpage to control elevator	ALL												
	User action buttons/icons implemented		Tori											
	Indicators/lights etc. implemented	Robyn												
	User action buttons developed to make changes to UI that are reflected in the database													
	User indicator buttons change - changes in database are reflected in UI													
2	2. Database Repository													
	Database set up and accepting values/commands from website													
	Database has fields for logging information placed on the CAN bus													
	3. Diagnostic Reports													
	Produced based on content in database	ALL												
	Displayed on webpage		Robyn											
	4. Server Program													
	Users are authorized/authenticated based on username/password													
	Server program provides logging data/accepts commands only from authorized users													
	5. Announce floor numbers (Function of the Client or Server Program) for users with Visual Impairments													
	Implement a Server Program for announcing the floor numbers													
	Speaker on elevator used to announce the current/next floor (speaker already connected to system)													
	6. Sabbath mode (Function of the Client Program)													
	Users may operate the elevator without having to push buttons (e.g. verbally, based on schedule/time)													
3	1. CAN/LAN network communication													
	Research and implement a CAN/LAN communication bridge													
	Ensure activity in CAN network changes values in database													
	Changes in User Interface/Website changes value in database and generate CAN messages													
	CAN Messages result in changes to database and GUI/Indicators													
	Commands from CAN/LAN networks are sent to the controller and move the elevator / indicate floor number													
	2. Controller Software Written													
	CAN Supervisor (Raspberry Pi) relays messages between CAN/LAN Network (see sample program on Rpi)													
	3. Finite State Machine logic written													
	Develop and Implement a finite state machine (using C/C++, PHP, Python to implement logic)	ALL												
	4. Additional capability that shows ingenuity													
	5. Ideas for Capstone (at least 3 per student)	ALL												
	6. Final Presentation & Demonstration													ALL

## Elevator Gui Draft:

# Design of Elevator GUI (Controls/Indicators)

Project Name: Unfortunate Series of Elevator Events

Date: May 16, 2022

Requirements (from Project Charter):

Each call station (Floor controller) will consist of

- Buttons and LEDs (virtual/software/online also possible) to indicate the selection.
- A display panel, which will show the elevator car's position and a speaker will chime to indicate that an elevator has reached its destination (floor).

There will be another controller (Elevator Car controller – call button at bottom of elevator) for:

- Floor selection buttons (possibly virtual buttons) with built-in LEF for each button, and
- LEDs (possibly virtual indicators) that indicate the floor the elevator is on.

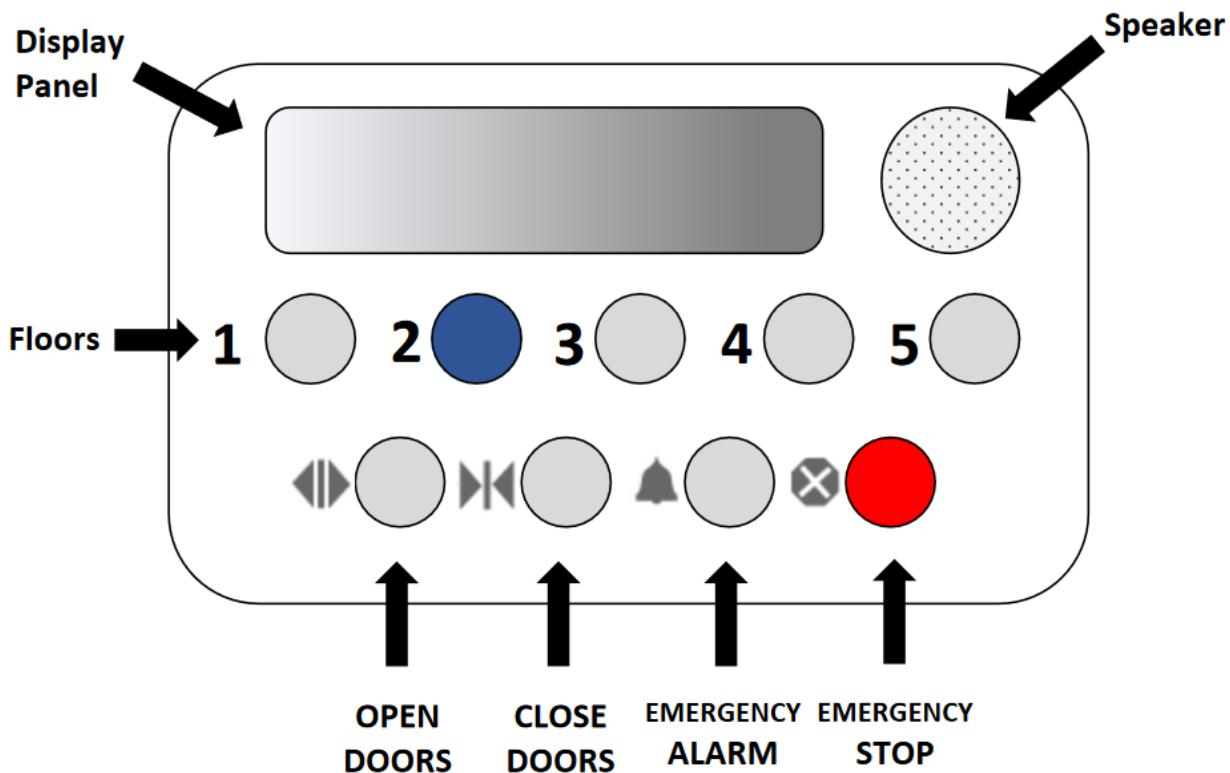
There will be another controller (Supervisory Controller – the Raspberry Pi) that:

- Stores the server, database and application for remote control of the elevator.
- Contains a speaker that may be used to announce the floor the elevator is on.

There will be another controller (Elevator Controller – HCS12) (already implemented for

- Sensing the height of the elevator.
- Controlling the height (floor) of the elevator.

This distance sensor is interfaced directly to the controller (HCS12) to provide a distance estimate, which can be displayed on an LCD or 7-segment LED display and ultimately used for positional feedback.



## Remote Camera Connection

