

# Arduino 101

## Day 1: Outputs

Presented by SCI and Area515

Instructors:

Bailey Mader

Brooke Hedrick

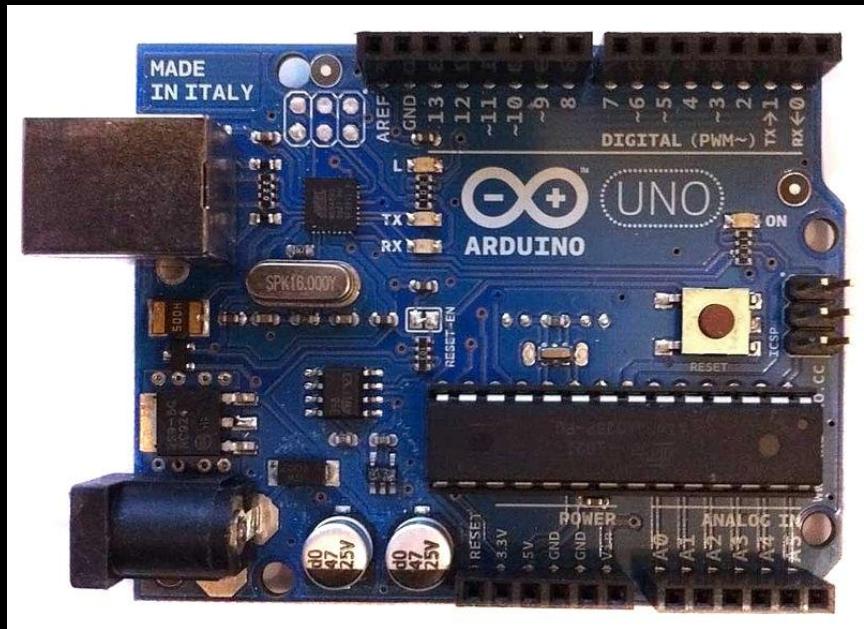
Brandon Hart

# Let's Get Set Up!

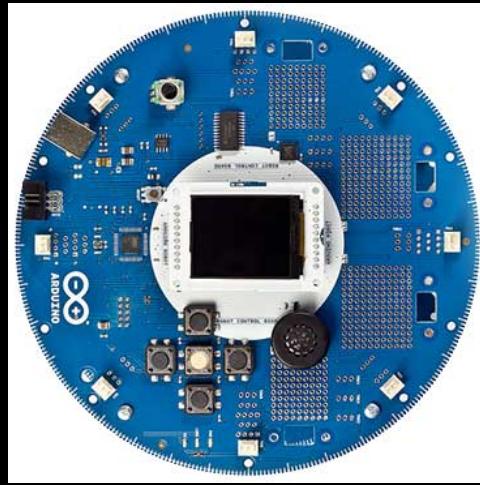
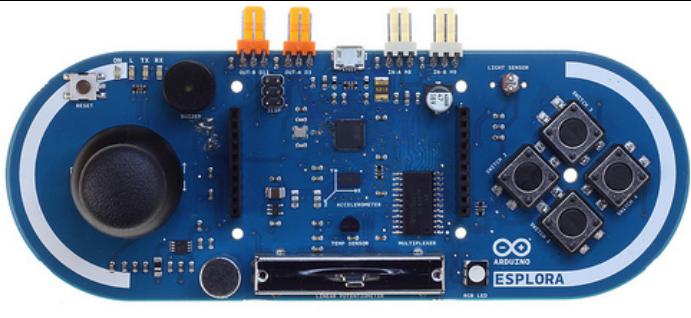
Download the Arduino software from [arduino.cc](http://arduino.cc)

Next, we'll look at what board everyone has.

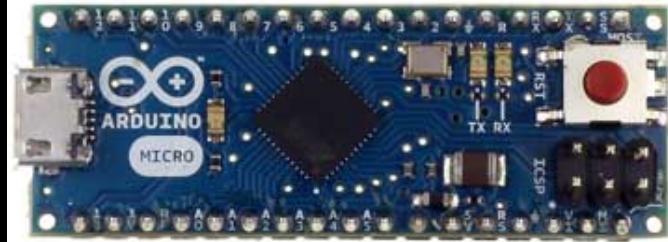
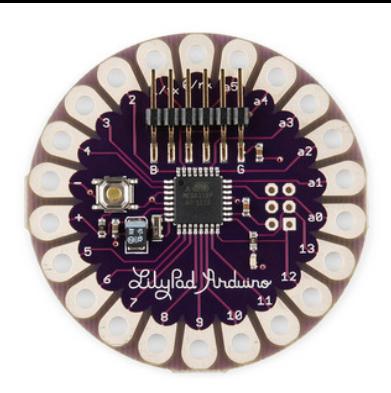
Picture from [shallowsky.com](http://shallowsky.com)



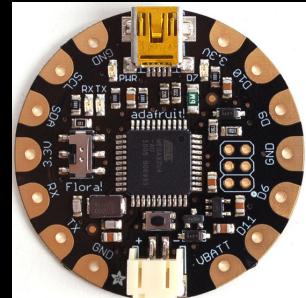
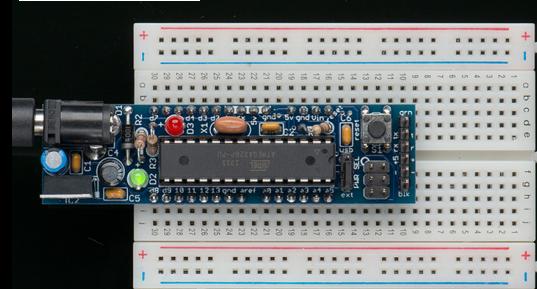
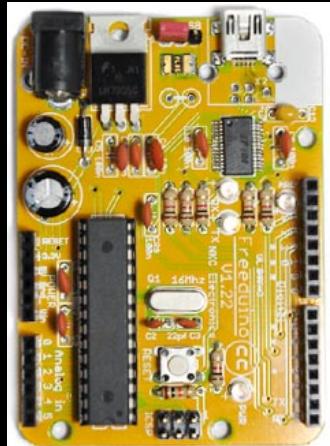
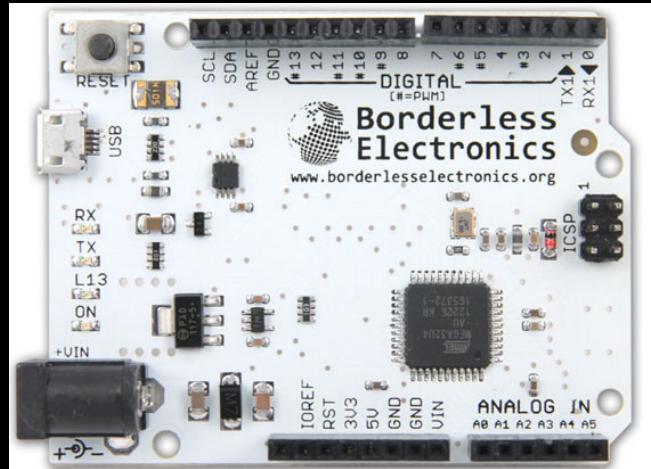
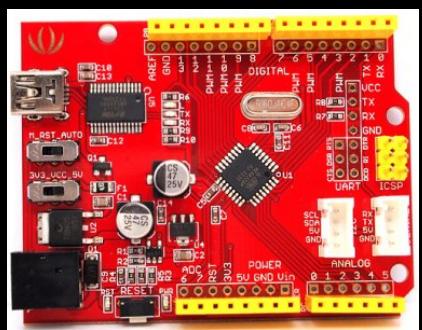
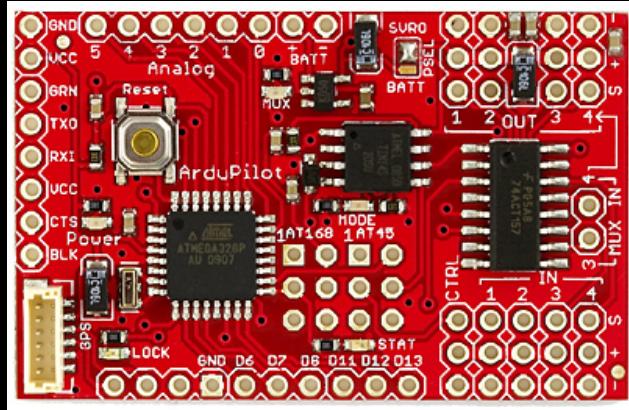
# More examples of types of boards



Pictures from arduino.cc

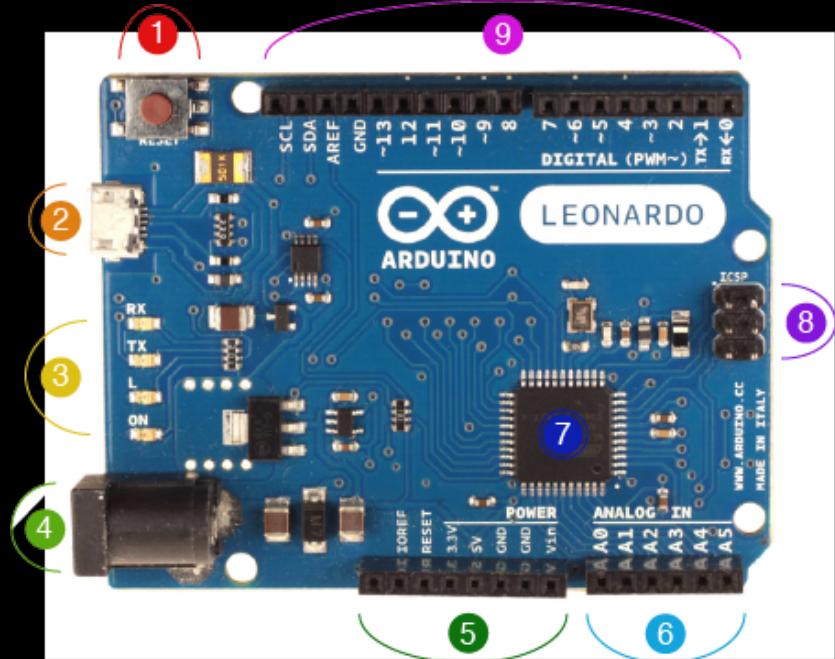


# Attack of the Clones



# Arduino Board

Each board will have the same basic set of components, though they may be arranged differently.



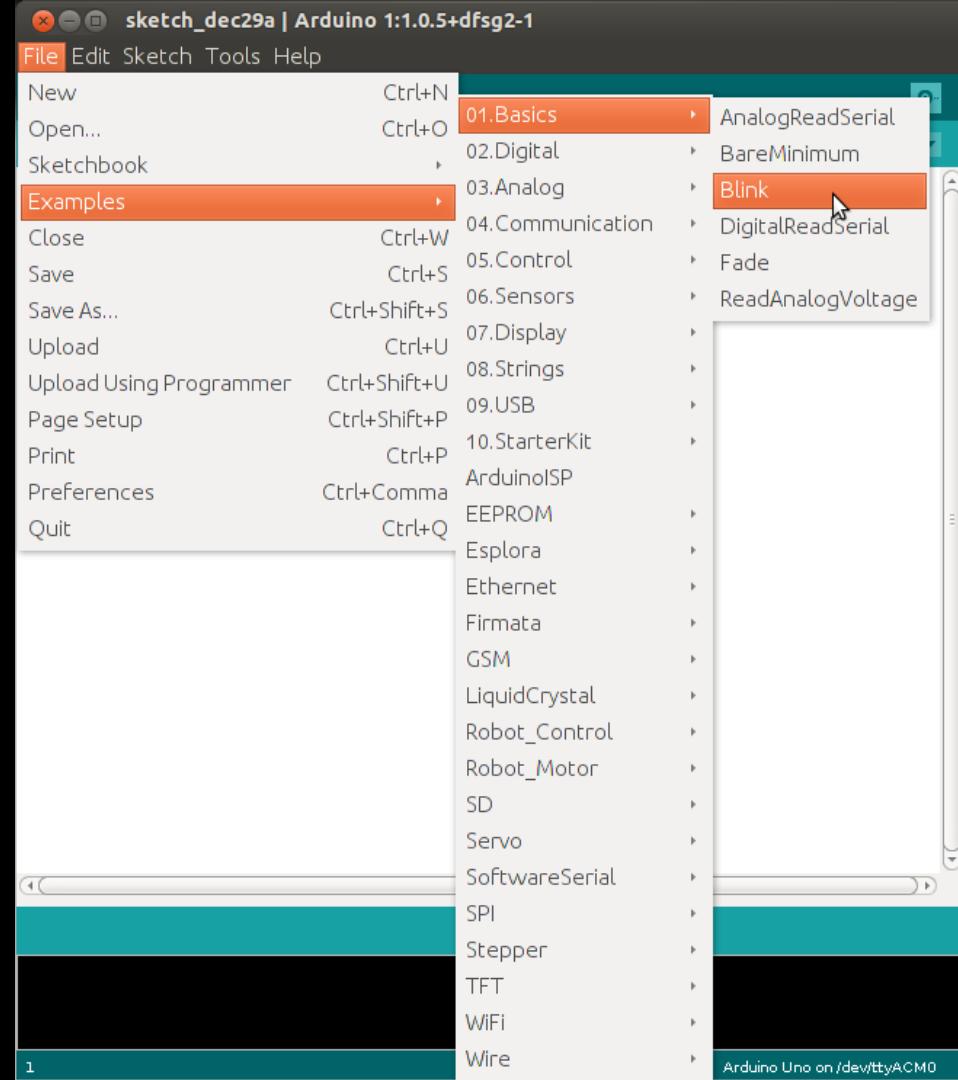
Color	Tech Name	What it does:
1	Reset Button	Restarts the Arduino
2	USB	Program your Arduino from your Computer
3	Indicator LEDs	Shows the user what the Arduino is doing
4	DC Power	Power your project - without USB
5	Power Pins	Connect power to your circuit
6	Analog Pins	Connect sensors to your Arduino
7	ATmega32u4	Brain of your Computer
8	ISP Connector	Update bootloader (software) on the Arduino
9	Digital I/O	Connect sensors or actuators to your Arduino

# Let's Blink!

```
//variable declarations
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);
    delay(1000);
    digitalWrite(led, LOW);
    delay(1000);
}
```



# Some notes about Arduino coding...

## Functions in **EVERY** Arduino sketch:

```
void setup( ){  
    //code goes here  
}  
void loop( ){  
    //code goes here  
}
```

## Setting pin mode:

- input: A pin mode that intakes information.
- output: A pin mode that sends information.

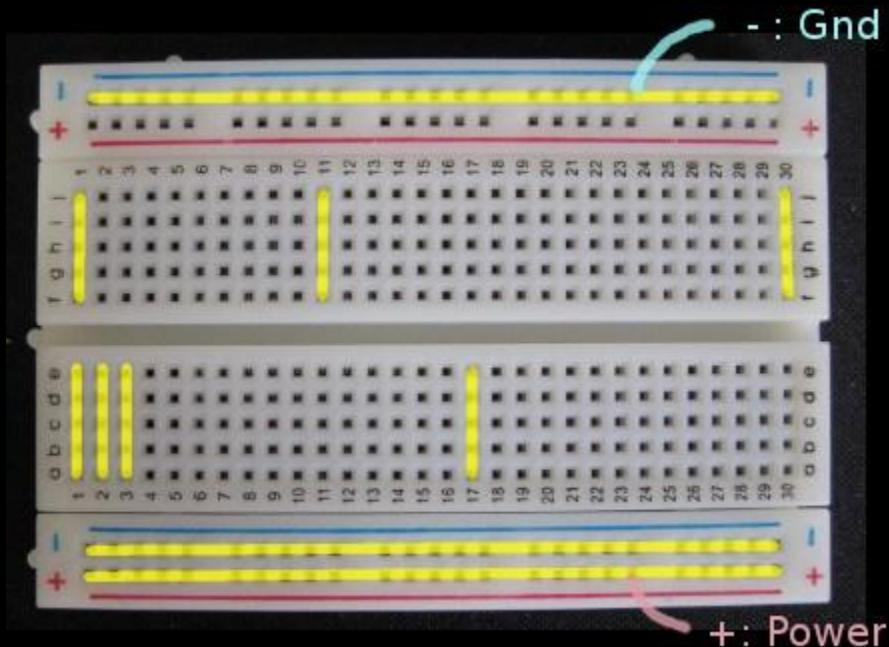
## Setting pin state:

- HIGH: Electrical signal present (5V or 3.3V for Uno). Also ON or TRUE in boolean logic.
- LOW: No electrical signal present (0V). Also OFF or FALSE in boolean logic.

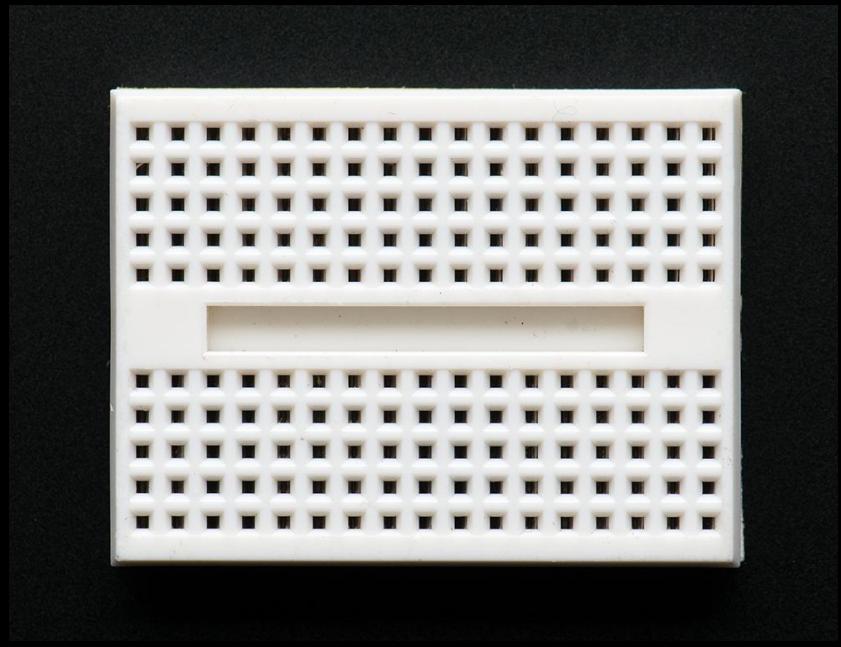
## The main four I/O functions:

- digitalRead: Get a HIGH or LOW reading from a pin already declared as an input.
- digitalWrite: Assign a HIGH or LOW value to a pin already declared as an output.
- analogRead: Get a value between or including 0 (LOW) and 1023 (HIGH). This allows you to get readings from analog sensors or interfaces that have more than two states.
- analogWrite: Assign a value between or including 0 (LOW) and 255 (HIGH). This allows you to set output to a PWM value instead of just HIGH or LOW.

# Breadboards



Picture from [shallowsky.com](http://shallowsky.com)



Picture from [adafruit.com](http://adafruit.com)

# Let's review a bit about electricity!

 Diode

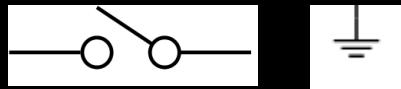
 Capacitor

 Inductor

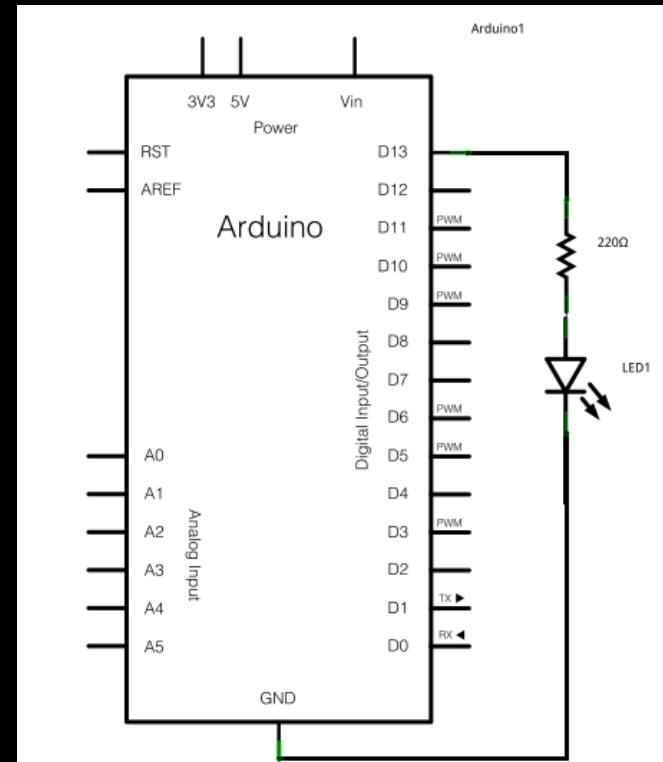
 Resistor

 DC voltage source

 AC voltage source



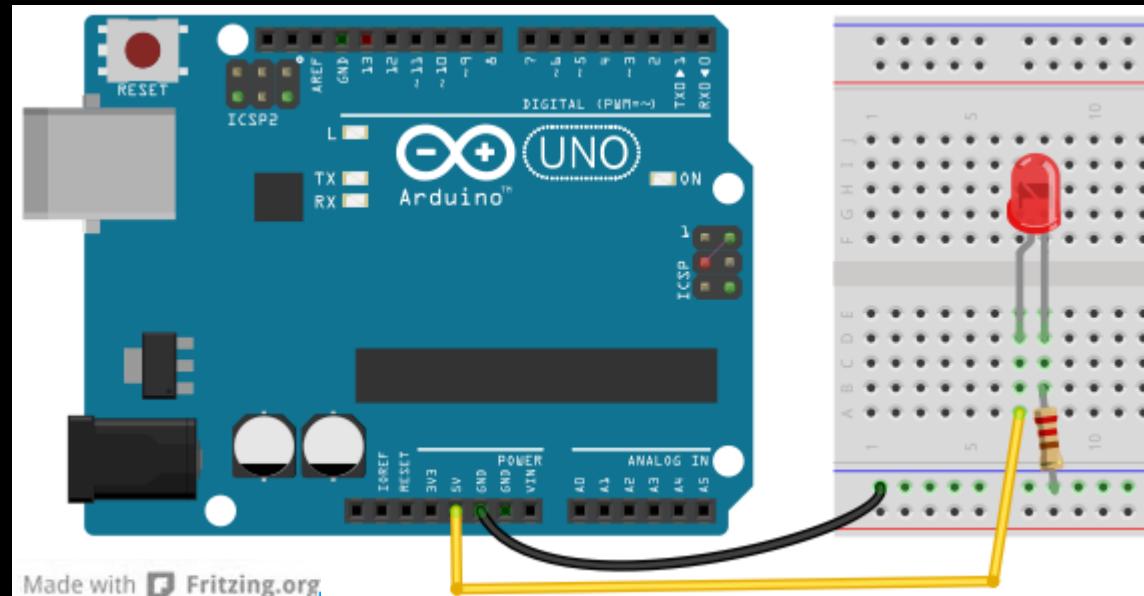
$$V = IR$$

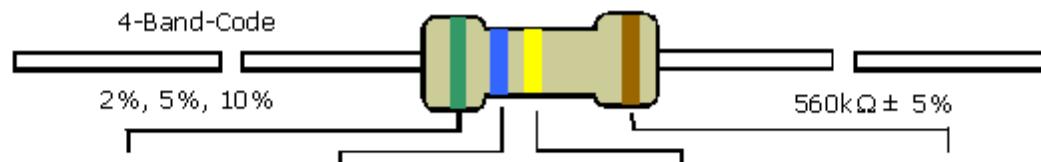


# LEDs and Resistors

LED = light emitting  
diode

LEDs are  
directional,  
resistors are not.

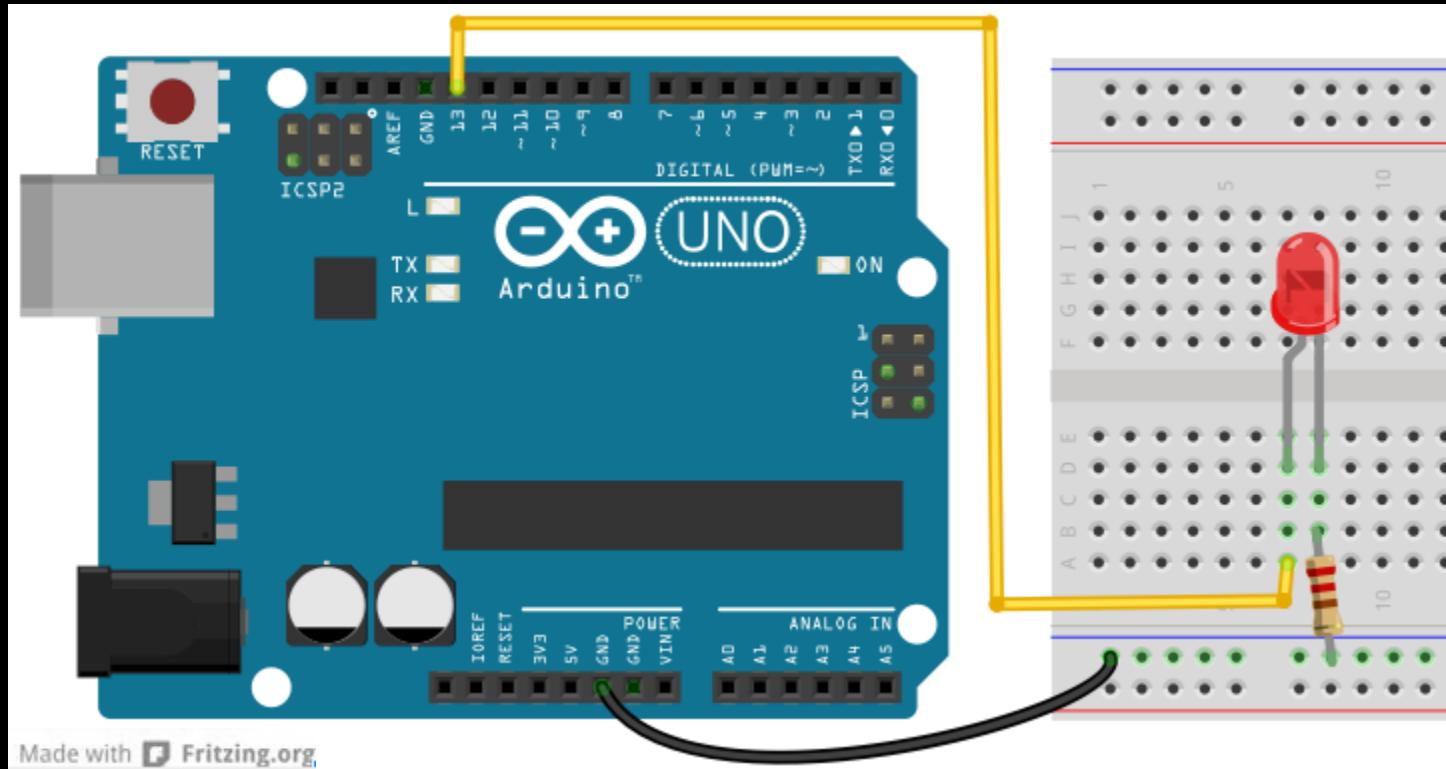




COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE	
Black	0	0	0	$1\Omega$		
Brown	1	1	1	$10\Omega$	$\pm 1\%$	(F)
Red	2	2	2	$100\Omega$	$\pm 2\%$	(G)
Orange	3	3	3	$1\text{K}\Omega$		
Yellow	4	4	4	$10\text{K}\Omega$		
Green	5	5	5	$100\text{K}\Omega$	$\pm 0.5\%$	(D)
Blue	6	6	6	$1\text{M}\Omega$	$\pm 0.25\%$	(C)
Violet	7	7	7	$10\text{M}\Omega$	$\pm 0.10\%$	(B)
Grey	8	8	8		$\pm 0.05\%$	
White	9	9	9			
Gold				0.1	$\pm 5\%$	(J)
Silver				0.01	$\pm 10\%$	(K)

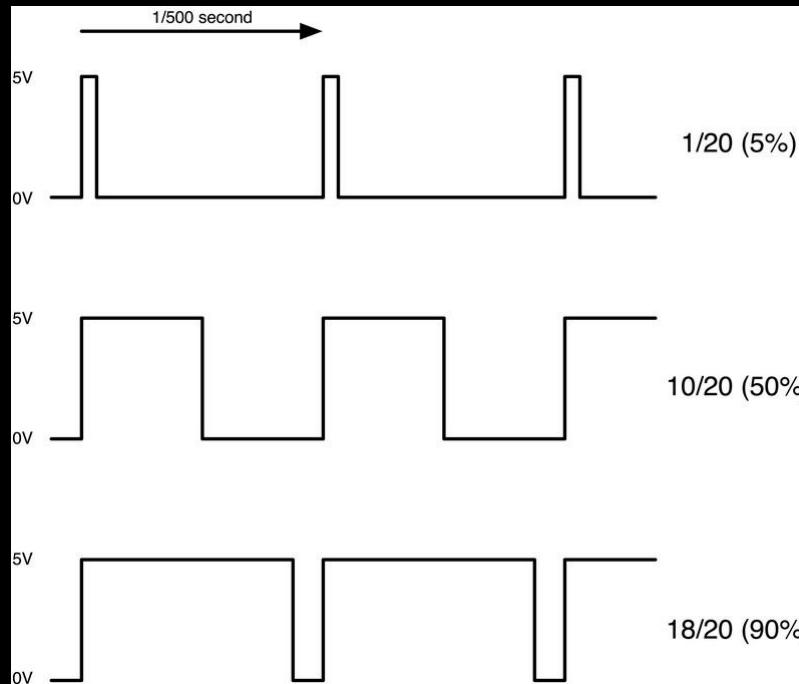


# Blink Hardware Setup



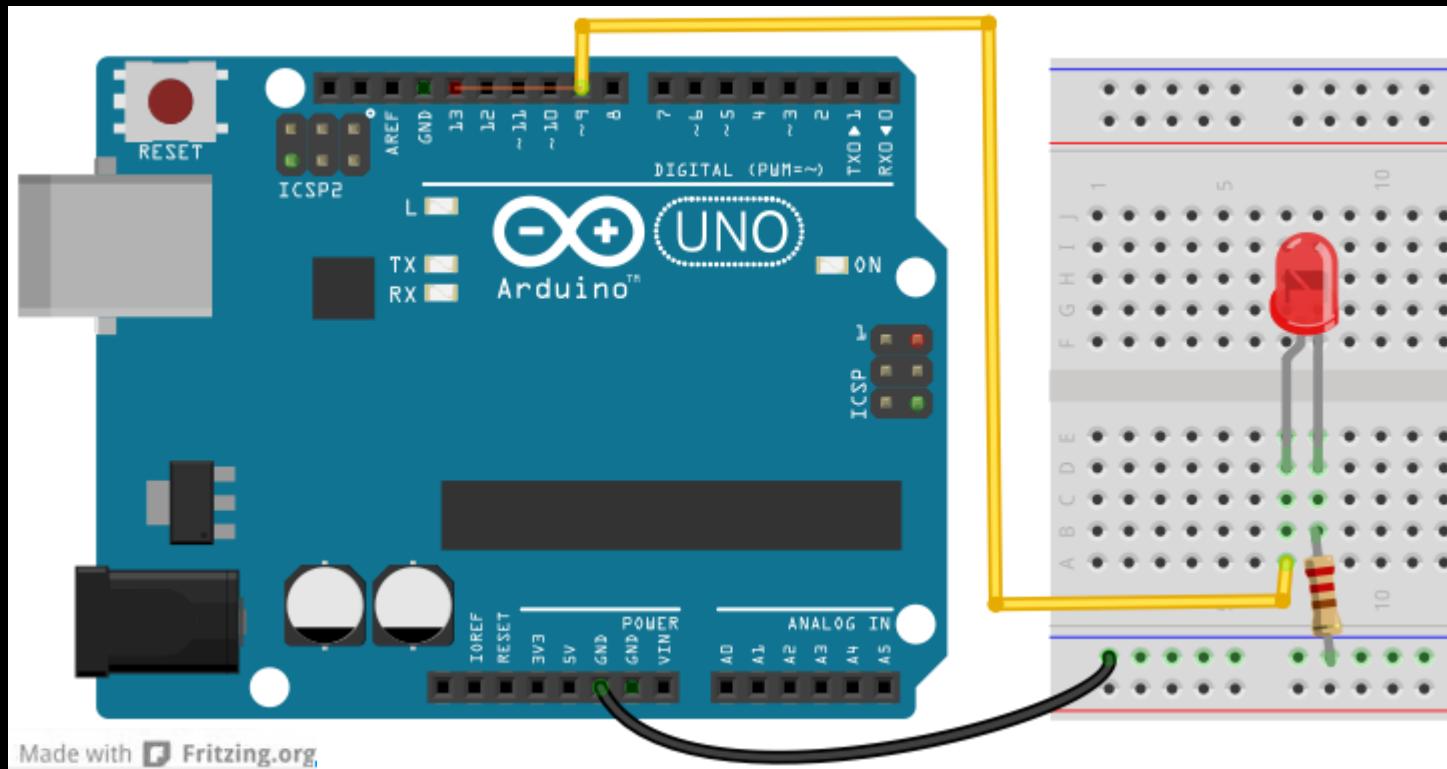
Made with Fritzing.org

# Pulse Width Modulation - PWM



Picture from [learn.adafruit.com](http://learn.adafruit.com)

# Fade Hardware Setup



# Let's Fade

```
int led = 9;      // the pin that the LED is attached to

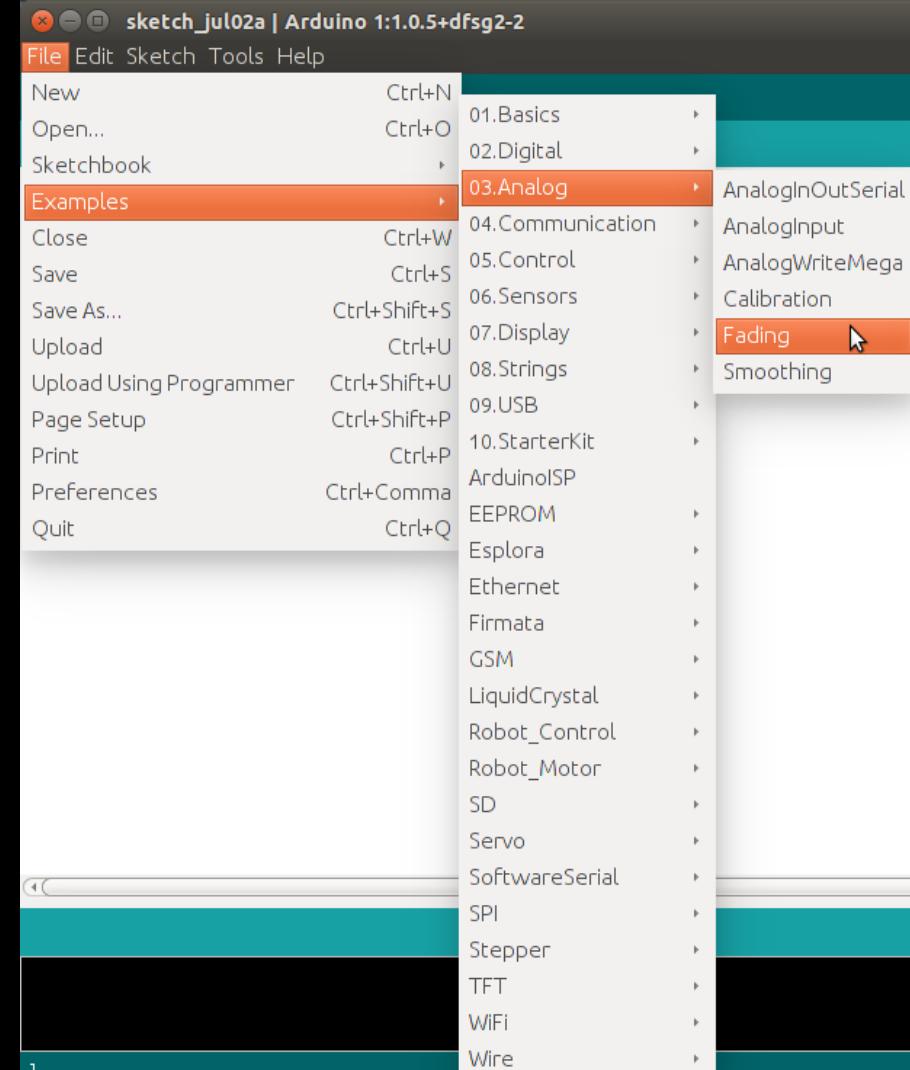
int brightness = 0; // how bright the LED is
int fadeAmount = 5; // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of the LED on pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```



# Fade variables...

	A	B
1	brightness	<u>fadeAmount</u>
2	0	5
3	5	5
4	10	5
5	15	5
6	20	5
52	...	...
53	250	5
54	255	-5
55	250	-5
56	245	-5
103	...	...
104	10	-5
105	5	-5
106	0	5
107	5	5
108	10	5
109	...	...

# Let's Make Some Noise!

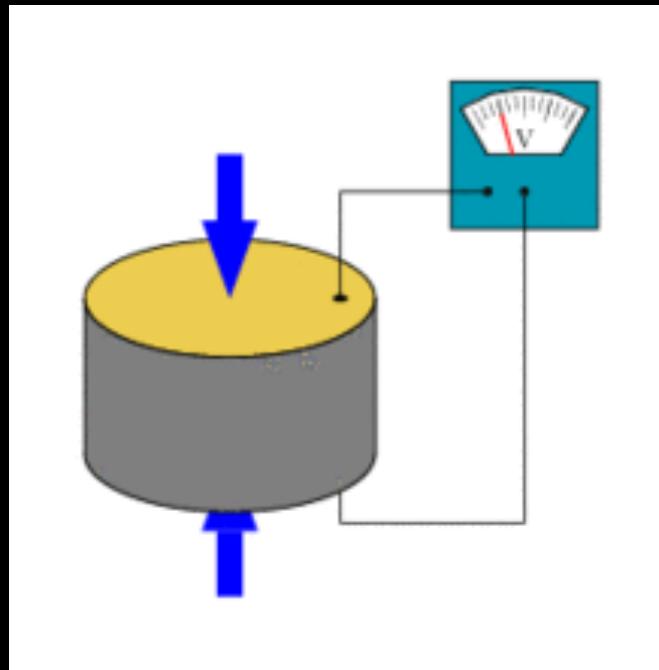
## Piezoelectricity

- \* electricity generated from mechanical stress
- \* process works both ways!



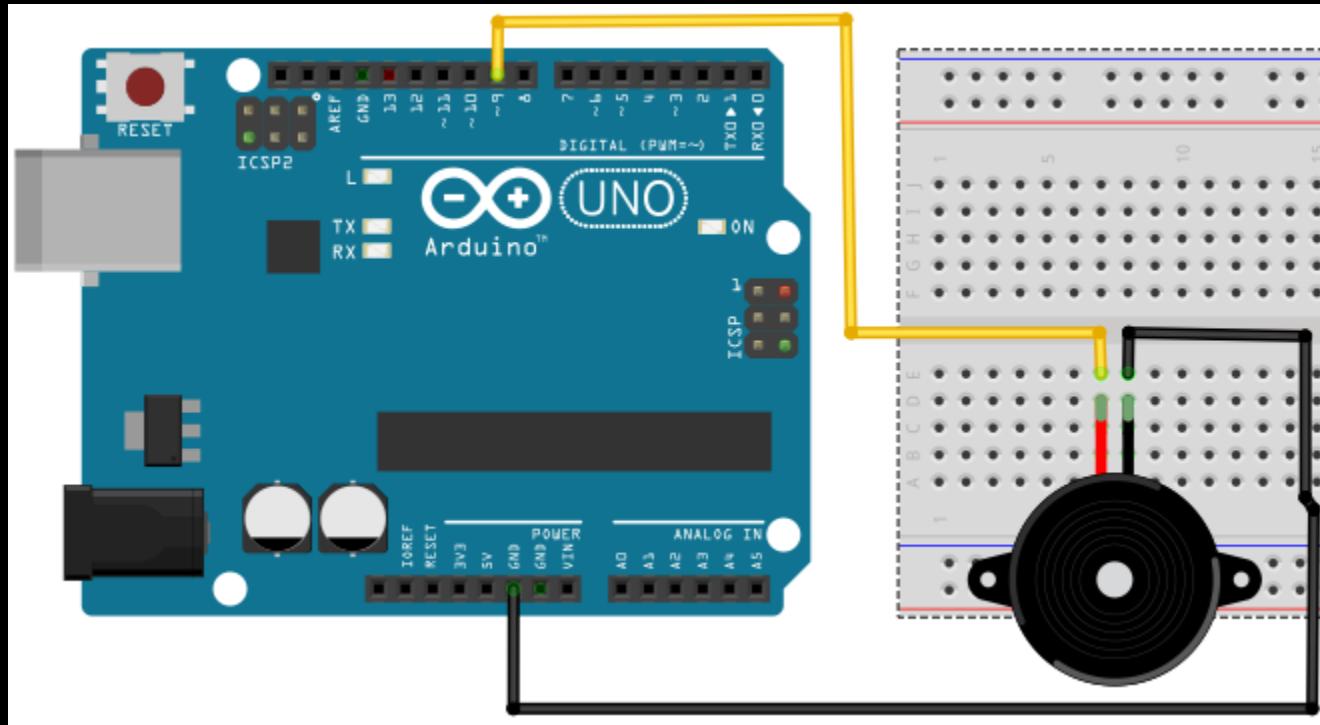
Picture from [solarbotics.com](http://solarbotics.com)

# How does a piezo work?



<http://en.wikipedia.org/wiki/Piezoelectricity>

# Try your Fade sketch with the Piezo



<http://www.arduino.cc/en/Tutorial/PlayMelody>

# Play some notes!

```
int speakerPin = 9;

int length = 15; // the number of notes
char notes[] = "ccggaagffeeddc "; // a space represents a rest
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;

void playTone(int tone, int duration) {
    for (long i = 0; i < duration * 1000L; i += tone * 2) {
        digitalWrite(speakerPin, HIGH);
        delayMicroseconds(tone);
        digitalWrite(speakerPin, LOW);
        delayMicroseconds(tone);
    }
}

void playNote(char note, int duration) {
    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

    // play the tone corresponding to the note name
    for (int i = 0; i < 8; i++) {
        if (names[i] == note) {
            playTone(tones[i], duration);
        }
    }
}
```

```
void setup() {
    pinMode(speakerPin, OUTPUT);
}

void loop() {
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // rest
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        // pause between notes
        delay(tempo / 2);
    }
}
```

# Further Explorations With Outputs

## Some Ideas:

- \* Find out how to play songs with your piezo speaker
- \* Change how the LEDs fade
- \* Have your LEDs flash out messages in morse code or patterns
- \* Have your program generate a random number in setup().

```
random(10); //number will be between 0 and 9 (10 numbers  
total)
```

Make the output do different things based on the number

What else did people bring in addition to LEDs and speakers?