Lights! Speed! Action!

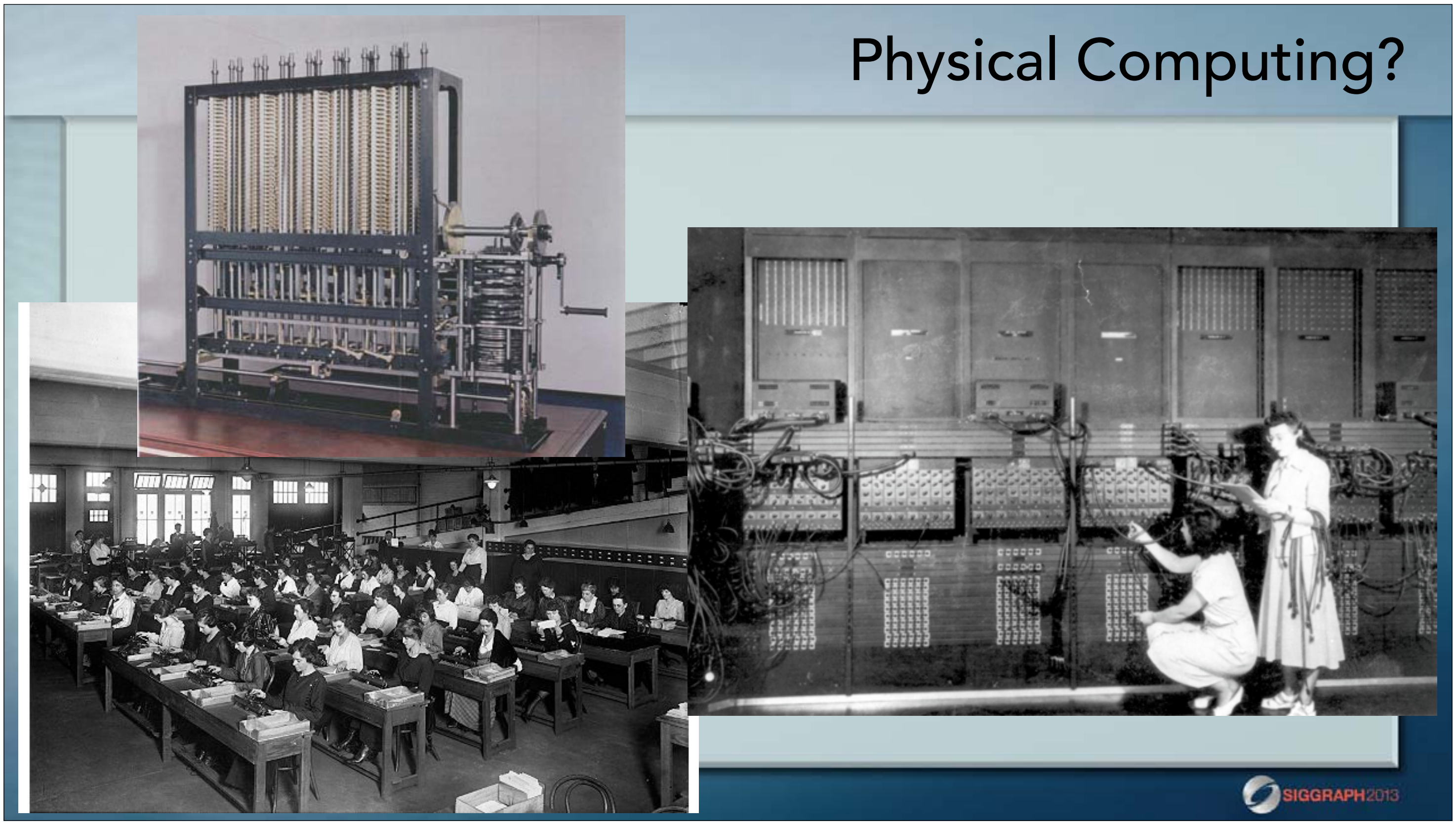
Fundamentals of Physical Computing for Programmers

SIGGRAPH2013

Erik Brunvand
School of Computing
University of Utah

THE UNIVERSITY OF UTAH

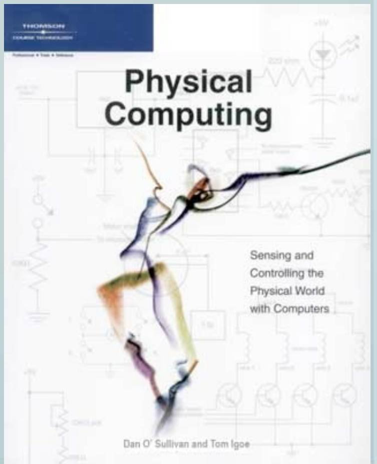
Physical Computing?



SIGGRAPH2013

Physical Computing

- Dan O'Sullivan and Tom Igoe's book (2004) has a title that nicely captures the idea
  - *Physical Computing: Sensing and Controlling the Physical World with Computers*

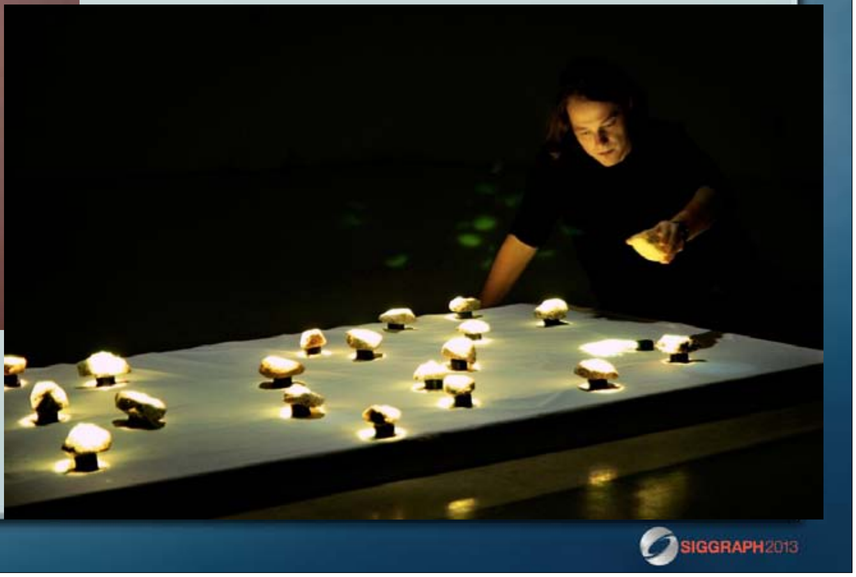

THOMSON
COURSE TECHNOLOGY
Physical Computing
Sensing and Controlling the Physical World with Computers
Dan O' Sullivan and Tom Igoe

Physical Computing

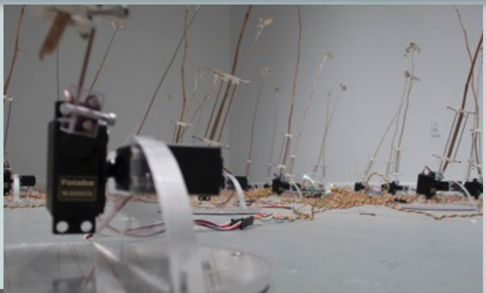
- Why should you care? You're into computer graphics!
  - But computer graphics can be about a LOT more than putting images on the screen…

The Bay Lights, Leo Villareal, 2013

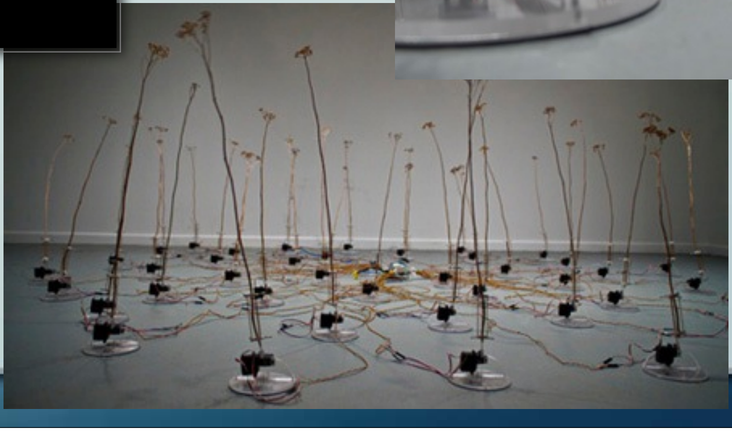
SIGGRAPH Art Gallery

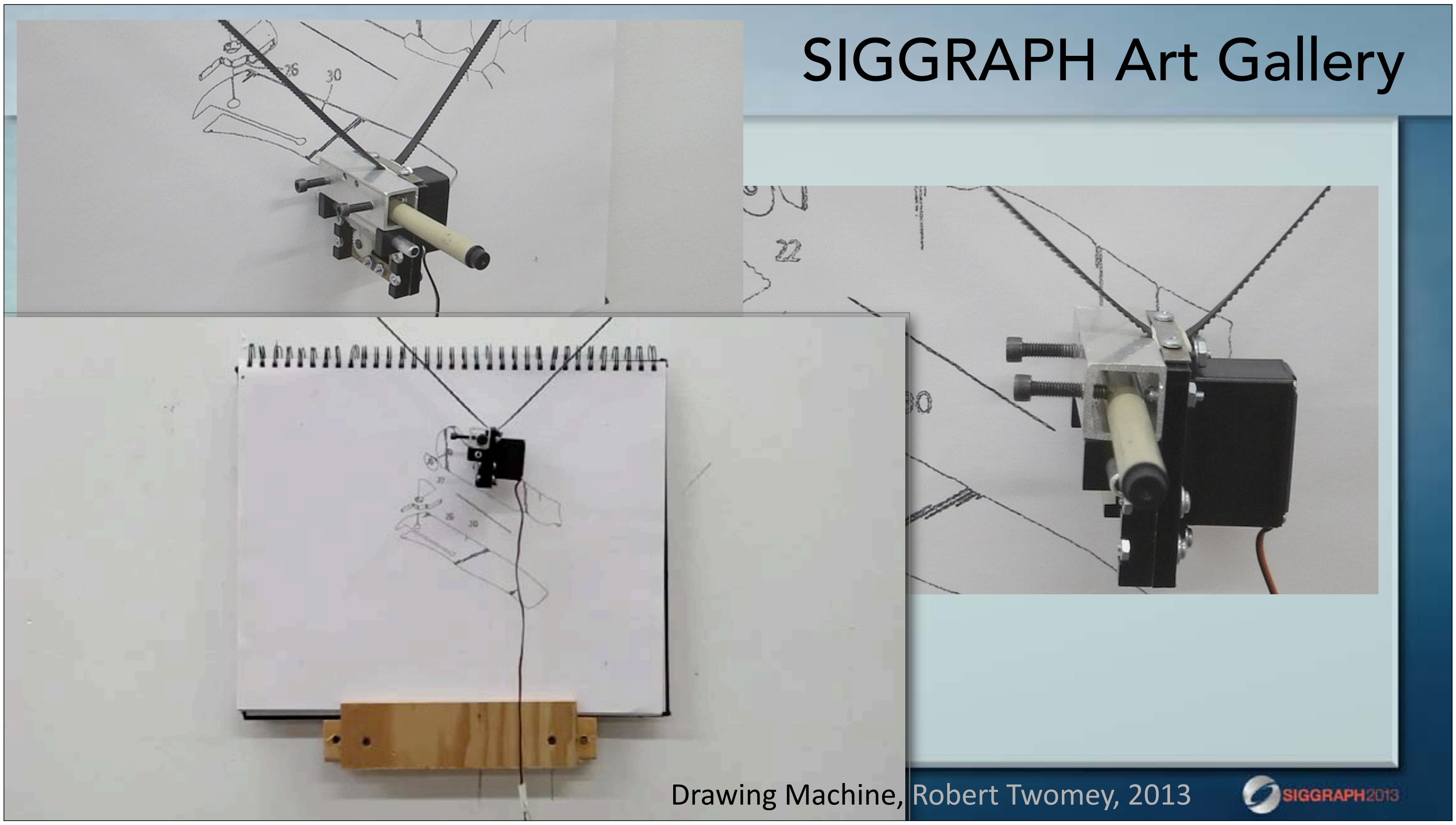


SIGGRAPH Art Gallery



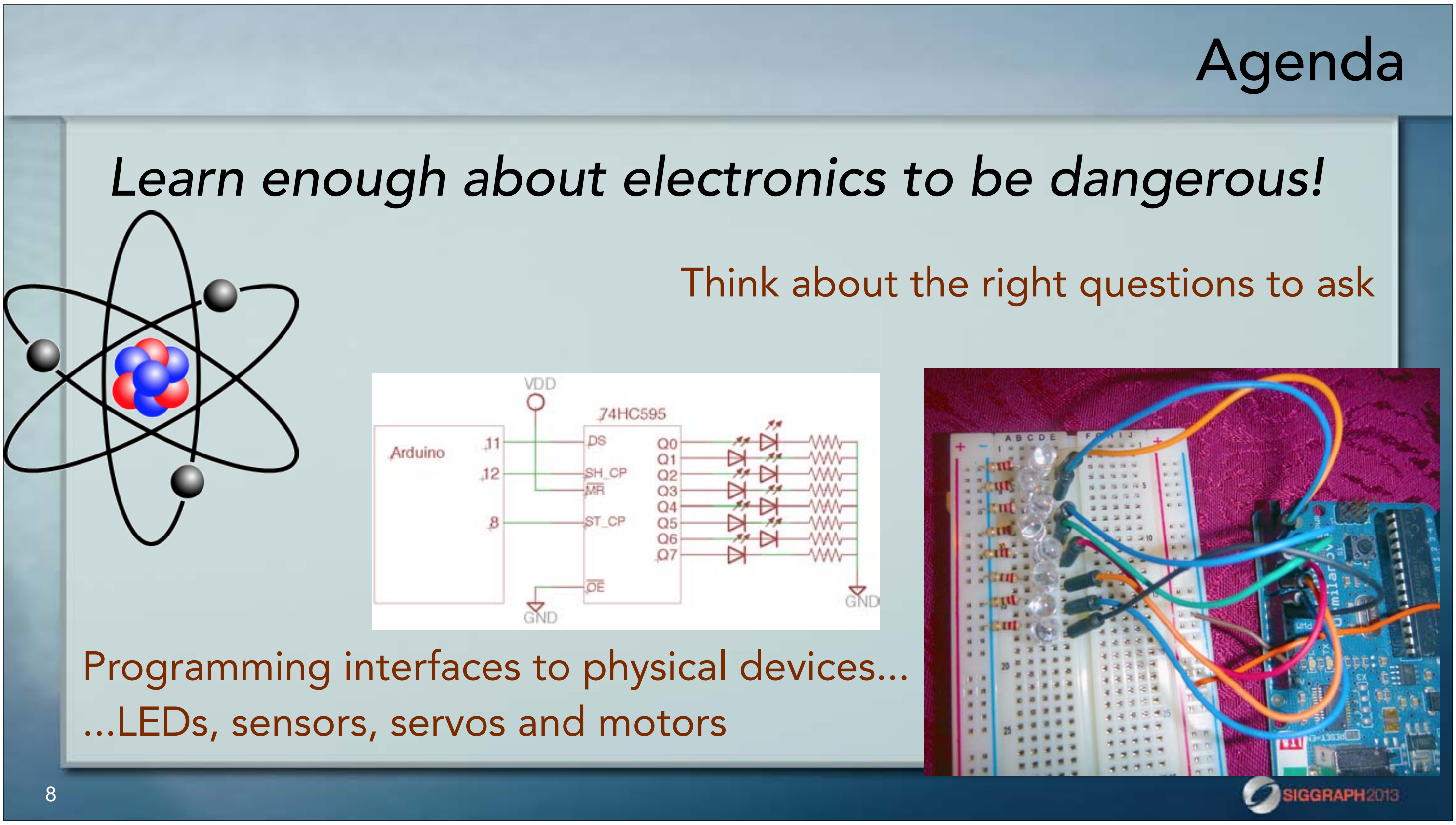
Telepresent Wind,
David Bowen,
SIGGRAPH 2010





SIGGRAPH Art Gallery

Drawing Machine, Robert Twomey, 2013



Agenda

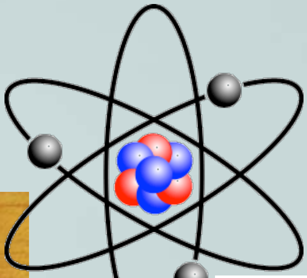
Learn enough about electronics to be dangerous!

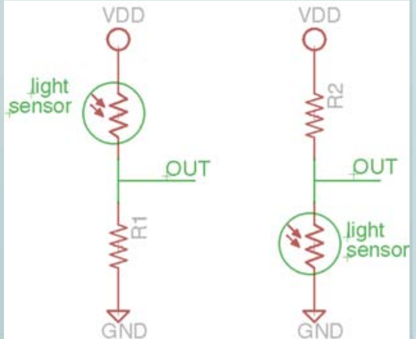
Think about the right questions to ask

Programming interfaces to physical devices…
…LEDs, sensors, servos and motors

More Detailed Agenda

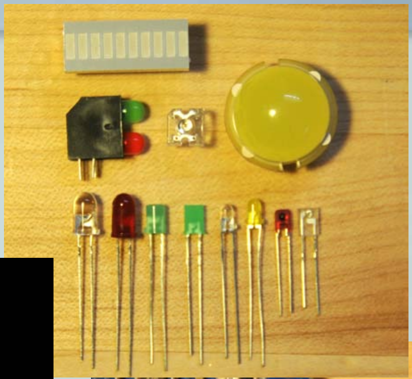
- Electronics Fundamentals

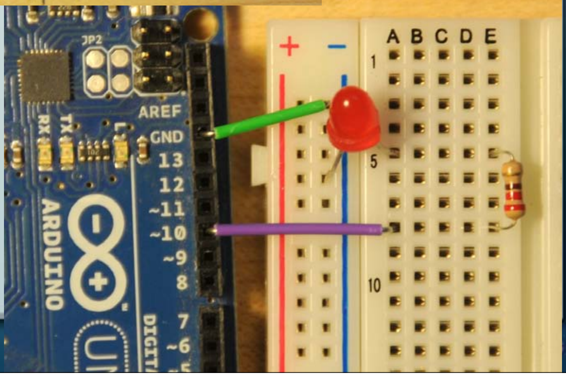




More Detailed Agenda

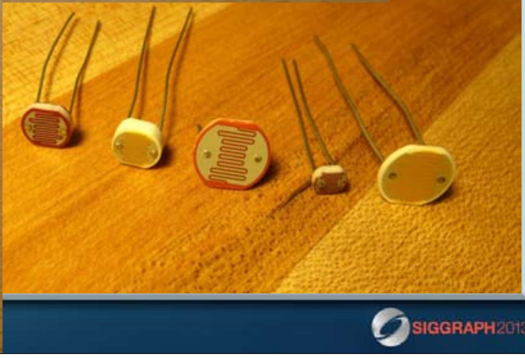
- Electronics Fundamentals
  - *Lights!* (LEDs)





More Detailed Agenda

- Electronics Fundamentals
  - *Lights!*
  - ***Speed!*** (Sensors)



More Detailed Agenda

- Electronics Fundamentals
  - *Lights!*
  - *Speed!*
  - ***Action!*** (Servos and motors)

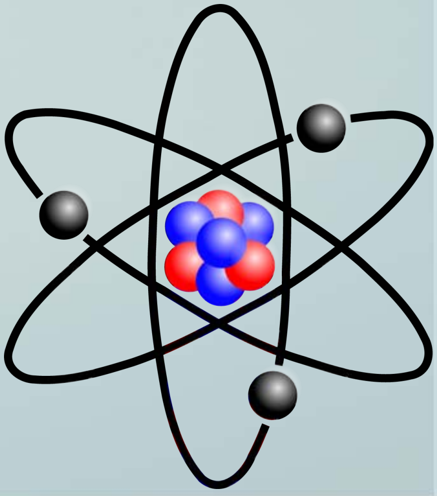
More Detailed Agenda

- Electronics Fundamentals
  - *Lights!*
  - *Speed!*
  - *Action!*
- Conclusions and context

Serpente Rosso, Erik Brunvand, 2012

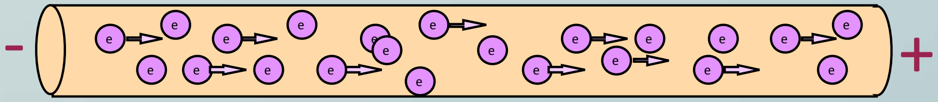
Electronics Fundamentals

- Electronics: A variety of phenomena related to ***charge moving*** in response to a ***force***
  - ***Charge?*** Charged subatomic particles
    - Protons and electrons
  - ***Moving?*** Electronic current
    - Like charges repel; Opposite charges attract
  - ***Force?*** Electromagnetic fields
    - Measured as voltage



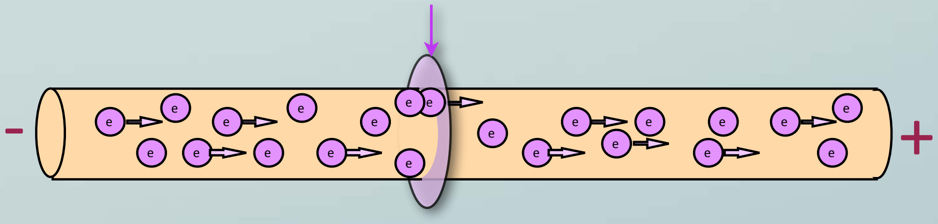
Moving Charge

- Apply an electric field
  - electrons are influenced by the field - move in response
- Charge is measured in coulombs
  - One coulomb = charge on 6.241×10^{18} electrons
  - Copper has 1.38×10^{24} free electrons / in^3



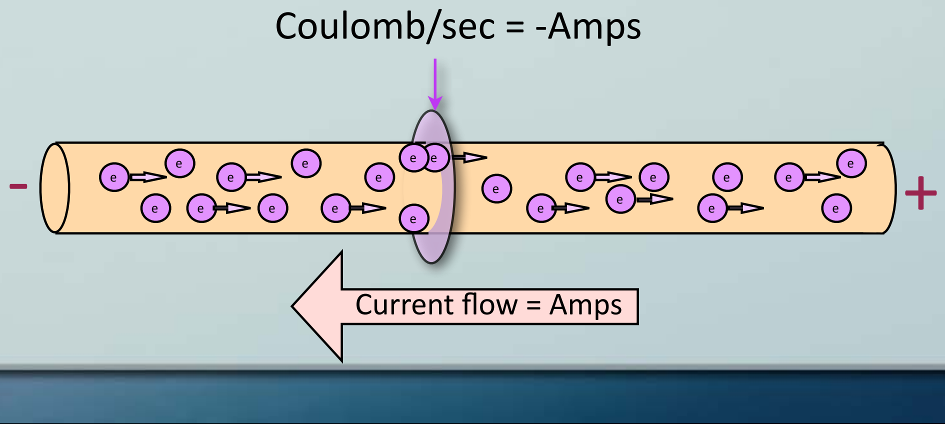
Current in Amperes (Amps)

- One Ampere of current is one Coulomb of charge moving past a point in one Second


Coulomb/sec = Amps

Confusing Concept #1: Current Direction

- Conventional current (positive current) flows from positive to negative!
  - As if charge carriers were positive…



Voltage: Force acting on charge

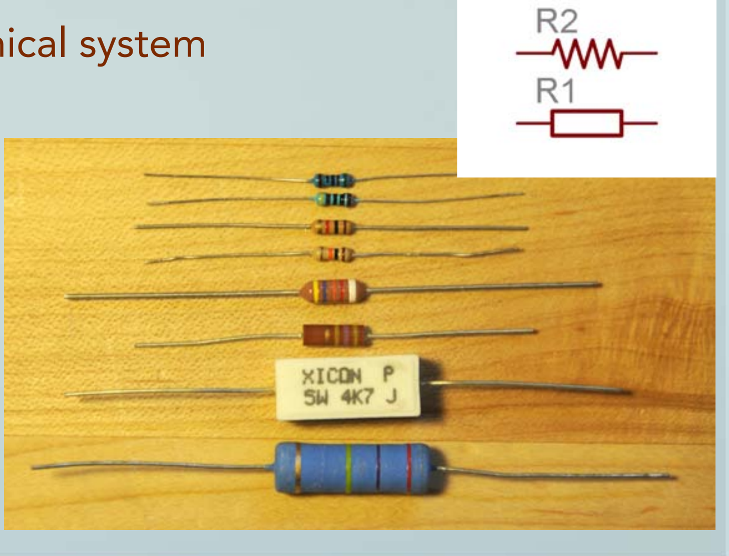
- Electrical force is measured in volts
  - Voltage is potential energy
  - 1v is the energy required to move 1 coulomb of charge
  - Two points in a circuit characterized by their ***voltage difference*** (not an absolute quantity)
- Arbitrary reference point for 0v called Ground (GND)
  - In your house, this is actually the ground…

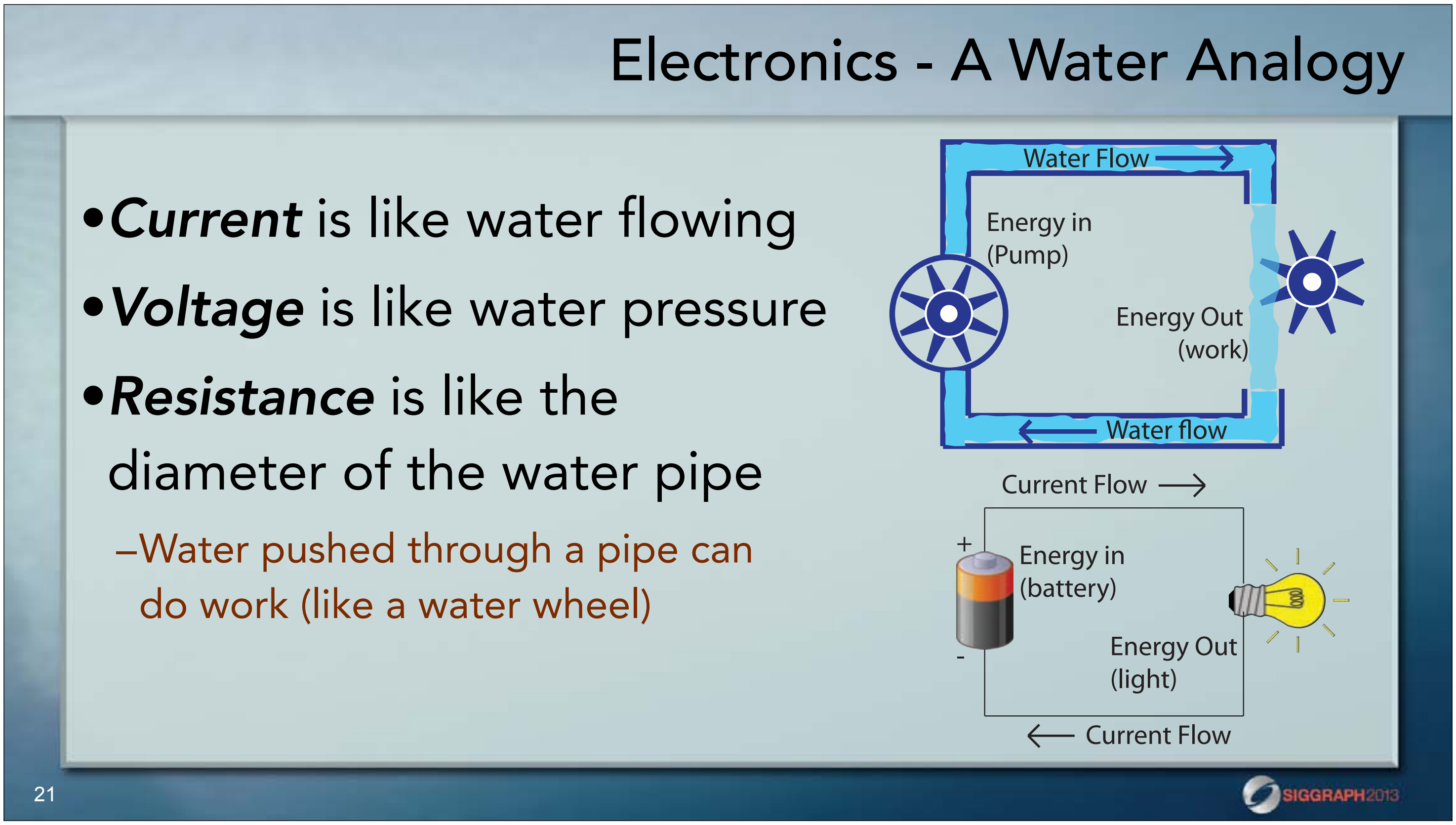
Controlling Charge: Big Idea #1

- ***Charge moving in a conductor (current) under the influence of a voltage is the main electrical activity that we’re interested in***
  - This phenomenon powers LEDs, makes motors move, and is the property that we’ll sense in a sensor to measure our environment.
  - ***Causing current to flow, and controlling that current, is one of our main goals!***

Resistance

- The property of a material to resist current flow
  - Similar to friction in a mechanical system
  - Measured in Ohms
  - Using the symbol Ω
- Color codes for values
  - Look for “resistor calculator” on the web…





Electronics - A Water Analogy

- ***Current*** is like water flowing
- ***Voltage*** is like water pressure
- ***Resistance*** is like the diameter of the water pipe
  - –Water pushed through a pipe can do work (like a water wheel)

Water Flow

Energy in (Pump)

Energy Out (work)

Water flow

Current Flow

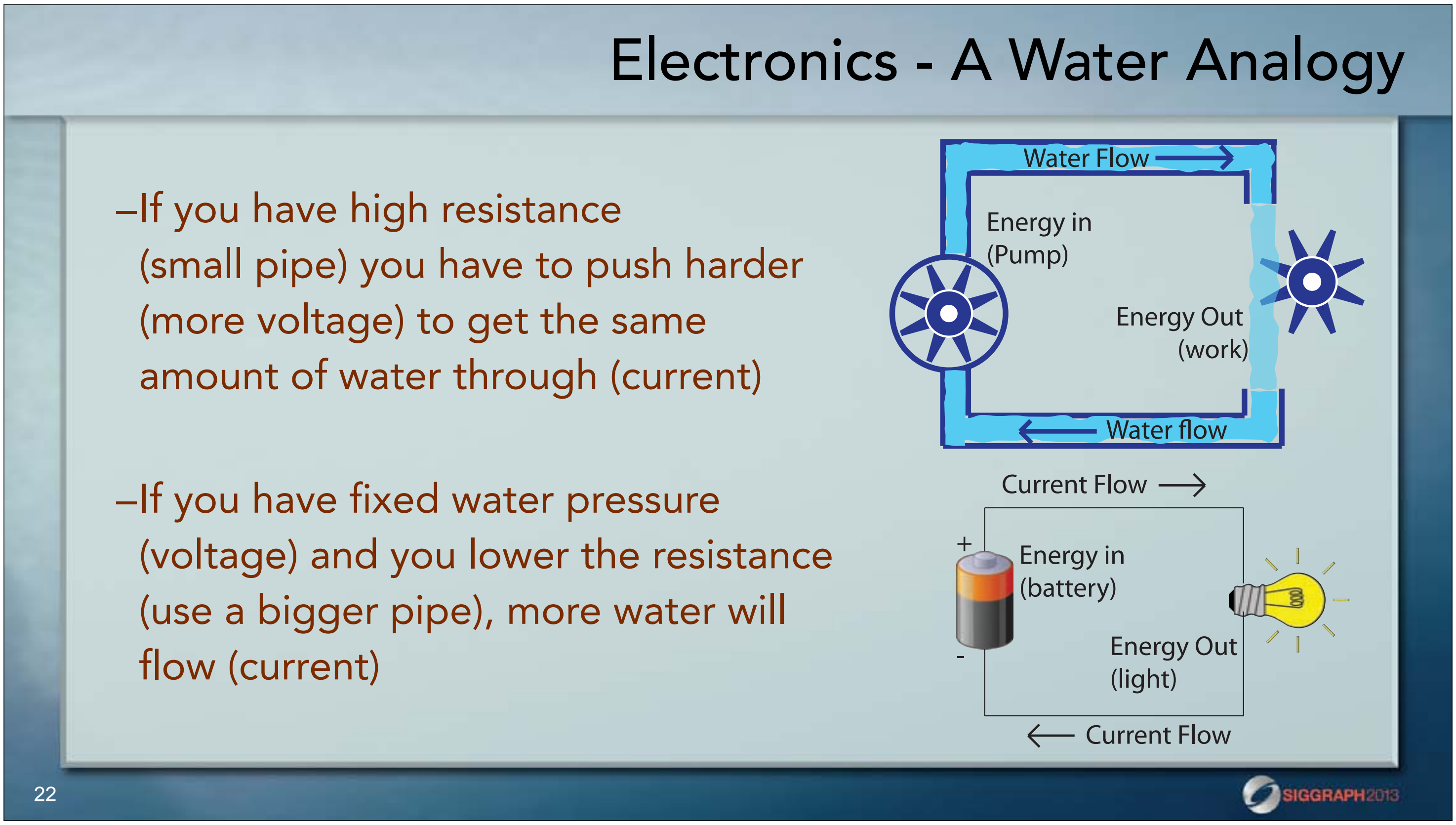
+

Energy in (battery)

Energy Out (light)

-

Current Flow



Electronics - A Water Analogy

–If you have high resistance (small pipe) you have to push harder (more voltage) to get the same amount of water through (current)

–If you have fixed water pressure (voltage) and you lower the resistance (use a bigger pipe), more water will flow (current)

Water Flow

Energy in (Pump)

Energy Out (work)

Water flow

Current Flow

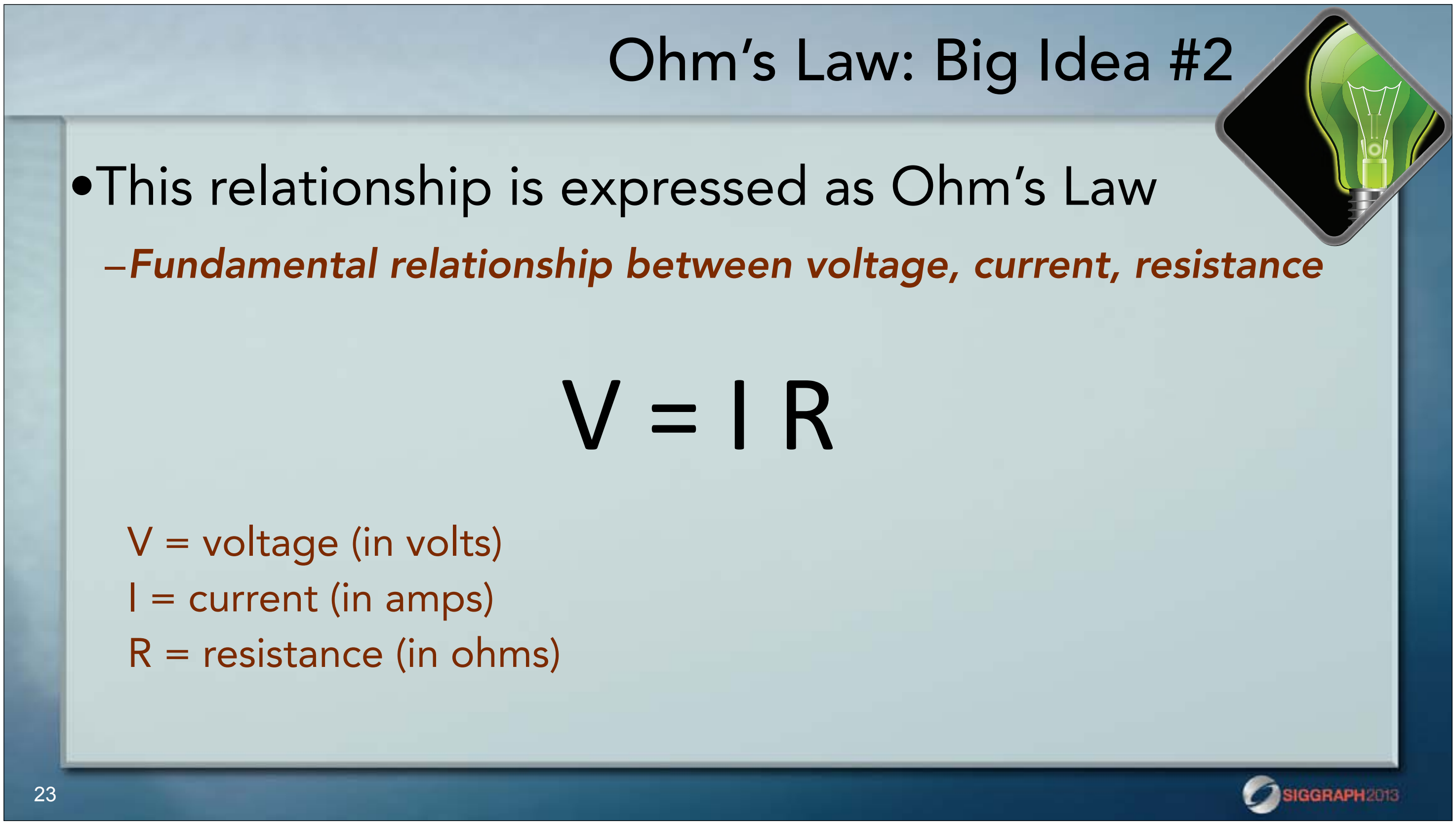
+

Energy in (battery)

Energy Out (light)

-

Current Flow

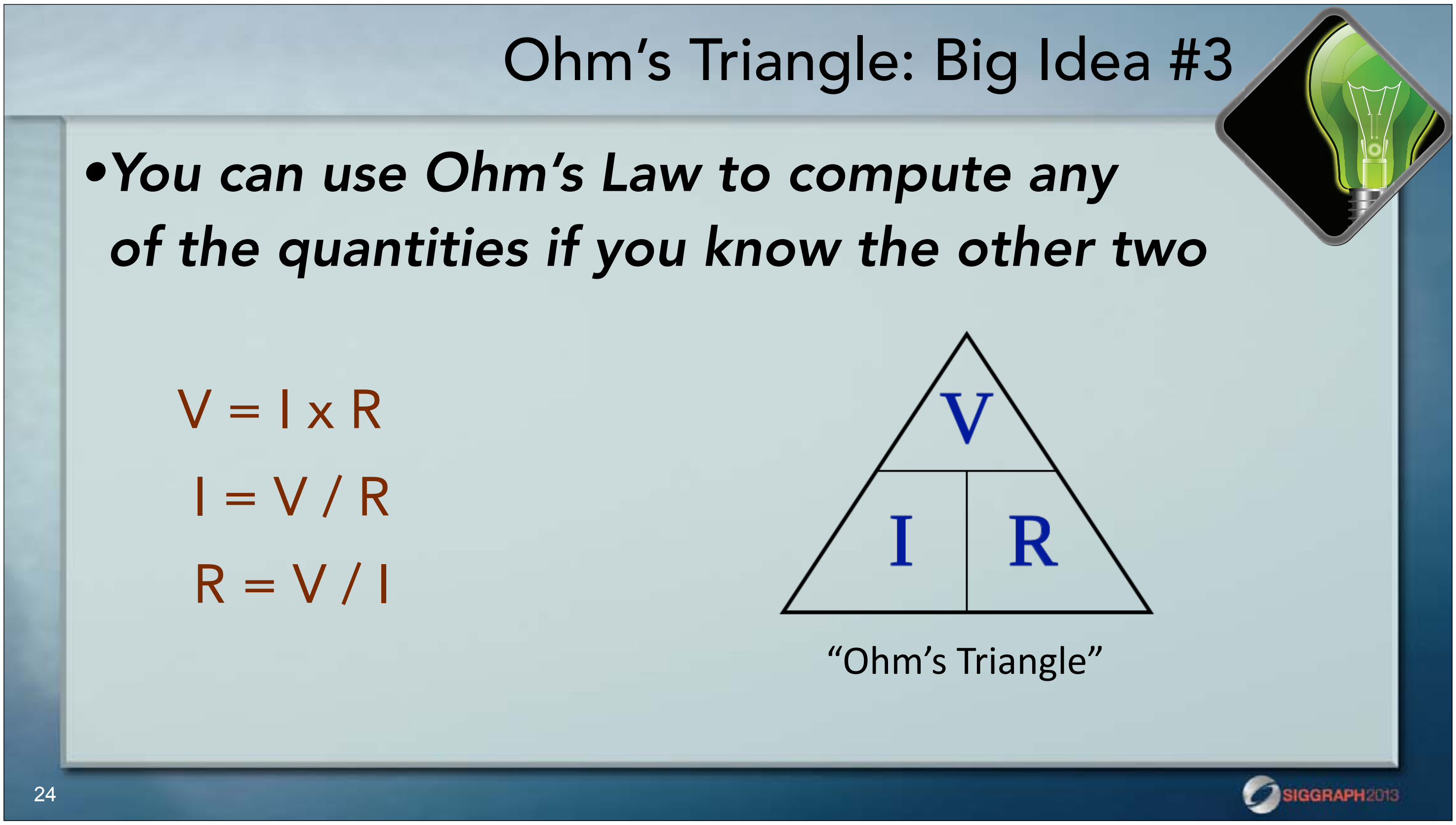


Ohm's Law: Big Idea #2

- This relationship is expressed as Ohm's Law
  - *Fundamental relationship between voltage, current, resistance*

$$V = I R$$

V = voltage (in volts)
I = current (in amps)
R = resistance (in ohms)



Ohm's Triangle: Big Idea #3

- *You can use Ohm's Law to compute any of the quantities if you know the other two*

$V = I \times R$

$I = V / R$

$R = V / I$

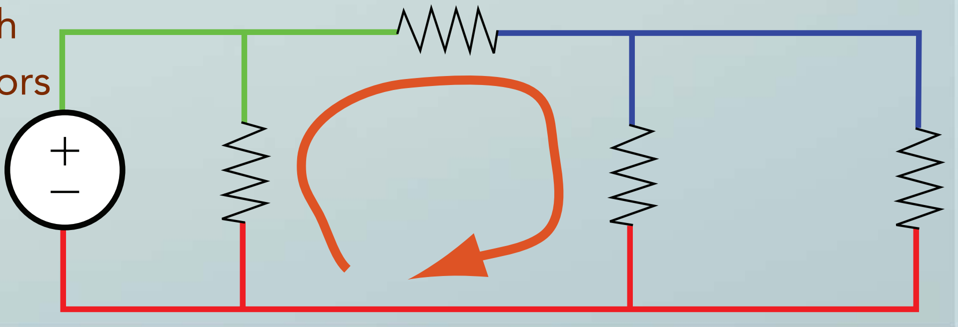
"Ohm's Triangle"

Electronics: Practical Matters

- Ranges of values you're likely to encounter
  - ***Current:*** A few amps (A) to milliamps (mA)
    - 1 mA = 0.001A
    - Current is indicted as i or I in circuits
  - ***Voltage:*** A few volts to millivolts (mV)
    - 1.5v, 3.3v, 5v, and 12v are common
  - ***Resistance:*** huge useful range from ohms (Ω), to kilo-ohms (kΩ), to mega-ohms (MΩ)
    - 1kΩ = 1,000Ω   1MΩ = 1,000,000Ω

Circuit Nodes: Big Idea #5

- Electrical node in a circuit
  - All points connected through a conductor are at the same electrical potential (same colors below)
- Electrical loop in a circuit
  - A connected path through conductors and components that ends up where it started



Kirchhoff's Laws

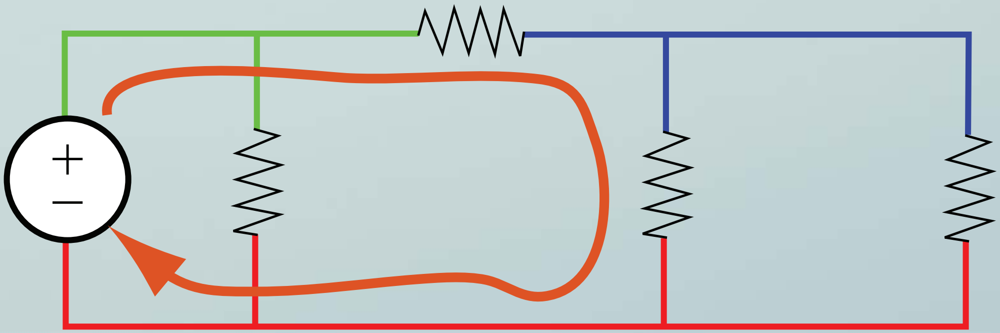
- ***Kirchhoff's Voltage Law (KVL)***
  - The sum of voltages around a circuit loop is 0v
  - Like a loop hike - just as much uphill as downhill

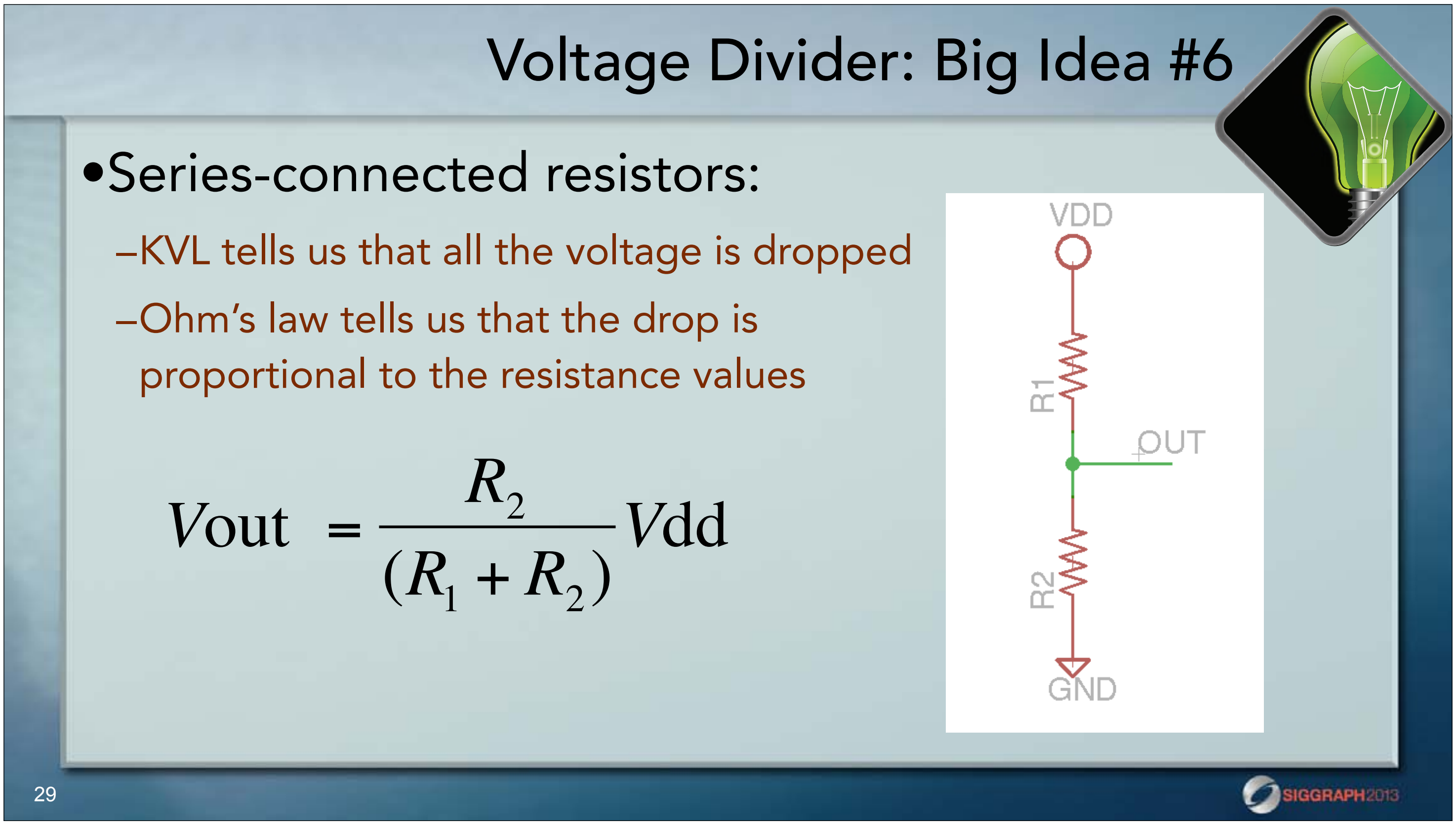
- ***Kirchhoff's Current Law (KCL)***
  - The sum of currents into and out of a node is 0A
  - Like a river splitting into streams:
    water in = water out

cc-by-sa-3.0|Courtesy Spinningspark at Wikipedia

"Voltage Drop" a la Kirchhoff

- Consider a loop with a battery in it
  - 5v supplied by battery, and resistors in the loop
  - KVL tells us that voltage is "used up" by the end of the loop
  - Where did it go? It's "dropped" across each component

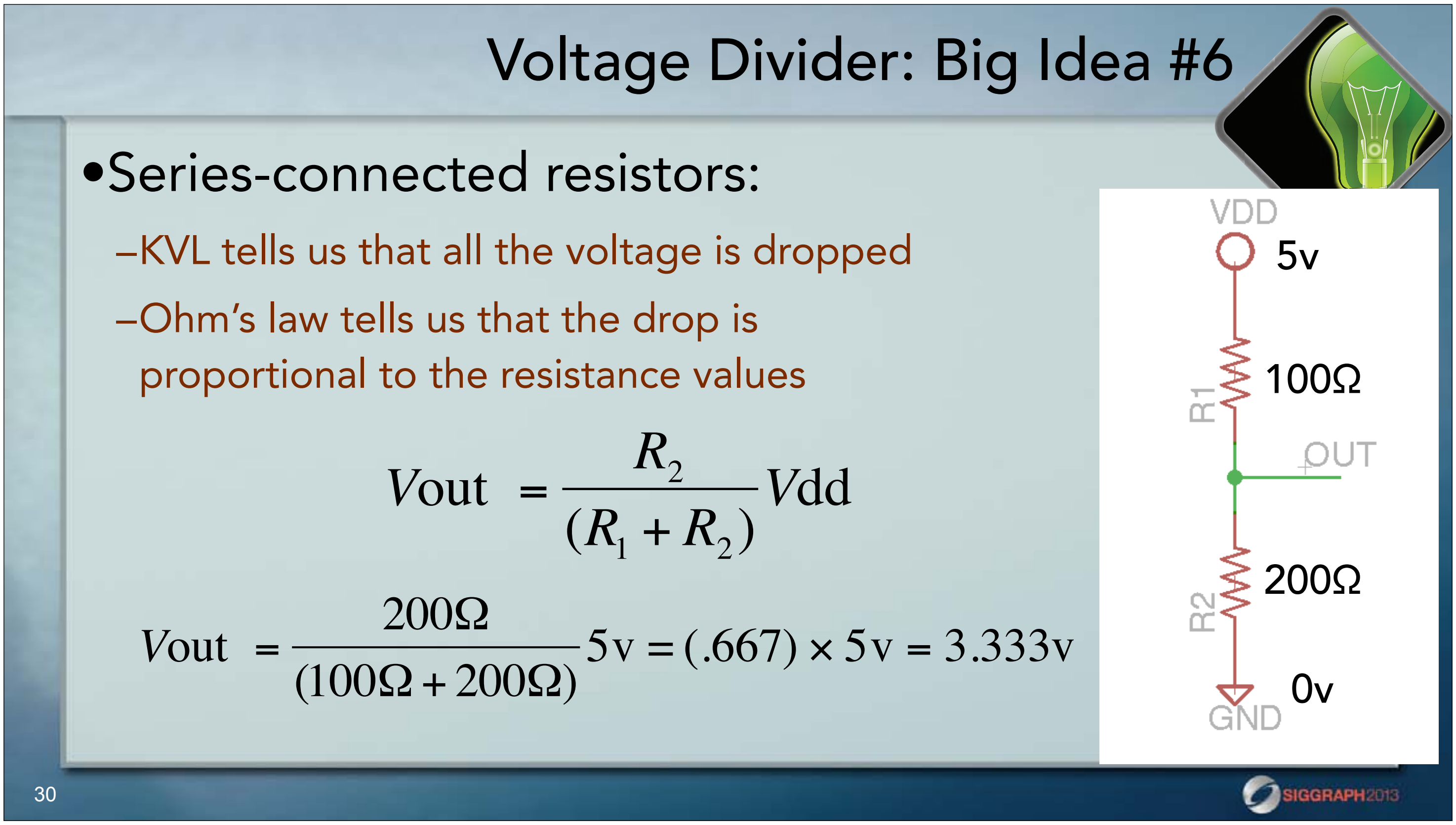




Voltage Divider: Big Idea #6

- Series-connected resistors:
  - KVL tells us that all the voltage is dropped
  - Ohm's law tells us that the drop is proportional to the resistance values

$$Vout = \frac{R_2}{(R_1 + R_2)} Vdd$$



Voltage Divider: Big Idea #6

- Series-connected resistors:
  - KVL tells us that all the voltage is dropped
  - Ohm's law tells us that the drop is proportional to the resistance values

$$Vout = \frac{R_2}{(R_1 + R_2)} Vdd$$

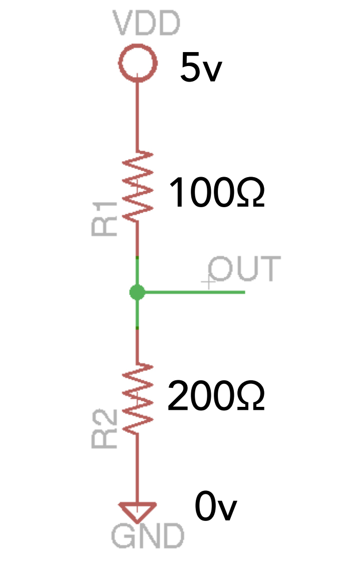
$$Vout = \frac{200\Omega}{(100\Omega + 200\Omega)} 5\text{v} = (.667) \times 5\text{v} = 3.333\text{v}$$

Voltage Divider: Big Idea #6

- Series-connected resistors:
  - –KVL tells us that all the voltage is dropped to GND
  - –KCL tells us that both resistors see the same current
  - –Ohm’s law tells us that the drop is proportional to the resistance values

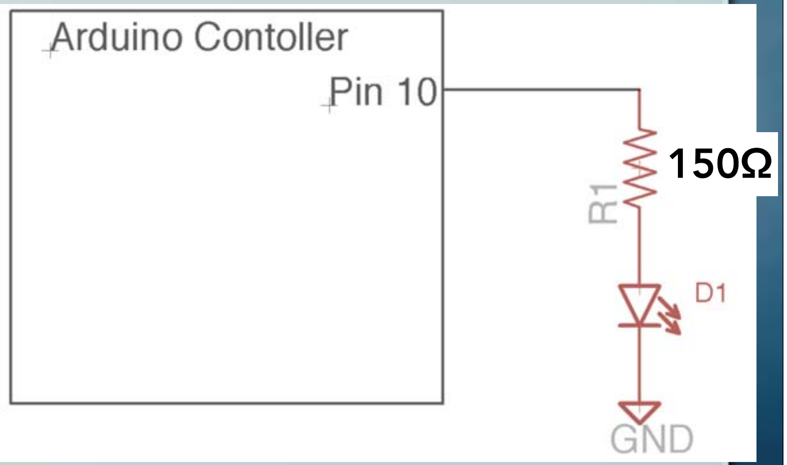
I = V/R = 5v / (100Ω + 200Ω) = .0167A

V1 = IR1 = .0167A x 100Ω = 1.667v (Vdd to OUT)
V2 = IR2 = .0167A x 200Ω = 3.333v (OUT to GND)

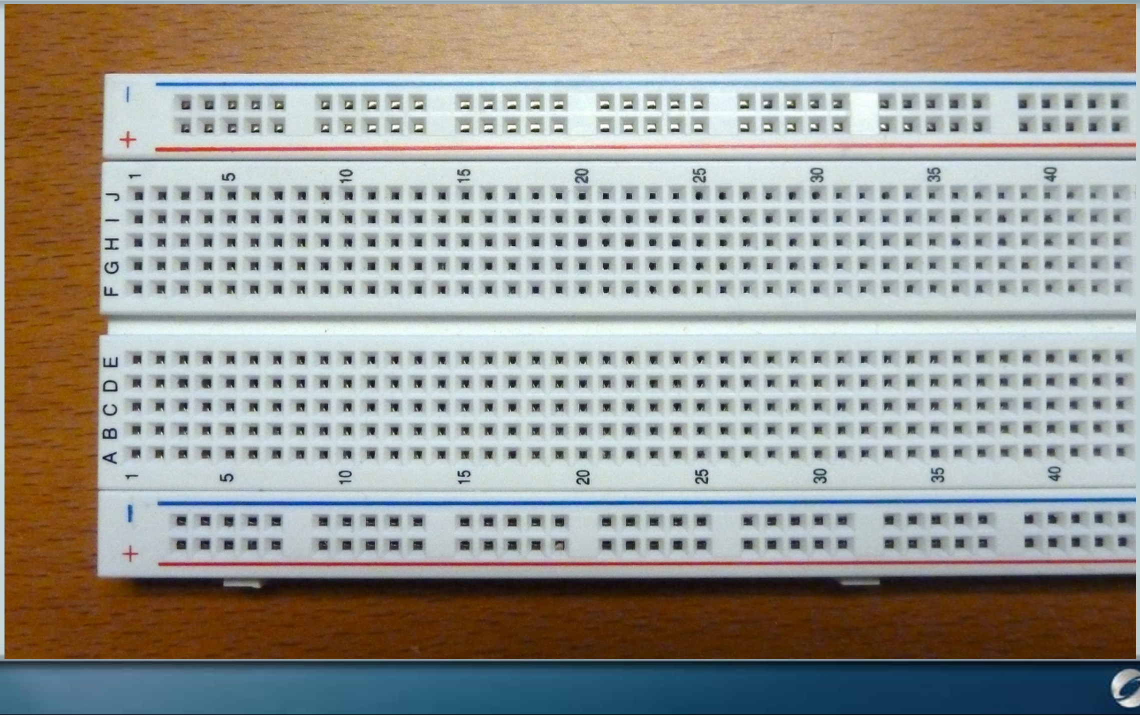


Circuit Practicalities: Wiring Things Up

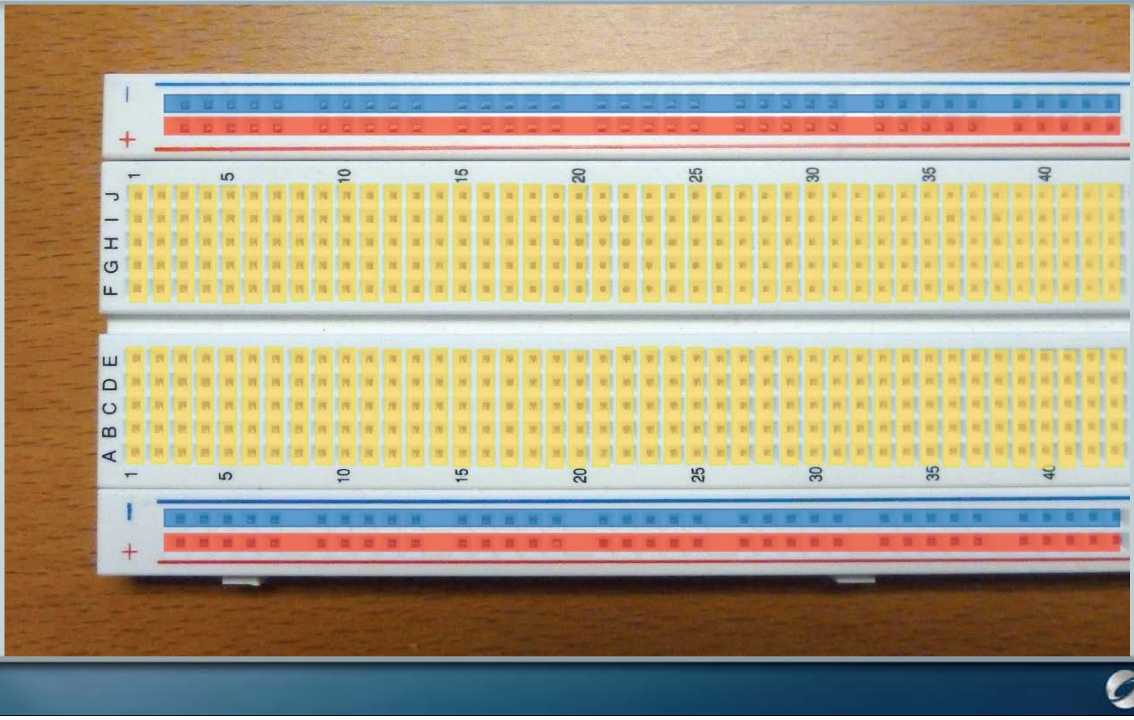
- Schematics describe the logical connections between components



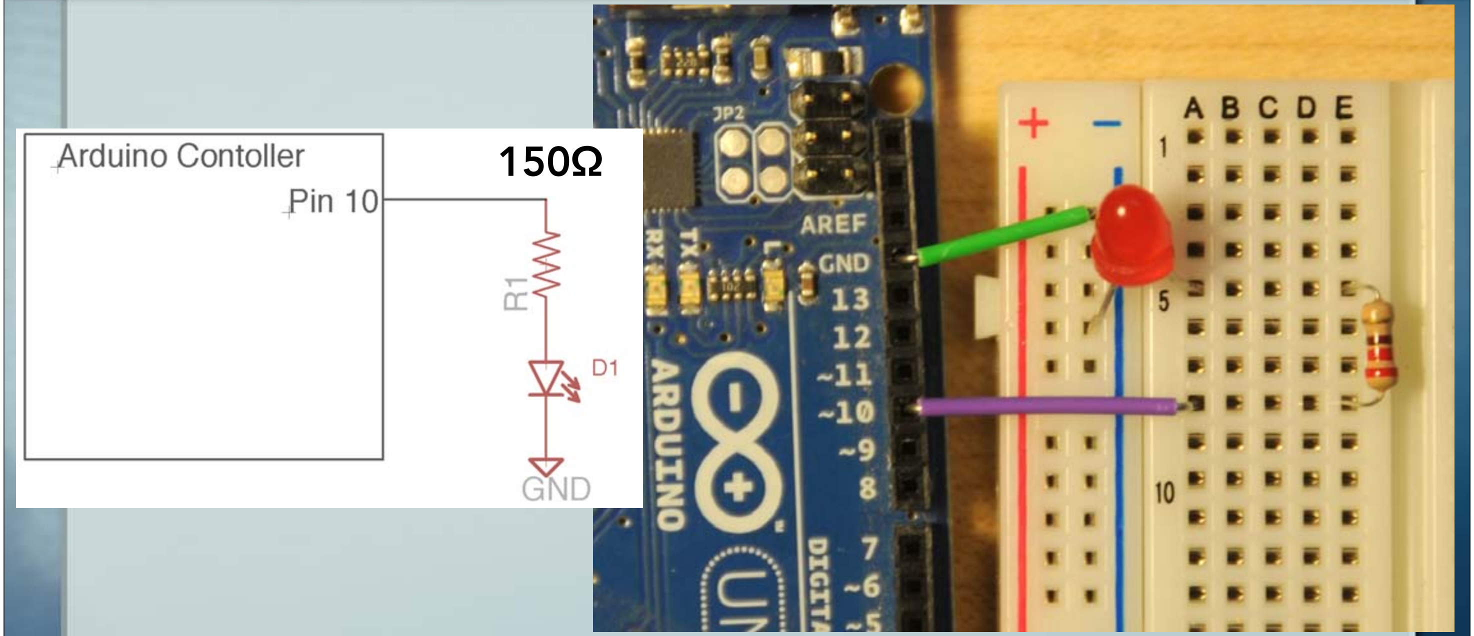
“Solderless Breadboad”



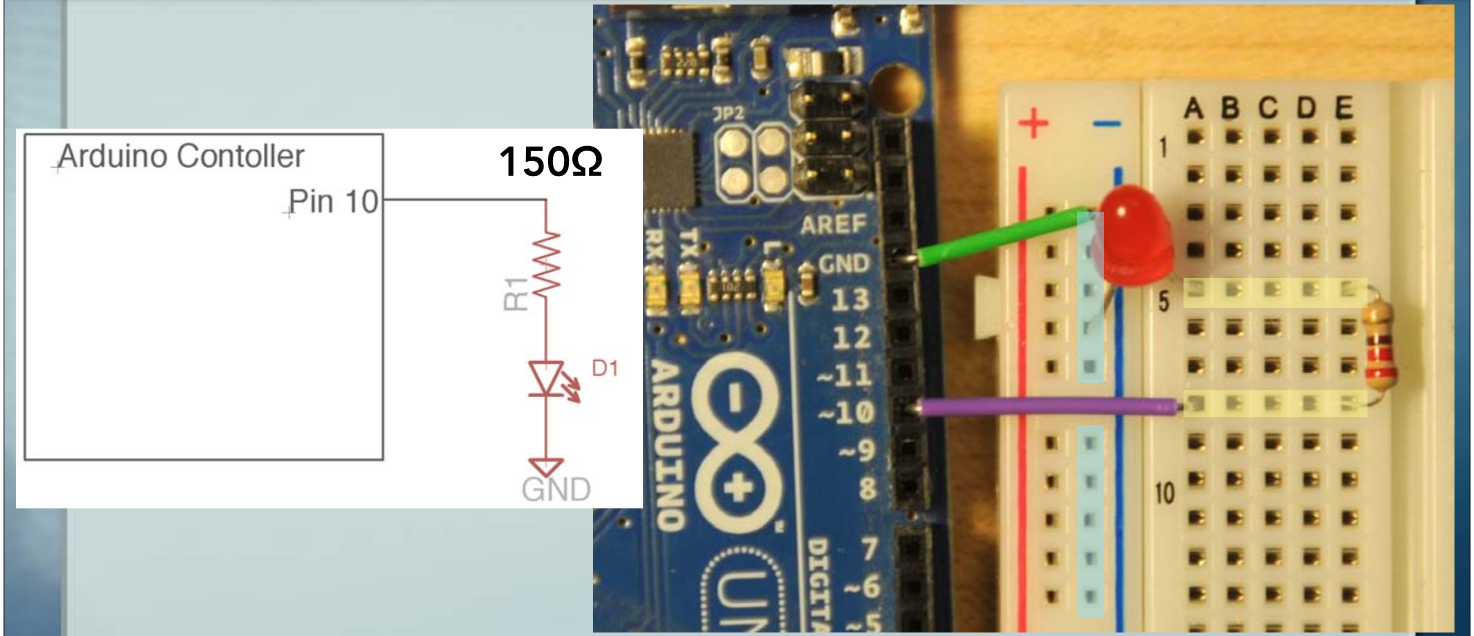
“Solderless Breadboad”



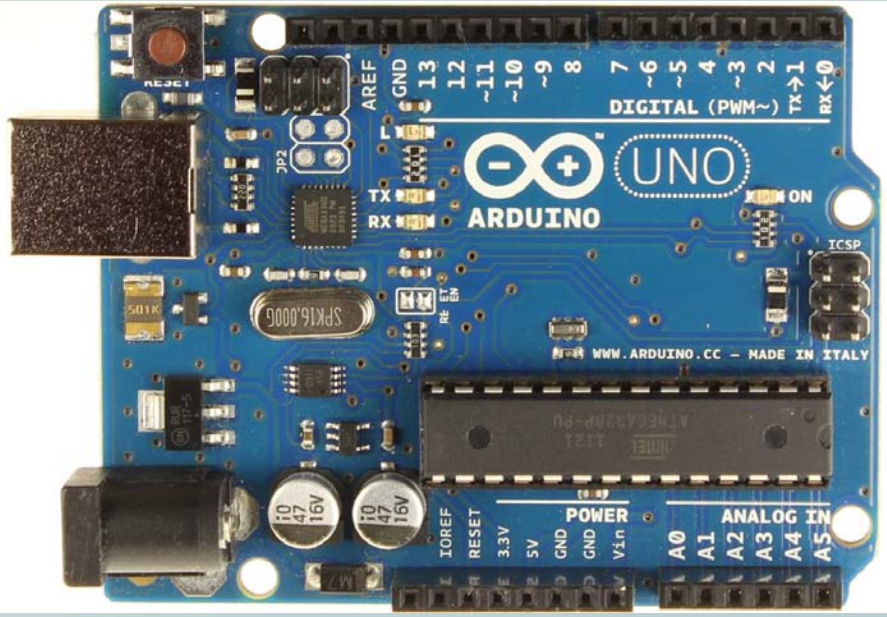
Circuit Practicalities: Wiring Things Up



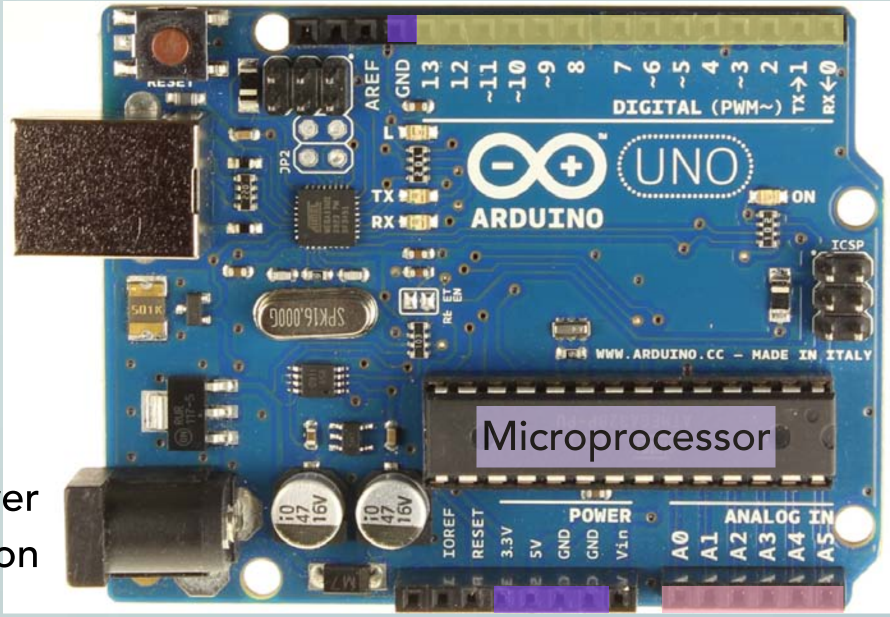
Circuit Practicalities: Wiring Things Up



Circuit Practicalities: Arduino



Circuit Practicalities: Arduino



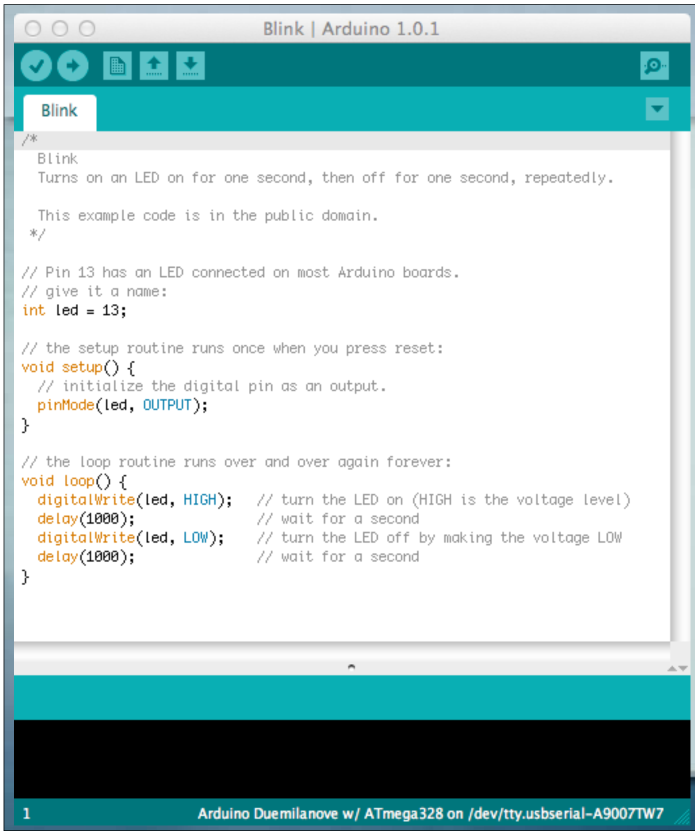
Digital Pins

USB Connection

Power Connection

Analog Pins

Power Pins


Blink | Arduino 1.0.1
Blink
1
Arduino Duemilanove w/ ATmega328 on /dev/tty.usbserial-A9007TW7

```
/*
  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
}
```

Software Platform: Arduino

- www.arduino.cc
  - Simple open source IDE
  - Arduino code is really C/C++
  - avr-gcc is the back end

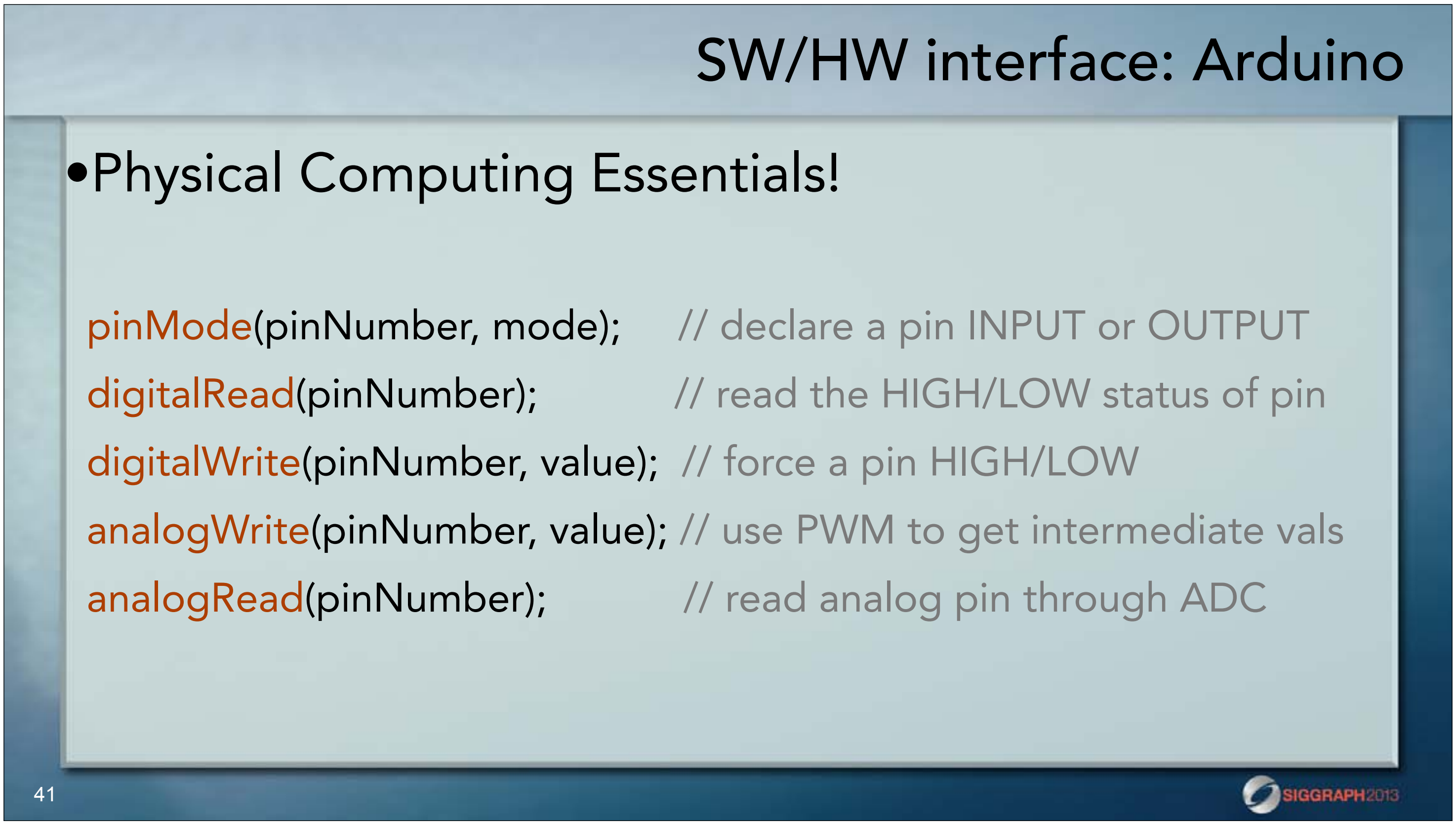
SW/HW interface: Arduino

- Two required functions

  void setup(){…}   // Runs once at startup

  void loop(){…}   // Loops forever after setup()

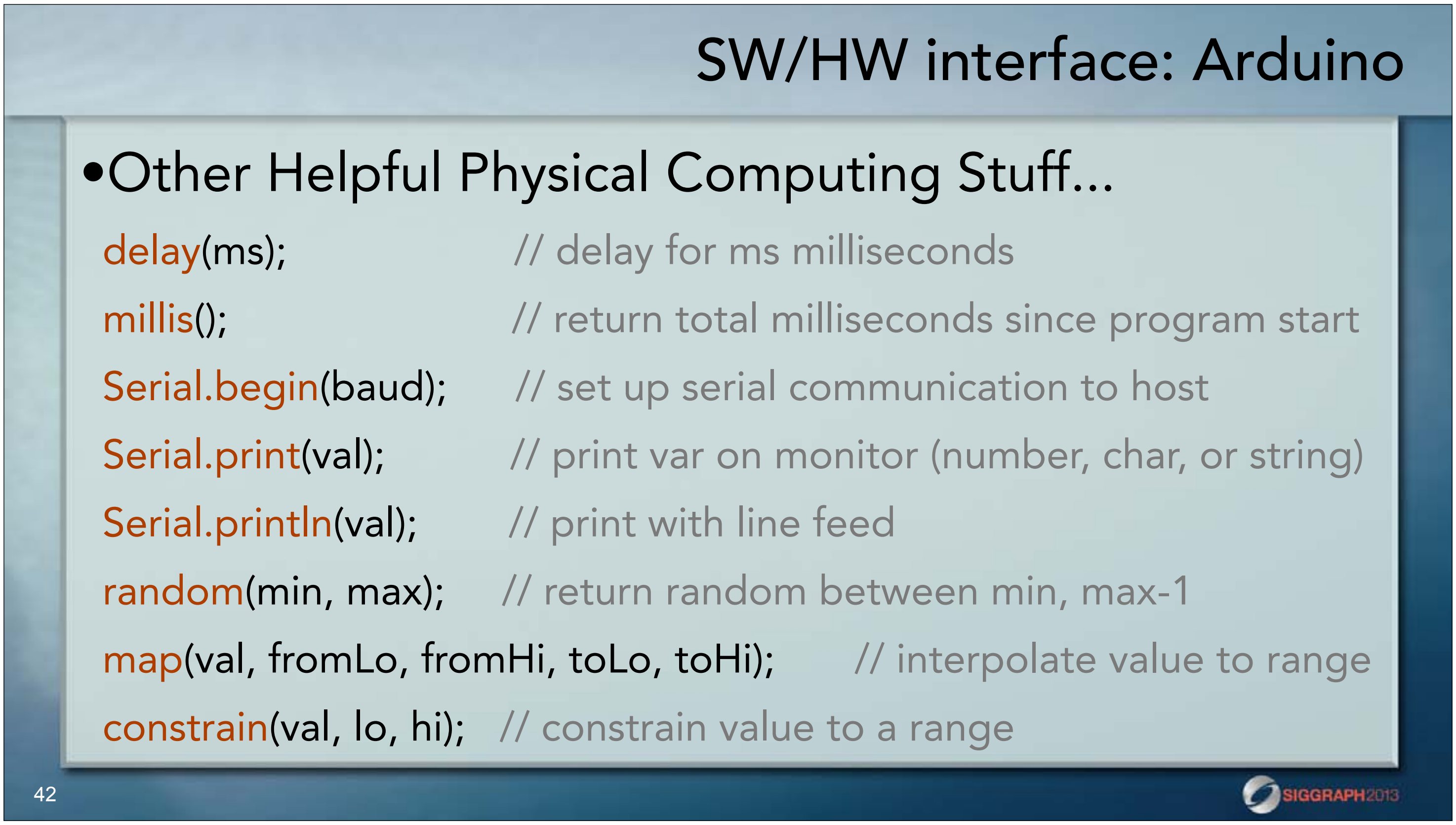
- Standard(ish) C/C++ data types
  - Boolean (1 bit)
  - char (signed 8 bits), byte (unsigned 8 bits)
  - int (16 bits), long (32 bits)
  - float (32 bits), double (32 bits)



SW/HW interface: Arduino

- Physical Computing Essentials!

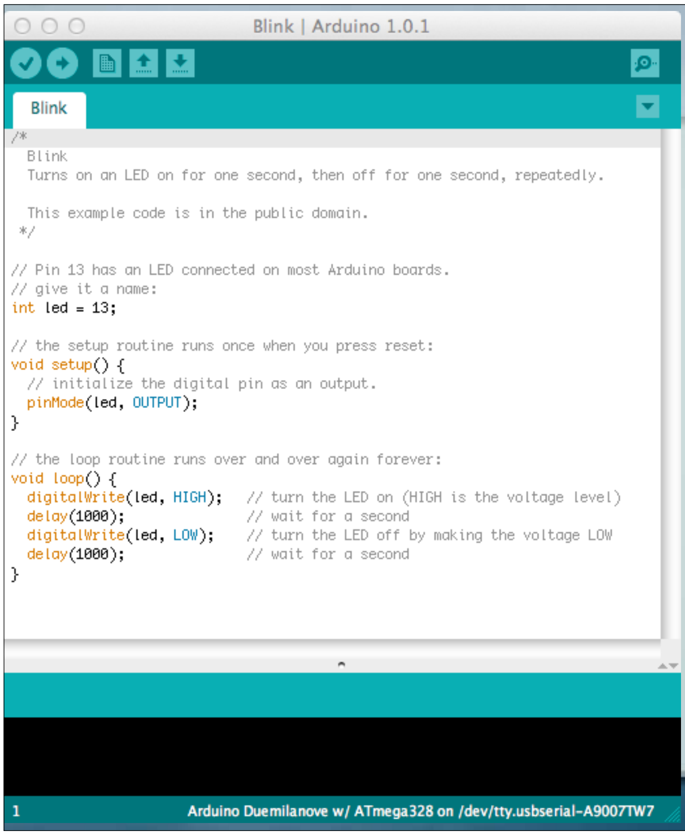
```
pinMode(pinNumber, mode);       // declare a pin INPUT or OUTPUT
digitalRead(pinNumber);         // read the HIGH/LOW status of pin
digitalWrite(pinNumber, value); // force a pin HIGH/LOW
analogWrite(pinNumber, value);  // use PWM to get intermediate vals
analogRead(pinNumber);          // read analog pin through ADC
```



SW/HW interface: Arduino

- Other Helpful Physical Computing Stuff…

```
delay(ms);                   // delay for ms milliseconds
millis();                    // return total milliseconds since program start
Serial.begin(baud);          // set up serial communication to host
Serial.print(val);           // print var on monitor (number, char, or string)
Serial.println(val);         // print with line feed
random(min, max);            // return random between min, max-1
map(val, fromLo, fromHi, toLo, toHi);      // interpolate value to range
constrain(val, lo, hi);      // constrain value to a range
```



```
/*
  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
}
```

```
1        Arduino Duemilanove w/ ATmega328 on /dev/tty.usbserial-A9007TW7
```

Example: Blink

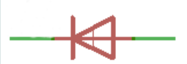
```
int led = 13;

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

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

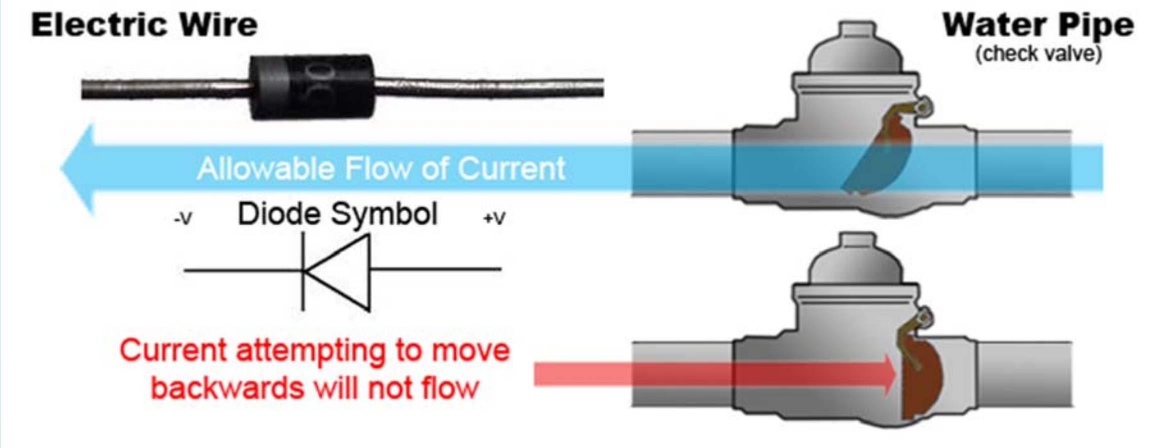
Lights!

LEDs

Diodes

- One-way valves for current

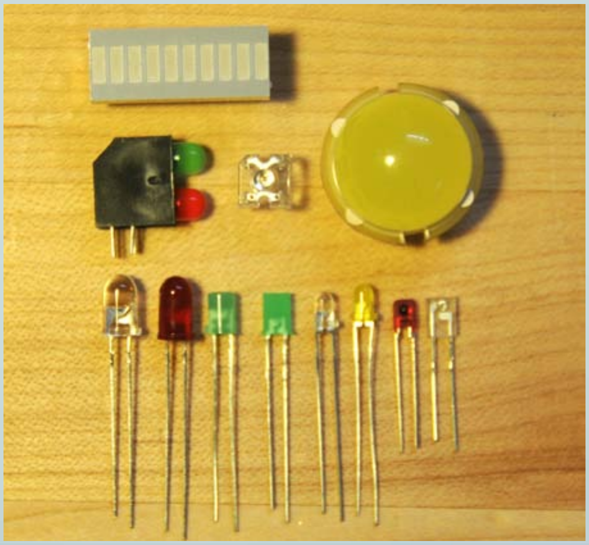
Cathode  Anode

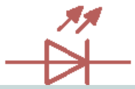

Electric Wire
Water Pipe
(check valve)
Allowable Flow of Current
-V
Diode Symbol
+V
Current attempting to move backwards will not flow

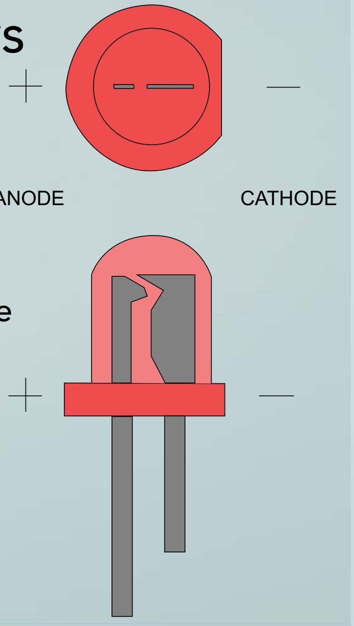
http://sargosis.com/

Light Emitting Diodes (LEDs)

- Diodes that light up when current flows



Anode
+

Cathode
-



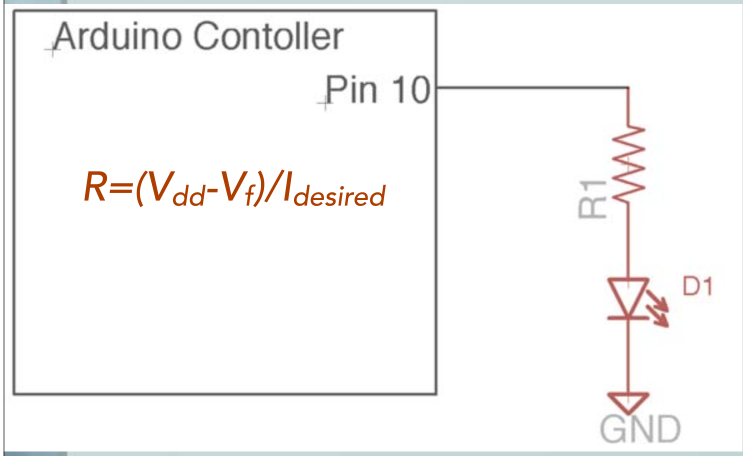
LED Practicalities

- Diodes have a *“forward voltage”* or *“diode drop”*
  - Typically V_f is around 0.7v for a diode, and 1.5v to 3.0v for an LED
- Diodes also have a current limit
  - Typically 20mA for an LED
  - If you don’t limit the current, they’ll burn out

CC-BY-SA-2.5:*GRL Graffiti research Lab http://graffitiresearchlab.com*

Current-Limiting Resistor: Big Idea #10

Remember, KCL says that all series connected components see the same current!

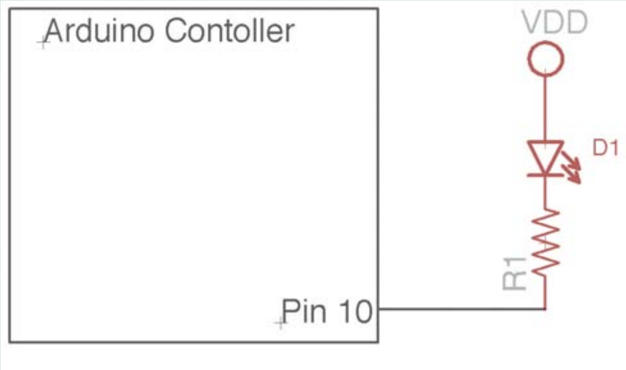


$R=(V_{dd}-V_f)/I_{desired}$

- Assume Pin10 can supply 5v
- Assume LED V_f is 2.0v
- (5v - 2v) = 3v remaining for R1
- We want 20mA
- R = V/I = 3v / .020A
- R = 150Ω
- In practice, 150Ω - 330Ω will work

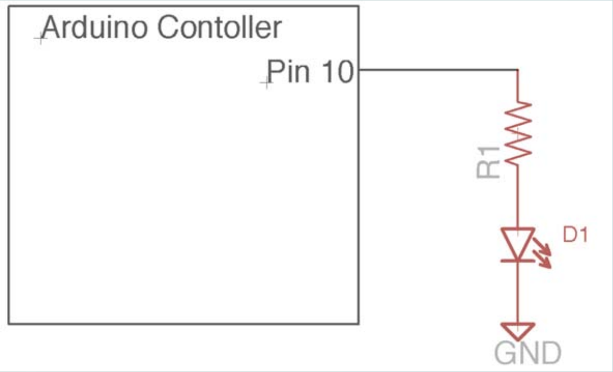
Current Sink vs. Source

Sink means “pull to ground”



LED lights up when Pin 10 is LOW (0v)

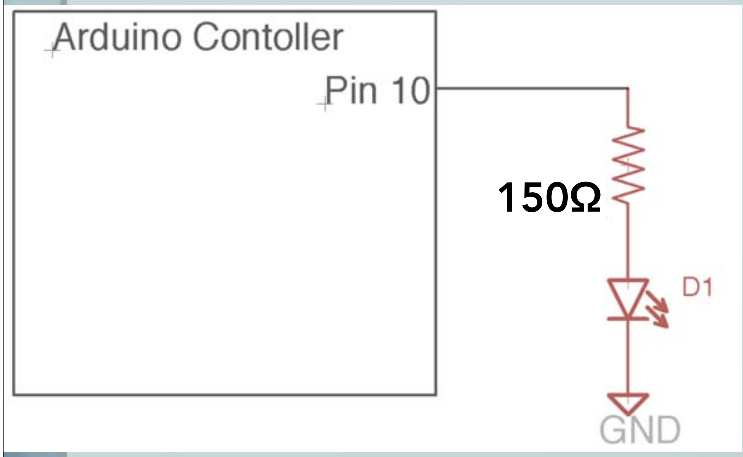
Source means “pull to Vdd”



LED lights up when Pin 10 is HIGH (5v)

Arduino digital pins source/sink a maximum of 40mA/pin
Arduino provides a max of 250mA total to all pins

Example Revisited: Blink

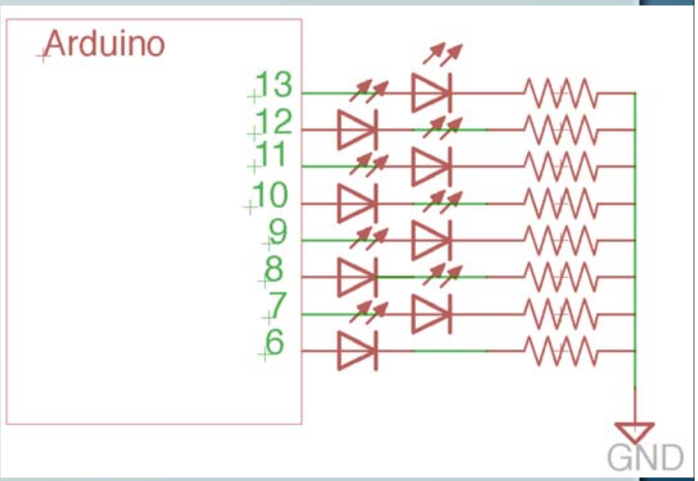


```
int led = 10;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Eight LEDs

- *Each LED needs its own current-limiting resistor!*

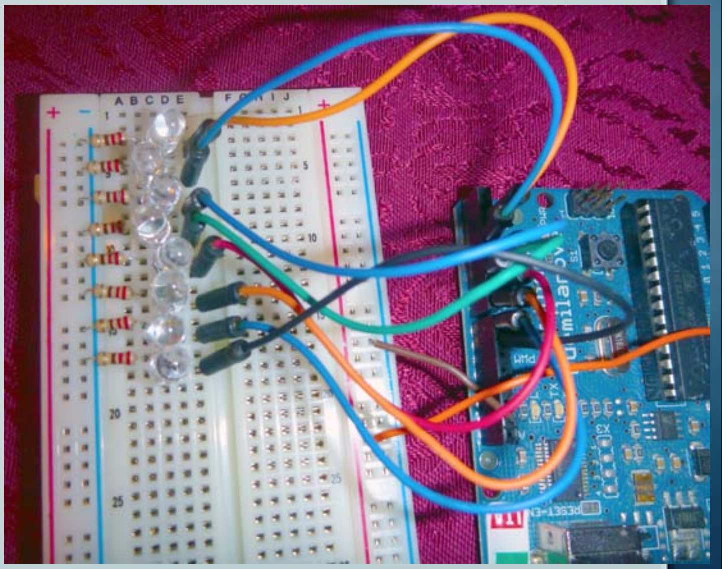
```
int leds[] = {6,7,8,9,10,11,12,13};
void setup(){
   for (int i=0; i<8; i++){
       pinMode(leds[i], OUTPUT);
}
void loop(){
    \\... set them HIGH and LOW
    \\ remember to delay(ms)!
}
```



Eight LEDs

- *Each LED needs its own current-limiting resistor!*

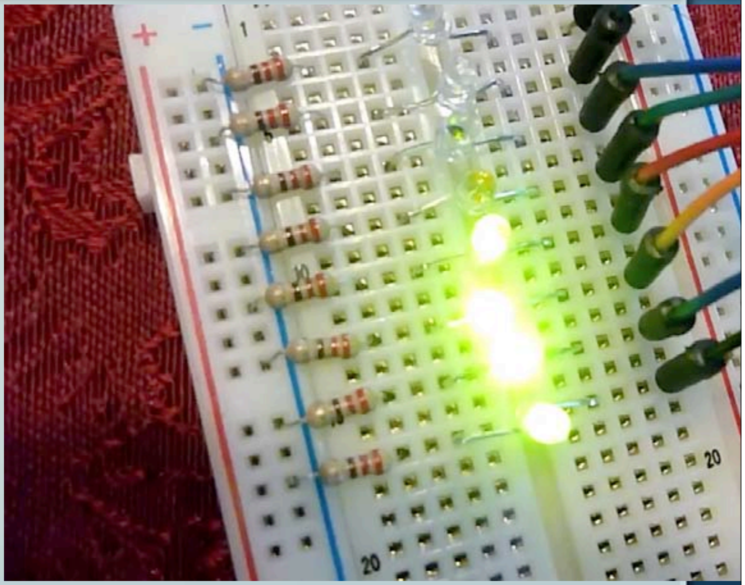
```
int leds[] = {6,7,8,9,10,11,12,13};
void setup(){
   for (int i=0; i<8; i++){
       pinMode(leds[i], OUTPUT);
}
void loop(){
    \\... set them HIGH and LOW
    \\ remember to delay(ms)!
}
```



Eight LEDs

- *Each LED needs its own current-limiting resistor!*

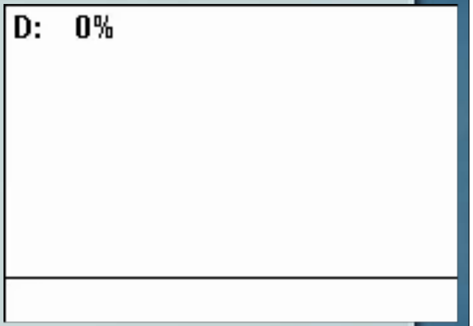
```
int leds[] = {6,7,8,9,10,11,12,13};
void setup(){
   for (int i=0; i<8; i++){
      pinMode(leds[i], OUTPUT);
}
void loop(){
     \\... set them HIGH and LOW
     \\ remember to delay(ms)!
}
```

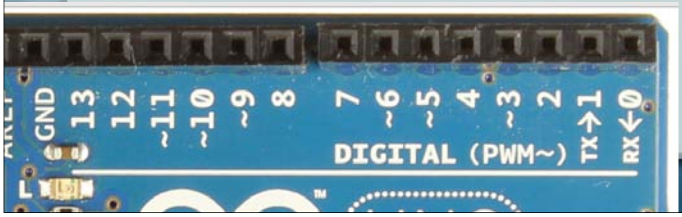


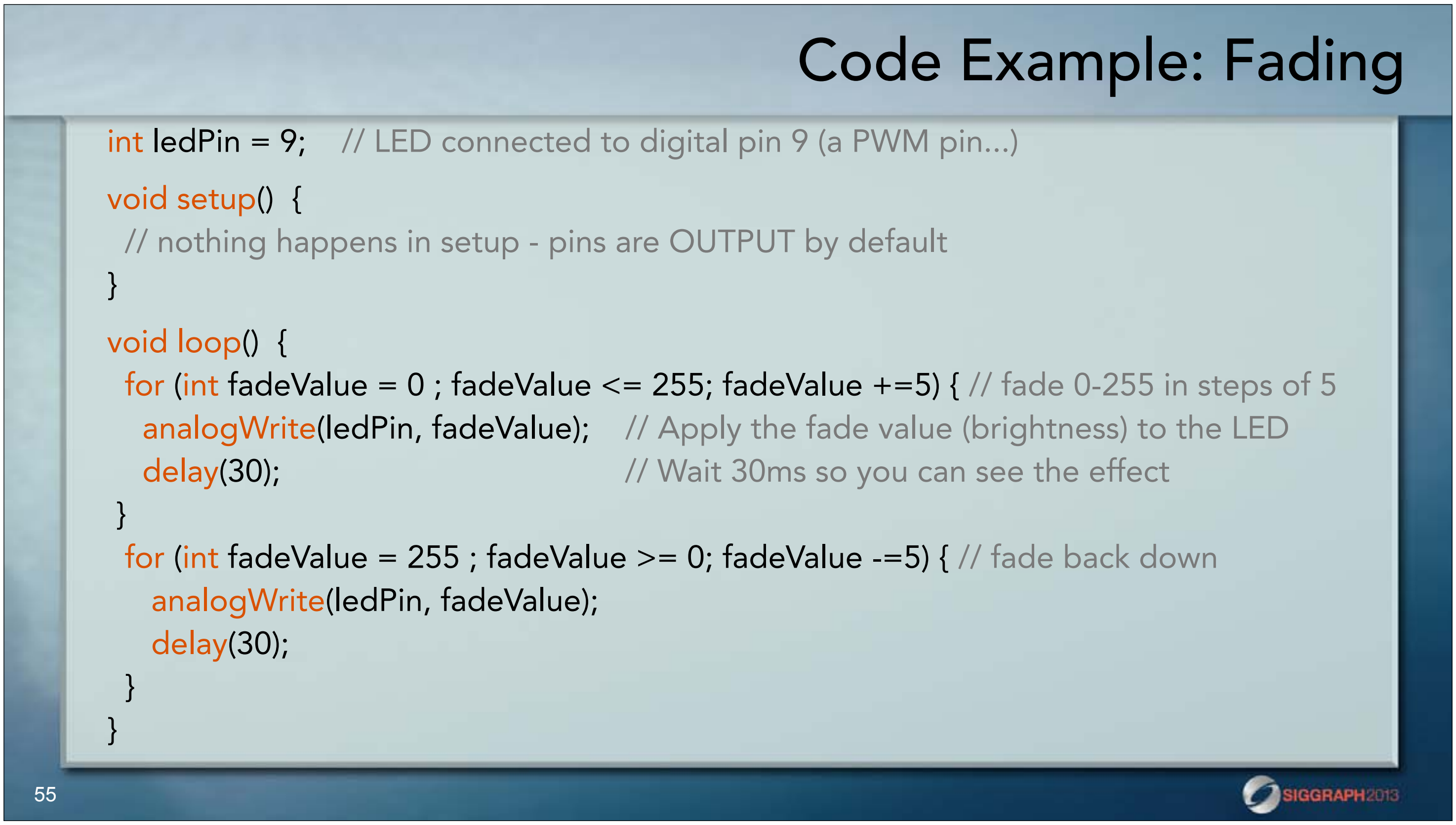
Dimming an LED

- LEDs are either all-on or all-off
  - –But, they go on and off really fast
  - –So if you flash them fast enough, they still look on, but dimmer
- ***"Pulse Width Modulation"   (PWM)***

```
analogWrite(pin, value); // value between 0-255
                         // Must be a "PWM pin"
```

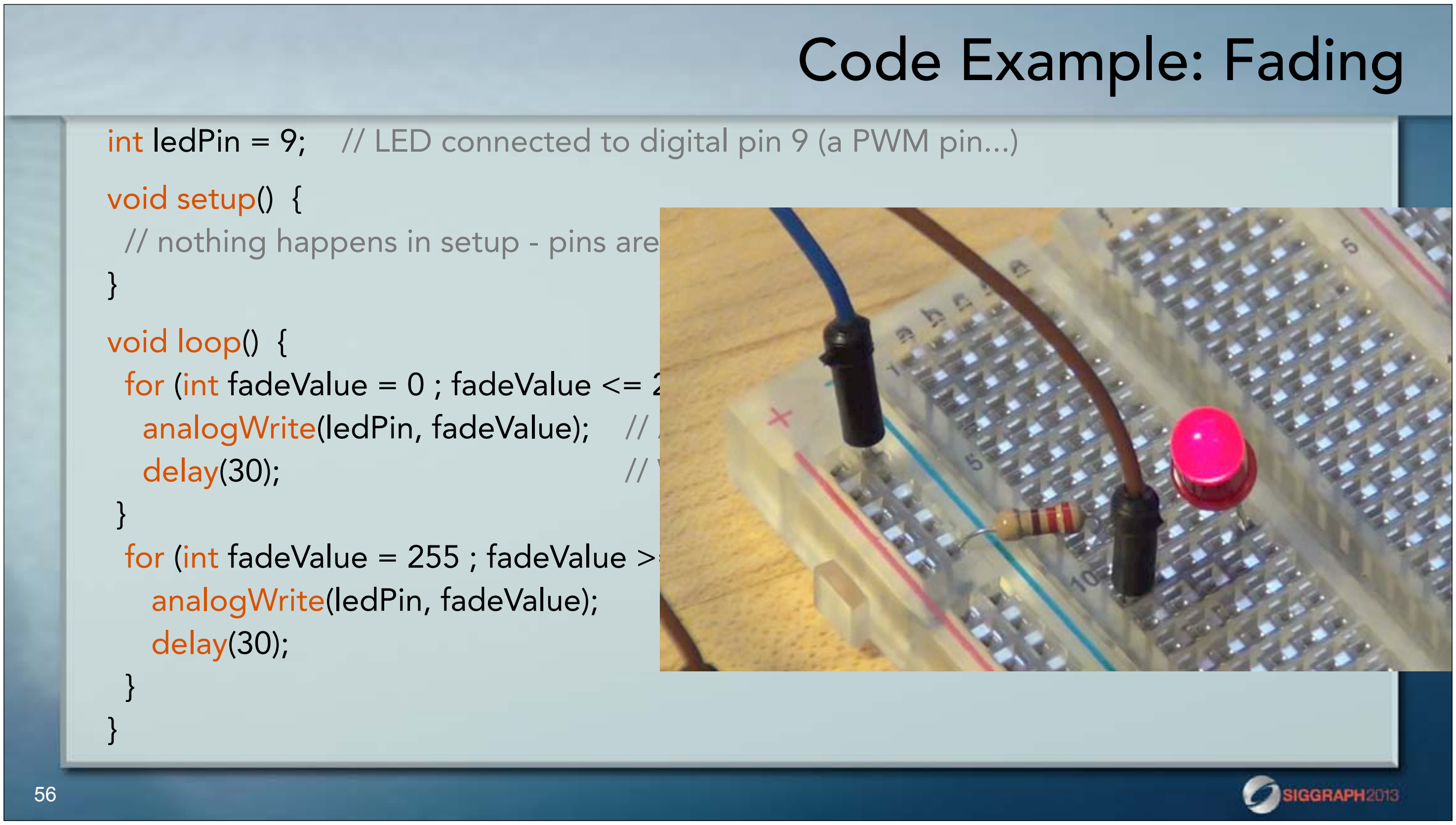






Code Example: Fading

```
int ledPin = 9;    // LED connected to digital pin 9 (a PWM pin...)

void setup()  {
  // nothing happens in setup - pins are OUTPUT by default
}

void loop()  {
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) { // fade 0-255 in steps of 5
    analogWrite(ledPin, fadeValue);    // Apply the fade value (brightness) to the LED
    delay(30);                         // Wait 30ms so you can see the effect
  }
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) { // fade back down
     analogWrite(ledPin, fadeValue);
     delay(30);
   }
}
```



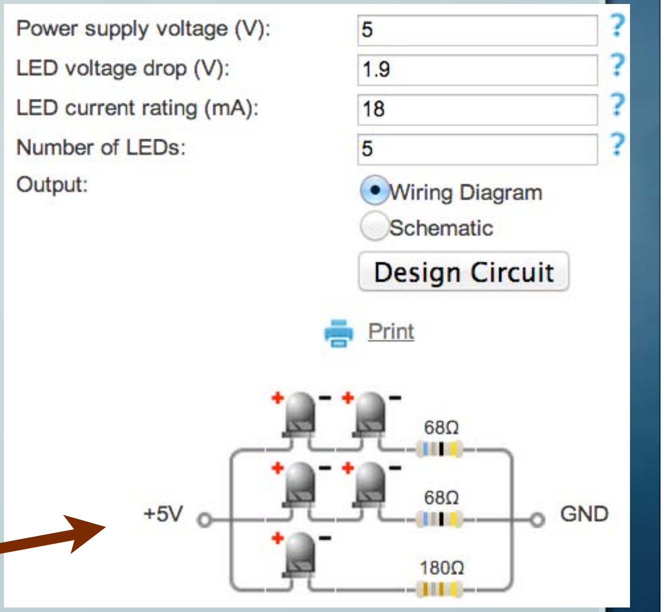
Code Example: Fading

```
int ledPin = 9;    // LED connected to digital pin 9 (a PWM pin...)

void setup()  {
  // nothing happens in setup - pins are
}

void loop()  {
  for (int fadeValue = 0 ; fadeValue <= 2
    analogWrite(ledPin, fadeValue);    // 
    delay(30);                         // 
  }
  for (int fadeValue = 255 ; fadeValue >
     analogWrite(ledPin, fadeValue);
     delay(30);
   }
}
```

Driving LOTS of LEDs

- Arduino only has 14 digital I/O pins
  - You could connect multiple LEDs to each pin
  - Web tools like ledcalculator.net
  - Remember current limits!

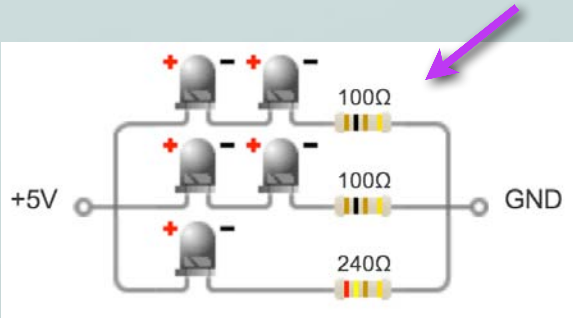
  - ***This circuit has a problem!!!***

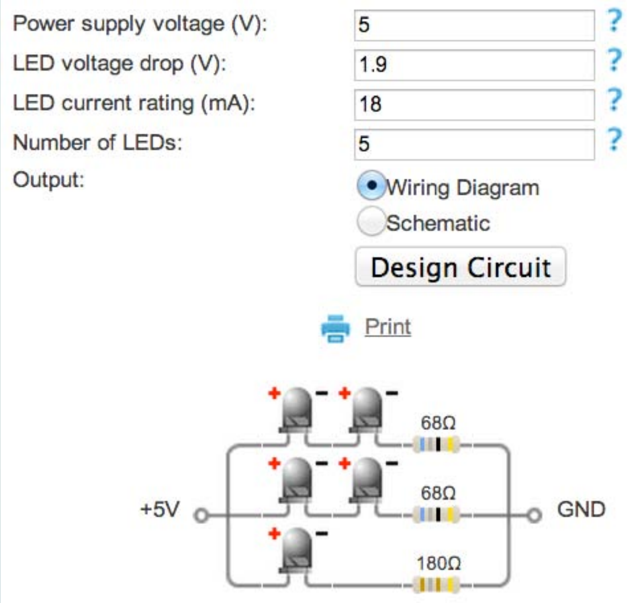


Power supply voltage (V):	5
LED voltage drop (V):	1.9
LED current rating (mA):	18
Number of LEDs:	5
Output:	● Wiring Diagram ○ Schematic

Driving LOTS of LEDs

- Arduino digital pins source/sink max of 40mA
  - KCL says 18mA / branch = 54mA
  - 13mA / LED max for 39mA total





Power supply voltage (V):	5
LED voltage drop (V):	1.9
LED current rating (mA):	18
Number of LEDs:	5
Output:	● Wiring Diagram ○ Schematic

Whew! Questions?