CS466 Homework/Lab 7 (Project) Assigned Wed, March 3, 2014
Due in class or via Angel at the end of the semester (the night of the final). This is worth 15% of your course grade.

Three Story Elevators:

This project is intended to be fun with an idea towards learning. It consists of several steps and should take much of the rest of the semester to complete. All of the steps are to be completed in the lab using your Rabbit microprocessor target hardware and using MicroC/OS-II. Generally, the steps will have to be done sequentially but you may skip around where it makes sense.

We may have some smaller labs in the interim.

You should already have code which reliably moves a stepper either forward or backward and should also possess code which will move multiple motors simultaneously (or are close to it). This lab will re-use most of your Lab6 software and use the same electronics.

Our project will be the control of a "simple" elevator. One or two students will control each half of the elevator. We do not have enough elevators for the entire class so be mindful of when you can work with a 'testbed' (e.g. a motor on the table) or when you need the full apparatus.

Requirements:

Metrics and Required Modes of Operation:

1) **(Mode 1)** The team must be able to run the elevator up and down at least 3 inches continuously for 2 minutes and return to precisely the same location. (This verifies that steps are not being gained or lost).
    1. Each team will be forced to use the same motor voltage to make the playing field as even as possible.
2) **(Mode 2)** The team must complete a time test to run the elevator up and down 10 times. The team who does it the fastest gets 10 extra points. Second place receives 5 extra points. Third and lower receive 0 extra points.
    1. While performing Mode 2, the list and number of prime numbers discovered must be indicated. This should be performed by writing numbers to a buffer and reporting after the 20 moves are complete.
    2. While doing Mode 2, the elevator must stop for one half second at each floor (Top, Middle, and Bottom).
    3. Each team will be forced to use the same motor voltage to make the playing field as even as possible.
    4. Mode 2 will initiate on a button press and report the time in seconds that it took to complete after providing the list and count of primes.
3) **(Mode 3)** Using LED's and buttons (discrete or off the demo board) implement a realistic three floor external elevator operator interface.

1. Ground Floor has only an ‘up’ button, middle floor has ‘up and ‘down’ buttons and the top floor has only a ‘down’ button
2. Led’s indicate if the button has been pressed as in a real elevator UI.
3. Mode 4 timing should be much slower than others to make sense of the UI and lights.
4. The Instructor will provide timing and acceleration data from an actual elevator and scaling information to map to our apparatus. Points will be given for the closest scaled match to acceleration and timing data acquired from your elevator. (sensor under construction)
5. Since we are not implementing the interior Elevator UI, Make some assumptions about operation (e.g. all ‘up’ requests go to the top floor and all ‘down’ requests go to the bottom.)
6. This step will require use of the SPI GPIO expander chips. I may provide the SPI interface library, similar to the Queue code.

It is clear that some sort of interface/protocol must be invented to communicate between the Rabbit boards. The communications should be bidirectional allowing either board to run the elevator system. The inbound command data should cause an interrupt on the receiving board so that polling is not necessary. Interface possibilities include.

- parallel I/O (single bits) implemented with GPIO pins
- RS232 communications
- Ethernet:
- Your Idea Here? (Smoke Signals?)

CAUTION: Make sure your circuit with the ULN motor driver is correctly isolating the Rabbit from the motor windings. The Rabbit board will not tolerate the voltage necessary to power the motor.

Project Requirements:

1) Each elevator will consist of a simple frame with 4 stepper motors mounted on the frame.
2) Each motor shaft should have a pulley mounted on it.
3) Each motor will lift 1 corner of the elevator.
4) One team will control two motors with one Rabbit while the other team will control the other two elevator motors with their Rabbit.
5) Both teams must be able to control the entire system (but only one at a time).
6) You are to do this simultaneously with a cpu intensive task of discovering Prime numbers.
7) The controlling system will gain extra points for using the LED’s and Buttons on the demo board as an Elevator UI
8) Verify that each team member can drive two motors forward and backward as many steps as you desire from your program indefinitely without losing or gaining steps.

9) Determine what sort of architecture/interface between the micros and generate code to support whatever choice is made
10) Come up with a software design
11) Verify that all four motors can be driven in synchronous motion
12) When each team can do a coordinated move, it is time then to hook up the elevators.
13) Each team must spend some time determining the fastest move probably in an empirical way.
14) All of this must be written up in a laboratory experiment which states what you did and what happened and presented by the due date. The instructor gets to decide if you have fulfilled each of the requirements. The above metrics will be conducted at this time.
15) A brief (5 minute) presentation will be given by your team to the class near the end of the semester.
16) Any "improvements" made to go faster and win the contest should be fully explained.
17) All teams must use the same motor voltage. (motor voltage tbd but probably in the range of 10-15 volts).
18) All teams must use the same motor driving electronics. There are lots of improvements that could be made over our drivers but they are beyond the scope of this class.
19) One other thing is to consider how to initialize the position of the elevator at power up of the micro. How could we do that? You must answer this question of how/what to do but do not necessarily have to implement same.
20) All source code must be included. Each team should have source code.
21) The write up should be well written, addressing all of the above points and using the standard lab format.
22) Lab will be due at the beginning of the final exam (I will allow a grace period to midnight that evening if submitted on Angel)