Computer Science 230
Assignment 2
Due before 11:55pm on October 26th 2018

Objectives

- More practice writing assembly language programs
- Introduction to peripherals: LEDs and buttons
- Introduction to subroutines

Academic Integrity

Prior to submitting your assignment, you should familiarize yourself with the University policy on Academic Integrity

We will use a plagiarism detection tool on all assignment submissions.

AVR Studio Project

As in assignment 1, you should create a new assembly language project in AVR Studio for this assignment. Download the sample code and add the files to the project.

Subroutines

In the sample code, subroutines use specific registers to pass parameters and return values. Registers are effectively global variables, so register usage must be consistent across every subroutine. The register usage is documented in the comments.

Rather than tracking register usage by hand, you may wish to update the subroutines to use the stack to save and restore registers. If you do this, be sure to initialize the stack pointer and update the comments appropriately regarding register usage.

Part I – Lighting the LEDs

As you've seen in the lab, the boards include a 6 LED strip, where each LED can be controlled independently from the others.

Although the LEDs are wired to adjacent Arduino pins, they are controlled using two AVR ports: PORTB and PORTL. The exact mapping is shown in the table below:

Pin	PORT	Bit #
42	L	7
44	L	5
46	L	3
48	L	1
50	B	3
52	B	1

You should modify the code in `a2q1.asm` so that it displays the value in R0 in binary using the 6 LEDs. You should consider the LED on Pin 42 to be the least-significant bit and the LED on Pin 52 to be the most-significant bit.

For example, if R0 contains the value 0x13, the LEDs should look like:

Pin	52	50	48	46	44	42
Value	OFF	ON	OFF	OFF	ON	ON

Be sure to test your code on several different values.

Your instructor's solution used a combination of `ANDI` and `BRNE` instructions to test the individual bits in `R0` and then used `ORI` to set individual bits in two output registers: one that is output to PORTL and another that is output to PORTB.

Be sure you understand the assignment 2 exercise posted with the assignment before you start programming. It will be the in class exercise on October 12, 2018.

Part II – Subroutines and timing

Using the code you wrote in Part I and the delay code provided to you in Lab 4, write a program that does the following:

```
counter = 0
start:
    display counter in binary on the LEDs
    counter = counter + 1
    delay between 100 and 300 milliseconds
    goto start
```

First you should turn the code you wrote in Part I into a subroutine called `display`. Then write a main program that implements the algorithm shown above. It will be easiest if you use `r0` for your counter.

Part III – Button press counting

Using the button code and delay code provided to you in Lab 4, write a program that does the following:

```
    counter = 0
start:
    if button pressed
        counter = counter + 1
        delay between 100 and 300 milliseconds
    display counter in binary on the LEDs
    goto start
```

Part IV – Improved button code

Improve the button code so that it correctly determines which button has been pressed. Study the comments in `a2q4.asm` for details on the values returned from the analog to digital conversion when the buttons are pressed.

You've been provided with a main program – you'll need to fix the button subroutine and include your LED display subroutine. Once you do that, a different light should come on for each button you press.

Your instructor's solution used the `BRSH` instruction.

Submission

Submit your `a2q1.asm`, `a2q2.asm`, `a2q3.asm` and `a2q4.asm` using connex. Do NOT submit your project file – just the .asm files.

Grading

If you submit a program that does not assemble you will receive 0 for that part of the assignment.

Question 1	4 marks
Question 2	4 marks
Question 3	4 marks
Question 4	2 marks

Total 14 marks