

Getting Started with Arduino

J.D. Zamfirescu
Michael Toren

What is Arduino?

- **Hardware** (Small programmable microcontroller)
- **IDE** (Software that runs on your computer)
- **Runtime Library** (Software that runs on the hardware)
- Learning platform
- Community of people

Super simplified overview: What can microcontroller do?

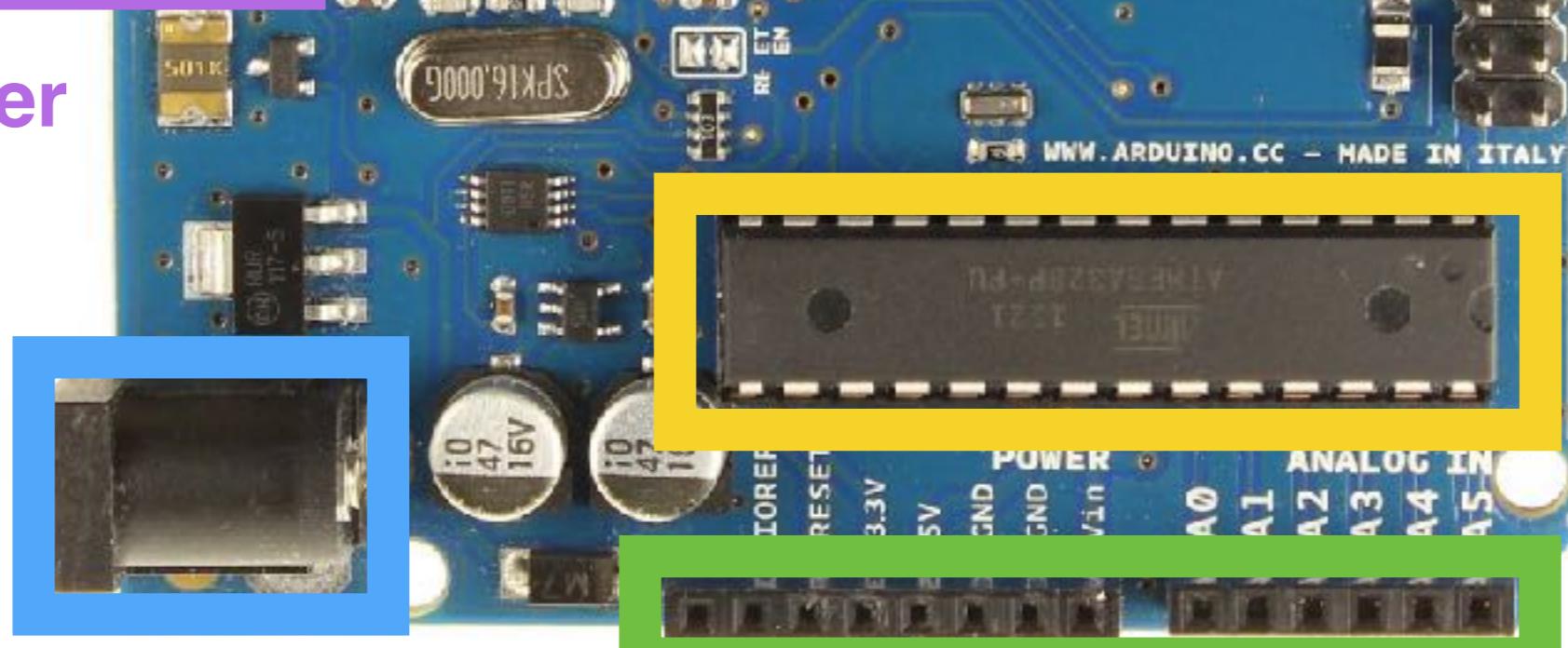
- Drive an output pin HIGH or LOW (sometimes very fast)
- Read if an input pin is HIGH or LOW
- Read how much voltage is on an input pin
- Talk to other chips

Reset Button



Digital Inputs & Outputs

USB / Power



ATmega 328

DC Power

Power & Analog Inputs

Arduino Hardware: UNO

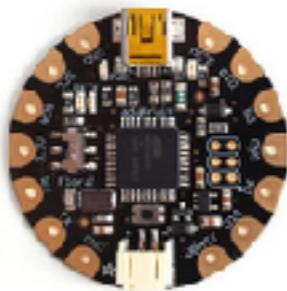
The Arduino Ecosystem



UNO



MEGA



FLORA

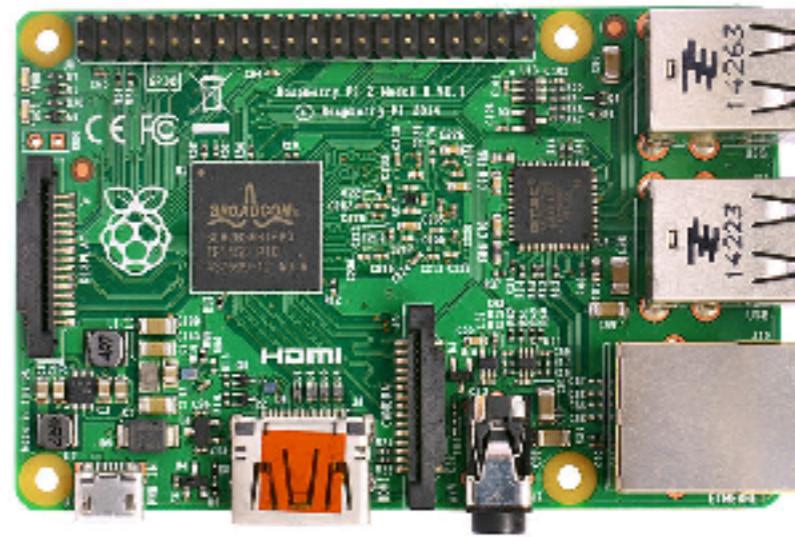


Pro Mini

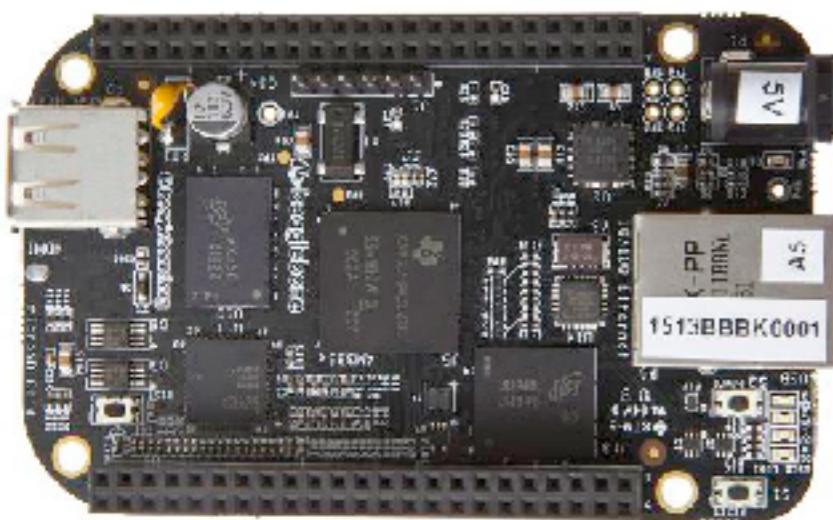
Arduino's Many Competitors



UNO



Raspberry Pi



BeagleBone Black



Particle Photon

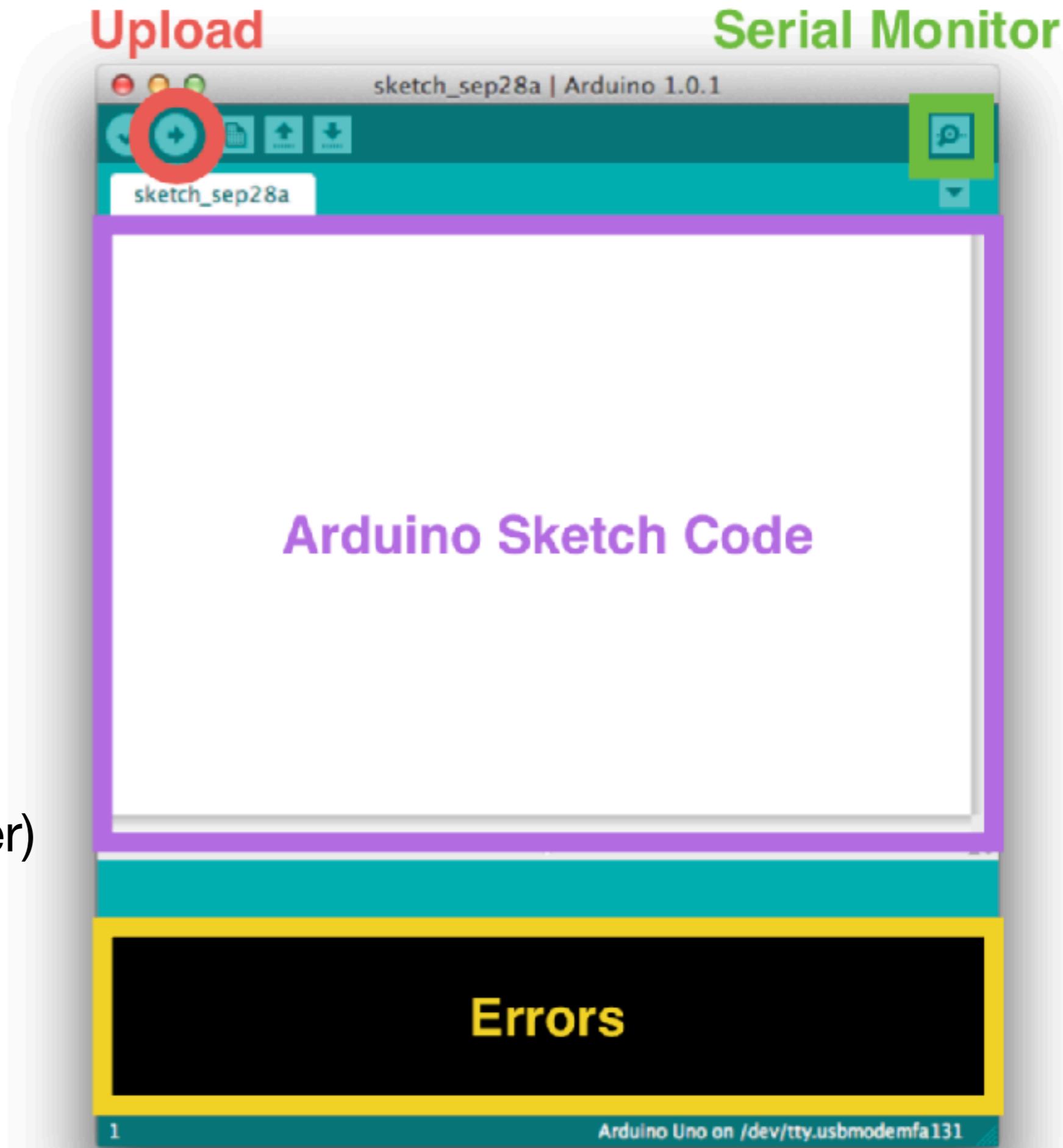


AdaFruit HUZZAH

Arduino IDE

Integrated
Development
Environment

(Runs on your computer)



Anatomy of an Arduino 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.
// Pin 11 has the LED on Teensy 2.0
// Pin 6 has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

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

Anatomy of an Arduino Sketch

```
int led = 13;

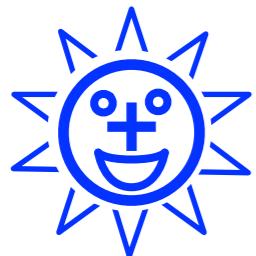
void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

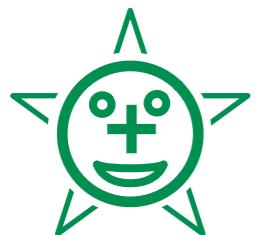
File > Examples > Basics > Blink

Electronics is...

Moving Charge



Highly energetic charge particle

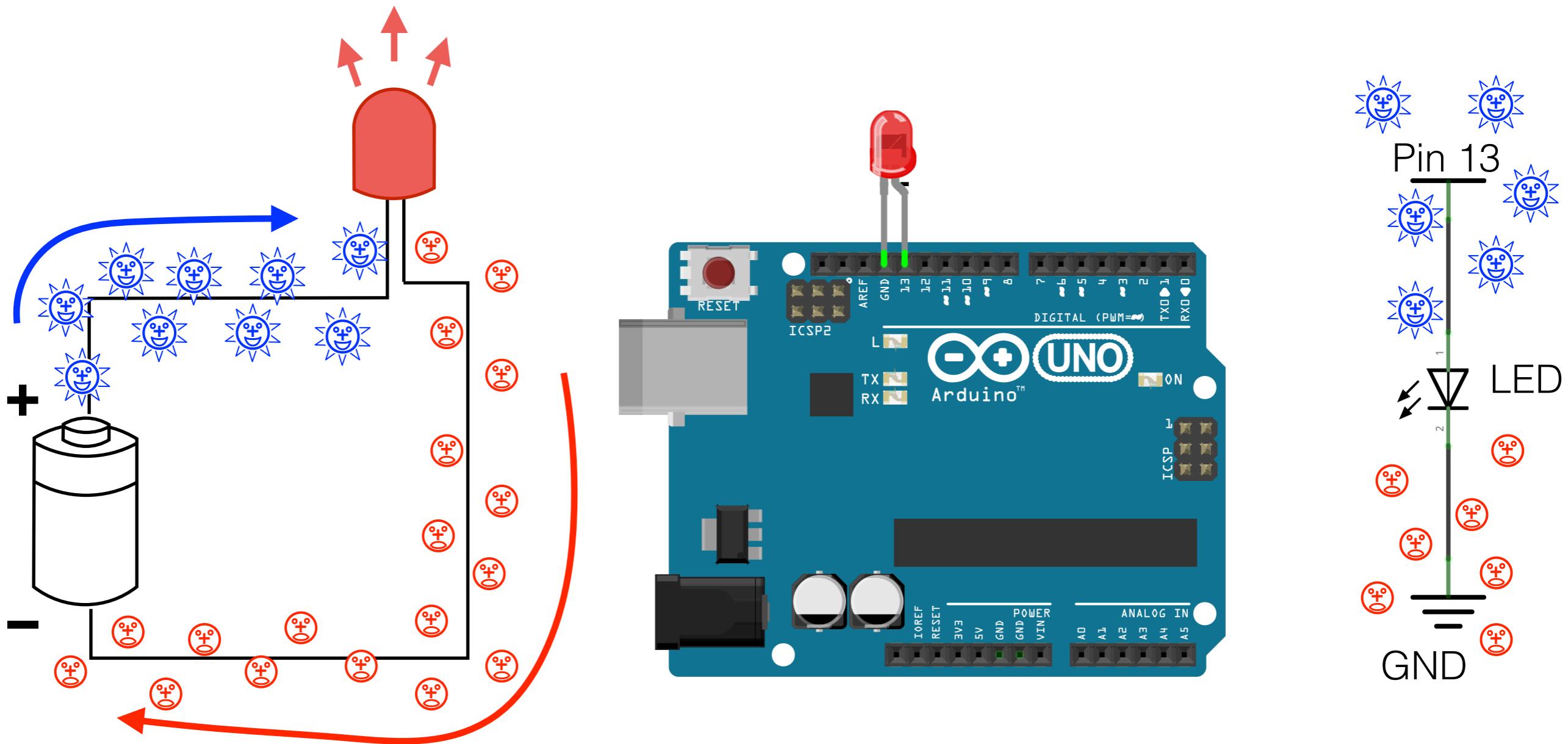


Less energetic charge particle

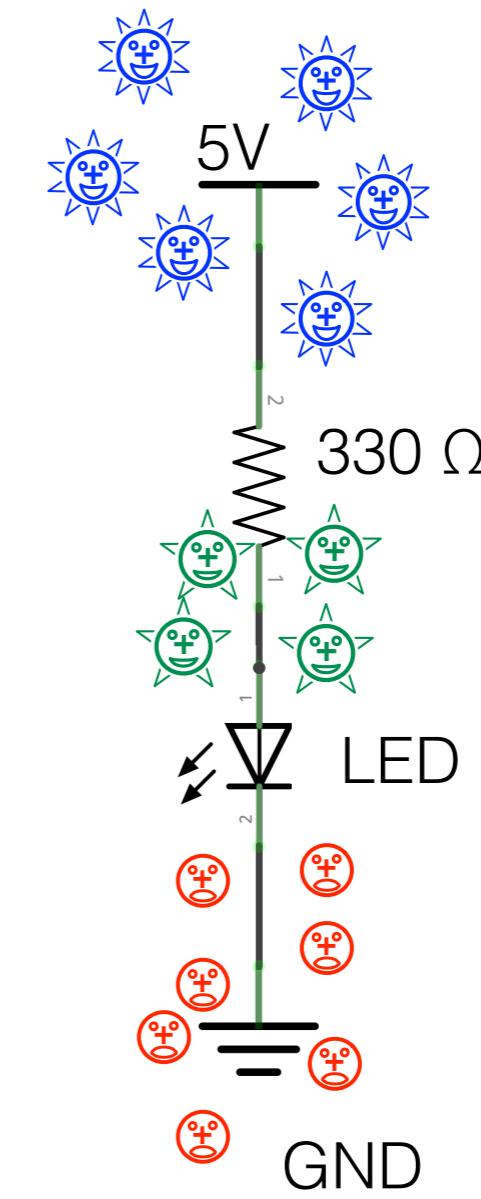
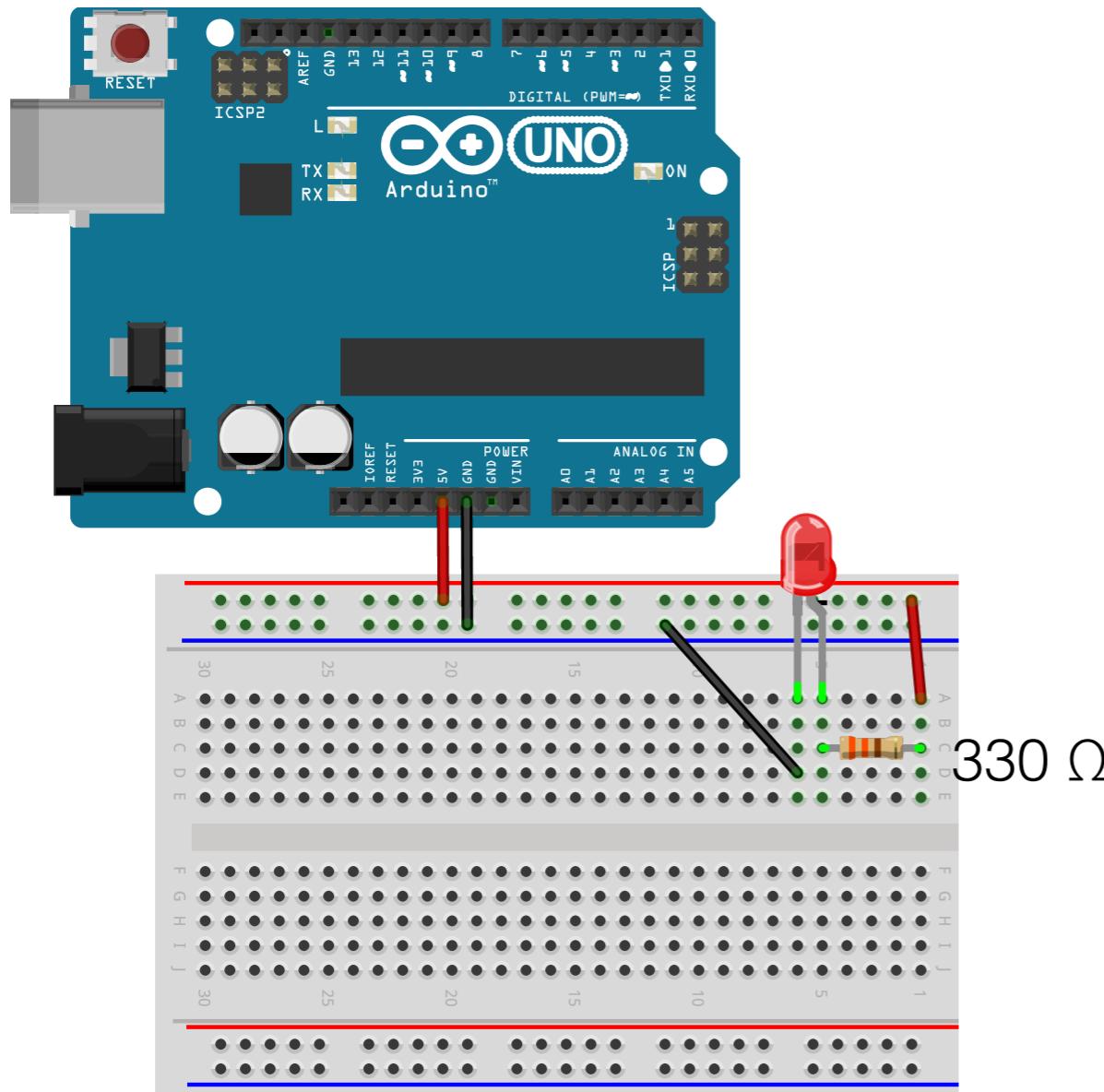


Exhausted charge particle

Simple Circuit

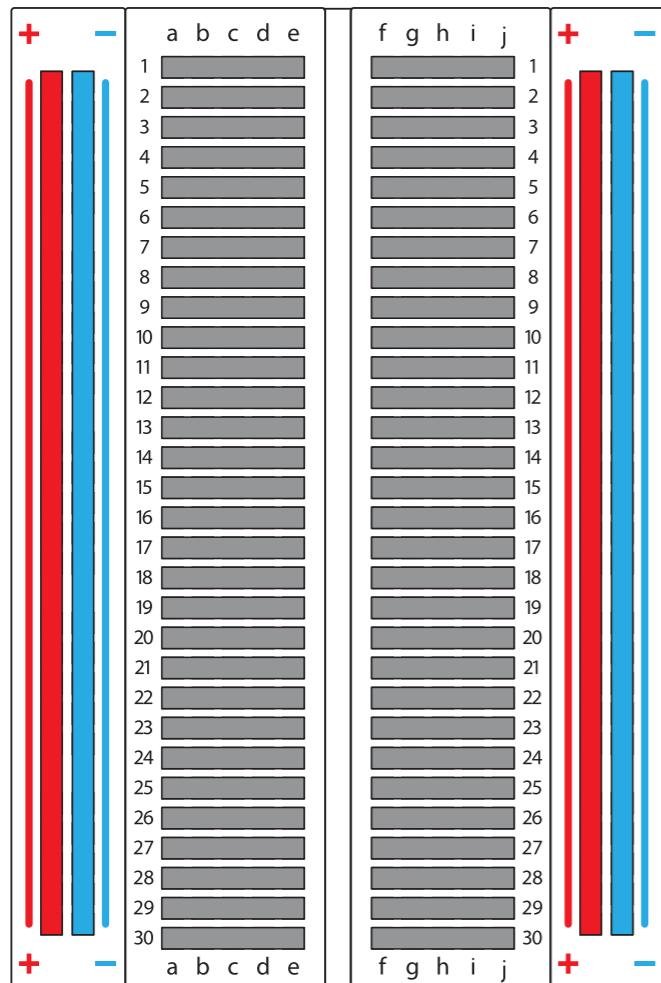


A Happier LED

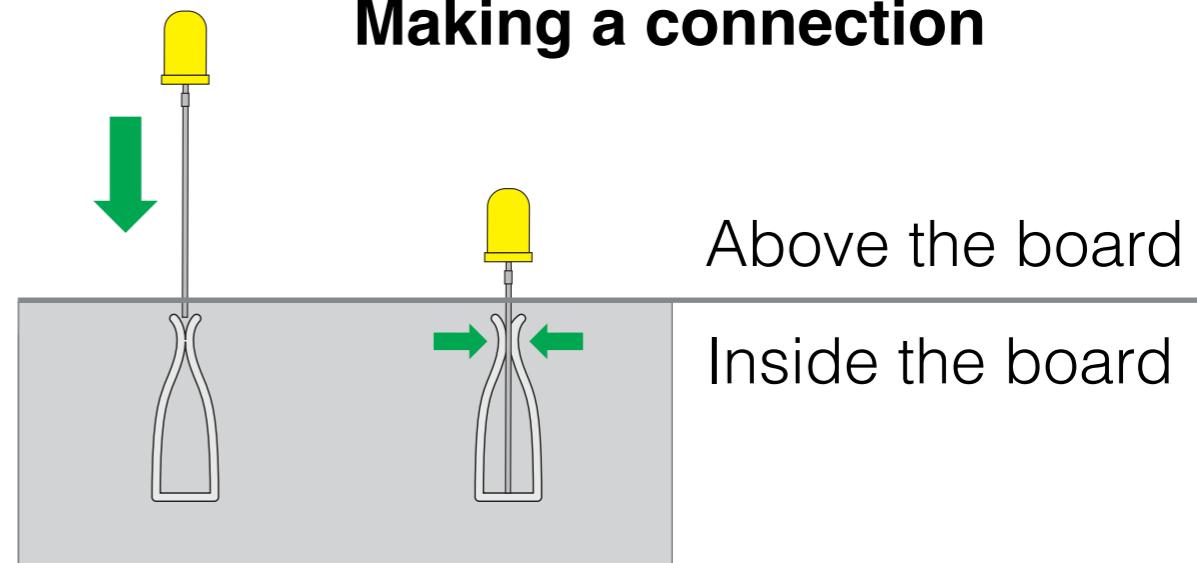


The Breadboard

- + Runs power along column
- Runs ground along column
- Each numbered row has 5 connected sockets

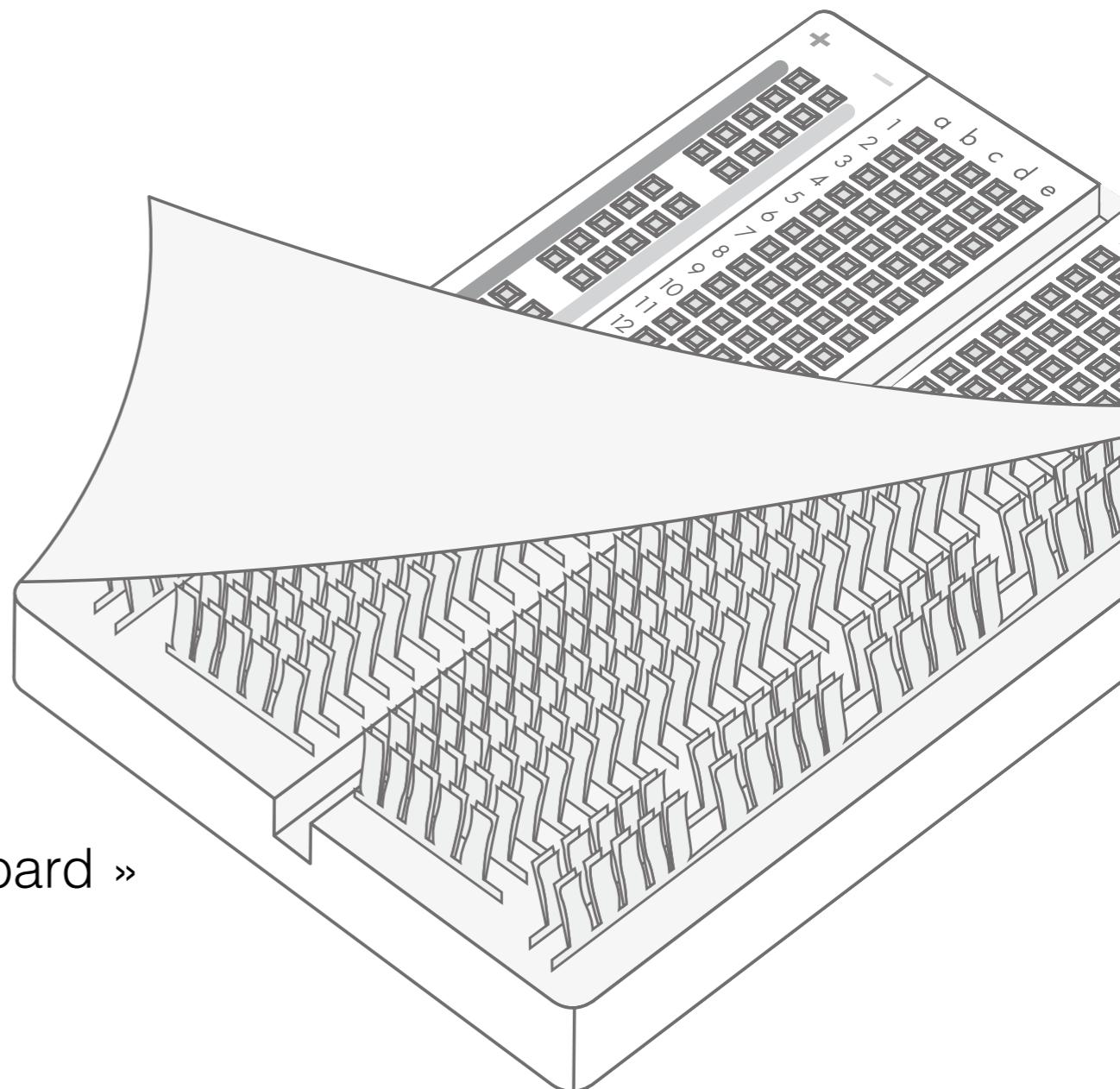


Making a connection



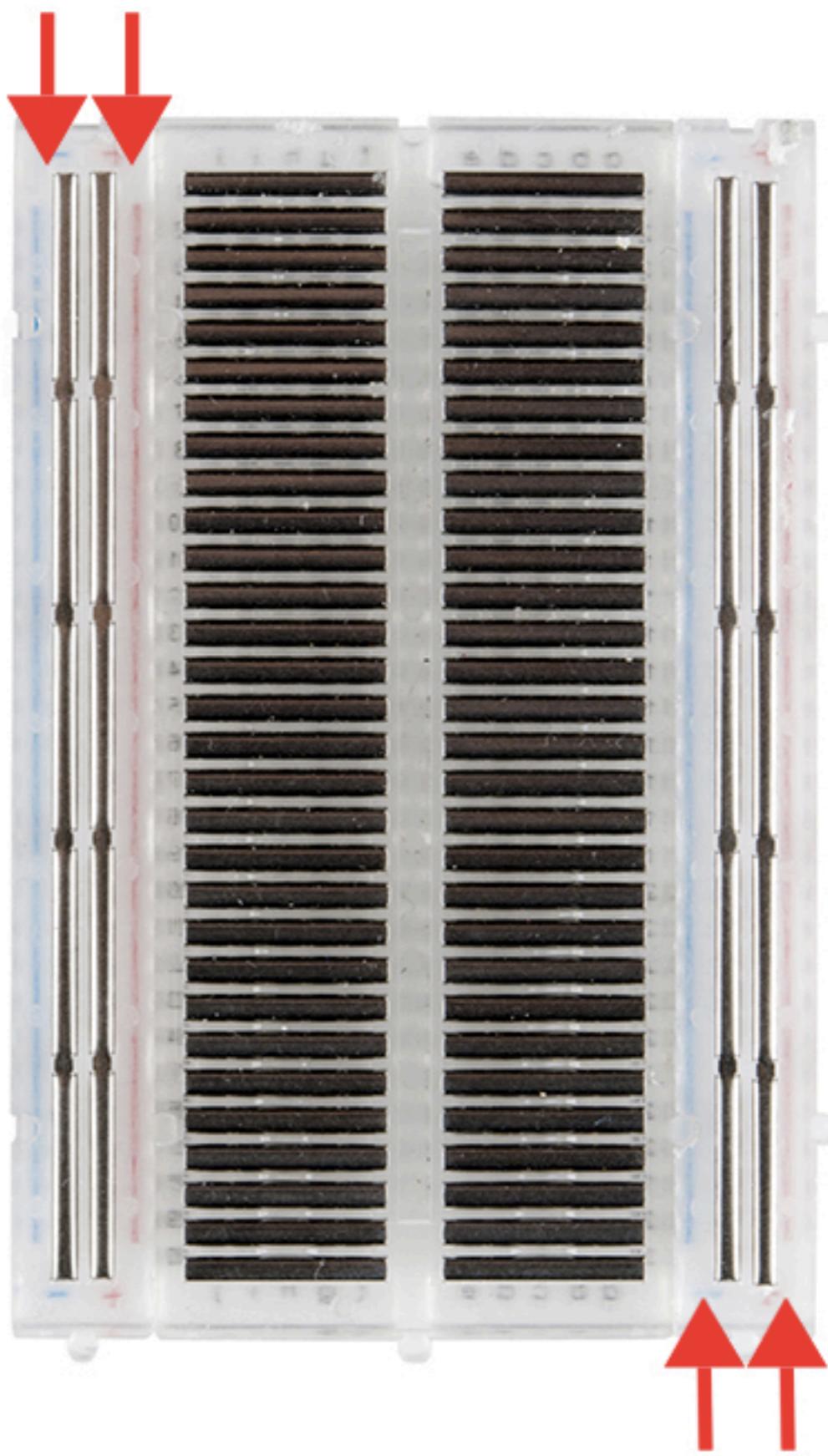
Above the board

Inside the board

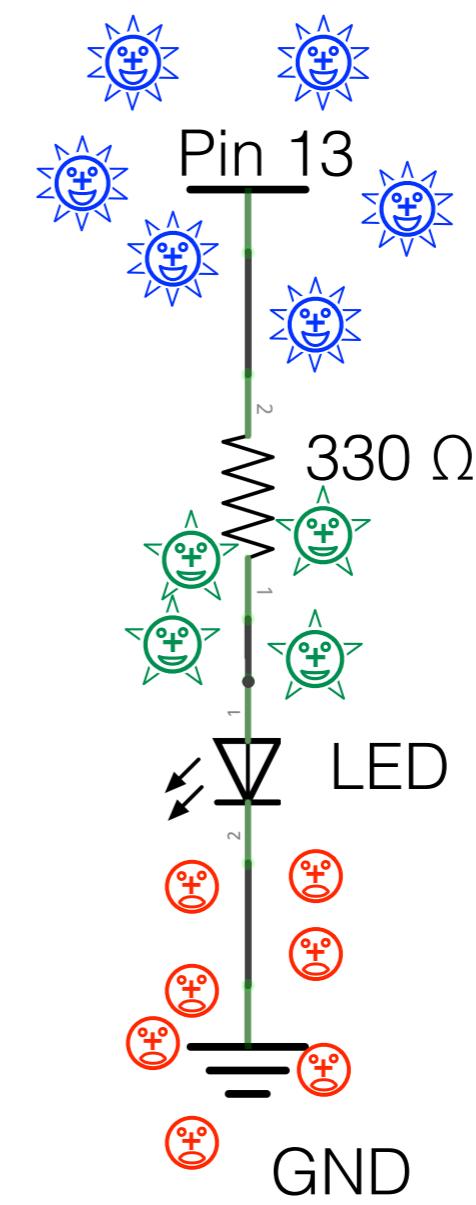
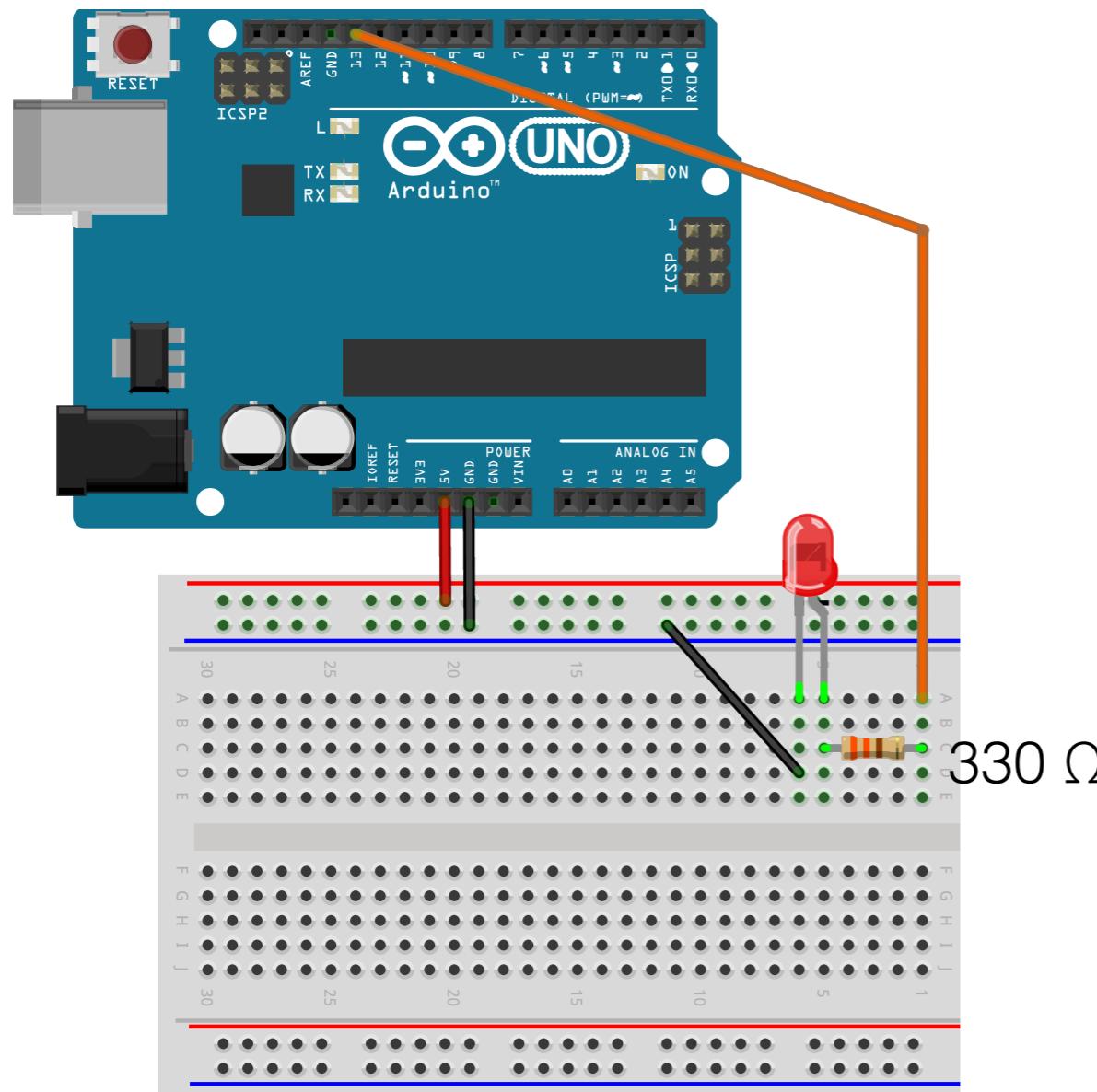


Inside the board »

	a	b	c	d	e	f	g	h	i	j		
1	■	■	■	■	■	■	■	■	■	■	3	
2	■	■	■	■	■	■	■	■	■	■	2	
3	■	■	■	■	■	■	■	■	■	■	3	
4	■	■	■	■	■	■	■	■	■	■	4	
5	■	■	■	■	■	■	■	■	■	■	5	
6	■	■	■	■	■	■	■	■	■	■	6	
7	■	■	■	■	■	■	■	■	■	■	7	
8	■	■	■	■	■	■	■	■	■	■	8	
9	■	■	■	■	■	■	■	■	■	■	9	
10	■	■	■	■	■	■	■	■	■	■	10	
11	■	■	■	■	■	■	■	■	■	■	11	
12	■	■	■	■	■	■	■	■	■	■	12	
13	■	■	■	■	■	■	■	■	■	■	13	
14	■	■	■	■	■	■	■	■	■	■	14	
15	■	■	■	■	■	■	■	■	■	■	15	
16	■	■	■	■	■	■	■	■	■	■	16	
17	■	■	■	■	■	■	■	■	■	■	17	
18	■	■	■	■	■	■	■	■	■	■	18	
19	■	■	■	■	■	■	■	■	■	■	19	
20	■	■	■	■	■	■	■	■	■	■	20	
21	■	■	■	■	■	■	■	■	■	■	21	
22	■	■	■	■	■	■	■	■	■	■	22	
23	■	■	■	■	■	■	■	■	■	■	23	
24	■	■	■	■	■	■	■	■	■	■	24	
25	■	■	■	■	■	■	■	■	■	■	25	
26	■	■	■	■	■	■	■	■	■	■	26	
27	■	■	■	■	■	■	■	■	■	■	27	
28	■	■	■	■	■	■	■	■	■	■	28	
29	■	■	■	■	■	■	■	■	■	■	29	
30	■	■	■	■	■	■	■	■	■	■	30	



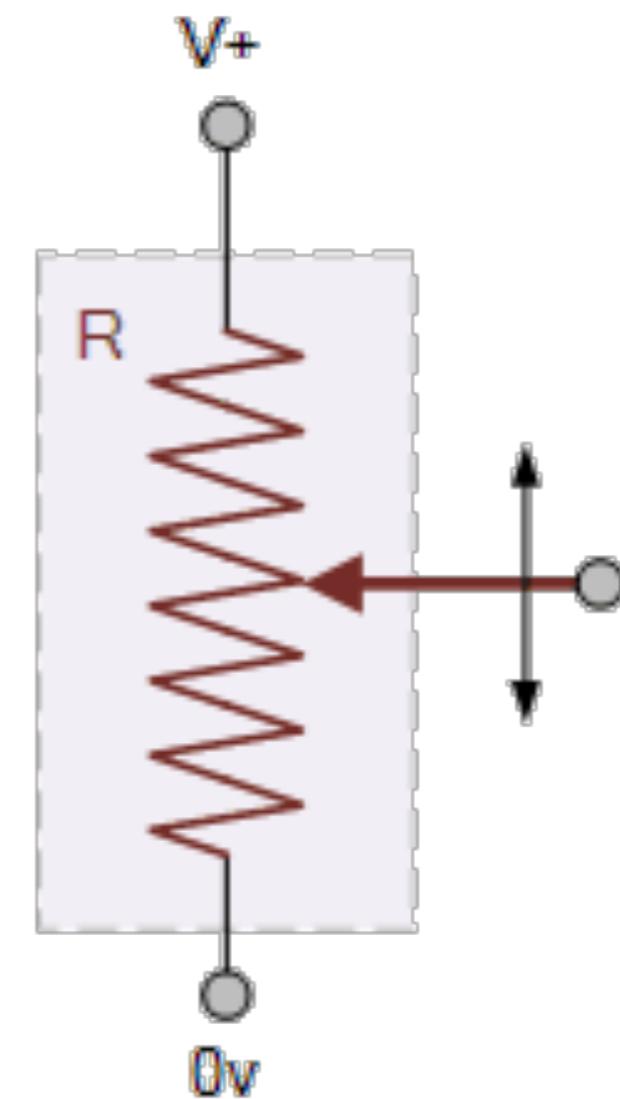
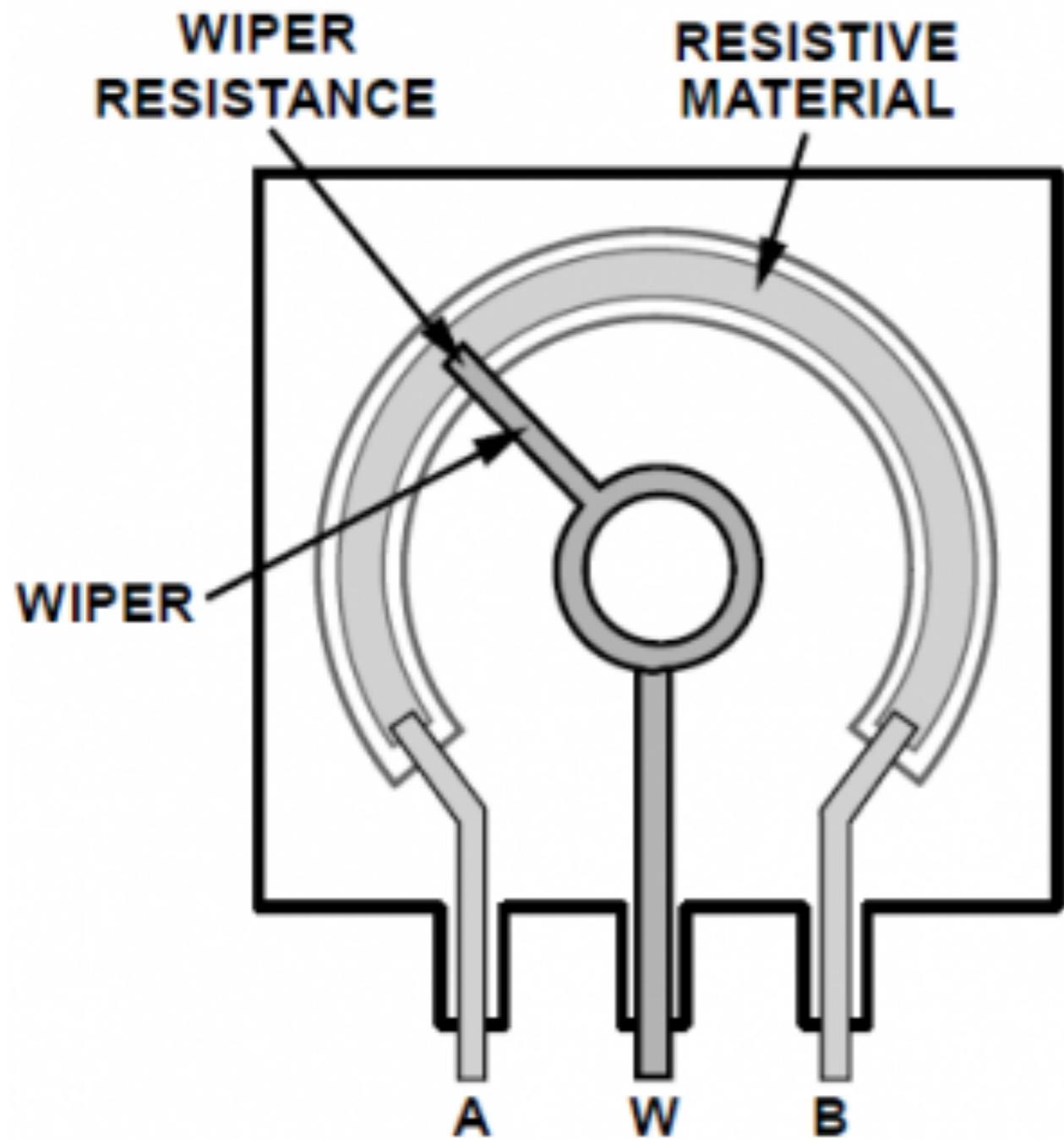
A Blinking LED



Potentiometers (Variable Resistors)

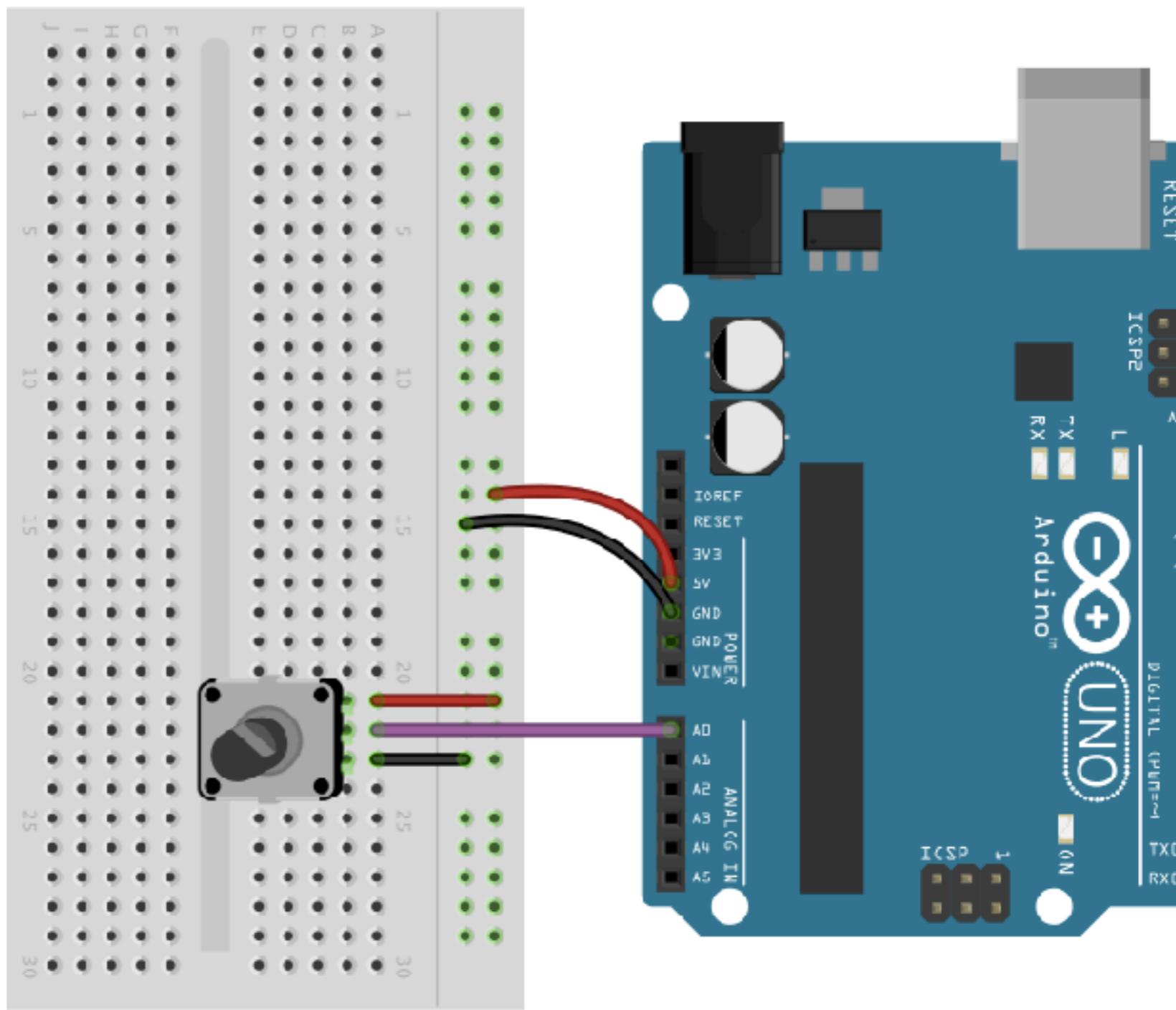


Potentiometers

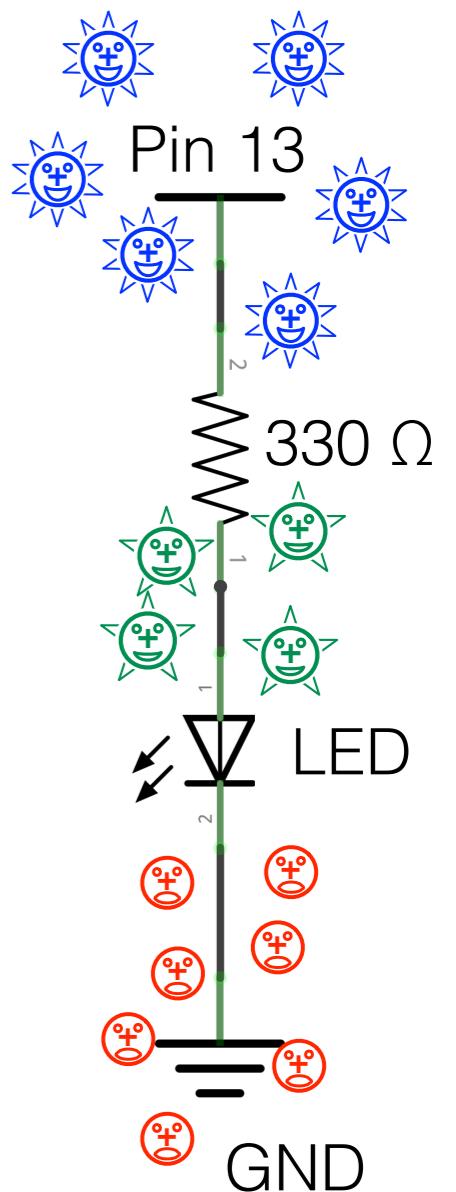
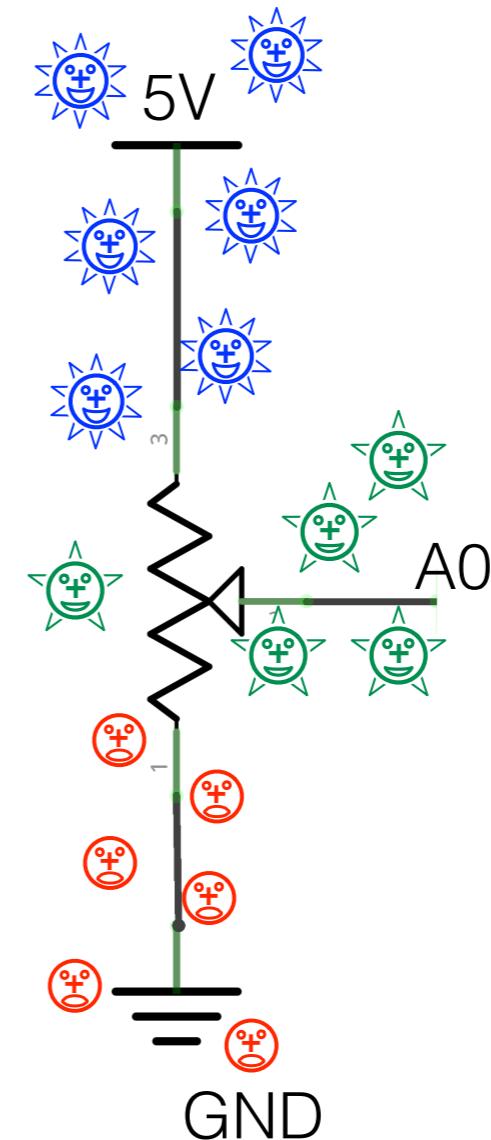
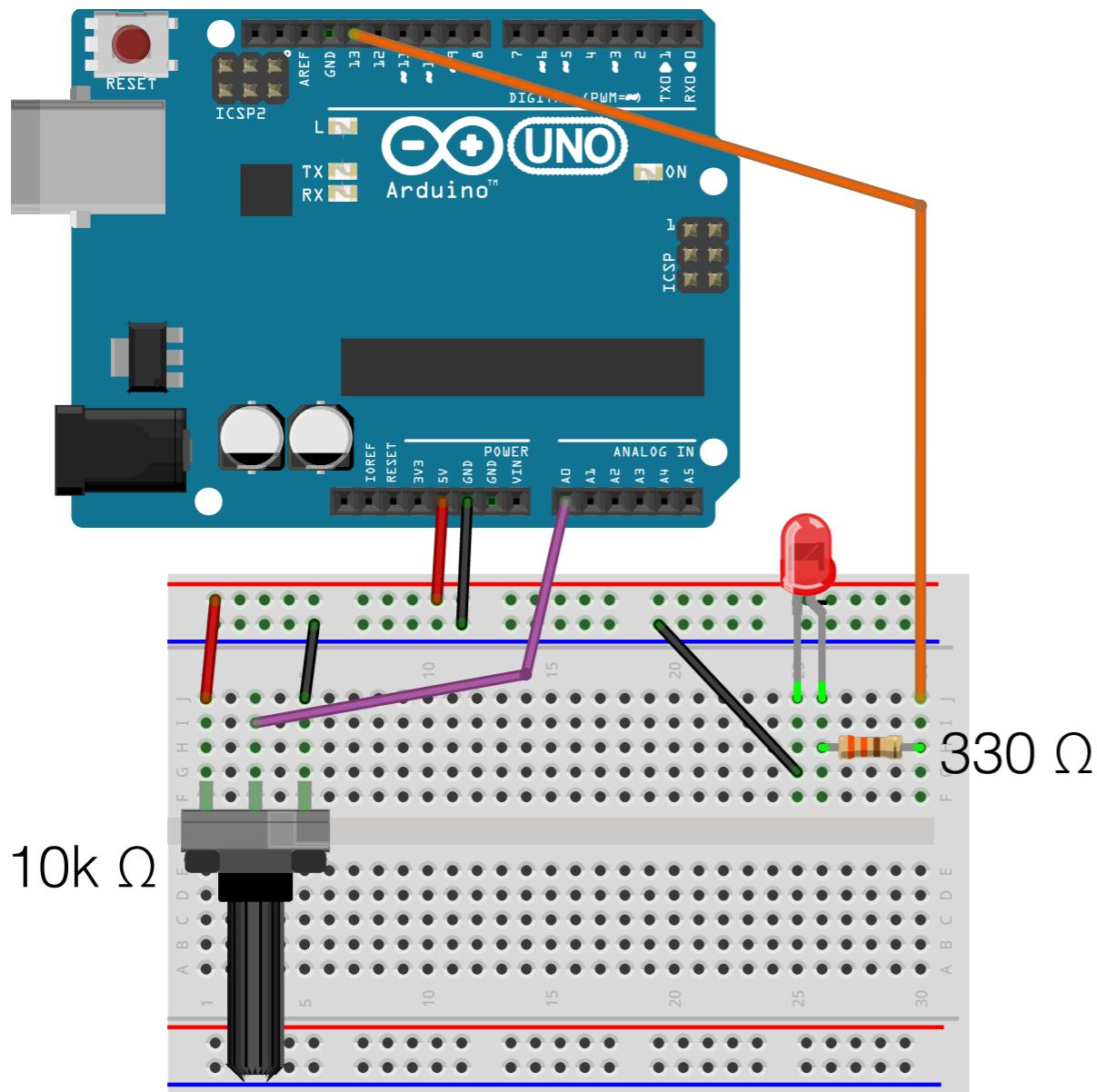


analogRead

- Analog to Digital Converter (ADC)
- Pins A0 to A5
- Returns a value from 0 to 1023
- 0 Volts? Returns 0
5 Volts? Returns 1023
2.5 Volts? Returns 512



Control the Blinking



Control the Blinking, Code

```
/*
Analog Input
[ ... ]

This example code is in the public domain.

*/
int sensorPin = A0;      // select the input pin for the potentiometer
int ledPin = 13;         // select the pin for the LED
int sensorValue = 0;     // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```

File > Examples > Analog > AnalogInput

Debug the Blinking

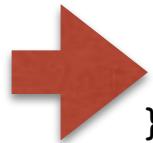
```
/*
Analog Input
[...]

This example code is in the public domain.

*/
int sensorPin = A0;      // select the input pin for the potentiometer
int ledPin = 13;         // select the pin for the LED
int sensorValue = 0;     // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
  // open a 9600-baud serial connection:
  Serial.begin(9600);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // write the sensor value to the serial interface:
  Serial.println(sensorValue);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```



File > Examples > Analog > AnalogInput

Examples!

- Make the potentiometer control brightness instead of blink rate.
*Hint: try flashing the LED really quickly! The **analogWrite** function might help!*
- Wire up 8 LEDs to 8 digital output pins. Use the potentiometer to control how many of the LEDs are on — a level meter!
- Control the red, green, and blue components of an RGB LED using three potentiometers.
- *Challenge:* Blink two LEDs, controlling the rate of each independently with its own potentiometer. *Hint: You can't use **delay()** anymore! Look at the **millis()** function and the **BlinkWithoutDelay** example.*

Arduino is...

- ✓ Small, programmable microcontroller.
- ✓ Software that runs on Mac, PC, and Linux. (IDE)
- Learning platform (for electronics & programming).
- Community of people sharing code & ideas.

Common Components



Motors! Locomotion!

- DC Motors
- Stepper Motors
- Servos

