CEEN - 1060 Microprocessor Applications Spring '07

Lab 9

Report Due: week of 4/23/07

Objective:

1) To learn the input/output port commands used for input of TekBot™ ATmega32 board DIP switch settings and output of patterns to LEDs.
2) To understand the principles of concurrent control and timing delays.

Procedure:

Output the following header to the LCD display:

```
*      NAME     *
*   CEEN 1060   *
```

Write a program to first display the header on the LCD screen. When the user hits either "bump" button your program will monitor the DIP switches of the ATmega32 board and light the appropriate LEDs for the following modes.

1) If all switches are "OFF" all LEDs will be off.

2) If a single switch is "ON" the associated LED will be lit continuously. For example: if switch number 5 is on, LED number 5 will light and remain lit until switch 5 is turned off.

3) If two switches are "ON" at the same time, the appropriate LEDs will light *alternately*. For example: if switches number 2 and 7 are on, LED 2 will light (LED 7 off), then LED 7 will light (LED 2 off), LED 2 will light (LED 7 off), then LED 7 will light (LED 2 off) …etc. Because of the speed of the ATmega32, you will have to delay each transition long enough to be perceptible; at least a half second would be appropriate.

4) If three or more switches are on, all eight LEDs will light and remain on.

These four functions are *concurrent*. This means if your program is operating in any one of the 4 modes the switches must be monitored and the mode *immediately* changed when the switches are changed. If all switches are off when the user first hits a "bump" button, your program must pass through modes 2 and 3 in order to get to mode 4. If your control program is in any mode and the switches change, the mode will change appropriately.

Your program will terminate when the user hits either "bump" button the second time and the normal "bump" TekBot™ behavior will continue until the ATmega32 is powered down or reset.

Report:

1. Listing of your program
2. Make sure the code that you submit is **well** documented. Code without comments will not be accepted.
3. Include a flow chart of your program including initialization, functions, and interrupt service routines.
4. Include a hierarchical chart of your program which clearly defines the calling structure, parameter passing, and organization of your program.
5. Your report text will discuss your efforts and your deliverables. Your results and conclusions will discuss problems encountered and lessons learned.

Demonstrate your program to your lab instructor and submit this page with your report.

Lab Instructor's Initials: ______________________ Date:________