EE260 – Lab 5

The objective of this lab is to learn how to use interrupts and pseudo instructions, and how to drive LCD displays.

Pre-Lab:
Read and understand the Lcd1.asm and Ex2.asm examples provided with the Wytec software. Find out the address of the SWI vector, and the addresses of port A, port B, Port C, and the DDRC register.

Lab:
   a. Write an assembly program that simulates the operation of an ATM machine.
   b. The PIN must be stored at address $E0004 of memory by using the fcb pseudo instruction. The max number of attempts allowed must be stored starting at address $E000 of memory using the fdb pseudo instruction. The number of attempts must be stored in a 2 bytes variable called cnt, and it must be defined using the rmb pseudo instruction.
   c. The user types in the PIN through the port C switches and uses the PA0 button to enter the value.
   d. Every time the PA0 button is pushed a software interrupt (SWI) is generated. The handler routine for SWI keeps track of the number of times that the PA0 button has been pushed.
   e. Incorrect PINs are displayed on portB’s LEDs.
   f. If the user does not enter the right PIN within 10 attempts the system is blocked, the LEDs are flashed once and the message “TOO MANY ATTEMPTS” is shown on the LCD display. Use the evplus2 I/O subroutines to drive the LCD display.
   g. If the user enters the right PIN the LED are flashed once and the message “TAKE THE MONEY!!” is shown on the LCD display. Use the evplus2 I/O subroutines to drive the LCD display.
   h. Write a report documenting your design. The report should have the following sections:
      - Objectives/Overview
      - Background Information
      - Design Description
      - Testing Strategy
      - Lab. Procedures
      - Conclusions
      - Appendix (attach your code, and the symbol table)

Hint:
If you push the PA0 button for too long the CPU will “think” that the button has been pushed several times. This problem is known as bouncing!! For the moment in order to work around this problem, 1) push the button gently, and 2) introduce a delay of 100-200 ms before consecutive checks of the button status.