

Design Overview of Elevator Simulation

Team Members and Contributions:

1. Karthik Raja – Interrupts and Main Logic of Elevator
2. Jacob Ferrer - Main Logic of Elevator, ASM to C Implementation
3. William Roper - Elevator Model, Wire Management, Stepper Motor
4. Zachary McPherson - Elevator Model, Wire Management, Interrupts, Keypad

Elevator Implementation:

Our elevator design allows the user to travel between multiple floors. Our design for the elevator uses various parts such as a keypad, seven segment display, two stepper motors, and five LEDs.

1. **Keypad:** The keypad acts as the grid of numbers inside the elevator, which allows the passengers to press which floor they want to go to. We implemented the keypad by using the keypad function that we implemented in Lab 04. In the main elevator function, we had the keypad function returning the integer that was pressed by the user. The reason we had the keypad returning which number was pressed was that we needed to know which LED to light up. We also allowed for multiple floors to be called while the doors of the elevator were open.
2. **Stepper Motors:** Two stepper motors were implemented in our design: one motor to control the elevator moving up and down and the other one to control the opening and closing of elevator doors. To implement this, we used the stepper motor function from Lab 05 and passed in a value in that function. If the value passed in was 1, then the stepper motor turns counterclockwise, and if the value was 0, then the stepper motor turns clockwise. The way we implemented the opposite direction was to do the same thing as the normal direction but the reverse order. Regarding the doors, we have implemented a delay that allows the passengers to get out before the doors close again.
3. **Seven Segment Display:** The seven segment display acts as the number display above the elevator, which displays the current level of the elevator. We implemented the keypad by using the keypad function that we implemented in Lab 03. In the main function, we had the current floor of the elevator being passed into the seven-segment function has an argument, which then displays that number. Whenever the elevator moves onto a different floor the seven segment display updates with the corresponding elevator floor number.
4. **LEDs: We implemented five LEDs in our design: four of which indicate elevator calls and one to detect when the admin button was pressed.** We implemented the LEDs to turn on using multiple external interrupts to help detect the button press. The LEDs both show which floor is

the destination floor and helped us implement the logic of the elevator depending on whether it was on or not. The way we implemented the admin interrupt was by setting its call to a higher priority than the floor interrupts. Then to keep the floor interrupts disabled during the admin process, each floor interrupt handler would check if the admin led was enabled. If so, the floor interrupt flag would be cleared.

5. **Logic**: We were able to implement an infinite while loop that checks our logic for the elevator infinitely. The first thing our logic does is check if the admin button is pressed down or not. If it is pressed down, then the elevator moves down to floor 1 and turns off all the elevator call LEDs. Furthermore, the logic checks if any of the elevator calls are on. If any of the elevator call LEDs are on, then the code moves on to a switch case that assigns which floor to check first and which to check last depending on the current floor. Finally, we go into another loop that checks if the current floor is greater or less than the destination floor. We implemented if-else statements corresponding to if the elevator should go up or down. Also, if the current floor is on a floor that has its LED on, then the doors open. But, if the admin LED is on, then none of these logics would occur.
6. **Elevator Model**: We were able to design, and 3-D build an elevator to help us demonstrate our simulation. This model helps the audience visualize how the elevator operates in real life. There are also 3-D printed doors that open and close which also helps us visualize the simulation.

Overall, our implementation of the elevator required many external parts, but by implementing them in an organized manner, we were able to create a working elevator model.

Design Constraints:

During our time working on this project, we encountered several design constraints that informed and shaped our decisions. Several of these are listed below.

1. Number of Usable Pins on the Microcontroller

The STM Nucleo-64 microcontroller we used has 4 GPIOs with 16 pins each, totaling 64 GPIO pins. The peripherals we used in our project had to have pins less than 64. Even then, some of the pins were unusable, such as PA2 and PA3, which required us to work around these reserved pins.

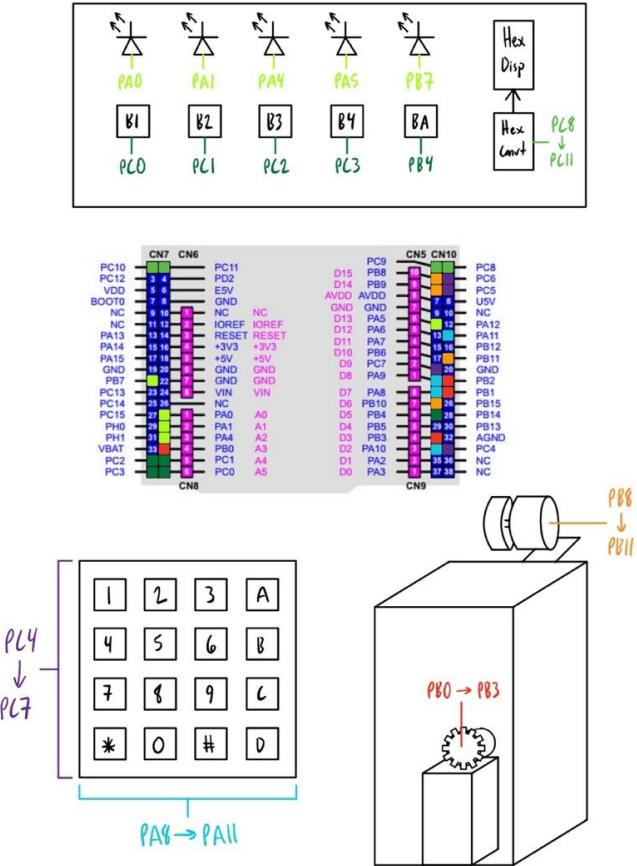
2. Use of ARM Assembly and C Languages

The specification of combining ARM Assembly with C had its own unique challenges. We decided to write the bulk of the high-level logic in a C file, while the low-level peripheral initialization and management was written in assembly. To combine these two, the EXPORT name of the assembly function could be called in C with any necessary parameters.

3. Floor Calls Need to be Received at Any Time

This design constraint of the elevator logic required us to implement external interrupts. We could not use polling since there could be instances where the floor call would not be received if another function (like the stepper motor turning) was currently running. We were able to modify the code found in chapter 8 of the textbook to implement interrupts and handlers for each floor call.

Schematic:



Successes and Failures Encountered:

Successes:

1. We were able to successfully use multiple external interrupts to detect the elevator button press and light up the corresponding LED. As well as using an admin interrupt to disable the other external interrupts.
2. We were able to design and print a small-scale elevator that helps us show the audience how an actual elevator would run with our code.

Failures:

1. We were unable to successfully display the elevator's status using the external terminal. We got all the messages to display successfully except for 'Moving Down.'