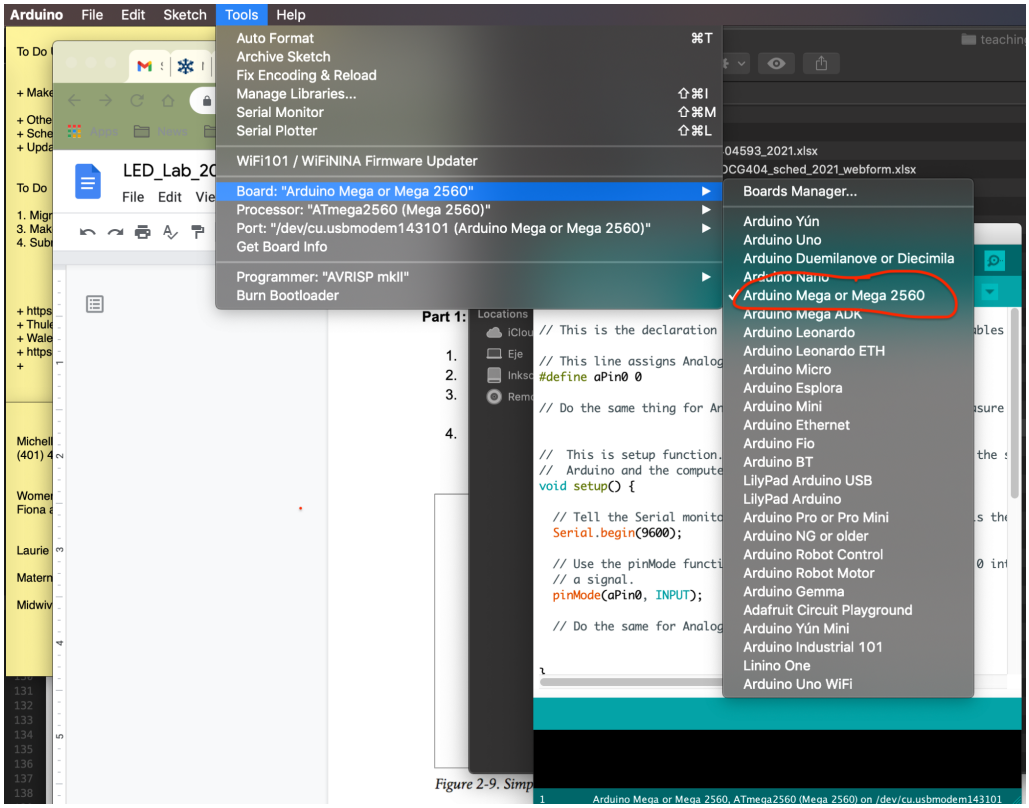
Arduino LED Lab

Lab Kit: Each group receives a kit that includes: (1) Arduino Mega 2560 microcontroller, (2) Bread board for prototyping circuits, (3) Jumper wires for connecting circuit components, (4) USB cable to power the Arduino board and communicate with the computer, (5) a single colored LED, and (6) a single bread board resistor with color coded bands.

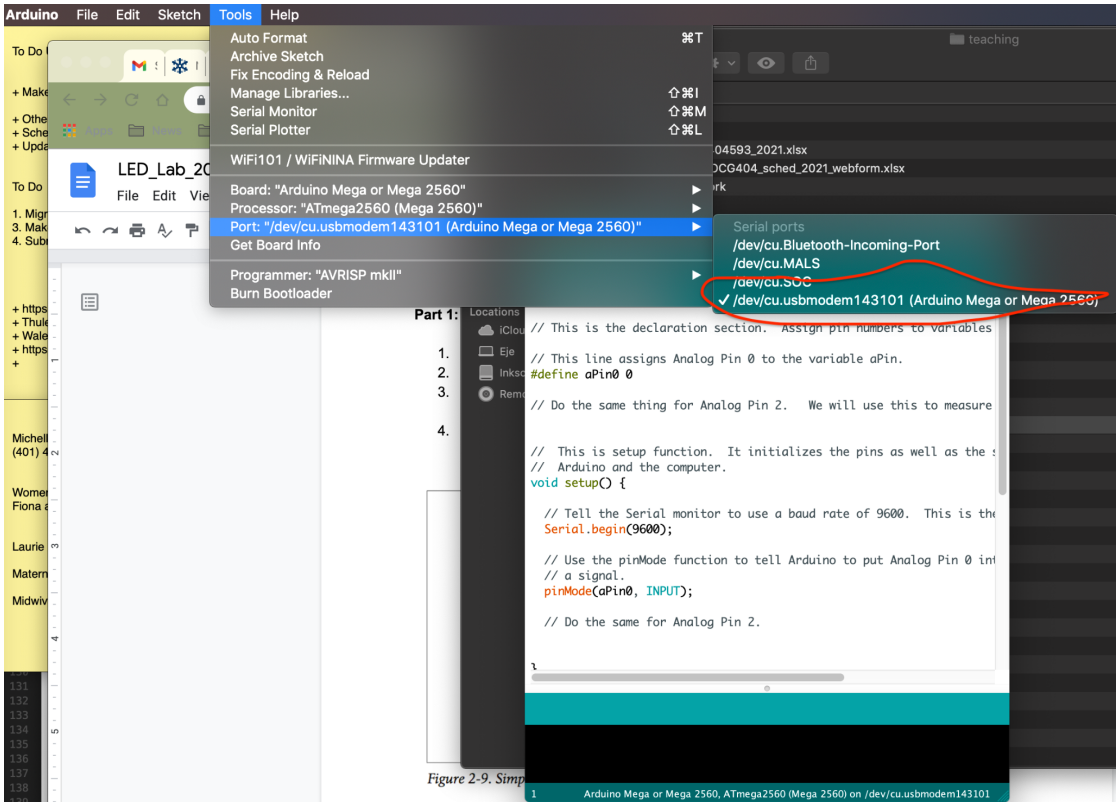
Part 1: Install and verify your copy of the Arduino IDE.

1. Visit https://www.arduino.cc/en/Main/Software_ and download the binary for your OS.
2. Download the led_voltage_analysis.ino from the course website.
3. Open led_voltage_analysis.ino from the Arduino IDE. (We will go over the language construction together, which is C++ with bumpers).
4. Verify that the correct hardware is selected in the Tools menu. This should be Arduino Mega 2560:



5. Verify that the correct serial port is selected on your computer. In Windows, you can find the serial ports in the Device Manager. In Mac/Linux, you can find a list of serial devices

by typing $ ls /dev at the command prompt.



Part 2: Build your circuit using the Arduino Mega, breadboard, resistor, LED and wire jumpers.

We are going to set up and analyze the LED circuit to understand how to use Ohm's law and Kirchoff's law to determine the properties of the circuit. We will use these basic circuit analysis tools to understand a thermistor circuit for measuring temperature and a soil moisture circuit for measuring soil water content. Both use variable resistance.

For now, focus your attention on Figure 2.9 from "Realworld Instrumentation", below.

1. This is intended as practice translating circuit diagrams to "actual" wiring. For power source, use the Vin pin on the Arduino. For Ground, use the GND pin adjacent to Vin. (*NOTE: LED's only permit current to flow in one direction. Attach the long pin closest to +3.3V and the short pin closest to GND.*)
    a. When you think you have the circuit completed, connect the USB cable to the Arduino and a computer. This will provide power to the Arduino. If the circuit is correct, the LED should light up and stay lit.

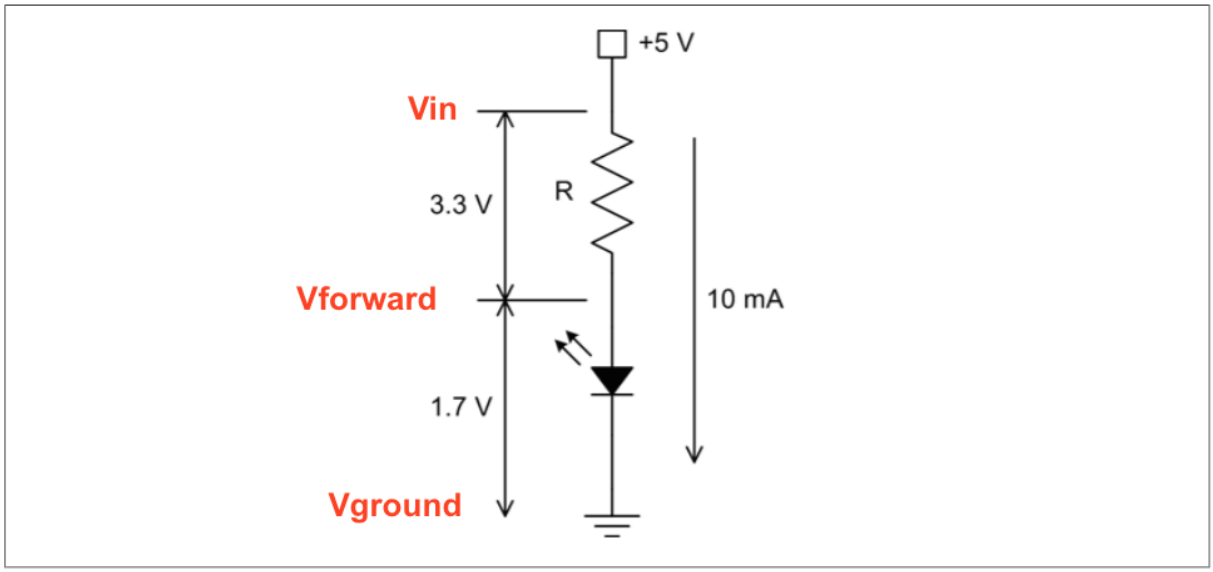
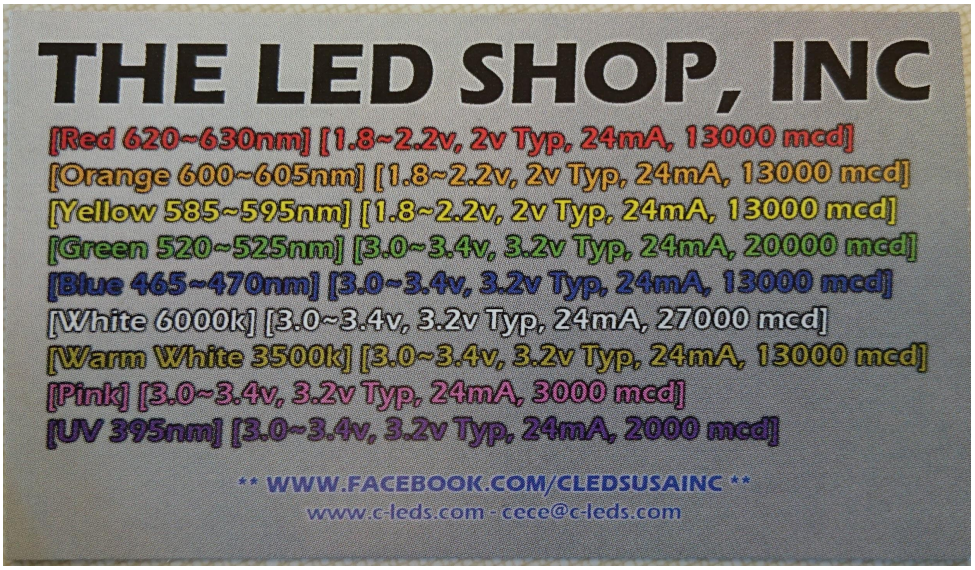


Figure 2-9. Simple LED circuit

Analyzing the LED circuit.

b.



1. What color is your LED? Based upon color, the LED shop figure tells us the expected voltage drop at the max allowable current. What should the voltage drop and max current through the LED be? (In Figure 2.9, this is labeled as 1.7 V. In our case, it might not be 1.7 V). Report the values here in your writeup.

2. The resistor is added to the circuit, to avoid passing too much current through the LED, thereby burning it out. Considering the voltage output of the Arduino, and the max allowable current for your LED, use Ohm's law (V = IR) to determine the value of the resistor that will be sufficient to protect the LED, given the max rated current in the LED SHOP chart. Report the value here.

3. Google 'resistor band chart' and use the colored bands on your resistor to determine its resistance in Ohms. Report the value here. Is it enough to protect the LED?

4. Now that you have a theoretical voltage drop across the LED (from question 1), use Kirchoff's voltage law to estimate the voltage drop across the resistor. $\sum_j V_j = 0;$ the sum of the voltages around a circuit must be zero. (In Figure 2.9, this is labeled as 3.3 V). Report the value here.

Part 3: Using the Arduino to measure V_{in} and $V_{forward}$. We want to compare the actual properties of the circuit to the ones we calculated using Ohm's law and Kirchoff's law.

5. Unplug the Arduino from the computer so it is not powered as you modify the circuit.

6. Now we will measure the voltage drop across the resistor using jumpers and the Analog IO pins. Connect jumpers to your breadboard so that they measure at positions $V_{forward}$ and V_{in} (see Figure 2.9). Connect Vin to A0 on the Arduino. Connect $V_{forward}$ to A2 on the Arduino.

7. Next, bring up the Arduino software on your computer (or download it if you don't have it). Make a sketch to report the values of Vin and Vf to the serial monitor.

8. Compute the voltage drop across the resistor from V_{in} and $V_{forward}$. Report the value here.

9. Use the voltage drop from question 8 and Ohm's law to determine the actual current flowing through the resistor, which is the same as the current flowing through the entire circuit. How does it compare to the max current specified by the LED SHOP figure? Are we protecting the LED?

At Home/In class: Write an Arduino algorithm to turn the LED on and off, based on keyboard input from your computer. Below is some pseudocode to get you started. This will require rewiring of the LED circuit so that Vin can be turned on and off. Use a digital I/O pin for this.

1. Initialize digital pin and assign to a variable.
2. Initialize serial input and assign to variable.
3. Create void setup() { } function.
4. Set pin mode for digital pin
5. Create void loop() { } function.
6. Check the state of the digital pin.
7. Use if/else statements to change its state.

What to turn in:

1. Upload a PDF of your group's answers to the questions 1-9 of the LED analysis section above. (This is an in-class assignment).
2. Upload a working copy of the led_voltage_analysis.ino. (This is an in-class/take home assignment).
3. Upload a working copy of the polled_led On/Off (polled) LED .ino sketch. (This is an in-class/take home assignment).