

# Lab Assignment 1 CSE474

Prof. Vikram Iyer<sup>1</sup>

Spring 22

University of Washington

## Getting Started with the Arduino Mega

### Learning Objectives

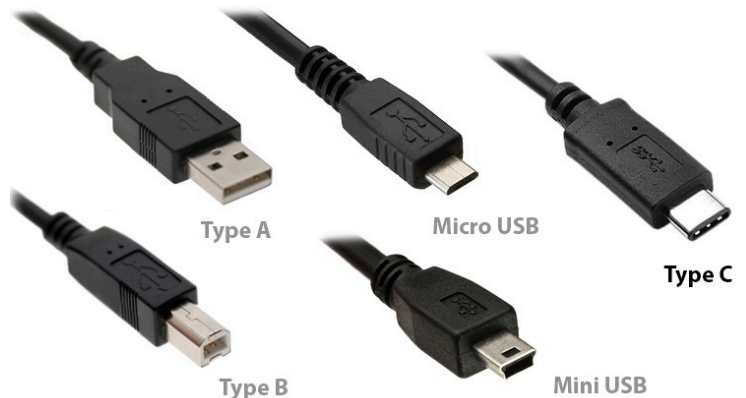
With successful completion of this lab, the student will be able to

- Install and set up Arduino IDE
- Build and run a basic sketch (program) using the Arduino Libraries
- Modify and demonstrate blinking light code and speaker output tone.



### Equipment ([See Equipment List with sources](#))

- 1) Arduino Mega Microcontroller board
- 2) External Arduino power supply
- 3) LEDs and 250 Ohm resistors
- 4) Small 8-Ohm Speaker
- 5) USB- Type-B cable (with USB-A or USB-C for computer end) (you probably got one with Arduino) (image: globetek.com)
- 6) Solderless Breadboard
- 7) Wires (either with pins (see the kits), or cut and strip your own 22AWG solid wire).



### Helpful Resources:

If you haven't worked with electronics before, or want to brush up, check out this list of compiled resources that will help you learn the basics. [\[LINK\]](#) Feel free to ask TAs any questions in office hours/Ed Discussion!

### Turn In and Demo Requirements:

[\[LINK\]](#)

## Technical Requirements

In this lab we will use some standard Arduino code packages to blink an LED and to make a tone on a speaker.

## Required Procedures

(suggestions for breaking down the lab, step-by-step testing, etc. )

### 1. **Part I** Intro to Arduino:

Install the open-source Arduino Interactive Development Environment (IDE) from [HERE](#) on your Windows, Mac OS, or Linux computer. In case of difficulty, try [Arduino Troubleshooting Guide](#), or [Arduino Help Center](#). Then if those don't solve the problem (along with random googling) please reach out to the ECE474 TA's.

- 1.1. Open the Arduino IDE by clicking its desktop icon (OS dependent).
- 1.2. Open the blink example by "File->Examples->0.1Basics->Blink".
- 1.3. "Break" your code by deleting the required semicolon (";") on line 28 (line number shows in the lower left). Hit the checkmark (upper left) and observe the orange error messages.
- 1.4. Fix your bug and build the code again (checkmark)
- 1.5. Connect your Arduino MEGA to your PC via USB (observe steady green "ON" light indicating power)
- 1.6. Configure IDE for your board and connection. Remember, because C is a compiled language and different chips have varying memory and peripherals we have to select the right target platform. To do this:
  - 1.6.1. "Tools->Board->Arduino Mega 2560 or Mega ADK"
  - 1.6.2. "Tools->Serial Port->" << select one and cross your fingers >> (if you see just one serial port, you're all set. If you see more than one, you have to experiment, unfortunately)
- 1.7. Upload "blink" to the Arduino (right arrow icon at the top). You should see a pair of LEDs on the board rapidly flashing. This indicates code downloading into your Arduino. After that your code automatically starts.
- 1.8. The LED labeled "L" near the USB connector should blink 1 time every two seconds.

### 2. **Part II:** Modifying your sketch and your LED

- 2.1. Change the delay values from 1000ms to 200ms. Recompile (checkmark) and re-upload your sketch.
- 2.2. The LED should go faster.
- 2.3. Now unplug the USB cable from your Arduino and plug in the 120VAC power adapter. The Arduino has non-volatile memory so the flashing light program should run again. If not, try pressing the reset button.

### 3. **Part III** External LED Hardware

- 3.1. Connect a 250Ohm resistor between the cathode (short wire) of an LED and a GND/0V pin on the Arduino. (See picture below for reference) Make the resistor-LED connection by either breadboard connecting, soldering, wire twisting, or alligator clip as you prefer.

- 3.2. Connect the long LED wire with IO/PWM pin 10.
- 3.3. Modify the blink.ino code on line 10 so that the LED pin is now pin 10
- 3.4. Compile and upload. Verify your LED attached to pin 10 is blinking as expected.
4. Part IV Multiple tasks
  - 4.1. Keeping your external LED wired up, wire the 8 Ohm speaker between pin 2 and +3.3V.
  - 4.2. Modify the blink “sketch” (code) to make clicks on the speaker.
  - 4.3. Now add new code to blink both the on-board LED (pin 13), and the external LED (pin 10) such that one is ON when the other is OFF and the speaker clicks when the LEDs change state.
5. Part V Differing Periodicities
  - 5.1. Modify your code so that the LEDs flash the same as 4.3, but instead of clicking, the speaker emits a continuous tone of 250Hz. **Pro tip (and lab requirement):** write your code so the sound turns off after a few seconds (it will drive you crazy!). Make sure your LEDs keep flashing in the correct pattern after audio stops (your code didn’t just crash).

NOTE: Do not use the tone() function for this. This can cause your program to freeze up and not allow you to blink the LED.

## Special Grading Criteria

- There are no special grading criteria for this lab see [Lab Guidance](#).

