

Lab 0

Overview

- Learn how to use the command line to navigate and manipulate files as well as compile and run your embedded programs
- Install the Arduino compiler and other software
 - We have the software on some USB drives you can grab if the network connection is slow
- Test your connection by modifying a pre-defined program and flashing it to the Arduino to see if it executes

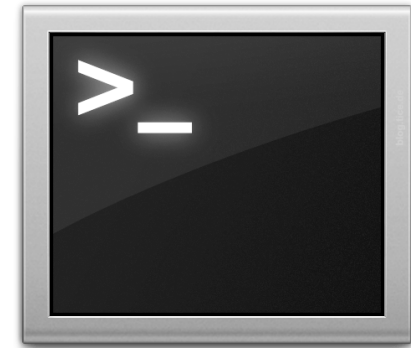
Using the Command line

- Startup Windows Command Prompt



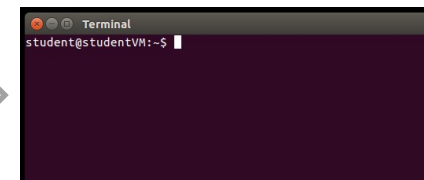
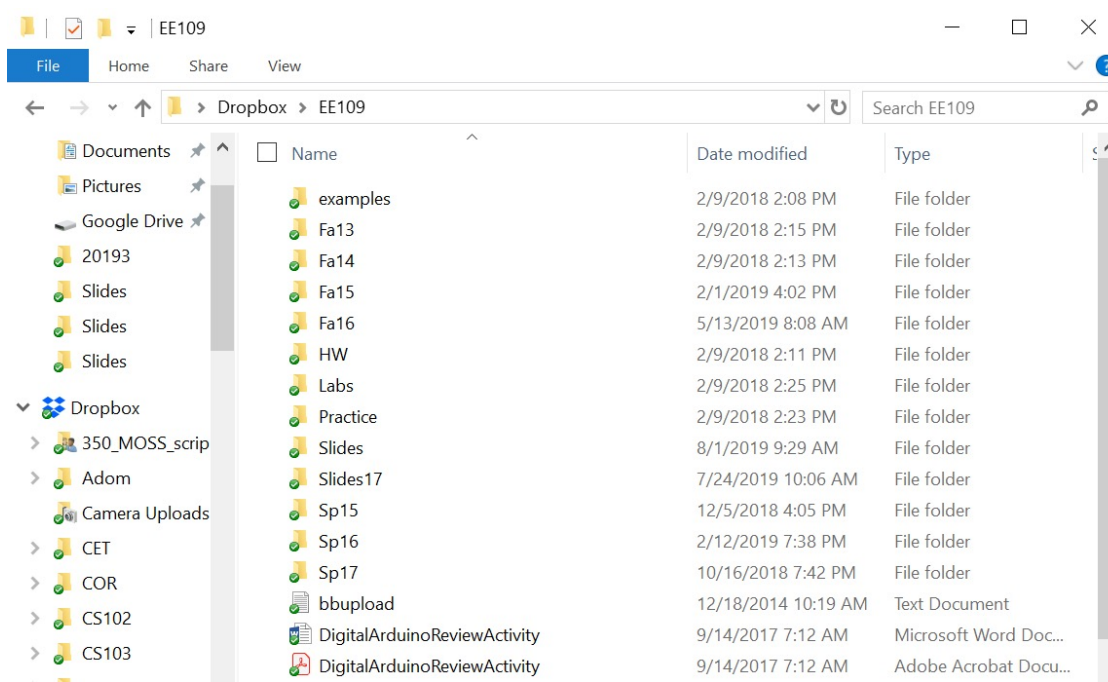
OR

- Startup Mac Terminal



Navigating the File System

- A file system has
 - Folders (directories)
 - Files
- They are organized in a hierarchy
- Everything we can do with a GUI we can do at the command line



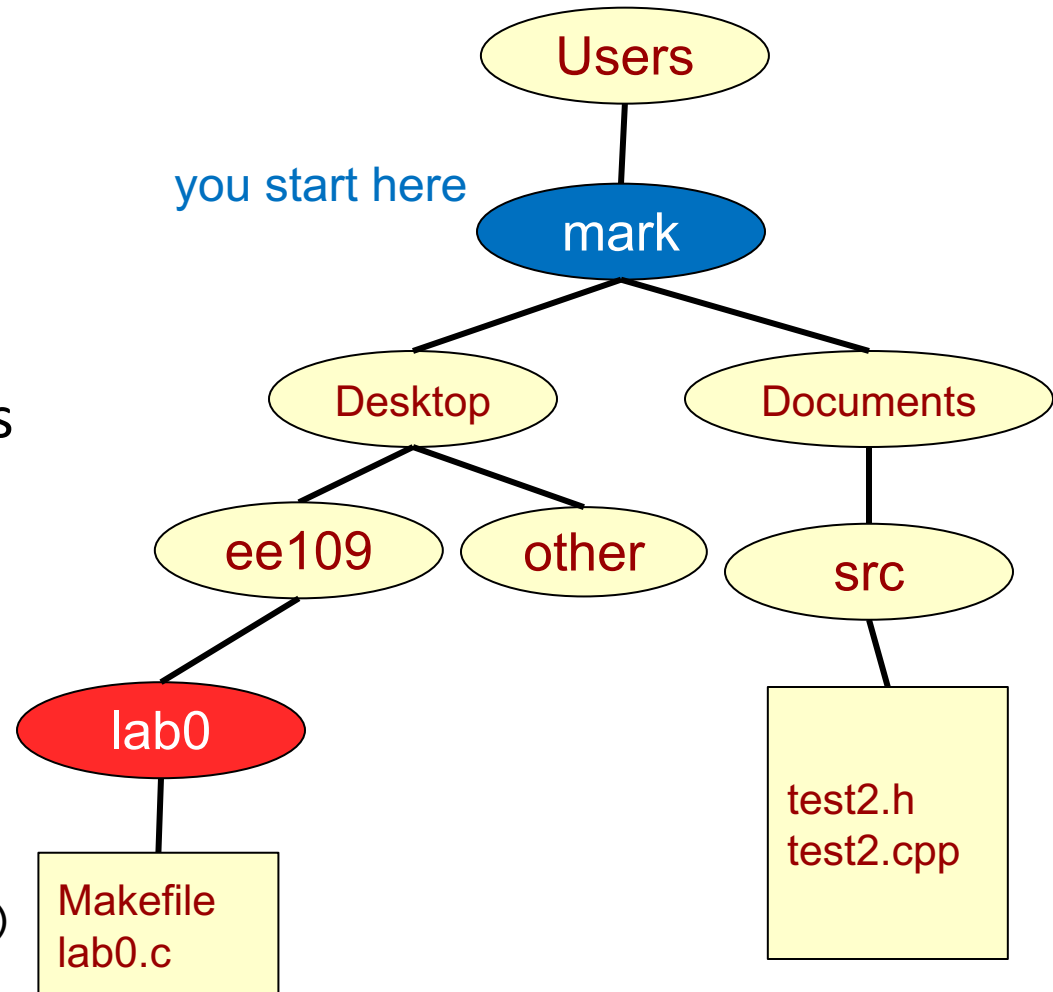
Some Basic Commands

- Here are some helpful commands to use at the command prompt

Command		
ls dir	(Mac) (Win)	List (see) all files in the current folder
pwd	(Mac)	Print working directory shows the current folder location of the terminal
cd <i>dirname</i>		Change directory to a new folder
cp <i>srcfiles dest</i> copy <i>srcfiles dest</i>	(Mac) (Win)	Copy file(s) to a new location
mv <i>srcfile dest</i> move <i>srcfile dest</i>	(Mac) (Win)	Move/rename files to a new name/location
rm <i>srcfiles</i> del <i>srcfiles</i>	(Mac) (Win)	Remove files from the current folder
mkdir <i>dirname</i>		Make directory / create a new folder

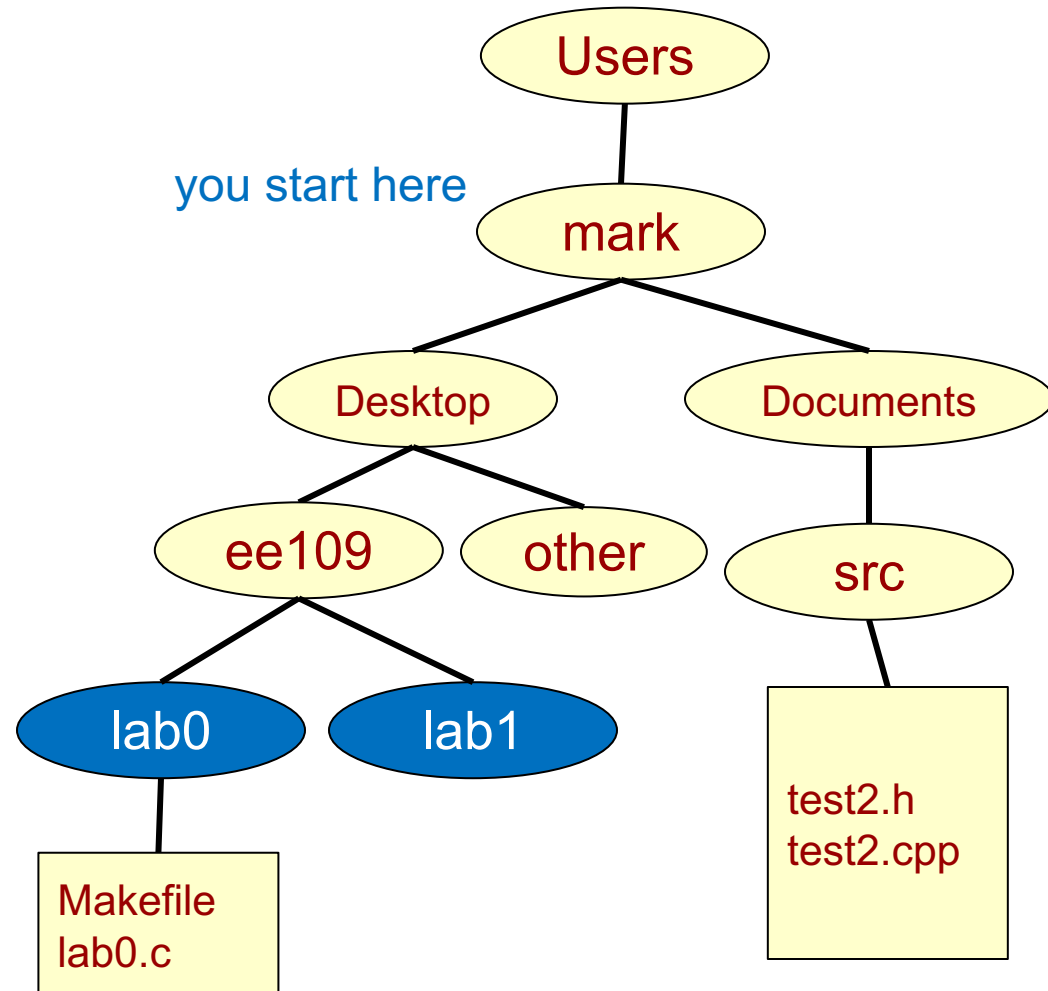
Directory Structure Ex. 1

- Each circle is a directory/folder
- Each name in the box is a file
- Starting from your home (e.g. 'mark') directory/folder...
- Use 'cd' to change directories (folders)
 - `cd Desktop`
 - `cd ee109`
 - `cd lab0`
- Or go multiple folders at a time
 - `cd Desktop/ee109/lab0` (Mac)
 - `cd Desktop\ee109\lab0` (Win)



Directory Structure Ex. 1

- To go up a level use
 - `cd ..`
- To go up 2 levels use
 - `cd ../../..` (Mac)
 - `cd ../../..` (Win)
- Let's make a 'lab1' directory (folder) under the 'ee109' folder using the 'mkdir' command (assuming you are in the lab0 folder)
 - `cd ..`
 - `mkdir lab1`



Shortcuts:

. = Current directory

.. = Parent directory (up one)

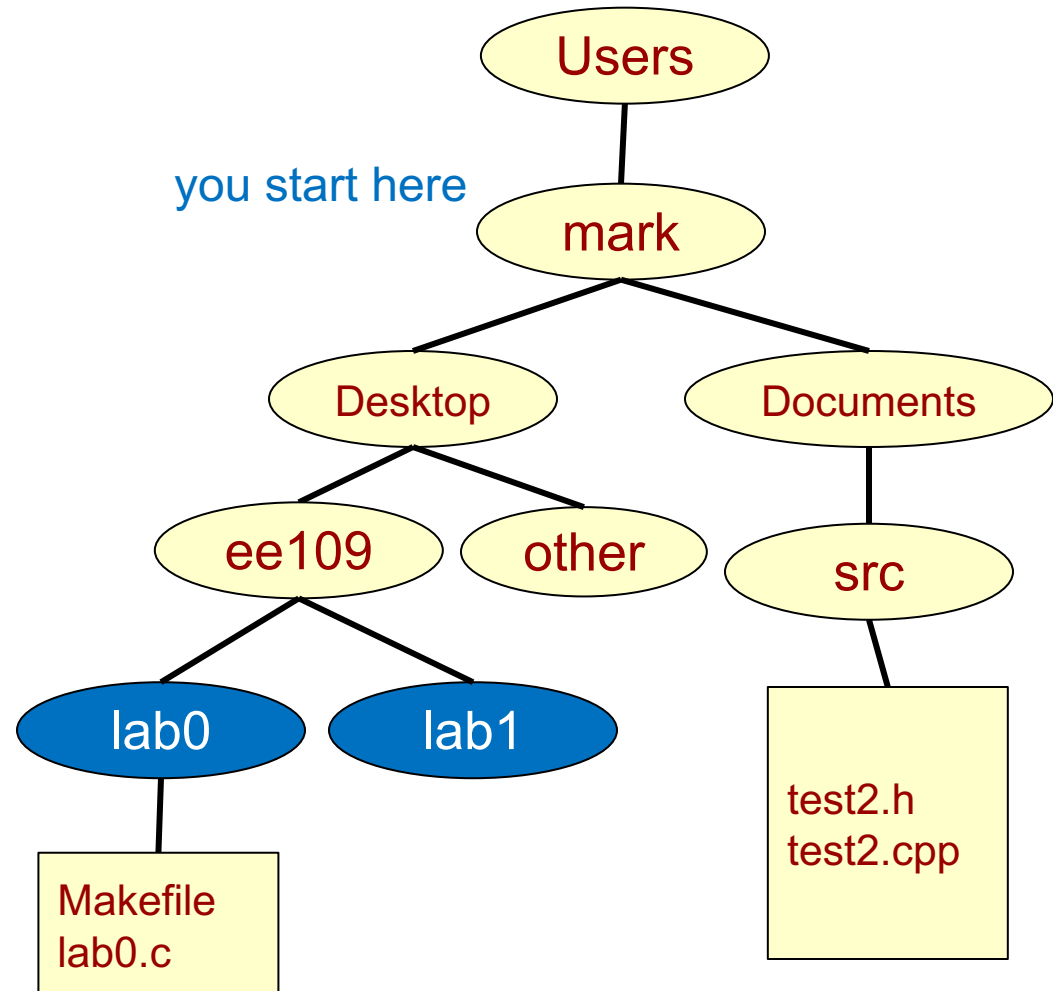
***** = Wildcard to match filenames

Unix commands:

pwd = Print current working dir

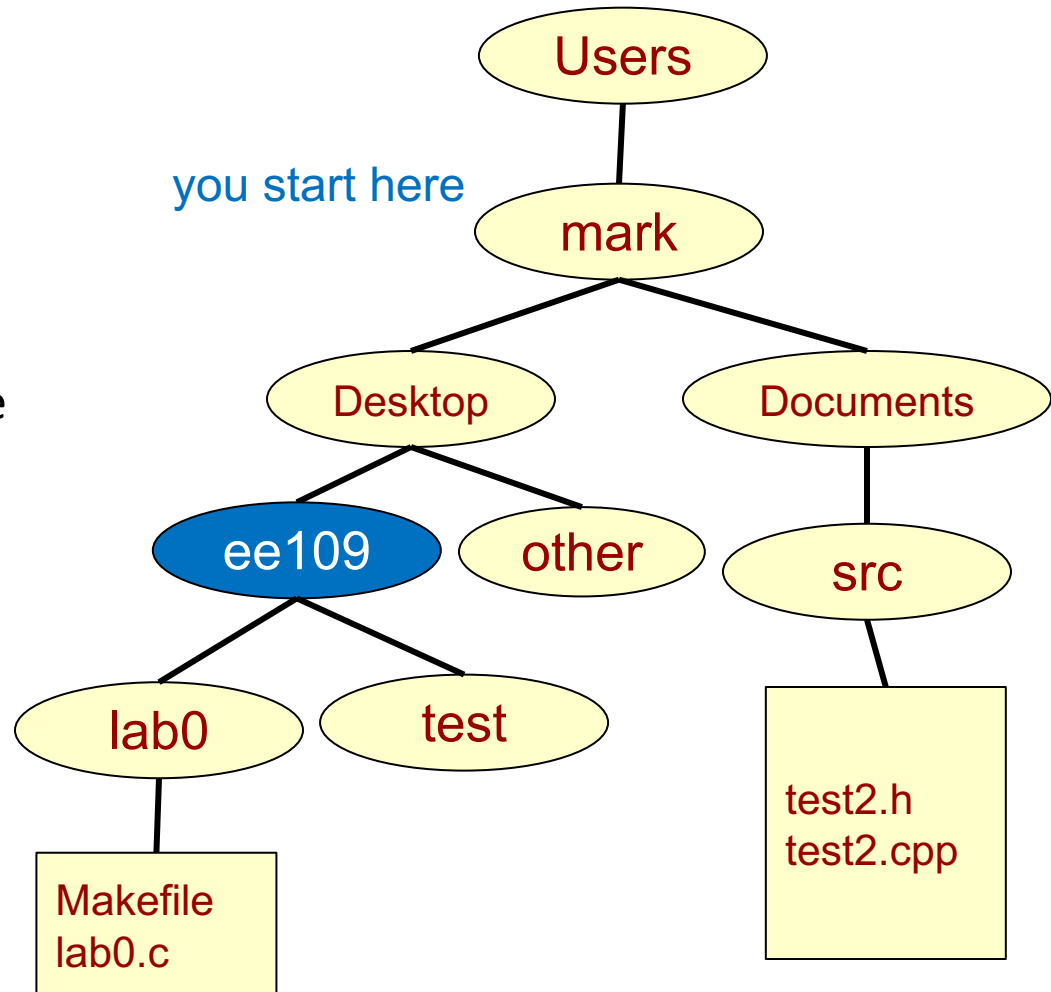
Directory Structure Ex. 1

- To go up a level use
 - `cd ..`
- To go up 2 levels use
 - `cd ../../..` (Mac)
 - `cd ../../..` (Win)
- Let's make a 'lab1' directory (folder) under the 'ee109' folder using the 'mkdir' command (assuming you are in the lab0 folder)
 - `cd ..`
 - `mkdir lab1`



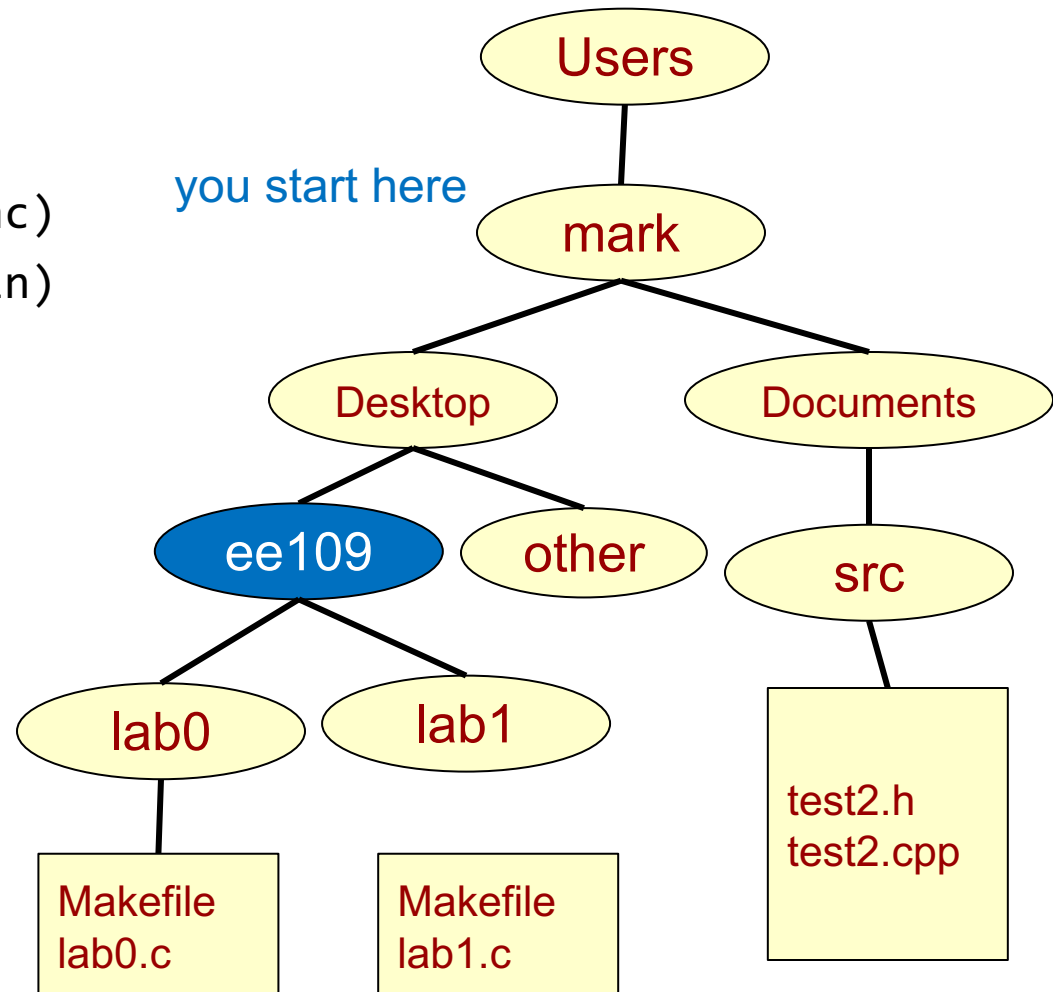
Directory Structure Ex. 1

- To go up a level use
 - `cd ..`
- To go up 2 levels use
 - `cd ../../` (Mac)
 - `cd ../../` (Win)
- Let's make a 'lab1' directory (folder) under the 'ee109' folder using the 'mkdir' command (assuming you are in the lab0 folder)
 - `cd ..`
 - `mkdir lab1`
- Let's say we want to start a new lab with a copy of our old work and just modify it. Let's copy our work
 - Recall I'm in ee109 folder currently
 - `cp lab0/* lab1/` (Mac)
 - `copy lab0* lab1\` (Win)



Directory Structure Ex. 1

- Let's now go into the test folder
 - `cd test`
- Now rename the lab0.c
 - `mv lab0.c lab1.c` (Mac)
 - `rename lab0.c lab1.c` (Win)



Tips

- Use up-arrow to cycle through commands
- Start a folder or filename and hit tab
 - `cd Desk [TAB]` will complete the command to be `cd Desktop`
- You will be using the Command Prompt or Terminal a lot this semester, so put a shortcut to it on your desktop so you can quickly start it up.

COMPILING AND RUNNING A PROJECT

Compilers

- Follow instructions on Lab 0 web page to install the toolchain programs
- Windows requires 2 installs
 - The first (WinAVR) is the compiler & tools
 - The second is the Arduino IDE that is only used to managing your USB connection
- Mac only requires 1 download (Arduino IDE)
 - On occasion some Mac OS'es raise a security violation when you try to install it.
 - Look for the solution in the lab writeup

Editors

- You need a text editor to write your code and edit the 'Makefile'
- We recommend using Visual Studio Code
- Other editors
 - Windows Notepad++
 - Mac BBEdit



Source Code & Makefiles

- Download a sample C-file and a 'Makefile'
 - Makefiles are scripts for how your program should be compiled and how it should be downloaded
 - With a correct Makefile you will only need to run 2 commands to compile and download your program:
 - make (will compile your code)
 - make flash (will download your code)
 - Actually just running 'make flash' will do both but doing them separately gives you the chance to see if the compilation ('make') step completed successfully
- Note 1: For each new assignment/project, you will need to edit the Makefile to point it to the correct filenames for compilation and for the right USB connection
- Note 2: There should be no extension on the filename 'Makefile'. Some editors add .txt (Makefile.txt). In the lab writeup we describe some commands to rename it back to 'Makefile' if that happens

Testing Your Installation

- Once you've edited the source code and Makefile connect the Arduino to your laptop and run "make flash"
- When run the first time you should see a small, yellow LED blinking at a rate of 1 second
- If you like, change the values of 500 in your code to 100, then rerun 'make' and 'make flash'. You should see the LED blinking at a much faster rate

OTHER IMPORTANT NOTES

Academic Integrity

- Labs should be done INDIVIDUALLY except for the first 2 hardware based labs where we explicitly allow teams of 2 or 3 students
- You should NOT show or discuss your actual code with your neighbors/friends but instead talk at the conceptual level or refer them to a TA/CP
- Important: You should not post your lab code publicly (e.g. on Github, etc.) nor attempt to find nor use code available online

WHEN YOU ARRIVE

When You Arrive...

- We will have the software available on a few USB thumb drives if the network connection is slow
- During the first lab session, the lab kits will be distributed.
 - Keep it with you for the rest of the semester
 - Bring it to **ALL** lab sessions
- We also have a few USB-C adapters if you need one, but you will need to buy your own