

Getting Started with Arduino

J.D. Zamfirescu
Arduino Robotics
CCA, Summer 2014

What is Arduino?

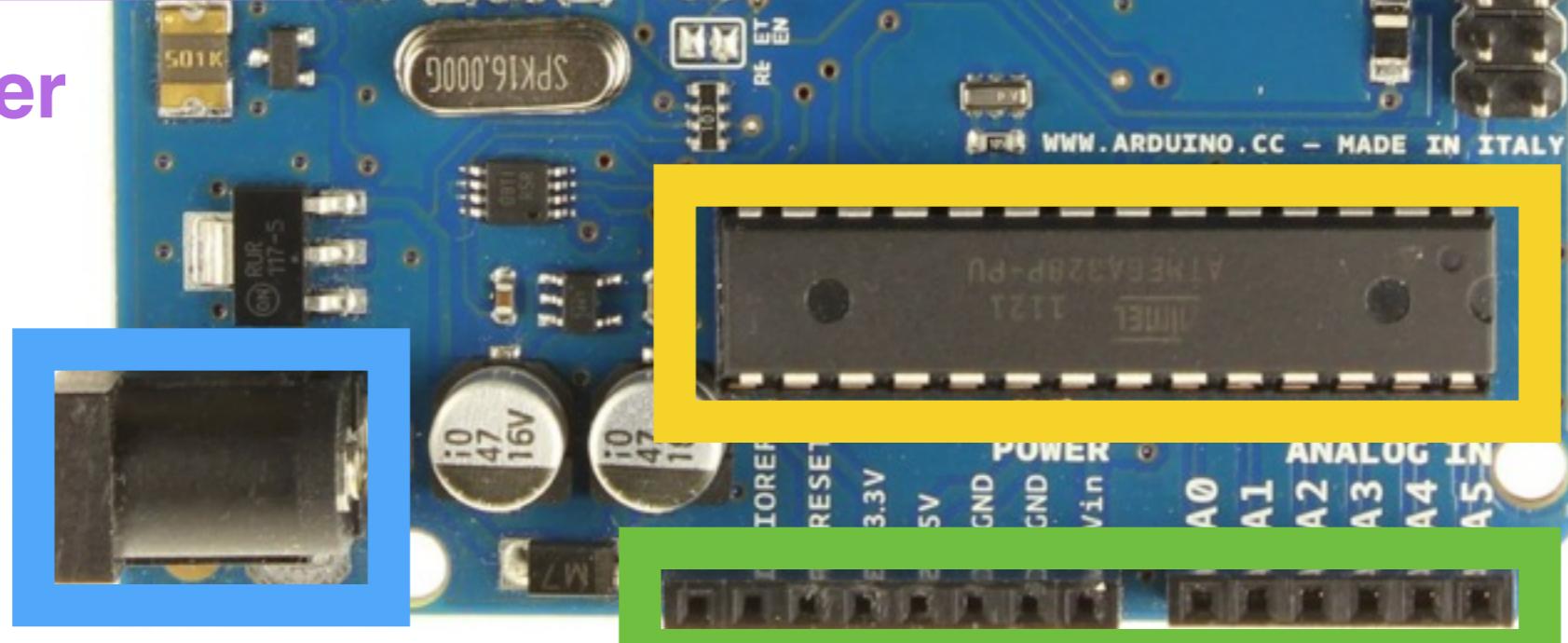
Arduino is...

- 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

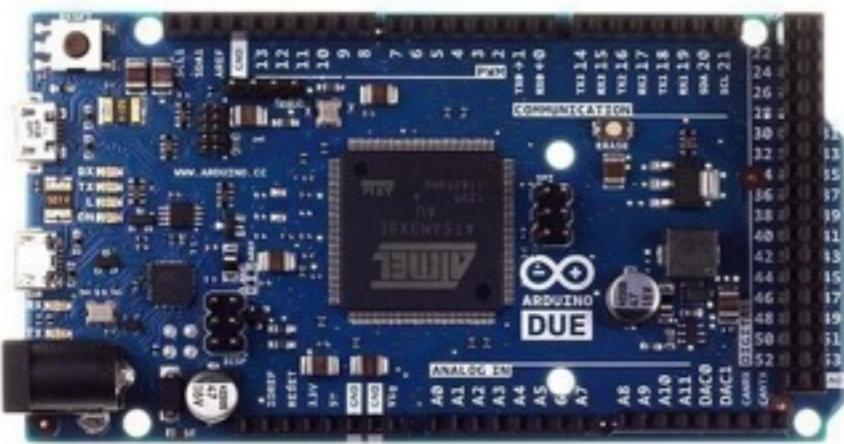
The Arduino Family



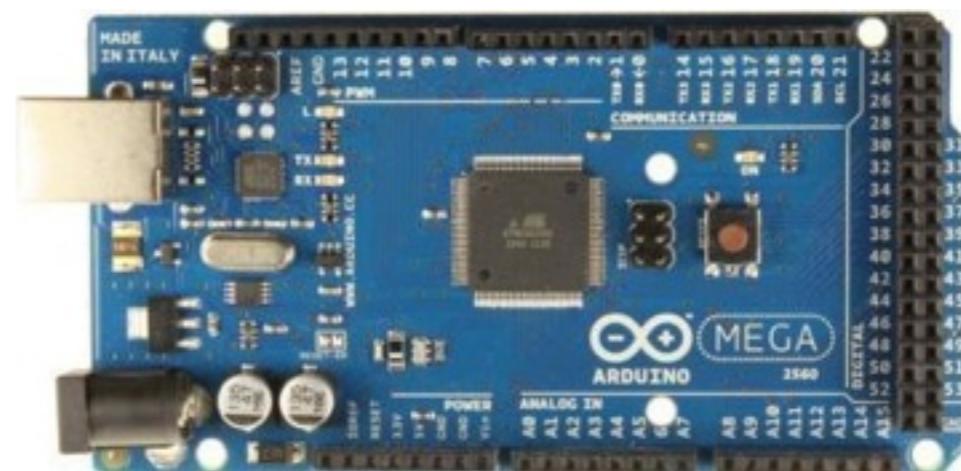
UNO



Leonardo

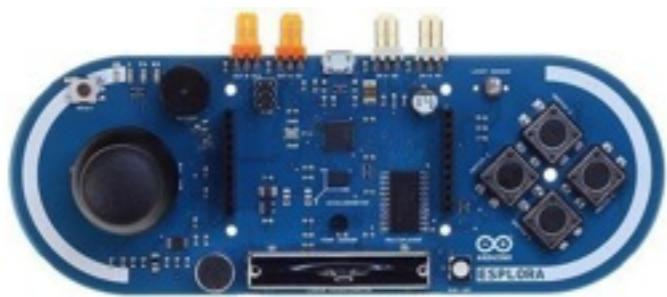


DUE

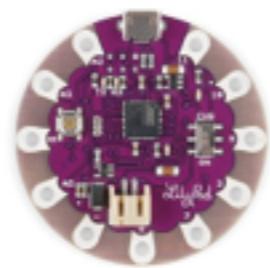


MEGA

More Family . . .



Esplora



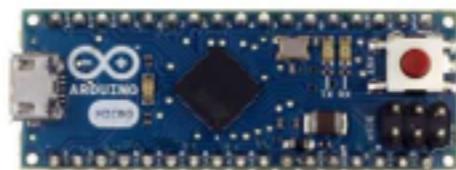
LilyPad



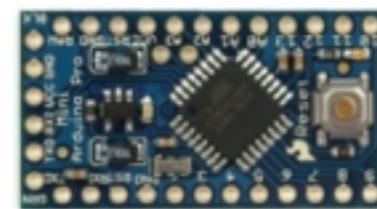
Pro



Arduino Ethernet



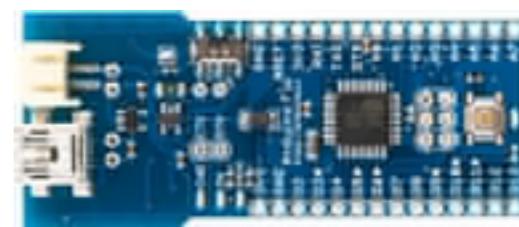
Micro



Pro Mini



Nano

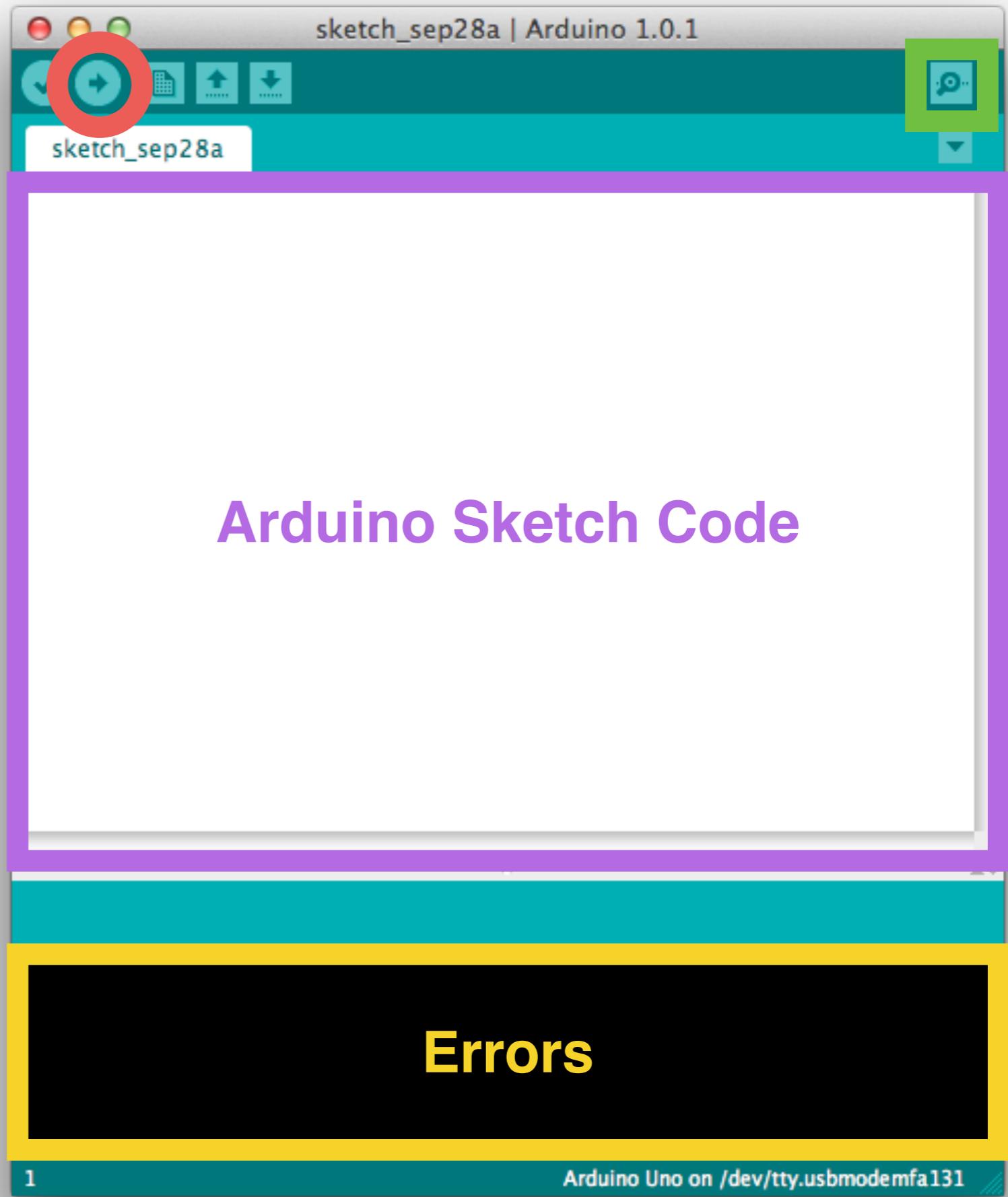


Fio

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

```
// Pin 13 has an LED connected on most Arduino boards.
```

```
// give it a name:
```

```
int led = 13;
```

Start Comment

```
// the setup routine runs once when you press reset:
```

```
void setup() {
```

```
  // initialize the digital pin as an output.
```

```
  pinMode(led, OUTPUT);
```

```
}
```

Line Comments

```
// 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 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.
*/



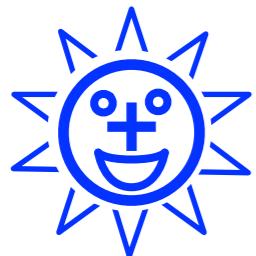
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

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

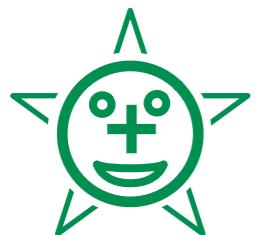
// the loop routine runs over and over again forever:
void loop() {
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);  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

Electronics is...

Moving Charge



Highly energetic charge particle

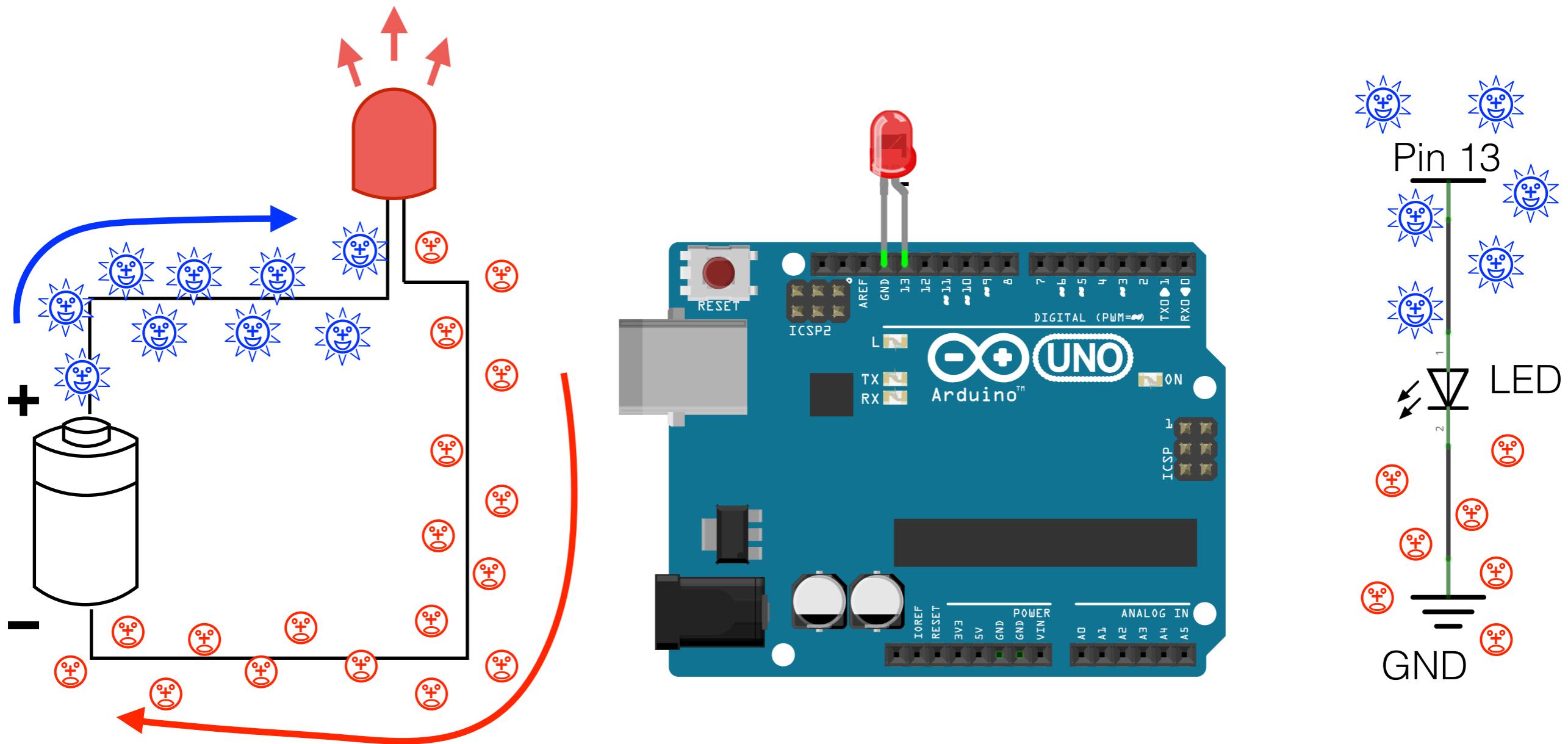


Less energetic charge particle

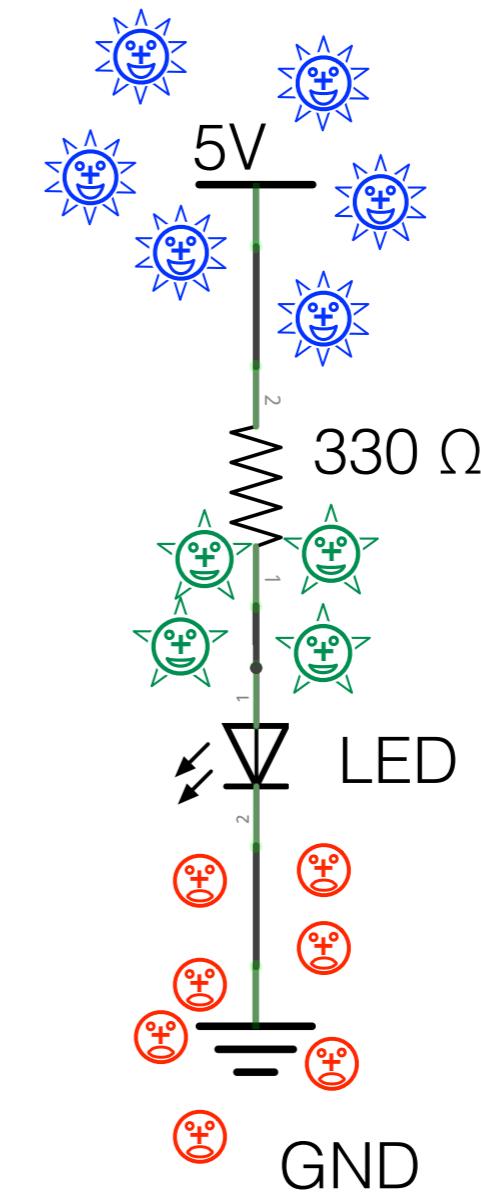
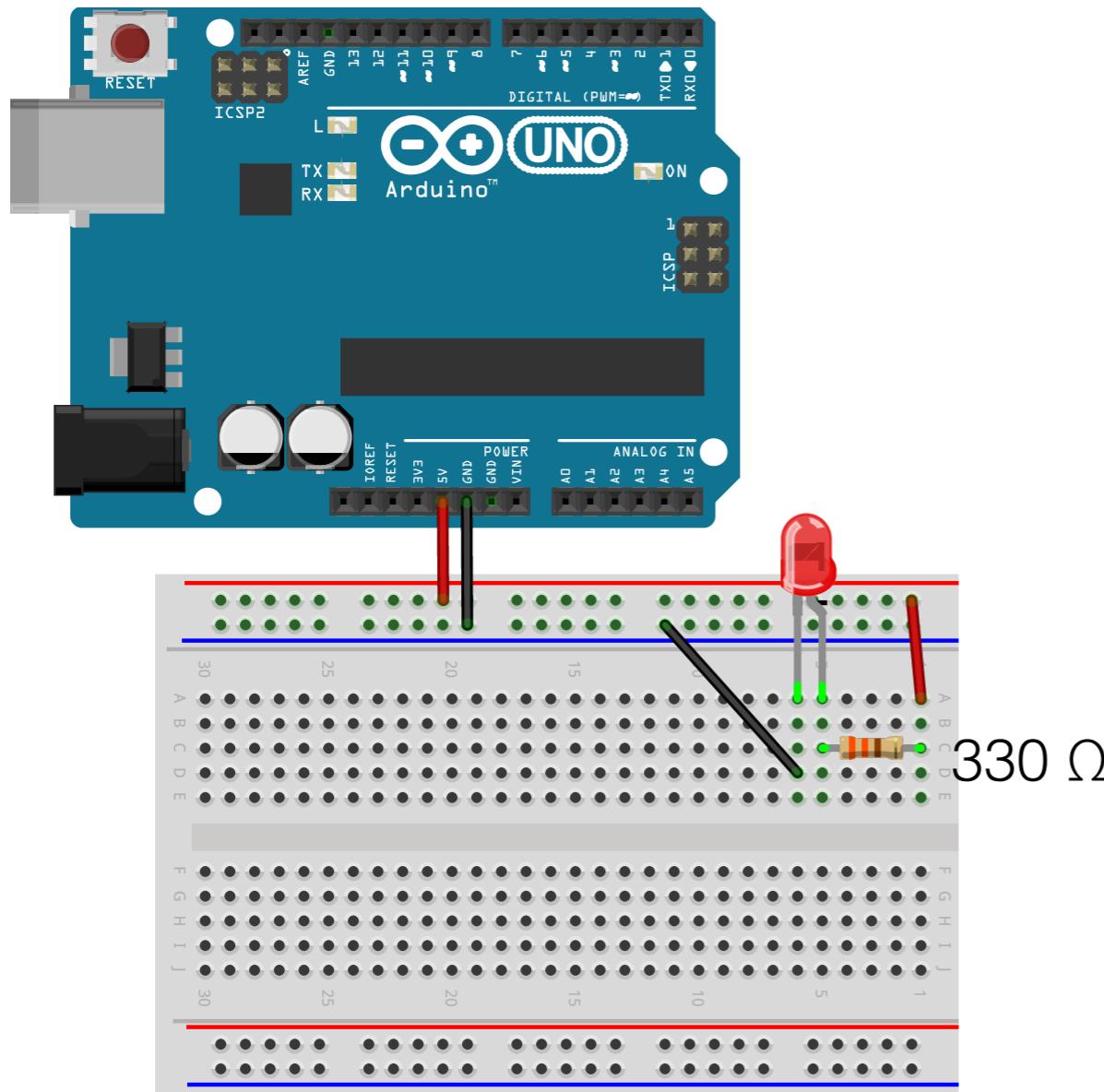


Exhausted charge particle

Simple Circuit

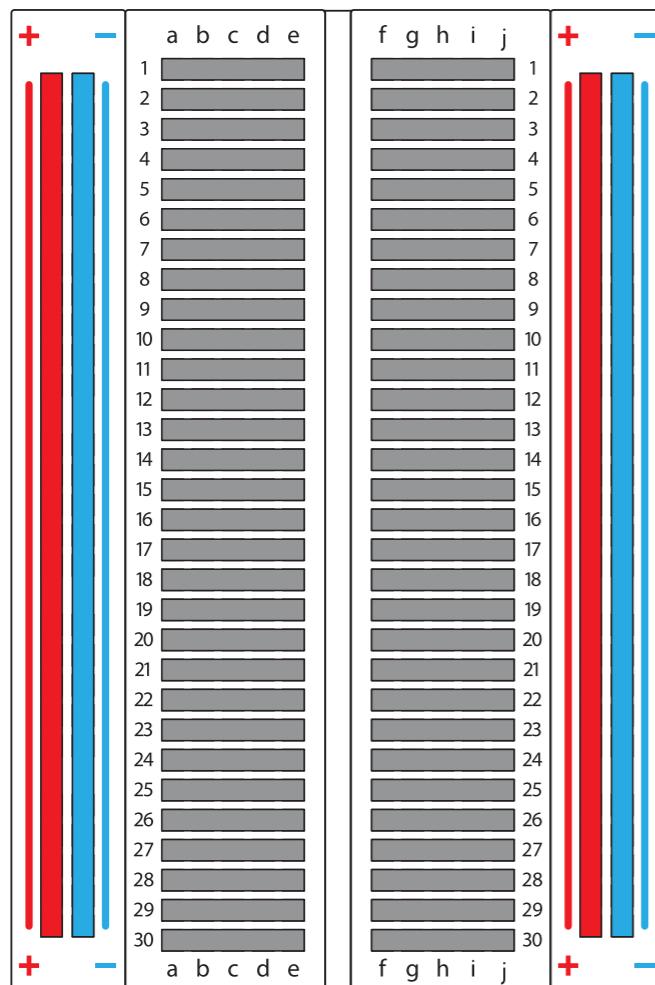


A Happier LED

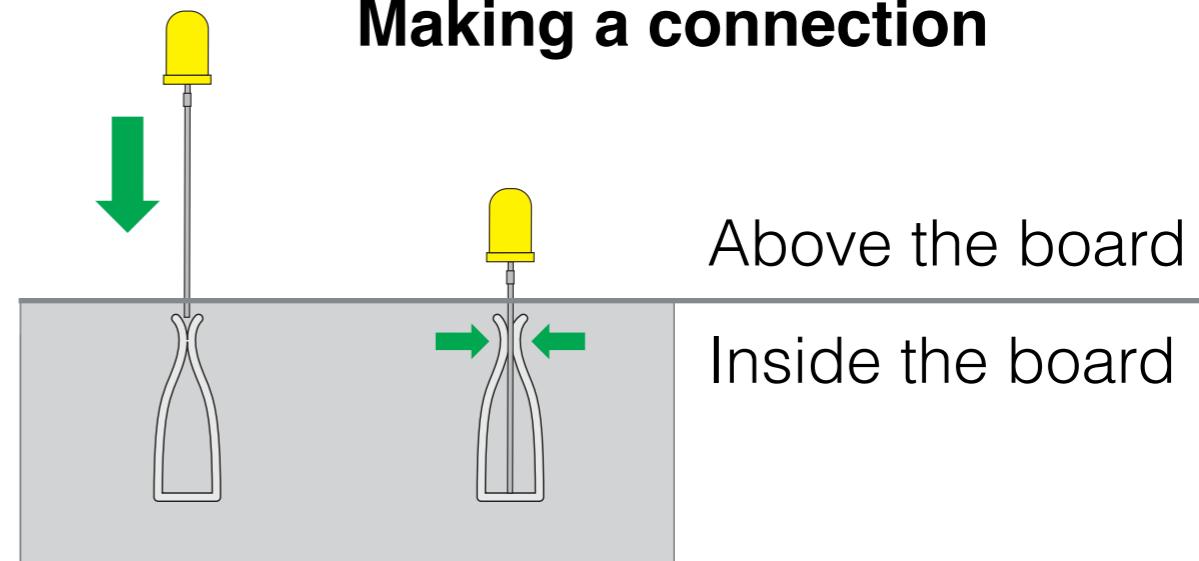


The Breadboard

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

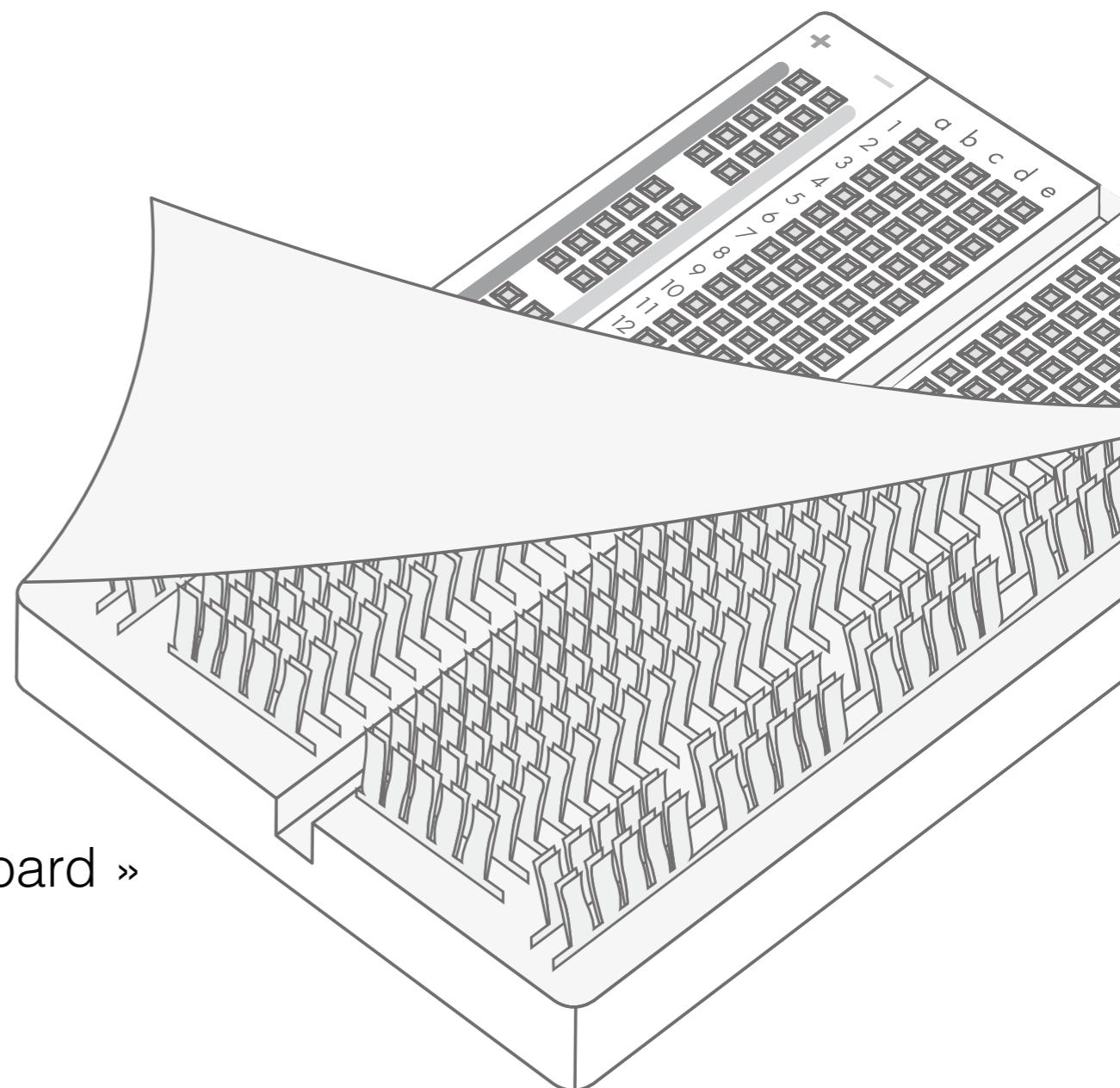


Making a connection

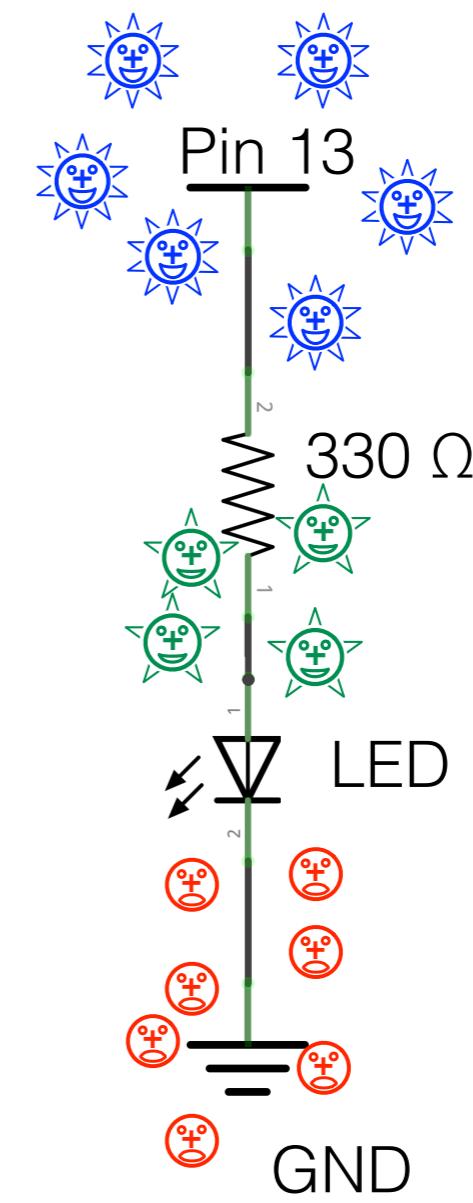
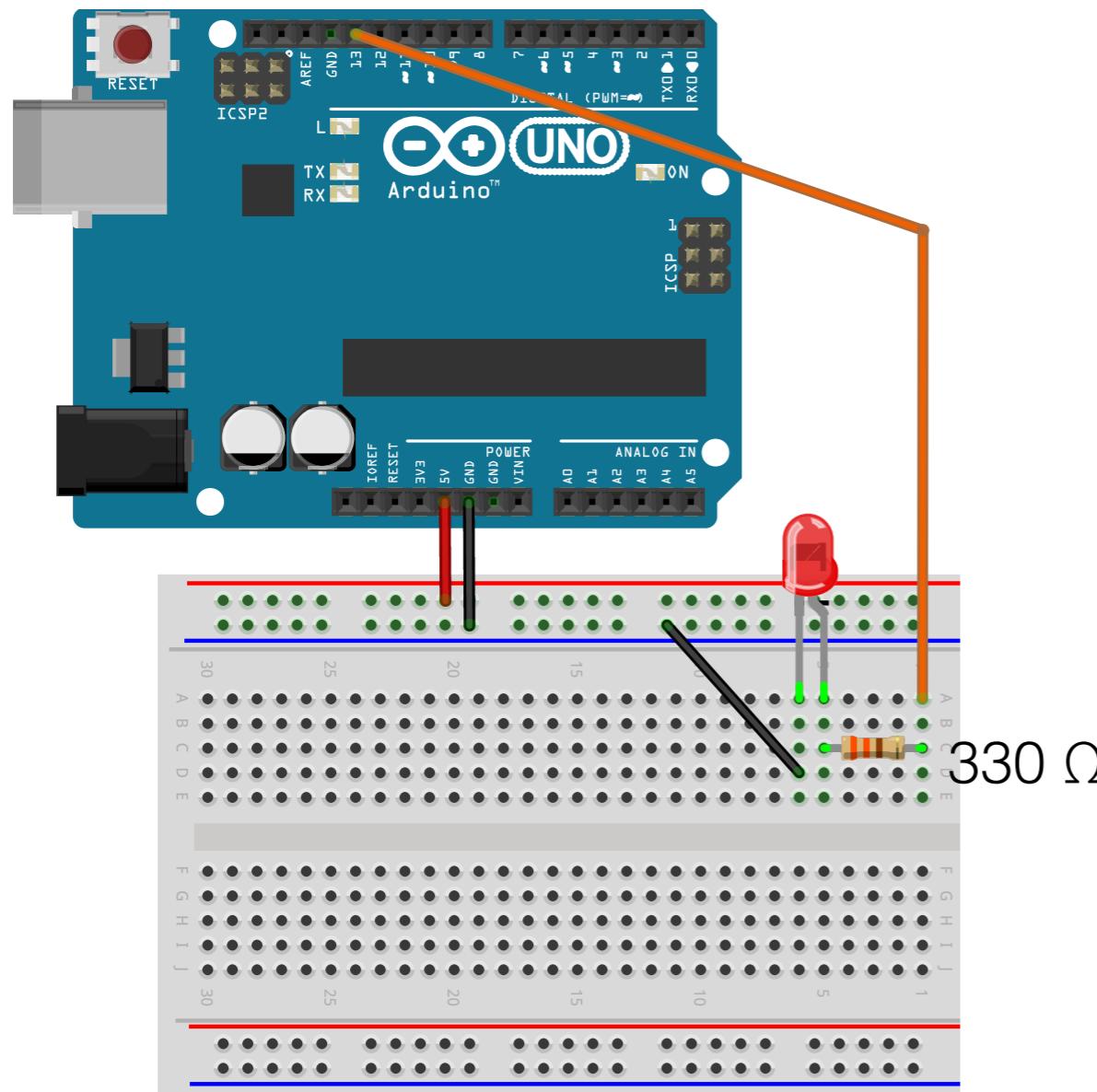


Above the board

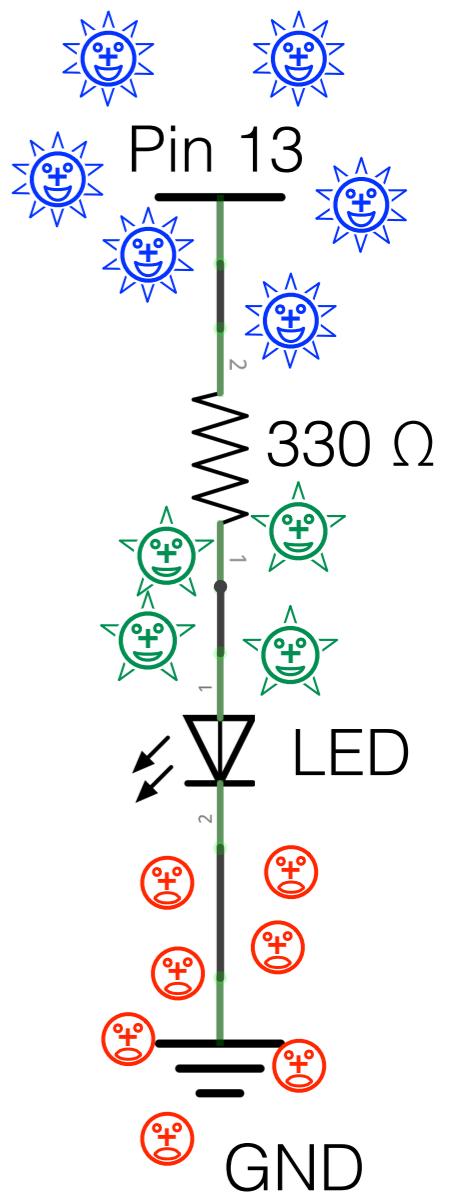
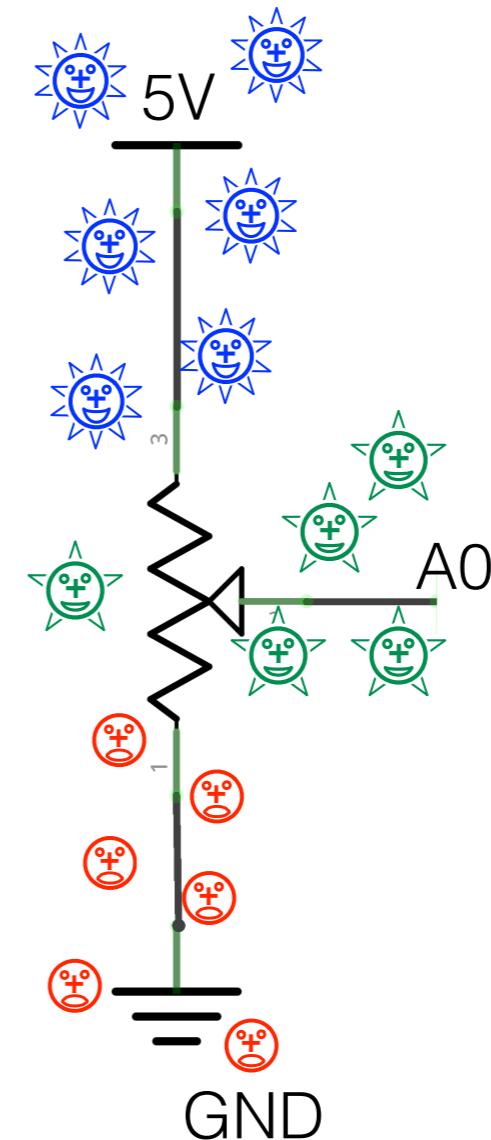
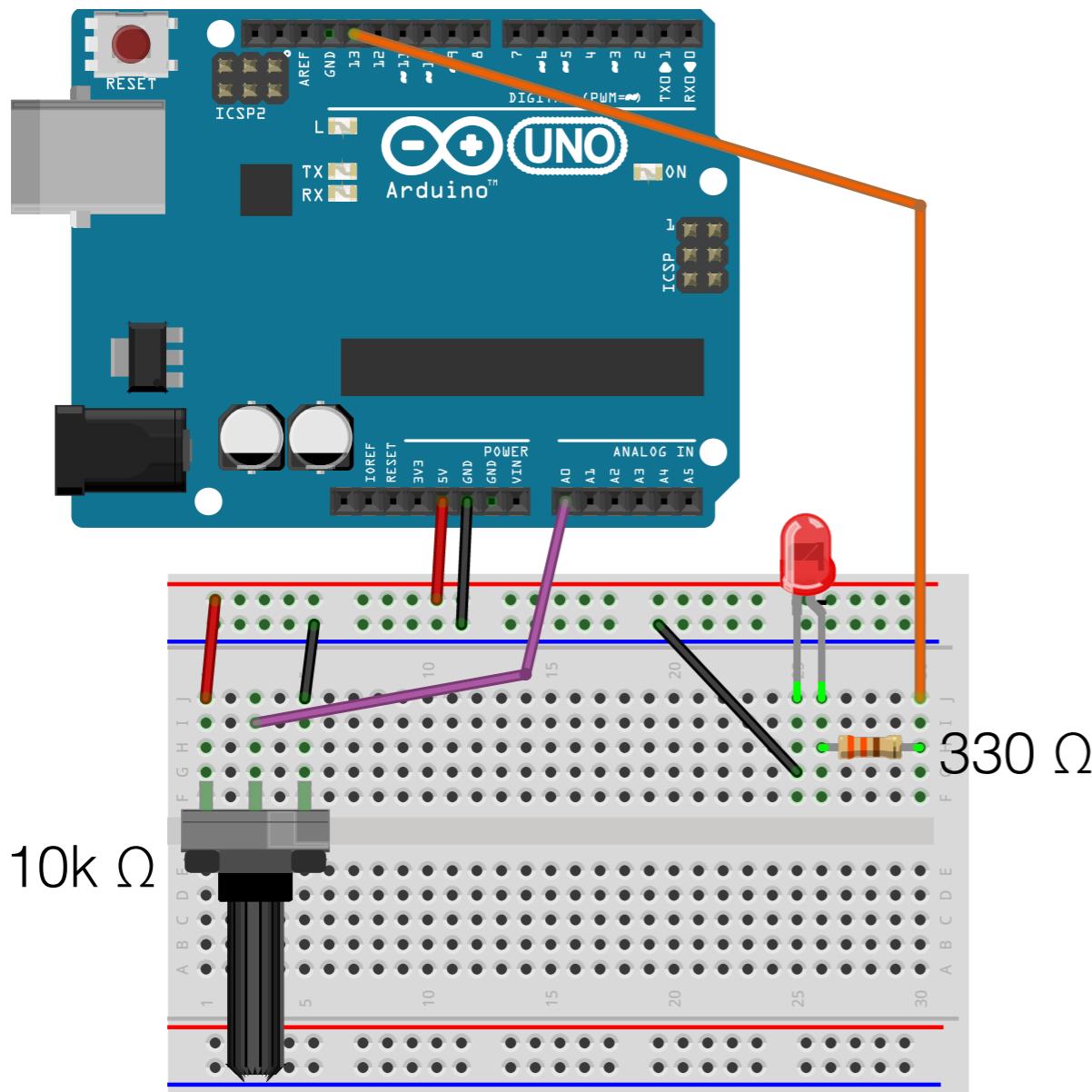
Inside the board



A Blinking LED



Control the Blinking



Control the Blinking, Code

```
/*
Analog Input
Demonstrates analog input by reading an analog sensor on analog pin 0 and
turning on and off a light emitting diode(LED) connected to digital pin 13.
The amount of time the LED will be on and off depends on
the value obtained by analogRead().

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);
}
```

Debug the Blinking

```
/*
Analog Input
Demonstrates analog input by reading an analog sensor on analog pin 0 and
turning on and off a light emitting diode(LED) connected to digital pin 13.
The amount of time the LED will be on and off depends on
the value obtained by analogRead().
```

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);
}
```

Challenges!

- Move the external LED to pin 12. Attach a **second** potentiometer, use its input to control pin 12's blink rate or brightness.
- Make one potentiometer control **brightness** instead of blink rate.
 - *Hint: try flashing the LED really quickly! The analogWrite function might help — look it up in the Arduino reference.*

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.