Use Lab 0 as a reference to this Lab

Click hyperlinks (blue underlined text) to learn more about a particular topic or concept.

Flashing Lights

Objectives

The purpose of this lab exercise is to:

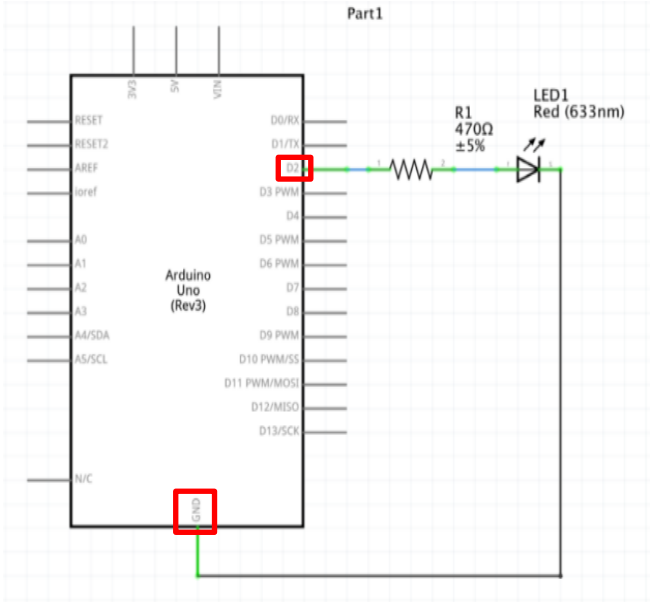
- Teach how to connect an LED to an Arduino
- Turn the LED on and off
- Explore the delay function
- Explore the Serial console

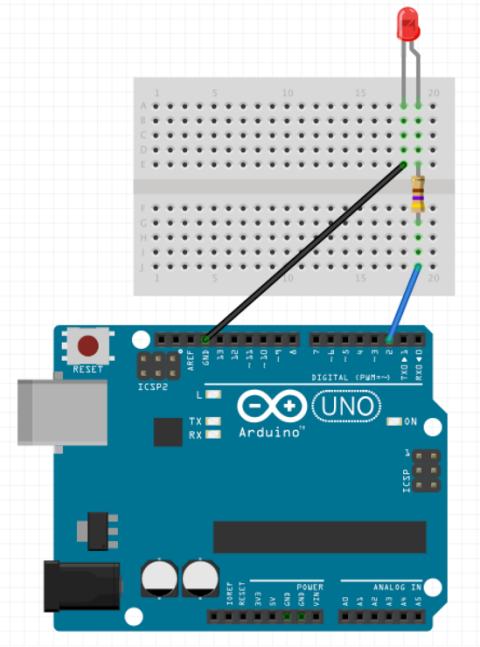
Parts/Equipment Required

- Arduino Uno
- Breadboard
- Laptop
- USB cable
- LED
- 470Ω Resistor
- Various wire

Hardware

Wire your breadboard to look like the provided schematic. You can use whatever colour wire you want connecting the resistor to the breadboard and Arduino, but you MUST use black to go from the breadboard to ground. Follow these steps to hardwire the Circuit before programming the Arduino:

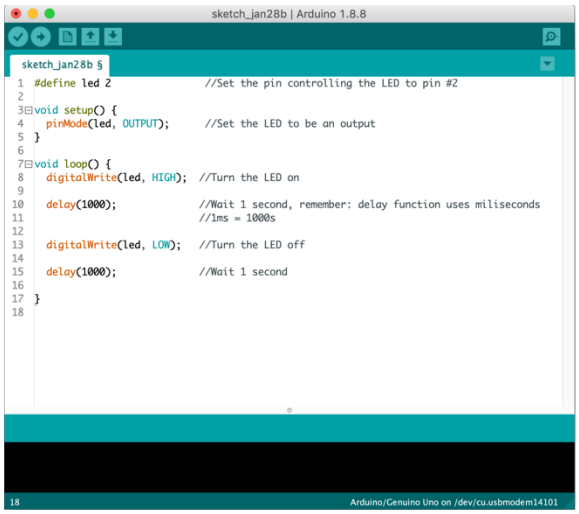




- Connect to Pin 2
- 470Ω Resistor connected to LED
- LED connected to GND
- Correct polarity of the LED

Software

Type this code into the Arduino IDE. If you wired the lab correctly you shouldn't have to modify anything to get it to work. If it doesn't work once you upload the code, try changing the polarity of the LED, maybe you wired it backwards by accident. If it still doesn't work, ask for help.

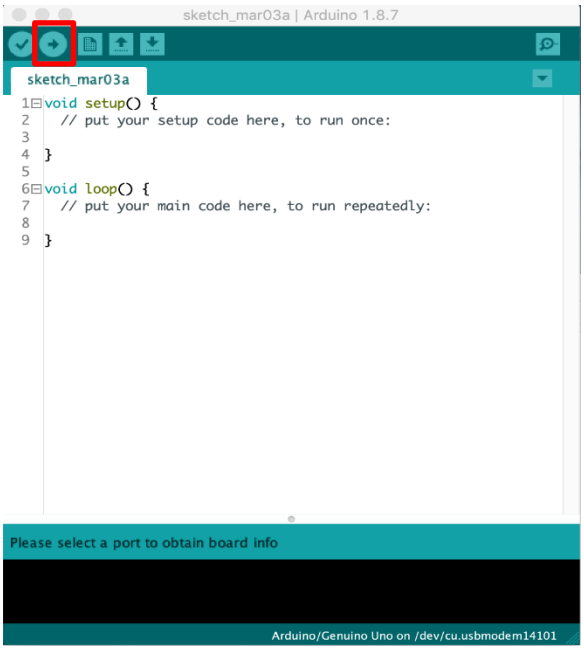


http://arduino.zaugg.ca/code/Lab1_Code.html

Uploading the code

Uploading, sometimes called flashing, is the process of translating the code you've written that is human readable into code that is machine readable, then getting the microcontroller on the Arduino to run the machine readable code.

1. Plug the Arduino into your computer using the USB cable
2. In the Arduino IDE, under the tools menu, select "Board:", then "Arduino/Geniuno Uno"
3. Still under the tools menu, select "Port:", then the serial port for the Arduino
4. To check that things are working, under the tools menu, click "get board info". A menu should open with a bunch of information about your Arduino such as the serial number. If no window opens, get help.
5. Click the Upload button to send the software onto the Arduino



Once the code has been uploaded onto the Arduino you can close the Arduino IDE on your computer and the code will still run. Better still, you can completely unplug the Arduino from your laptop and use a wall adaptor to power the Arduino and the code will still run.