

# Getting Started with Arduino

J.D. Zamfirescu  
Fall 2017

# What is Arduino?

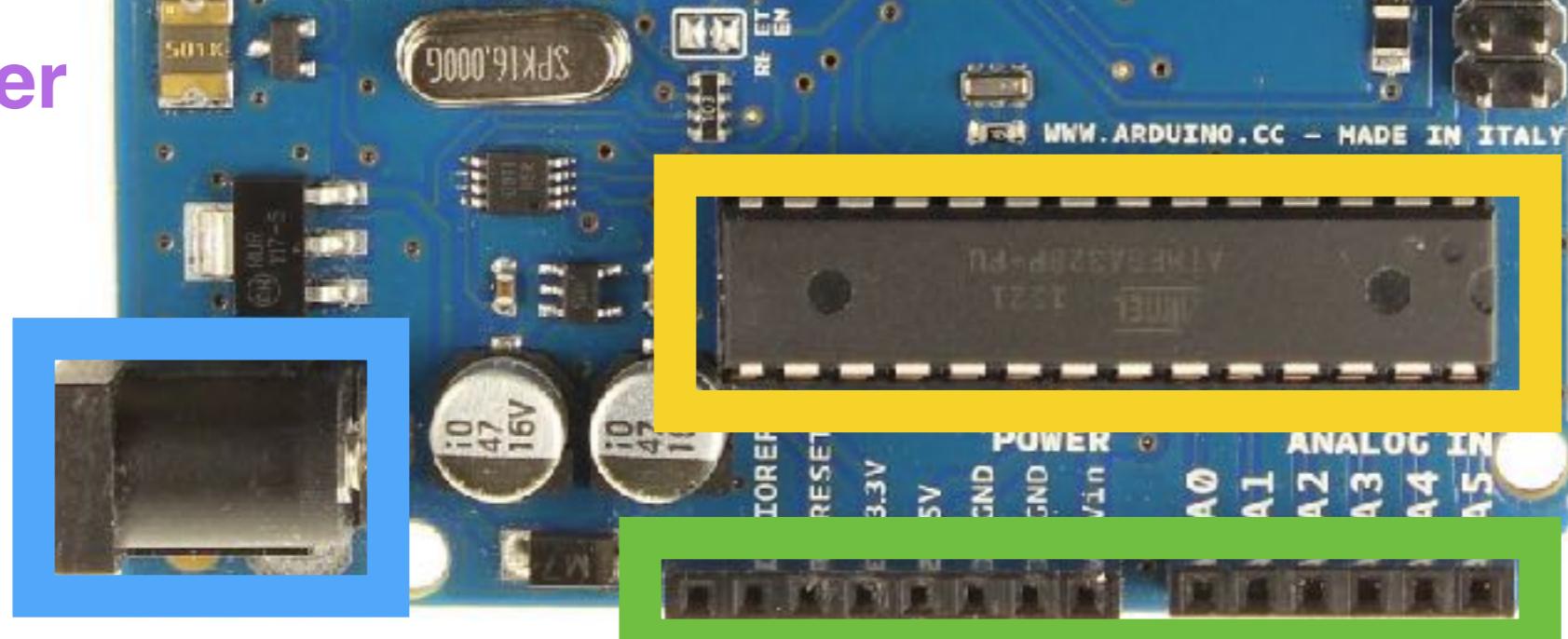
# Arduino is...

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

Reset Button



USB / Power



DC Power

Digital Inputs & Outputs

ATmega 328

Power & Analog Inputs

# Arduino Hardware: UNO

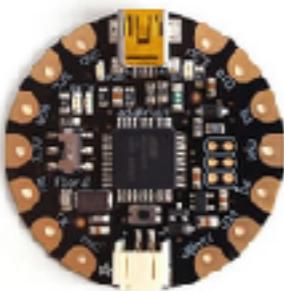
# The Arduino Ecosystem



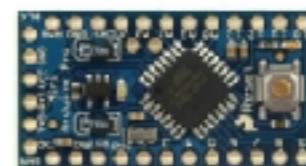
UNO



MEGA



FLORA

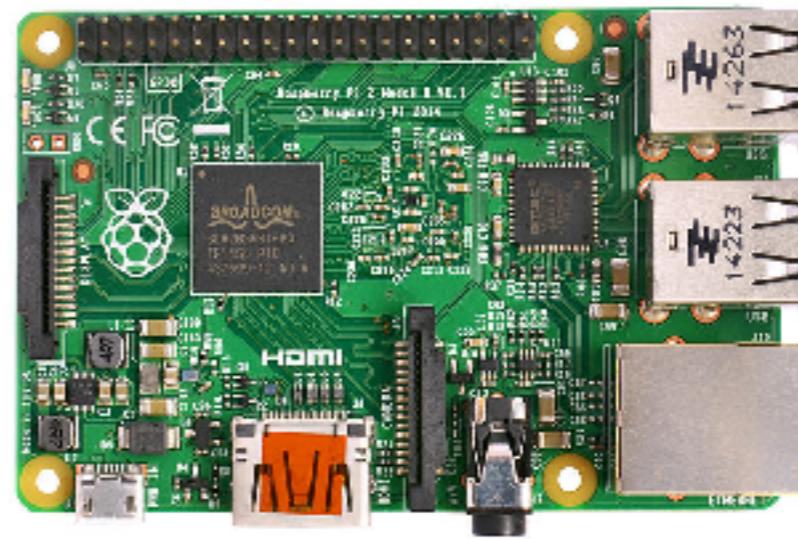


Pro Mini

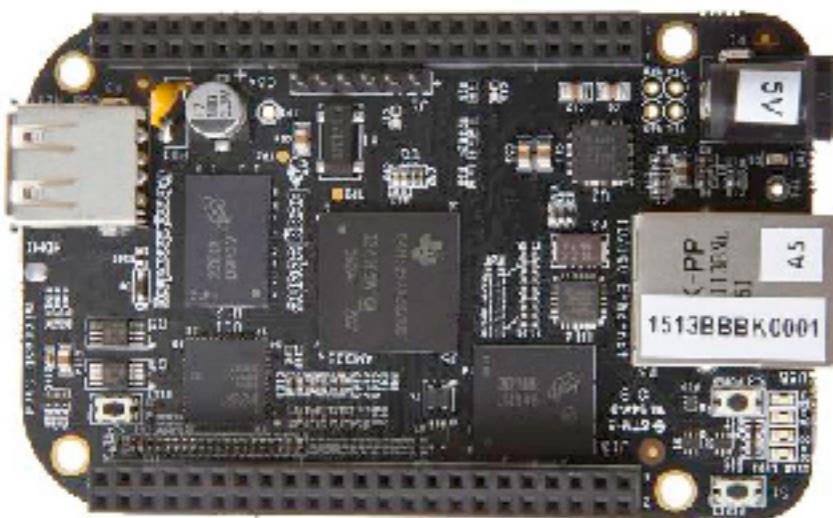
# Arduino's Many Competitors



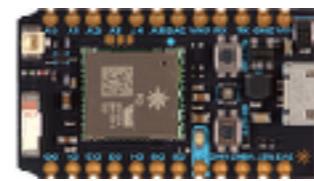
UNO



Raspberry Pi



BeagleBone Black



Particle Photon

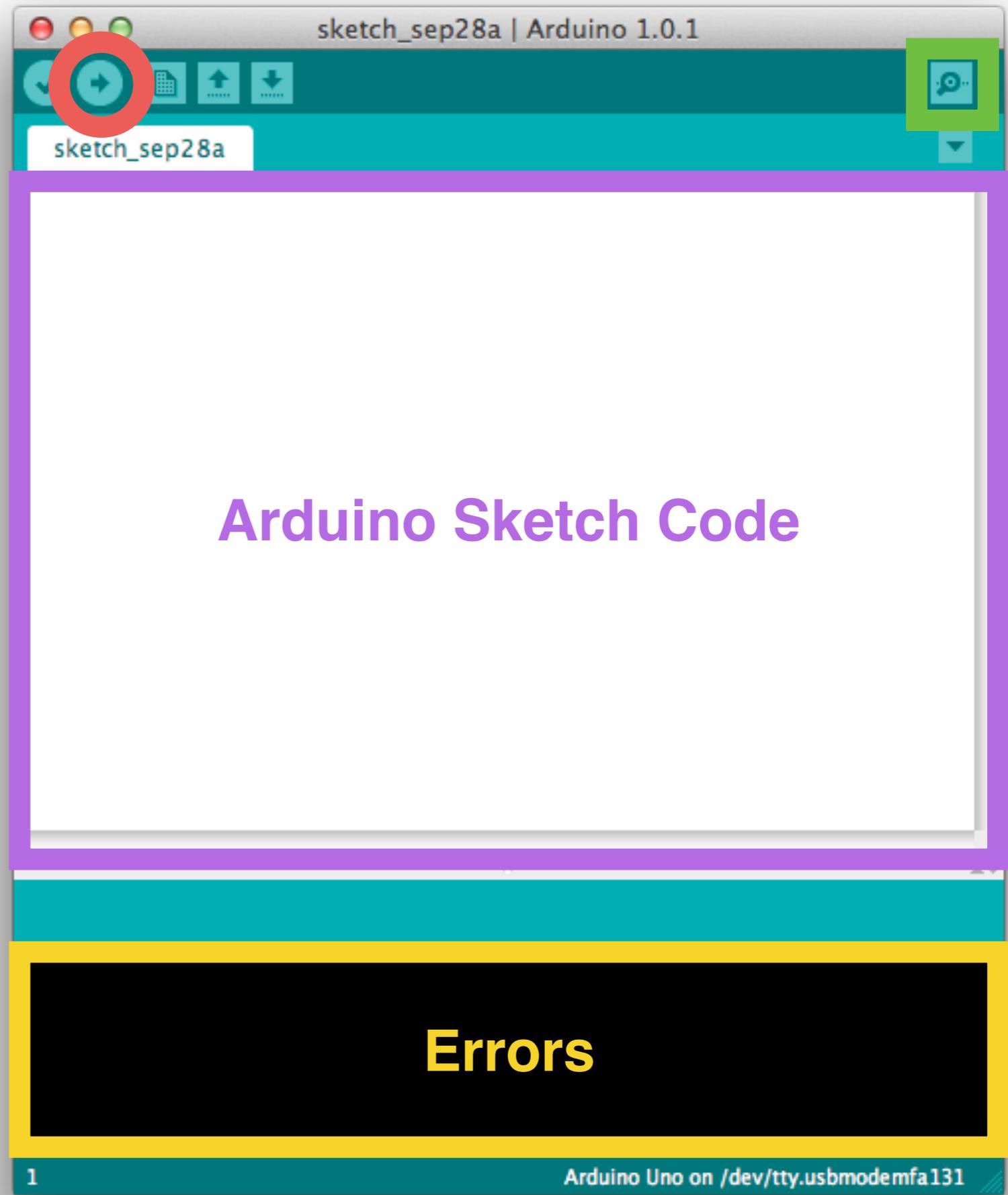


AdaFruit HUZZAH

# Arduino Software

Upload

Serial Monitor



# Anatomy of a Sketch

**Start Comment**

Blink

Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

**End Comment**

**Descriptive Comment**

**Start Comment**

```
// 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);
}
```

**Line Comments**

```
// 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
}
```

***File > Examples > Basics > Blink***

# Anatomy of a Sketch

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/



// 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
}
```

***File > Examples > Basics > Blink***

# Anatomy of a Sketch

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

*/
Hey Arduino,
here's how you setup

// 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);
}
```

**File > Examples > Basics > Blink**

# Anatomy of a Sketch

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/



// the setup function runs once when you press reset or power the board
void setup() {
    // initializes the digital pin as an output.
    pin
}

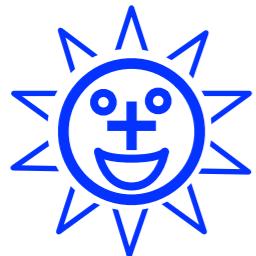
// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);           // turn the LED on (HIGH is the voltage level)
    delay(1000);                    // wait for a second
    digitalWrite(13, LOW);          // make the voltage LOW
    delay(1000);                    // wait for a second
}
```

Hey Arduino,  
here's how you **loop**

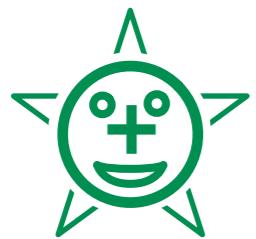
Code block for **loop**

# Electronics is...

## Moving Charge



Highly energetic charge particle

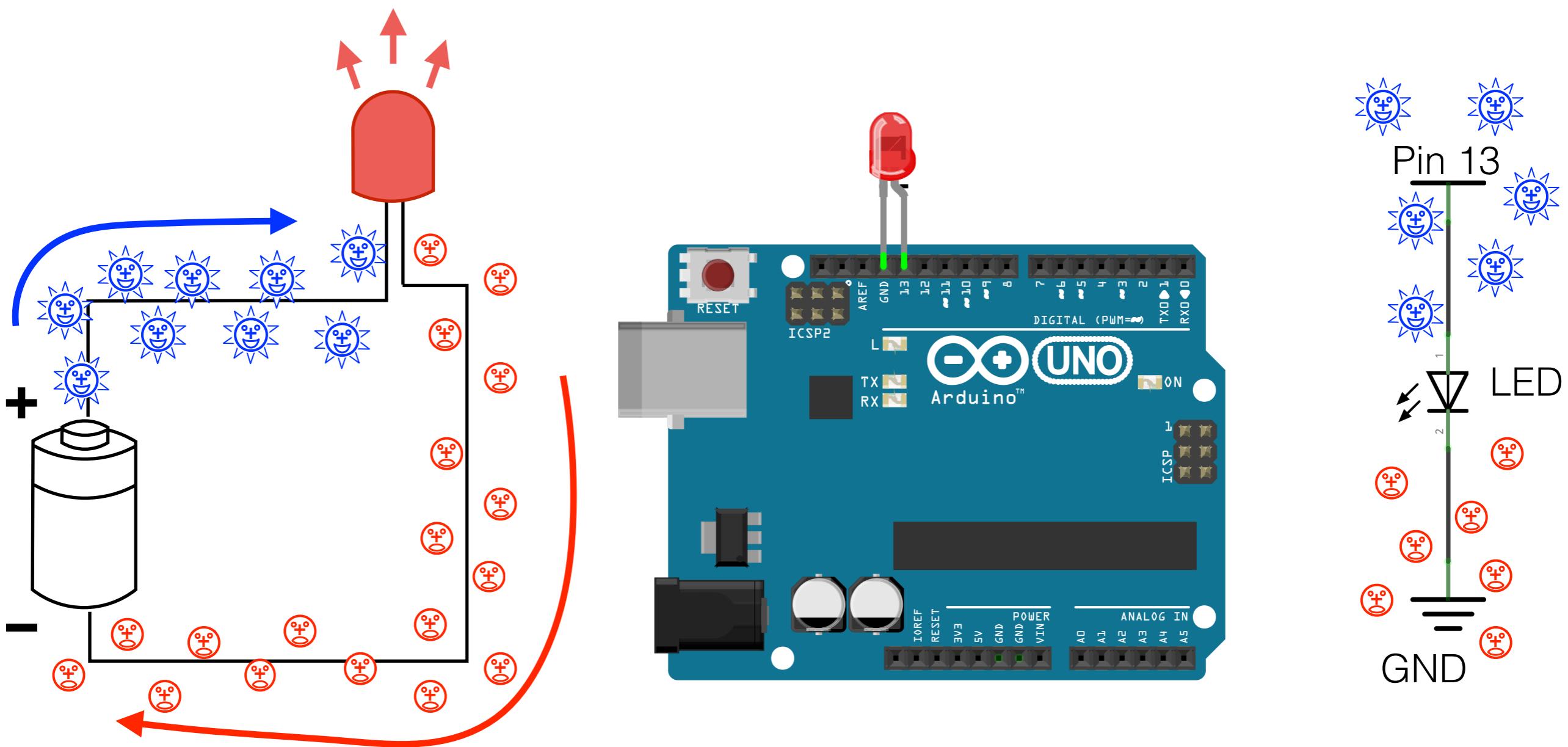


Less energetic charge particle

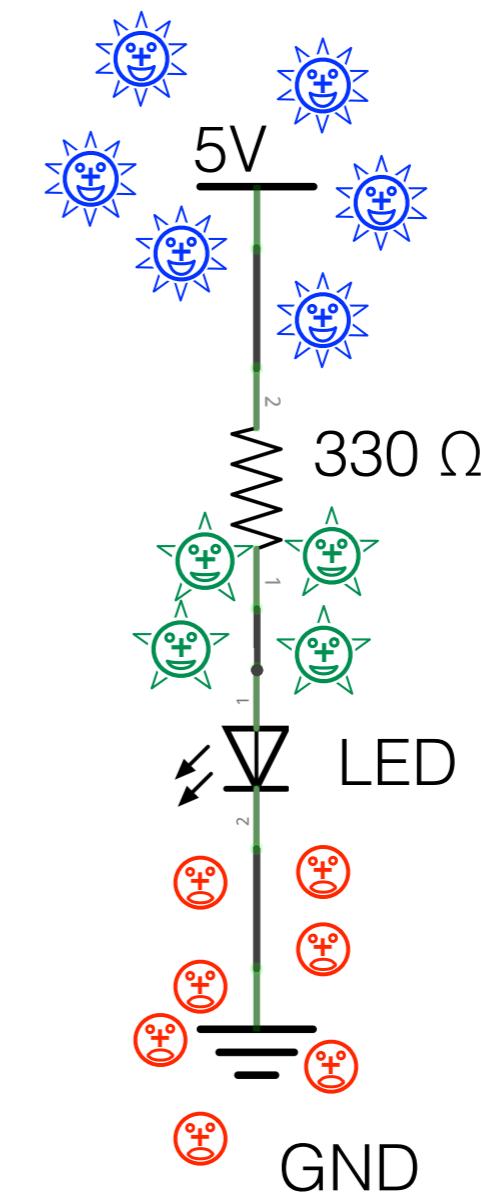
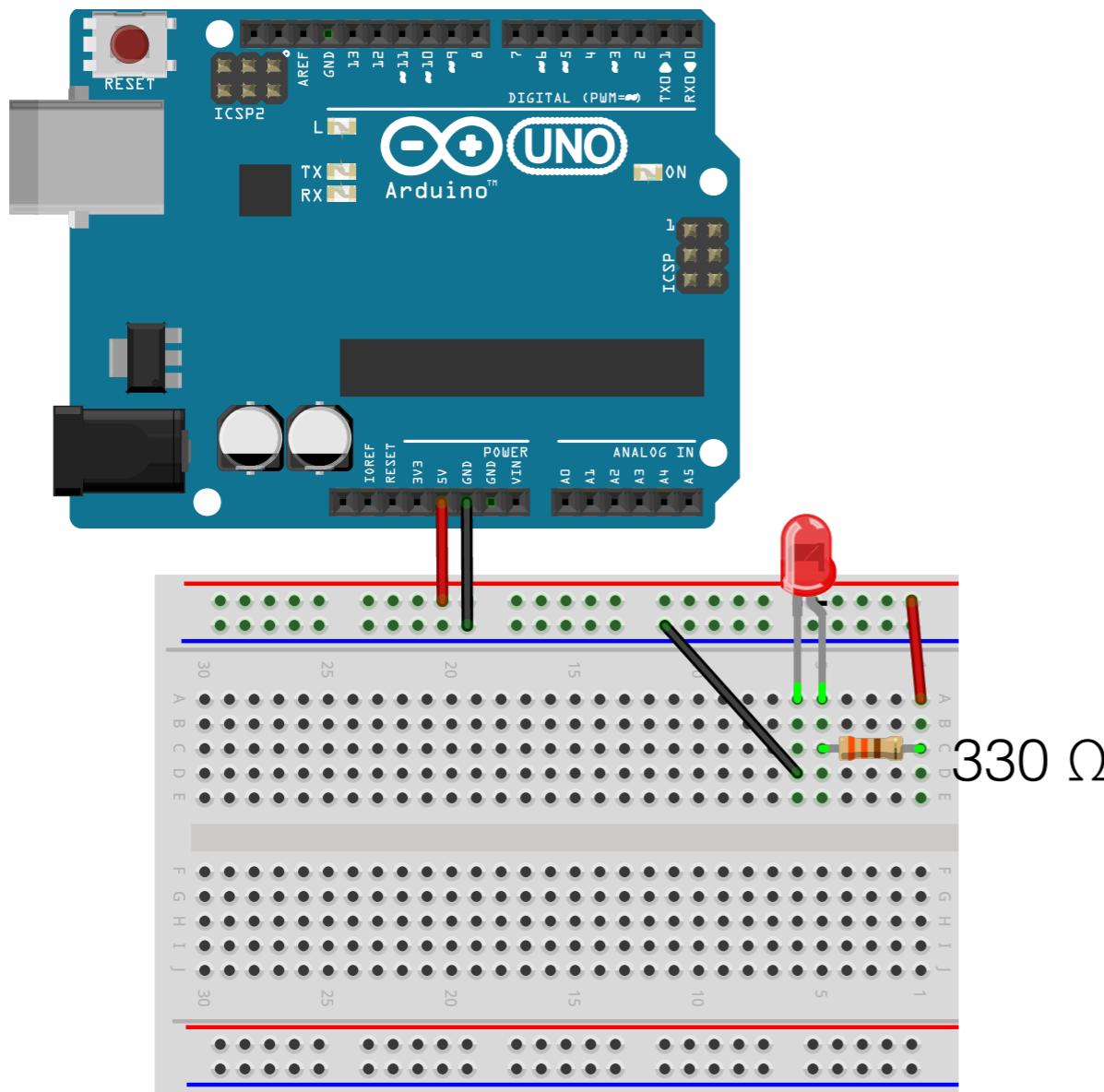


Exhausted charge particle

# Simple Circuit

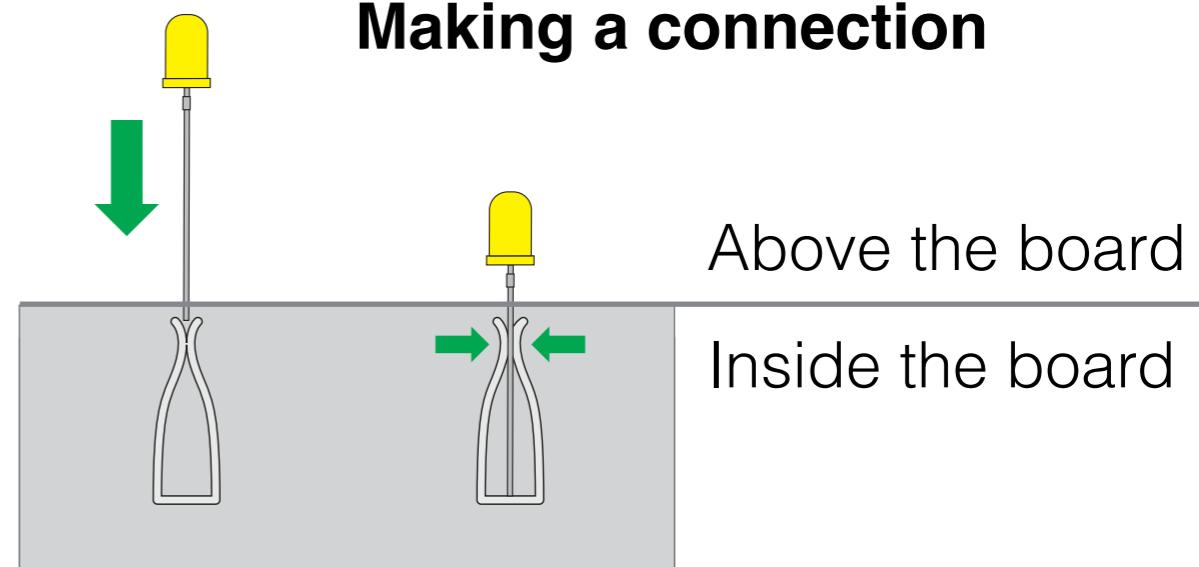
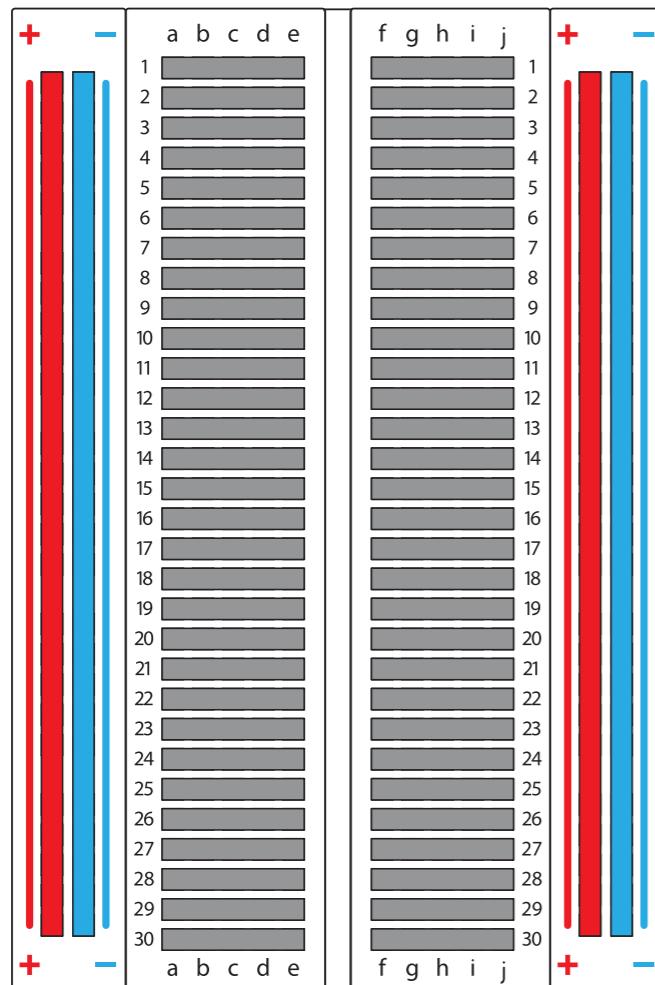


# A Happier LED

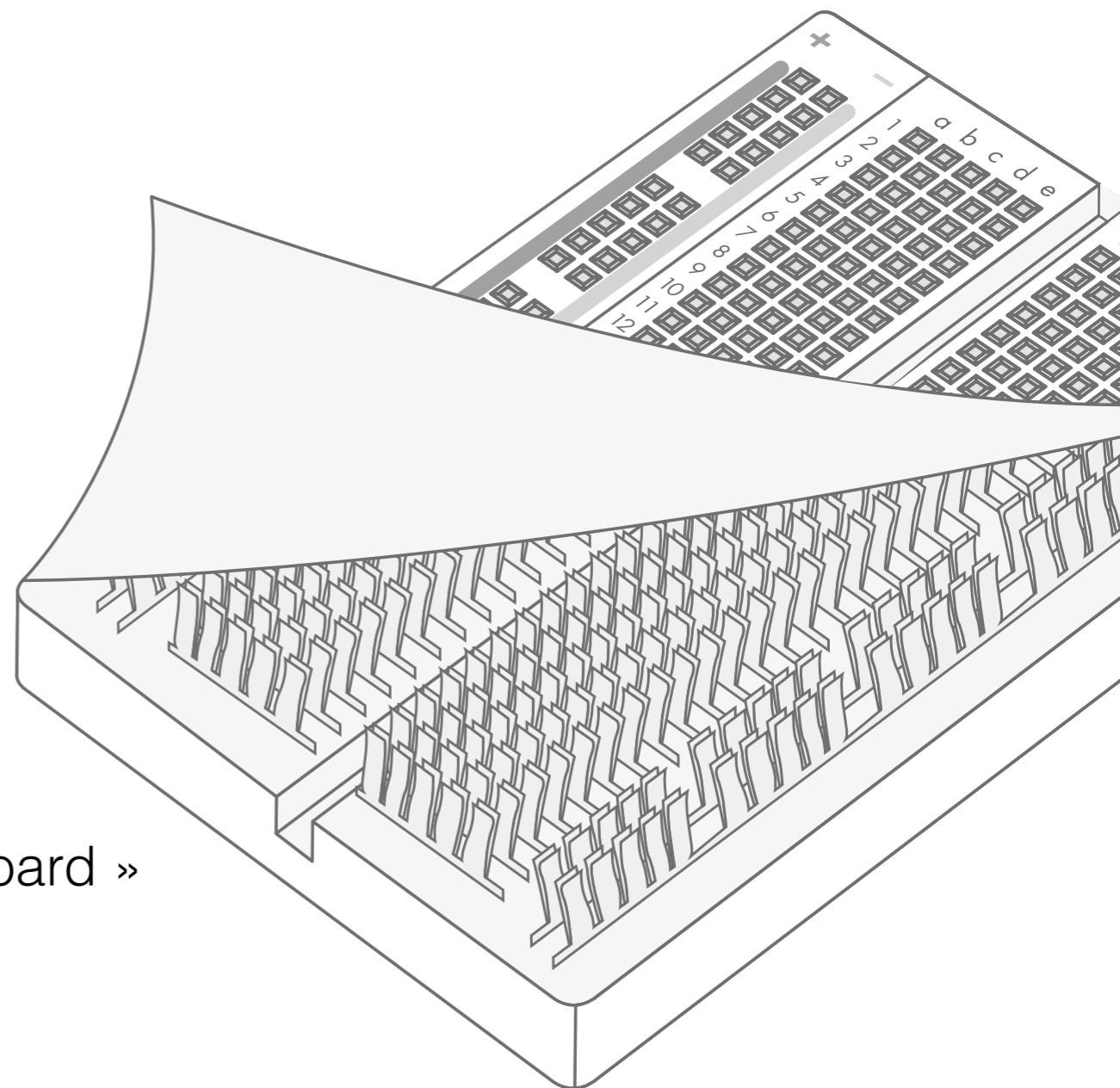


# The Breadboard

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

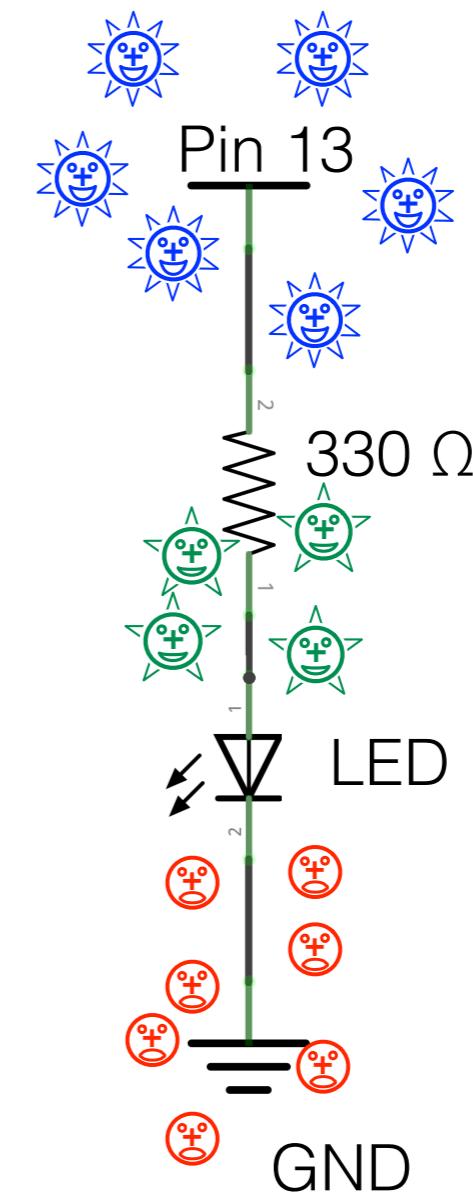
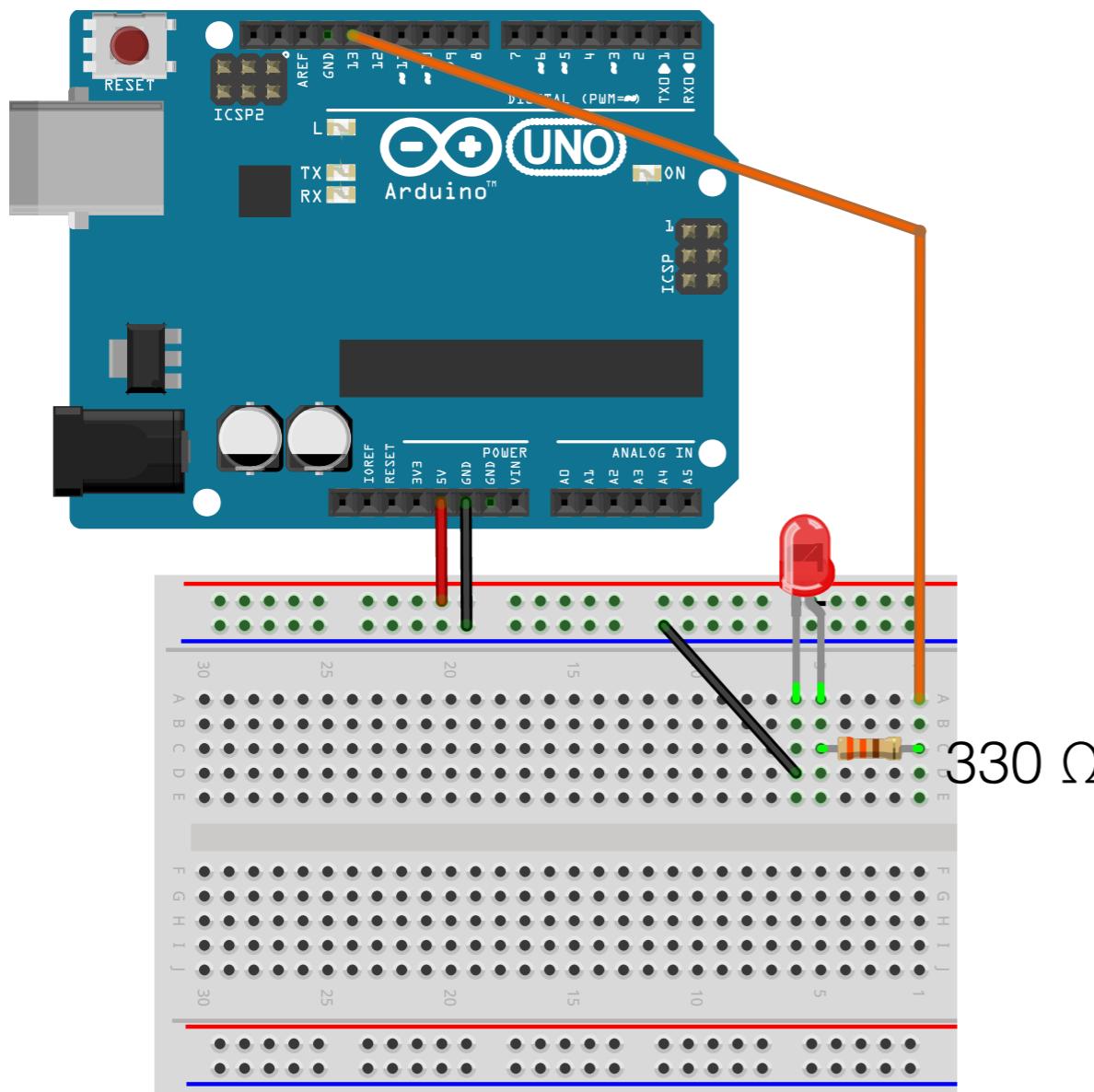


Inside the board »

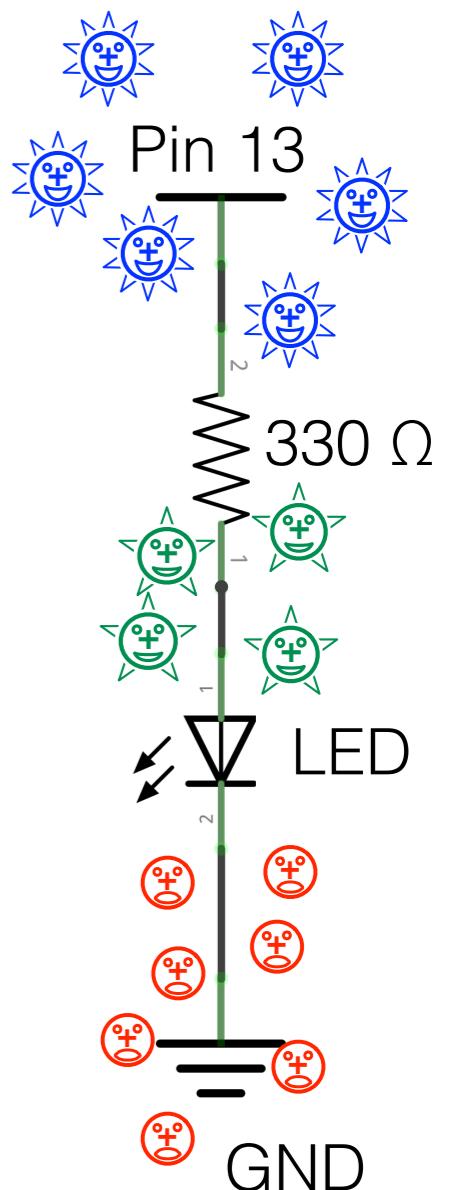
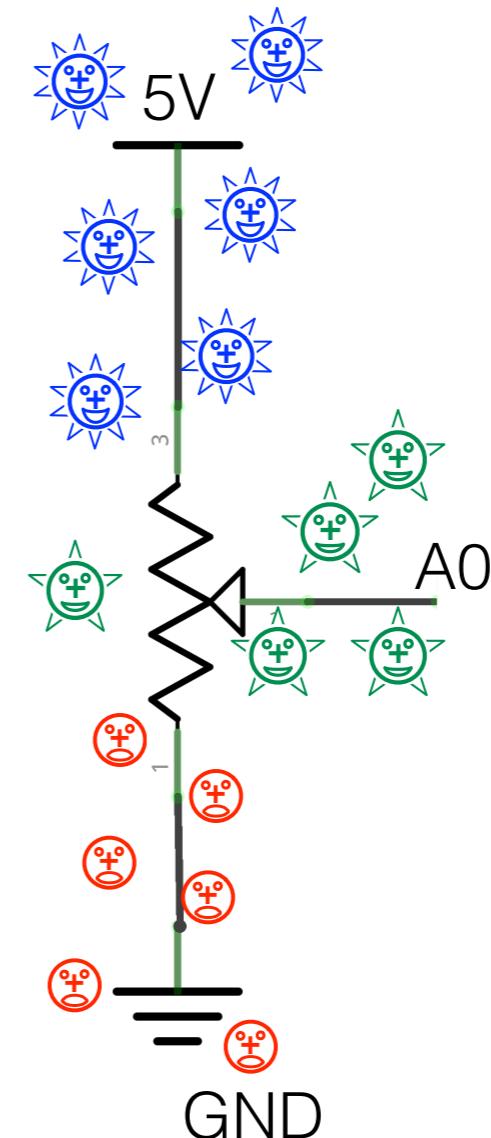
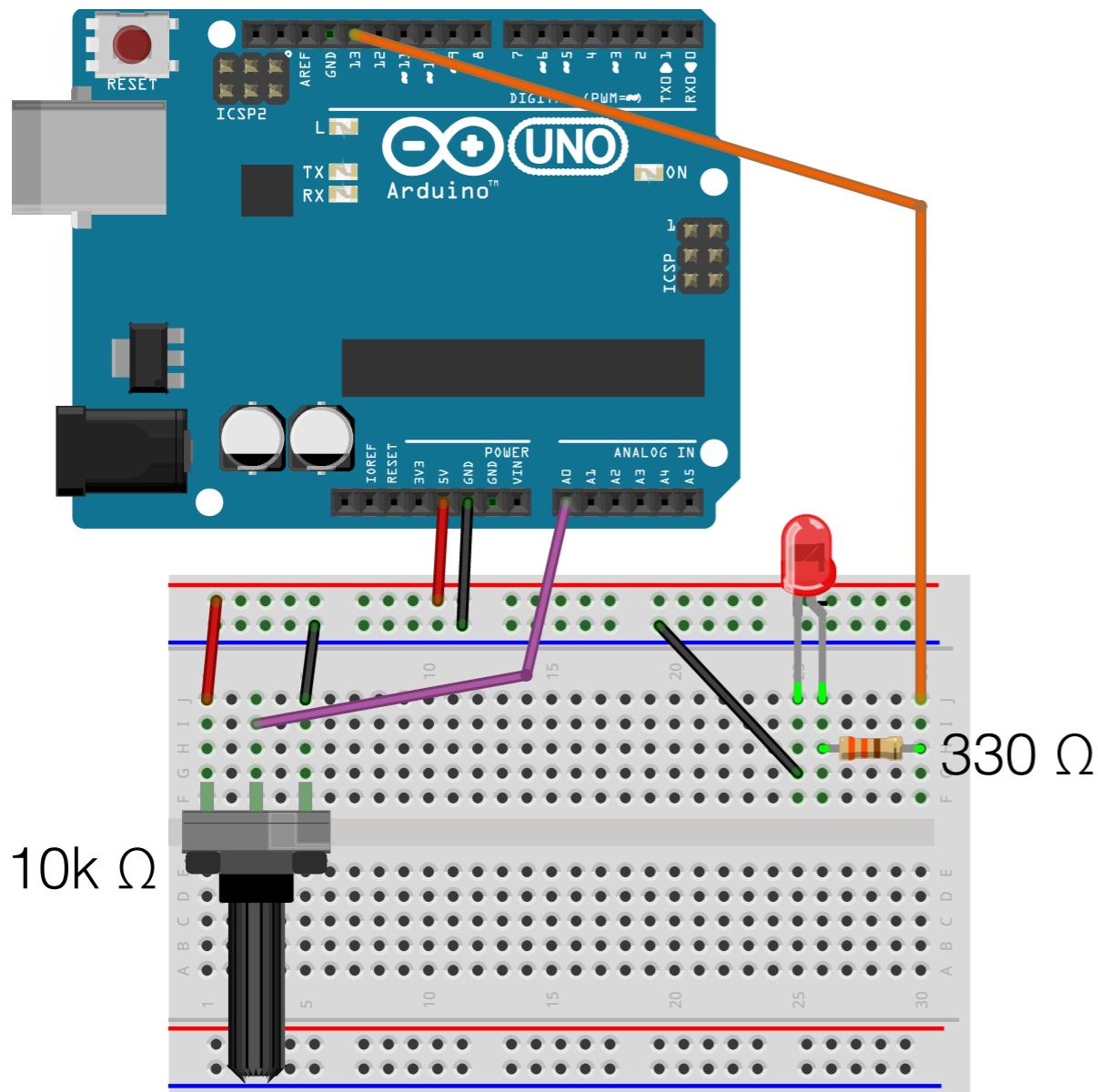


Making a connection

# A Blinking LED



# 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.