

# Robotic Arts Intro

Week 3: arduino is here  
[Connecting/Architecture/Digital IO]

# Arduino consists of 2 parts

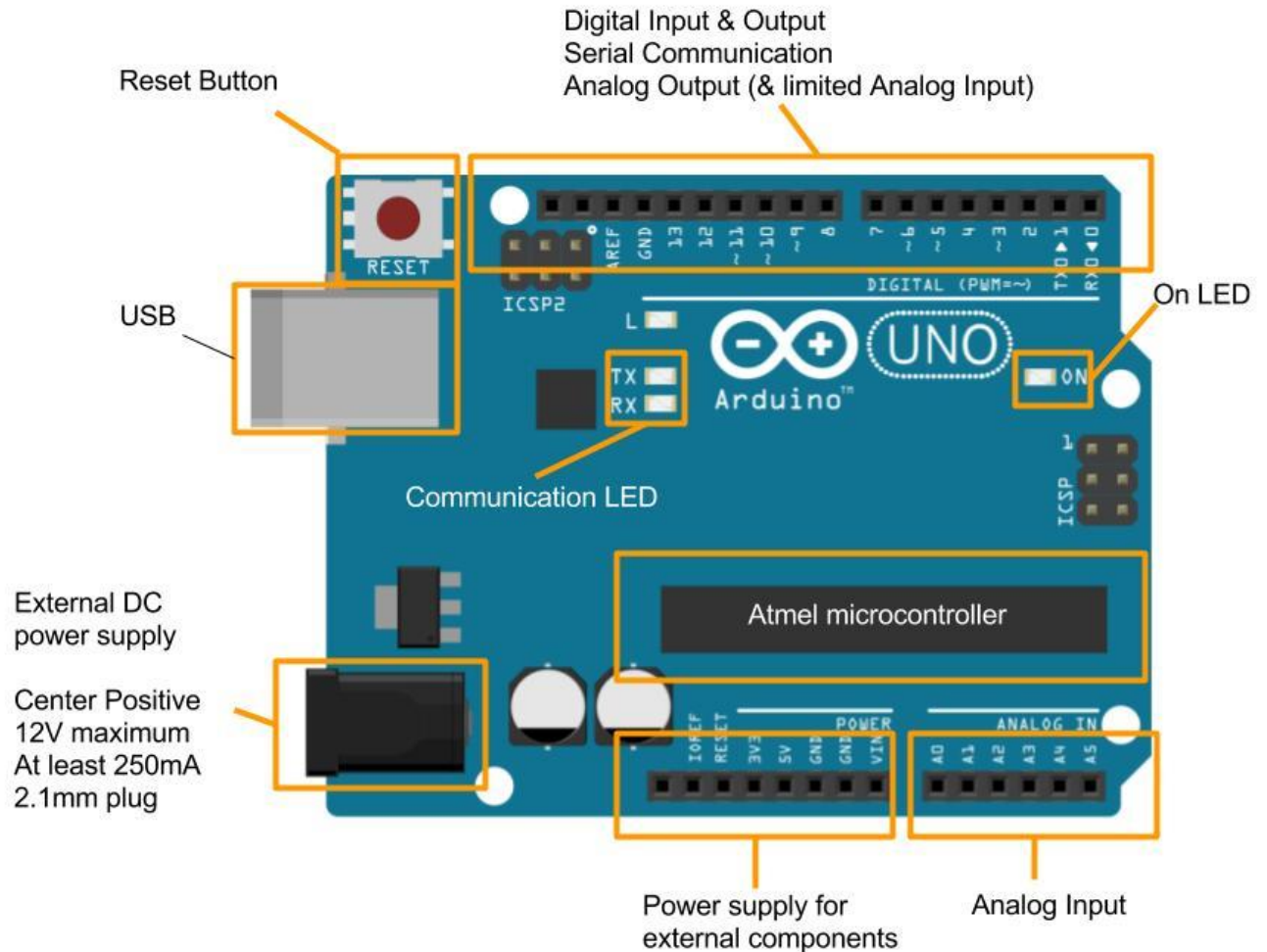
Hardware: The Arduino board

Software: The Arduino software

We write programs\* (code) using the Arduino software which we send (upload) to the Arduino hardware.

\* Program is another word for instructions.

# Arduino UNO architecture



# Arduino Software

Verify checks your code. Upload sends it to the Arduino.



## 3rd part... Arduino Community

A key part of Arduino's success has been due to the community behind it.

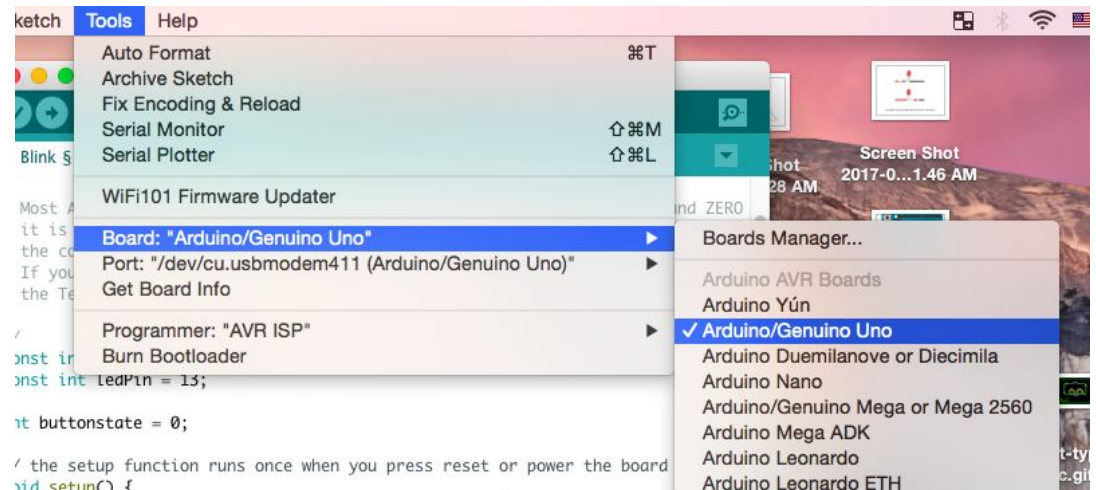
Not only does the Arduino host a fantastic programming reference, it also has a very active user base which help solve problems online:

<http://forum.arduino.cc/>

<http://playground.arduino.cc/>

# Connecting software with hardware

Select your board  
Arduino Uno



- Select your Arduino's port
- Plug Arduino in first!



# Syntax

## Comment

1. single line comment `// comment here`
2. Multi line comment `/* comment here */`

## Semicolon

1. At the end of every statement you will need a semicolon ;
2. Compiler recognizes semicolons

## Function

1. Series of operations packaged in one line
2. `void setup()` `void loop()`

## Brackets

1. `()` and `{}` contains values and variables

# Digital Signals

Digital signals are two state signals which can either be produced (OUTPUT) or received (INPUT) by the Arduino.

Digital signals are represented as:

5V or 0V

0 or 1

HIGH or LOW



# Tonight's Code

We'll be relying on examples which come with the Arduino software tonight.

These can be found in:

File > Examples > 01. Basics > Blink

*and*

File > Examples > 02. Digital > Button

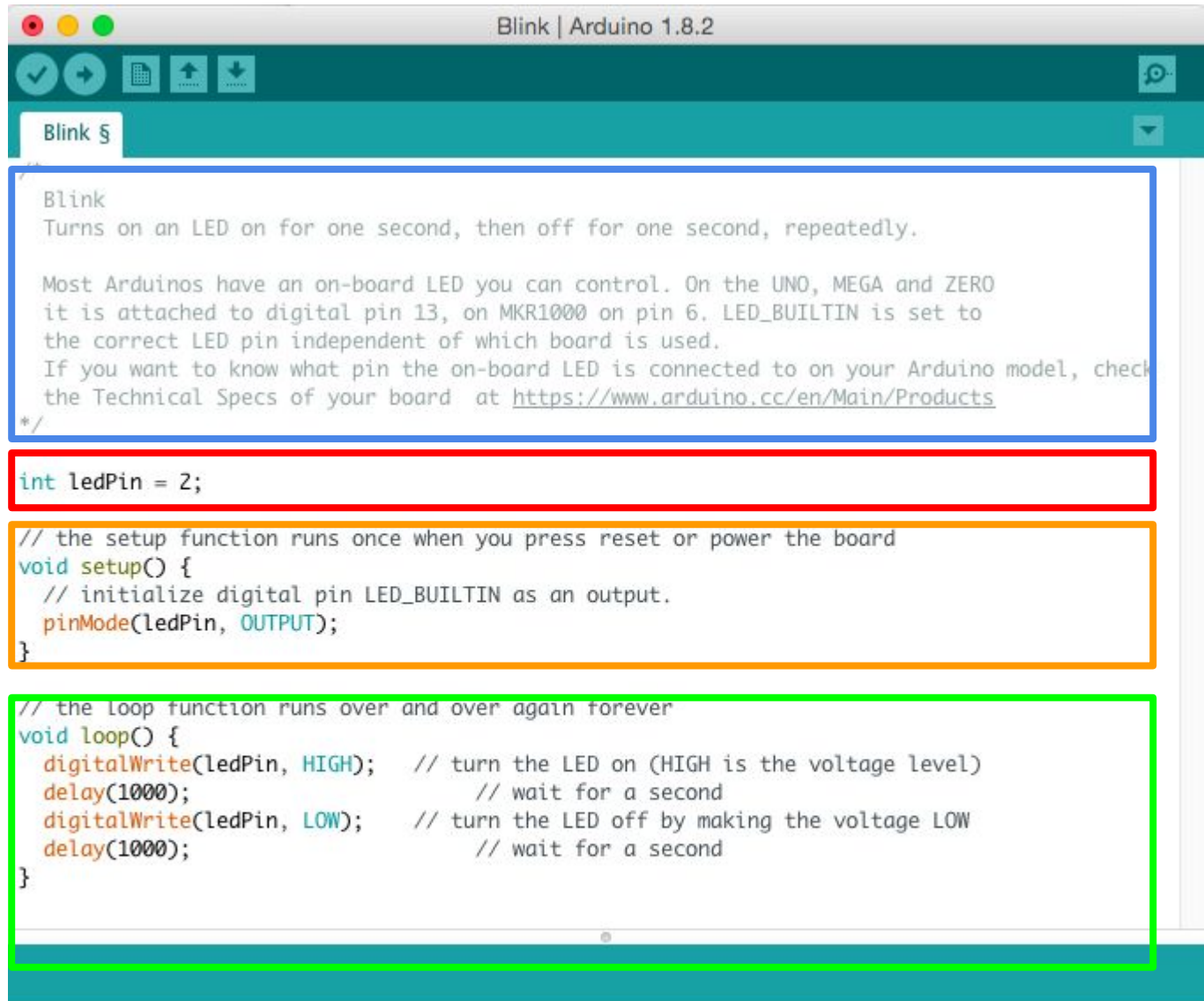
# Structure of code

Comments

Variables

Set up

Loop



```
Blink | Arduino 1.8.2
Blink §
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino model, check
 * the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
 */
int ledPin = 2;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(ledPin, OUTPUT);
}

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

# Circuit Diagrams

Getting power from the Arduino to the breadboard.

Connecting an LED.

Connecting LEDs in series to a single pin.

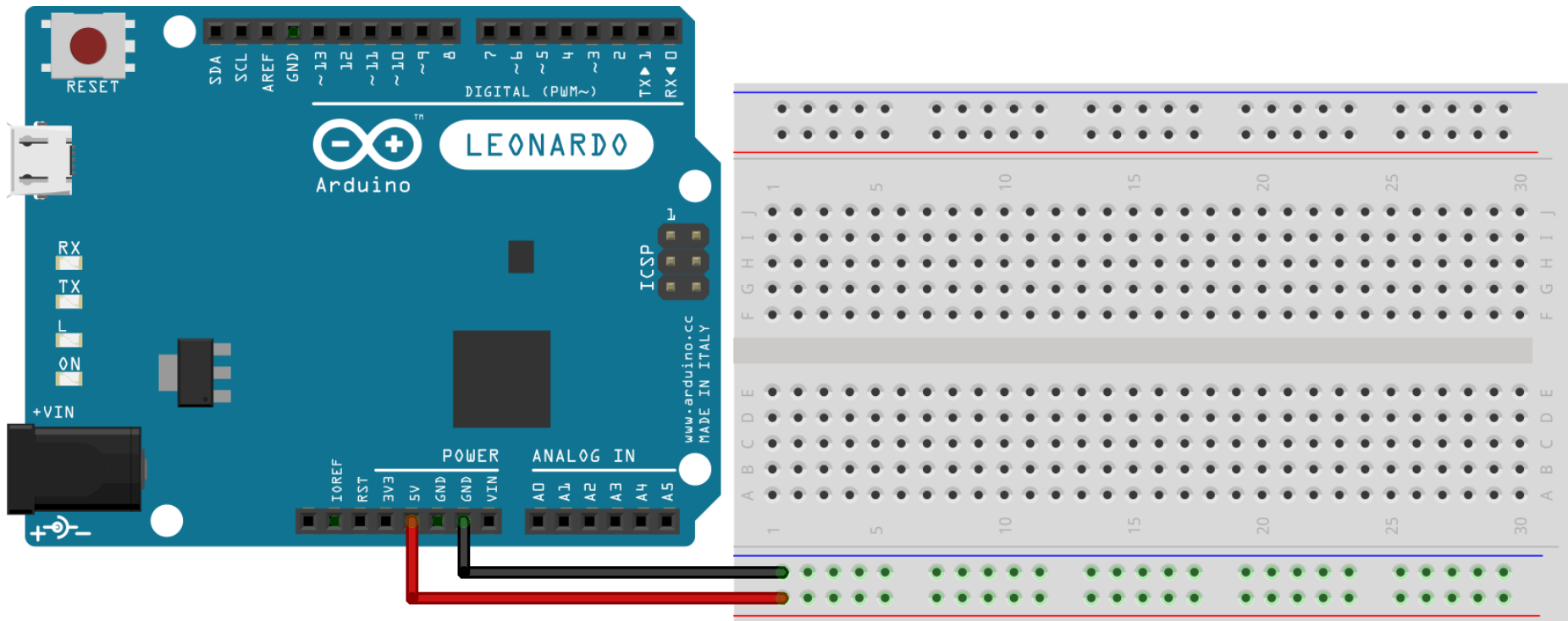
Connecting buttons/switches.

# Modify the pin number

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(2, OUTPUT);
}

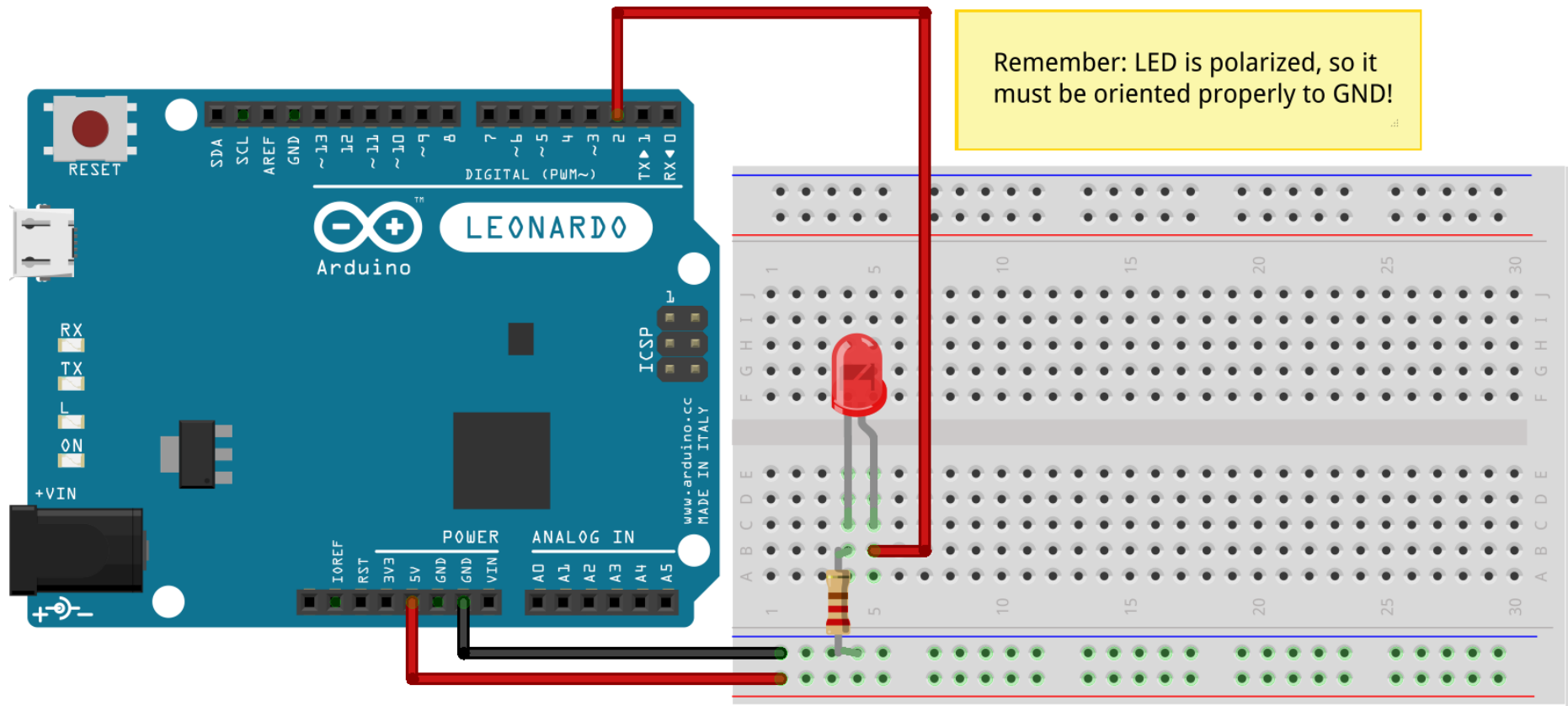
// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(2, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

# Getting Power from the Arduino



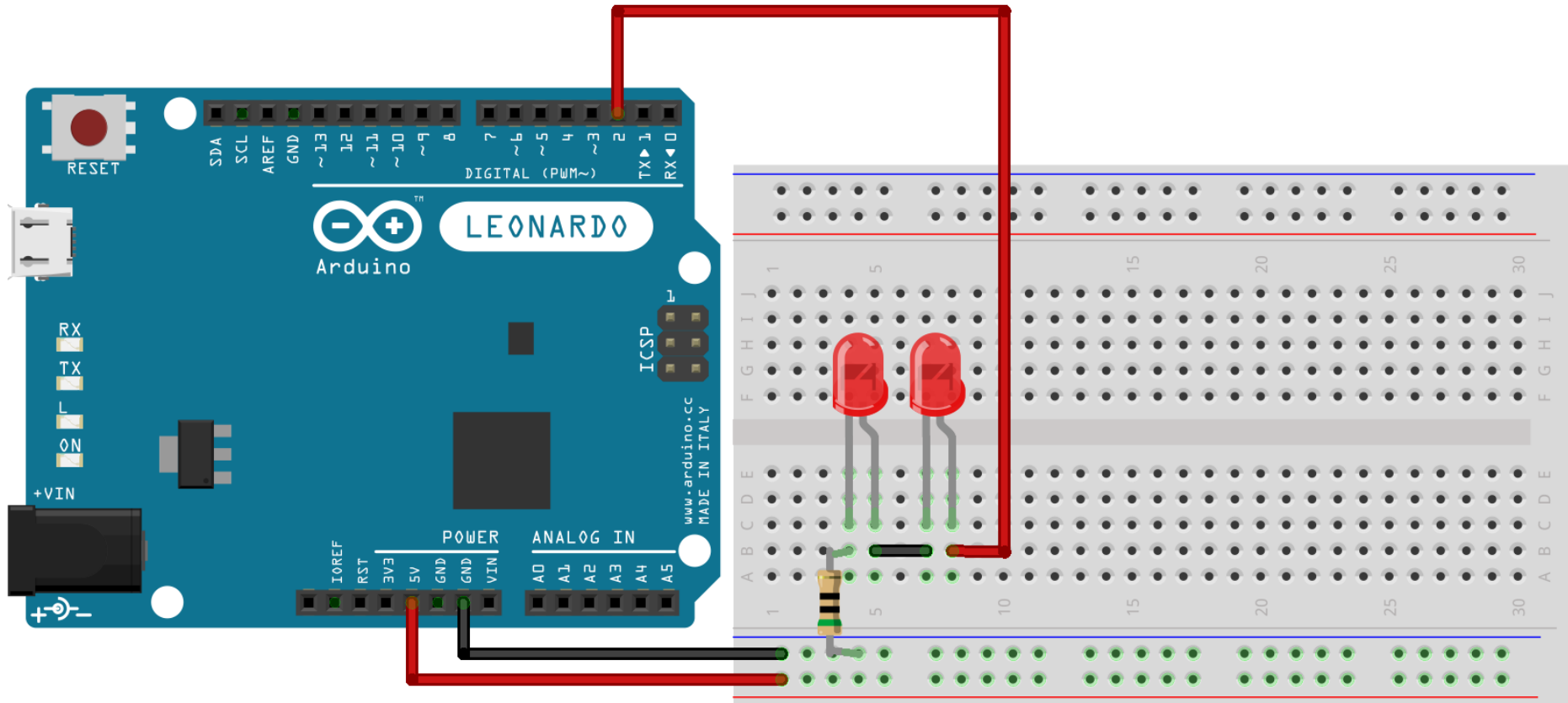
This adds 5V from the Arduino to the breadboard.

# Connecting an LED to the Arduino



A 150Ω resistor is optimal, but 220Ω is more common.

# LED Series Circuit



Ohm's Law can be used to find optimal resistance:

$$5V - (2V + 2V) / .02A = 50\Omega$$

# Exercise

1. Measure a voltage between two LED's anode and cathode
2. Vary the numbers in delay(####);
3. Blink an LED in two different delays speeds(one quick blink one slow blink)
4. Use two LEDs with two different output pins



# Variables

Data goes in here.



Variables are containers for data. They allow us to use the same data over and over again.

Variables can also act like labels, making it easier for us to keep track of things in our code.

# Declaring variables

```
int ledPin = 13;  
int buttonPin = 2;
```

- Creating a variable is referred to as “declaring a variable”.
- Variables are declared before void setup() so they can be used anywhere in our sketch.
- A variable can only hold one type of data, but you can choose what that is (and create as many variables as you’d like!)
- A variable name cannot contain any spaces.

# Data types

- Computers store information as data.
- There are specific types of data for storing different kinds of information.
- We'll commonly use one called `int` (integer).

\*An `int` is a whole number (no decimals) between -32,768 & 32,767

# Using variables

To use a variable we just refer to it by name:

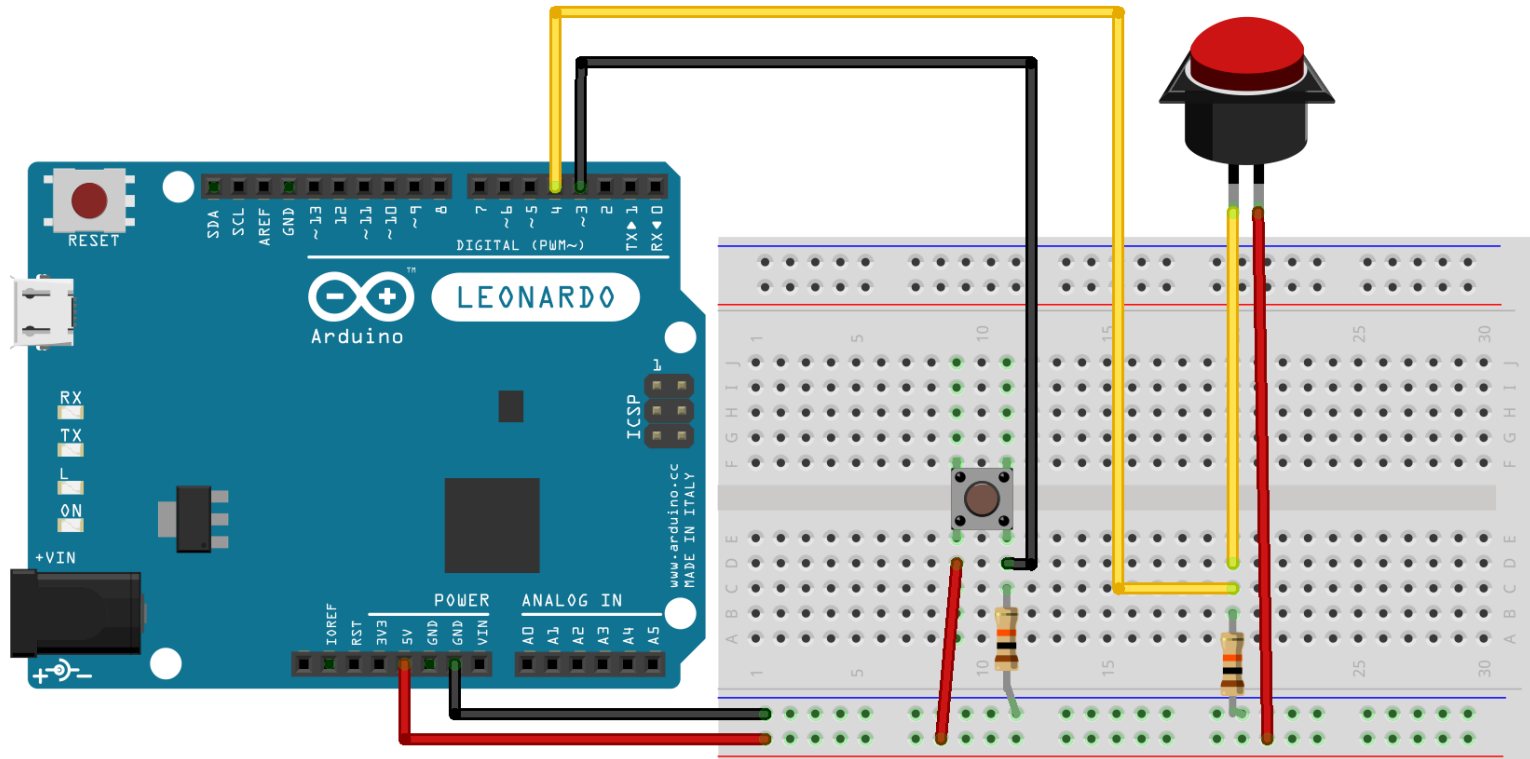
```
buttonState = digitalRead(buttonPin);
```

```
digitalWrite(ledPin, HIGH);
```

\*HIGH and LOW are actually variables which are used to store 1 and 0.

# Connecting Buttons/Switches

A button/switch can have either 2 or 4 legs.



10KΩ is a standard pull-down resistor

The pull-down resistor ensures that we get a LOW digital signal when the button is not pressed.

# LED + Button Combination

