Lab Final Project – Elevator simulation

Objective:

Use the previous knowledge gathered during the lab experiments and theoretical classes to simulate the solution of a real problem, implementing it on the Dragon12 plus board in the lab.

Project Rules

- The lab project counts for **45% of lab grade**.
- This project is to be done in a group of two students, during the start of this lab session each group is supposed to submit the names of their group members.
- Performance of groups in the project will be only their own, students are not allowed to copy the code and strategies used by other groups.
- Students can ask the TA for help but the help provided will be very limited, as this project is the final evaluation for this lab. If you ask questions that you should already know, the TA may mark down points from the final grade of the project – so, first try looking up in your notes, books, manuals or references.
- Students can download the Code Warrior on their personal computers and work at home or off campus. They can test their codes on the Dragon EVB during the regular lab sessions.
- If any students want to use the lab facility other than lab sessions, he/she can contact the TA during office hours.
- Projects will be graded during the last lab session.
- **10 %** credit is for a **project report** which will also be collected on the final lab day, TA will explain the format for the lab report during the lab session.
- The report has to be well documented, with flowcharts, meaningful comments of your code, and anything else that the TA may require.
- Remember Google is your best friend and Book is the best resource. Check the Motorola/Freescale/NXP manuals on e-learning and on the internet.

Part1:

You are to implement the control of an elevator of a 4 floor building, (1 to 4).

The **inputs** are going be:

- "Internal" Keypad to choose the floor to go to (1 to 4), and 2 keypad buttons to open or close the door;
- 4 "external;" Push Buttons for calling and opening the door at each floor.

For simplification, these buttons only open the door if the elevator is already stopped in the floor.
If it is not there, but stopped somewhere else, then it serves to call the elevator to that floor (a floor LED will indicate that it is not moving). If it is moving, the buttons are not active.

The **outputs** are:

- LCD for the messages interfacing to the user;
- Buzzer (speaker) to indicate the arrival at the chosen floor with a 2 second 1 KHz tone;
- 2 "internal" LEDs to indicate if it is going up or down (stopped both are off);
- 4 "external" LEDs to indicate if the elevator is stopped and available to be called;
- 2 "internal" LEDs that simulate the motors to open and close the door (LED1=1 open, LE2=1close, never activate both simultaneously).

Note: the elevator can only be called when stopped. Once you choose a floor, it only stops on the target floor, not intermediate ones.

Actions:

You start by pressing the open button outside the first floor. The door opens, a message "door opening" shows on the LCD and you get in. Every time you close or open the door, turn on the corresponding LED, and display the corresponding message on the second line ("door opening" or "door closing"), wait 1 sec to simulate the time a real door would take to open or close and then change back to the previous message ("current floor #").

If you don't press anything, the door closes in 2 seconds.

If before the 2s you press the Close Button, it should close the door and stop and wait for you to press on the keypad which floor to go to **or** the open button to get out.

If before the 2s you press a floor number on the keypad, it waits 2s and closes the door, then start moving.

If you are in a hurry, after you press the floor key you can press the Close button before the 2 seconds, then it "closes" the door immediately.

To simulate the elevator moving, use a 1second delay for each floor, and display in the LCD the target floor on the first line ("going to floor #") and on the second line the current floor ("current floor #") which keeps changing after each second, simulating the movement. While moving, the 4 external LEDs (1 for each floor) should remain off to indicate that the elevator cannot be called.

When it arrives at the target floor, open the door and play the speaker/buzzer for 2 sec and then close it again, turning all the external LEDs on to indicate that the elevator is available to be called.

Notes:

1. The delays don't need to be exact, it does not need to be done with the timer and interrupts. For example a 1 s could 1.05s or 0.99s.
2. You will have to look up the Dragon12 plus manual for the addresses of the speaker, push buttons, keypad and LCD (or in previous labs where you used some of them).
3. **First do a flowchart** of the actions to be taken, then start coding, putting **meaningful comments** to permit you to understand what is going on.
4. Also try to use **Equates** and **subroutines/macros** to make the code more readable, otherwise you will have a very hard time. If possible, use look up tables where possible, to simplify your code.
For example, create a delay subroutine (or macro) for 1 sec, passing the amount of seconds via a parameter on register or memory location. Create subroutines (or macros) to open and close door, to turn off/on all external LEDs, etc… and then in the main program you just refer (call) to them, making it more readable.
5. Use a **flag** (register or memory position) to indicate the current floor.

HINT: when you want to check several conditions, an easy way is to use "**flags**" to represent the condition. For example a memory position or a register can be used to represent the states "stopped", "going up" or going down". Then you can check for this anywhere in you code, even within subroutines. A flag can be binary (true or false, 1 or 0) or use more than one bit (n bits) to represent 2^n states.