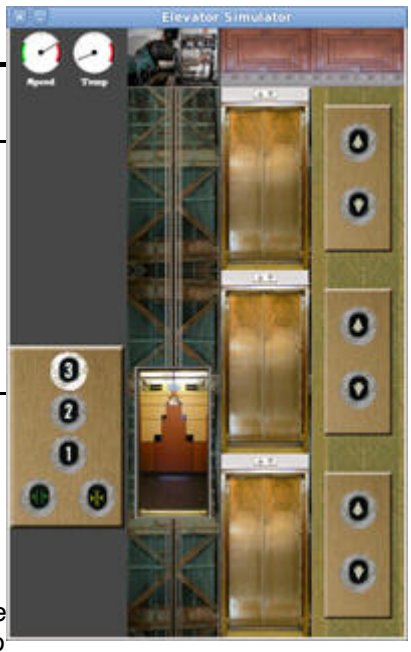
# Project 2: Elevator Simulator Part B




## Resources

Please refer to this section for resources to help complete the project.

- Project Main Page
- The ElevatorLib Library Documentation
- Project 2 Part B Help Page

## Part B Tasks

Make the elevator work. Here's how it should operate.

When not in use the elevator sits idle.

Button presses call the elevator or send it to a floor.

At any point in time the elevator should be either going up or going down. The function WhatFloorToGoTo should be called when the elevator is stopped to determine what floor to go to. If the value returned is -1, the elevator stays idle. The function WhatFloorToGoToInThisDirection should be called when the elevator is moving to decide what floor to stop at in that direction.

When the elevator stops on a floor the doors should open. Once open, they should stay open for 3 seconds, then close.

Don't move the elevator until the doors have completely closed.

You are *not required* to implement the door open and door close buttons on the elevator car panel.

When done, turn in only the file elevator.c via http://www.cse.msu.edu/handin

## Grading Criteria

- Working PID controller: 10%
- Uses state machine: 15%
- Not having IF statements for each floor, generalized solution: 5%
- Program works properly: 70% (broken down further, below)

The following are the steps I take to ensure that the program works properly, along with the percentage that will be lost if it doesn't work.

Note: Panel button is the button inside the elevator, call button is the button outside the elevator.

From start (Elevator stopped at first floor):

1. (10%) With the elevator on the first floor, push down call button on third floor, then immediately push down call button on second floor.
   **Expected result:** Elevator goes to third floor, then stops at second floor.
2. (10%) With the elevator at the second floor, push panel button for third floor, then first floor.
   **Expected result:** Elevator goes to third floor, then first floor.
3. (5%) With the elevator at the first floor, push panel button for third floor, then up call button on second floor when the elevator is past the safe stopping point for the second floor (i.e, less than one-half floor from the second floor). Once the elevator reaches the third floor, push the panel button for the first floor.
   **Expected result:** Elevator goes to third floor (not stopping at second floor), then first floor (again not stopping at second floor), then back up to second floor.
4. (5%) With the elevator on the first floor, push both up and down call lights on second floor and the down call light on third floor so all are lit at the same time.
   **Expected result:** Elevator goes to second floor, turns off the up call light, goes third floor, turns off the down call light, then down to second floor, turning off the down call light.
5. At all stops, the elevator must turn off appropriate call lights (10%), open and close (10%) the doors after a 3-second delay (10%).

## Hints and Suggestions

This part of the project will require a state machine. This will be covered in Lecture/Step 10.

You don't need any button handler functions for this functionality. The default button behavior is sufficient. In fact, you won't need to check any buttons at all, since WhatFloorToGoTo and WhatFloorToGoToInThisDirection do this all for you.

Consider writing a function for each state rather than putting all of the code in the main function.

Consider making your state and maybe your desired speed values global. A global variable is one outside of any function, usually at the beginning of the program. Global variables are often frowned upon in computer programming, but are fairly common in embedded systems.

CSE 251