Introduction to Robotics: Homework

Andrei Dumitriu andrei.dumitriu@fmi.unibuc.ro

Last Updated: October 23th, 2023, 20:18

1 Homework #3 - Elevator simulator wannabe

Deadline (hard): Your respective lab in the week of October 30th-November 5th, 2023.

1.1 Description

This assignment involves simulating a 3-floor elevator control system using LEDs, buttons, and a buzzer with Arduino. By the end of this task, you will gain experience in using button state change, implementing debouncing techniques, and coordinating multiple components to represent real-world scenarios.

You must finish the work before your laboratory session and present it there. This includes everything: hardware, coding and github documentation.

Example: `https://www.youtube.com/watch?v=7BQCW9OsQV0`. Big thanks to Adrian Pascu for doing it on such a short notice.

1.2 Components

- LEDs (At least 4: 3 for the floors and 1 for the elevator's operational state)
- Buttons (At least 3 for floor calls)
- Buzzer (1) - optional for Computer Science, mandatory for CTI
- Resistors and wires as needed

1.3 Technical Task

Design a control system that simulates a 3-floor elevator using the Arduino platform. Here are the specific requirements:

- **LED Indicators:** Each of the 3 LEDs should represent one of the 3 floors. The LED corresponding to the current floor should light up. Additionally, another LED should represent the elevator's operational state. It should blink when the elevator is moving and remain static when stationary.

- **Buttons:** Implement 3 buttons that represent the call buttons from the 3 floors. When pressed, the elevator should simulate movement towards the floor after a short interval (2-3 seconds).
- **Buzzer (optional for Computer Science, mandatory for CTI):** The buzzer should sound briefly during the following scenarios:
  - Elevator arriving at the desired floor (something resembling a "cling").
  - Elevator doors closing and movement (**pro tip**: split them into 2 different sounds)
- **State Change & Timers:** If the elevator is already at the desired floor, pressing the button for that floor should have no effect. Otherwise, after a button press, the elevator should "wait for the doors to close" and then "move" to the corresponding floor. If the elevator is in movement, it should either do nothing or it should stack its decision (get to the first programmed floor, open the doors, wait, close them and then go to the next desired floor).
- **Debounce:** Remember to implement debounce for the buttons to avoid unintentional repeated button presses.

1.4 Publishing Task

1. Add the code to the Git repository.
2. Update the repository's README with:
   - Task Requirements and/or Description
   - A photo of your setup
   - A link to a video showcasing functionality (YouTube is recommended, but any accessible platform is acceptable). Ensure the video is in the correct orientation.
3. Once your Git repository is up to date, submit your homework on MS Teams.

1.5 Coding Task

- Maintain clean and structured code. Following best practices is essential.
- Replace "magic numbers" with constants that have meaningful names.
- Refer to the style guide and any provided documentation to ensure consistency and clarity in your code.

1.6 Common Mistakes to Avoid

1. Using delay() instead of millis().
2. Using non-descriptive variable names such as **btn1**, **led1**. Opt for more descriptive names, such as **floor1Button**, **floor1LED** (or maybe an use an array).
3. Ignoring button debounce which may result in unintended elevator movement or multiple buzzer sounds.
4. Forgetting to set the elevator's operational LED to blink when the elevator is moving.
5. Overlooking the buzzer feedback for specific events such as doors closing or elevator movement.

1.7 Possible bonus points

1. Getting creative with the physical design.
2. Adding a "decision-stack". When using a normal elevator, you usually press multiple numbers and the elevator takes you there in order. You can implement that here as a bonus.
3. Buzzers for Computer Science students (CTI has them part of mandatory requirements).