ECE 376 - Homework #5

Keypads, Stepper Motors, NeoPixels

note: You can combine the two programs into one which uses a keypad, a stepper motor, and a neopixel.

Keypads & Stepper Motors

Design an embedded system which uses the numeric keypad and the stepper motor. Some suggestions are...

- Roulette Wheel: Bet on a winning number with numbers (0..7). Press RB0. The stepper motor then spins two rotations at 10ms / step and then stops at the winning number (N*25 steps)
- Count-Down Timer: Input how long you want to wait from X = 000 to 999 x 100ms. The stepper motor then turns X steps to start. It then steps back to 000 steps at a rate of 10ms / step and stops when time is up.
- Stepper Motor Position Control: Input a number with the keypad from (000 to 999) then hit RB0. The stepper motor then turns to that angle (in steps) at a rate of 10ms / step.
- Other...

1) Requirements: Specify the inputs / outputs / how they relate.
2) C code, flow chart, and resulting number of lines of assembler
3) Validation: Collect data in lab to verify you met the requirements.
4) Demo. Video or in person.

Keypads & NeoPixels

Design an embedded system which uses the keypad and a NeoPixel. Some suggestions are...

- Alarm Clock: Input a number from X = 000 to 999 x 100ms (00.0 to 99.9 seconds) using the keypad. When you press RB0, the time decrements to 00.0 every 100ms. Set the color of the NeoPixel according to how much time is left.
- Alarm Clock (take 2): Input a number from 000 to 255 seconds using the keypad. When you press RB0, the time decrements every 1.00 second. Display the time left on the NeoPixel as a binary number 000 to 255.
- Roulette Wheel: Bet on a winning number with the keypad (0..7). Press RB0. The lights on the NeoPixel then spins and stop at the winning number (0..7). If you win, your bank increases by $8. If you lose, it decreases by $1.
- Combination Lock: Input the correct password (0000 to 9999) using the keypad and press RB0. If input correctly, the NeoPixel flashes on for one second then off.
- Other...

5) Requirements: Specify the inputs / outputs / how they relate.
6) C code, flow chart, and resulting number of lines of assembler
7) Validation: Collect data in lab to verify you met the requirements.
8) Demo. Video or in person.