Lab 8: System Integration rev. 1.0

This lab integrates the knowledge you have obtained throughout the semester. When you finish, you will have your working clock! This lab has one section and one demonstration. You will need to email your zipped code to grw1@msstate.edu BEFORE the start of lab, on November 19, 2001.

Pre lab

none

I/O pin utilization

pin	*Port A*	*Port B*	*Port C*
0	I2CM_SCL	RE_A	LED_DATA
1	I2CM_SDA	RE_B	LCD_RS
2	SOLENOID	PB1 (temp select)	LCD_RW
3		PB2 (pot select)	LCD_E
4	-n/a-	PB3 (time select)	LCD_D4
5	-n/a-	PB4 (solenoid select)	LCD_D5
6	-n/a-	LED_CLK	LCD_D6
7	-n/a-	LED_LATCH	LCD_D7

Main (FG thread) program

The FG thread contains the same four state s/w FSM as the previous lab. With the exception of the solenoid software changes below, all other functions (from the previous lab) must continue working without error.

ISR Modifications

To finish your project, you need to do two things. The first is to divide your LED "screen" into character positions. As the pendulum oscillates, your software will "wait" for the LEDs to arrive at each character position. Then, your software will send the appropriate "character" to the LED array by lighting each character's scan line in a timely fashion. The width of each character is controlled using through delay following each scan line.

Modify your solenoid code in the following way:

- Decrement solCounterH. If zero, then jump to solReset to toggle SOLENOID.
- At the label solReset, toggle SOLENOID and divide the solenoid "half-period" (solCounterH) by 32. Place this value into a variable solStep. Also, a variable solNext should be set equal to solCounterH-solStep and a variable solPos is set to 31.
- If solCounterH is not zero after decrementing, compare solCounterH with solNext. When solCounterH is less than or equal to solNext, then
    1. "new" solNext value is set to the "old" solNext value minus solStep,
    2. solPos is decremented, and
    3. compare solPos with SOLENOID_PRINT_BEGIN and SOLENOID_PRINT_END. If solPos is between these two numbers, then set the ledCanPrint flag, else clear ledCanPrint.

To summarize, the new solenoid software variables are

solStep	number of solenoid counter "ticks" between each character position
solNext	solenoid tick number where the next character position begins
solPos	number of current character position

Now you have your character positions. You need to simply write the clock time characters into these character positions. Modify the solenoid ISR to do the following (after the above code has finished executing):

- If ledCanPrint is clear, then jump to solExit
- If ledCanPrint is set and ledIsBusy is set, then jump to solExit.
- If ledCanPrint is set and ledIsBusy is clear, then load the appropriate clock digit into ledData, set ledIsBusy, and clear ledCanPrint. To determine the clock digit, use the following flags (located LED bank variable ledFlags that already contains ledCanPrint and ledIsBusy): ledOnHours, ledOnMins, ledOnSecs, ledOnTens, ledOnOnes, and ledOnColon. Create a state machine that uses these flags to determine which value from clkHours, clkMins, and clkSecs to copy to ledData. (NOTE: This code is located in the solenoid ISR, so make it very efficient!)

You will need to create lookup table entries for the LED scanlines to print the numbers 0-9, and ":" (colon).

Functional requirements

Your pendulum LED array must print the current time (with "changing" seconds). You can print the time during either "swipe". (Printing during both swipes is coming soon, so start thinking about it.)

Adjust the values SOLENOID_PRINT_BEGIN and SOLENOID_PRINT_END to center your time display in mid-air. (These values vary from base to base depending on the physical properties of the hacksaw blade, LED array mass, epoxy stiffness, etc.)

You must demonstrate your program. Failure to write the program with the proper techniques will result in no points for that section. Late submissions will be penalized 10 points. A submission is considered late if the code used for the demo is not in my email inbox by 6:00pm, November 19, 2001.

Grading will be as follows:

Strict adherence to coding conventions	30 Points
Proper time display	50 Points
LCD Display/Buttons	10 Points
Proper submission	10 Points
AM/PM time display on LEDs	5 Points BONUS