# Lab 1: AVR Toolchain Review

*UCR CS 122A*

This lab reviews the use of concurrent synchSMs and the AVR toolchain.
**Before you start developing**, you need to create a Github account for yourself if you do not already have one. The files that you create on the lab machines are only temporary. You will need to push the changes to github to save your work.

## Pre-lab

If you have not worked with the new Linux AVR Toolchain on the lab machines, you will want to review Lab 2 and Lab 5 from CS120B.
You will want to refresh your memory on working with and programming an Atmega1284. We will be working directly with the hardware in the first lab, so make sure you have your parts kits and are ready to work. You may want to prepare your own laptops and test to make sure you can program the Atmega1284.

- Windows
    - Atmel studio
- Mac and Linux (Lab machines)
    - AVR-GCC
    - AVR-GDB
    - AVRDude (AVRDudess)
    - SimAVR
    - UCR CS120B AVRTools

Getting started:

- Review the course syllabus and subscribe to the course online textbook (it should be free)
- Review the "Recommended CS122A Workflow"
- Review the relevant header files (timer.h, lcd.h, scheduler.h)

## Exercises

1. Create a system to blink three LEDs on PORTB[2:0] in order (B0, B1, B2, B0 … etc.) with the LEDs changing every 500 ms. Additionally, have an LED on B3 that blinks on and off every 1000 ms. You will also have a button that will pause the lights blinking (all of them) when it is toggled. Each independent press should toggle the LED sequences on then off, etc.

```
$ cp source/main.c turnin/[cslogin]_lab1_part1.c
$ git commit -m "Completed part 1"
```

2. Extend the problem from above to create a game. Everytime you press the button when the middle LED (B1) is lit up your score is incremented by 1. Everytime you miss, your score is decremented by 1. Your score cannot go below 0. Your score should be outpt on the LCD screen.

```
$ cp source/main.c turnin/[cslogin]_lab1_part2.c
$ git commit -m "Completed part 2"
```

# Submission

Each student must submit their source files (`.c`, `.h`) and test files (`.gdb`) according to instructions in the lab submission guidelines.

```
$ tar -czvf [cslogin]_lab1.tgz turnin/
```

**Don't forget to commit and push to Github before you logout!**