# EE 109 Lab Detailed Debugging

## Compilation and Programming

- ☐ Be sure you are editing the `.c` files and `Makefile` in the same folder as you are compiling. At the prompt, type `pwd` to see the current folder where you are compiling. In most editors, mouse-over the tab of the file you are editing and ensure the paths are the same.
- ☐ Did you address or verify any warnings are truly not problems? Run `make clean` and just do `make`. Read the warnings. What do they say? Are you sure they don't indicate a problem? (If you are unsure, search them online and see what may be causing these).

## LCD issues

- ☐ Disconnect any wires to your protoboard (this is important), press down on the LCD to make sure it is firmly seated (you should not see the pins when you look from the side at your Arduino and LCD), and then go back to your `lab4` folder and run `make test` to see if anything appears on your LCD. If it does, it's something about your software.
- ☐ Does your LCD go dark as soon as you connect wires to your protoboard OR when you press a button? You most likely have wired your circuit wrong and created a short-circuit between power and ground. The LCD will turn off if too much current flows through it which is indicative of a short circuit.

## General Debugging (My Code Doesn't Work)

- ☐ FIRST, don't try to write the FULL implementation immediately. Write small parts. Rather than outputting a full Morse code sequence when you press a button, can you just turn an LED on (and leave it on) when you sense a button is pressed? Can you do that for each button one at a time, recompiling in between? If nothing appears on your LCD, can you make anything appear on your LCD (just sit in a while loop and show a letter on the screen)? If your up/down counter isn't working, can you just make it count 0-9 (in the up direction) on the LCD regardless of buttons/direction/etc. (i.e. a much simpler task)?
- ☐ Use the DMM and some probes to measure the voltage at the button output / Arduino inputs. Does it produce 5V when you ARE NOT pressing and 0 when you are? If not, have you configured everything correctly (DDRs, pull-up resistors, etc.).
- ☐ If an LED doesn't light up, are you sure it is plugged in correctly? Can you write a small program that outputs a constant '1' = high voltage from your corresponding PORT bit and see if the LED can light up in that case? Add an infinite loop above your normal while loop just for temporary testing. Does the LED light up?

  ```
  while(1) { PORTC |= (1 << 2); }
  ```

- [ ] I have gone back to the slides for specific configuration registers and verified all the bits I think I should set are set correctly. I have looked carefully at my bit fiddling commands to ensure parentheses and comparisons seem correct.
- [ ] Similarly to the advice above, when your code doesn't work COMMENT OUT SECTIONS / FEATURES of code until you CAN get something to work. Then comment these sections back in to see which one causes it to potentially break again.
- [ ] Does it seem like your ISR may not be working? Just temporarily put an `lcd_stringout("here")` in the ISR (even though it's generally incorrect to have long-latency work done in an ISR) and then run the program. When you trigger the interrupt behavior, do you see that text? Alternatively, pick an unused output bit, configure its DDR bit as an **output** and flip/toggle the bit (`PORTC ^= (1 << 2);`) in your ISR. Connect it to an LED or the oscilloscope. Do you see it flipping (you may need to use Single trigger mode if the ISR runs aperiodically.