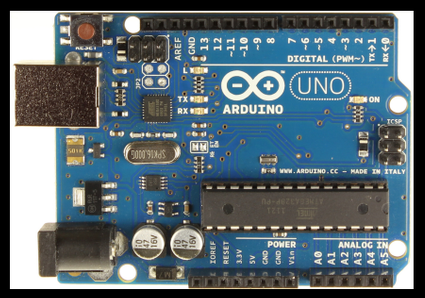
How to:
Control and Sense

General Schema

PC

IDE

INPUT



OUTPUT

µController/µProcessor

Arduino IDE

- IDE is Integrated Development Environment
  - It's what you program the Arduino with
- "Official" Version available here:
  - https://www.arduino.cc/en/Main/Software
  - There are "Nightly" versions, useful if "stable" version has issues
- Adafruit fork of Arduino IDE here:
  - https://learn.adafruit.com/adafruit-arduino-ide-setup/arduino-1-dot-6-x-ide
  - If you are using Flora, Gemma, or Micro, use Adafruit's IDE

µController/µProcessor

- µController/µProcessor is a device you can program to receive input and produce outputs
- Arduino is one class of µControllers
  - Uno, Due, Micro, Nano, Flora, Gemma, etc
  - Arduino can do simple computation as well
  - Mostly programmed in C-ish
- If you need to do more heavy (local) computation, a µProcessor is a better option
  - Raspberry Pi, BeagleBone, Galileo, etc.
  - Can be programmed in other languages
- µController and µProcessors for the most part are relatively interchangeable

Output Exercise

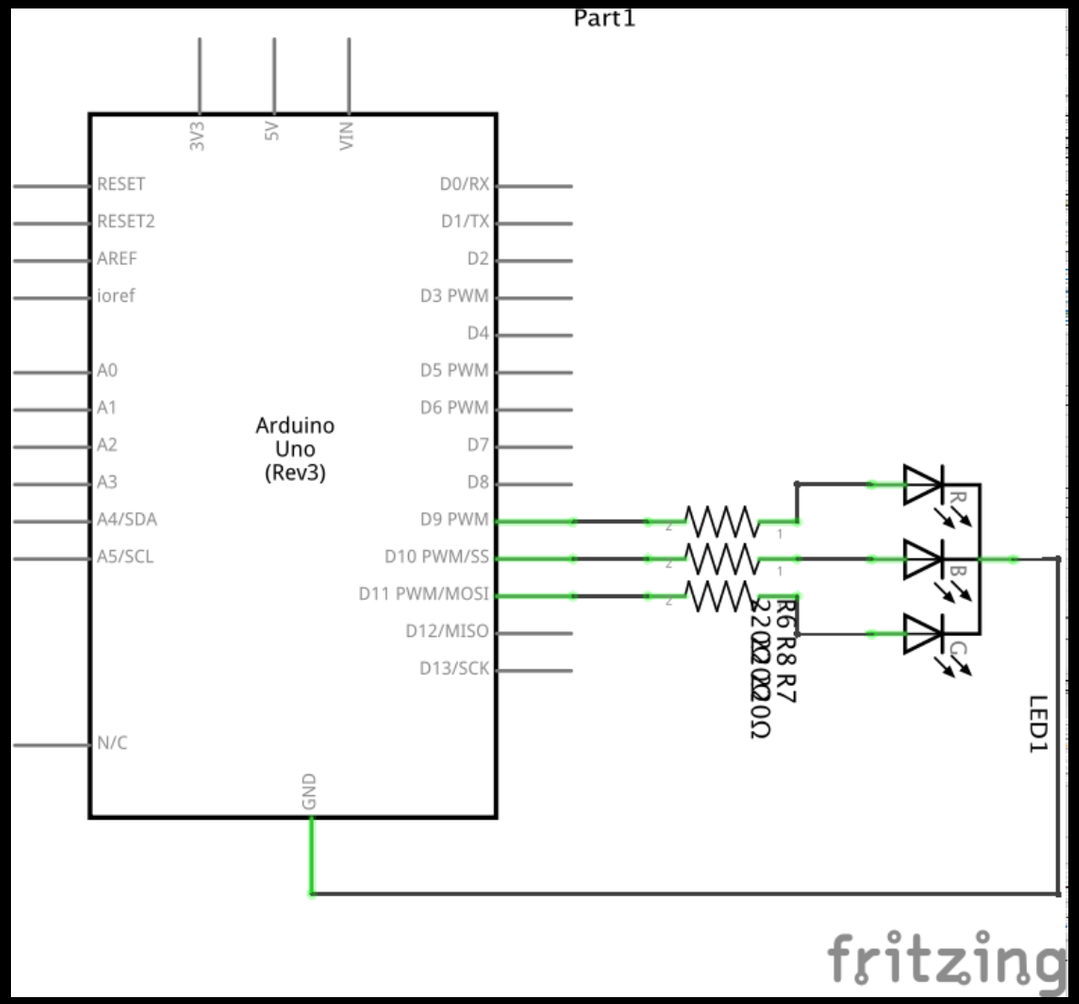
- For this workshop, you have three LEDs (or an RGB LED), 3 resistors, a breadboard, an Arduino, and a USB cable.

- The goal of this exercise is to fade on then fade off LED1, wait 1000ms, then fade on/off LED2, wait, etc.

- I will upload my version to the course website.

Output Circuit

- The RGB LED is really 3 LEDs stacked on top of each other.
- The “R”, “G”, and “B” legs are connected to pins 9,10,11 in series with the resistor. The ground is connected to ground.
- For some applications, like LEDs, we need to put a resistor in series in order to limit the current. In lecture, I used a 1K resistor. See: https://en.wikipedia.org/wiki/File:Diode-IV-Curve.svg



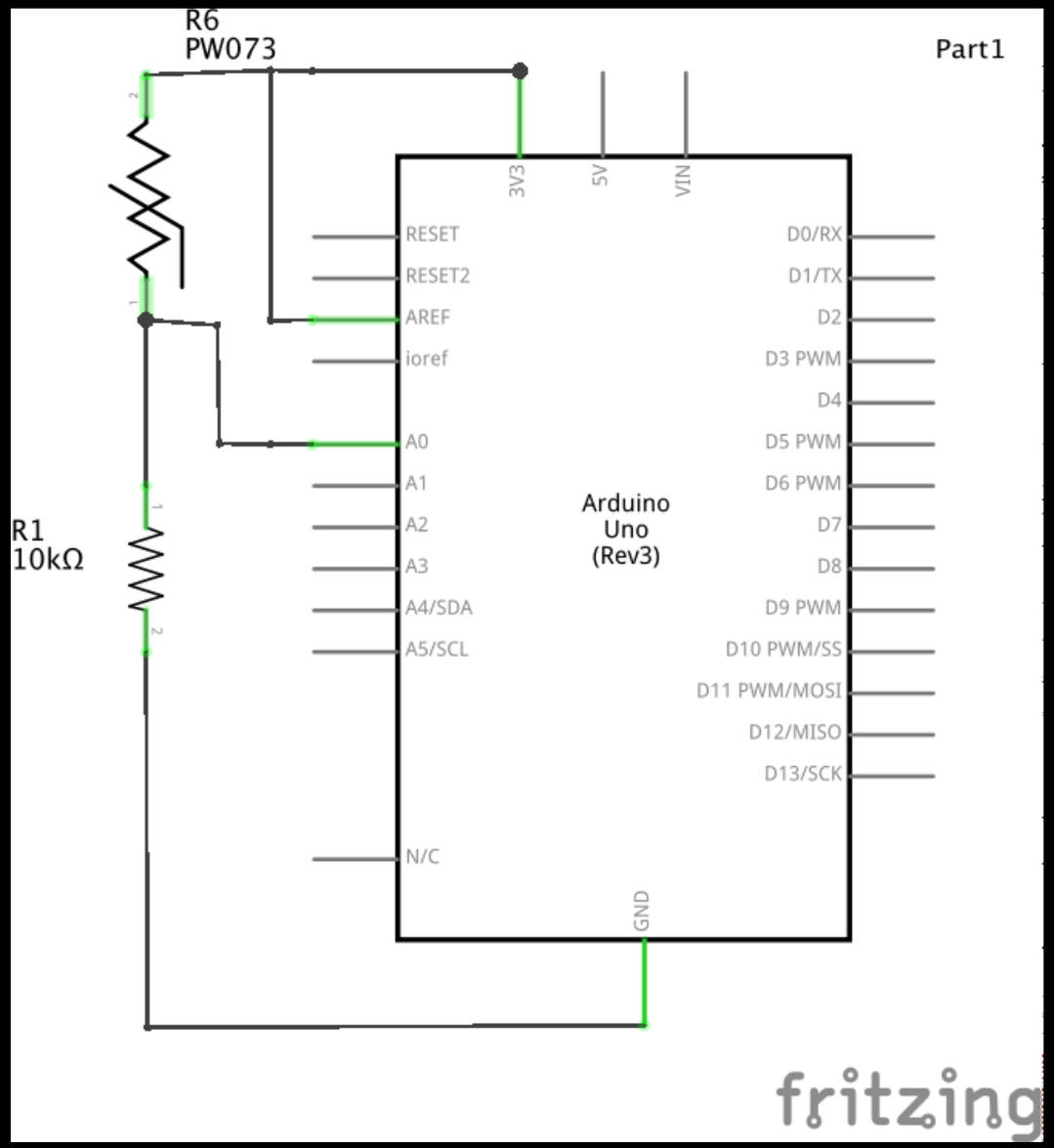
Input Exercise

- For this workshop, in addition, you now have a sensor and a 10K resistor.
- The goal of this exercise is to measure inputs from a sensor and log them to serial so that a computer can record them.
- Use the known value of the resistor to estimate the resistance of the sensor.
- You can use MatLab (or your favorite language) to interact through serial

```
MATLAB R2014a
s = serial('/dev/cu.usbmodem1411'); % You'd use 'COM#' in Windows.
set(s,'BaudRate',9600); % Set the Baud in Arduino, make sure it matches!
fopen(s); % This opens the connection.
out = fscanf(s); % This is what actually saves it to a variable.
```

Input Circuit

- The thermistor decreases it's resistance as it's temperature increases
- In order to read its resistance, we create a simple voltage divider with a known resistor (10K) in order to measure its resistance.
- We use the A0 (Analog In) pin to read the voltage across the thermistor (0-1023)
- Dividing the value of the series resistor (10K) by the voltage gives us the resistance of the thermistor
- We can then use the Steinhart equation to get the temperature.



Control Exercise

- For this workshop, in addition, you now have a toy motor.
- Combine the skills you've learned from the previous two workshops to control the speed of a motor from a sensor input.
- I will upload my version to the course website.

Control Circuit

- In this circuit, keep everything from the Input Circuit but swap out the thermistor for the pressure sensor.
- We are now introducing a MOSFET, a toy DC Brushed Motor, another 10K resistor, a Flyback Diode, a Bypass Capacitor and a battery pack
- Using the resistance of the pressure sensor, we can drive the motor by PWM'ing the gate of the MOSFET.
- Note, this is not a particularly good way to drive a large motor, just a small toy vibration motor. If you want to drive a motor, please come and talk to us.