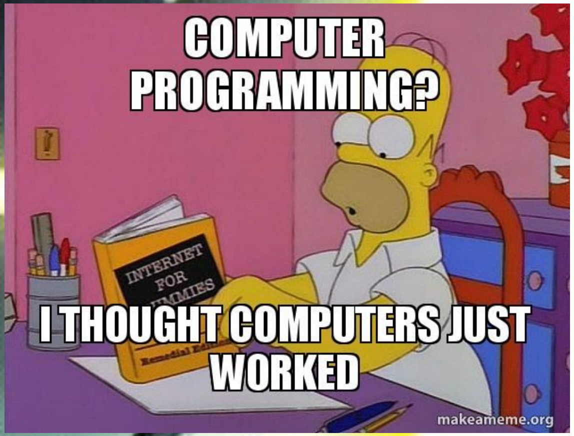
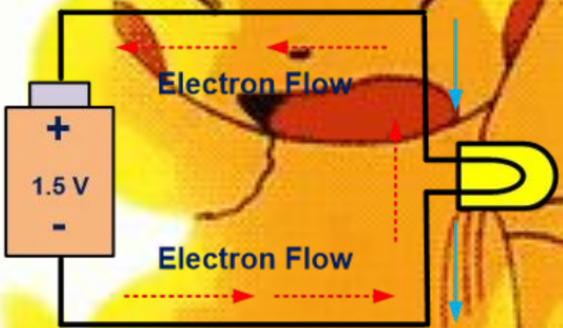
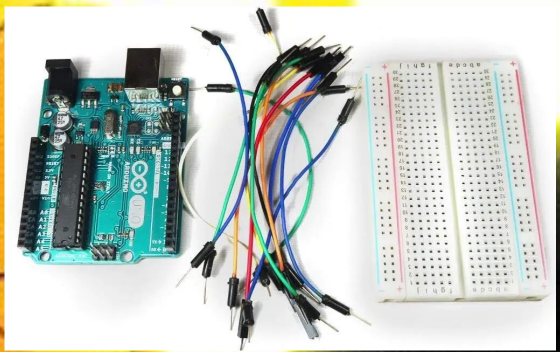
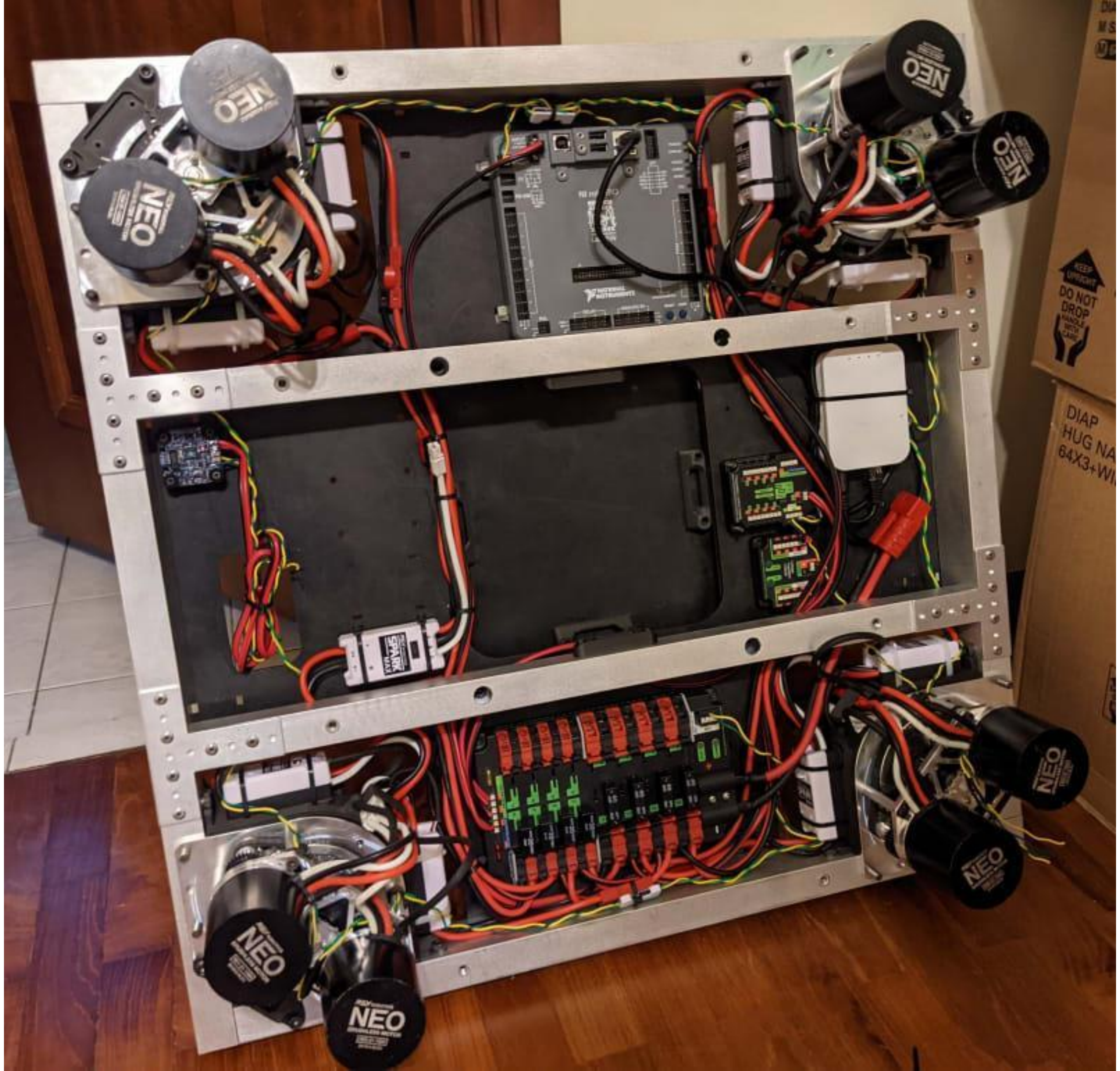
JAMCODERS

Circuits!





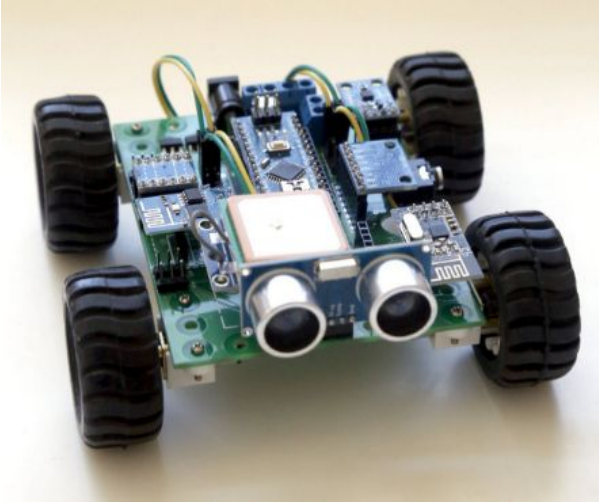




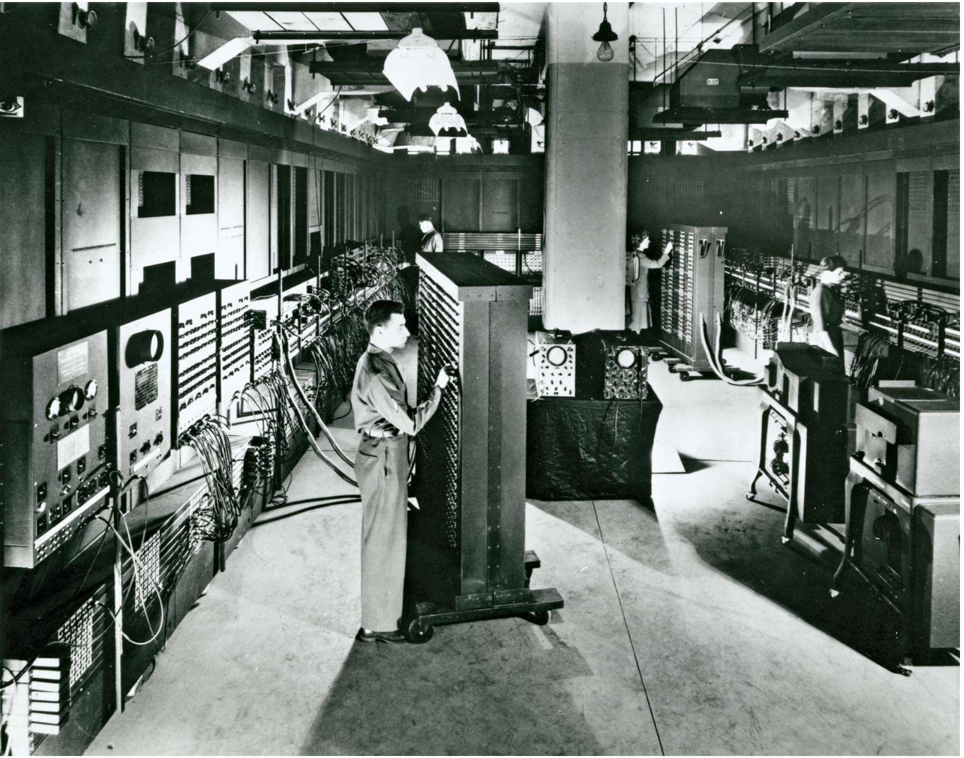
NEO
NEO
NEO
NEO
NATIONAL INSTRUMENTS
SPARK MAX
NEO
NEO
NEO
NEO
KEEP UPRIGHT
DO NOT DROP
HANDLE WITH CARE
DIAP
HUG NA
64X3+WI

In JamCoders, so far: Software

Today: Hardware!

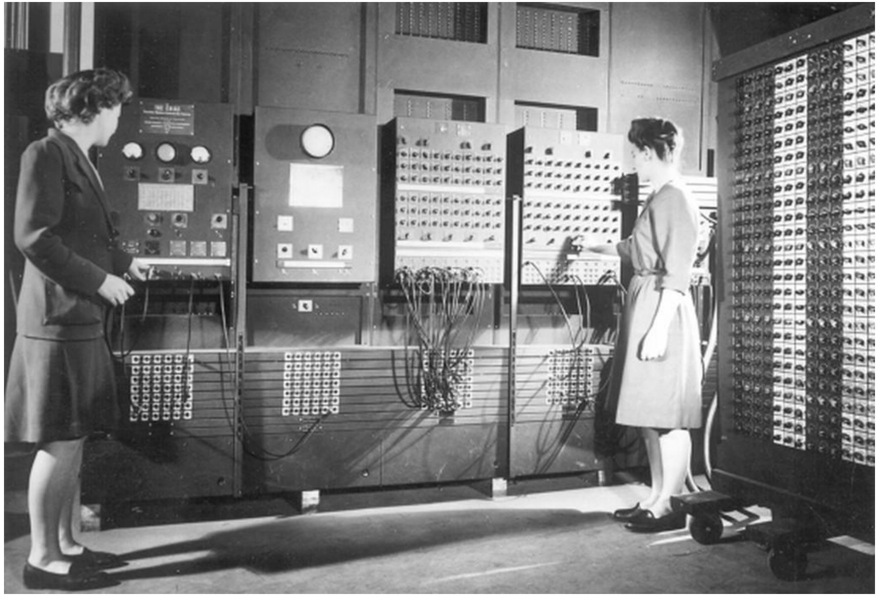


Electronic Numerical Integrator and Computer (ENIAC)



- Invented 1946
- University of Pennsylvania (U Penn)

ENIAC: 5000 calculations per second



M2Ultra: 31.6 trillion operations per second

Who is Grace Hopper?

Grace Hopper: Born 1906

Early computer scientist

Navy admiral

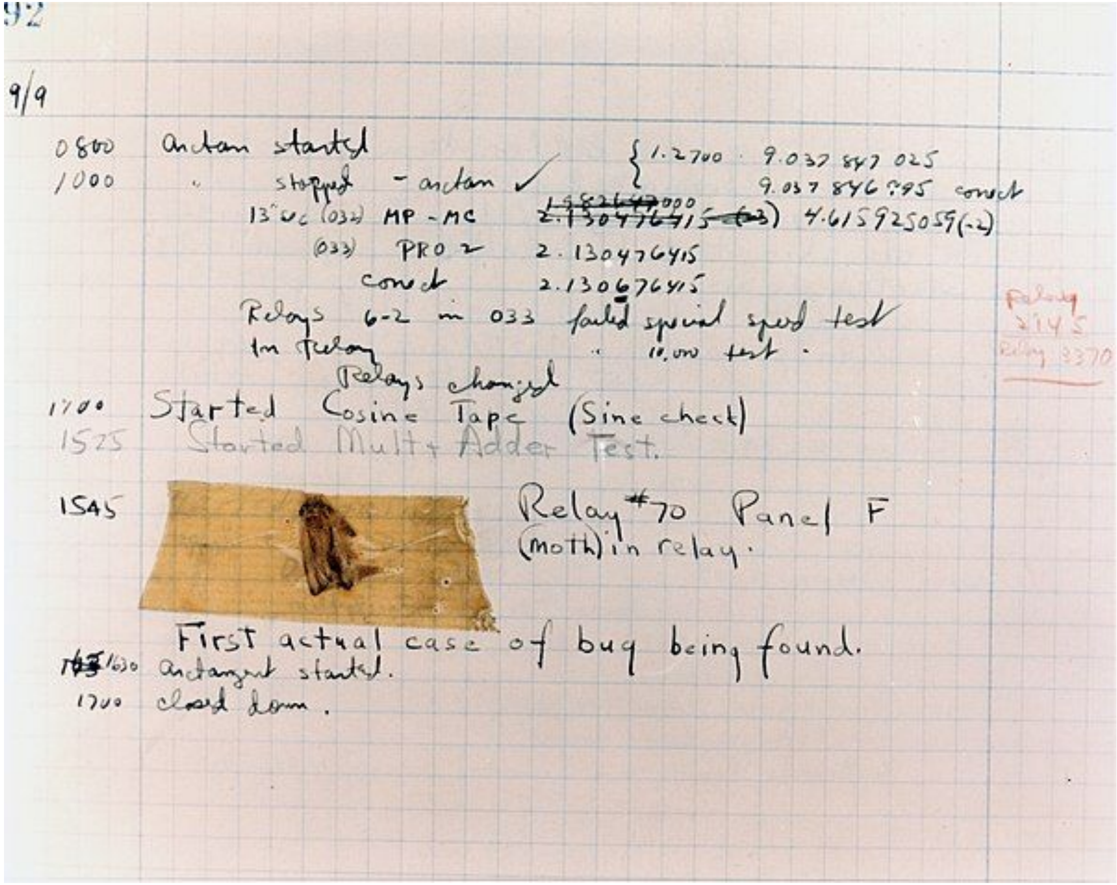
Known for:

- Inventing one of the first linkers
- Theory of machine-independent programming languages

What is debugging?

A bug died inside Grace Hopper's computer and impeded the relay operations

The term "debugging" was born



92

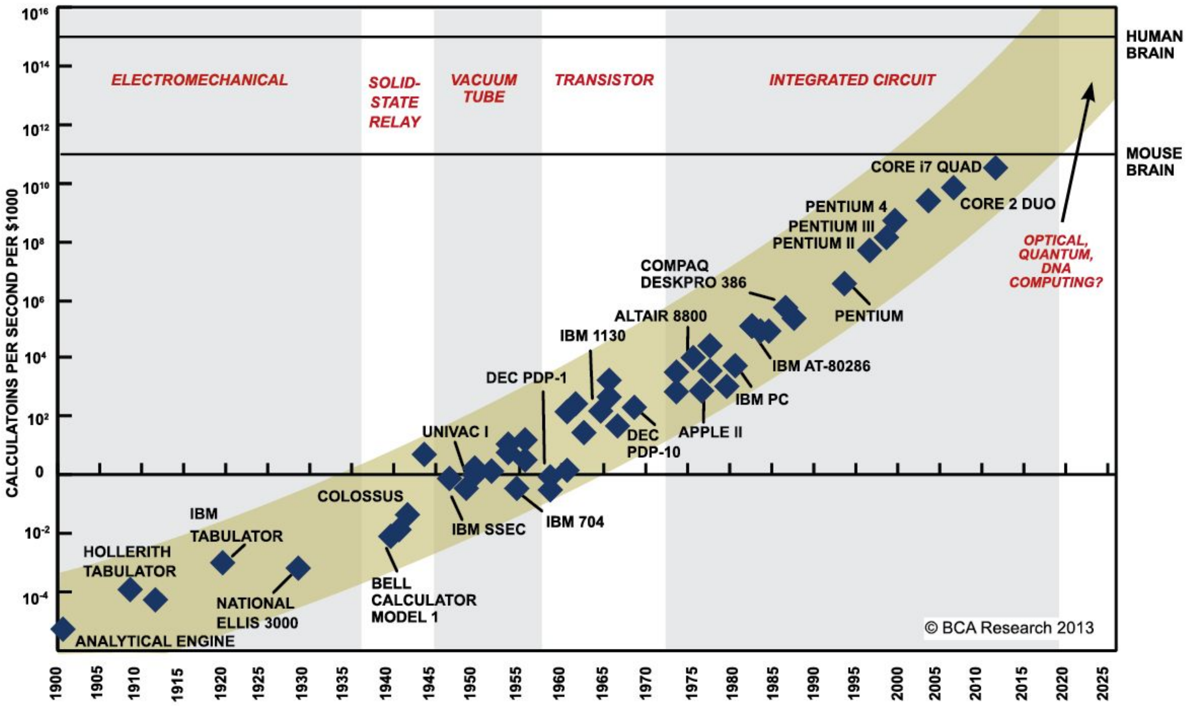
9/9

0800 arctan started
1000 " stopped - arctan ✓ {1.2700 9.037 847 025
9.037 846 995 correct
13" uc (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ ~~(-3)~~) 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
In relay " 10,000 test .
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult+ Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1545~~ 1630 antangent started.
1700 closed down.

Relay 2145
Relay 3370

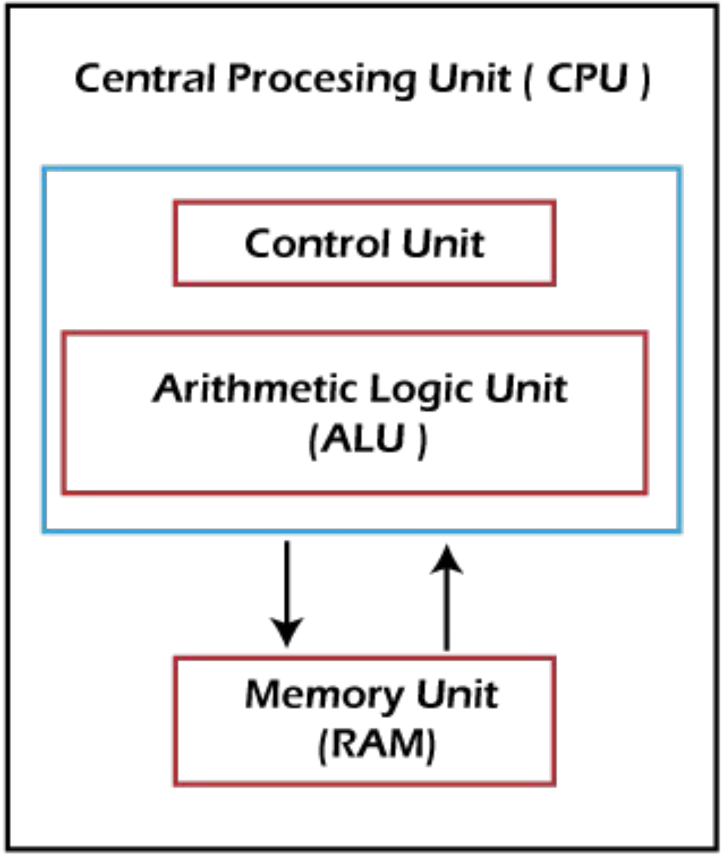
Debugging is still difficult, but luckily doesn't involve real bugs

Today, computers are faster, store more data, and take up less space.



© BCA Research 2013

SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

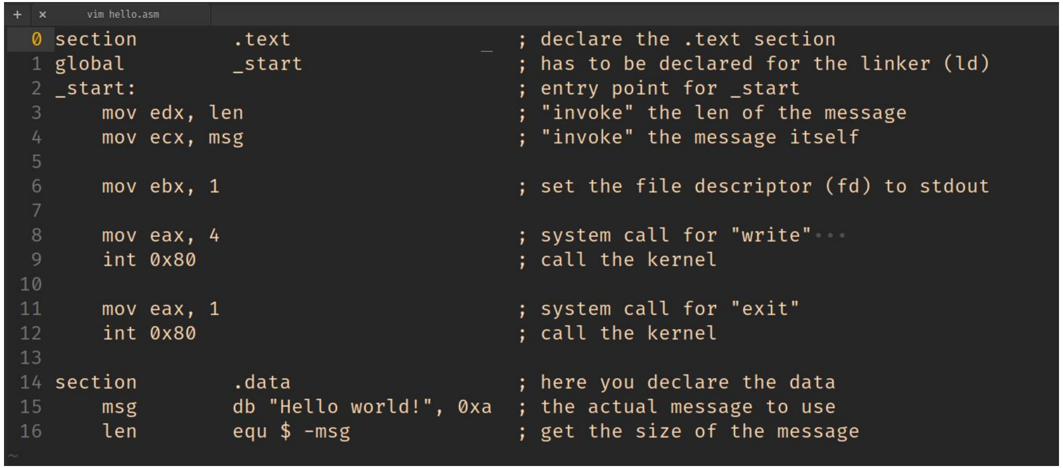


Central Procesing Unit (CPU)

Control Unit

Arithmetic Logic Unit (ALU)

Memory Unit (RAM)



```
section        .text                   ; declare the .text section
global         _start                  ; has to be declared for the linker (ld)
_start:                                ; entry point for _start
    mov edx, len                       ; "invoke" the len of the message
    mov ecx, msg                       ; "invoke" the message itself

    mov ebx, 1                         ; set the file descriptor (fd) to stdout

    mov eax, 4                         ; system call for "write"
    int 0x80                           ; call the kernel

    mov eax, 1                         ; system call for "exit"
    int 0x80                           ; call the kernel

section        .data                   ; here you declare the data
    msg        db "Hello world!", 0xa  ; the actual message to use
    len        equ $ -msg              ; get the size of the message
```

How to make computers faster?

- Get data faster.
    - Access to data is limited by distance (speed of light).
    - Multiple tiers of cache on a chip.

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds

How to make computers faster?

- Get data faster.
    - Access to data is limited by distance (speed of light).
    - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)

How to make computers faster?

- Get data faster.
    - Access to data is limited by distance (speed of light).
    - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	

How to make computers faster?

- Get data faster.
    - Access to data is limited by distance (speed of light).
    - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)
SSD: 100,000ns (100us)	

How to make computers faster?

- Get data faster.
    - Access to data is limited by distance (speed of light).
    - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)
SSD: 100,000ns (100us)	~7 hours (go to the library today)

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)
SSD: 100,000ns (100us)	~7 hours (go to the library today)
Disk: 10,000,000ns (10,000us, 10ms)	

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)
SSD: 100,000ns (100us)	~7 hours (go to the library today)
Disk: 10,000,000ns (10,000us, 10ms)	~29 days (Jamcoders!)

How to make computers faster?

- Get data faster.
  - Access to data is limited by distance (speed of light).
  - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)
SSD: 100,000ns (100us)	~7 hours (go to the library today)
Disk: 10,000,000ns (10,000us, 10ms)	~29 days (Jamcoders!)
Network (JAM ↔ NYC): 70,000,000ns (70,000us, 70ms)	

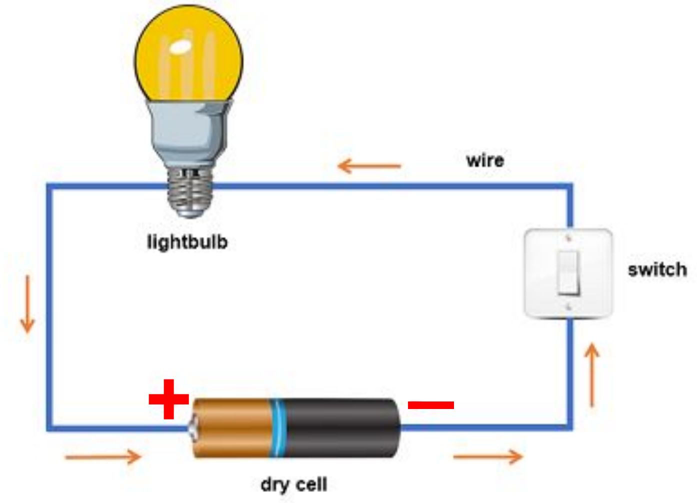
How to make computers faster?

- Get data faster.
    - Access to data is limited by distance (speed of light).
    - Multiple tiers of cache on a chip.

Computer	**Human Analogy**
L1 Cache: 1ns	0.25 seconds (fact just learned)
L3 Cache: 16ns	4 seconds (fact from a month ago)
RAM: 100ns	25 seconds (look in a book on your desk)
SSD: 100,000ns (100us)	~7 hours (go to the library today)
Disk: 10,000,000ns (10,000us, 10ms)	~29 days (Jamcoders!)
Network (JAM ↔ NYC): 70,000,000ns (70,000us, 70ms)	~202 days (a year of school)

What is voltage?

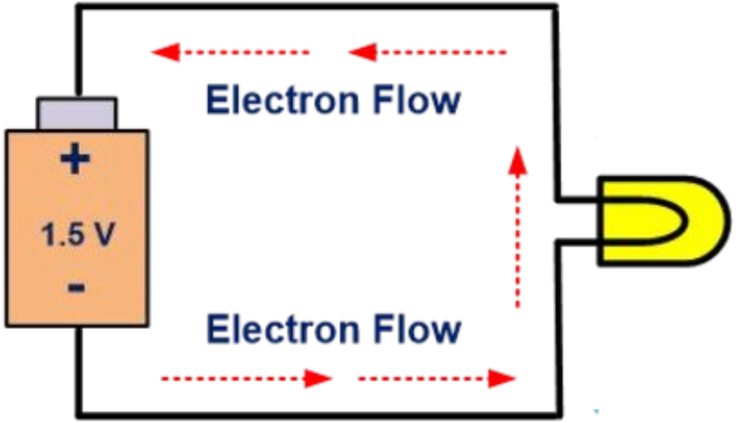
Electric potential difference per unit charge between two points in a circuit's electric field.



- The battery supplies voltage, causing an energy imbalance to exist between the + and -
- This causes current (electrons or ions) to flow

What is current?

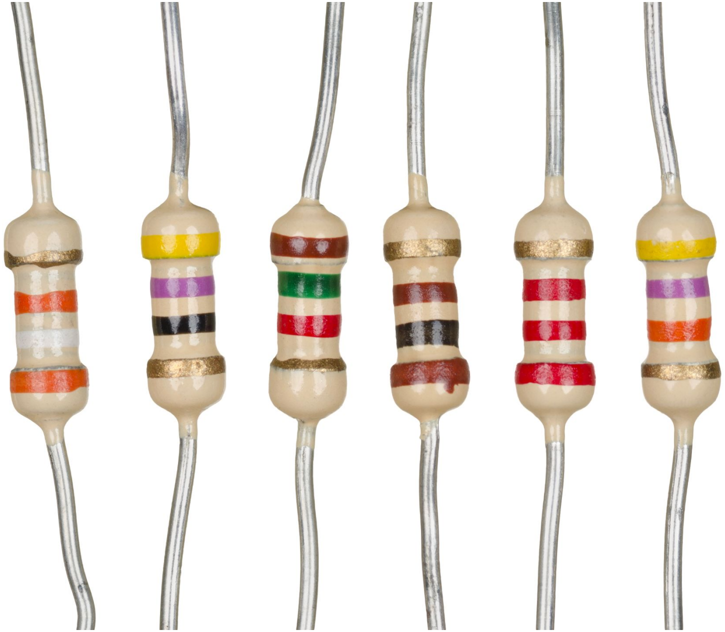
Electrical current is a stream of charged particles, such as electrons or ions, moving through an electrical conductor.



Resistors

Resistors provide resistance to electrical current.

They are often needed to regulate the current and voltage in a circuit.

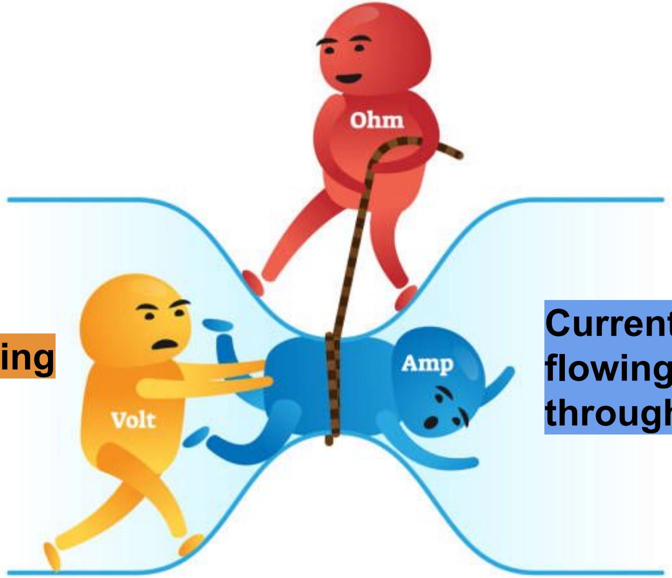


Ohm’s Law

Ohm’s Law captures the relationship between:

- Current (I)
- Voltage (V)
- Resistance (R)

$$V = IR$$



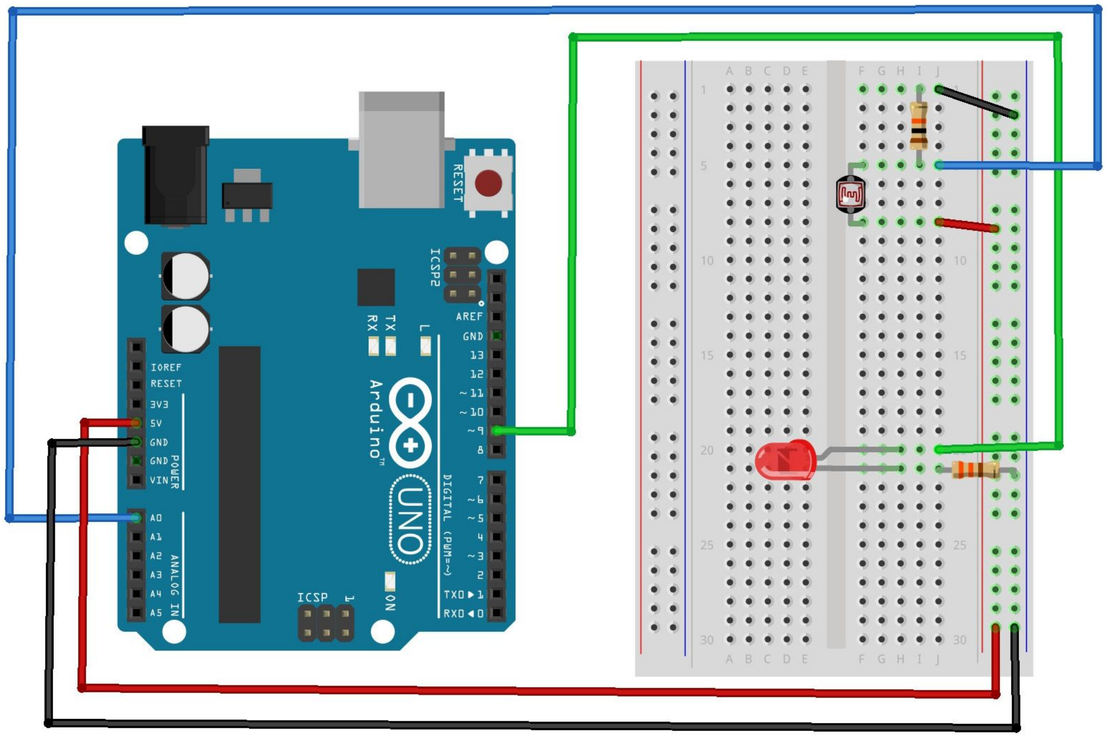
Resistance opposes electron flow

Voltage is causing current to flow

Current is flowing through wire

Arduino Chip

Breadboard



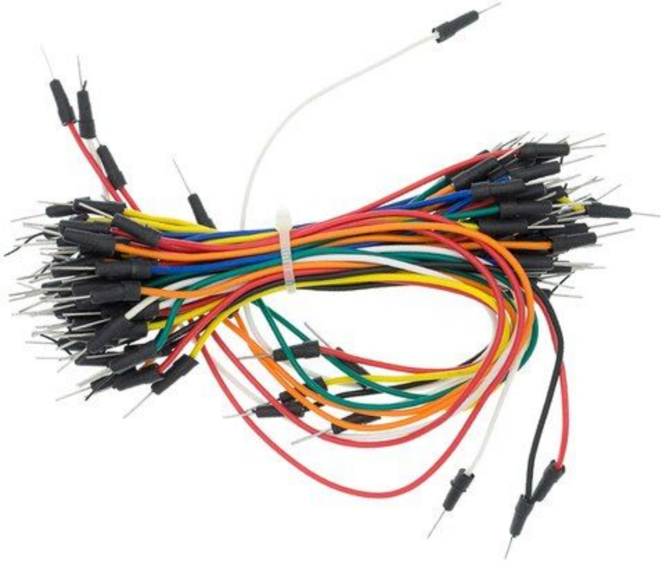
Circuit Components

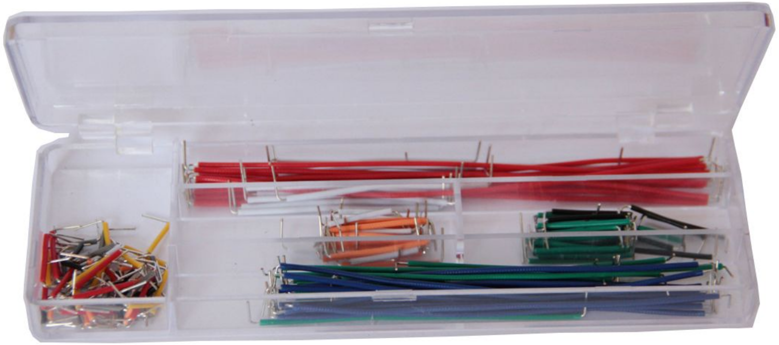
- Wires
- Resistors
- Capacitors
- Transistors
- Multiplexers
- Switches
- OR, AND, and NOT gates
- The list goes on…

Wires

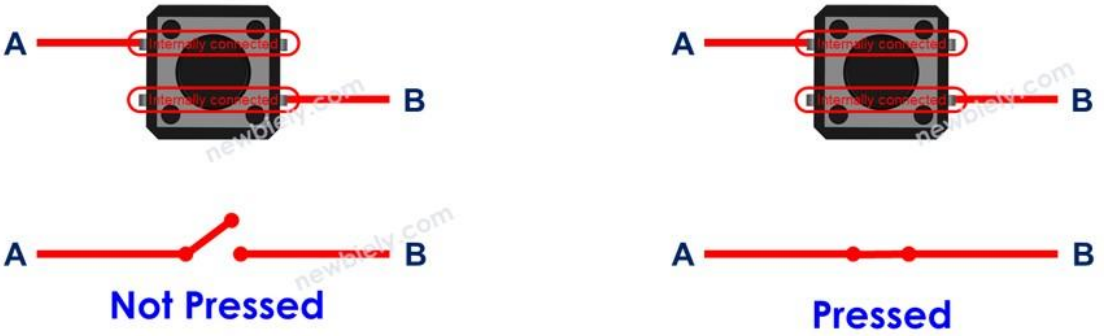
I know you know what wires are, here are some photos of the ones we're using:

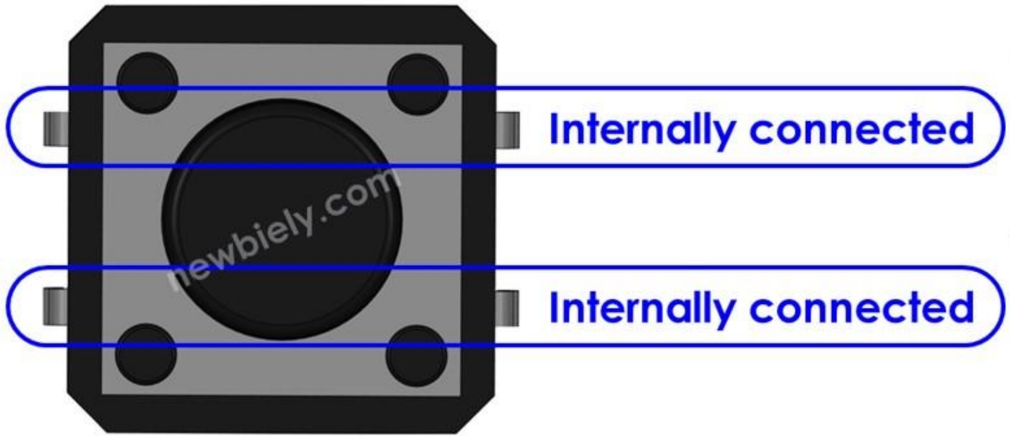
The metal ends enter the breadboard and Arduino pin holes

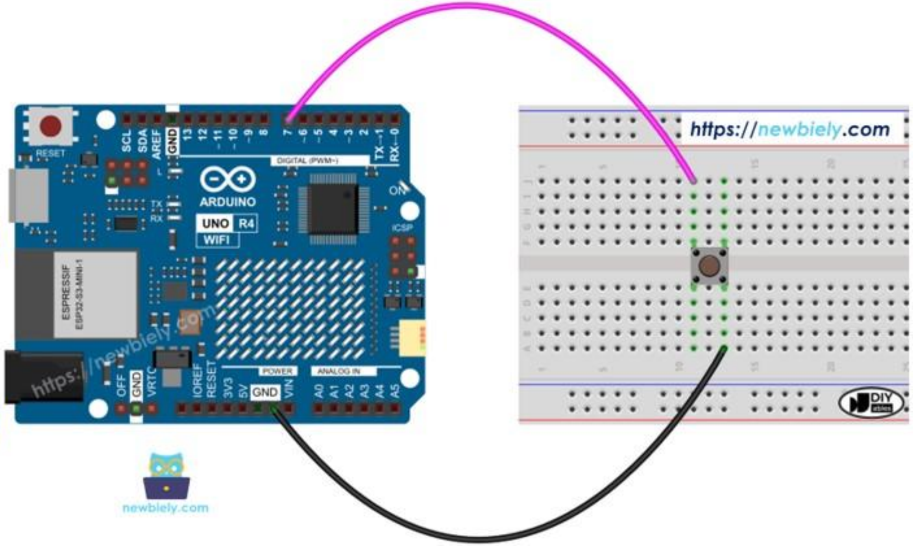


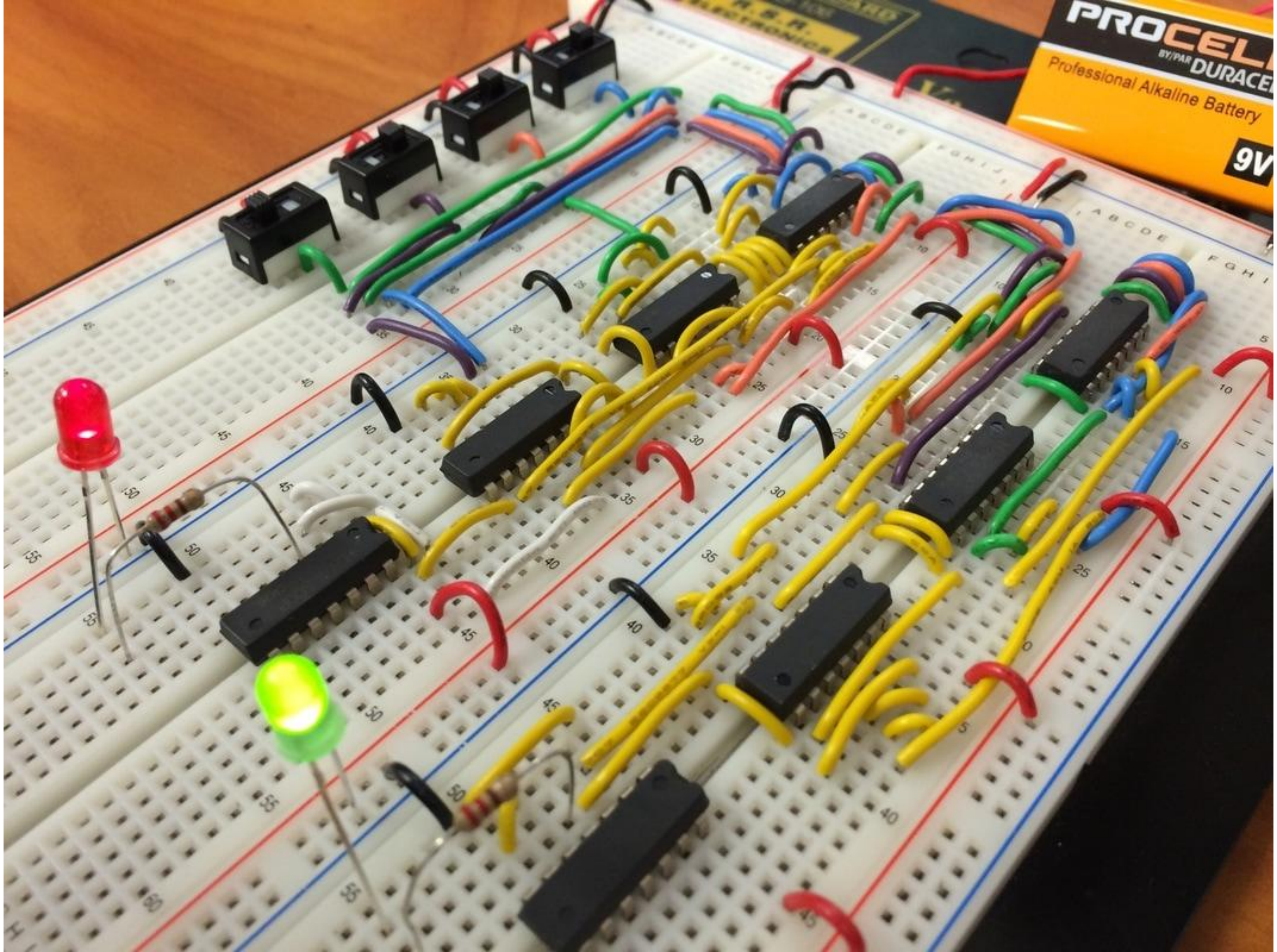


LED Lightbulbs

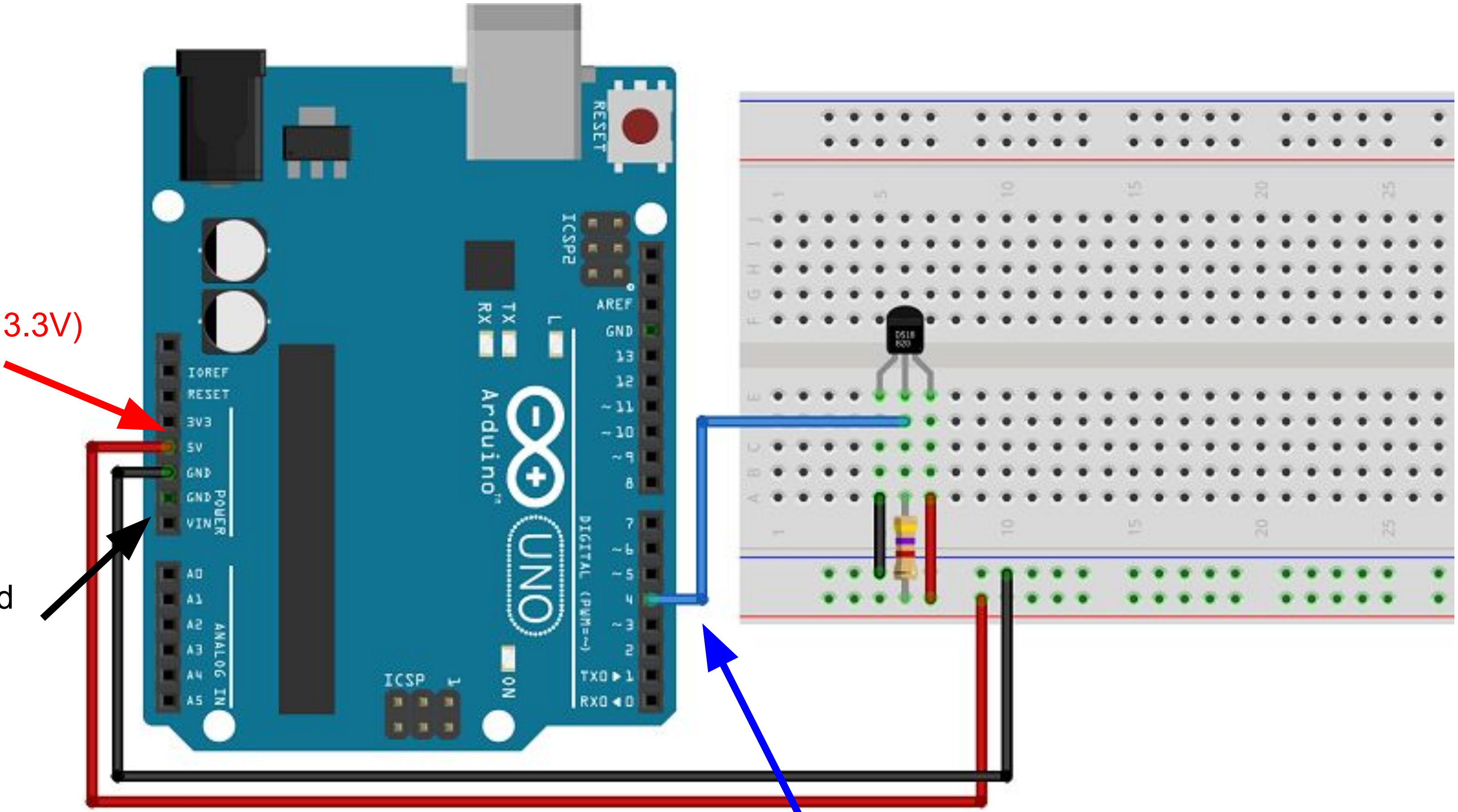

A
Internally connected
Internally connected
B
A
B
Not Pressed
A
Internally connected
Internally connected
B
A
B
Pressed


Internally connected
Internally connected
newbiely.com


RESET
SCL
SDA
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO R4
WIFI
ON
ICSP
ESPRESSIF
ESP32-S3-MINI-1
OFF
GND
VRTC
IOREF
RESET
3V3
5V
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
https://newbiely.com
newbiely.com


PROCELL
BY/PAR DURACELL
Professional Alkaline Battery
9V

Power
(5V or 3.3V)

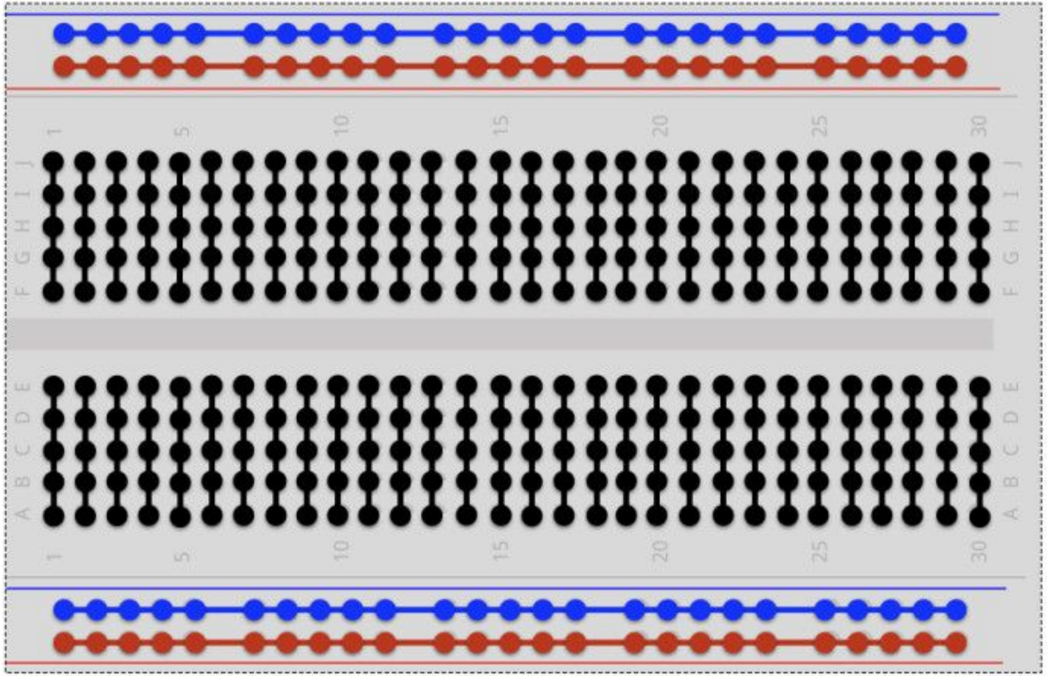


Ground
(GND)

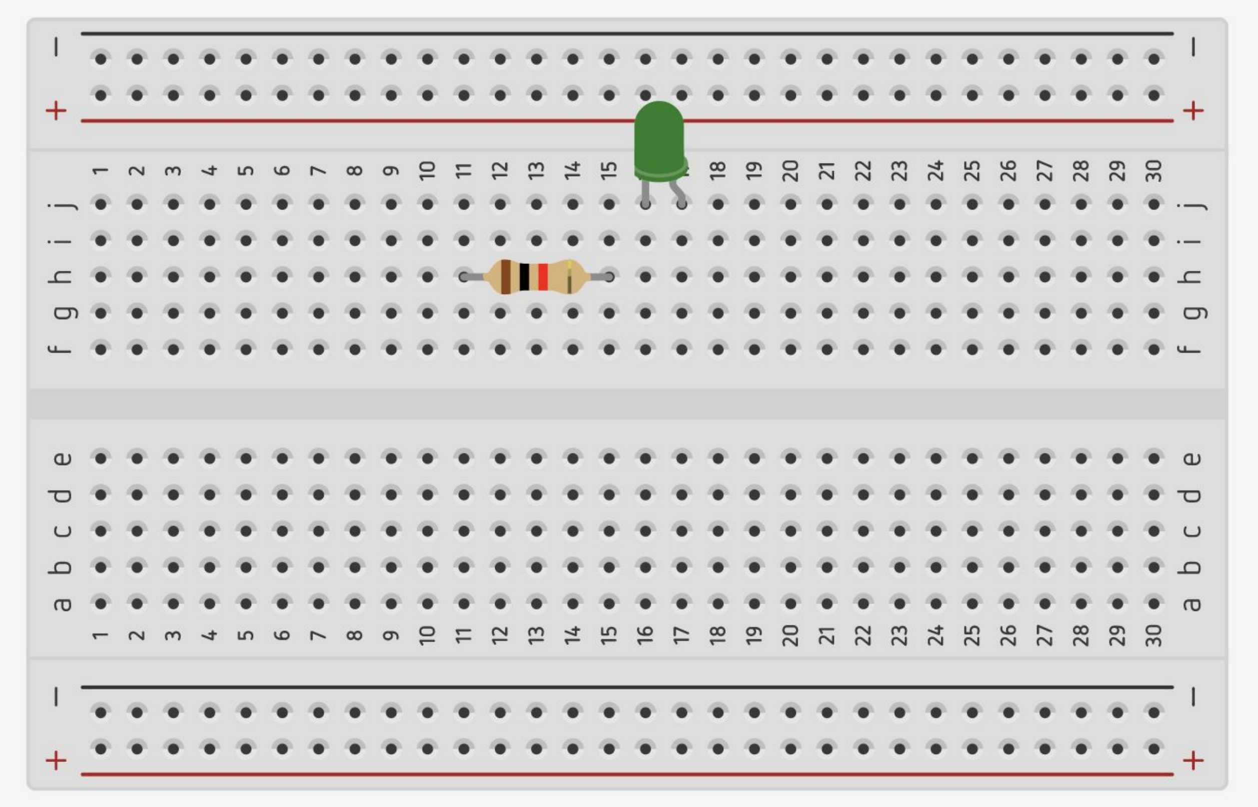
Pins, here pin 4 is in use

Breadboard

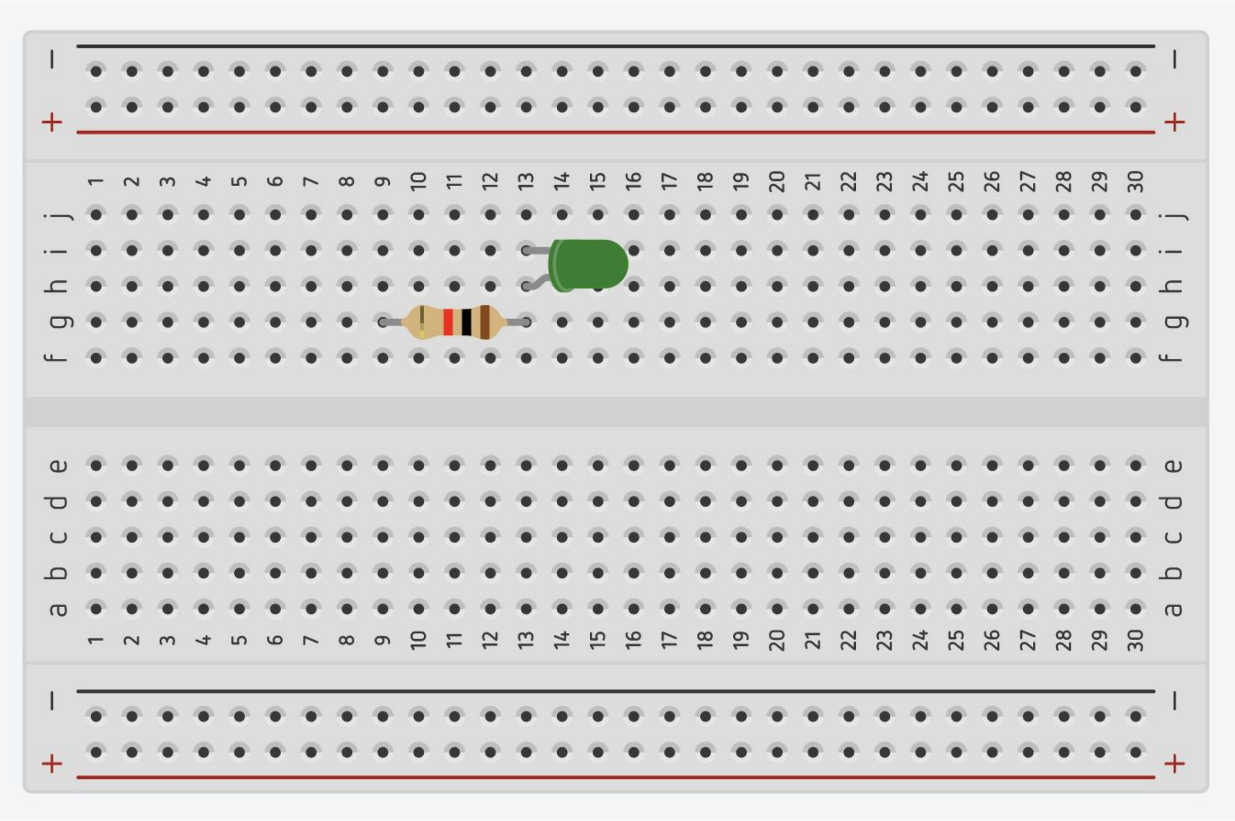
Breadboards are used to organize and connect circuit elements.



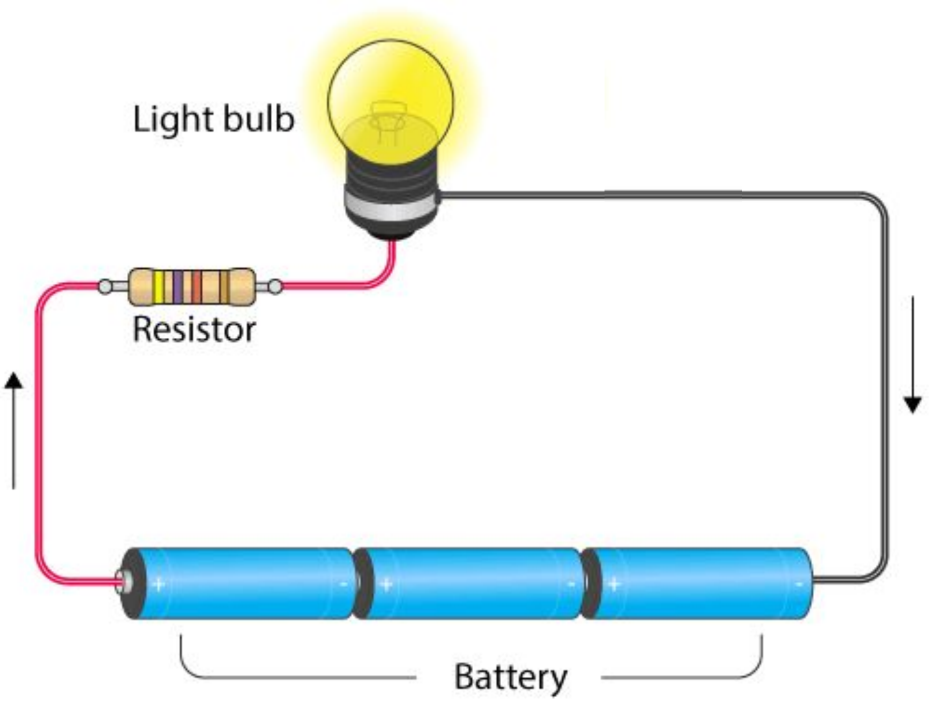
Are the resistor and the lightbulb connected?



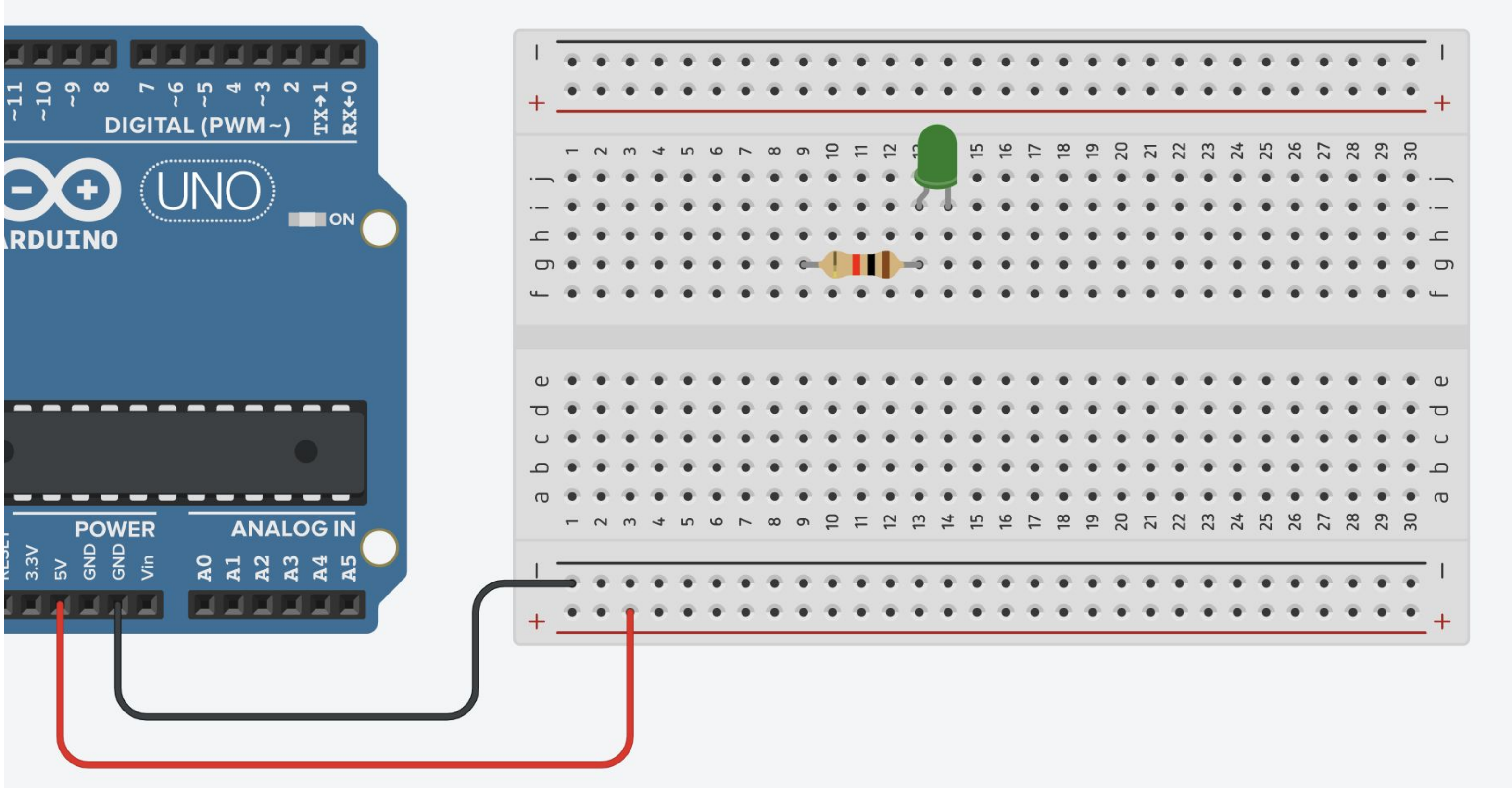
What about now? Is there anything wrong with this connection?



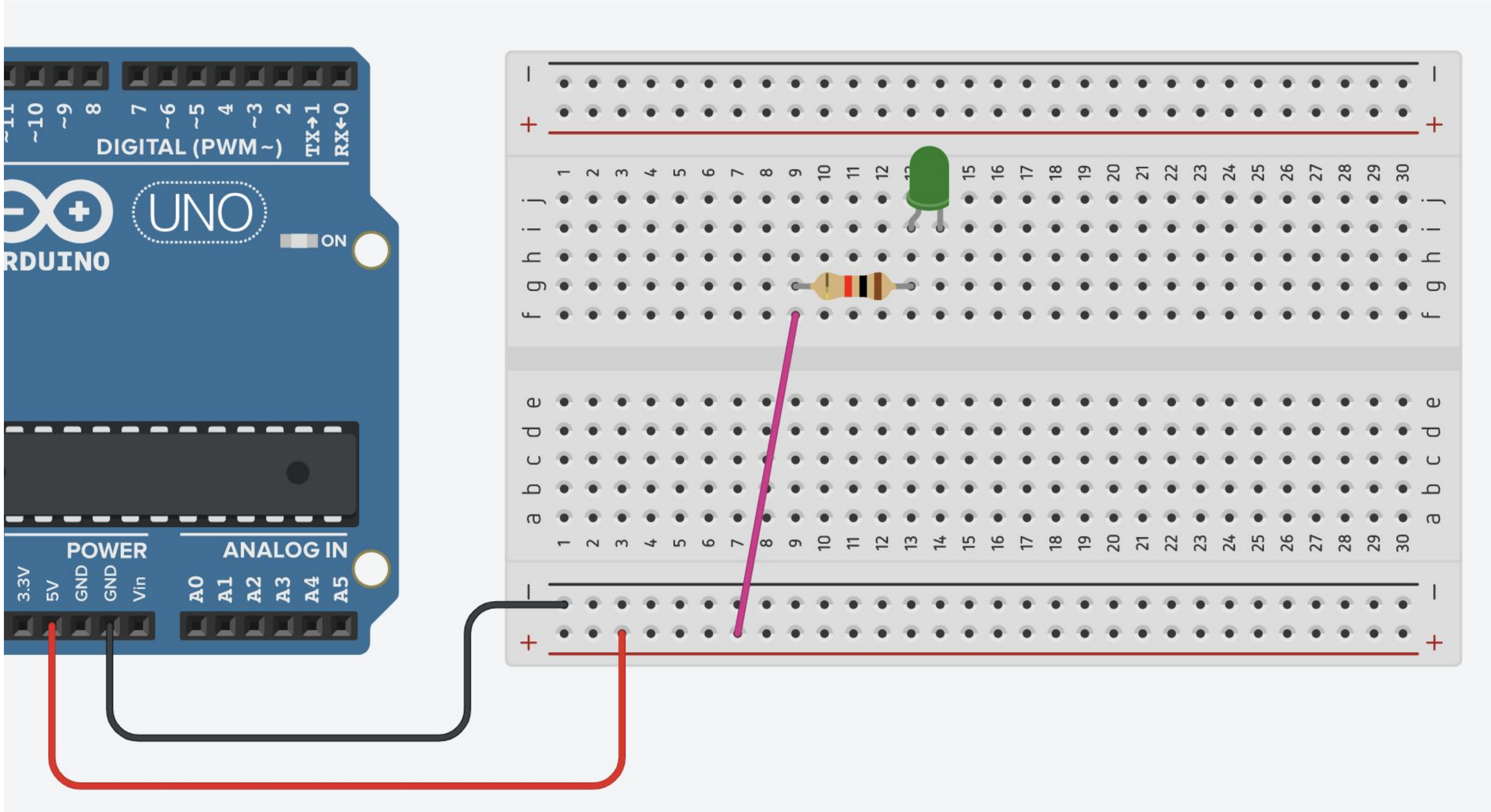
Simple Light Switch Circuit



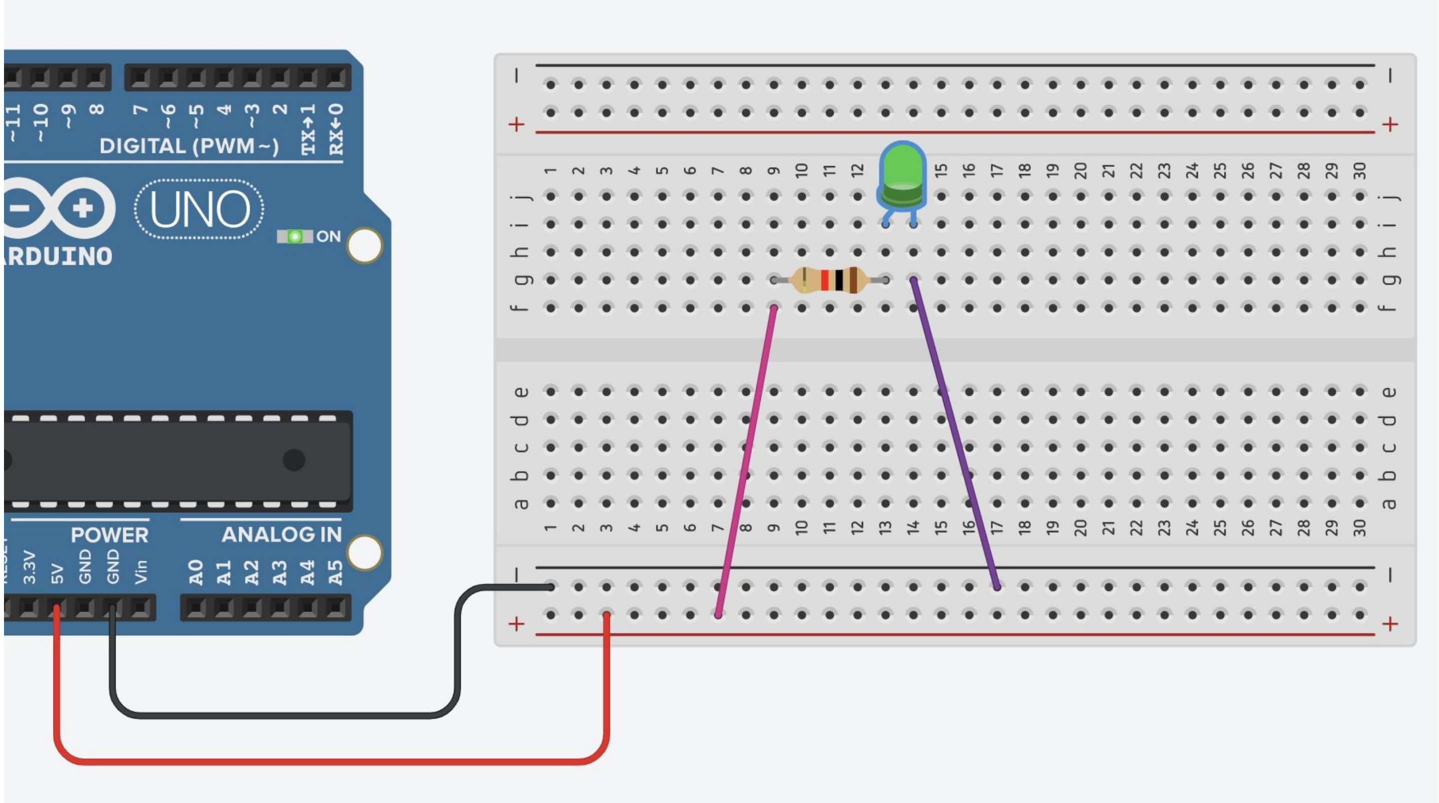
Let’s connect our circuit board to +/- voltage:



Now let’s connect the positive voltage to the resistor:



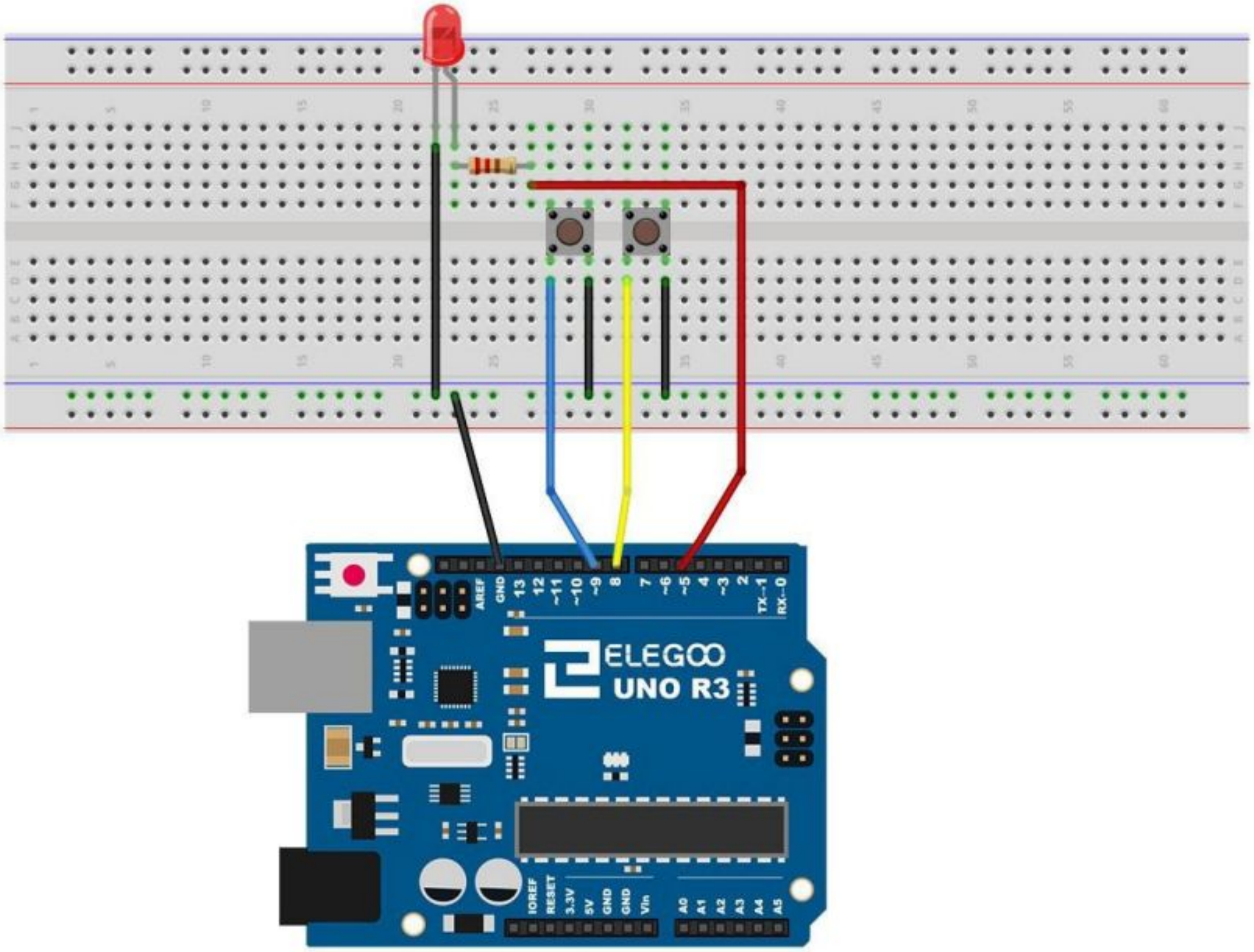
Now let’s connect the negative voltage to the lightbulb:



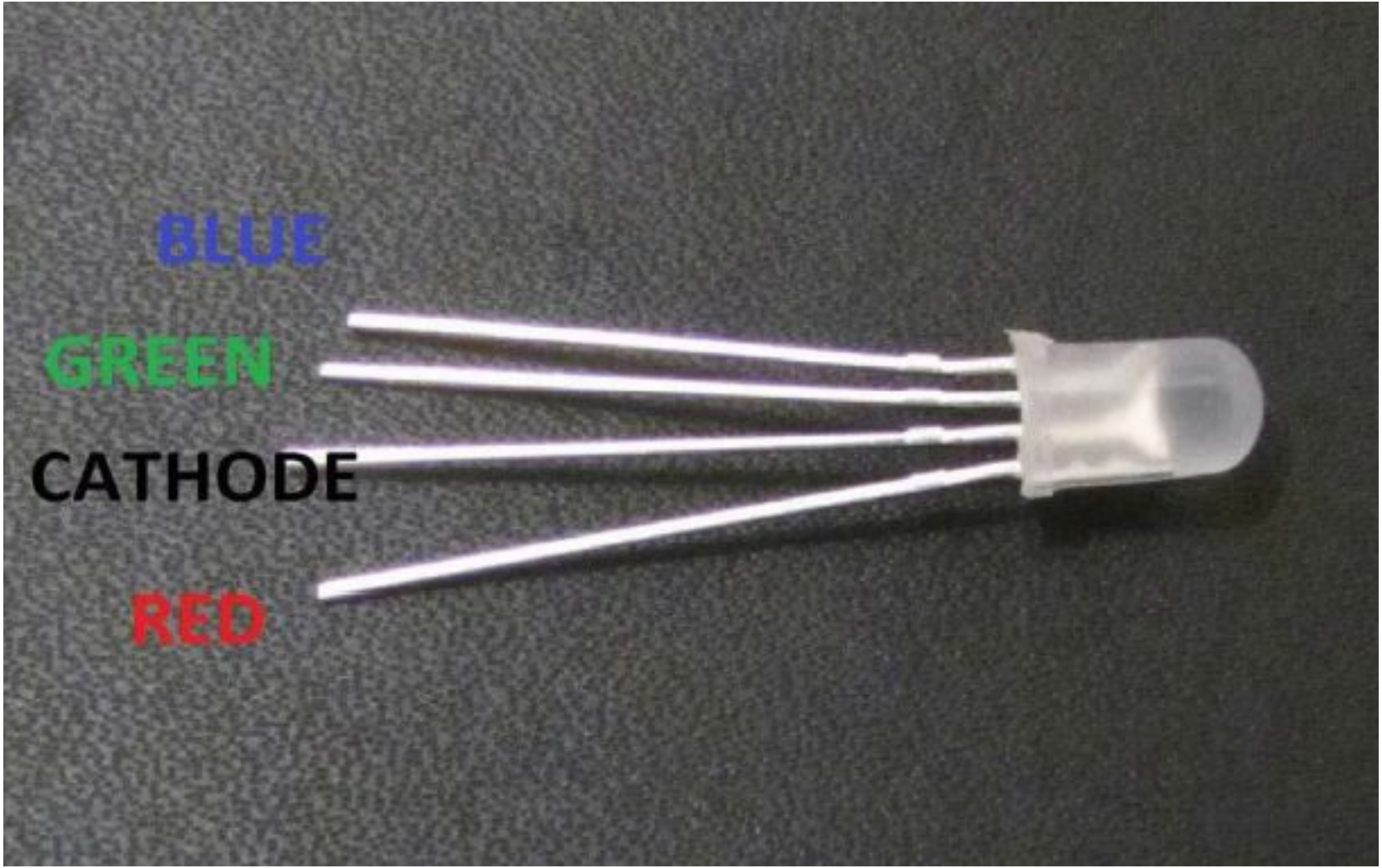
Nice :)

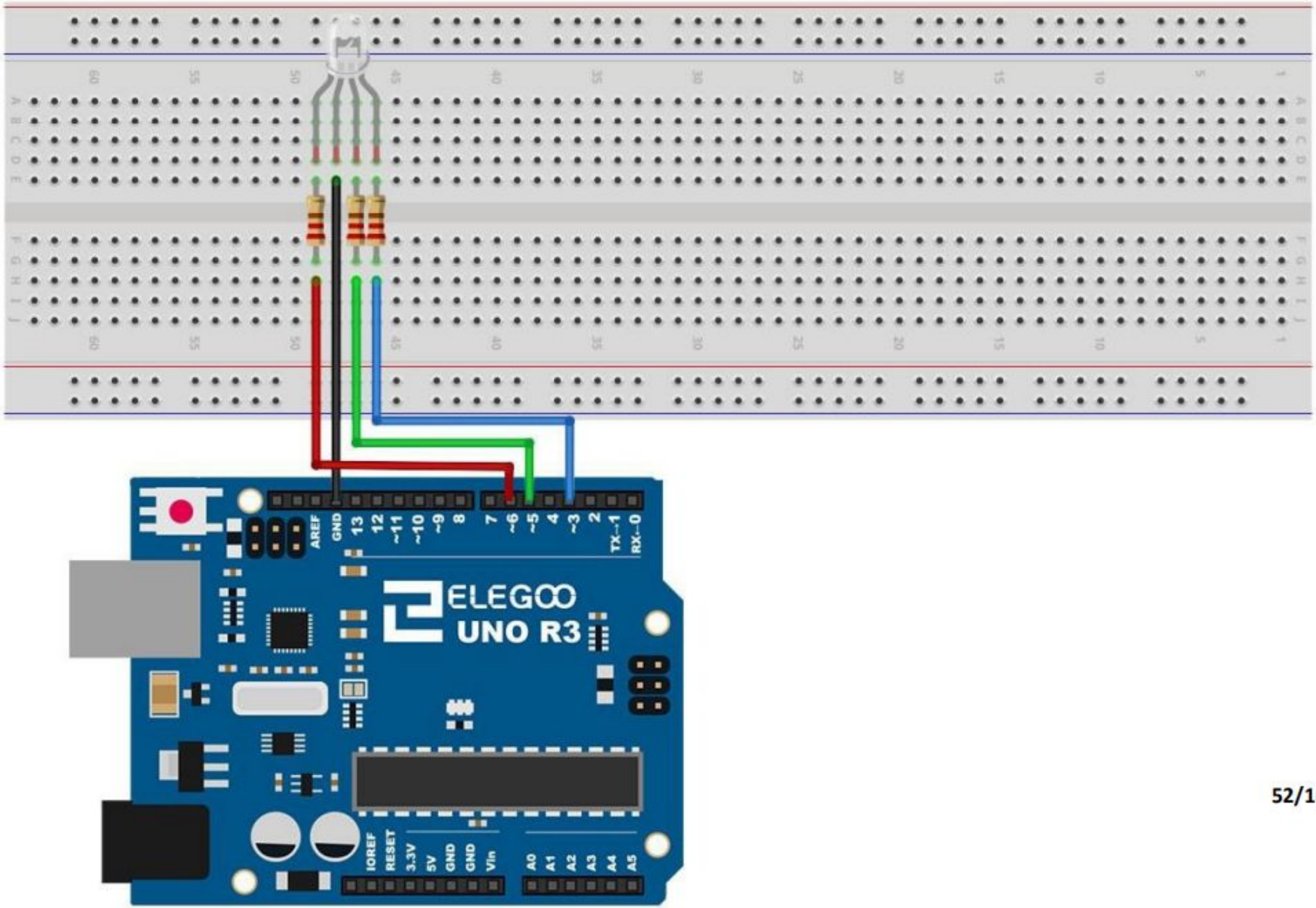
- Open Blink_LED.ino

Arduino chips run code which controls the voltage and current applied via wires into our circuit


1
5
10
15
20
25
30
35
40
45
50
55
60
J
I
H
G
F
E
D
C
B
A
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
ELEGOO
UNO R3
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

- Open LED_buttons.ino


BLUE
GREEN
CATHODE
RED



ELEGOO
UNO R3
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

- Open rgb_fade.ino

Now put it together!

Make a circuit that uses TWO buttons to control the colour of an RGB LED.

- Open rbg_buttons.ino