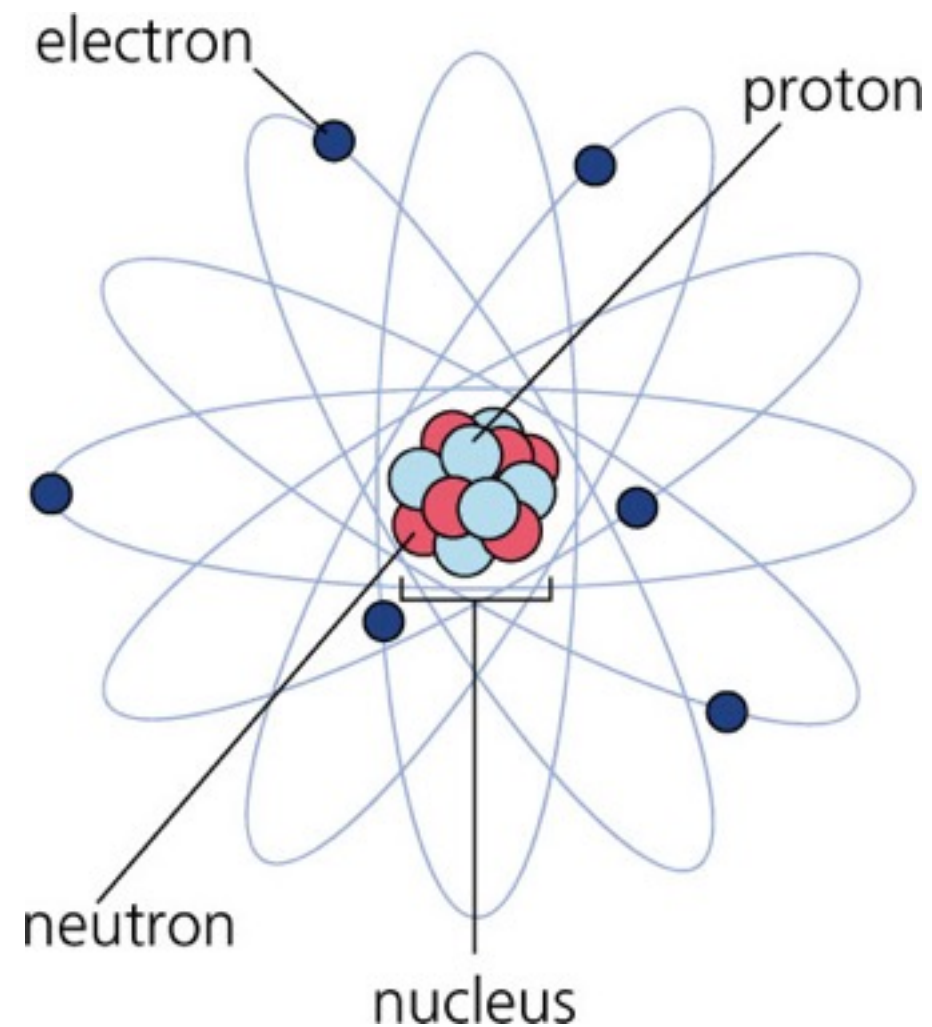


WELCOME TO THE DUCK TAPE FILES

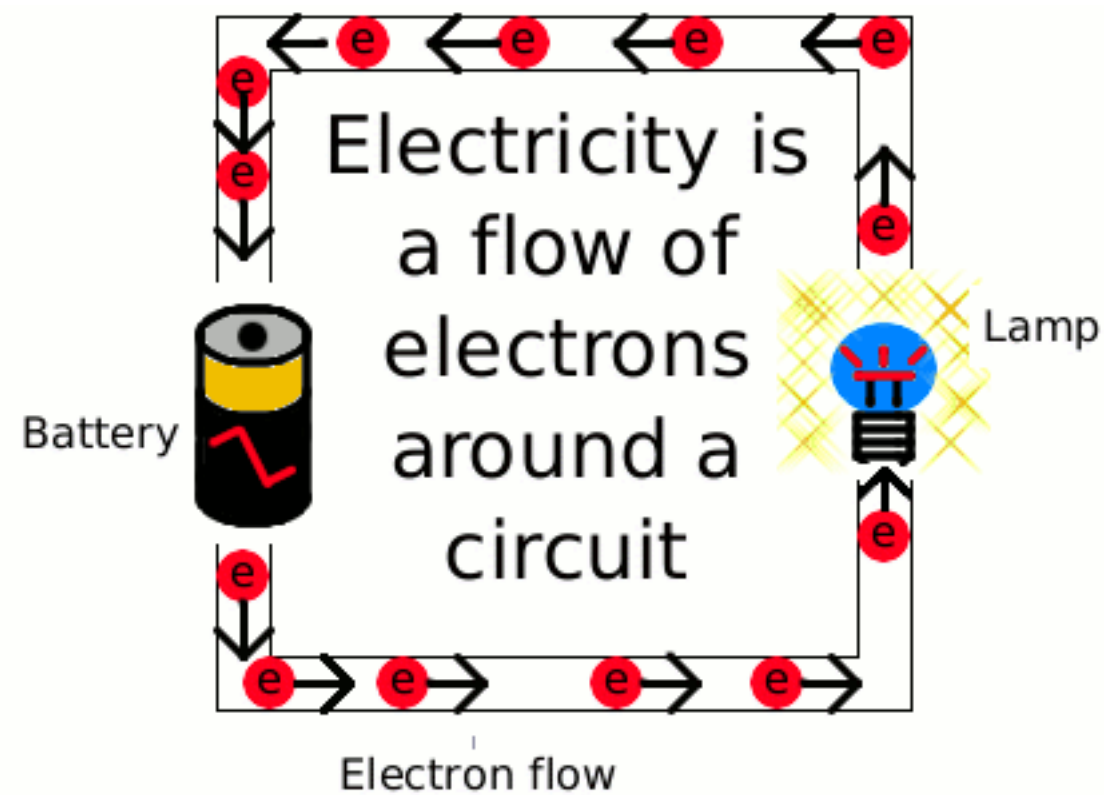
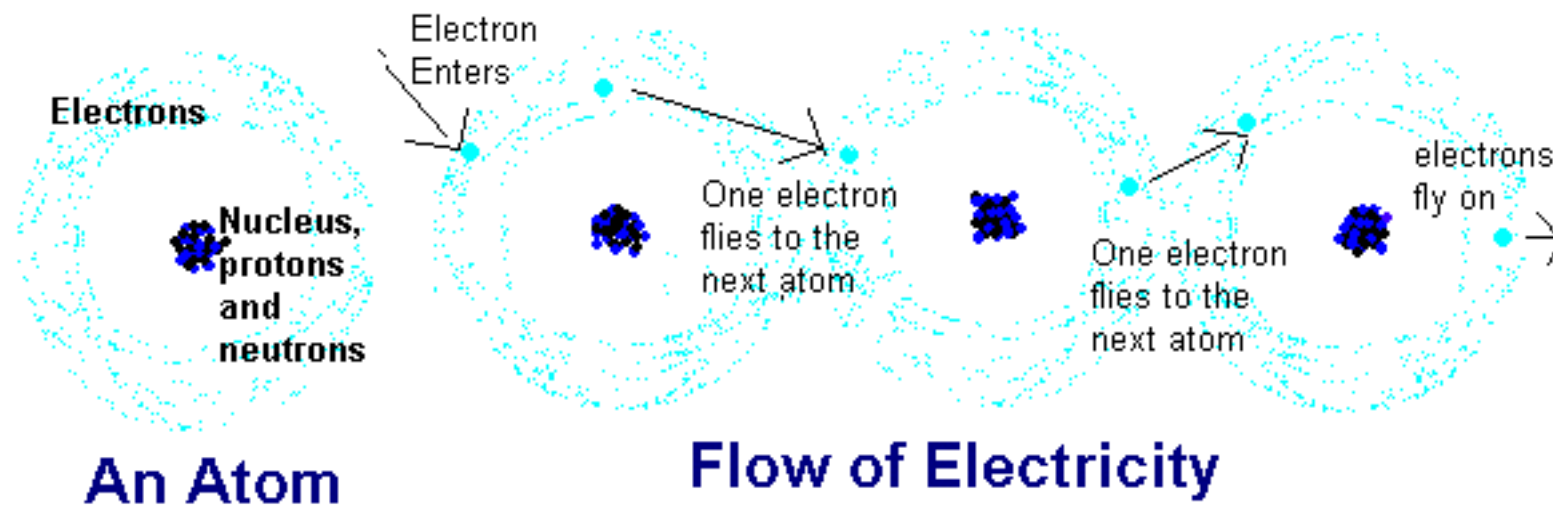


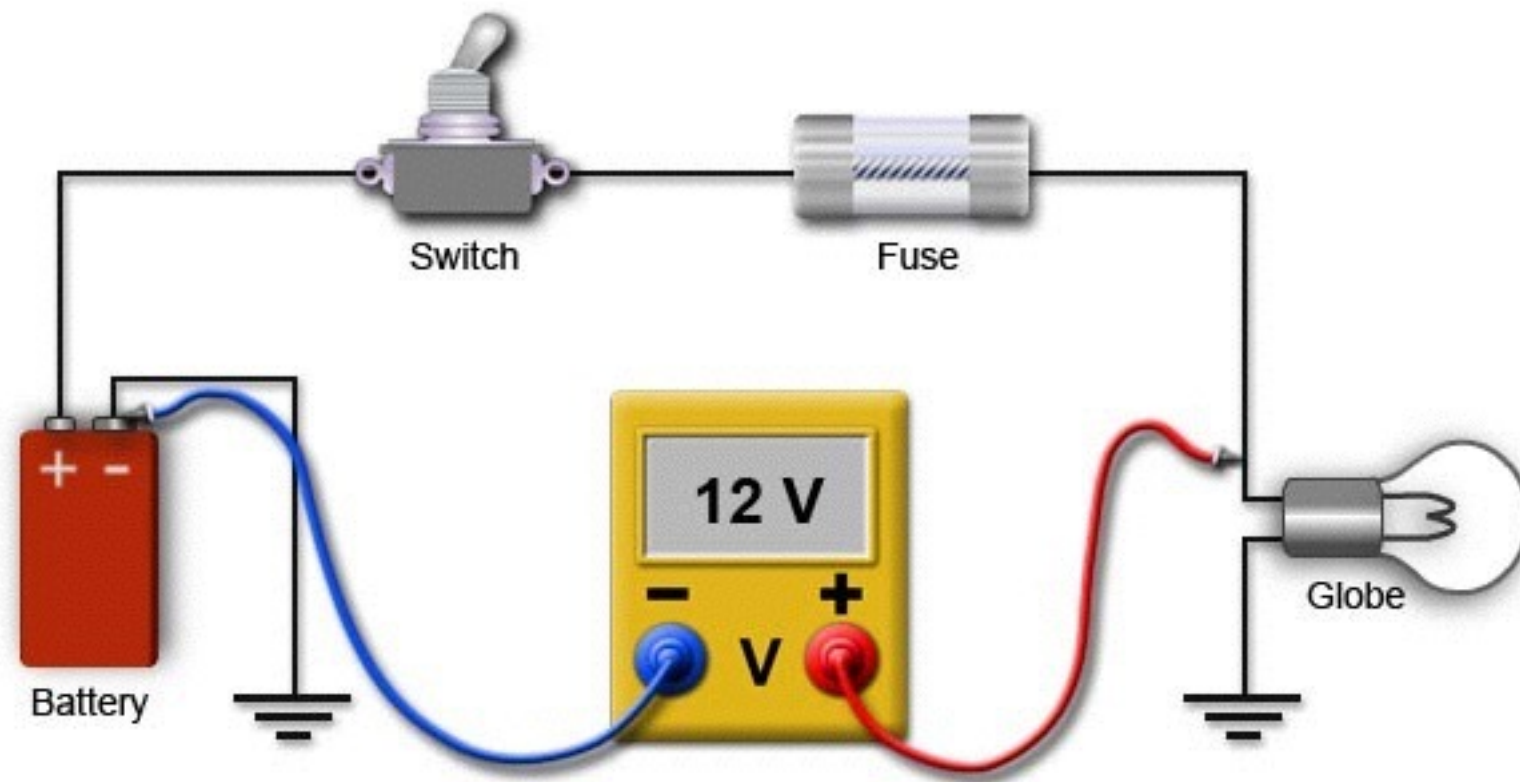
FAILURE IS **LEARNING**

It's all about balancing the
electrons....



Nature likes
balance





Voltage (V)

also a volt or an EMF or as potential difference . It gets used up as it goes around the circuit. Think of it like your energy if you're running a race.

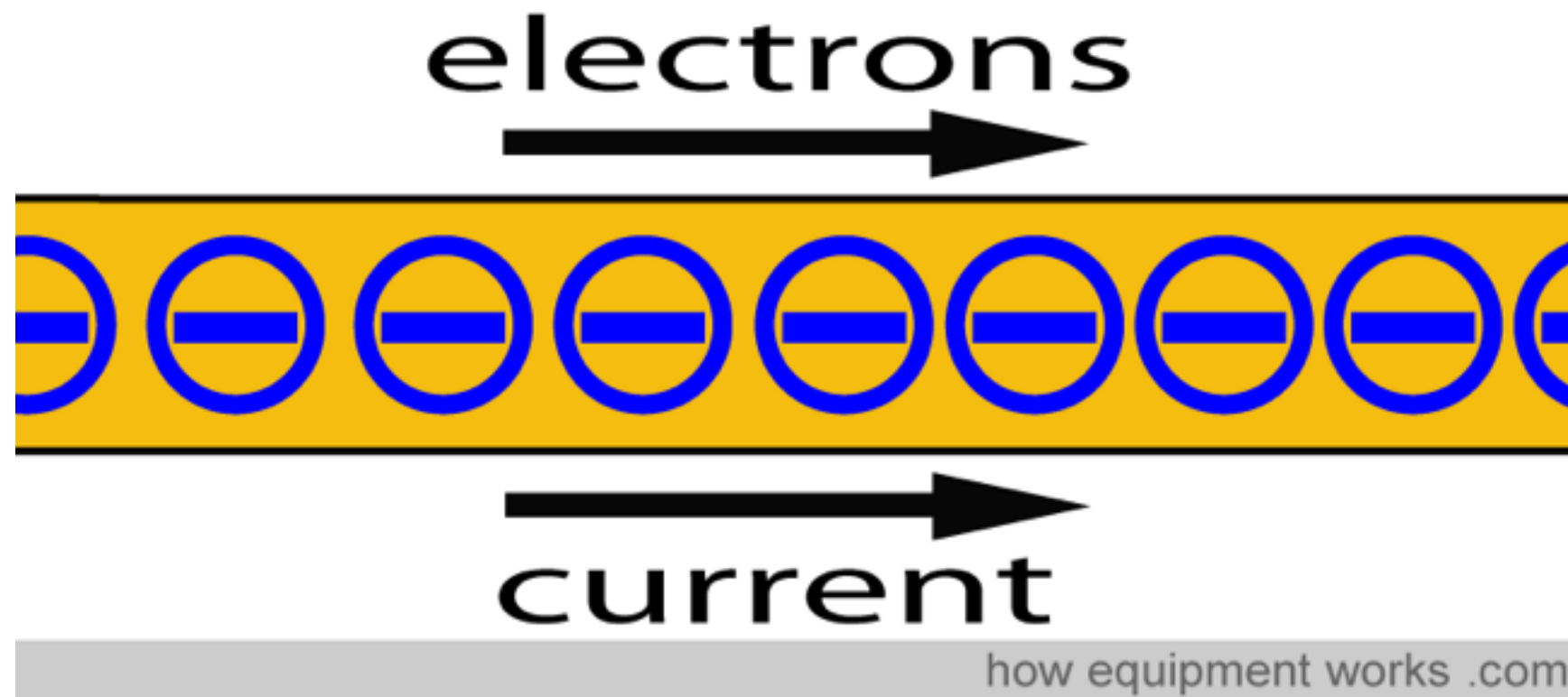
1. What's an electron?
2. What's Arduino?
3. What's electricity?



**KEEP
CALM**

ITS

**QUESTION
TIME**

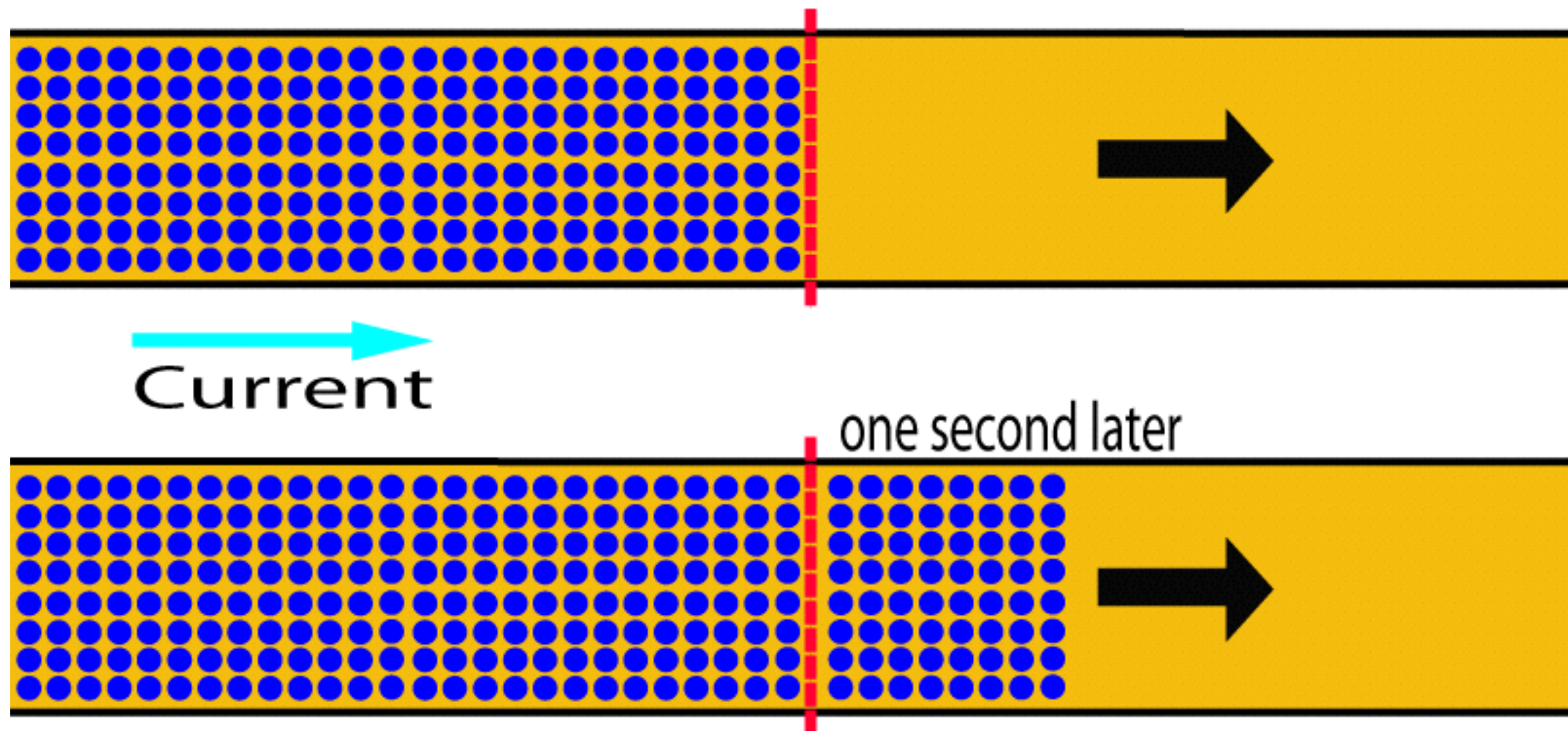


Current (I)

Electrons can be thought of as negatively charged “particles”. The movement of these electrons is called current.

Amperes (Amps)

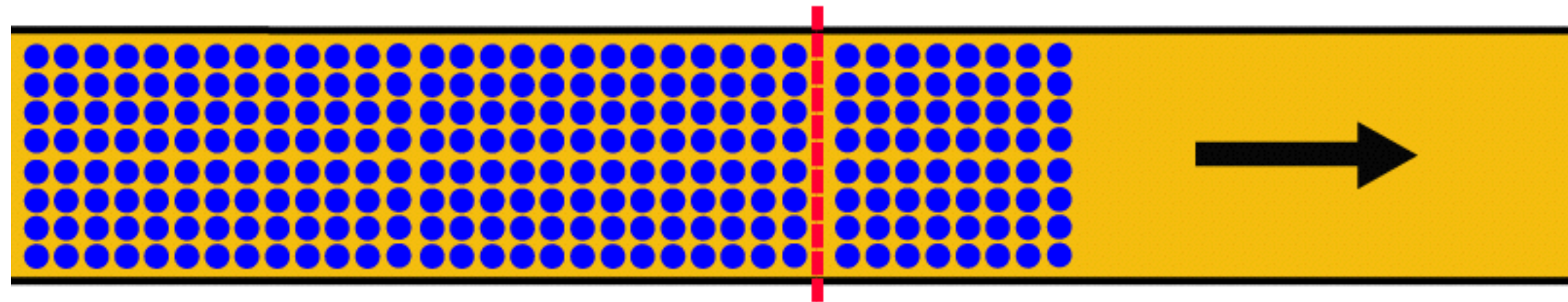
Current is measured in units called 'amperes'. The number of amperes in a wire relates to how many electrons pass a cross section of the wire per second.



Prefix	Symbol	1000 ^m	10 ⁿ	Decimal	Short scale	Long scale	Since ^[n 1]
yotta	Y	1000 ⁸	10 ²⁴	1 000 000 000 000 000 000 000 000 000	Septillion	Quadrillion	1991
zetta	Z	1000 ⁷	10 ²¹	1 000 000 000 000 000 000 000 000	Sextillion	Trilliard	1991
exa	E	1000 ⁶	10 ¹⁸	1 000 000 000 000 000 000 000	Quintillion	Trillion	1975
peta	P	1000 ⁵	10 ¹⁵	1 000 000 000 000 000	Quadrillion	Billiard	1975
tera	T	1000 ⁴	10 ¹²	1 000 000 000 000	Trillion	Billion	1960
giga	G	1000 ³	10 ⁹	1 000 000 000	Billion	Milliard	1960
mega	M	1000 ²	10 ⁶	1 000 000	Million		1960
kilo	k	1000 ¹	10 ³	1 000	Thousand		1795
hecto	h	1000 ^{2/3}	10 ²	100	Hundred		1795
deca	da	1000 ^{1/3}	10 ¹	10	Ten		1795
		1000 ⁰	10 ⁰	1	One		–
deci	d	1000 ^{-1/3}	10 ⁻¹	0.1	Tenth		1795
centi	c	1000 ^{-2/3}	10 ⁻²	0.01	Hundredth		1795
milli	m	1000 ⁻¹	10 ⁻³	0.001	Thousandth		1795
micro	μ	1000 ⁻²	10 ⁻⁶	0.000 001	Millionth		1960
nano	n	1000 ⁻³	10 ⁻⁹	0.000 000 001	Billionth	Milliardth	1960
pico	p	1000 ⁻⁴	10 ⁻¹²	0.000 000 000 001	Trillionth	Billionth	1960
femto	f	1000 ⁻⁵	10 ⁻¹⁵	0.000 000 000 000 001	Quadrillionth	Billiardth	1964
atto	a	1000 ⁻⁶	10 ⁻¹⁸	0.000 000 000 000 000 001	Quintillionth	Trillionth	1964
zepto	z	1000 ⁻⁷	10 ⁻²¹	0.000 000 000 000 000 000 001	Sextillionth	Trilliardth	1991
yocto	y	1000 ⁻⁸	10 ⁻²⁴	0.000 000 000 000 000 000 000 001	Septillionth	Quadrillionth	1991

A current of one ampere relates to a certain number of electrons passing a cross section of the wire in one second. This number is absolutely huge, and please don't bother to remember it. The wire below is carrying one ampere of current.

A wire carrying 1 ampere carries about



6,241,000,000,000,000,000
electrons across it per second !

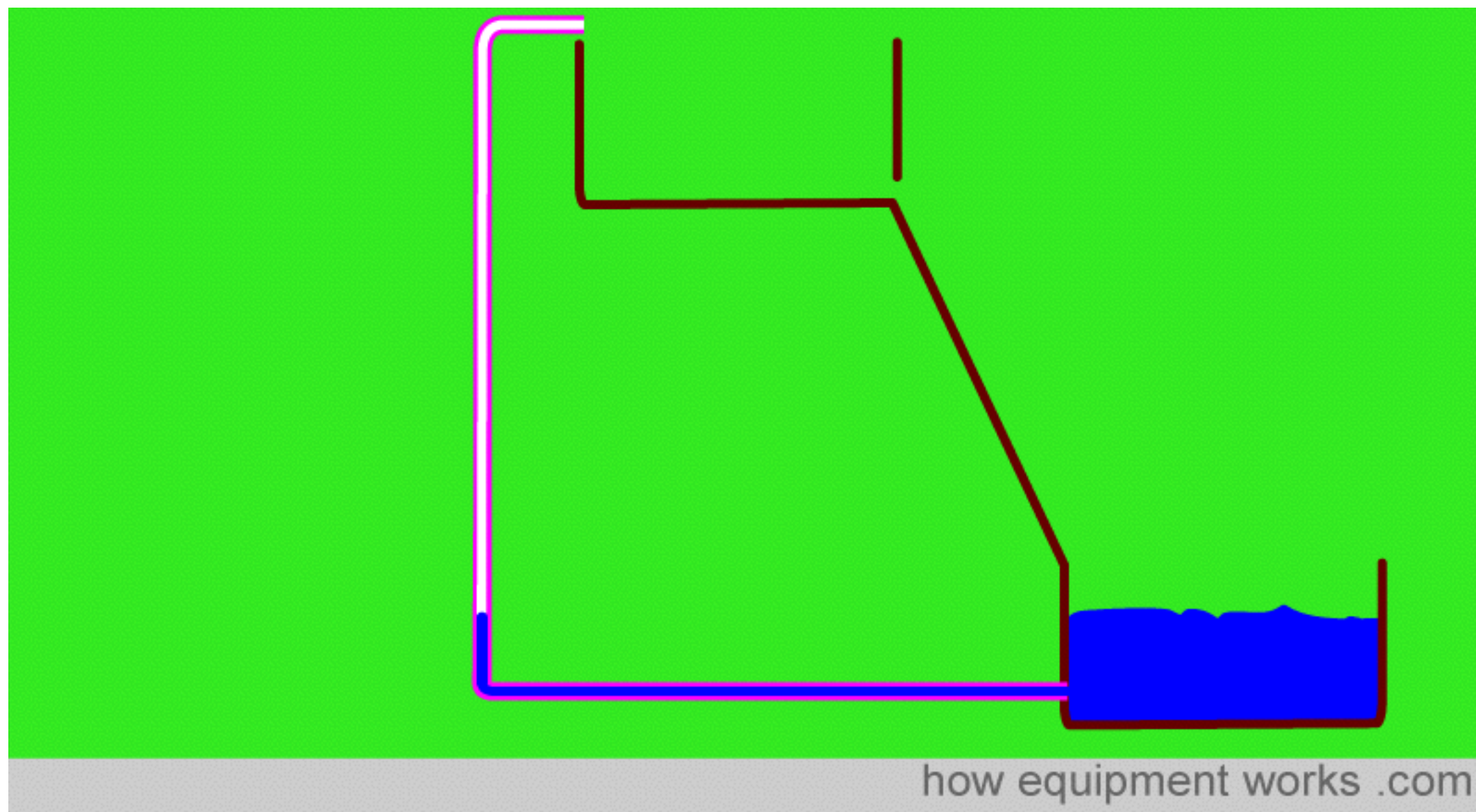
1. What's V?
2. What's I?
3. What's an R?



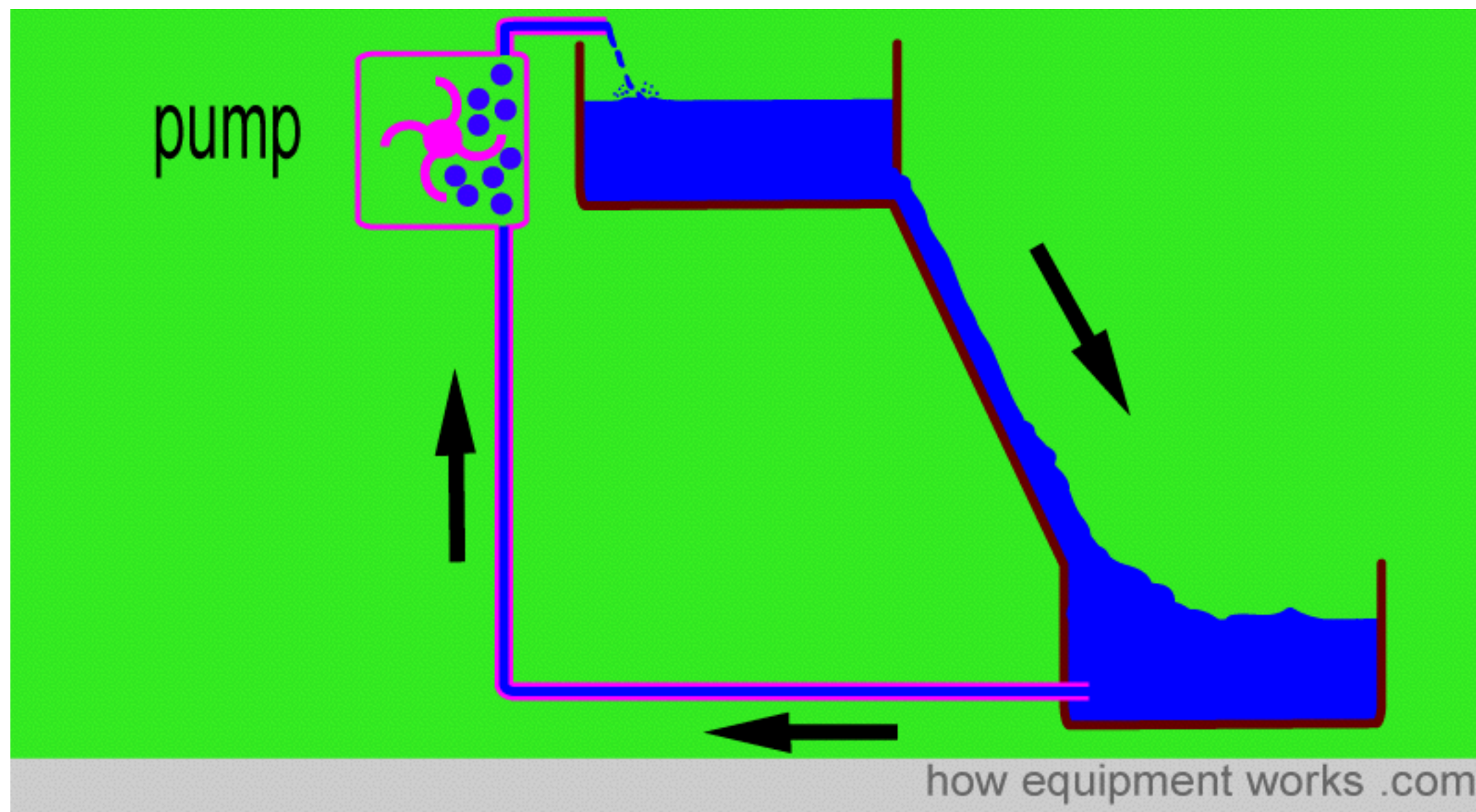
**It's convention to
refer to current as
flowing downhill**

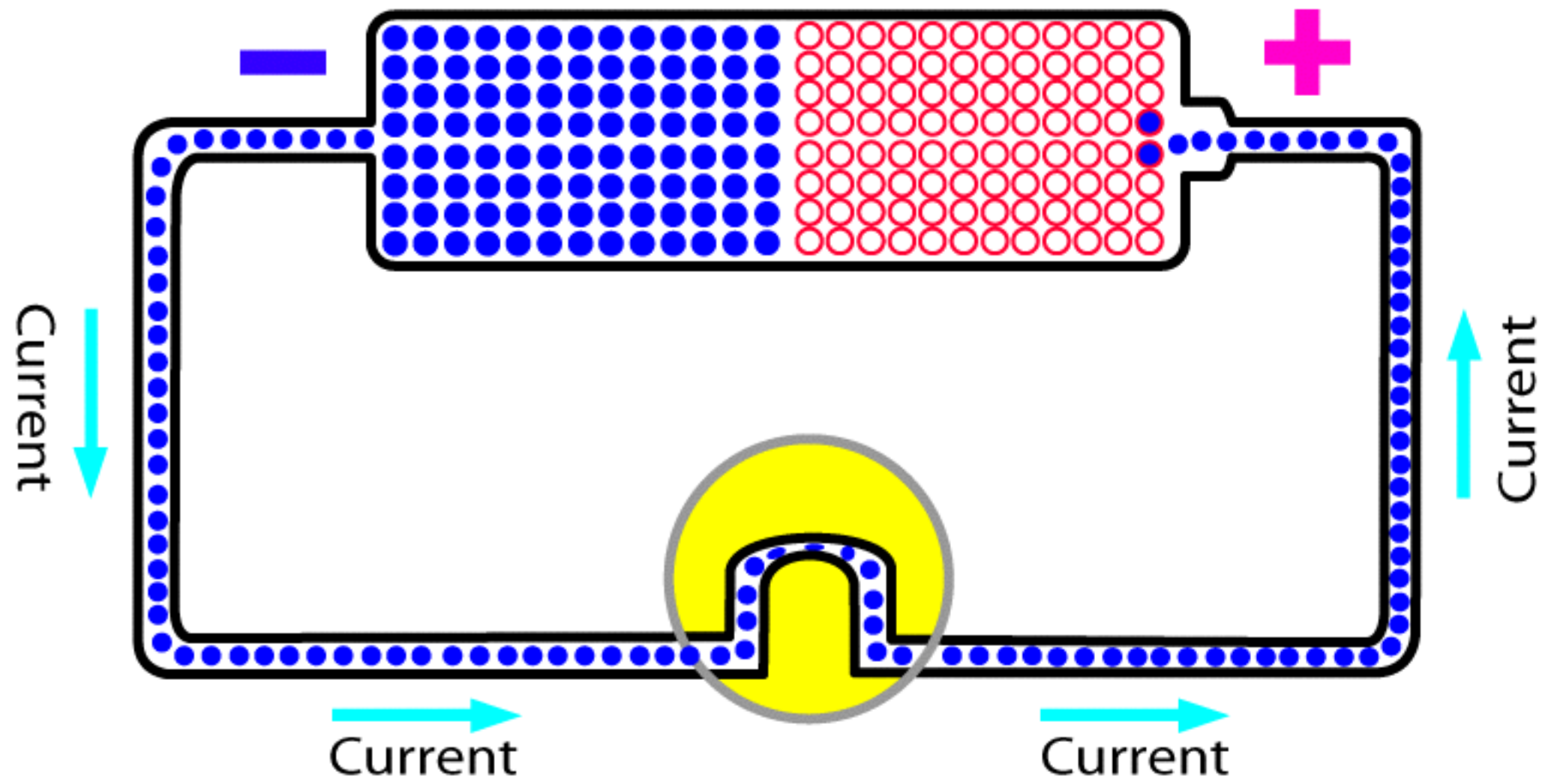
even though the
electrons are going
from negative to
positive. Why?
Because metaphor.

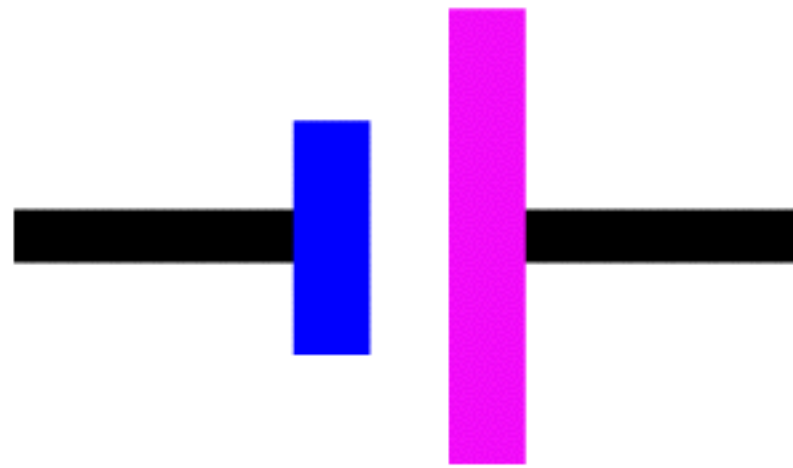




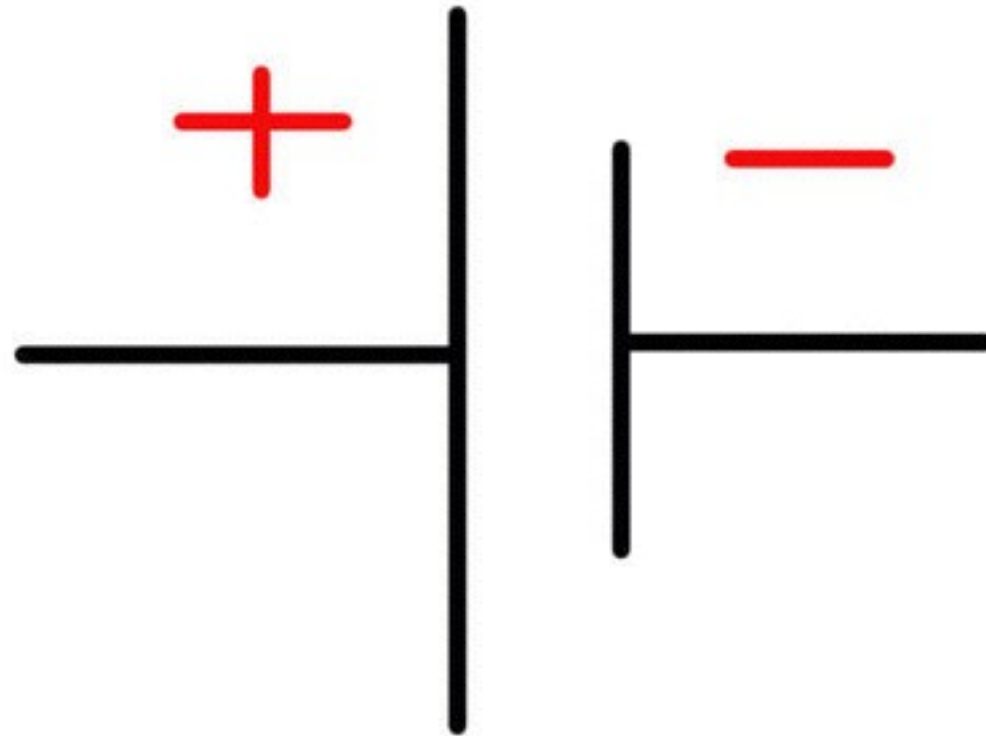
What's a circuit?







negative positive



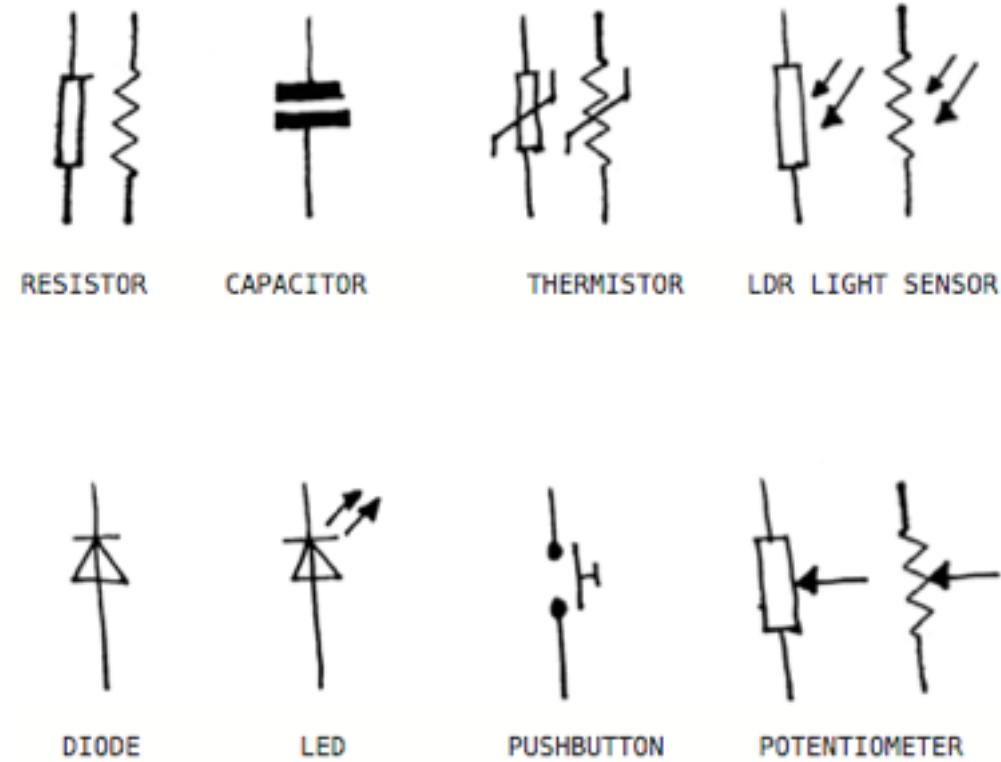
BATTERY



longer lead is positive.

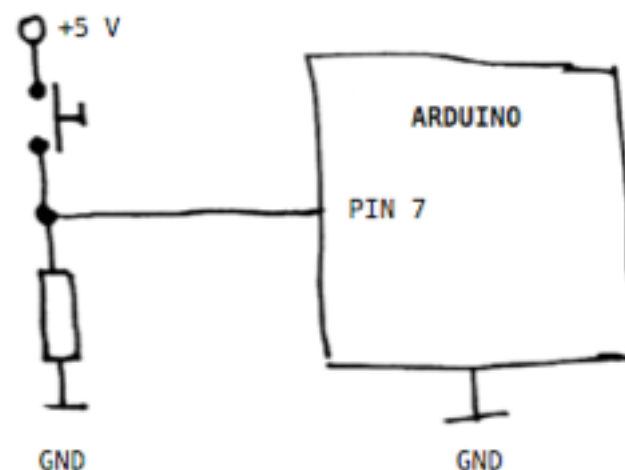
<http://www.instructables.com/id/HOW-TO-READ-CIRCUIT-DIAGRAMS/>

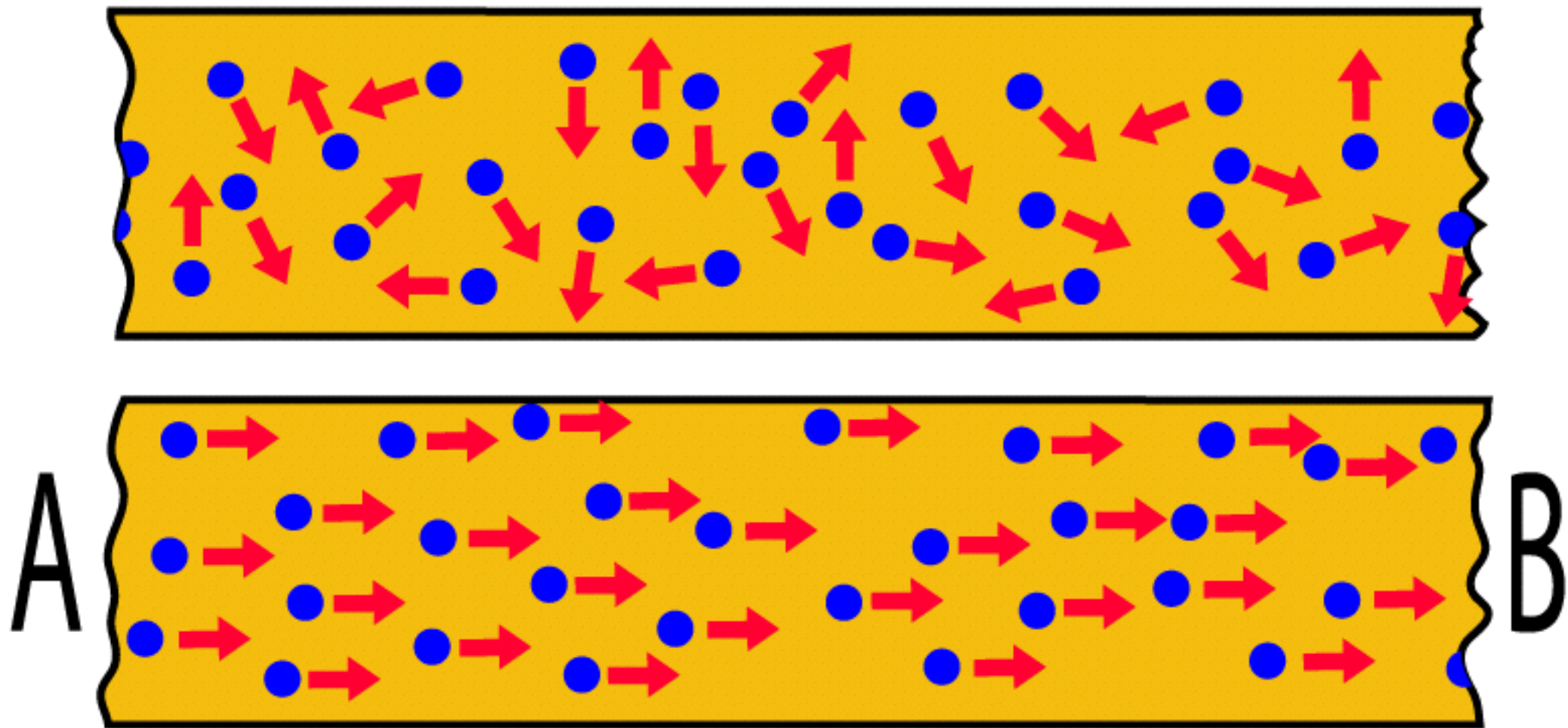
This is all you need to understand basic schematics. Here is a more comprehensive list of symbols and their meanings:



You may encounter variations in these symbols (for example, both variants of resistor symbols are shown here). See en.wikipedia.org/wiki/Electronic_symbol for a larger list of electronics symbols. By convention, diagrams are drawn from left to right. For example, a radio would be drawn starting with the antenna on the left, following the path of the radio signal as it makes its way to the speaker (which is drawn on the right).

The following schematic describes the push-button circuit shown earlier in this book:

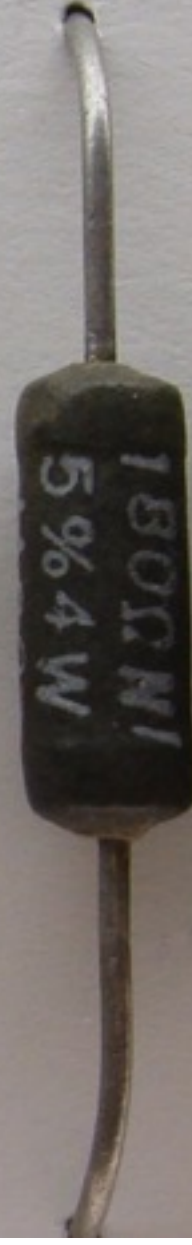




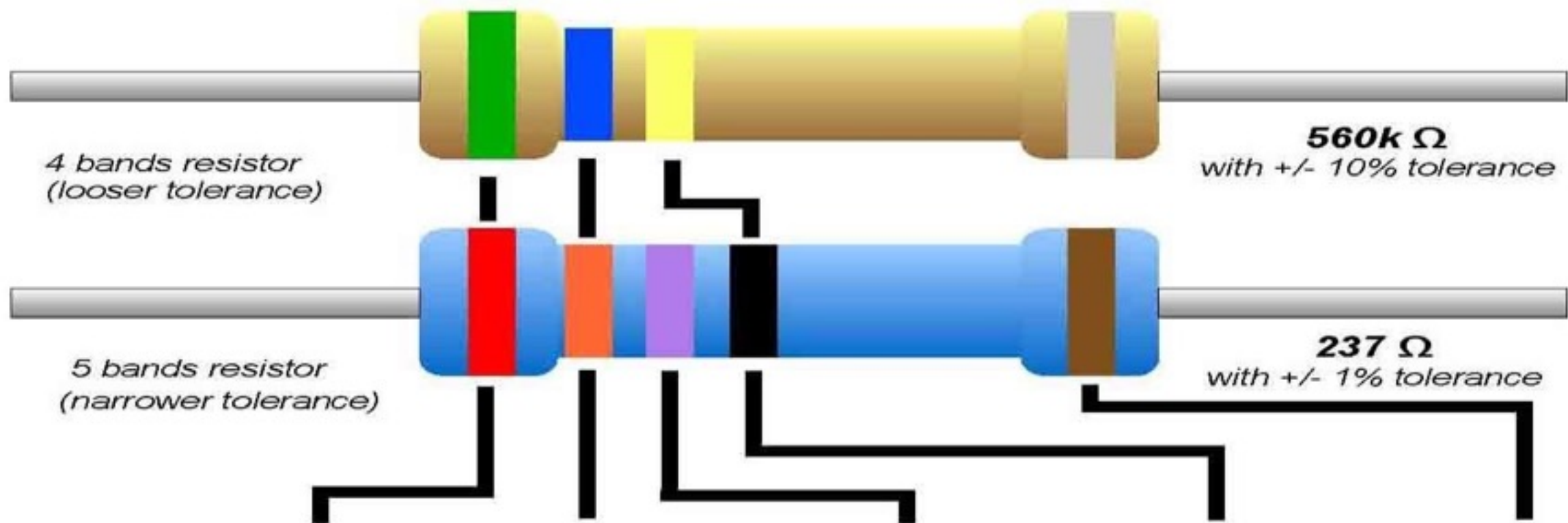


Resistance (R)

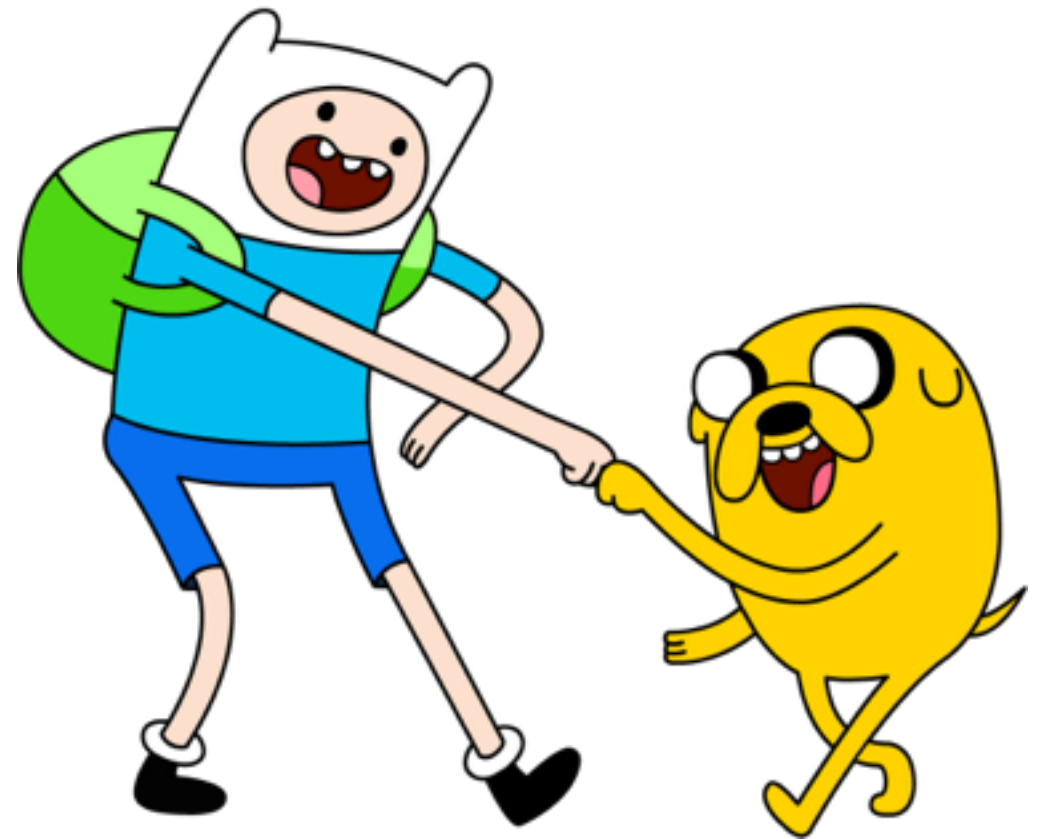
How easily can the current flow?



Resistor Color Code



Color	1 st Band	2 nd Band	3 rd Band	Multiplier	Tolerance
Black	0	0	0	$\times 1 \Omega$	
Brown	1	1	1	$\times 10 \Omega$	$\pm 1\%$
Red	2	2	2	$\times 100 \Omega$	$\pm 2\%$
Orange	3	3	3	$\times 1K \Omega$	
Yellow	4	4	4	$\times 10K \Omega$	
Green	5	5	5	$\times 100K \Omega$	$\pm 5\%$
Blue	6	6	6	$\times 1M \Omega$	$\pm .25\%$
Violet	7	7	7	$\times 10M \Omega$	$\pm .1\%$
Grey	8	8	8		$\pm .05\%$
White	9	9	9		
Gold				$\times .1 \Omega$	$\pm 5\%$
Silver				$\times .01 \Omega$	$\pm 10\%$



1. What's a circuit?
2. Which foot of the LED is positive
3. What's a resistor?

Ohm's law

- Ohm's law says *that the tension is equal to the product of the intensity and the resistance*

$$V = R \cdot I$$

- *This is equivalent to:*

$$I = V/R \leftrightarrow R = V/I$$



How much power is spent?

$$P = VI$$

power is referred to as watts W

The power goes into heat usually or sometimes
mechanical work (like a robot motor)

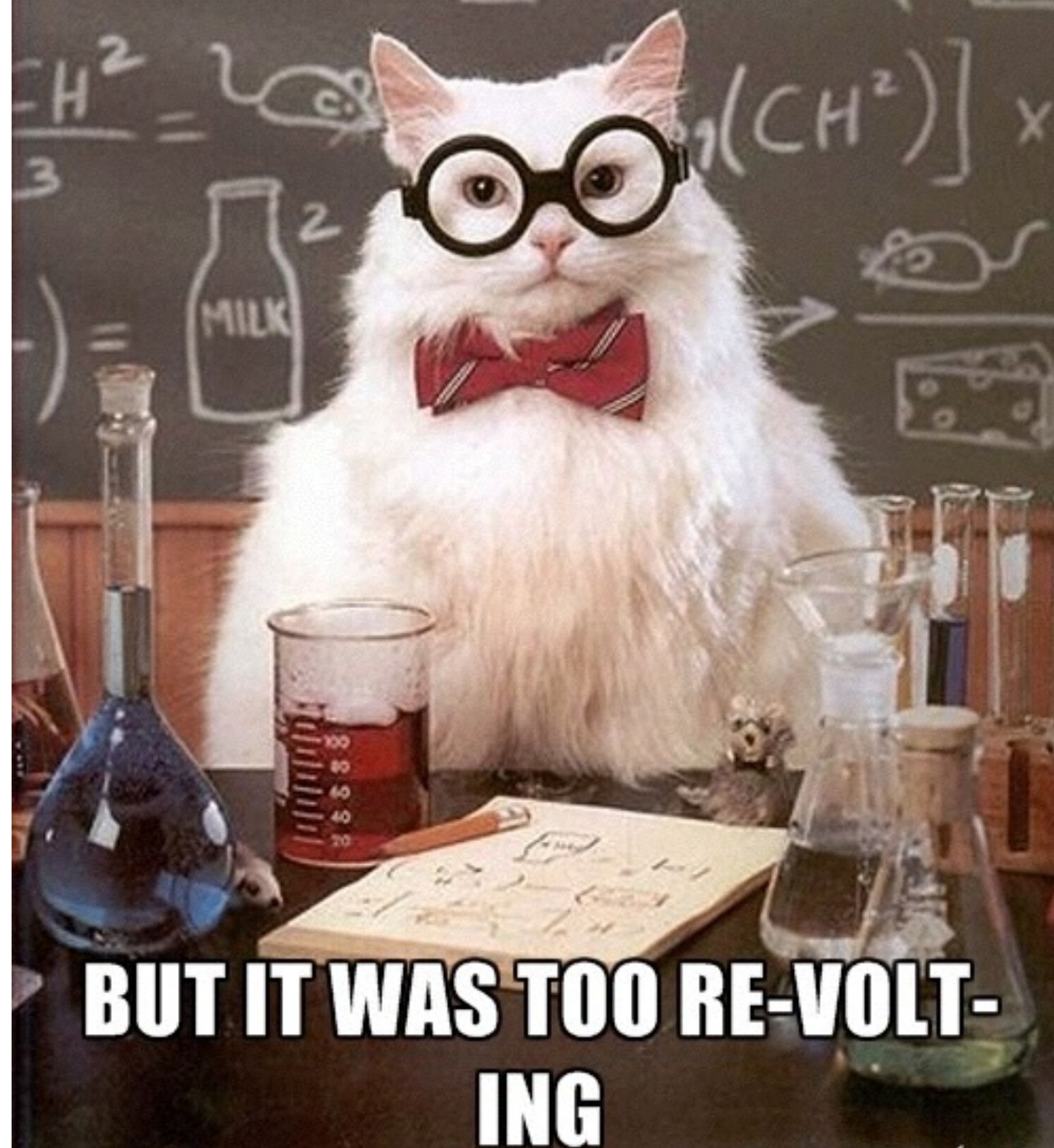
this is why your laptop can get hot

mechanical work = **motor**

radiated energy = **lamps, transmitters**

stored energy = **battery, capacitors, inductors**

**I WAS GOING TO TELL YOU A JOKE ABOUT
ELECTRICITY**



**BUT IT WAS TOO RE-VOLT-
ING**

And now on to the
microcontroller

Usually you'll be prototyping on a **breadboard**

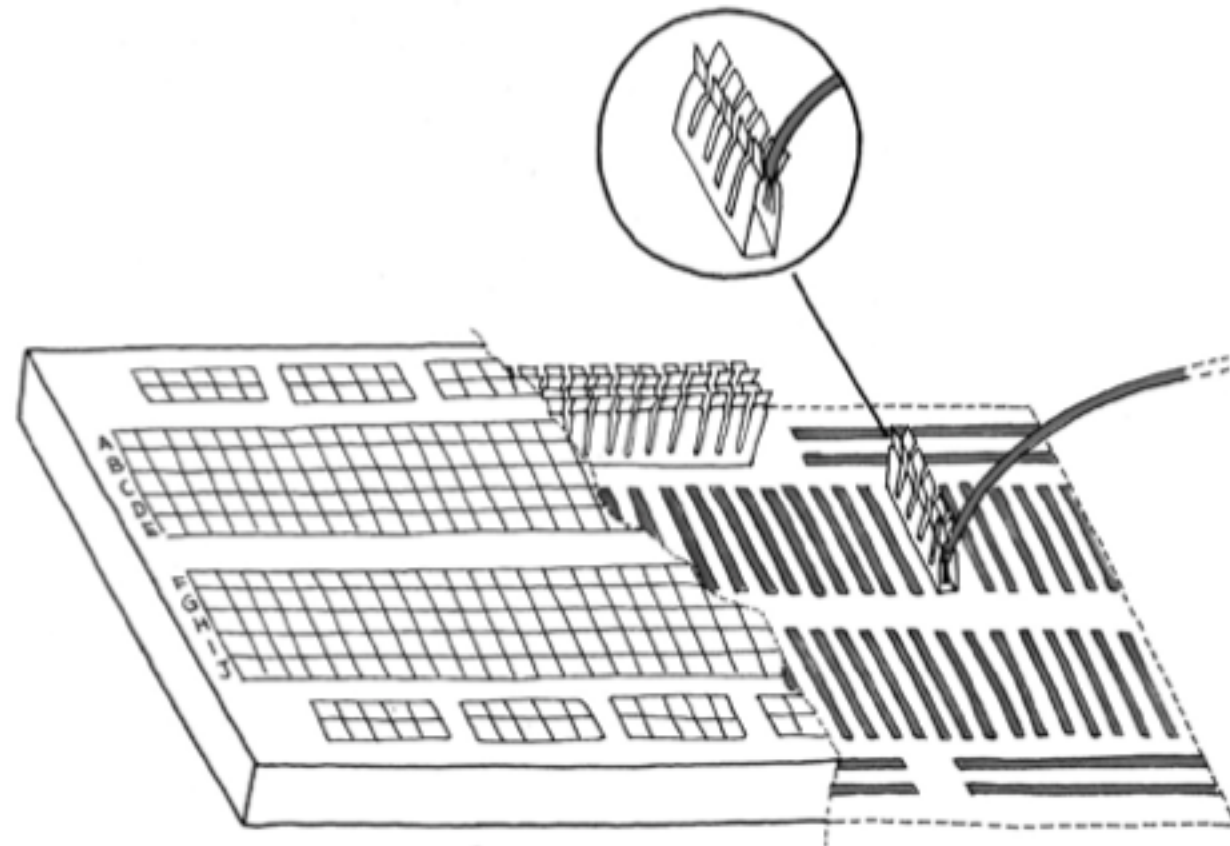
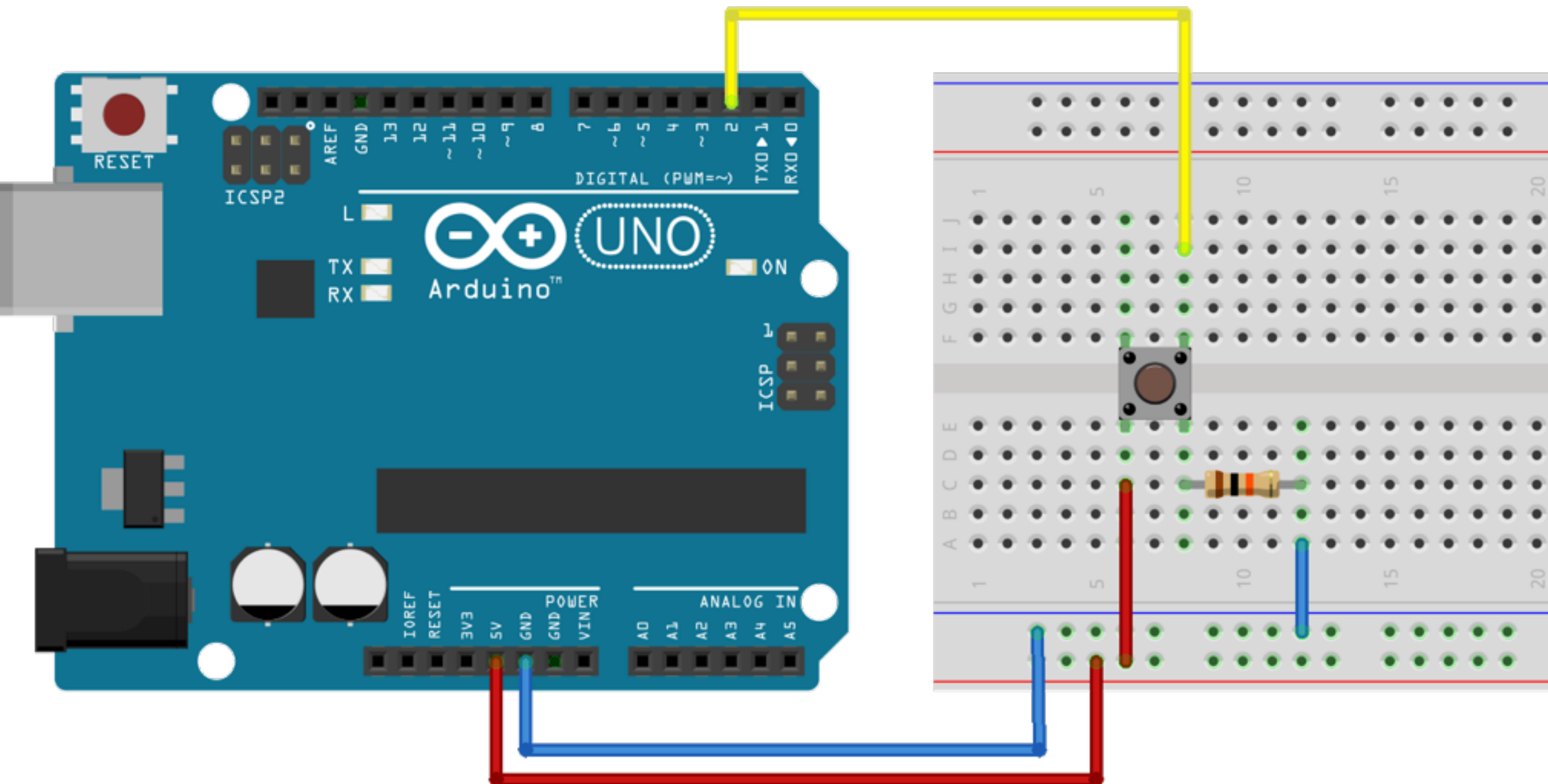
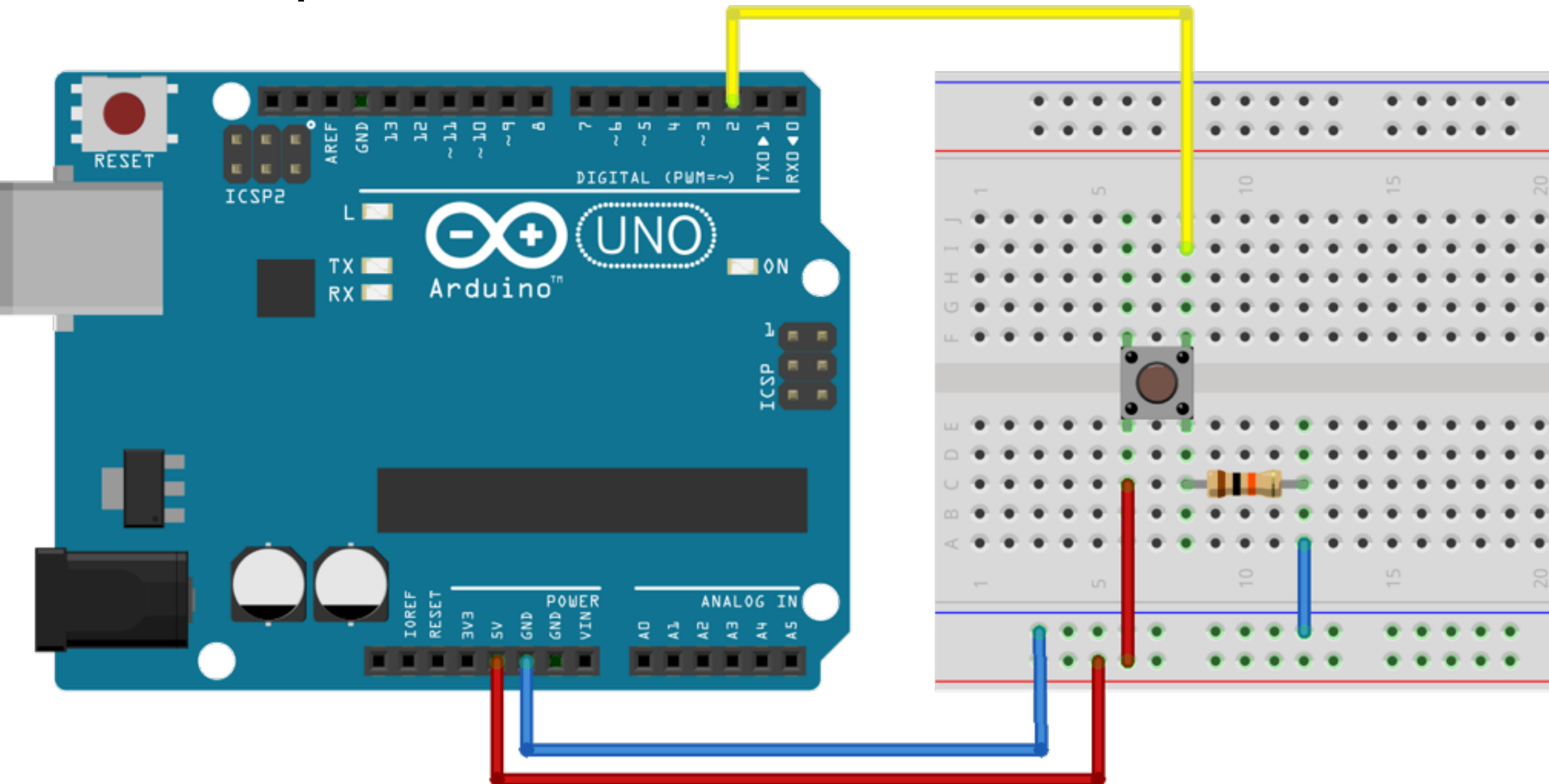


Figure A-1.
The solderless breadboard

You will be looking to connect your sensors to the **arduino pins**



If set to output, they send out **five volts**
and up to **40 mA of current**



THE TWO MOST COMMON USE OF PINS

Digital & Analog* (PWM)

Get the data in
Send the data out

DIGITAL
ON/OFF
0V or 5V

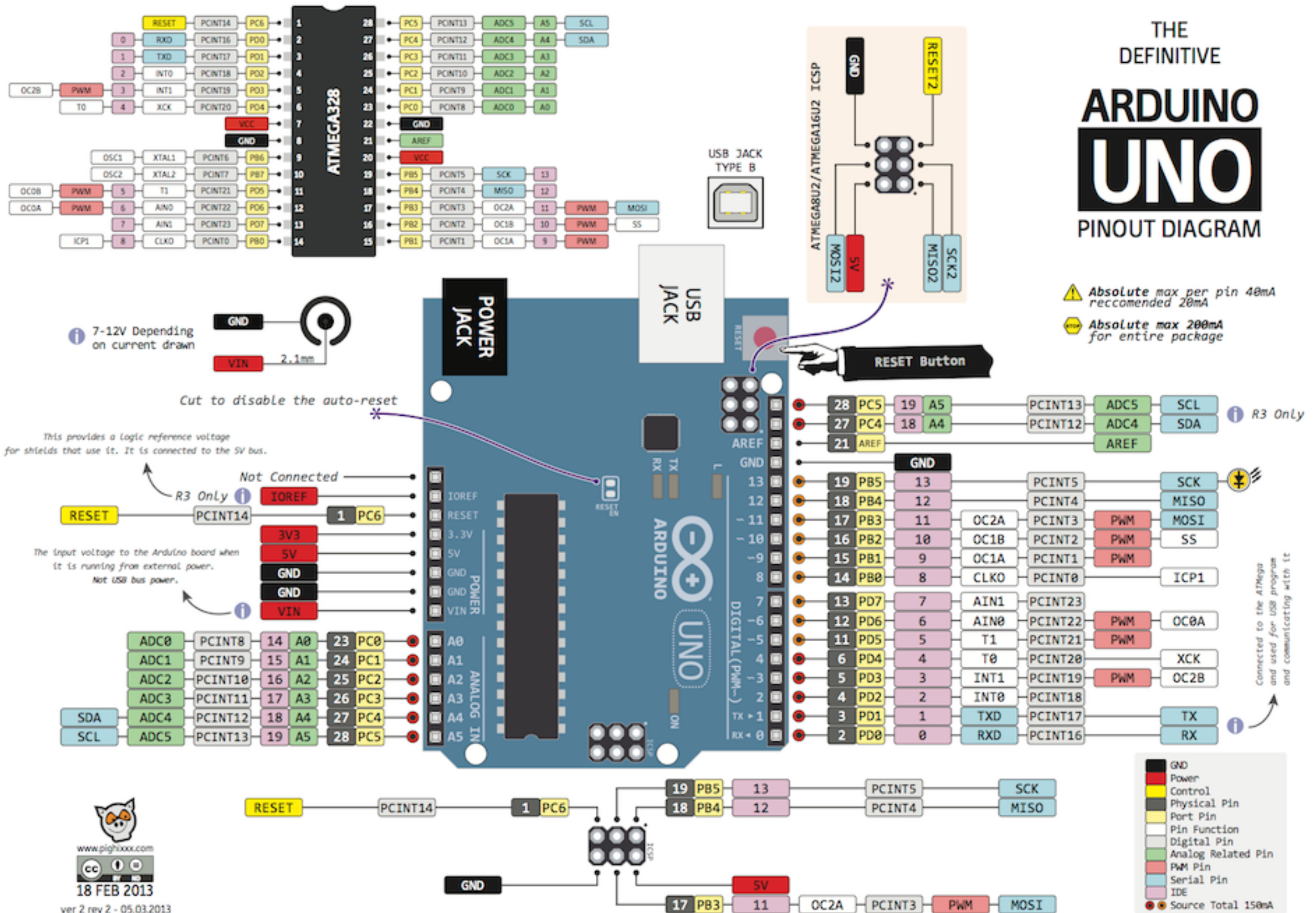
PWM / ANALOG

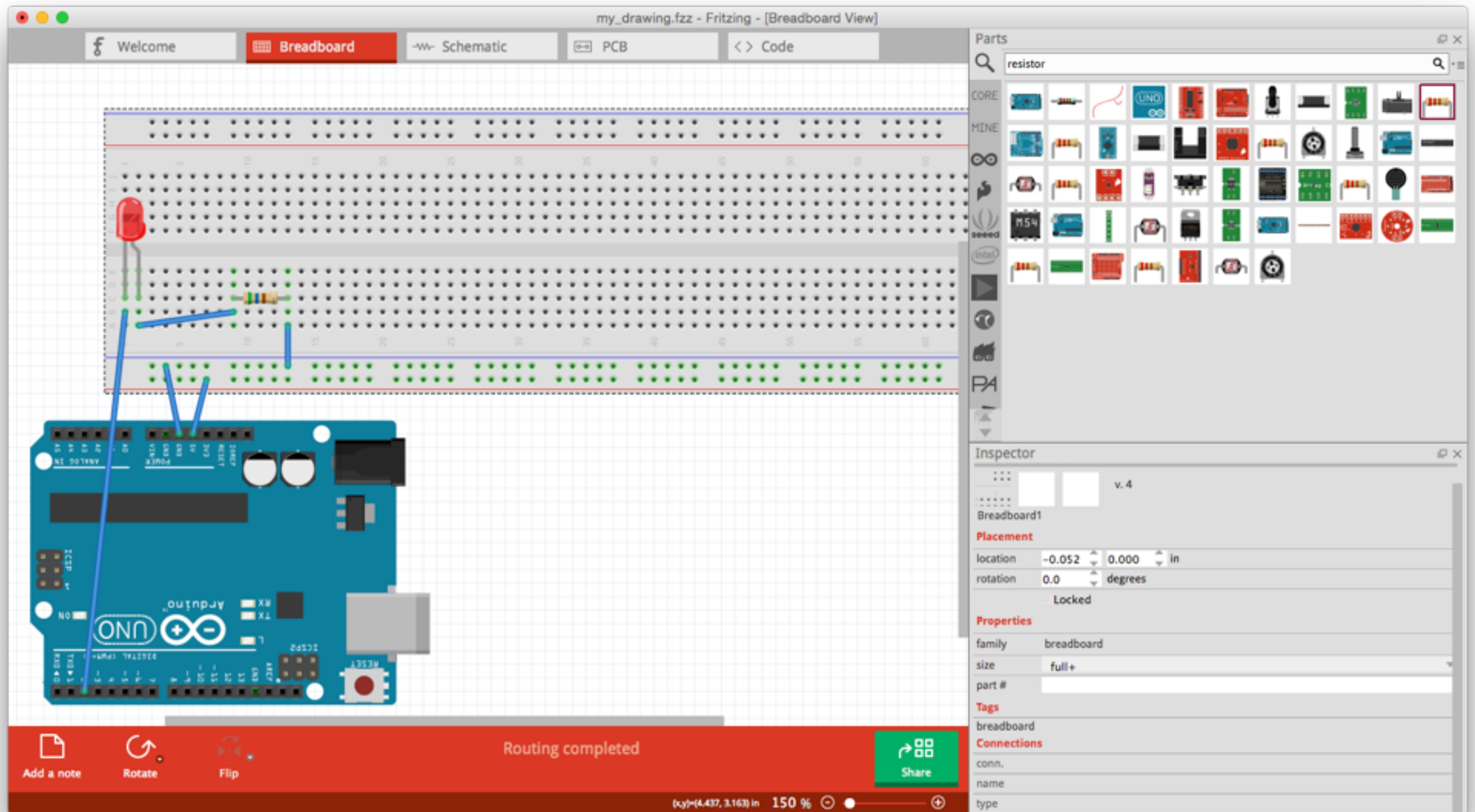
0-255

Voltage varies between 0 and 5V.

Same with current.

THE
DEFINITIVE
ARDUINO
UNO
PINOUT DIAGRAM





<http://fritzing.org/home/>

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 and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at <http://arduino.cc>

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board

```
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}
```

// the loop function runs over and over again forever

```
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);            // wait for a second  
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);            // wait for a second  
}
```

Arduino has two main loops just like processing.

1. setup(){}

Set your pins up here as input or output using the pinMode function.

signature: **pinMode(pinNumber, MODE)**

Mode is either **INPUT** or **OUTPUT** mainly...

2. loop(){}

Write or read to your pins and do any timing or computation.

Steps to get / send data for each pin

1. Create an int for the pin number outside the scope of your functions
2. Set the pinMode as input or output in setup()
3. Write to the pin or read from pin in loop()
4. If you read, you'll want to save that data to another variable in loop()

BUILDING A HEALTHY **HACKER SPACE**

<https://wiki.hackerspaces.org/Hackerspaces>





RESOURCES

<http://www.allaboutcircuits.com/>

MAKE MAGAZINE

INSTRUCTABLES

ADAFRUIT TUTORIALS