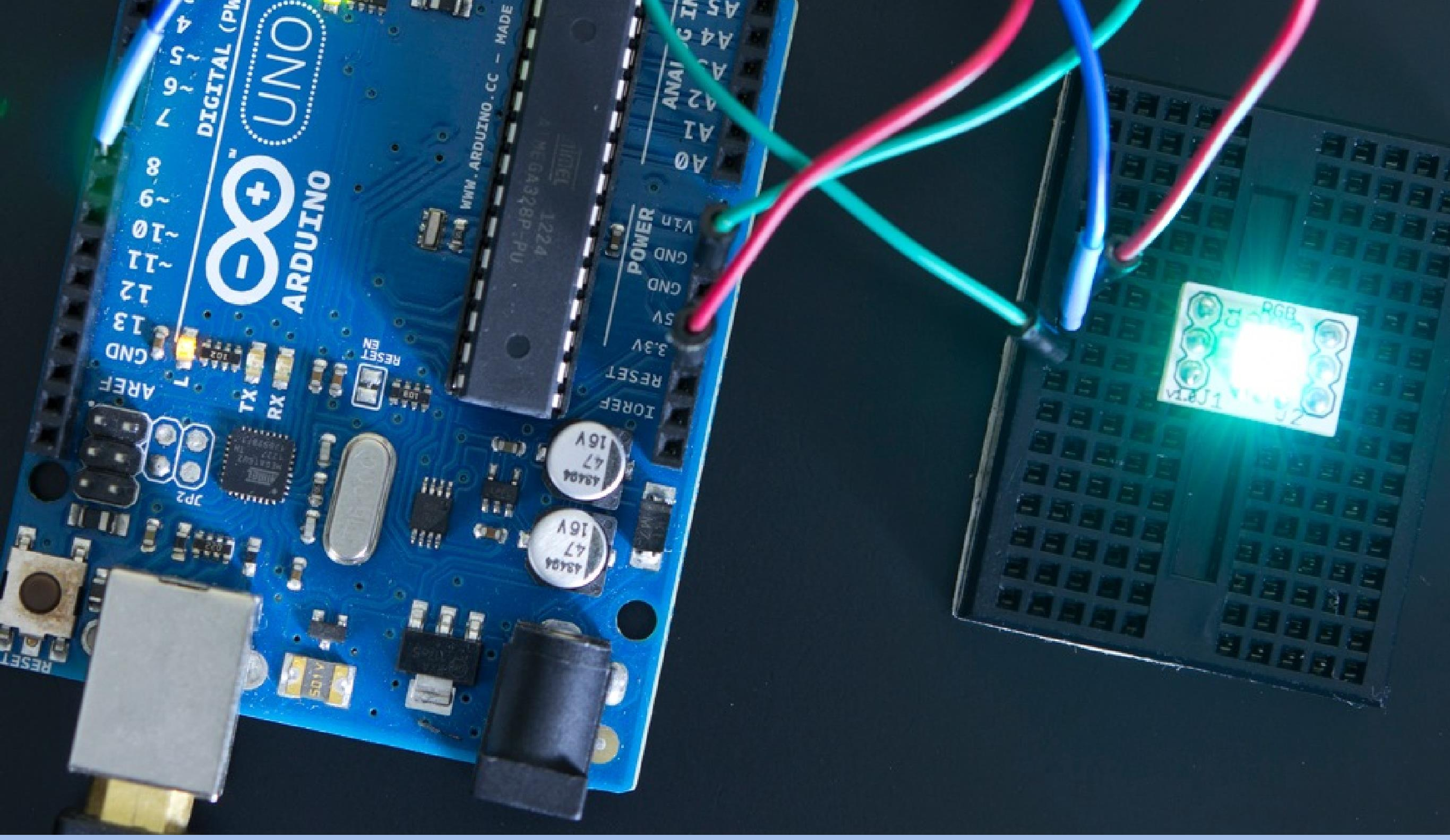


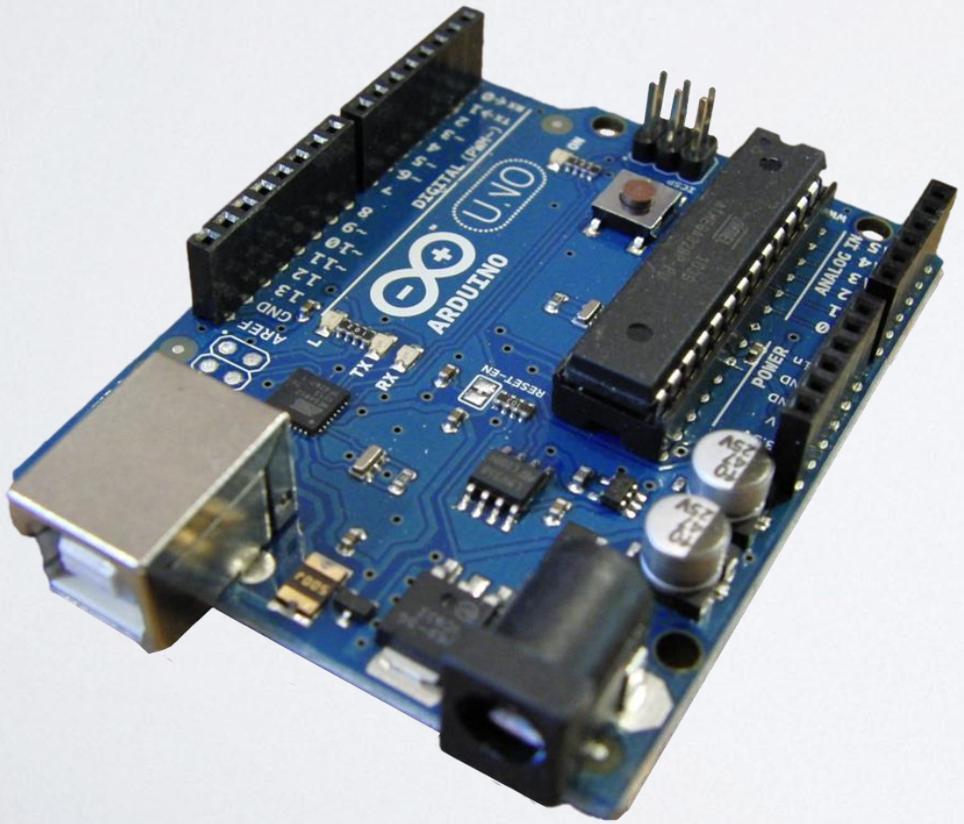
DSC IOT TRACK-1 : Intro to Arduino

What is Arduino?

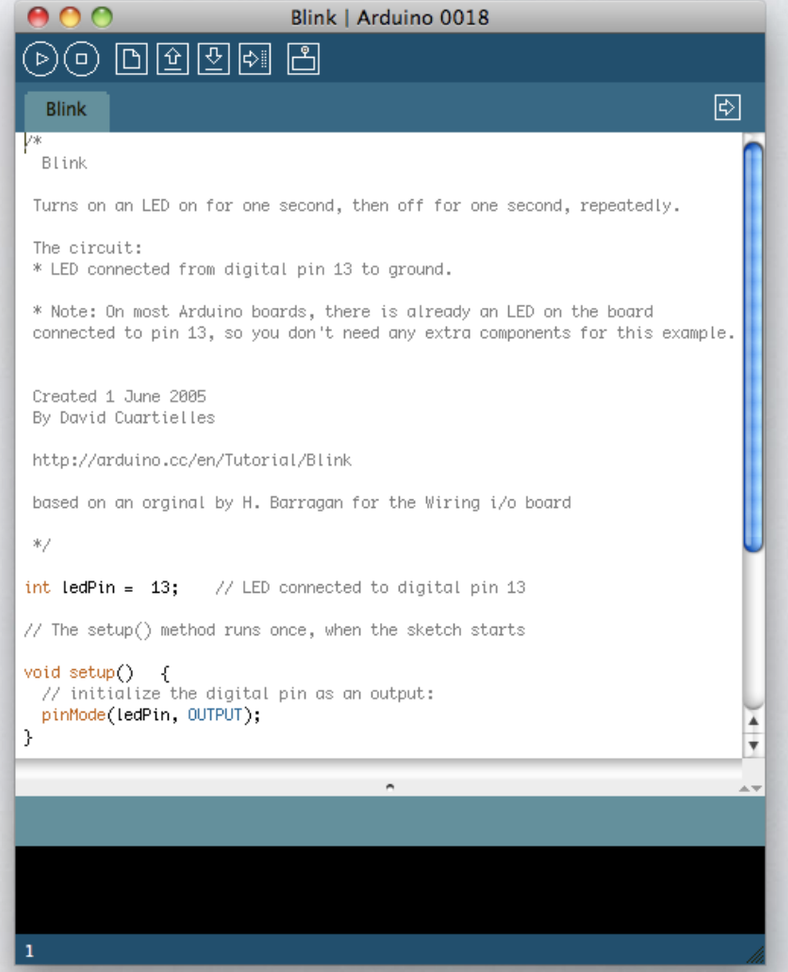
Arduino is the *brand* of a toolset that enables anyone to interact with the physical world (buzzword bingo: *physical computing*).

Physical world interactions occur through analog and digital inputs and outputs.

The **Arduino** toolset is comprised by both hardware and software, that allow us to perform these interactions with ease.



+



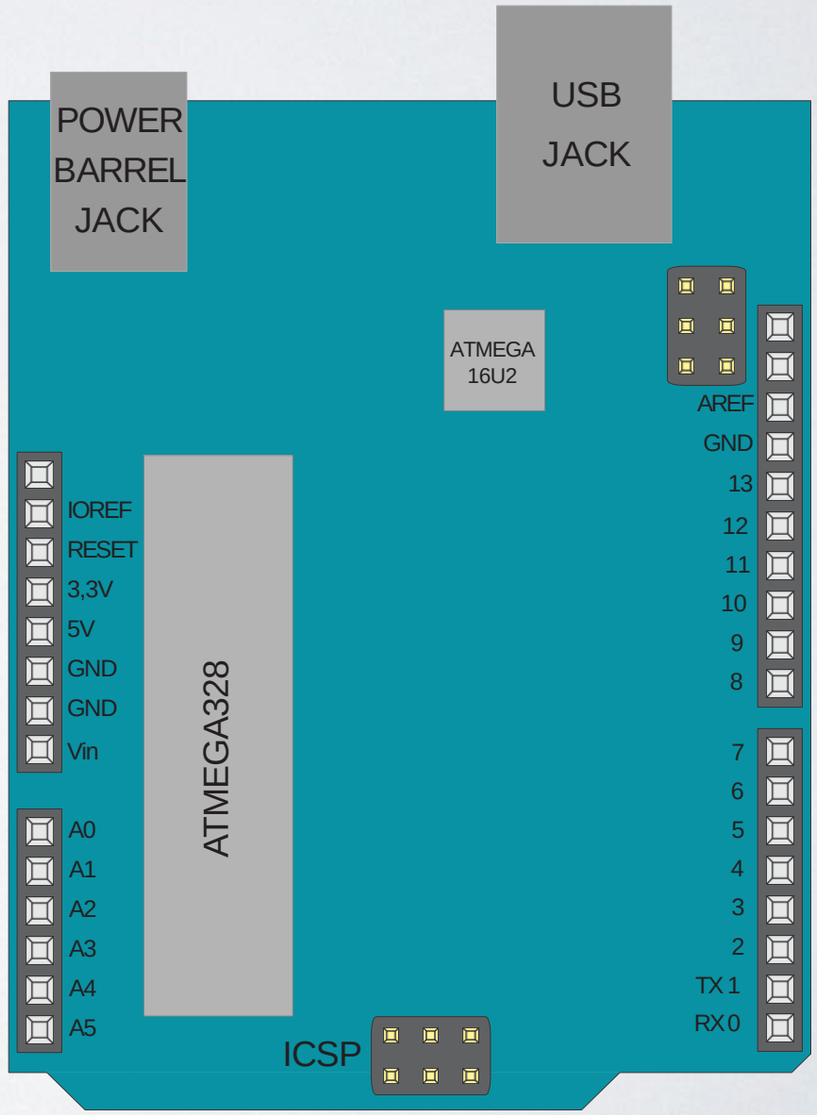
What is Arduino?

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?

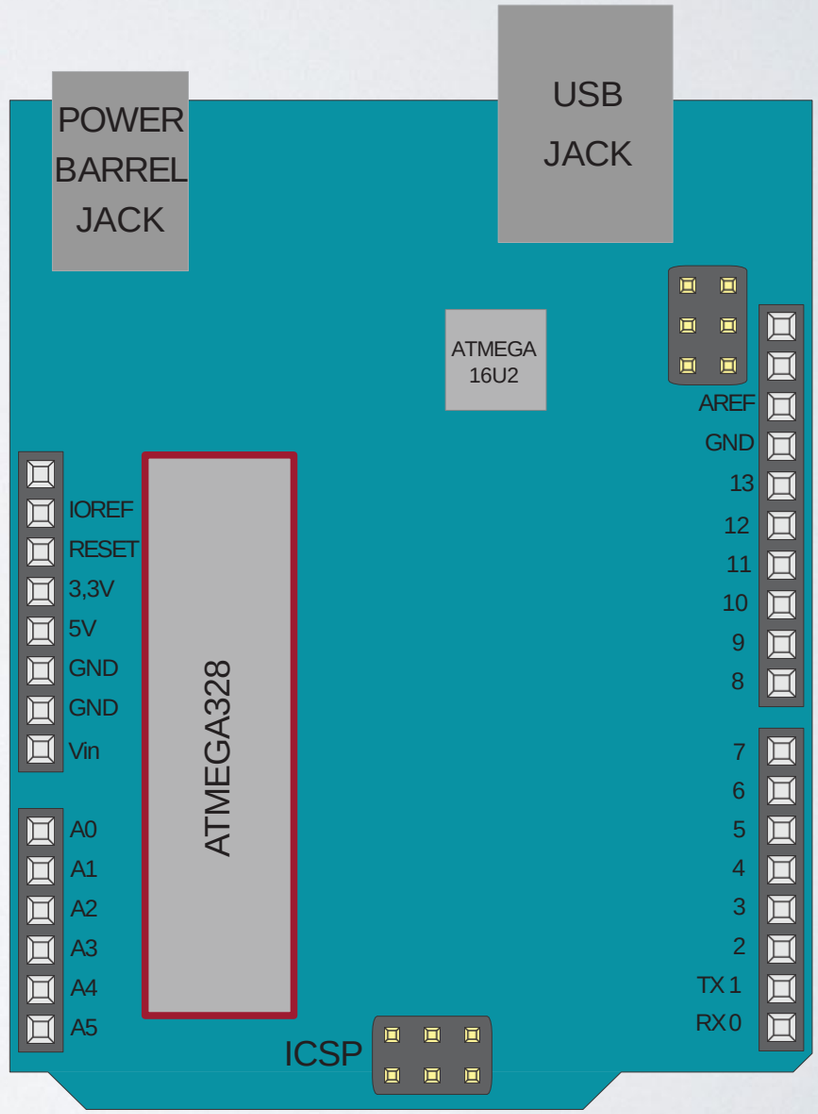
Arduino Hardware (plural!)

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**!



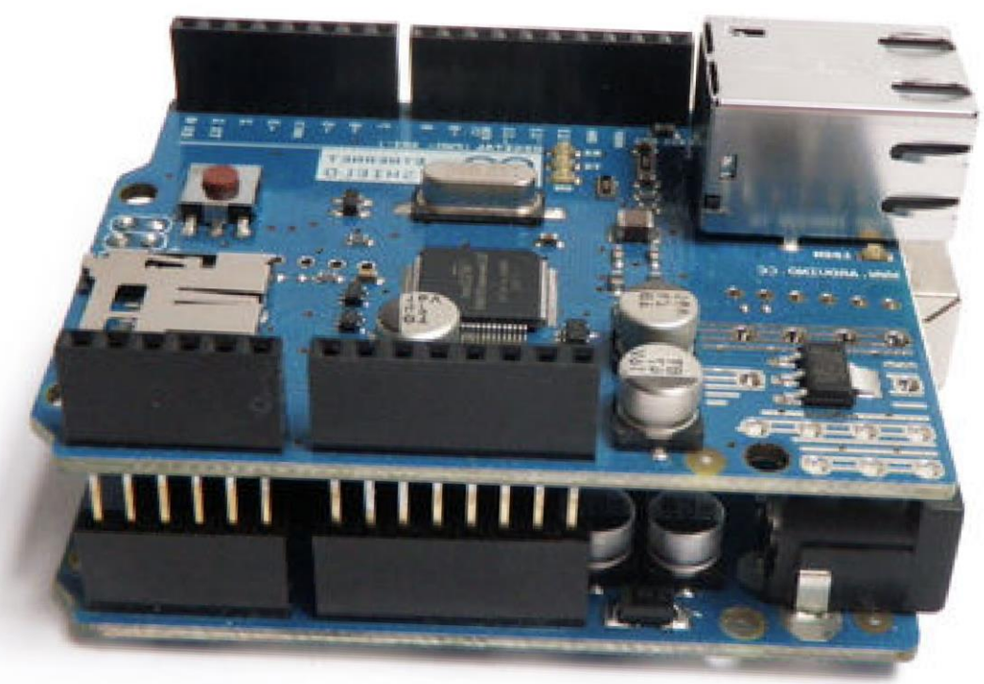
What is Arduino?

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!)

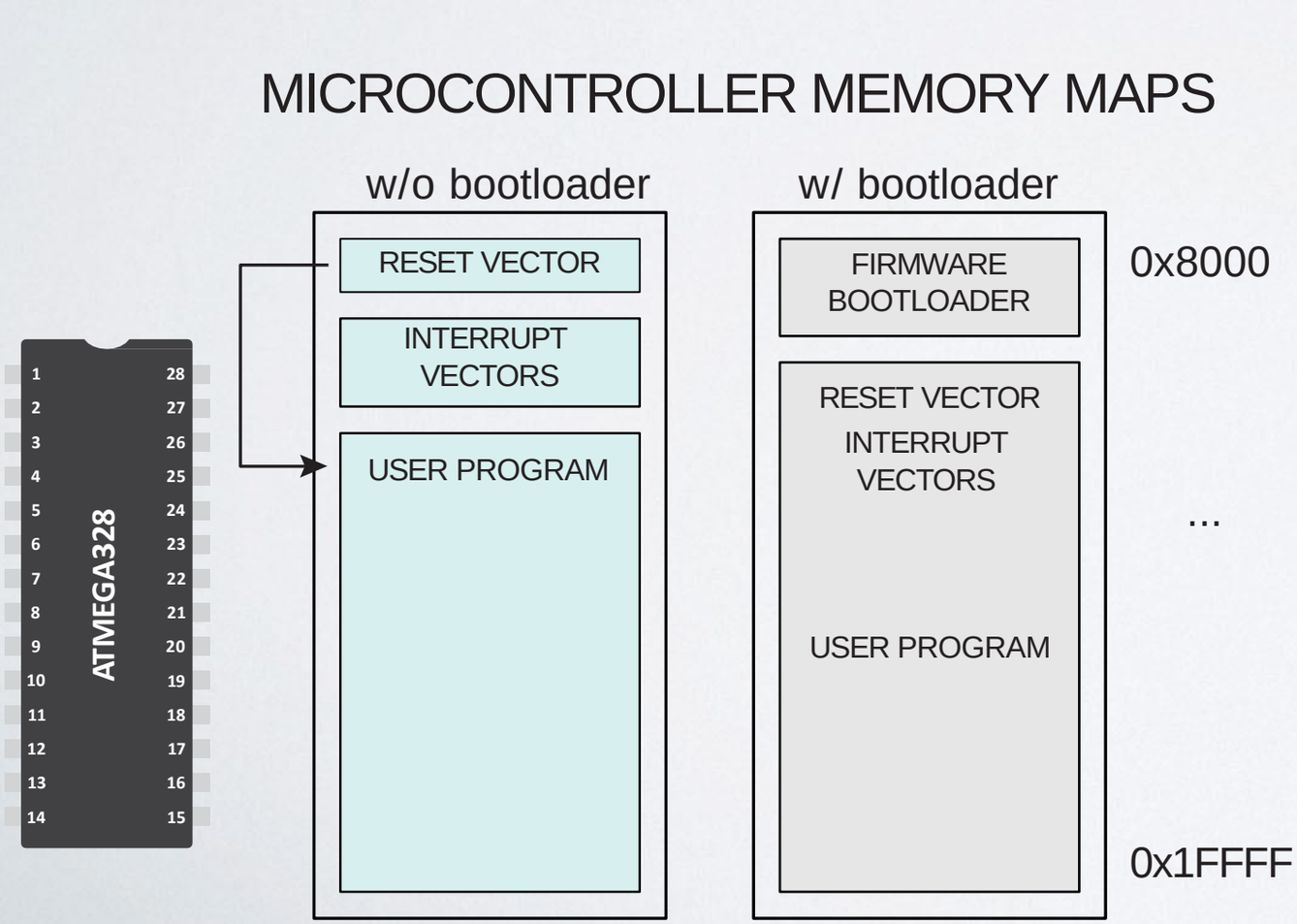
There are many, Many, MANY Arduino boards and clones that vary in form factor, computing power, number and type of peripherals, etc.

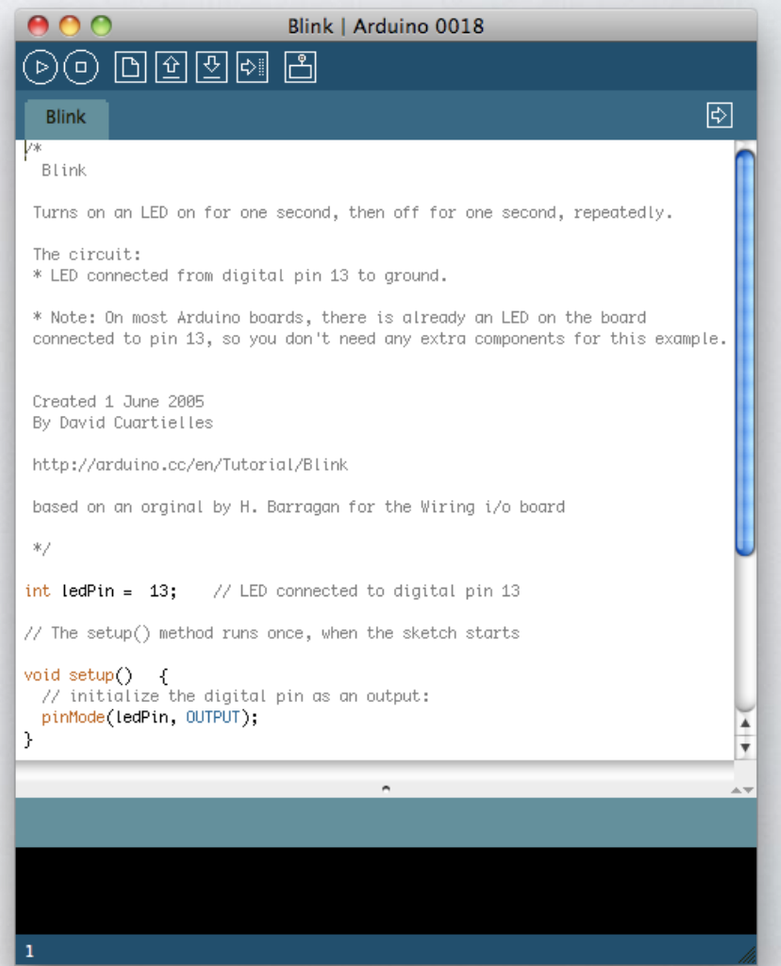
http://blog.arduino.cc/wp-content/uploads/2013/11/ArduinoEvolution_make.jpg

What is Arduino?

Arduino Software (plural!)

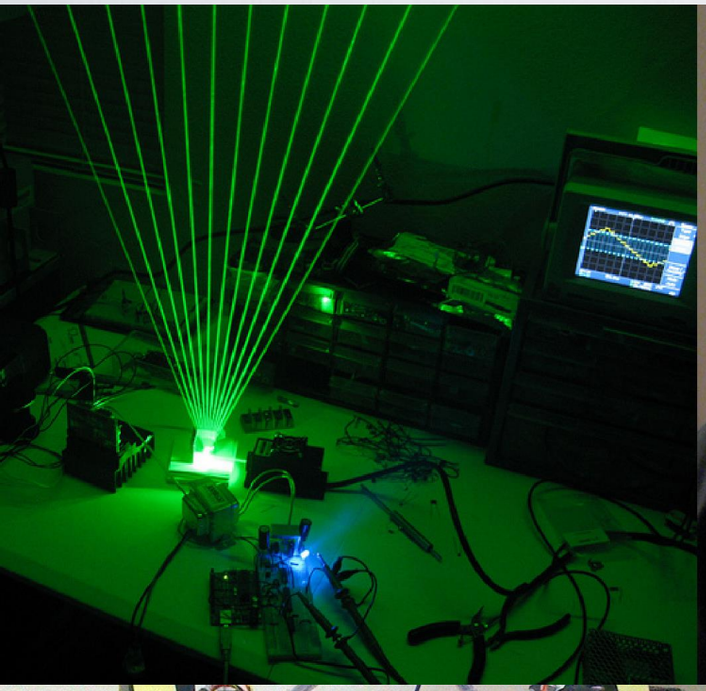
The Arduino software is comprised by:
- A *firmware bootloader* that enables us to easily load programs onto the MCU
- A computer application (*IDE*) to edit, compile, and upload our programs, as well as communicate via USB with the Arduino boards

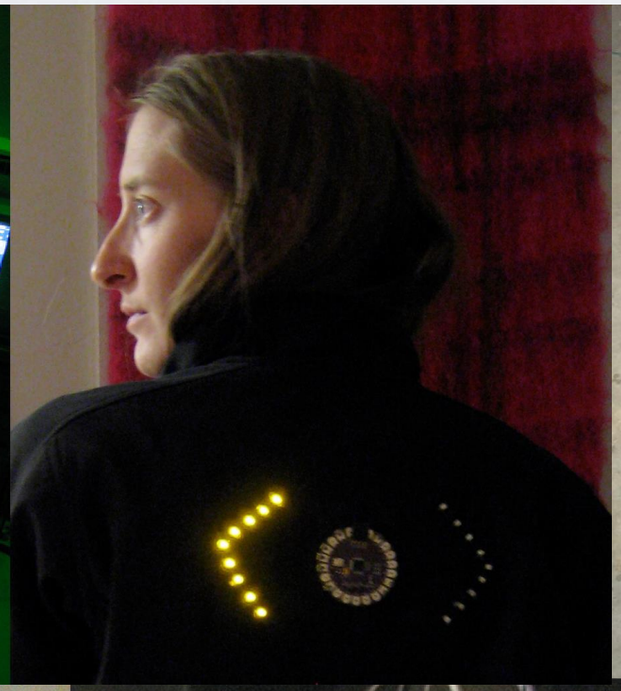


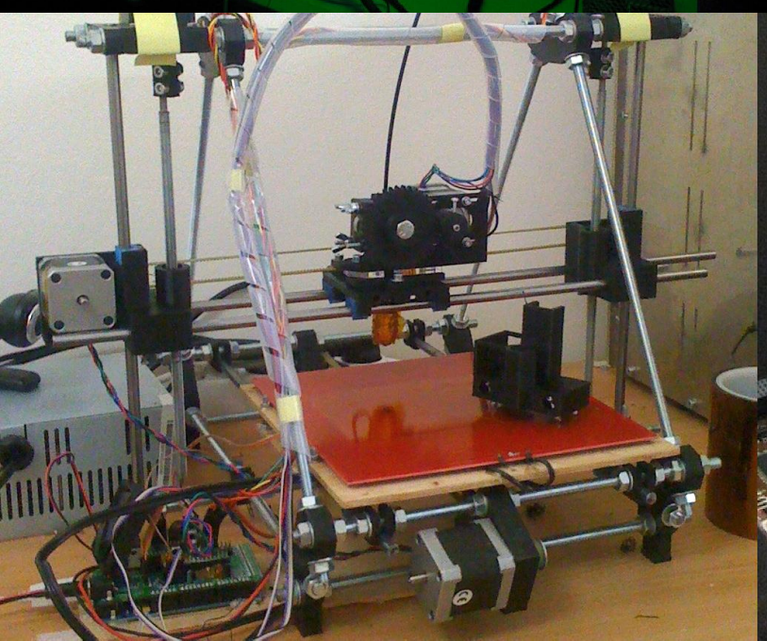


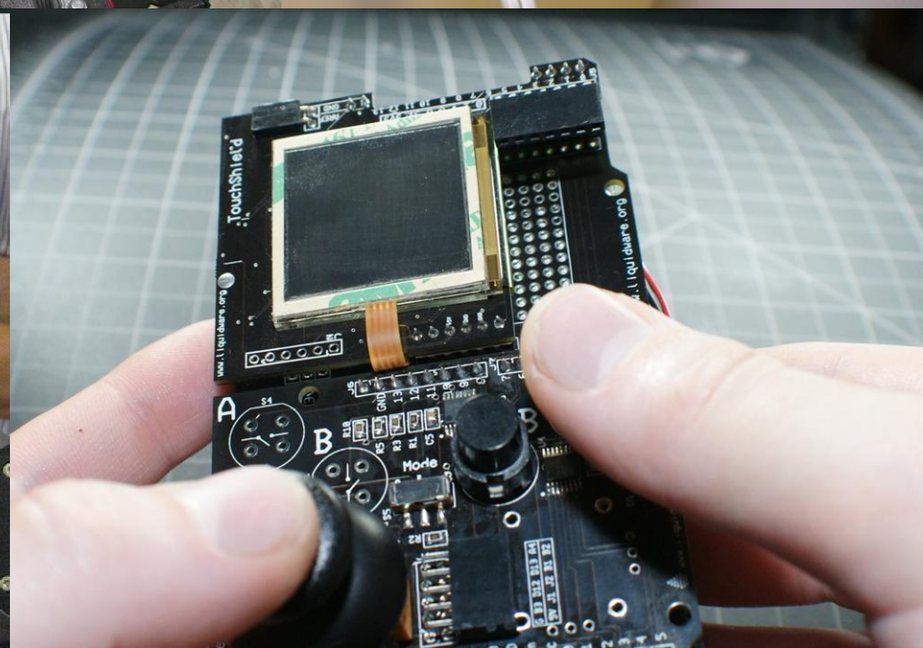
What uses does the Arduino toolset have?

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?

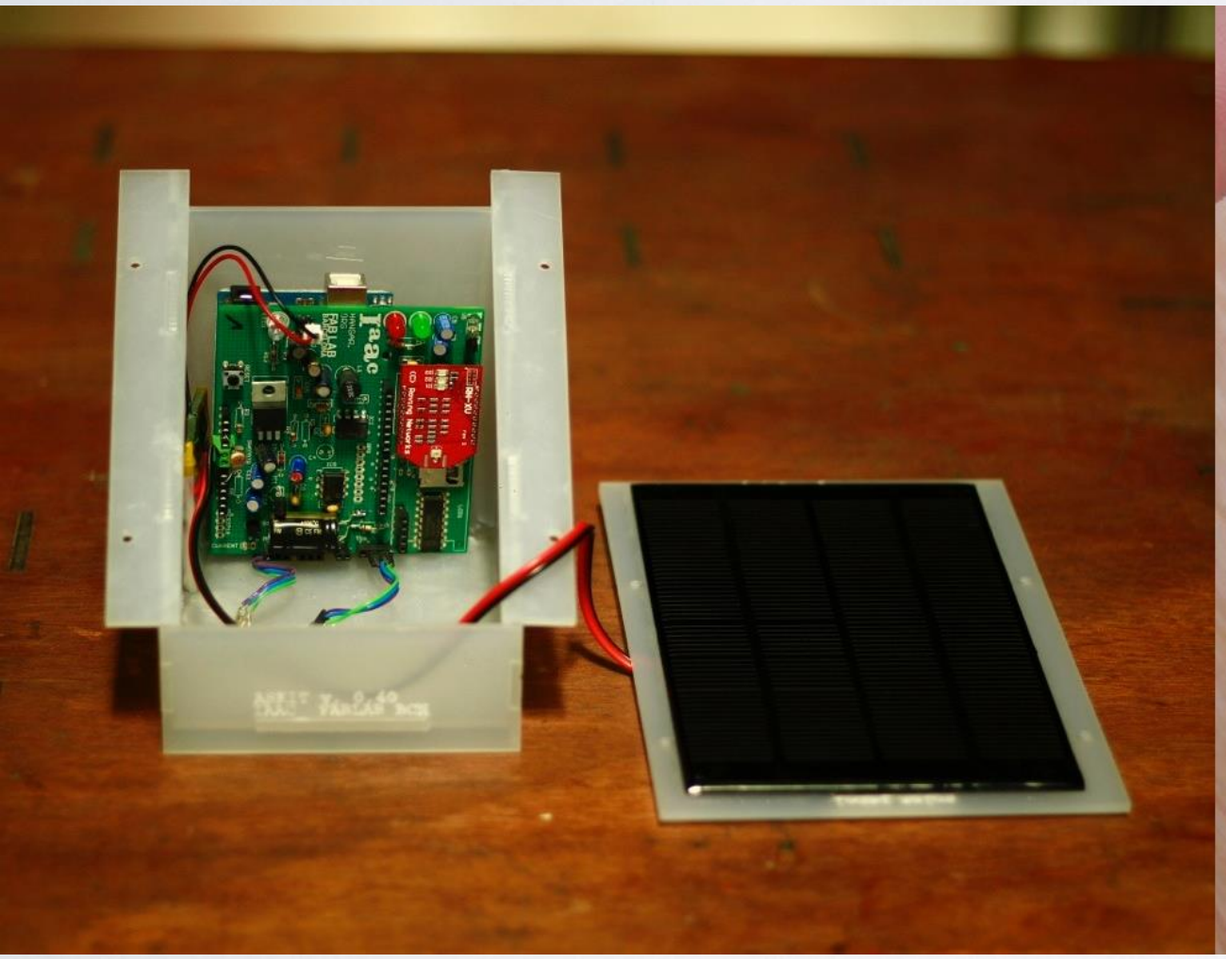
From the horse's mouth:

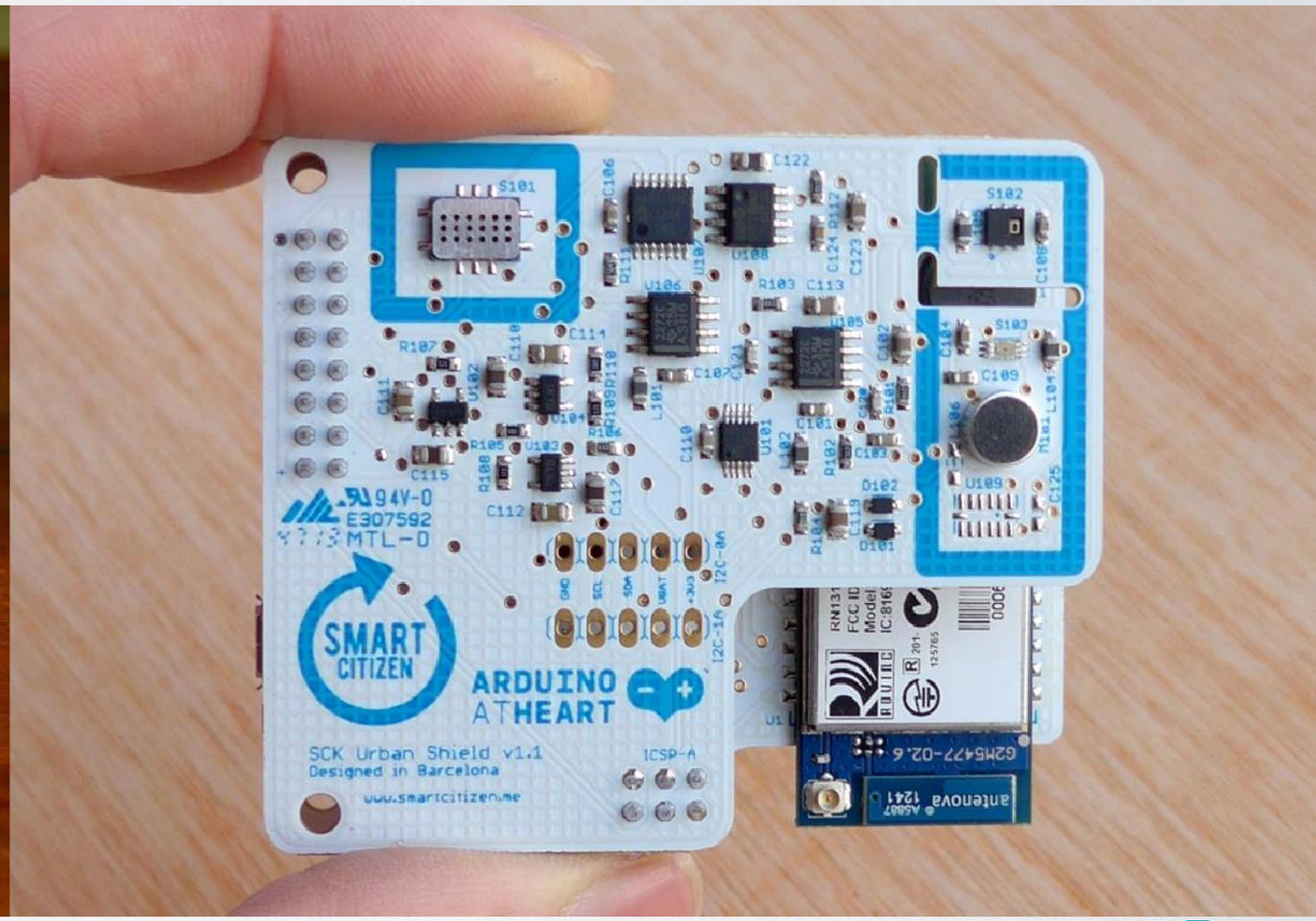
https://www.youtube.com/watch?v=UoBUXOOdLXY&t=175s

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)





open source hardware

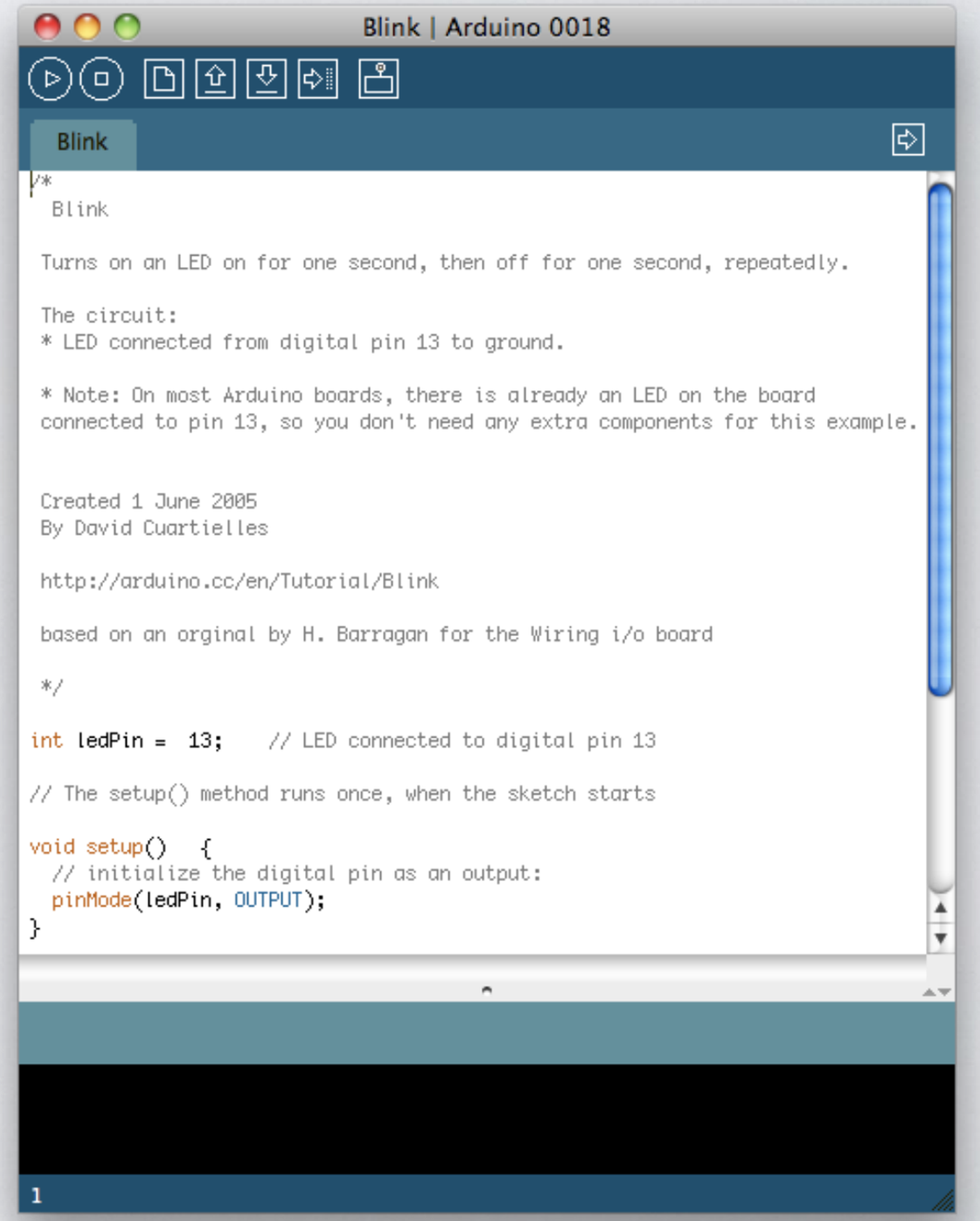
Getting Started with Arduino

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+)

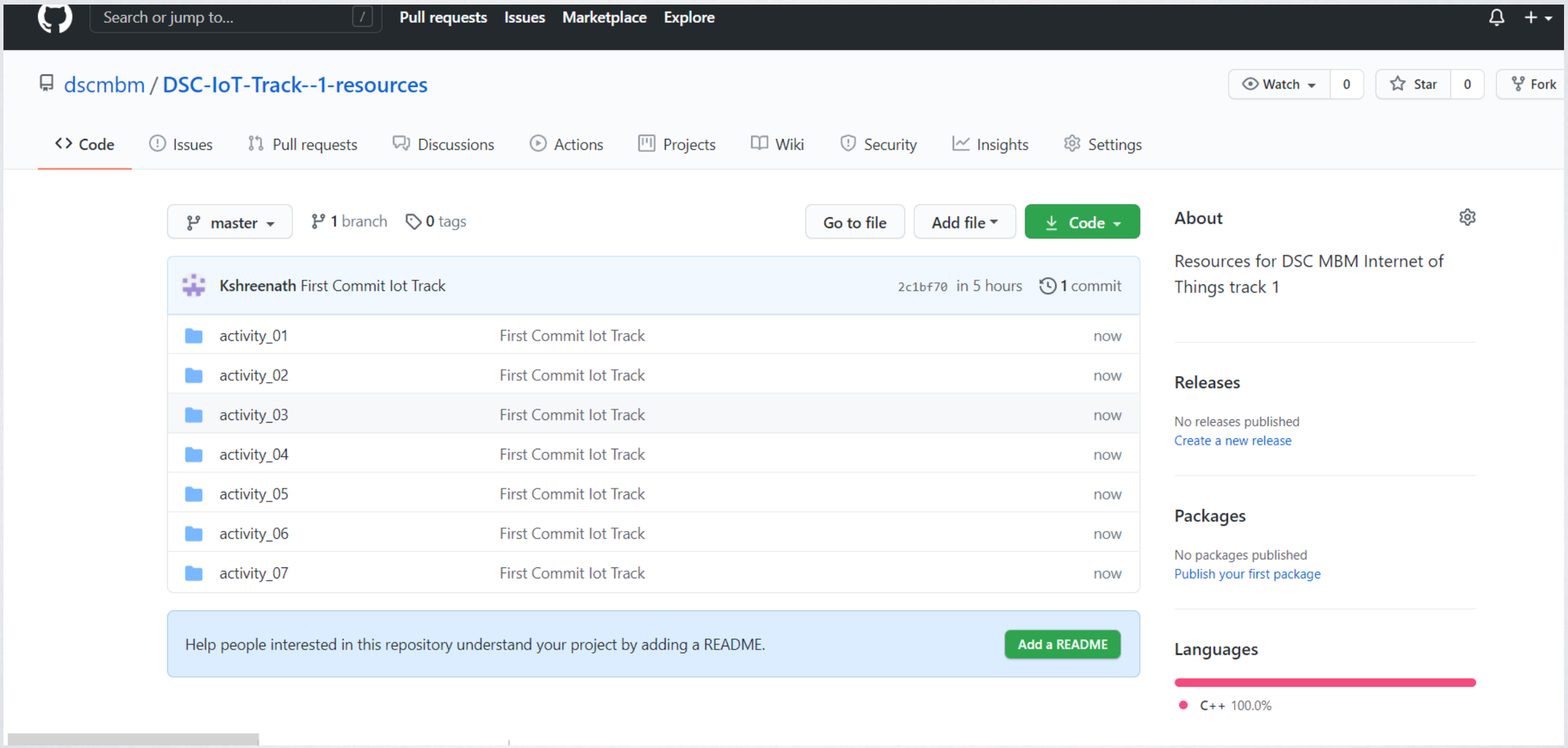
Linux (32-bit, 64-bit)



Getting Started with Arduino

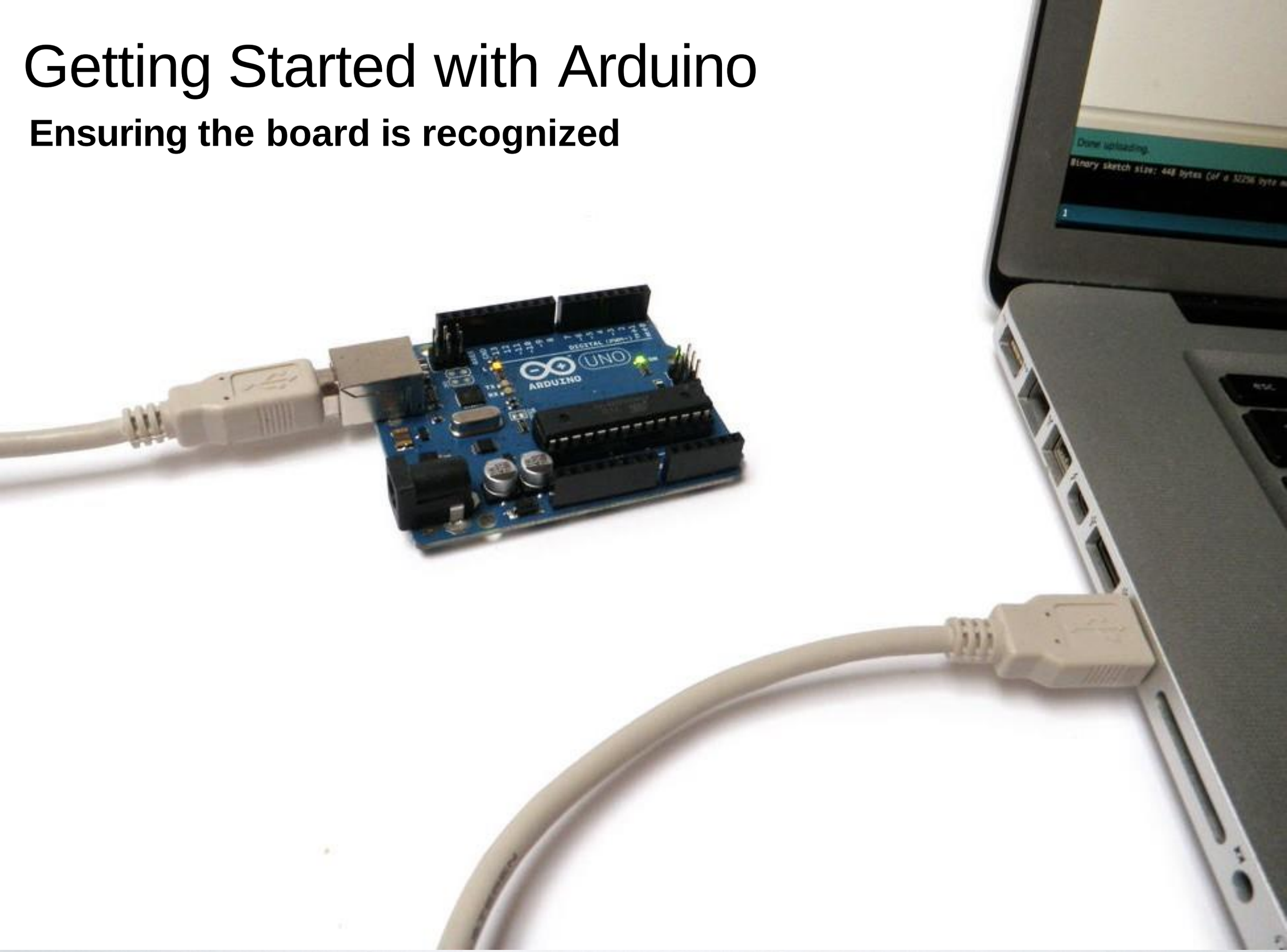
Downloading the code for this class—all our code is free and Open-Source, so download it, use it, share it (when you make millions, remember us (-:)

https://github.com/dscmbm/DSC-IoT-Track--1-resources



Getting Started with Arduino

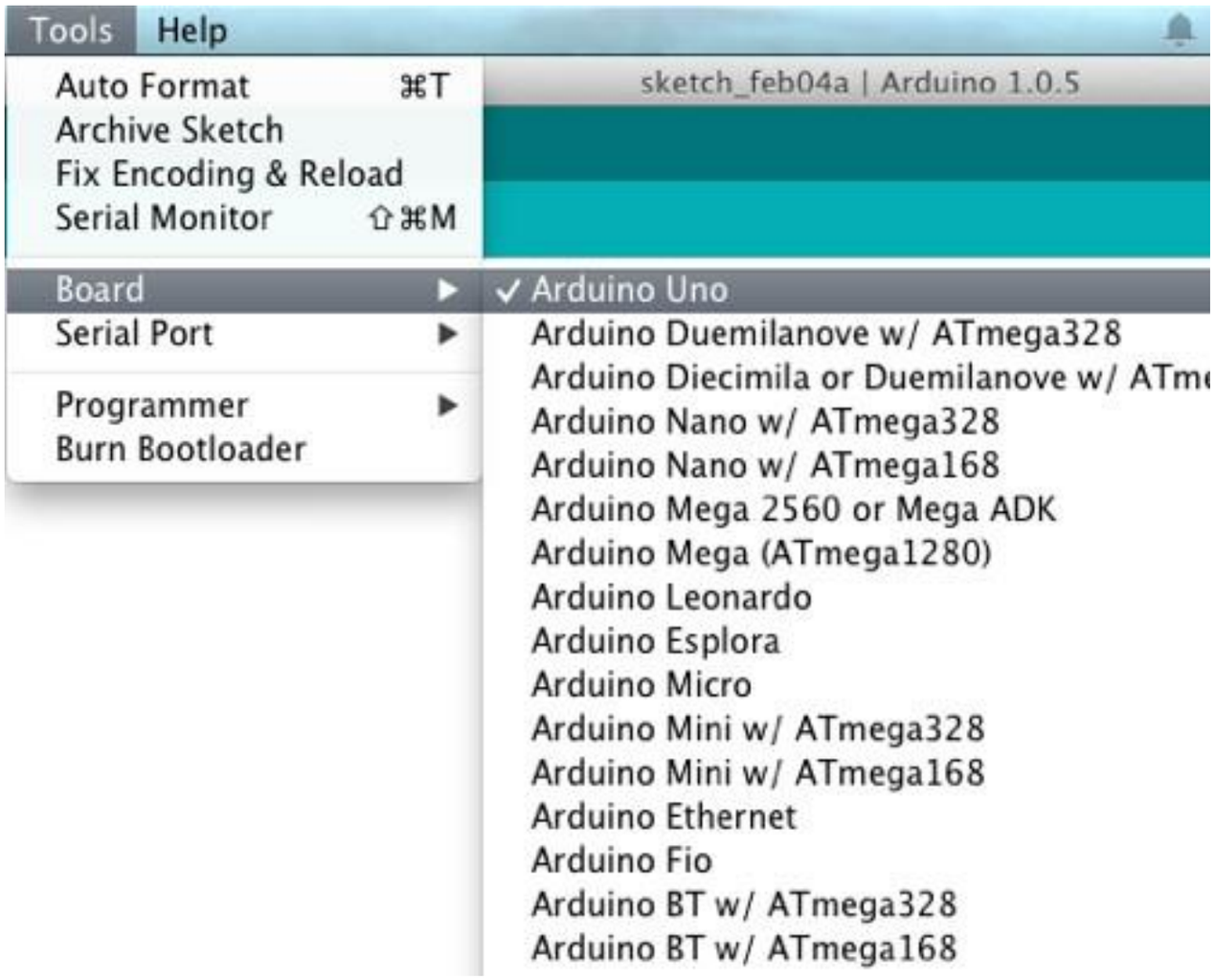
Ensuring the board is recognized

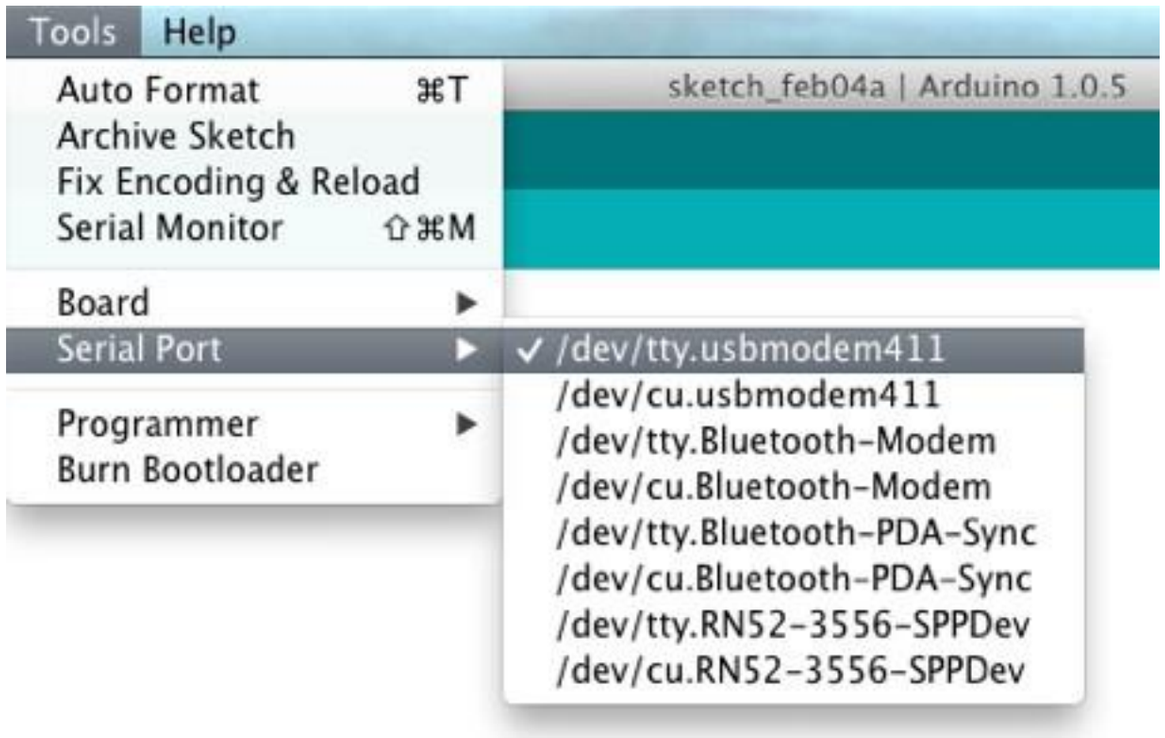


Getting Started with Arduino

Ensuring the board is recognized

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



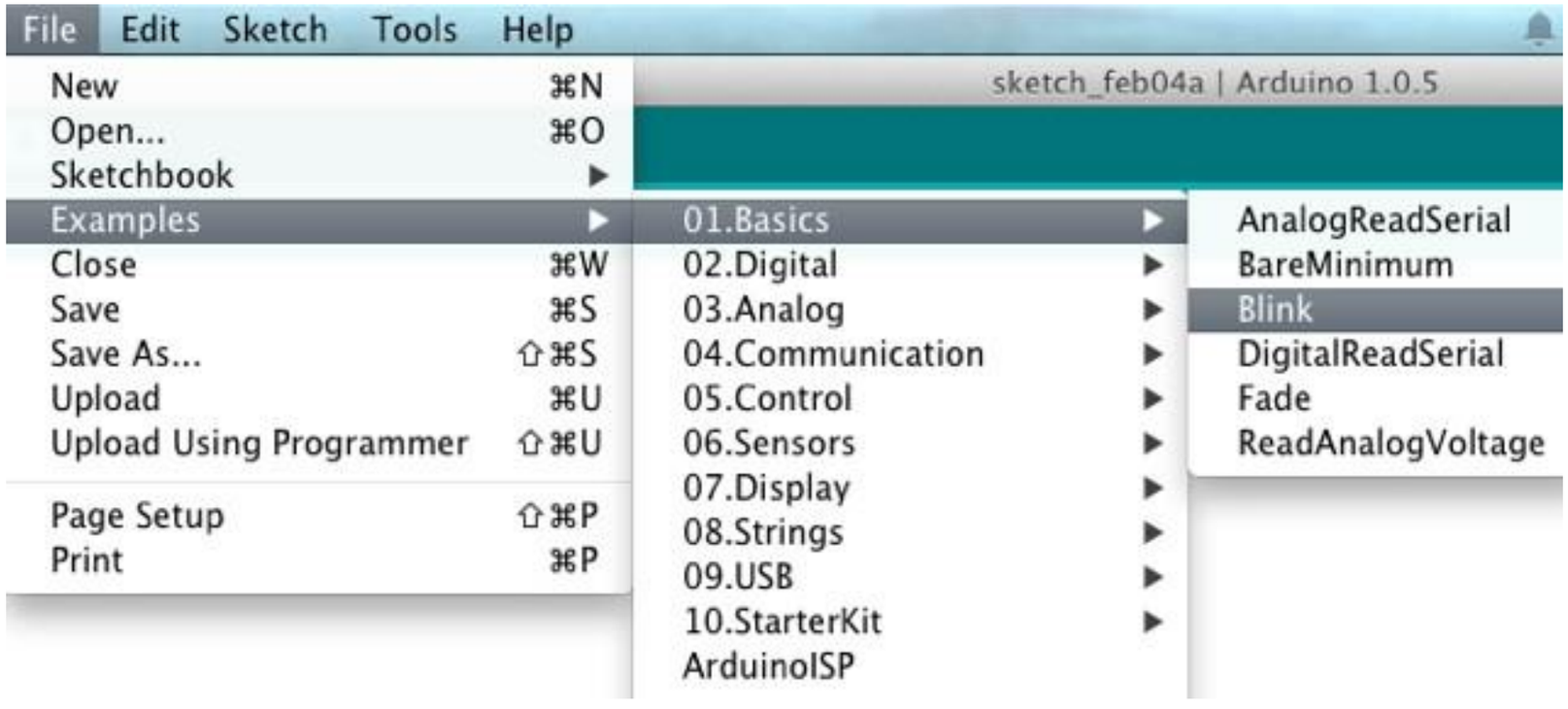


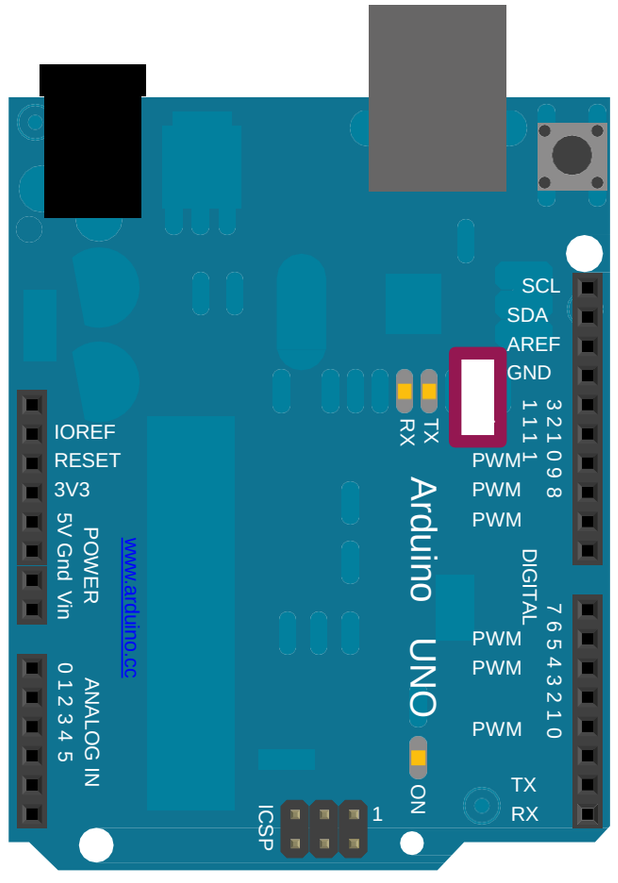
Windows users might need to install/update the board driver at the time of installation by keeping board plugged in.

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:

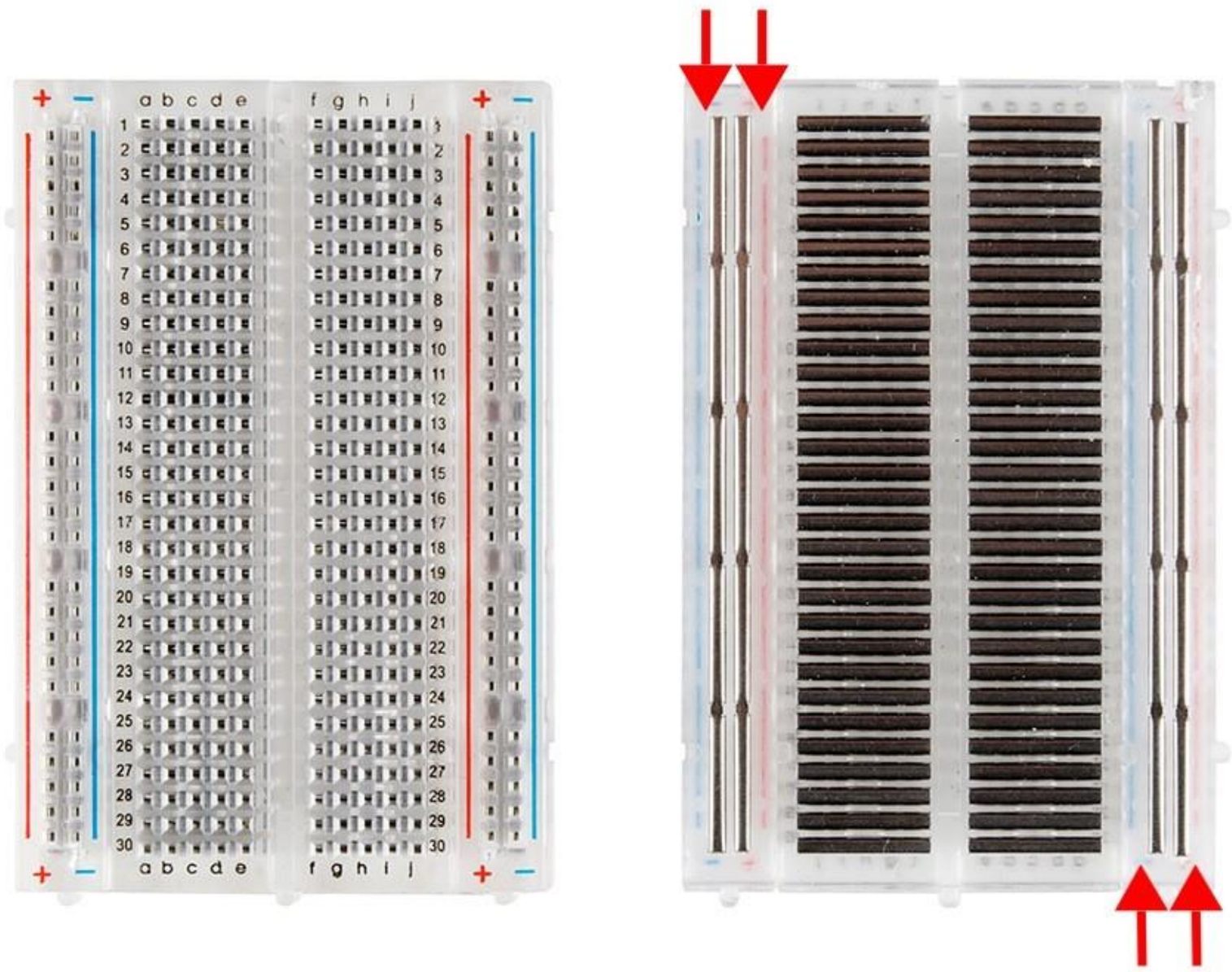


Photo credit: Sparkfun Electronics, Inc.

Getting Started with Arduino

Reading Resistor Codes:

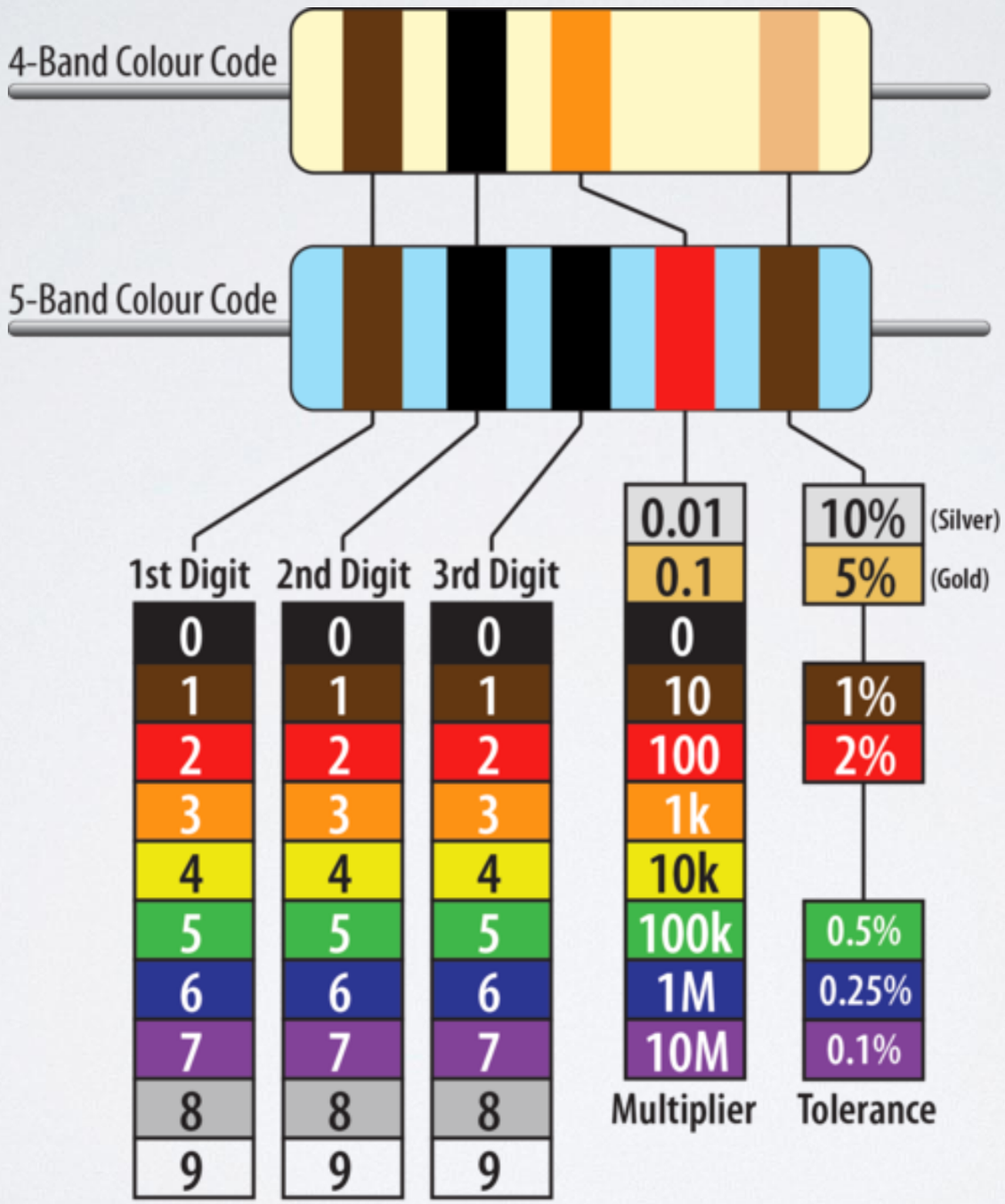
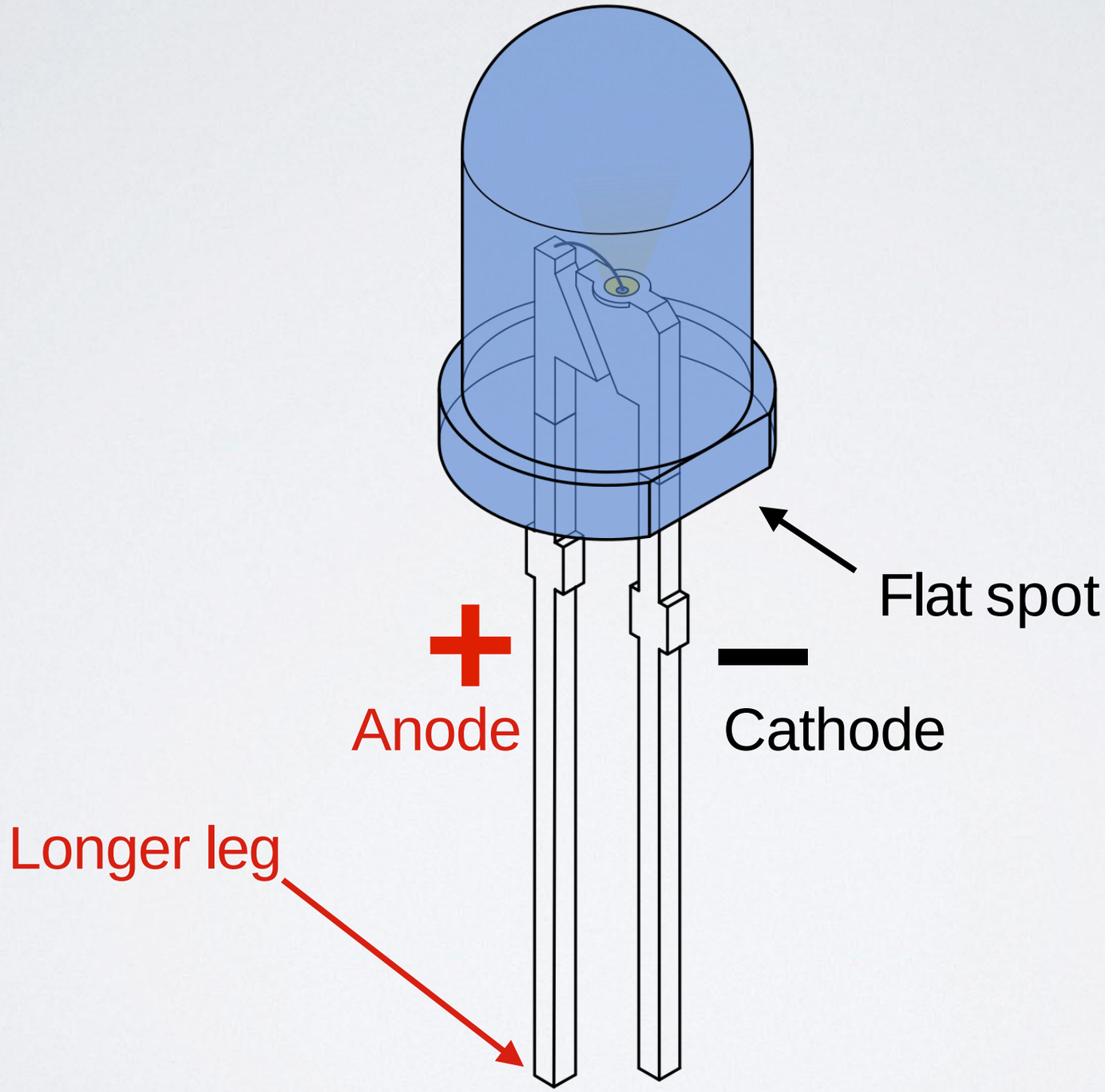


Photo credit: www.freetronics.com

Getting Started with Arduino

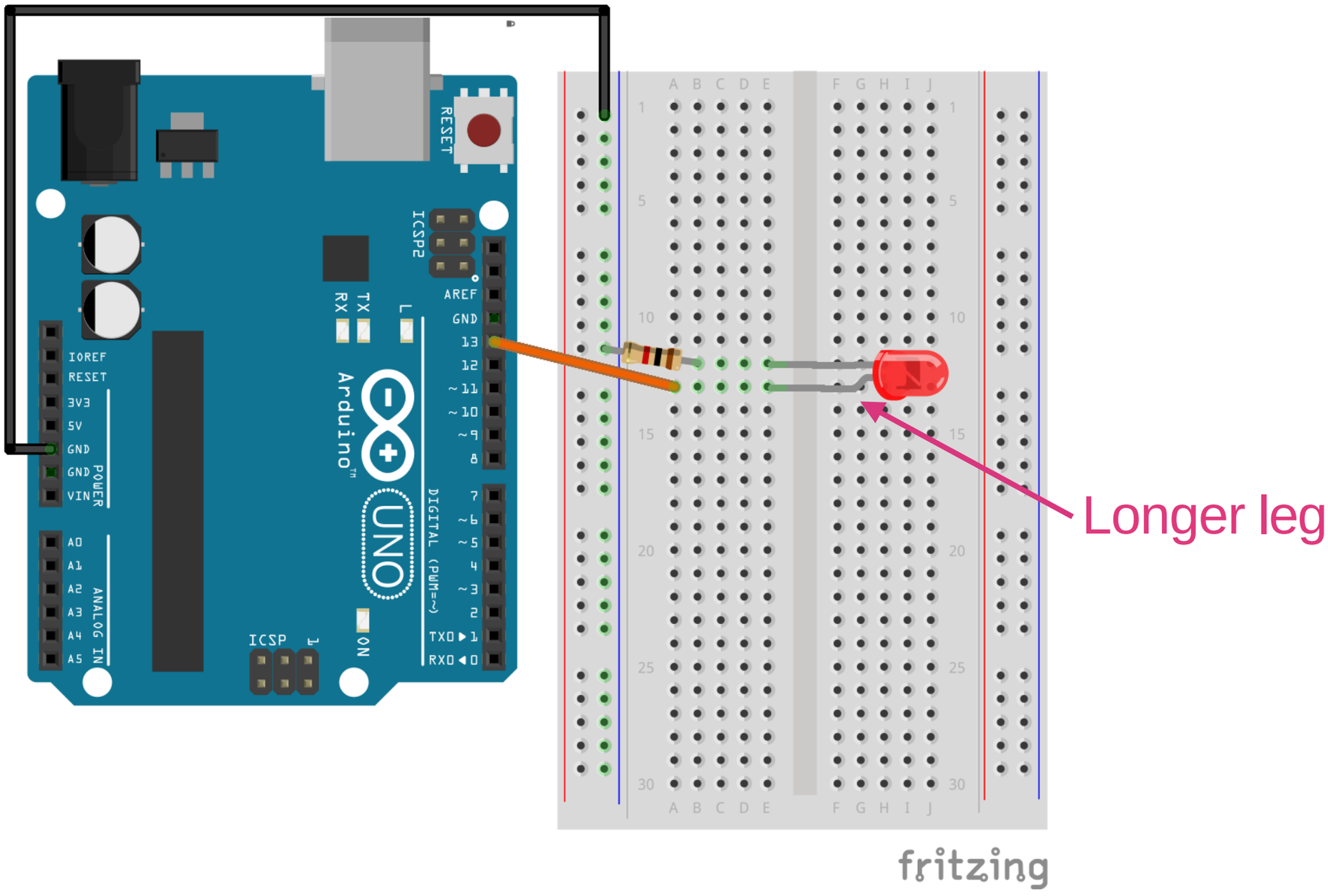
Reading LED polarity:



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

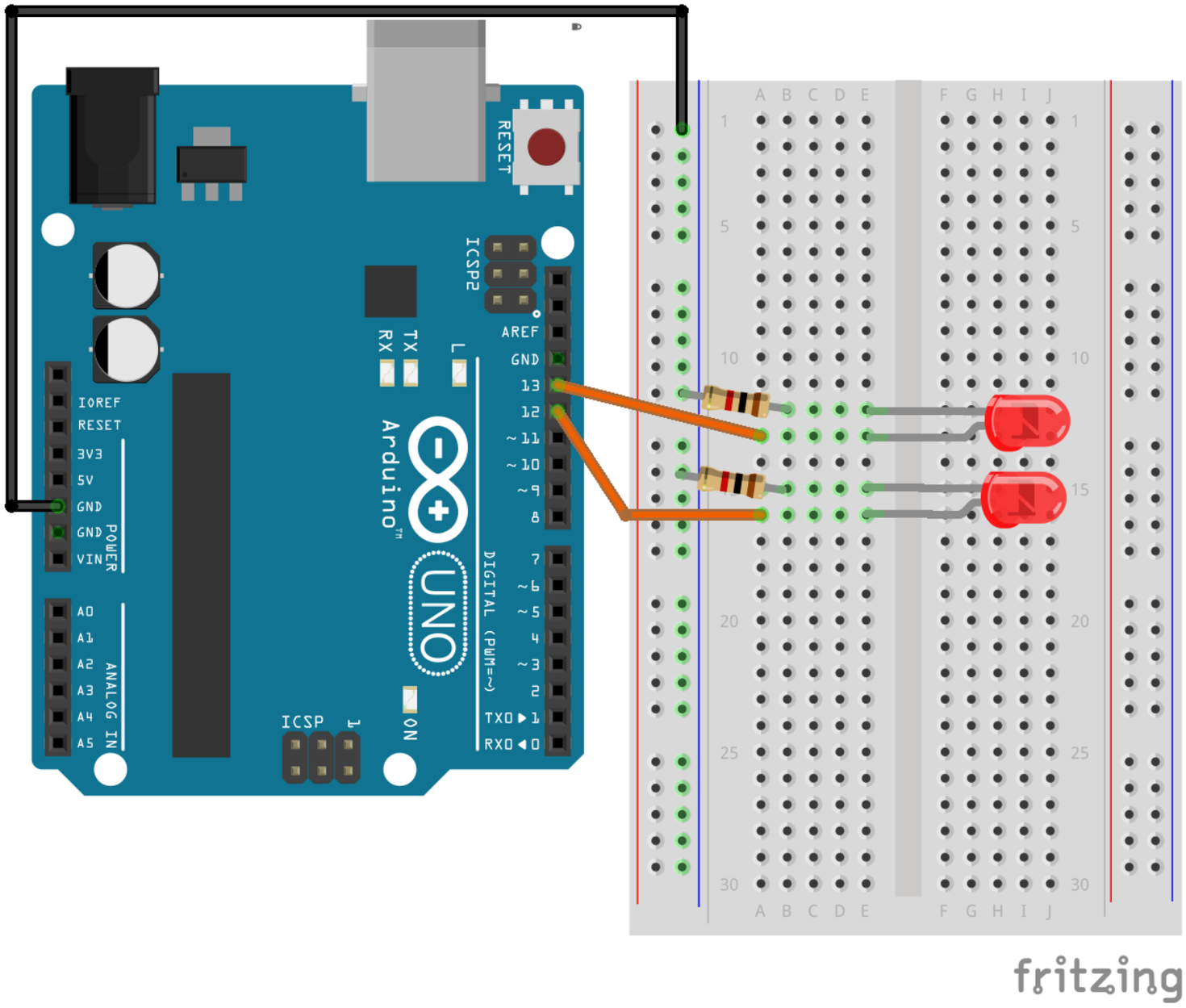
Anatomy of an Arduino Program ('sketch'):

```
/*
  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]

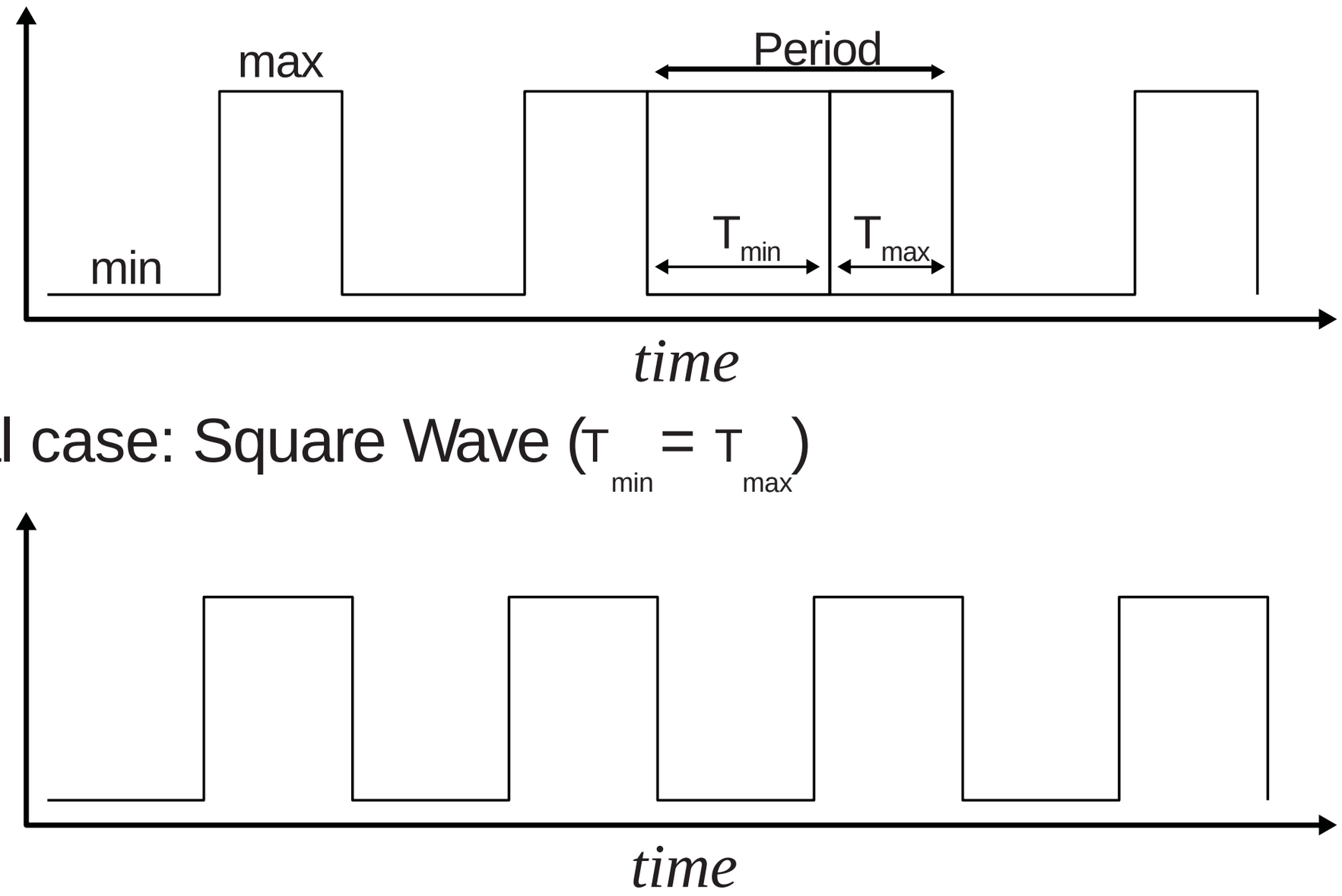
What happens when we decrease the on/off delay (T_{min})?



Getting Started with Arduino

Activity 2: Pulse Width Modulation (PWM)

Rectangular or Pulse Wave



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

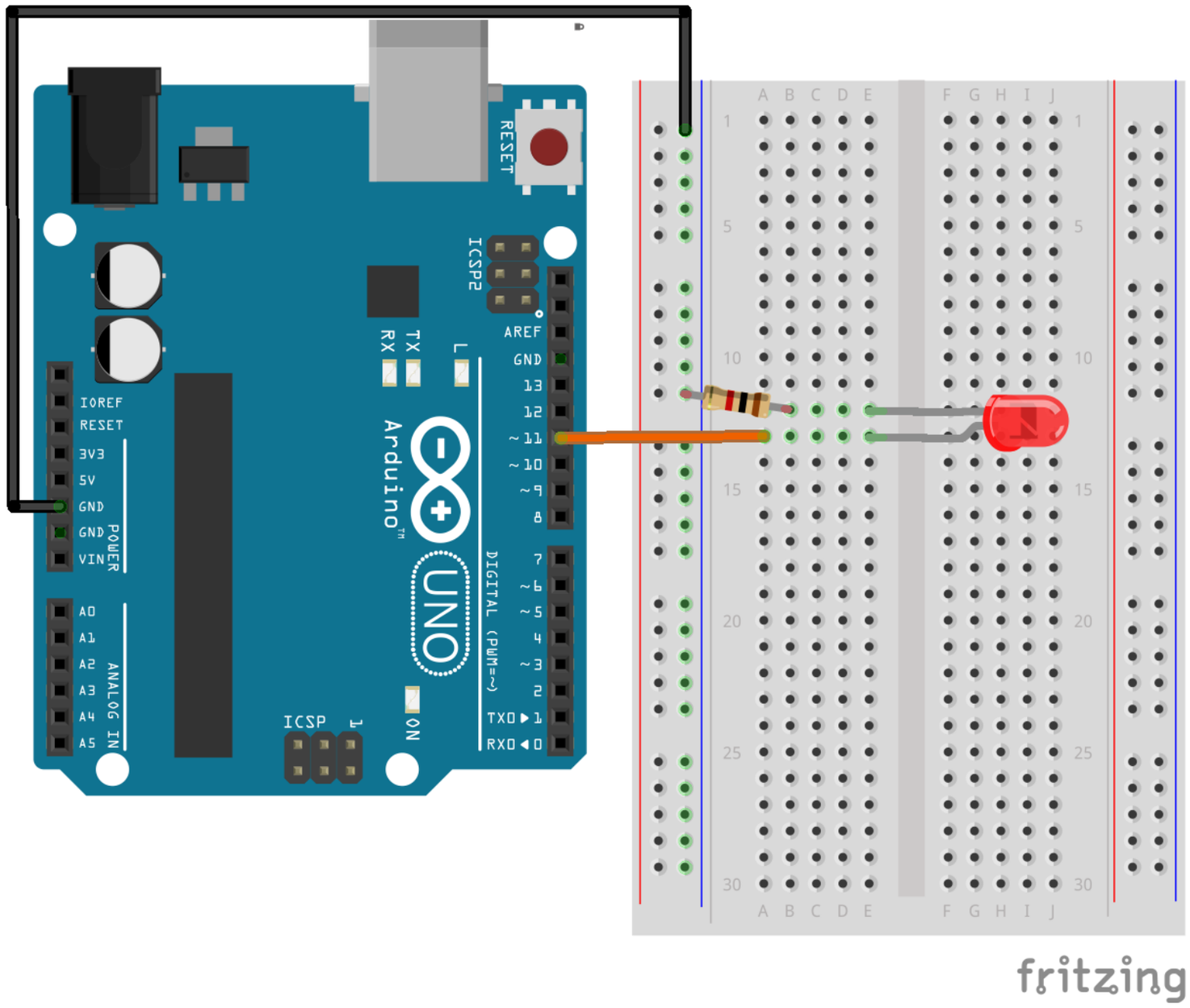
Frequency = 1/Period

Duty Cycle = 100% x $T_{max}/(T_{max}+T_{min})$

Getting Started with Arduino

Activity 2: Pulse Width Modulation (PWM)

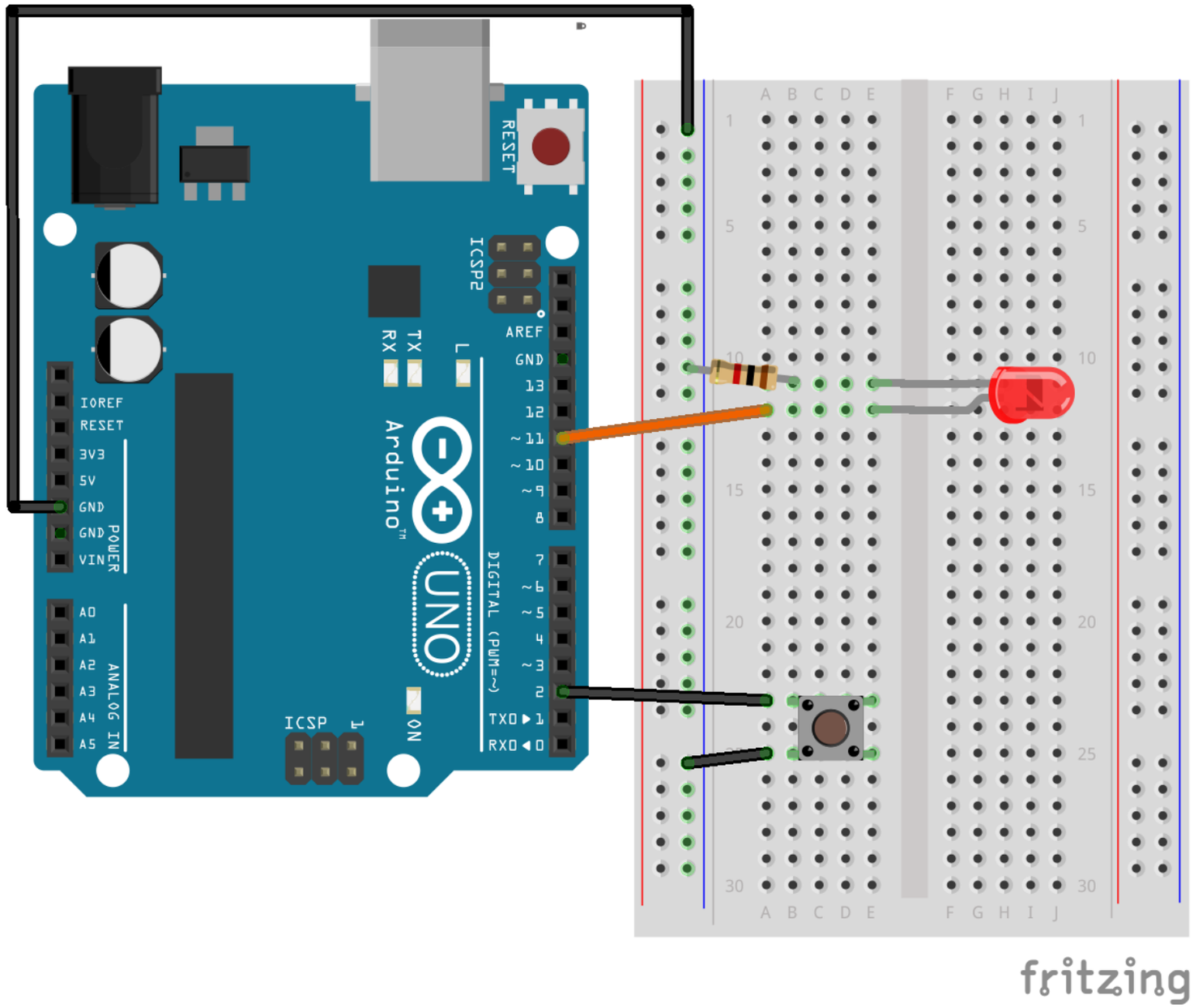
What happens when we modulate the on/off delay (T_{min})? [heartbeat.ino]



Getting Started with Arduino

Activity 3: Digital Inputs [button_hold.ino]

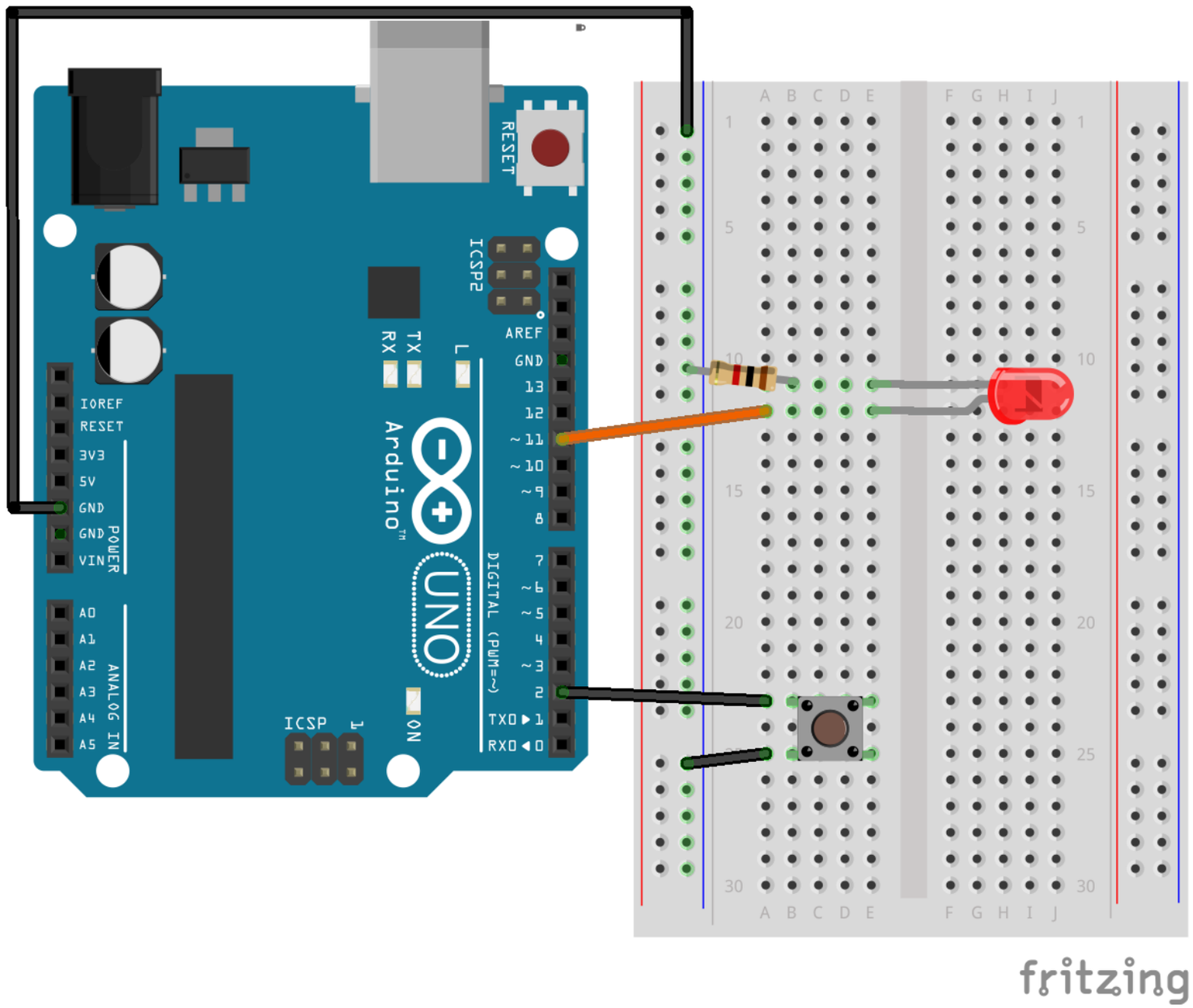
Using the digitalRead() function, we can determine the Open/Closed state of the button.



Getting Started with Arduino

Activity 3: Digital Inputs [button_toggle.ino]

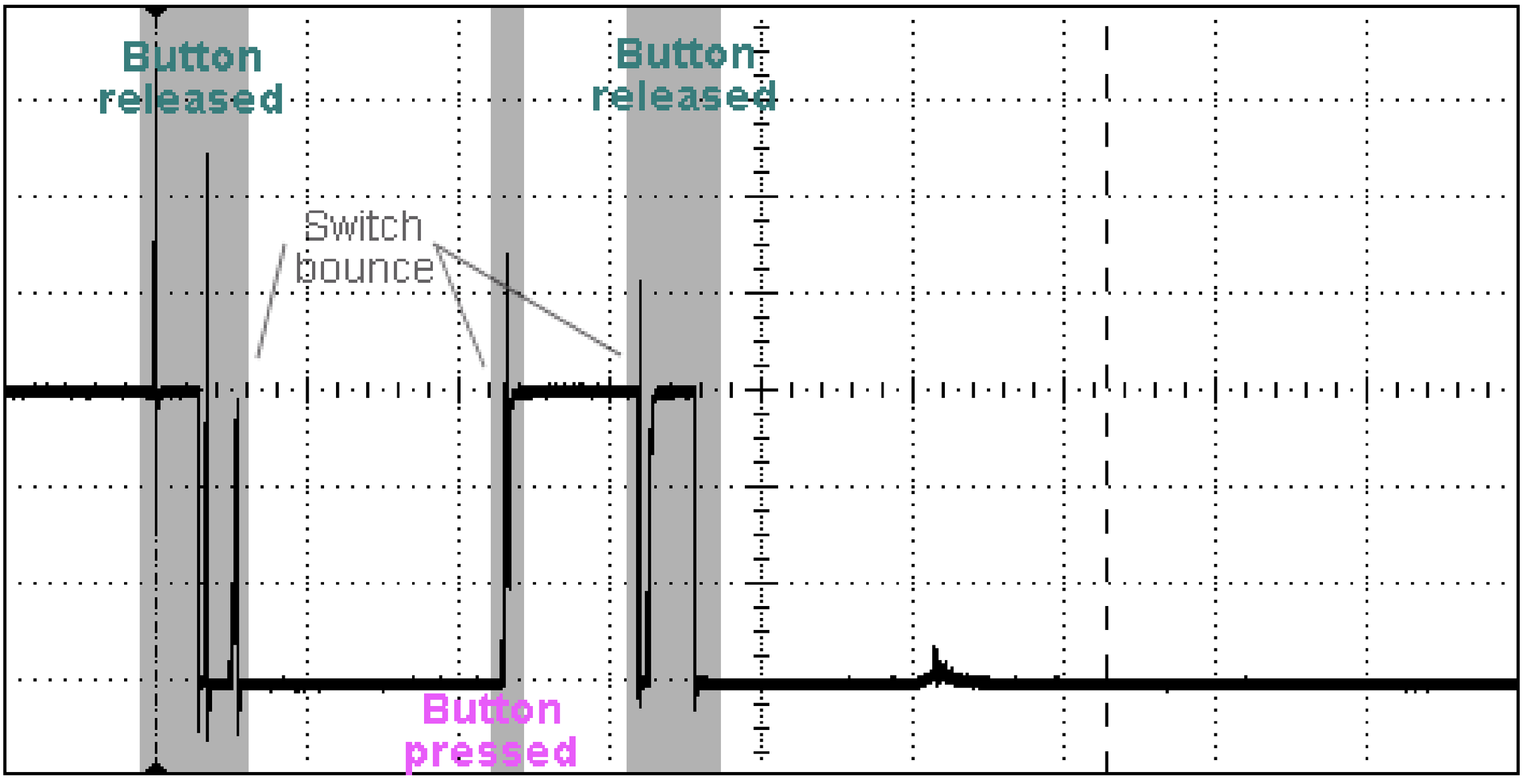
Using the Serial Object, we can transmit information via USB from/to the Arduino board.



Getting Started with Arduino

Activity 3: Digital Inputs [button_toggle.ino]

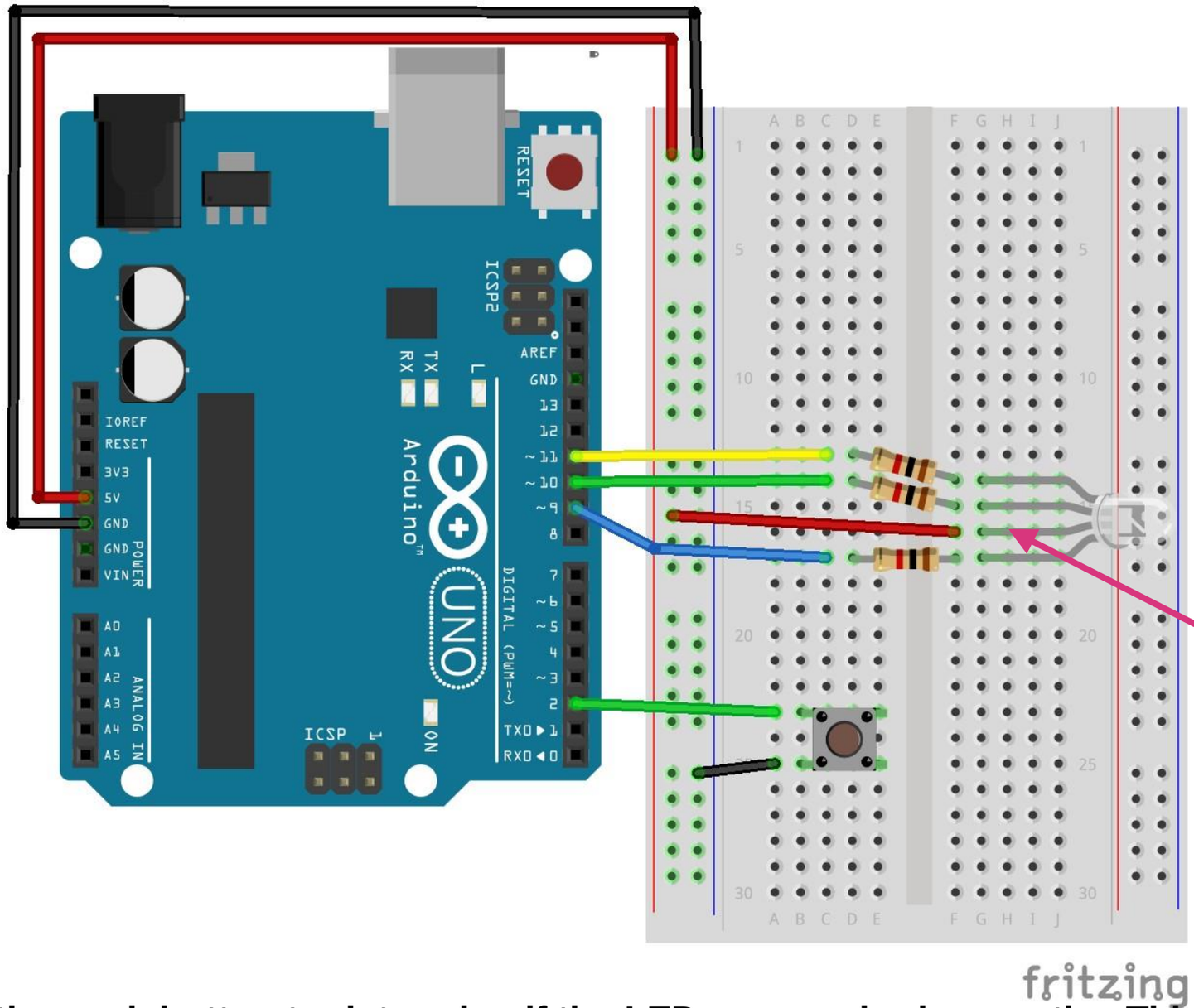
Electromechanical artifacts can affect your devices. Electrical bouncing is one such artifact; it can be mitigated with software, hardware, or both!



Getting Started with Arduino

Activity 4: Multiple Digital Outputs [rgb_led_cycle.ino]

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

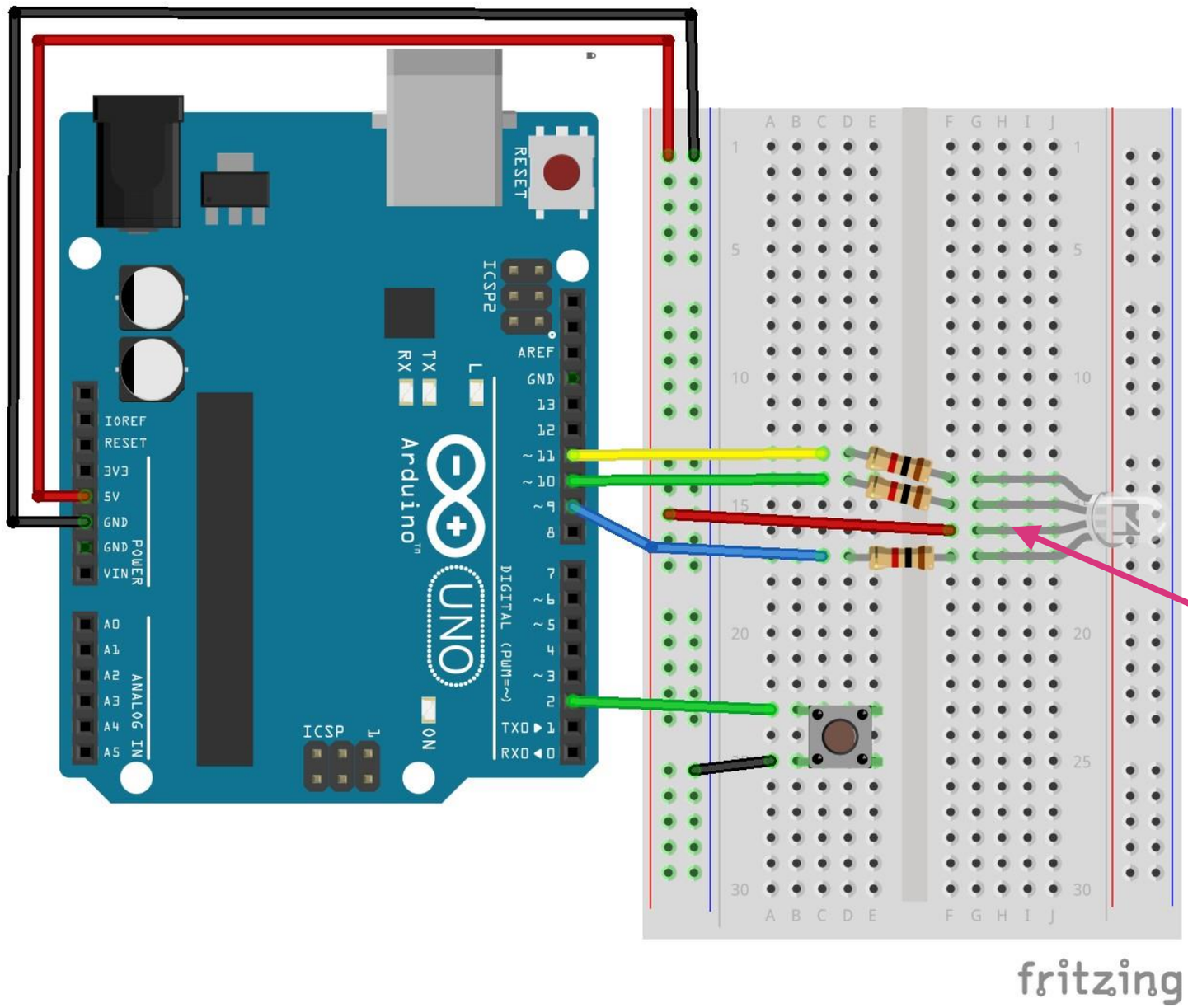


Note: our program uses the push button to determine if the LEDs were wired correctly. This is an example of a code written for user interaction!

Getting Started with Arduino

Activity 4: Multiple Digital Outputs [rgb_led_fade.ino]

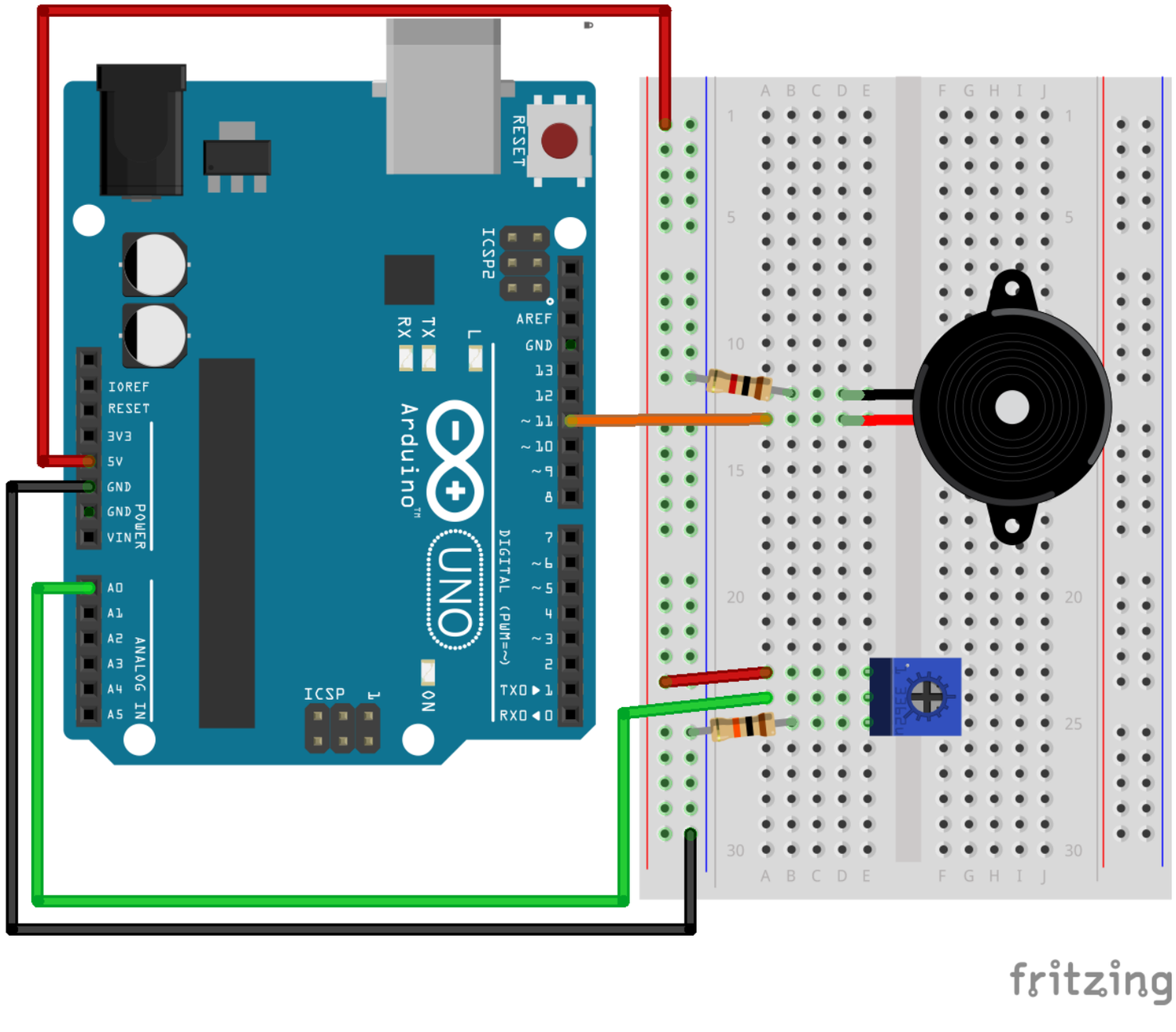
We have 2^{24} possible color combinations as each LEDs accept a brightness level between 0 and 255.



Getting Started with Arduino

Activity 6: Analog Input [analog_input_vol.ino]

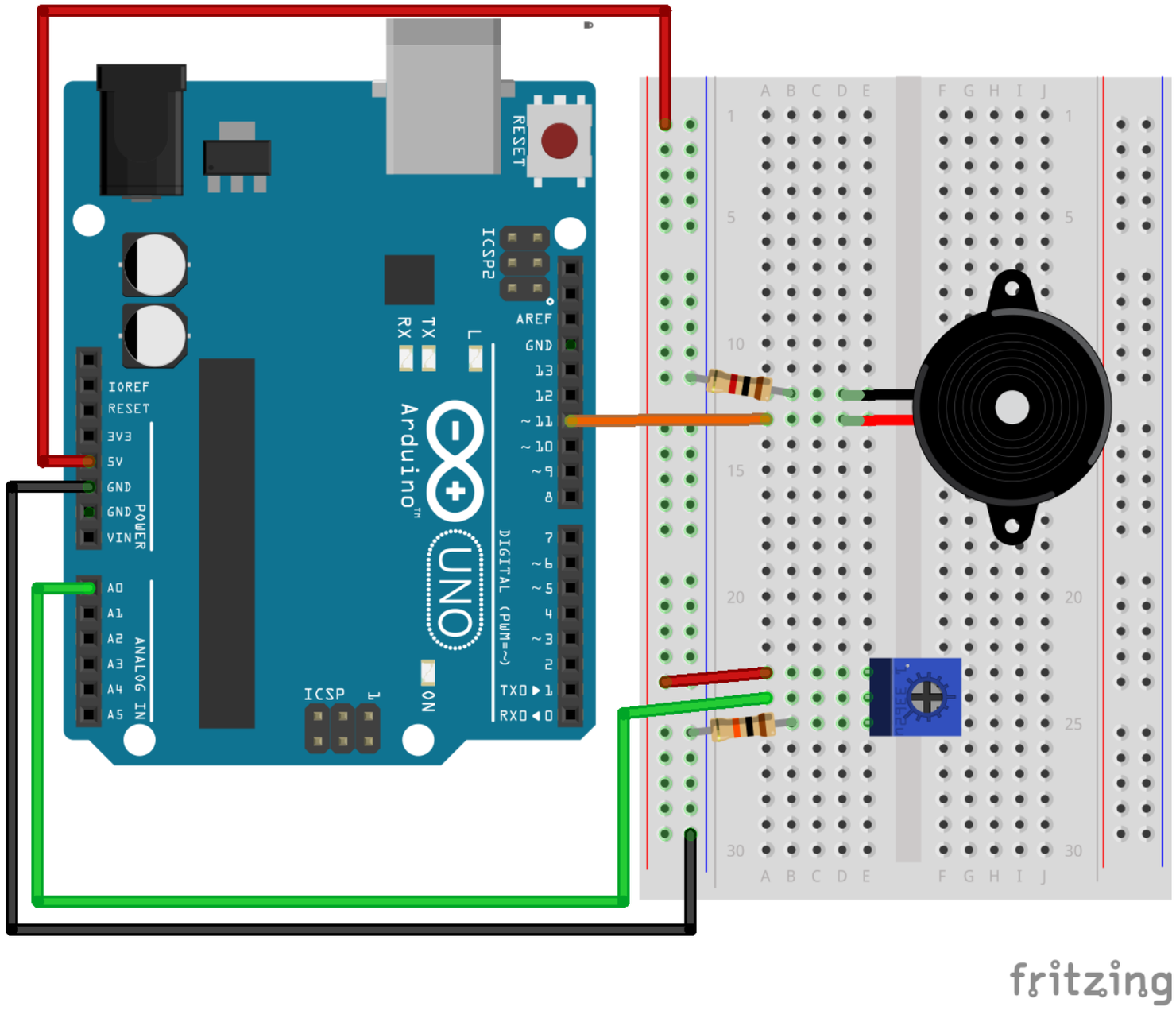
The function analogRead(…) uses the built-in Analog to Digital Converter to measure the analog voltage on the pot's sweeper pin.



Getting Started with Arduino

Activity 6: Analog Input [analog_input_freq.ino]

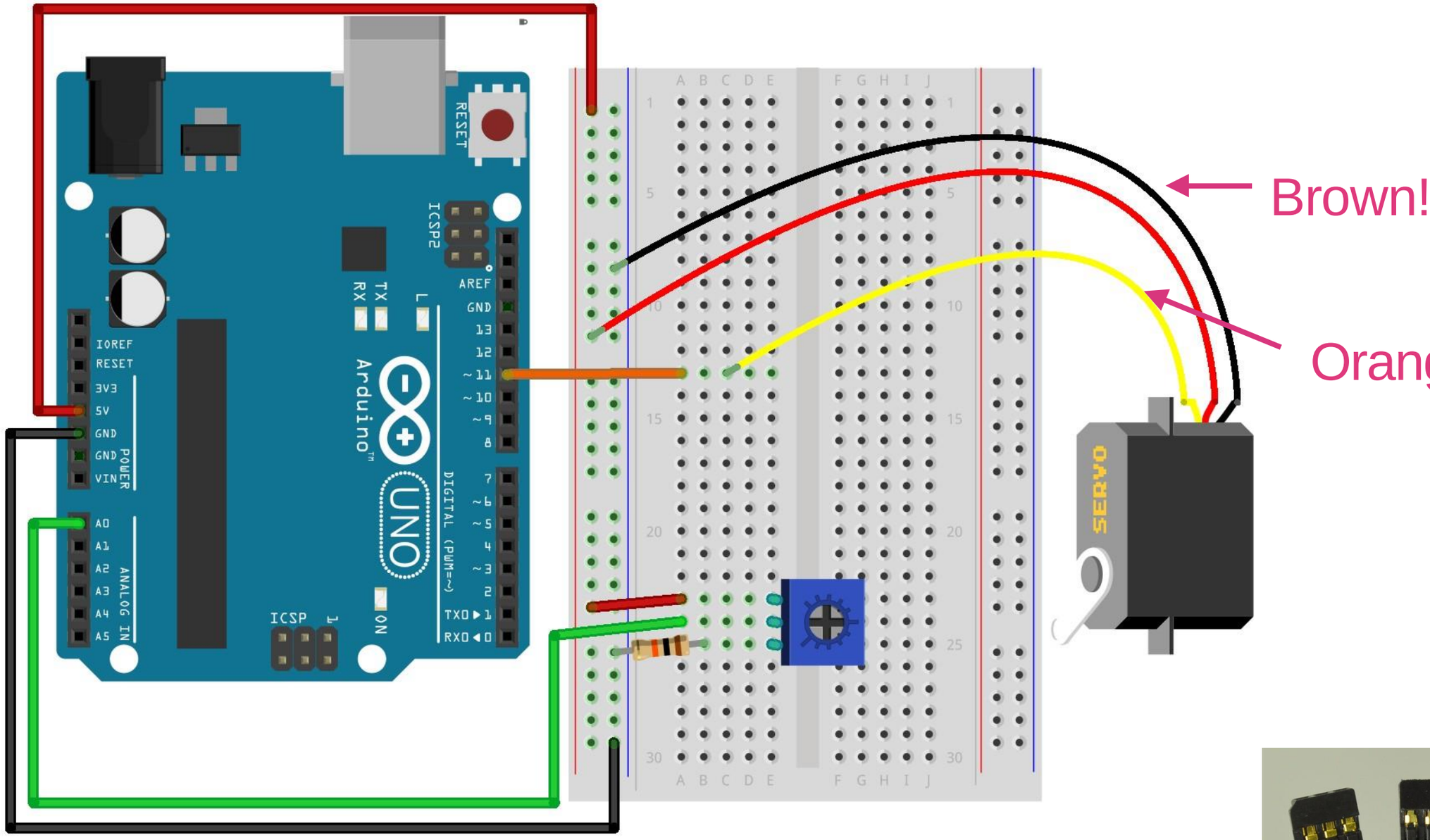
The function tone(…) changes the PWM frequency while keeping a 50% duty cycle!

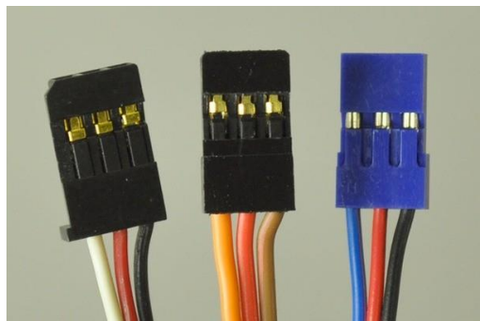


Getting Started with Arduino

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!

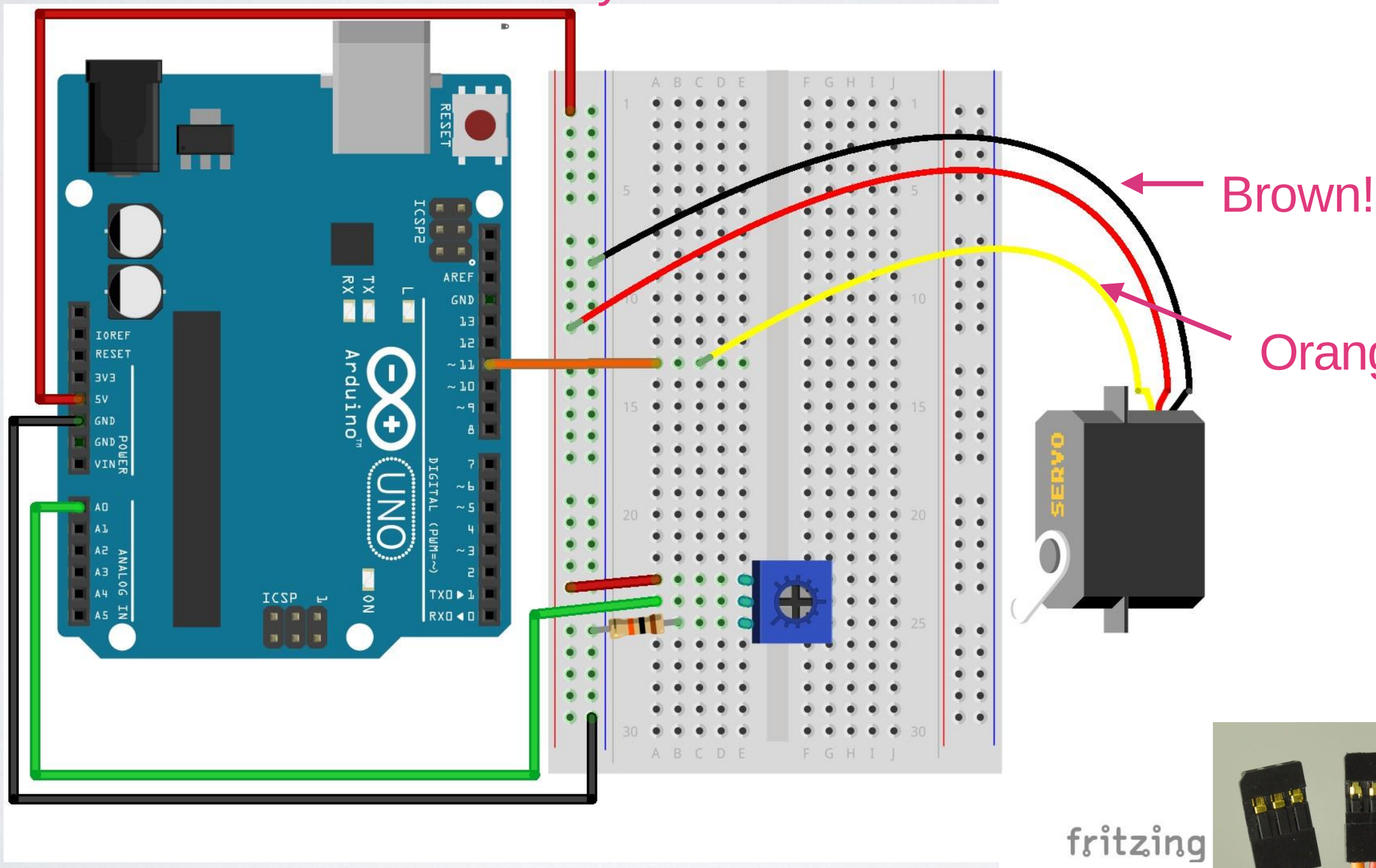




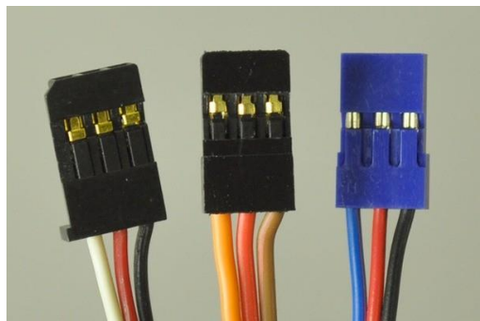
Getting Started with Arduino

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!



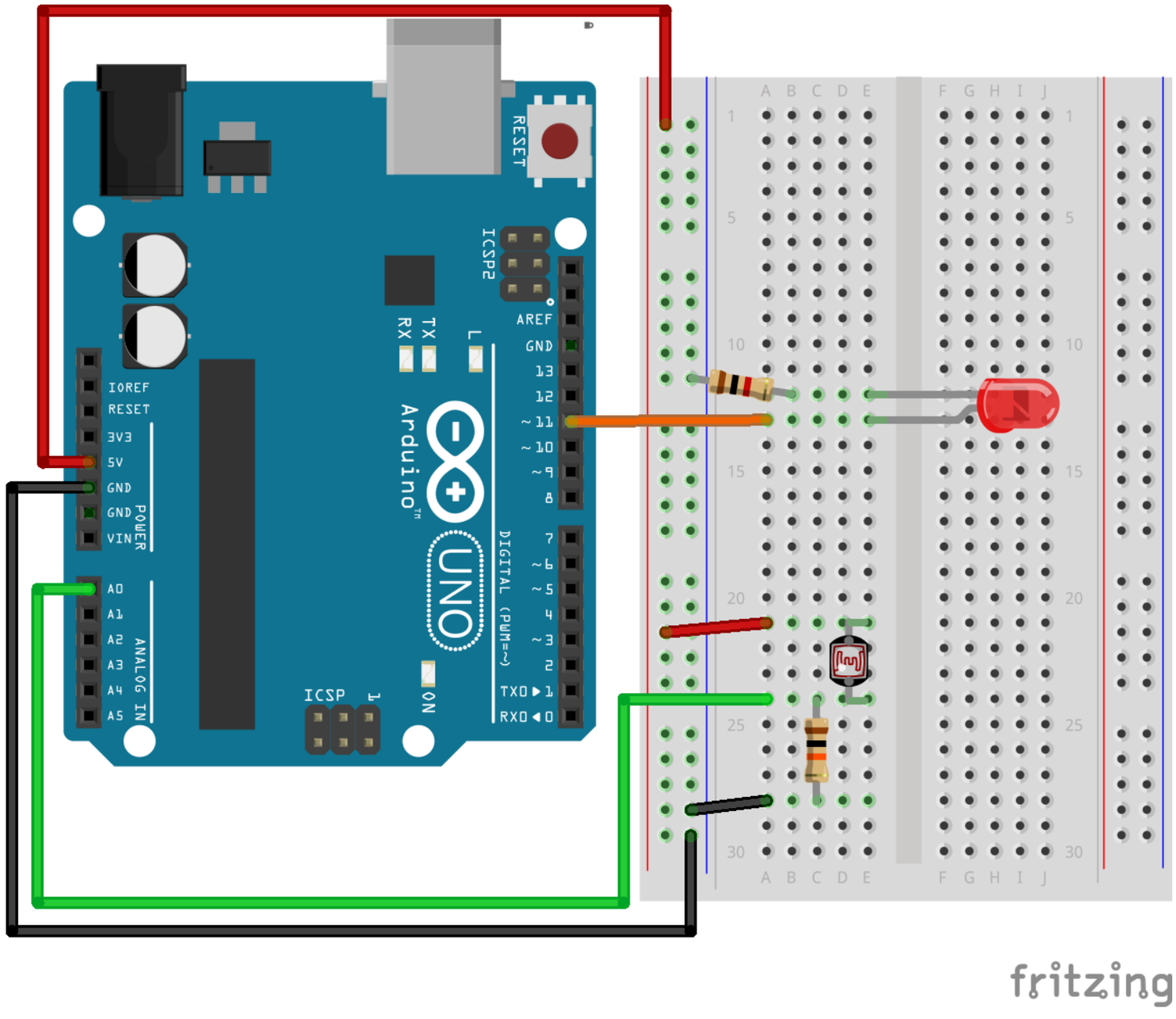
fritzing



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!



In a similar fashion some additional interfacing projects are also specified in this repository. Do check it out.

And for further queries regarding the track, share your doubts on our Repositiory’s Discussion tab