



## Wi-Fi Settings

**SSID:** AcroboticGuest  
**Password:** 13meetup37

**Class: Intro to Electronics with Arduino!**

# Welcome!

## Class outline

- Overview of Arduino and hardware electronics [25min]
- Q&A [5–10min]
- Hands-on, instructor-led activities [60min]
- Break [5–10min]
- Hands-on, instructor-led activities [60min]
- Experimentation [if time permits]

# Welcome!

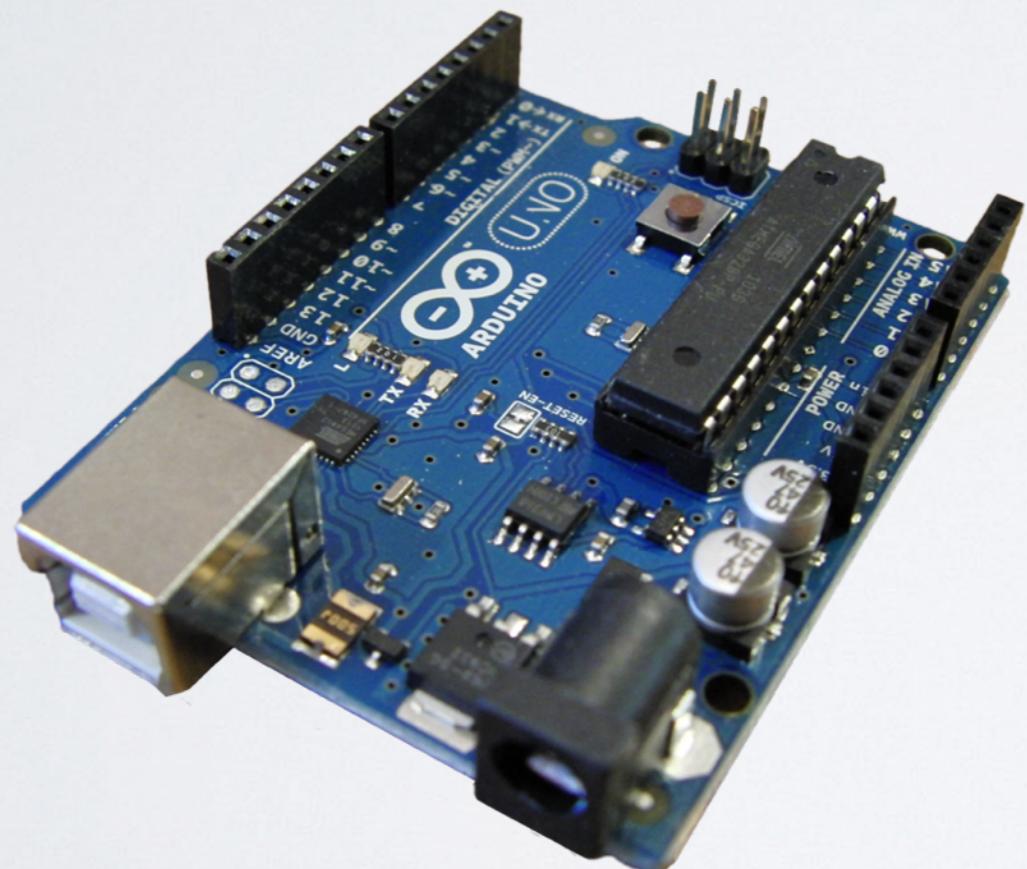
## Class outline

- **Overview of Arduino and hardware electronics [25min]**
- Q&A [5–10min]
- Hands-on, instructor-led activities [60min]
- Break [5–10min]
- Hands-on, instructor-led activities [60min]
- Experimentation [if time permits]

# What is Arduino?

Arduino is the *brand* of a toolset that enables anyone to interact with the physical world.

The Arduino toolset is comprised by both hardware and software:



+

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0018". The main window displays the "Blink" sketch. The code is as follows:

```
/*
 * Blink
 *
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * The circuit:
 * * LED connected from digital pin 13 to ground.
 *
 * Note: On most Arduino boards, there is already an LED on the board
 * connected to pin 13, so you don't need any extra components for this example.
 *
 * Created 1 June 2005
 * By David Cuartielles
 *
 * http://arduino.cc/en/Tutorial/Blink
 *
 * based on an orginal by H. Barragan for the Wiring i/o board
 */
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
```

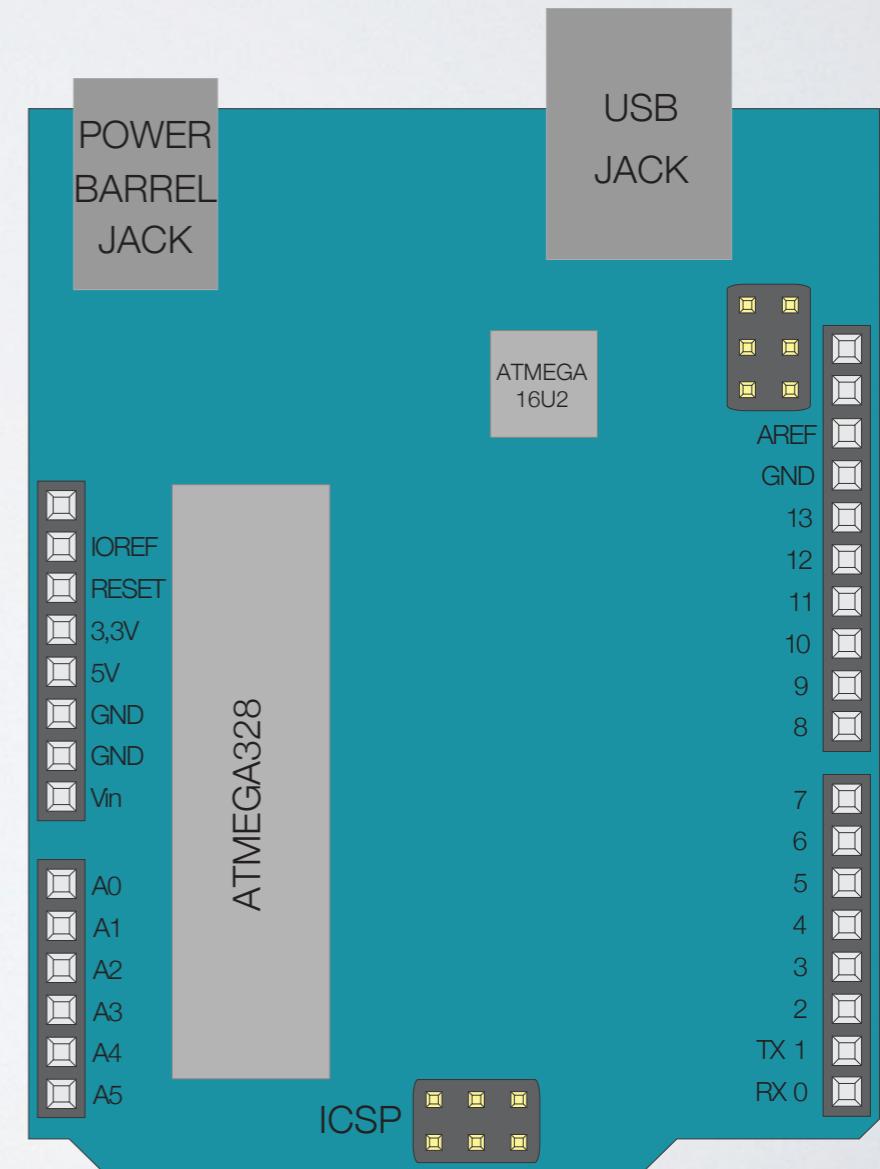
# What is Arduino?

## Arduino Hardware (plural!)

Arduino hardware are development boards for Microcontrollers—small computer chips that contain a processor core, memory, and programmable input and output.

The Arduino boards typically comprise:

- DC power regulation circuitry
- Serial communication port (USB)
- Microcontroller Unit (MCU)
- Digital Input/Output (GPIO)
- Analog Input

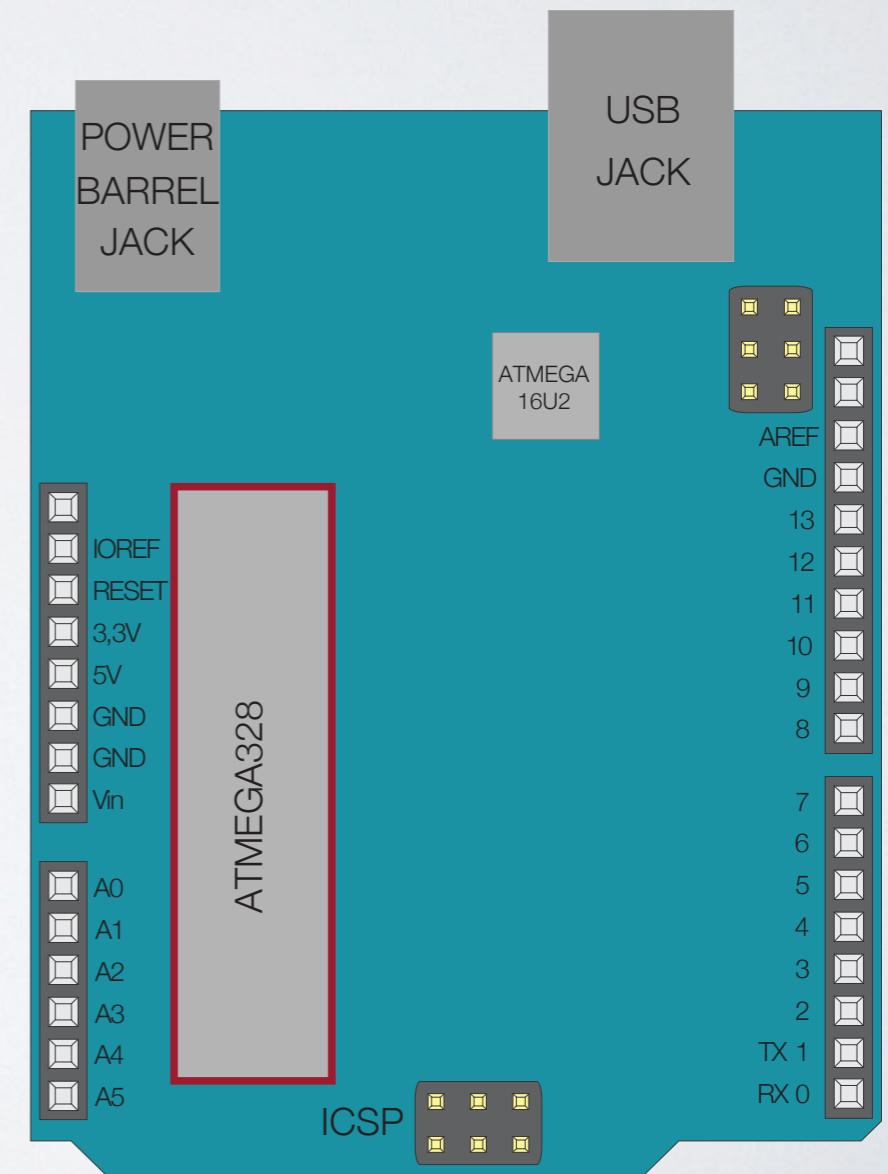


# What is Arduino?

## Arduino Hardware (plural!)

The Arduino boards facilitate the process of programming a microcontroller by allowing us to:

- Program the onboard MCU from a computer
- Connect electronic components and devices to the MCU's Digital and Analog I/O (e.g., LED, temperature sensor, motor, LCD screen)
- Add capabilities to the MCU using **Arduino Shields!**



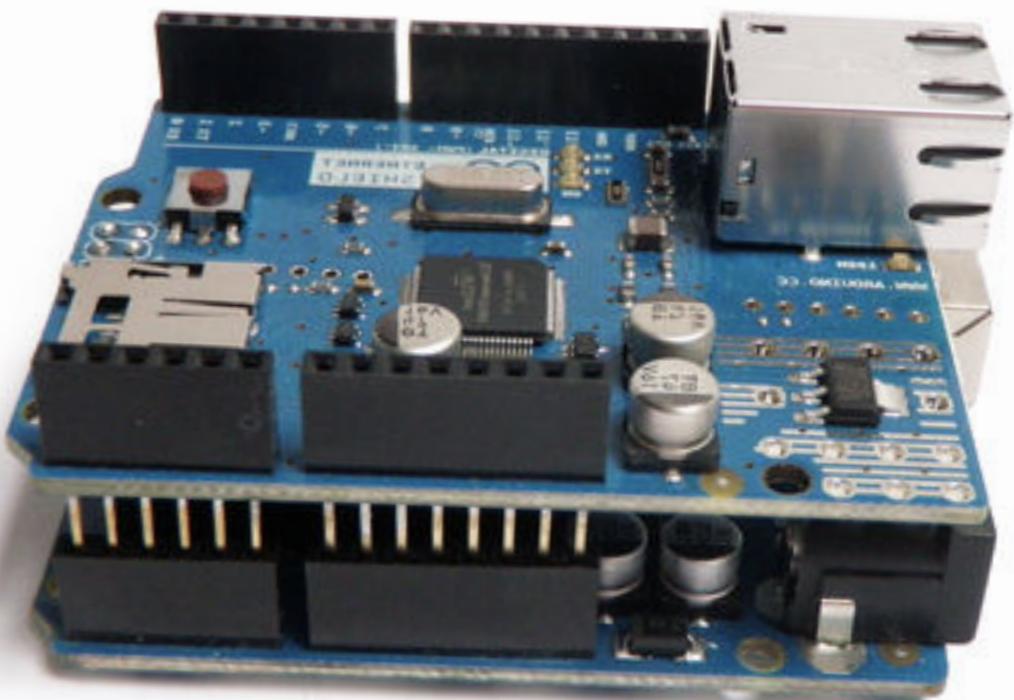
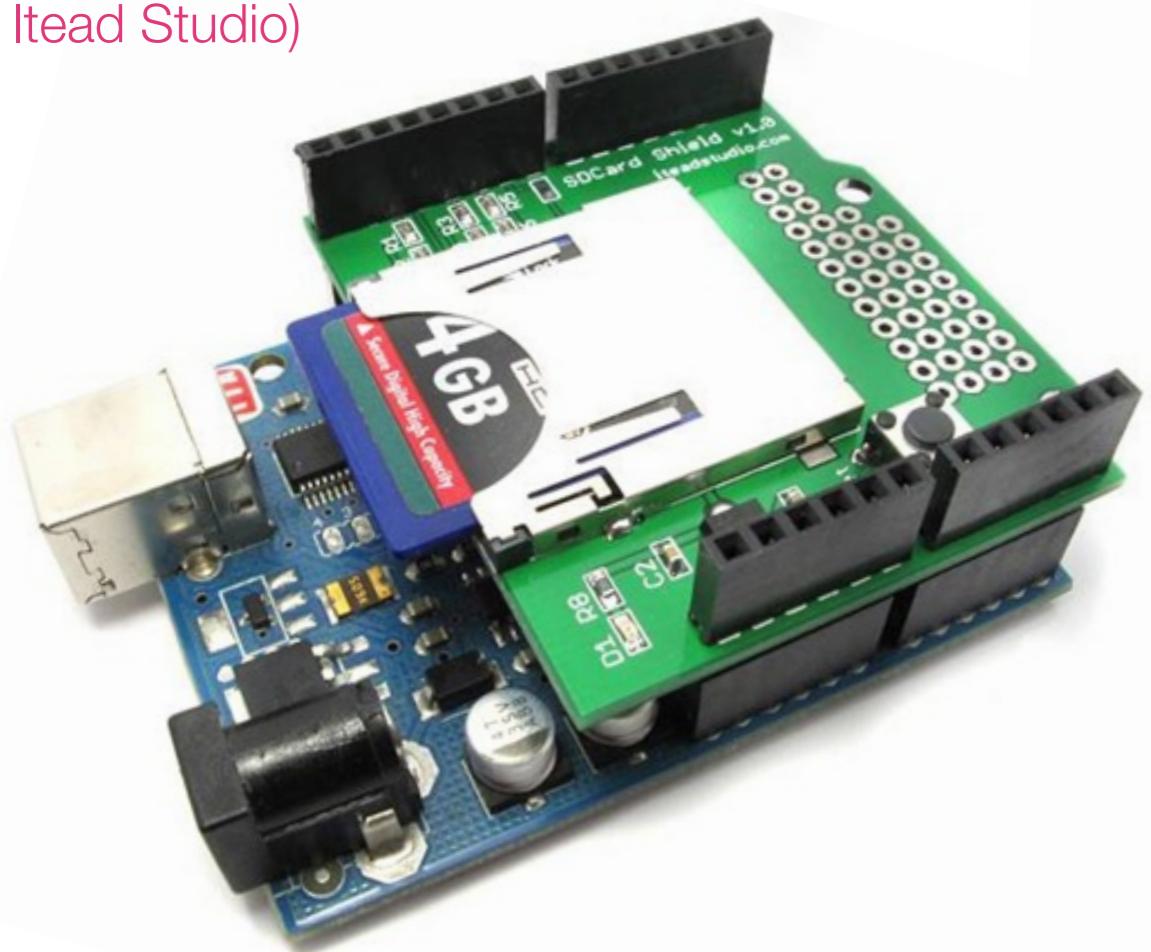
# What is Arduino?

## Arduino Hardware (plural!)

The Arduino boards are usually stackable so that users connect daughterboards (shields) to add any desired capability.

SDCard Shield for the Arduino Uno board

(by Itead Studio)



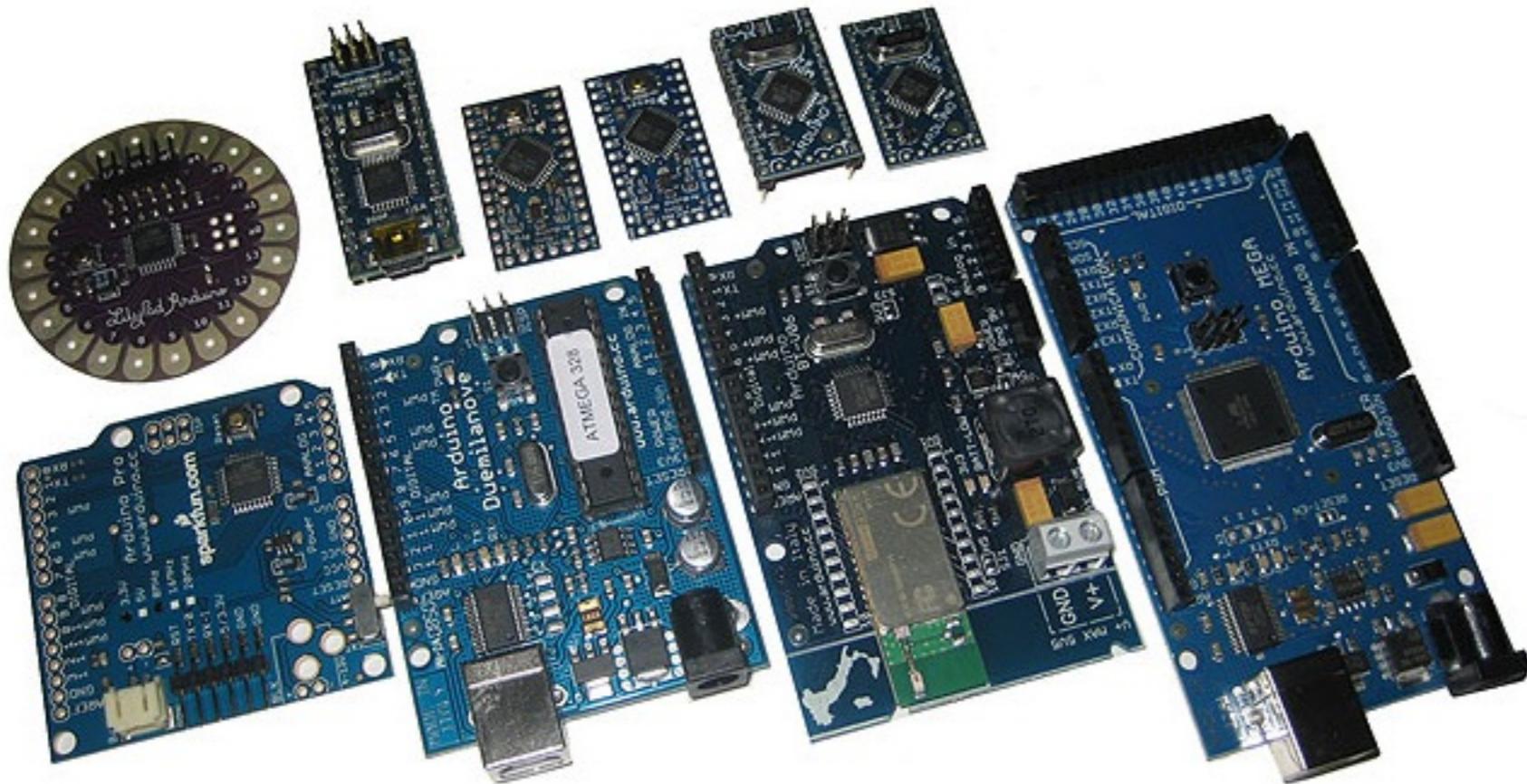
Ethernet Shield for the Arduino Uno board

(by Arduino)

# What is Arduino?

## Arduino Hardware (plural!)

There are many, Many, MANY Arduino boards and clones that vary in form factor, computing power, number and type of peripherals, etc.



[http://blog.arduino.cc/wp-content/uploads/2013/11/ArduinoEvolution\\_make.jpg](http://blog.arduino.cc/wp-content/uploads/2013/11/ArduinoEvolution_make.jpg)

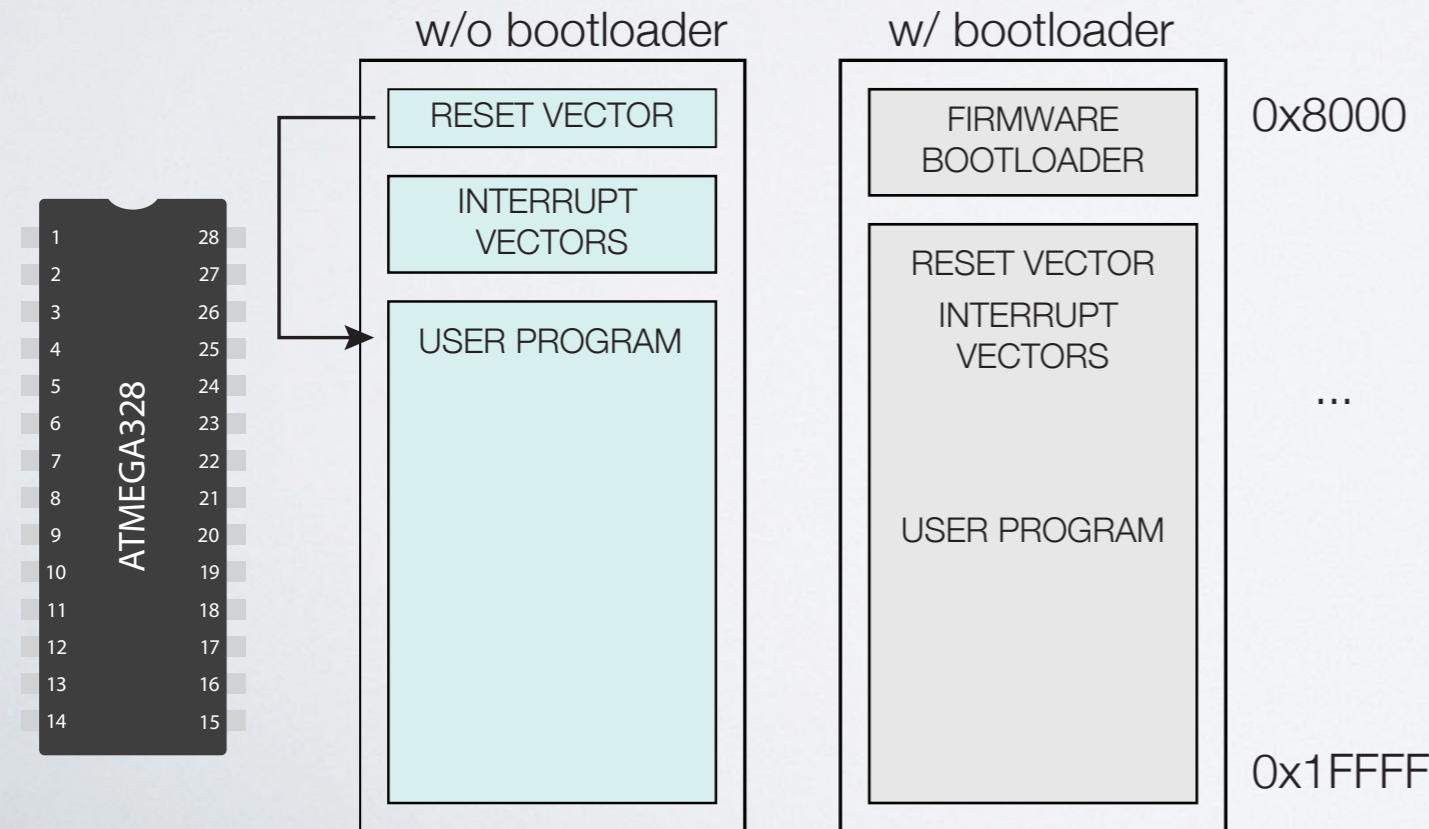
# What is Arduino?

## Arduino Software (plural!)

The Arduino software is comprised by:

- A *firmware bootloader* that enables us to easily load programs onto the MCU
- A computer application (*IDE*) to edit, compile, and upload our programs, as well as communicate via USB with the Arduino boards

### MICROCONTROLLER MEMORY MAPS



A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0018". The central area displays the C code for the "Blink" sketch. The code is as follows:

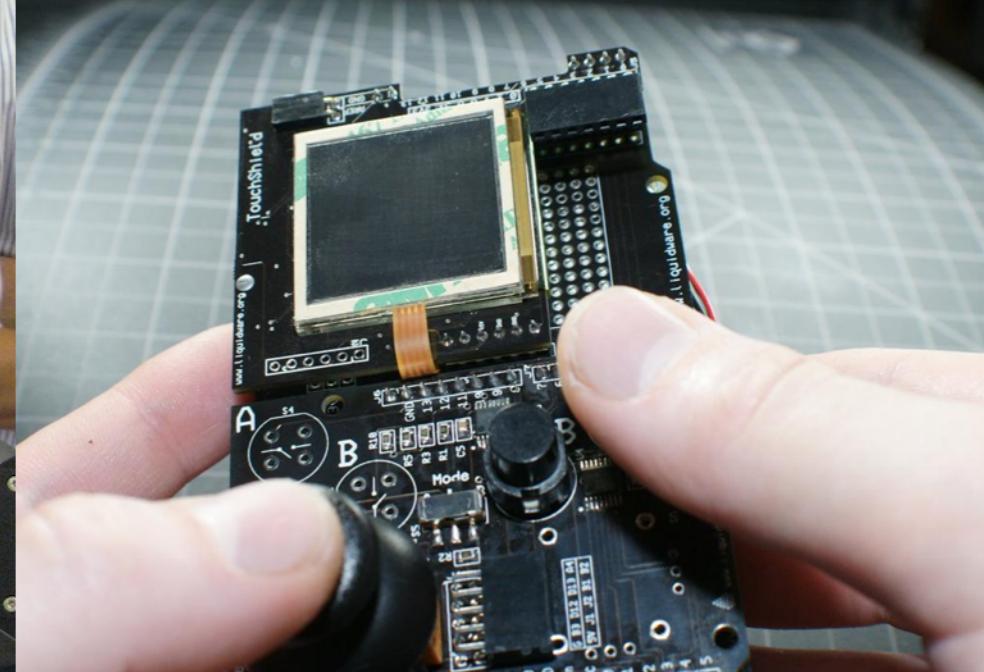
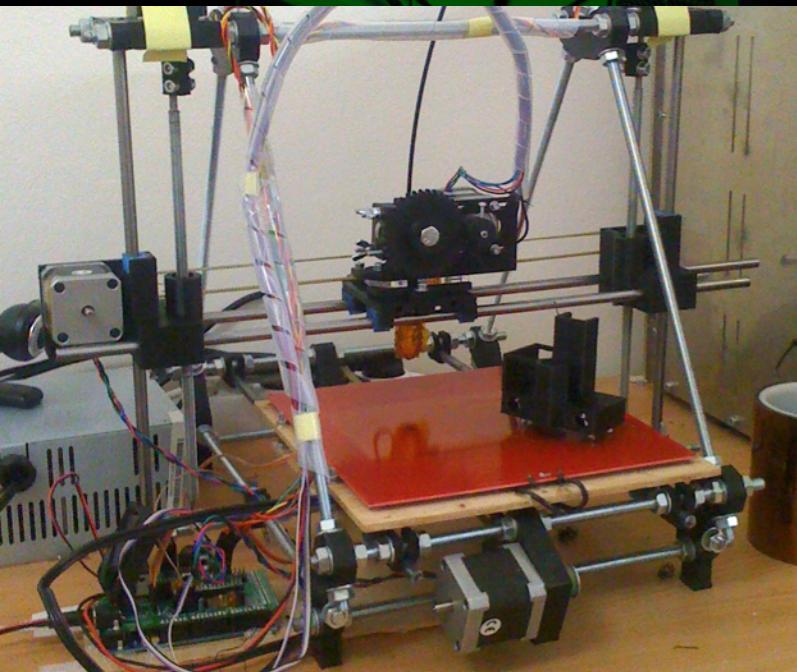
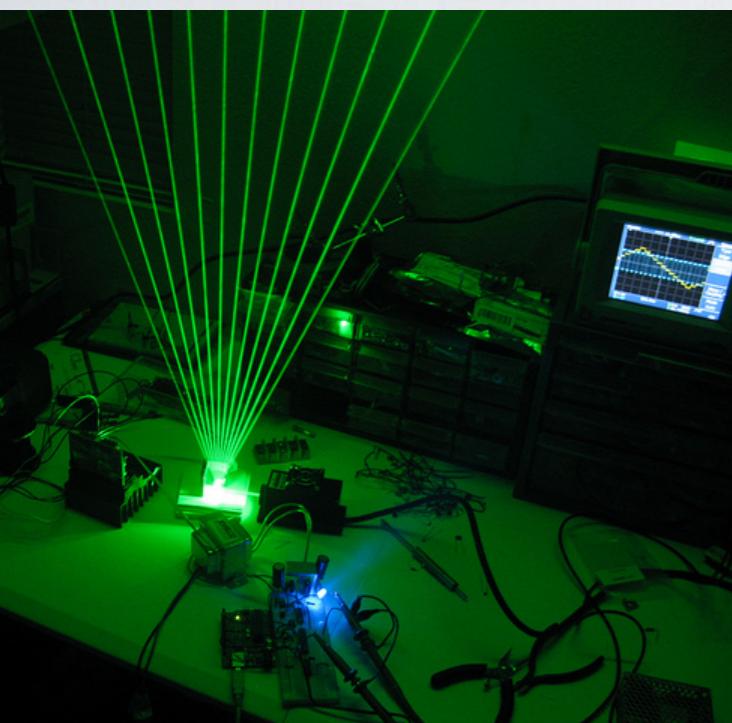
```
/*
 * Blink
 *
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * The circuit:
 * * LED connected from digital pin 13 to ground.
 *
 * Note: On most Arduino boards, there is already an LED on the board
 * connected to pin 13, so you don't need any extra components for this example.
 *
 * Created 1 June 2005
 * By David Cuartielles
 *
 * http://arduino.cc/en/Tutorial/Blink
 *
 * based on an orginal by H. Barragan for the Wiring i/o board
 */
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
```

# What uses does the Arduino toolset have?

The Arduino toolset enables us to interact with the physical world. As such, the ever-increasing range of applications is only limited by the users' imagination.



# What uses does the Arduino toolset have?

From the horse's mouth:

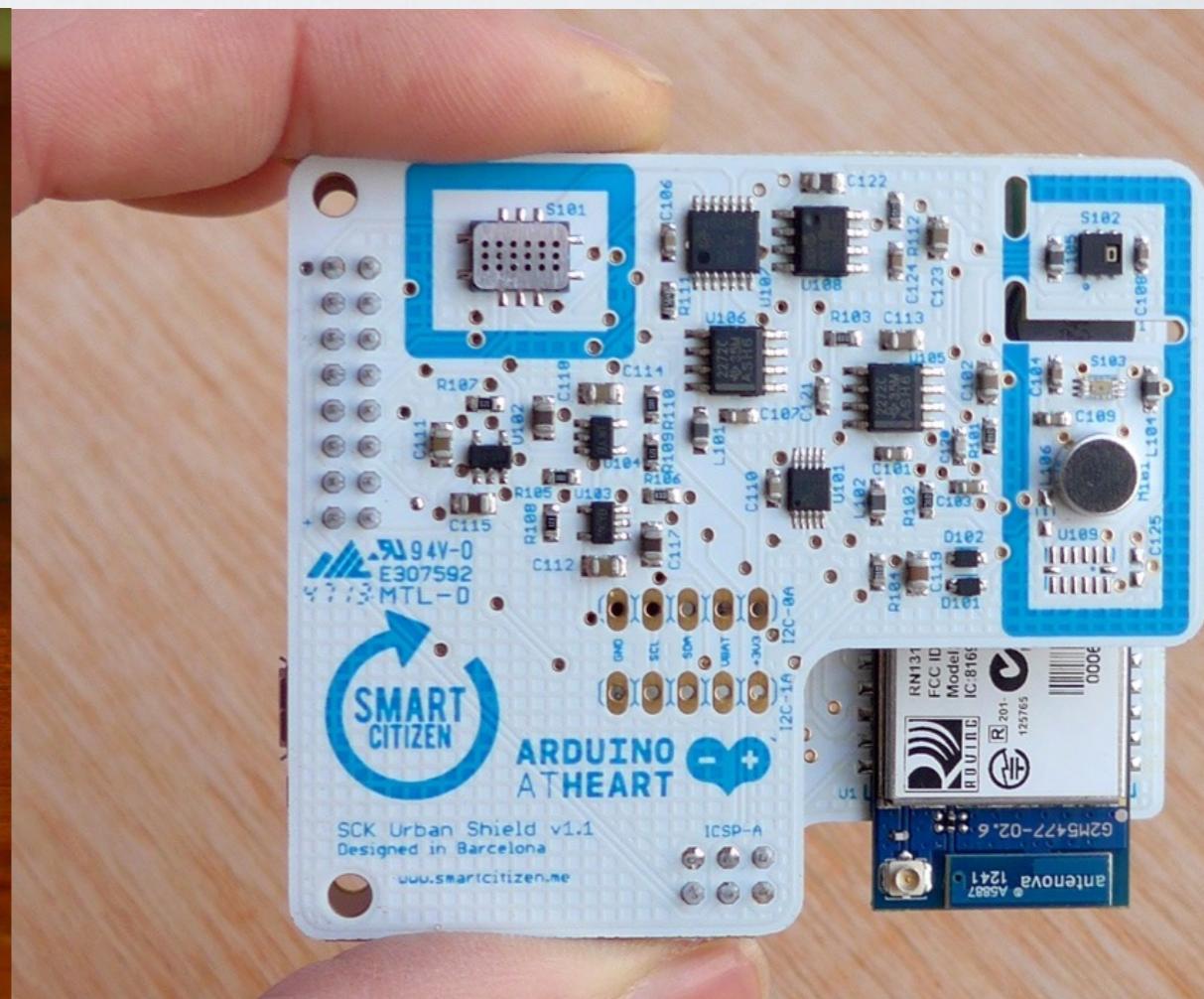
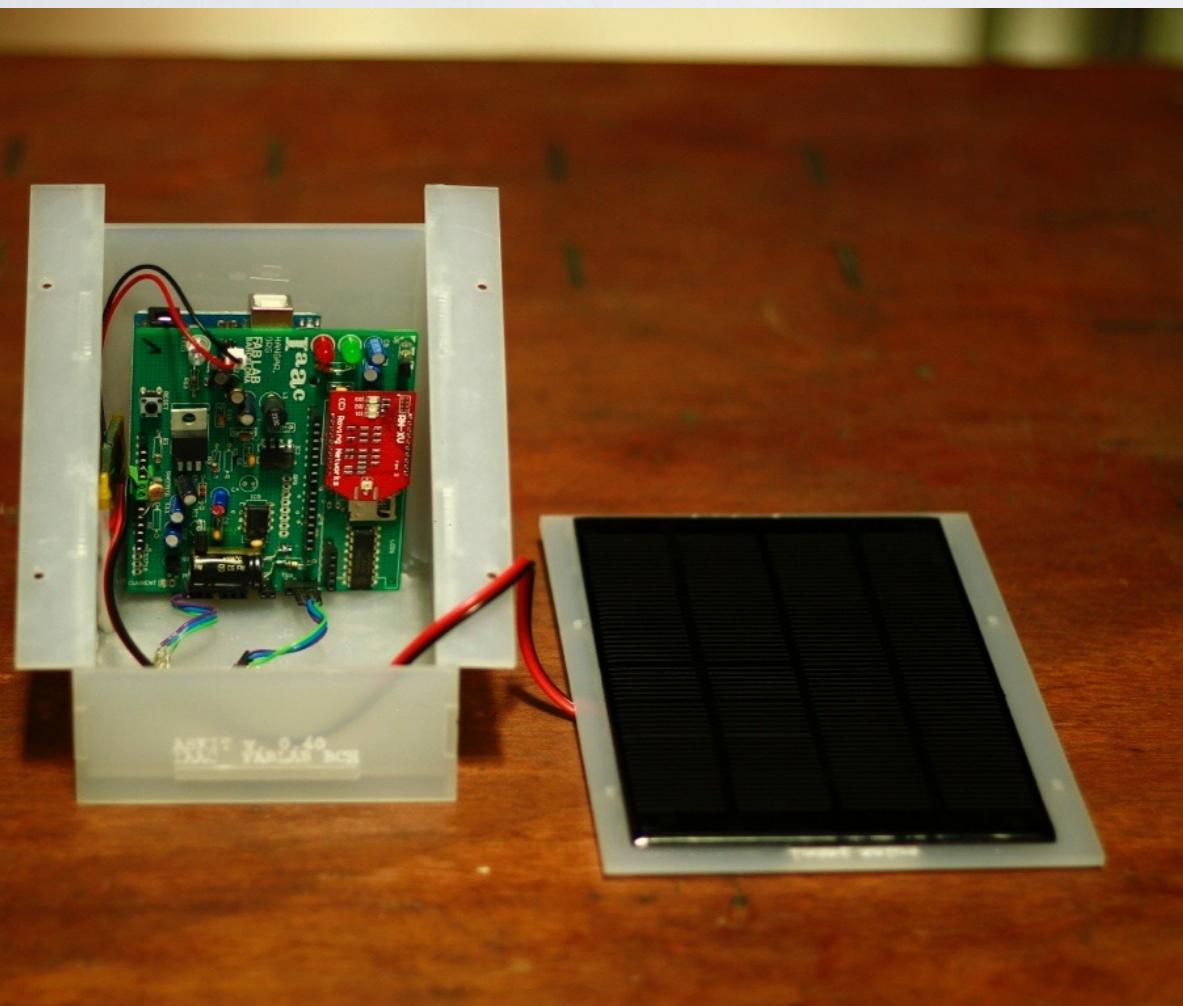


<https://www.youtube.com/watch?v=UoBUXOOdLXY&t=175s>

# What use(s) do Arduino boards have?

The most prominent use, however, is rapid prototyping in electronic projects.

Thanks to Arduino's **Open Source** licenses, we can remix the software/hardware and turn it into our own products! (**just don't call it Arduino**)



# Welcome!

## Class outline

- Overview of Arduino and hardware electronics [25min]
- **Q&A [5–10min]**
- Hands-on, instructor-led activities [60min]
- Break [5–10min]
- Hands-on, instructor-led activities [60min]
- Experimentation [if time permits]

# Welcome!

## Class outline

- Overview of Arduino and hardware electronics [25min]
- Q&A [5–10min]
- **Hands-on, instructor-led activities [60min]**
- Break [5–10min]
- Hands-on, instructor-led activities [60min]
- Experimentation [if time permits]

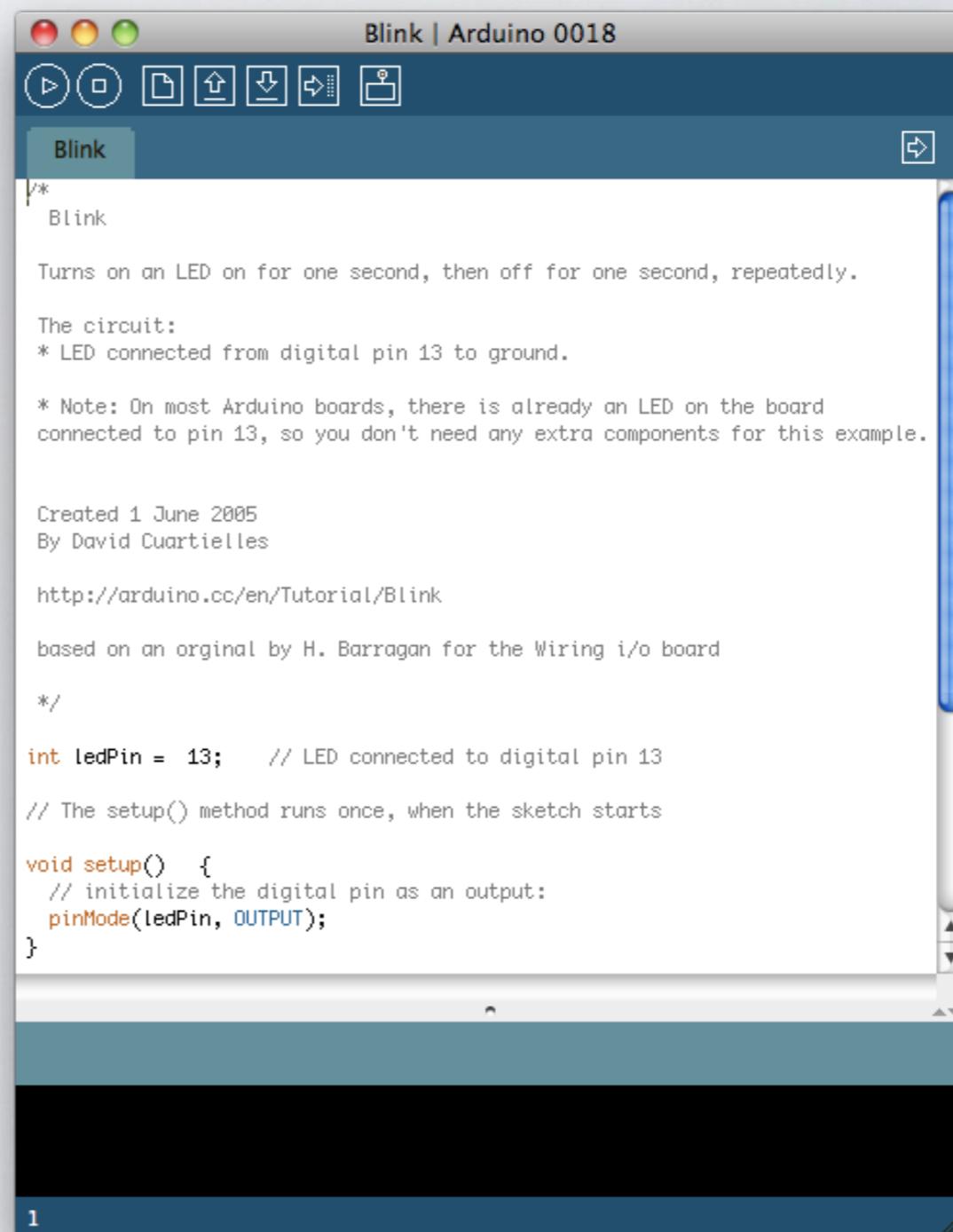
# Getting Started with Arduino

**Installing the Arduino IDE**—a computer application to edit, compile, and upload our programs, as well as communicate via USB with the Arduino boards

[Windows](#)

[Mac \(OSX 10.5+\)](#)

[Linux \(32-bit, 64-bit\)](#)



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 0018". The main window displays the "Blink" sketch. The code is as follows:

```
/*
 * Blink
 *
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * The circuit:
 * * LED connected from digital pin 13 to ground.
 *
 * Note: On most Arduino boards, there is already an LED on the board
 * connected to pin 13, so you don't need any extra components for this example.
 *
 * Created 1 June 2005
 * By David Cuartielles
 *
 * http://arduino.cc/en/Tutorial/Blink
 *
 * based on an original by H. Barragan for the Wiring i/o board
 */
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
```

# Getting Started with Arduino

**Downloading the code for this class**—all our code is free and Open-Source, so download it, use it, share it (when you make millions, remember us (-:)

<https://github.com/acrobotic/Ai Intro Arduino>

updating activity 6

1sc0 authored 7 hours ago latest commit d57a54cc39

activity_01	fixing some descriptions and file names	5 months ago
activity_02	updating	11 days ago
activity_03	fixing some descriptions and file names	5 months ago
activity_04	updating RGB LED tutorial	a month ago
activity_05	updating	14 hours ago
activity_06	updating activity 6	7 hours ago
activity_07	fixing some descriptions and file names	5 months ago
activity_08	fixing some descriptions and file names	5 months ago
activity_09	updating	14 hours ago
presentation	updating activity 6	7 hours ago

Wiki    Pulse    Graphs    Settings

HTTPS clone URL: <https://github.com/>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

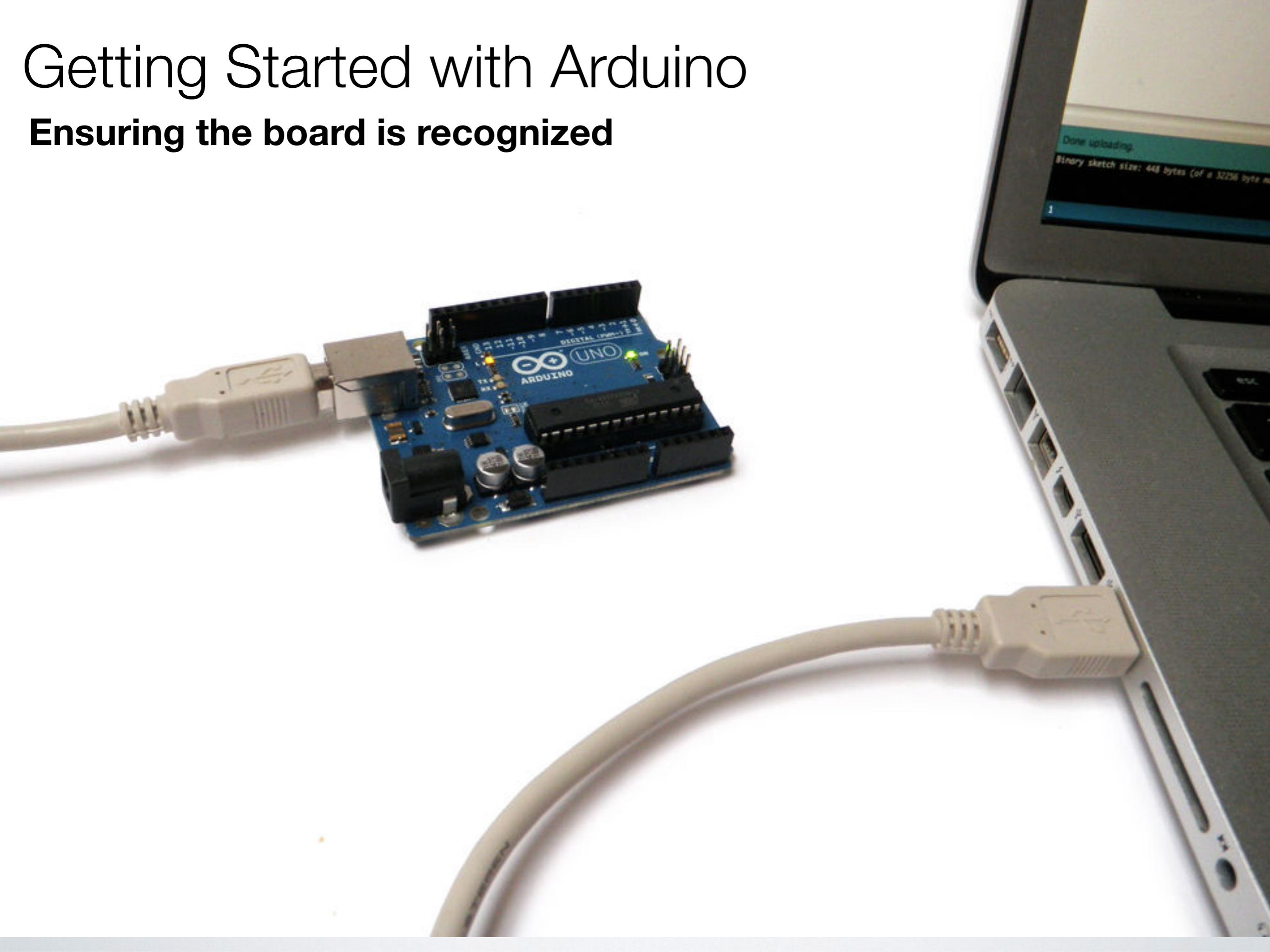
Clone in Desktop

Download ZIP

Help people interested in this repository understand your project by adding a README! [Add a README](#)

# Getting Started with Arduino

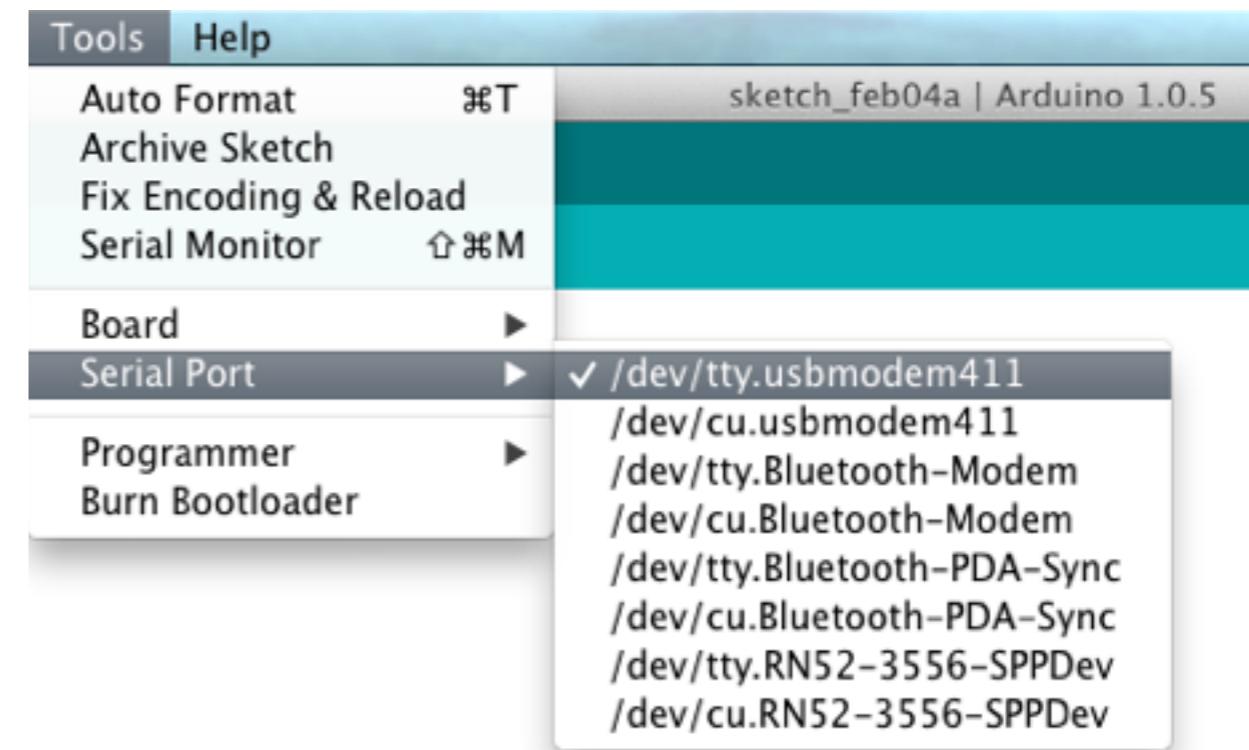
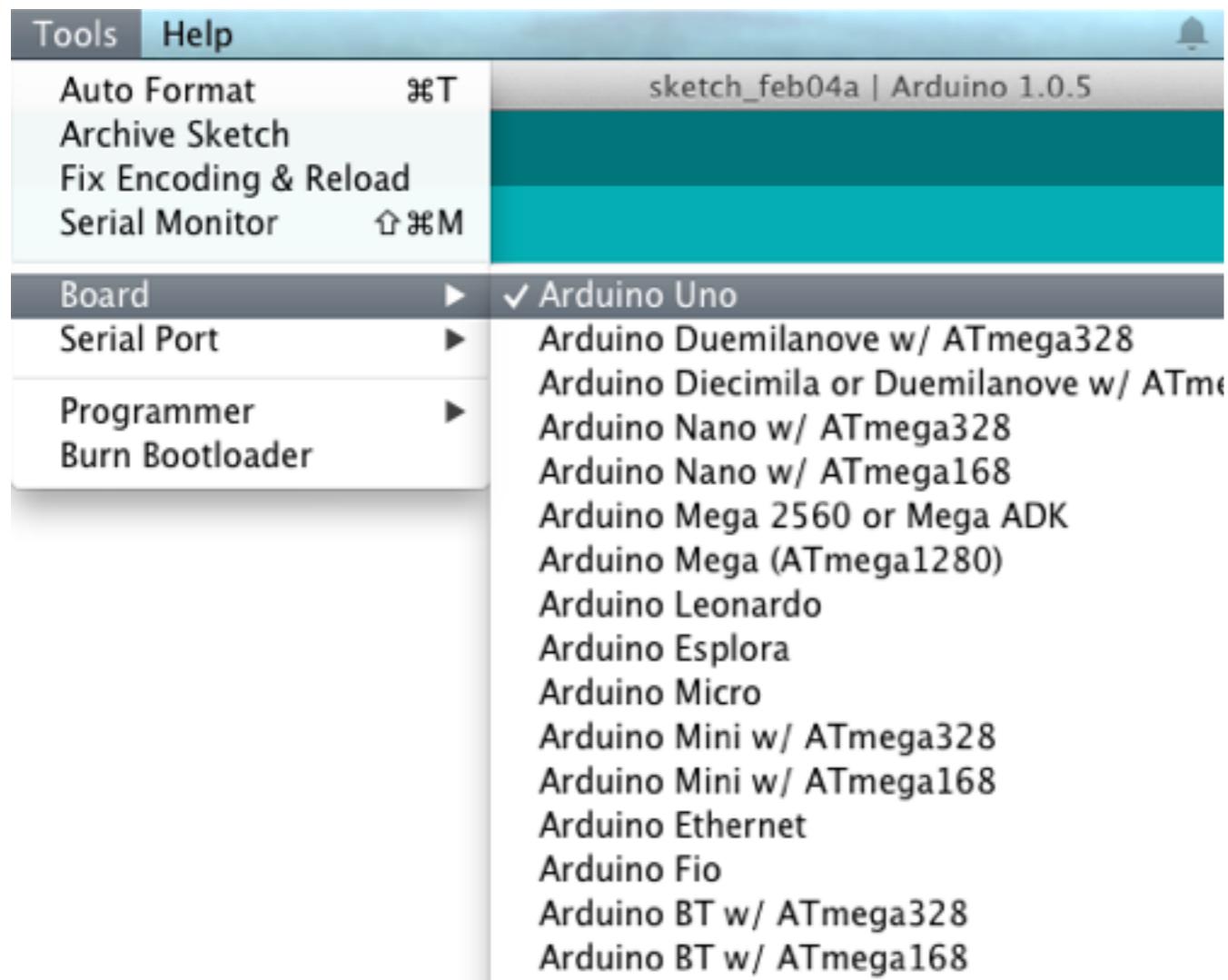
## Ensuring the board is recognized



# Getting Started with Arduino

## Ensuring the board is recognized

Using the Arduino IDE menu options, select the Board and Serial Port for the connected device.



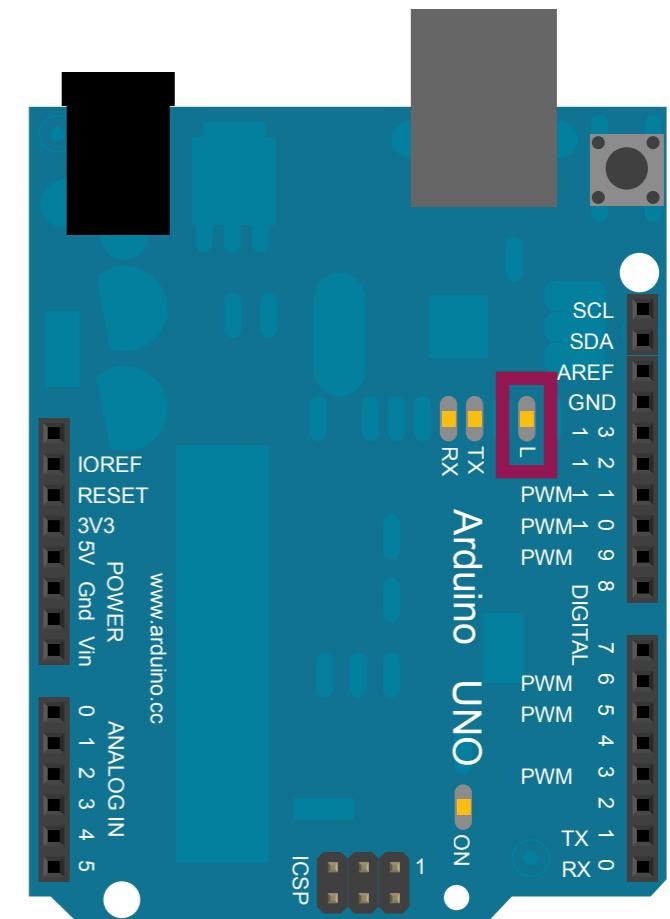
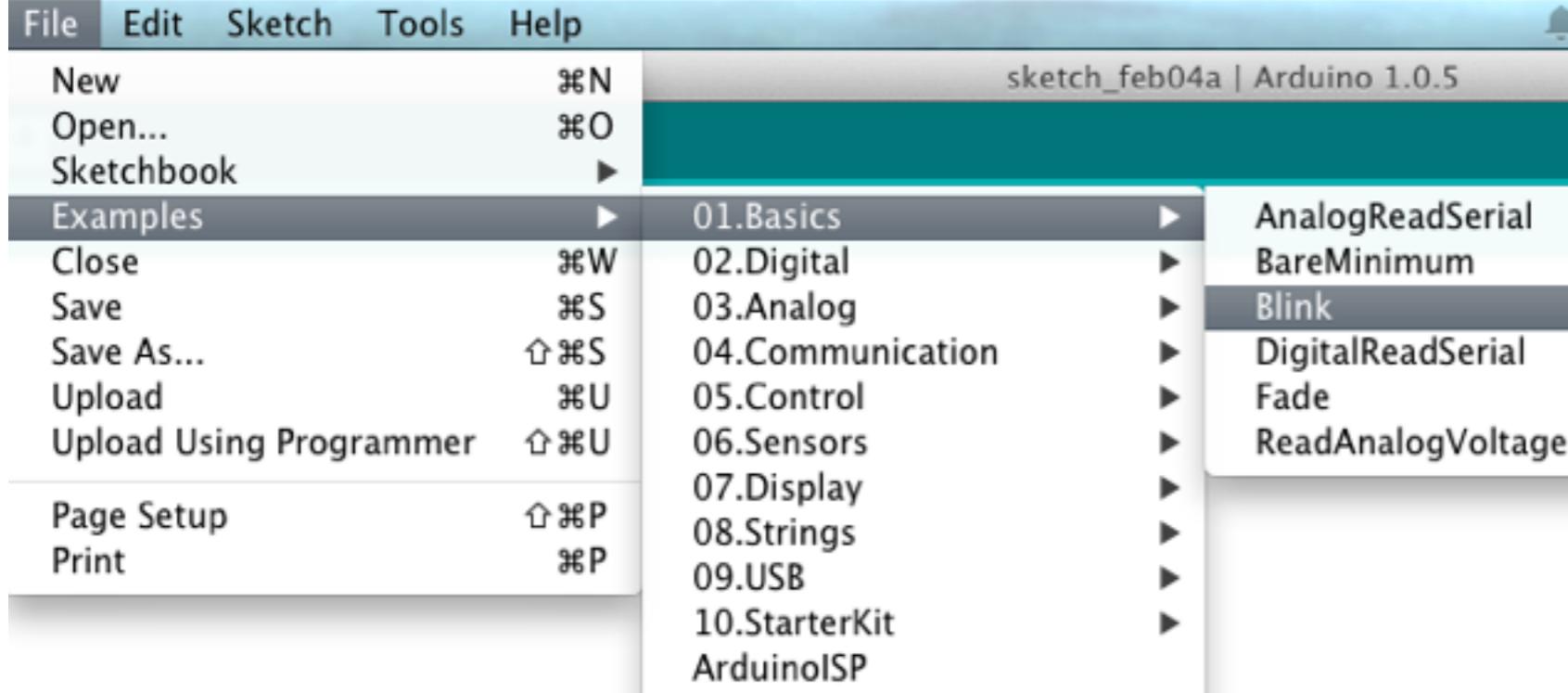
Windows users might need to install/update the board driver:

<http://learn.acrobotic.com/tutorials/intro-to-electronics-with-arduino#step-3>

# Getting Started with Arduino

## Running our first program!

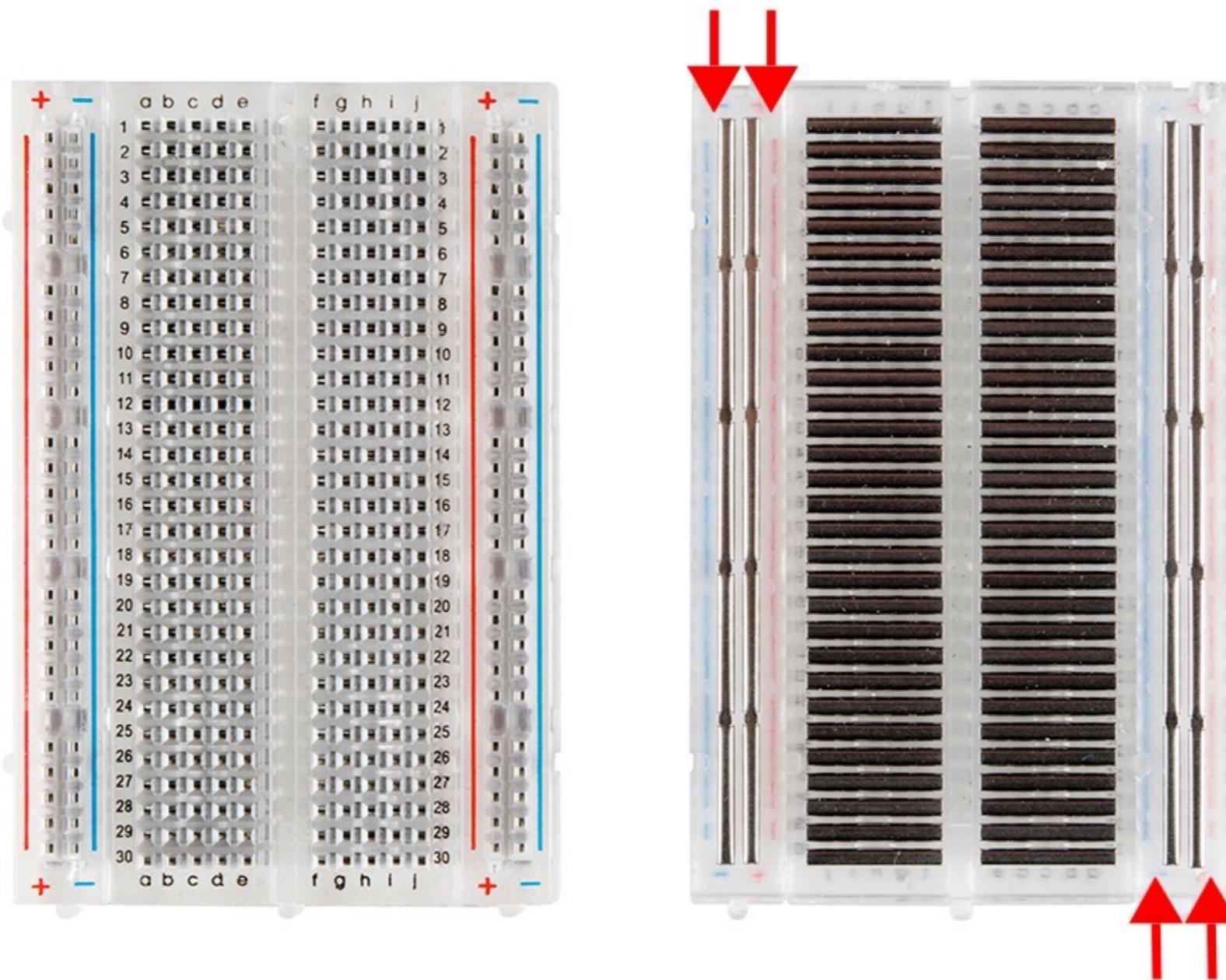
Using the Arduino IDE menu options, select the Blink example, compile it (optional), and upload it to the board.



If everything goes well, the yellow LED by digital pin 13 should start 'blinking'.

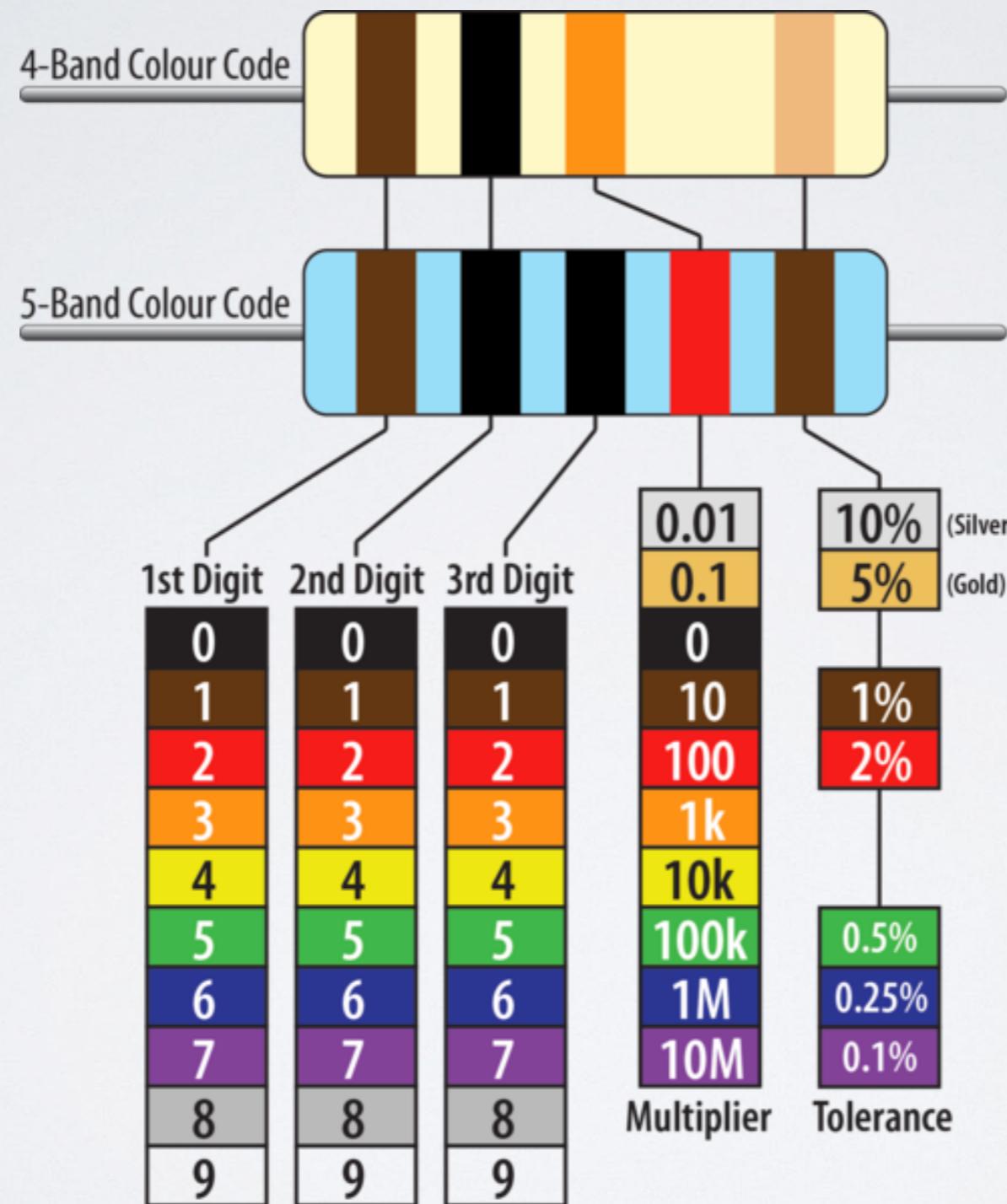
# Getting Started with Arduino

## How the solderless breadboard works:



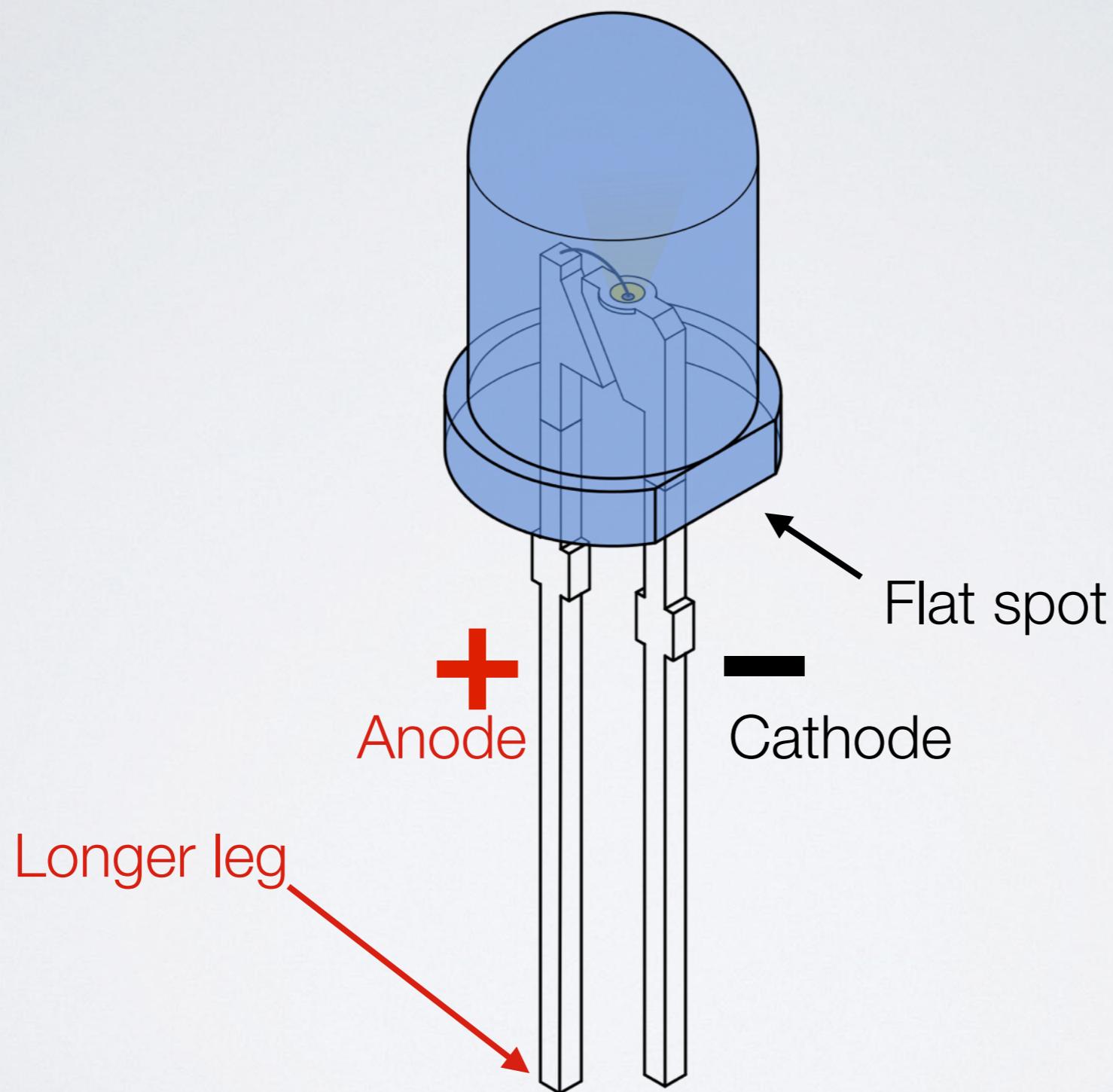
# Getting Started with Arduino

## Reading Resistor Codes:



# Getting Started with Arduino

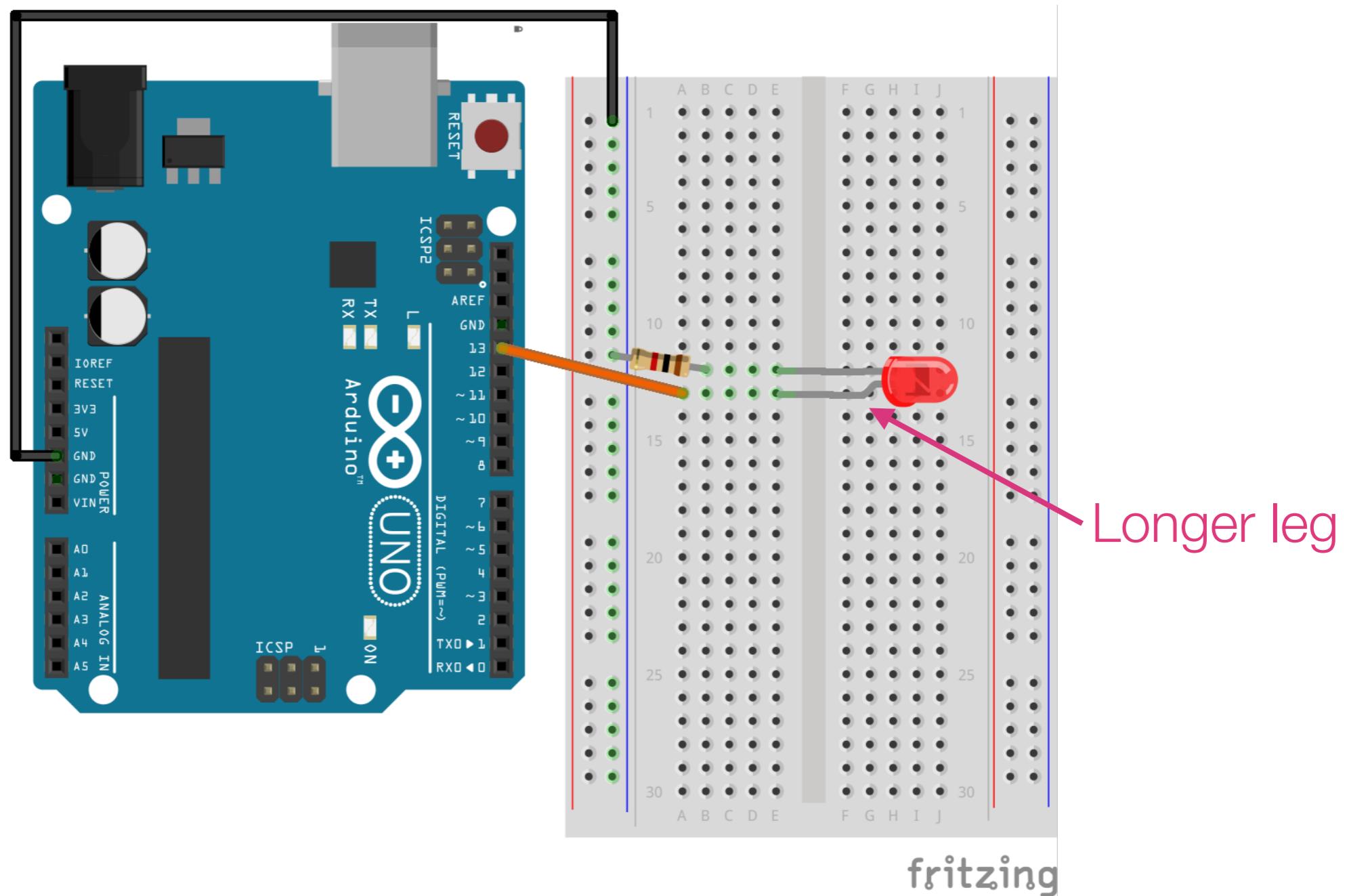
## Reading LED polarity:



# Getting Started with Arduino

## Activity 2: Digital Outputs [blink\_external.ino]

What happens when we connect an LED to pin 13?



# Getting Started with Arduino

## Anatomy of an Arduino Program ('sketch'):

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

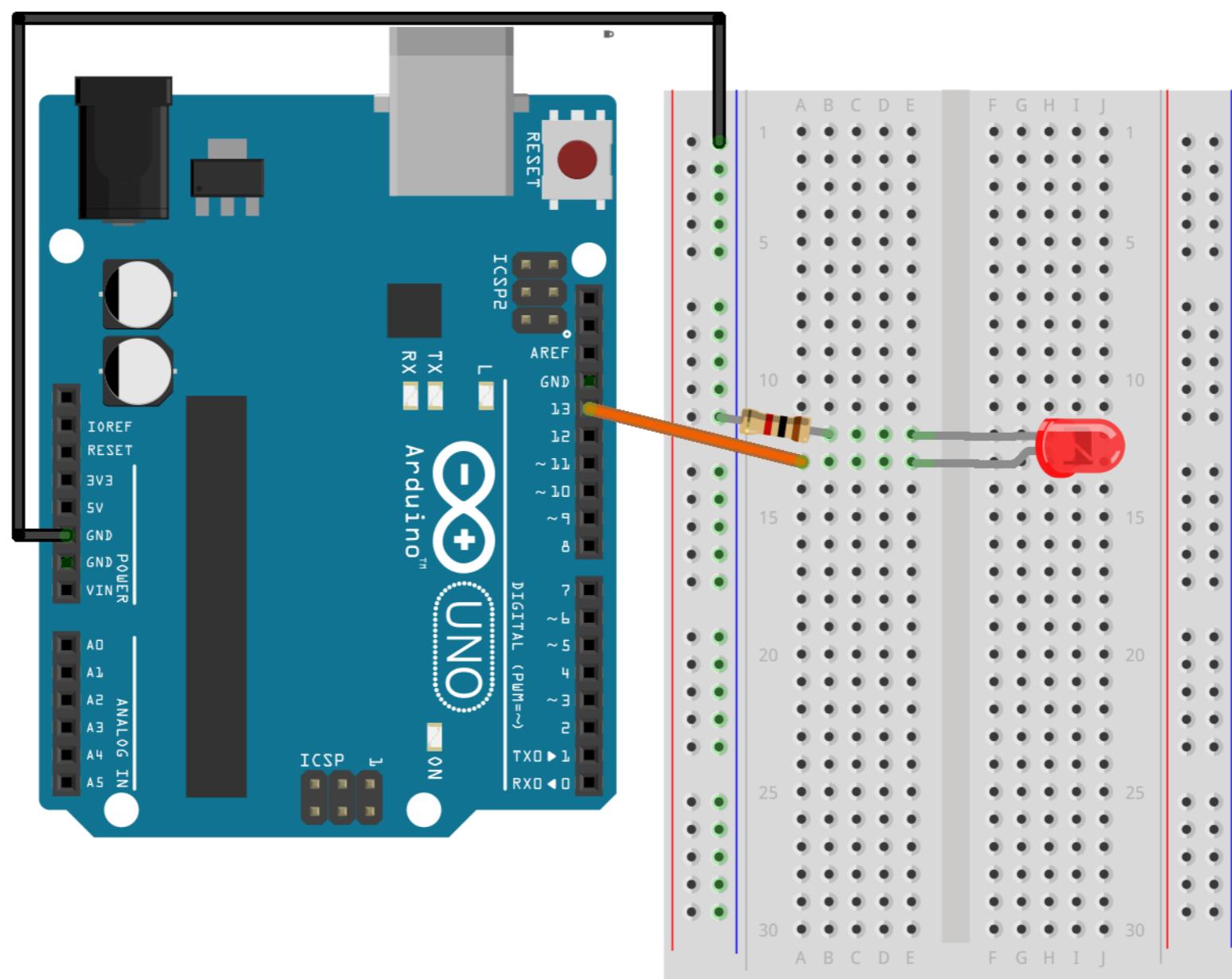
// the setup routine runs once when you press reset:
void setup() {
    // initialize the digital pin as an output.
    pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                  // wait for a second
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                  // wait for a second
}
```

# Getting Started with Arduino

## Activity 2: Pulse Width Modulation (PWM) [blink\_compare.ino]

What happens when we decrease the on/off delay ( $T_{min}$ )?

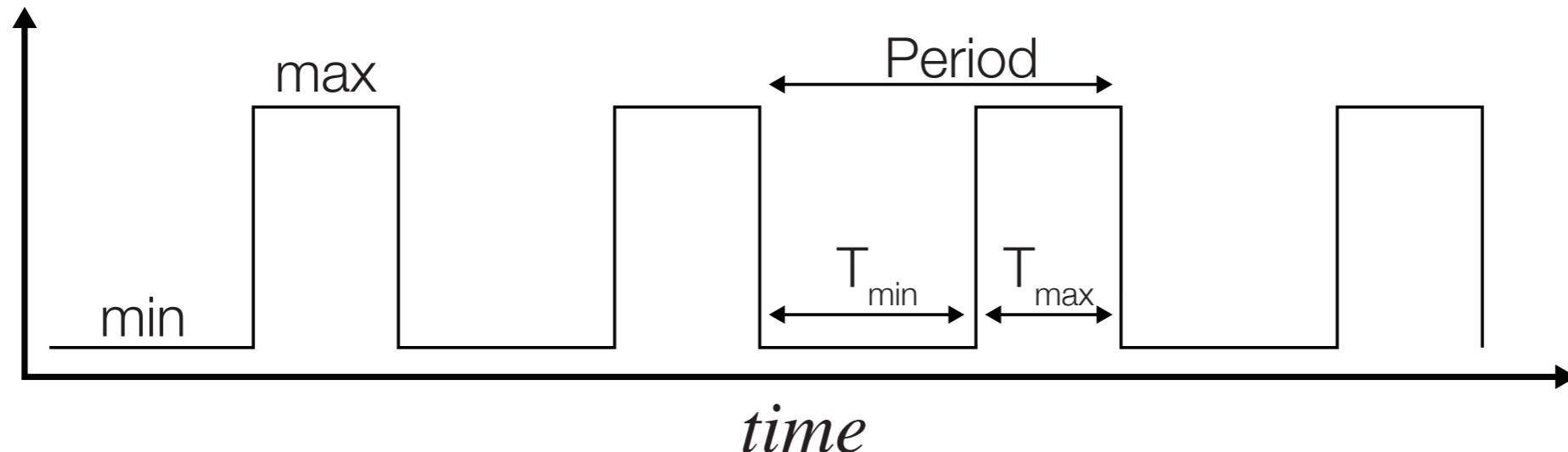


fritzing

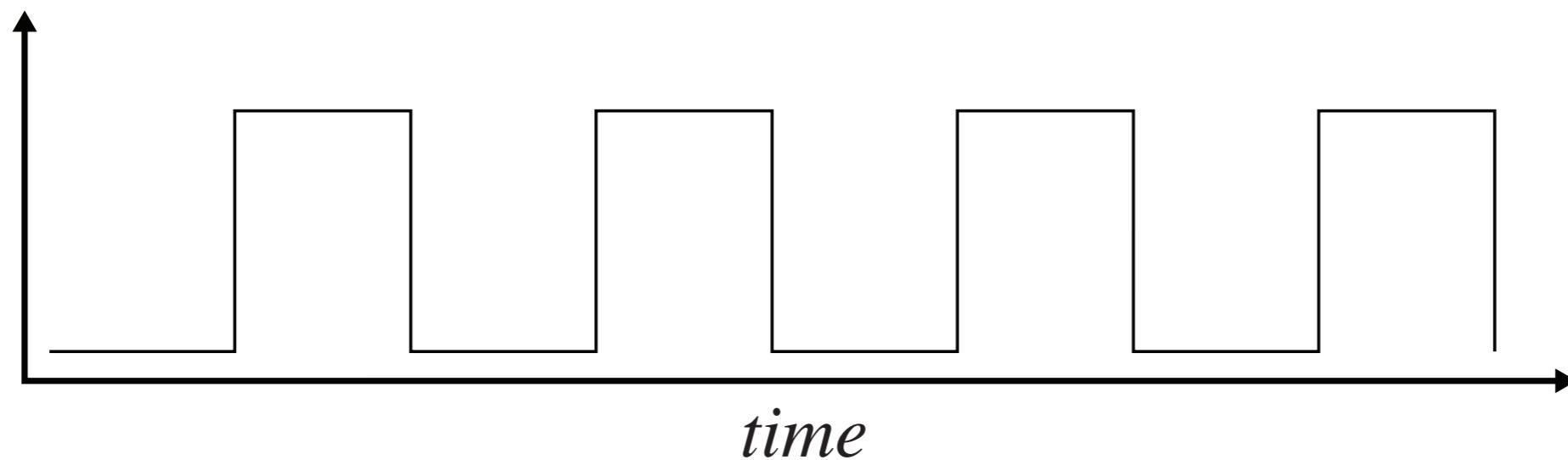
# Getting Started with Arduino

## Activity 2: Pulse Width Modulation (PWM)

Rectangular or Pulse Wave



Special case: Square Wave ( $T_{\text{min}} = T_{\text{max}}$ )



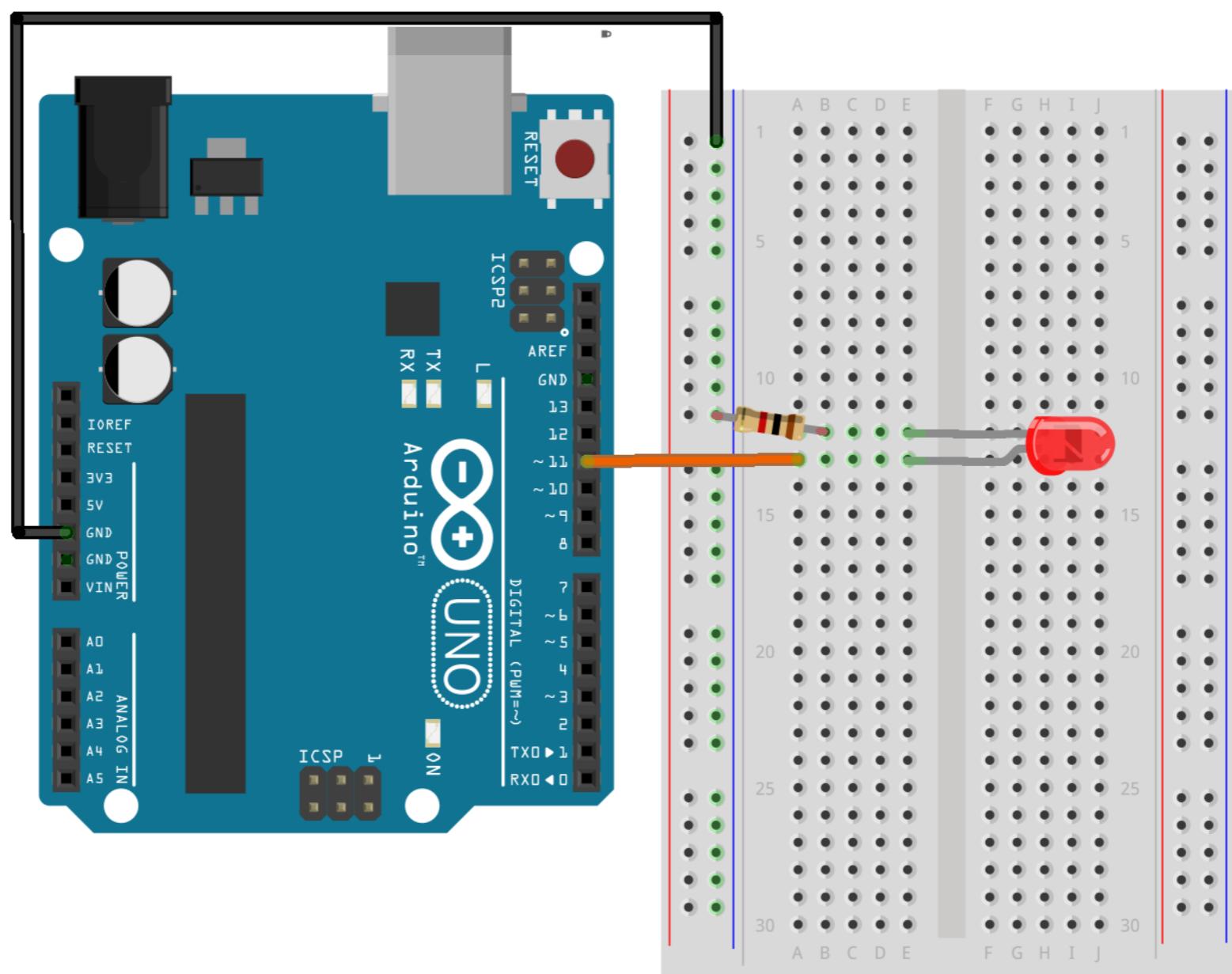
$$\text{Frequency} = 1/\text{Period}$$

$$\text{Duty Cycle} = 100\% \times T_{\text{max}} / (T_{\text{max}} + T_{\text{min}})$$

# Getting Started with Arduino

## Activity 2: Pulse Width Modulation (PWM)

What happens when we modulate the on/off delay ( $T_{min}$ )? [heartbeat.ino]

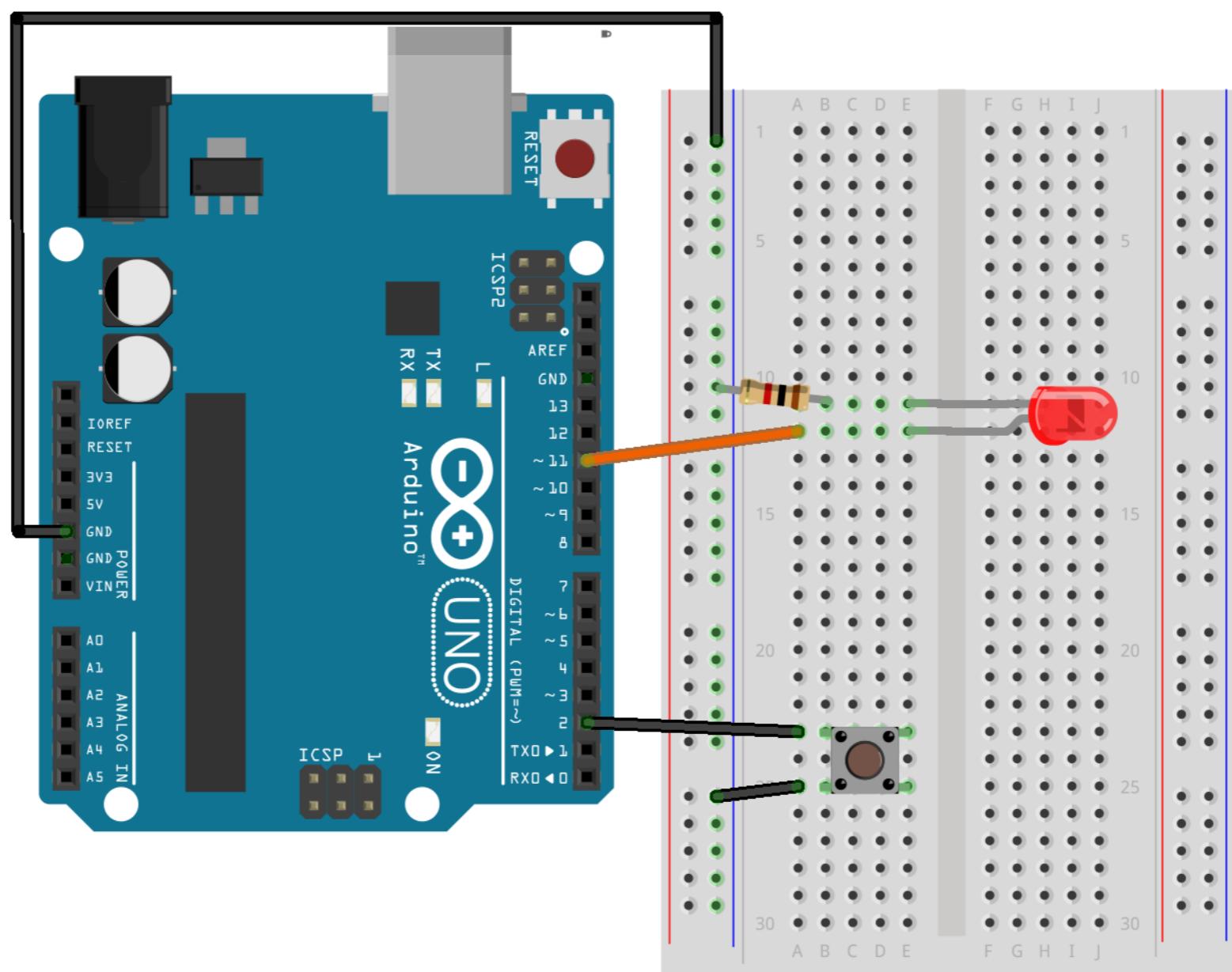


fritzing

# Getting Started with Arduino

## Activity 3: Digital Inputs [button\_hold.ino]

Using the `digitalRead()` function, we can determine the Open/Closed state of the button.

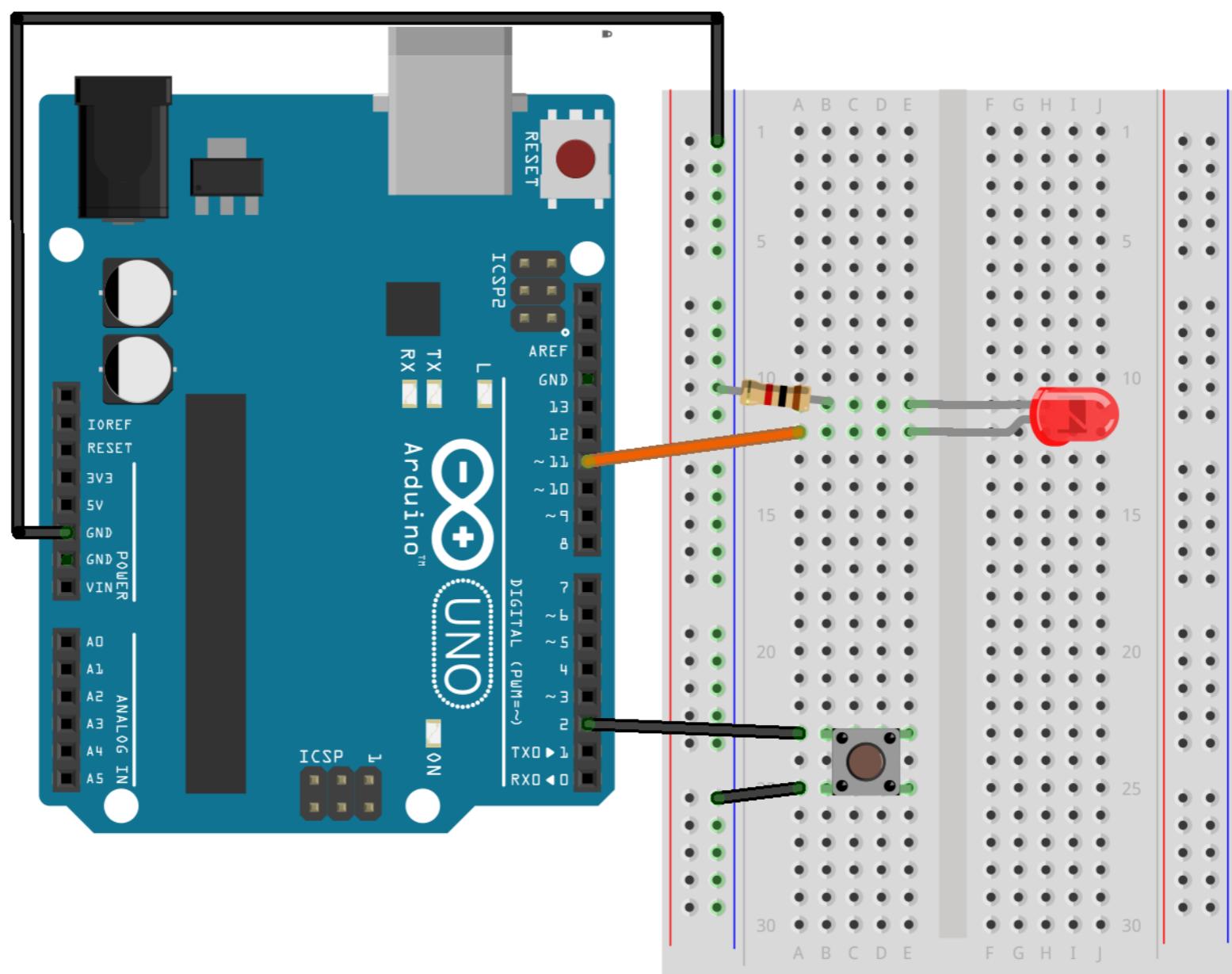


fritzing

# Getting Started with Arduino

## Activity 3: Digital Inputs [button\_toggle.ino]

Adding the `Serial.print('...')` line allows us to debug the code.



fritzing

# Welcome!

## Class outline

- Overview of Arduino and hardware electronics [25min]
- Q&A [5–10min]
- Hands-on, instructor-led activities [60min]
- **Break [5–10min]**
- Hands-on, instructor-led activities [60min]
- Experimentation [if time permits]

# Welcome!

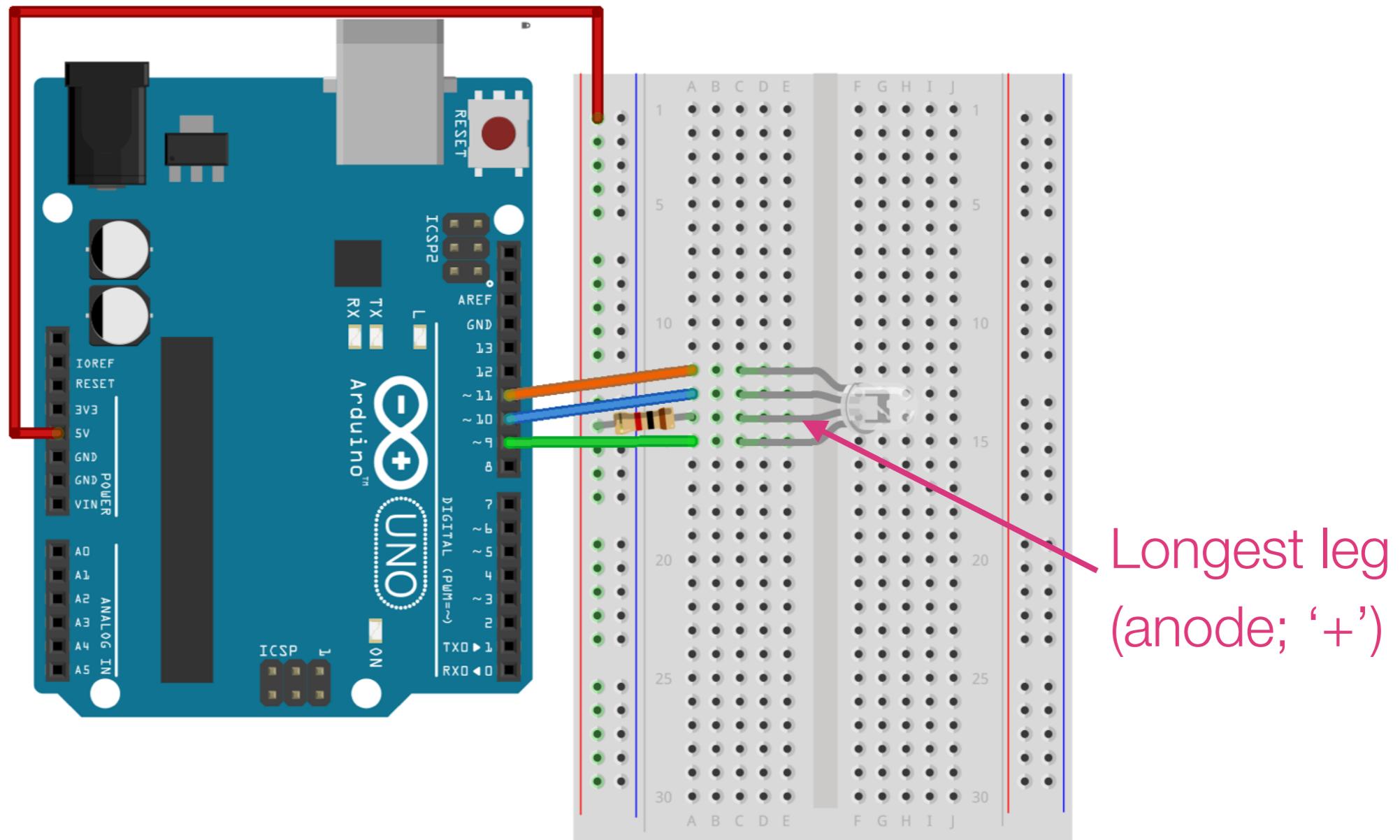
## Class outline

- Overview of Arduino and hardware electronics [25min]
- Q&A [5–10min]
- Hands-on, instructor-led activities [60min]
- Break [5–10min]
- **Hands-on, instructor-led activities [60min]**
- Experimentation [if time permits]

# Getting Started with Arduino

## Activity 4: Multiple Digital Outputs [rgb\_led\_cycle.ino]

Colored light can be obtained by mixing 2 or more LEDs!



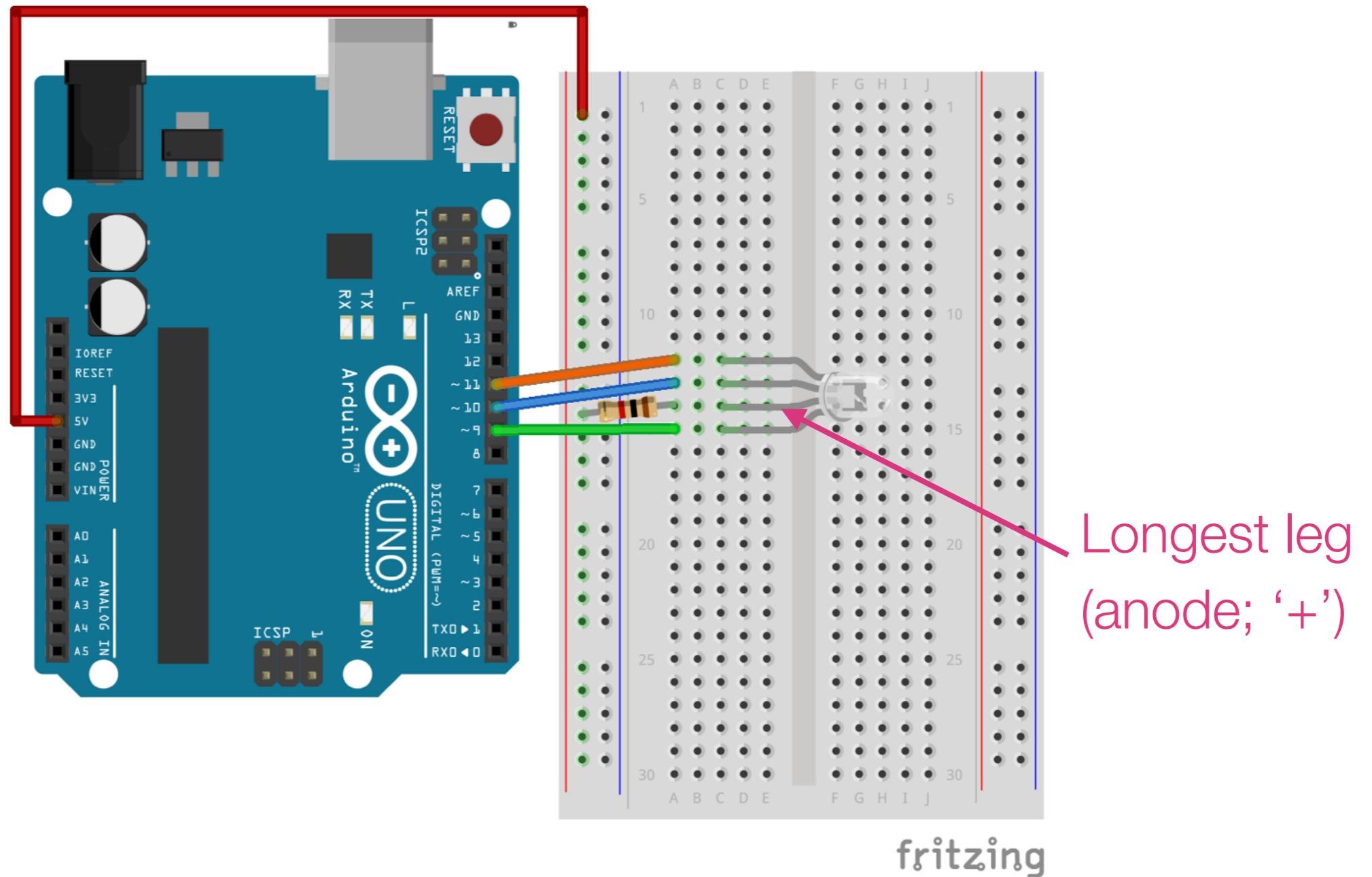
fritzing

Note: double-check the wiring so that when `rgb_led_cycle.ino` runs the blue LED lights-up for 2-seconds, turns-off, and then the red LED does the same.

# Getting Started with Arduino

## Activity 4: Multiple Digital Outputs [rgb\_led\_fade.ino]

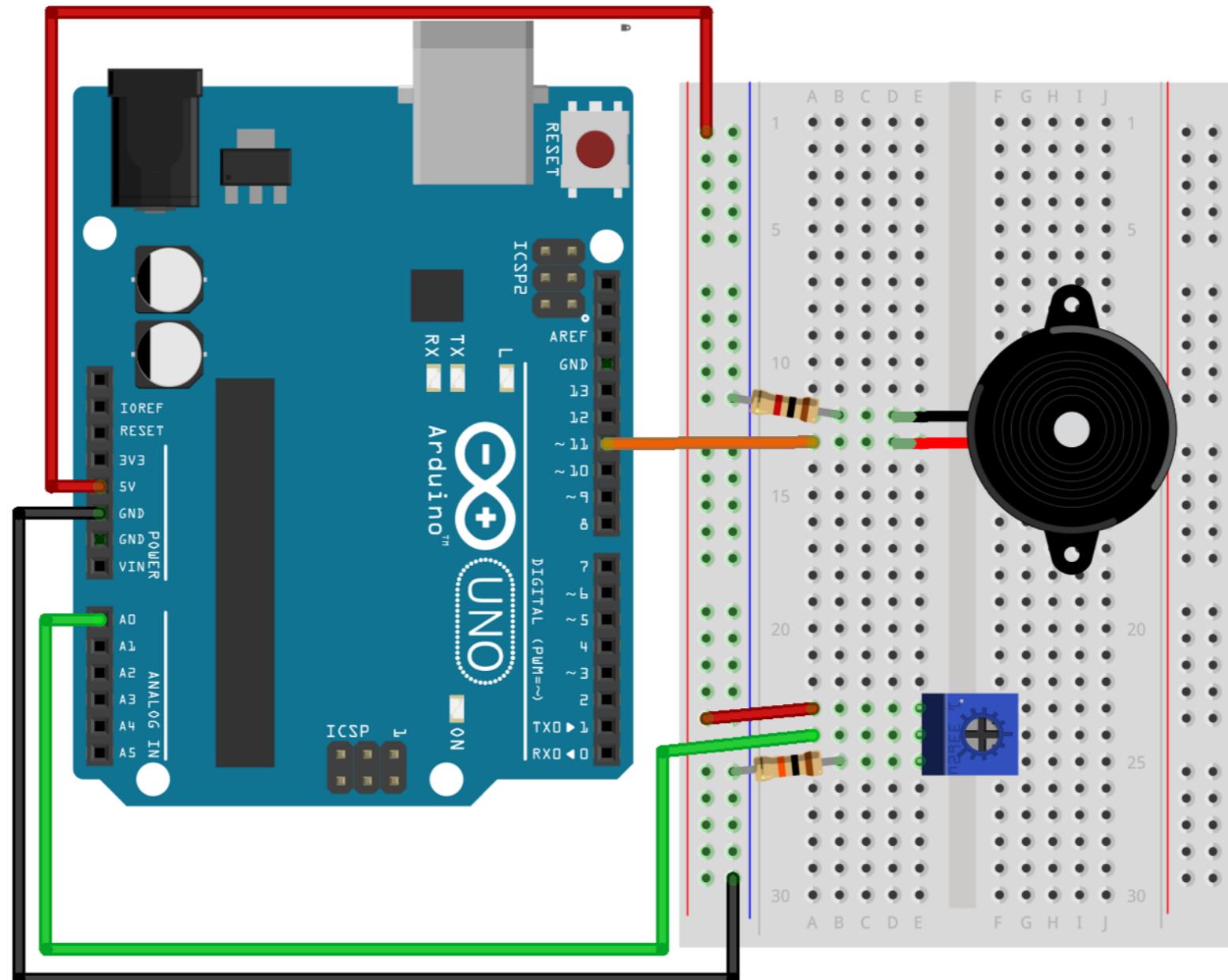
We have  $2^{24}$  possible color combinations as each LEDs accept a brightness level between 0 and 255.



# Getting Started with Arduino

## Activity 6: Analog Input [analog\_input\_vol.ino]

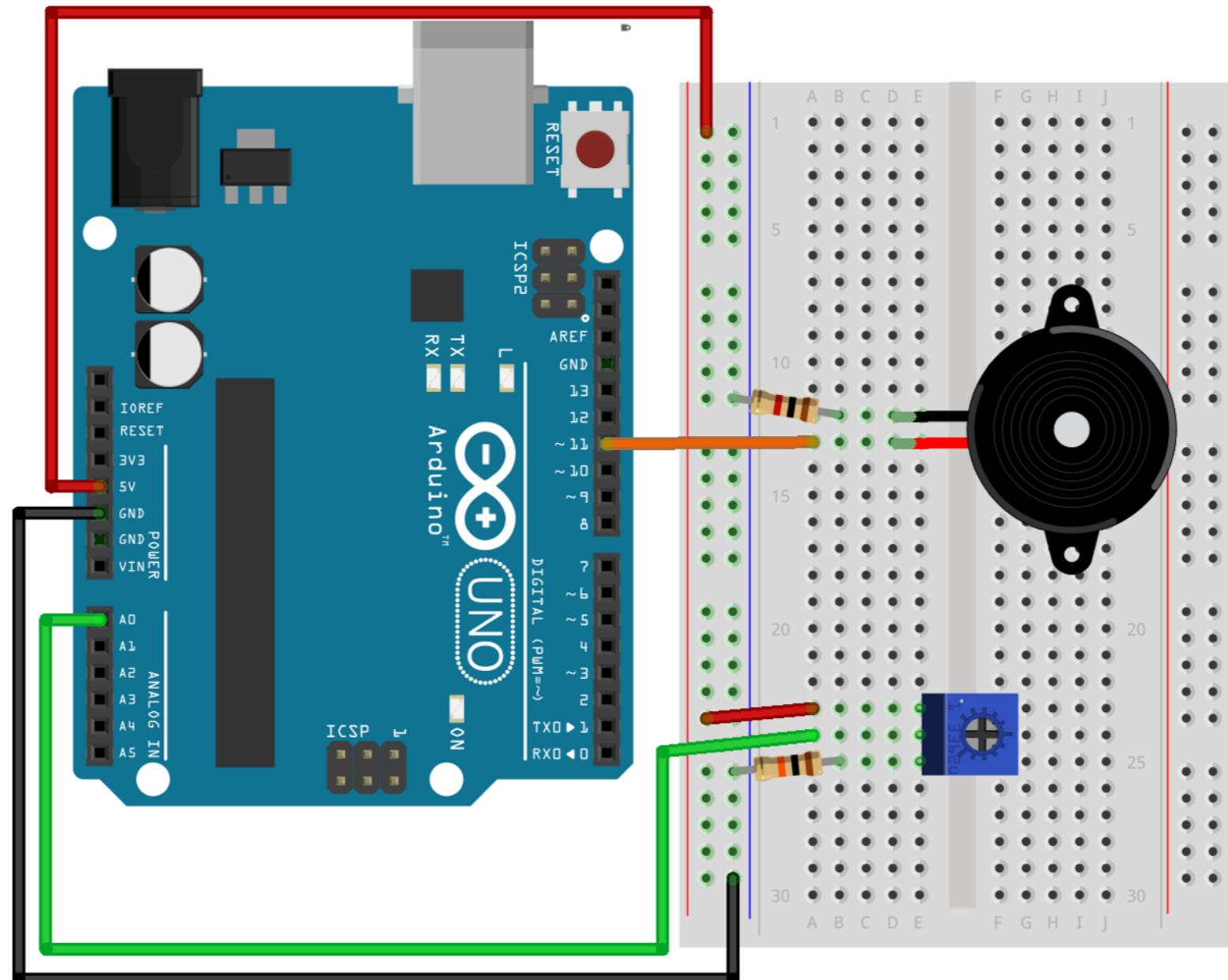
The function `analogRead(...)` uses the built-in Analog to Digital Converter to measure the analog voltage on the pot's sweeper pin.



# Getting Started with Arduino

## Activity 6: Analog Input [analog\_input\_freq.ino]

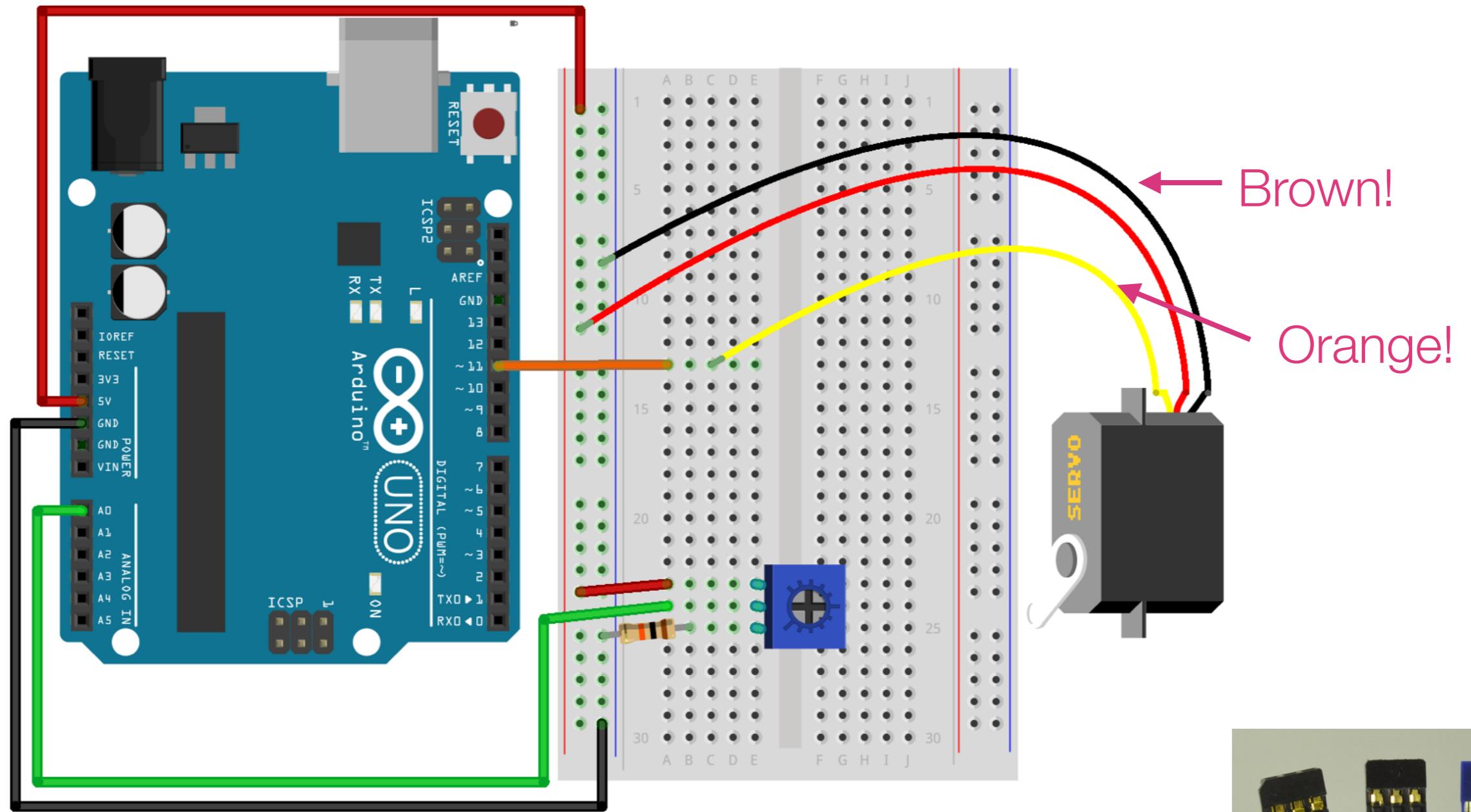
The function `tone(...)` changes the PWM frequency while keeping a 50% duty cycle!



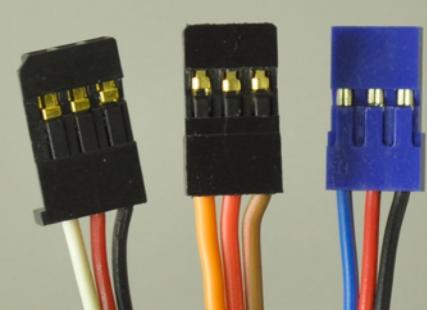
# Getting Started with Arduino

## Activity 7: Motor Control [servo\_sweep.ino]

By making use of the built-in Arduino Libraries, we don't have to re-invent the wheel!



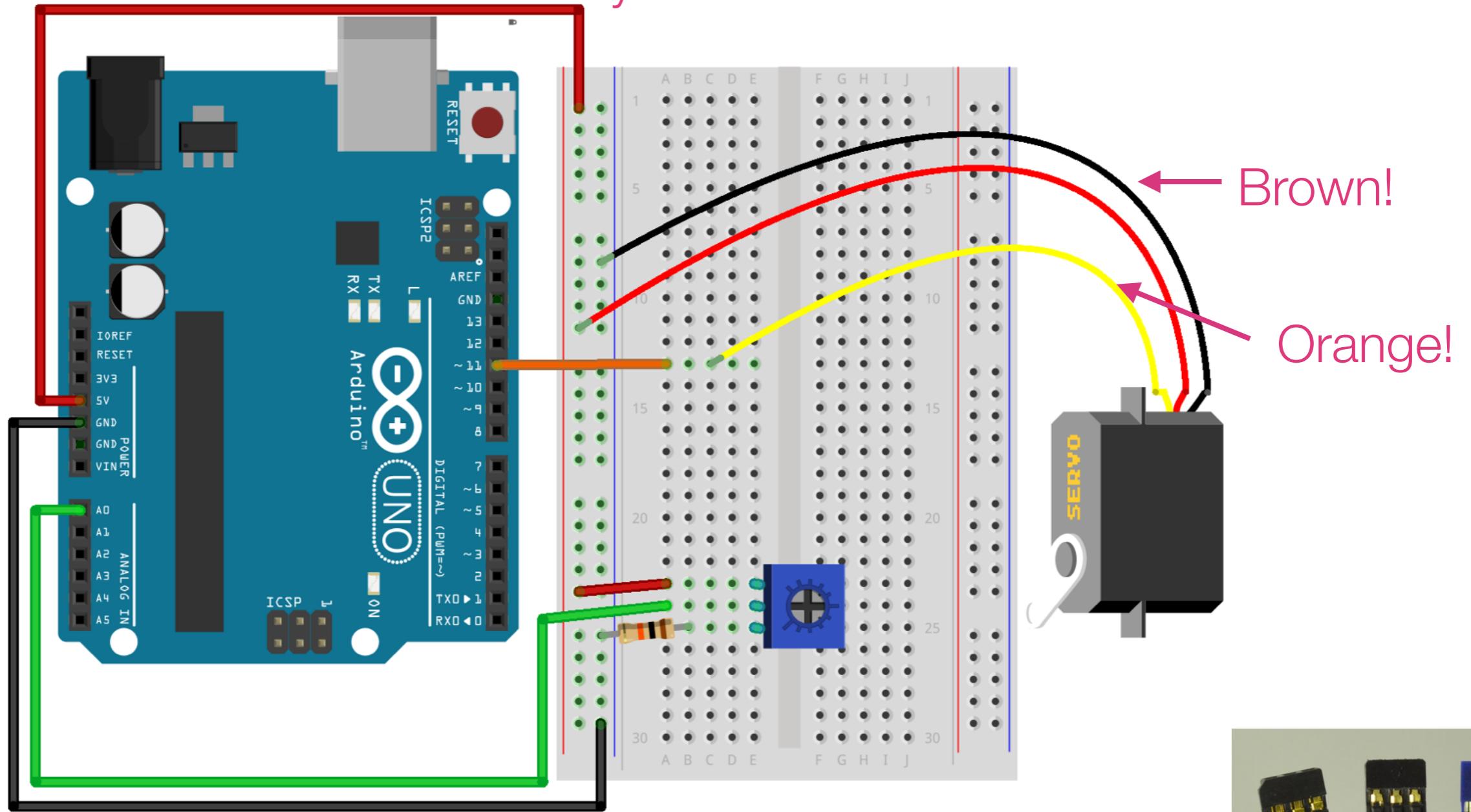
fritzing



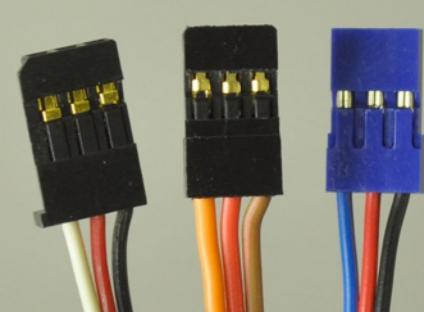
# Getting Started with Arduino

## Activity 7: Motor Control [servo\_knob.ino]

We don't have to worry about frequency or duty cycle of the PWM signal; it is taken care of within the Servo library!



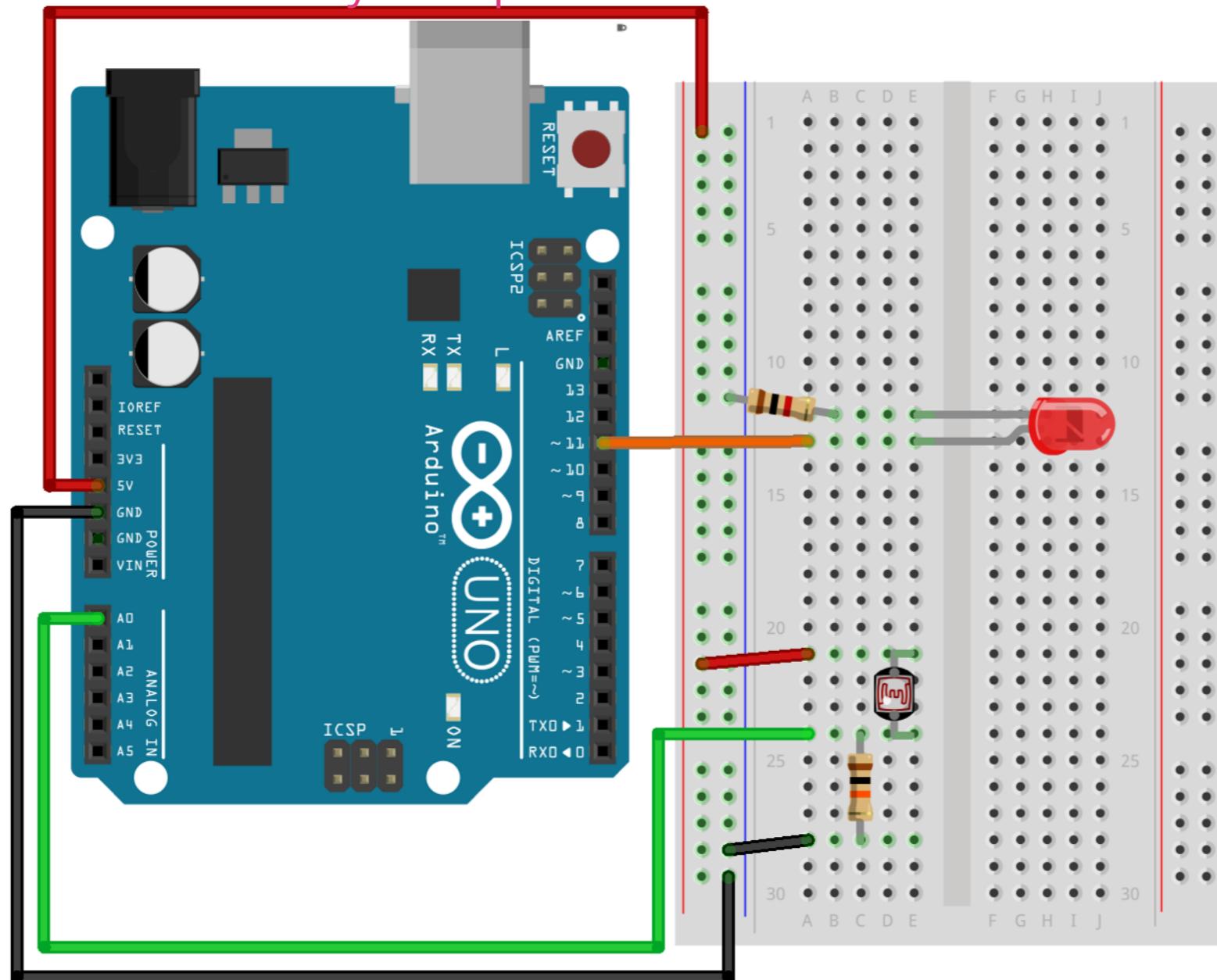
fritzing



# Getting Started with Arduino

## Activity 8: Light Sensor [photoresistor.ino]

Repurposing the code in activity 5 we can control LED brightness based on the light measured by the photoresistor!

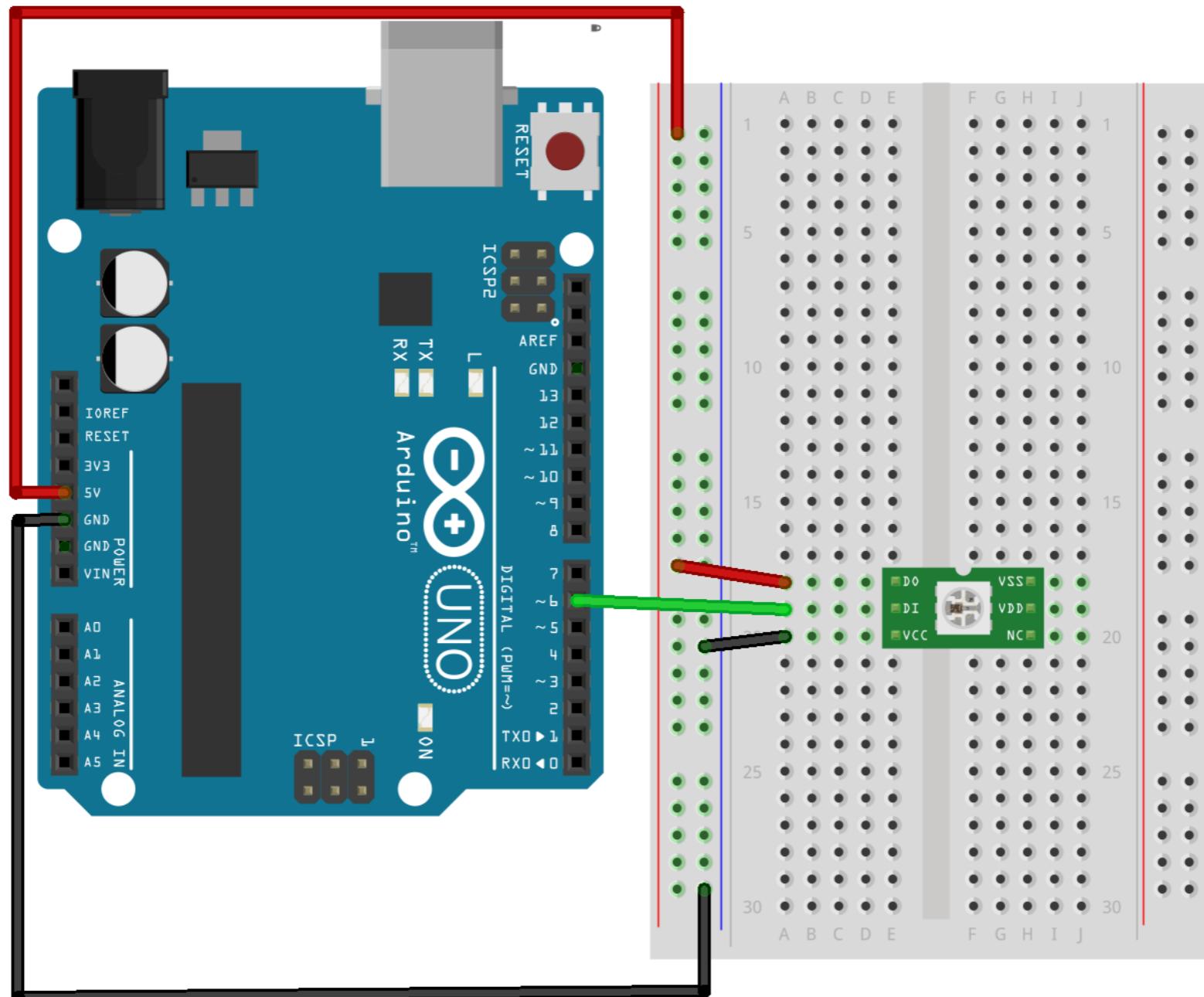


fritzing

# Getting Started with Arduino

## Activity 9: Smart RGB LED [ws2812.ino]

We can use additional libraries that are not built-in the Arduino IDE. Copy them to the libraries folder, restart the IDE, and voilà!

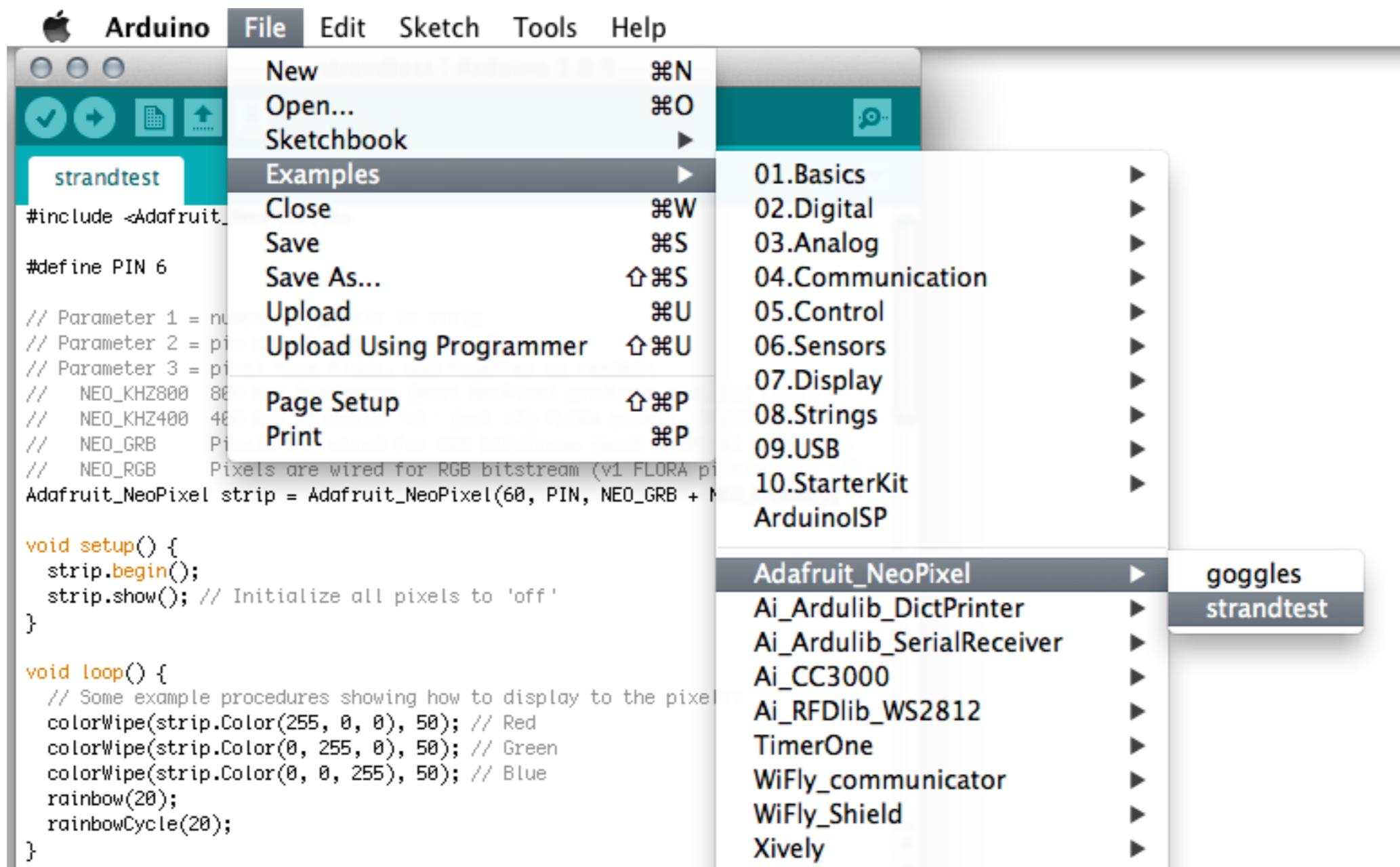


# Getting Started with Arduino

## Activity 9: Smart RGB LED [File>Examples>...]

Mac OS: /Users/username/Documents/Arduino/libraries/

Windows: C:\Users\xisco\Documents\Arduino\libraries\



# Welcome!

## Class outline

- Overview of Arduino and hardware electronics [25min]
- Q&A [5–10min]
- Hands-on, instructor-led activities [60min]
- Break [5–10min]
- Hands-on, instructor-led activities [60min]
- **Experimentation [if time permits]**

Thank You!