LAB 120

Introduction to Arduino and Electronics

Class 1

Giving Credit

This courseware is a mashup of Tod E. Kurt's Bionic Arduino course, taught at Machine Project in LA and Lutz Hamel's Intro to Arduino course taught here at AS220

Class Info

- Thumbdrive is being passed around, with:
  - PDF version of these notes
  - Arduino software for Mac OS X & Windows
  - Source code ("sketches") used in class
  - Copy files off, then pass thumbdrive around
- Tuesday classes: 2.5 - 3 hours
  - with some review at the beginning

What's for Today

- Introduction to Arduino
- Setting up your Arduino Environment
- Your first Arduino sketch
- Basic digital input and output
- Basic digital sensor inputs
- Making LEDs glow and blink on command
- How to read buttons & switches

Class Kit Manifest

- Arduino Duemilanove board
- Solderless breadboard
- USB cable
- 5 Red LEDs (large, clear)
- 1 RGB LED (diffuse, com. anode)
- two tactile switches
- 50K Potentiometer
- resistors:
  - 5 x 220 ohm (red-red-brown)
  - 5 x 500 ohm (green-black-brown)
  - 5 x 1k (brown-black-red)
  - 5 x10k (brown-black-orange)
  - 5 x1M (brown-black-green)
- capacitors
  - 3 x 10uF
  - 3 x 220uF
- phototransistor (small,clear)
- LM386 amp chip
- H-Bridge Chip
- TIP120 Transistor
- Motor
- Hitec HS311 Servo
- Speaker
- Microphone
- IR LED
- IR Photo Transistor
- Force Sensitive Resistor
- 4 colors of hookup wire
- 9V battery and case
- rubber bands

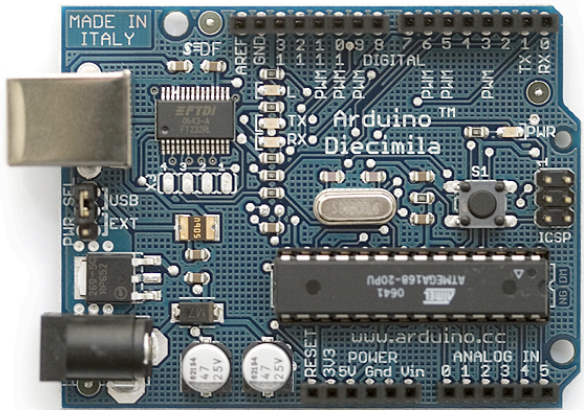
A Word on Safety

- Electronics can hurt you
  - Lead in some of the parts
  - Wash up afterwards
- You can hurt electronics
  - Static-sensitive: don't shuffle your feet & touch
  - Wires only bend so much

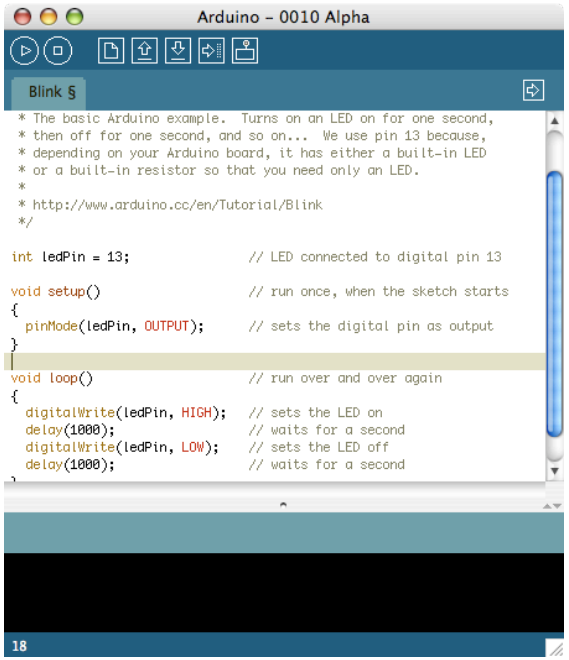
What is Arduino?

The word "Arduino" can mean 3 things

A physical piece of hardware



A programming environment



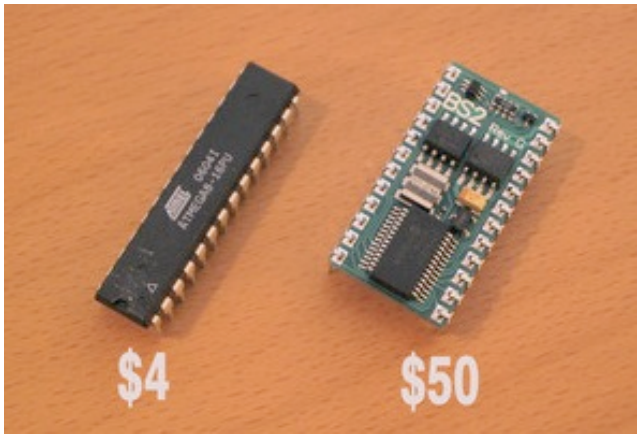
A community & philosophy

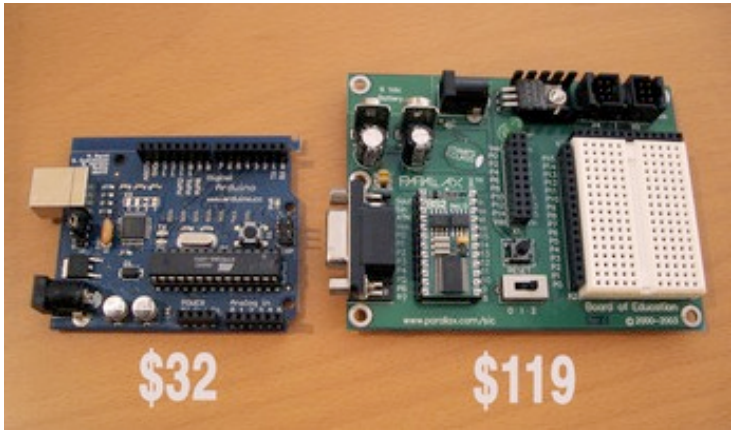
Arduino
Philosophy & Community

- Open Source Physical Computing Platform
  - "open source hardware"
  - open source: free to inspect & modify
  - physical computing. er, what? ubiquitous computing, pervasive computing, ambient intelligence, calm computing, everyware, spimes, blogjects, smart objects...
- Community-built
  - Examples wiki (the "playground") editable by anyone
  - Forums with lots of helpful people

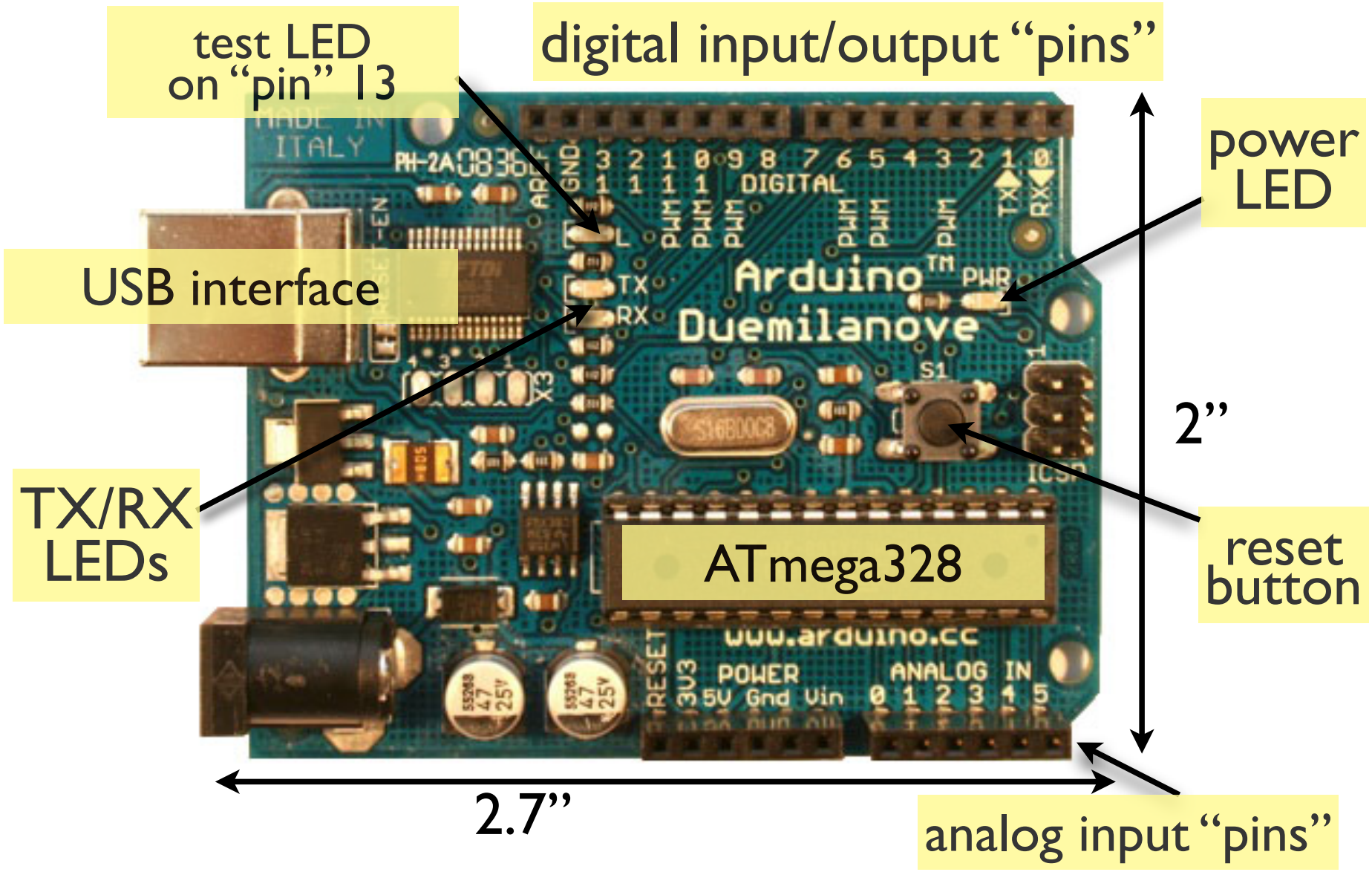
Arduino Hardware

- Similar to Basic Stamp (if you know of it)
  - but cheaper, faster, & open
- Uses AVR ATmega328 microcontroller chip
  - chip was designed to be used with C language





Arduino Duemilanove Board

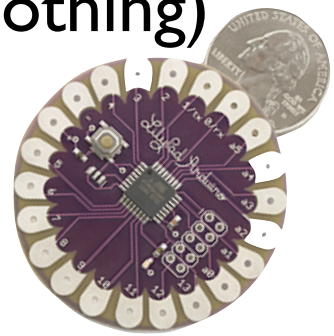


Arduino Capabilities

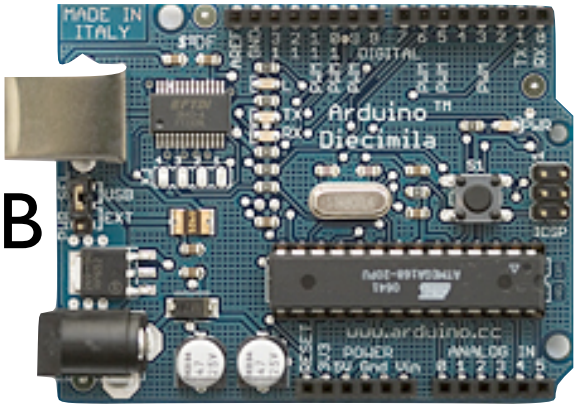
- 32 kBytes of Flash program memory
- 2 kByte of RAM
- 16 MHz (Apple II: 1 MHz)
- Inputs and Outputs
  - 13 digital input/output pins
  - 5 analog input pins
  - 6 analog output pins*
- Completely stand-alone: doesn't need a computer once programmed

Arduino Hardware Variety

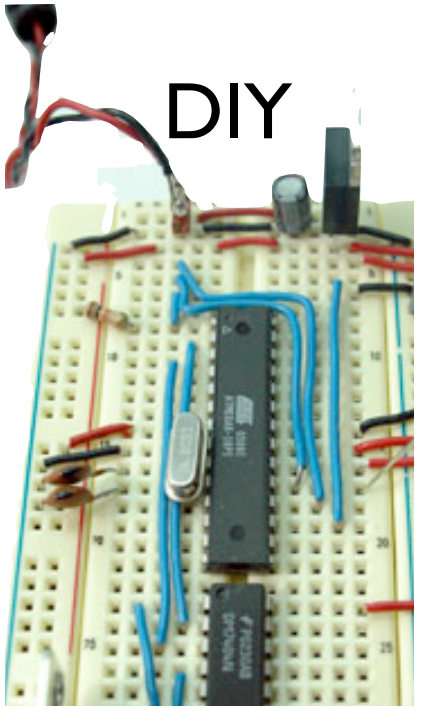
LilyPad
(for clothing)

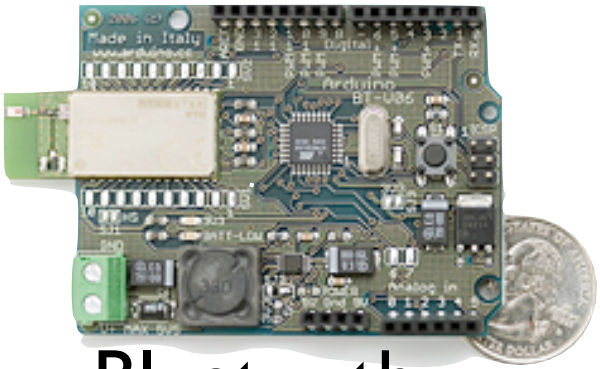


USB



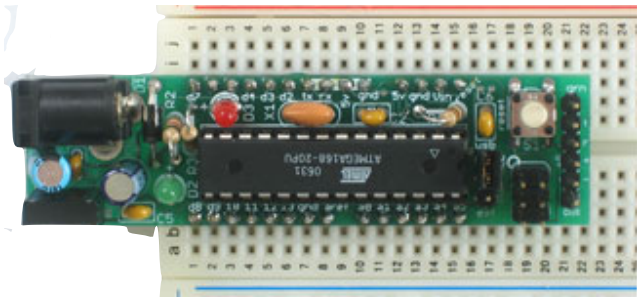
DIY



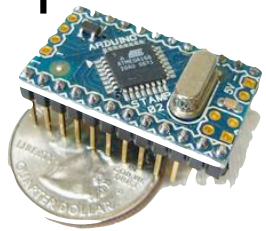


Bluetooth

Boarduino Kit



"Stamp"-sized



many different variations to suite your needs

Arduino Terminology

"*sketch*" – a program you write to run on an Arduino board

"*pin*" – an input or output connected to something.

e.g. output to an LED, input from a knob.

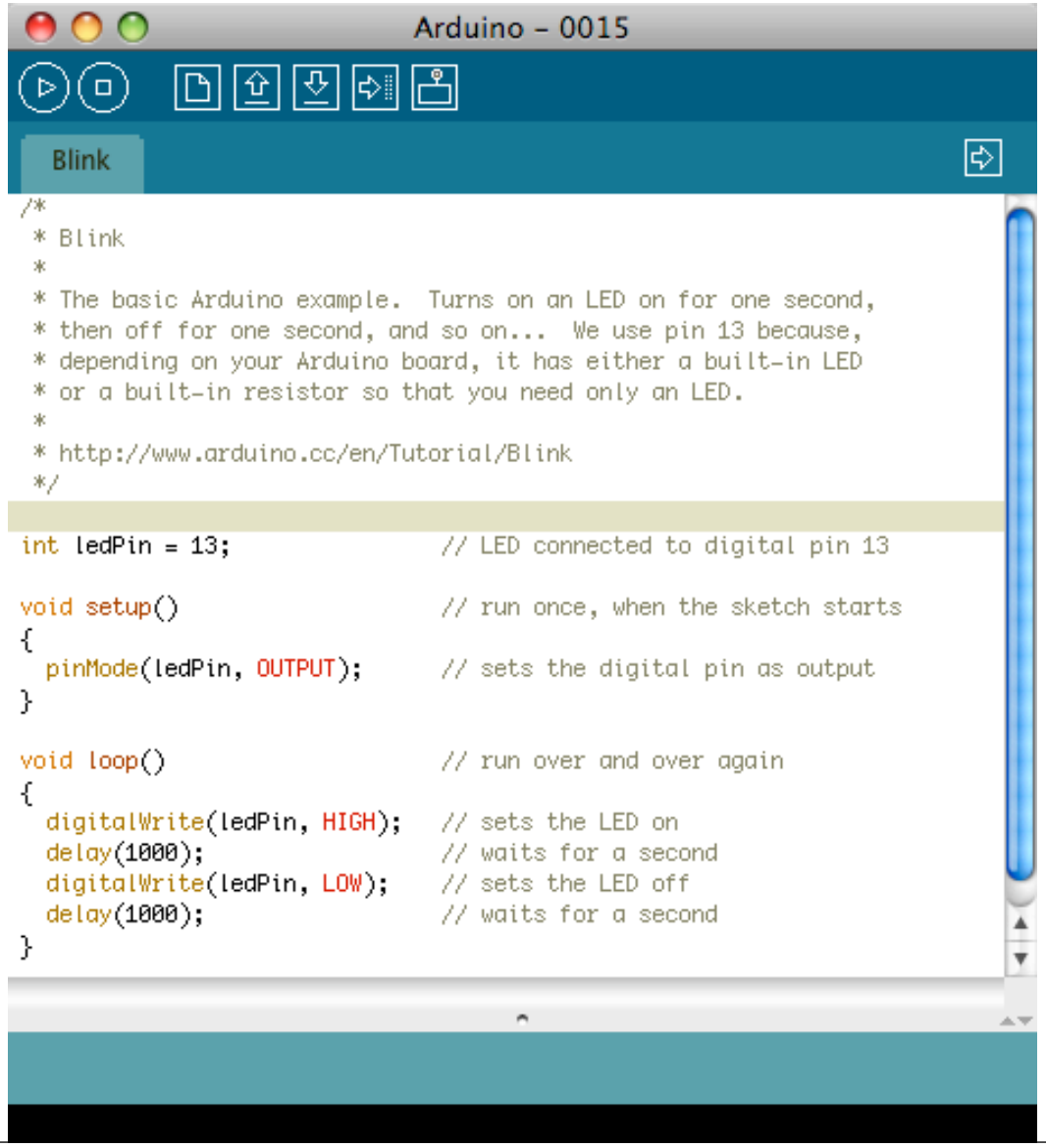
"*digital*" – value is either HIGH or LOW.

(aka on/off, one/zero) e.g. switch state

"*analog*" – value ranges, usually from 0-255.

e.g. LED brightness, motor speed, etc.

Arduino Software



- Like a text editor
- View/write/edit sketches
- But then you program them into hardware

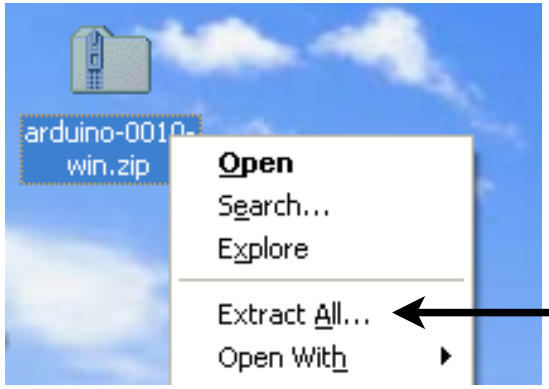
Installing Arduino

The Steps

1. Get the Arduino software & unzip it
2. Plug in Arduino board
3. Install the driver
4. Reboot
5. Run the Arduino program
6. Tell Arduino (program) about Arduino (board)

Getting and Unpacking

- On the thumbdrives
  - "arduino-0016-win.zip" for Windows
  - "arduino-0016-mac.zip" for Mac OS X
- Unzip the zip file. Double-click on Mac



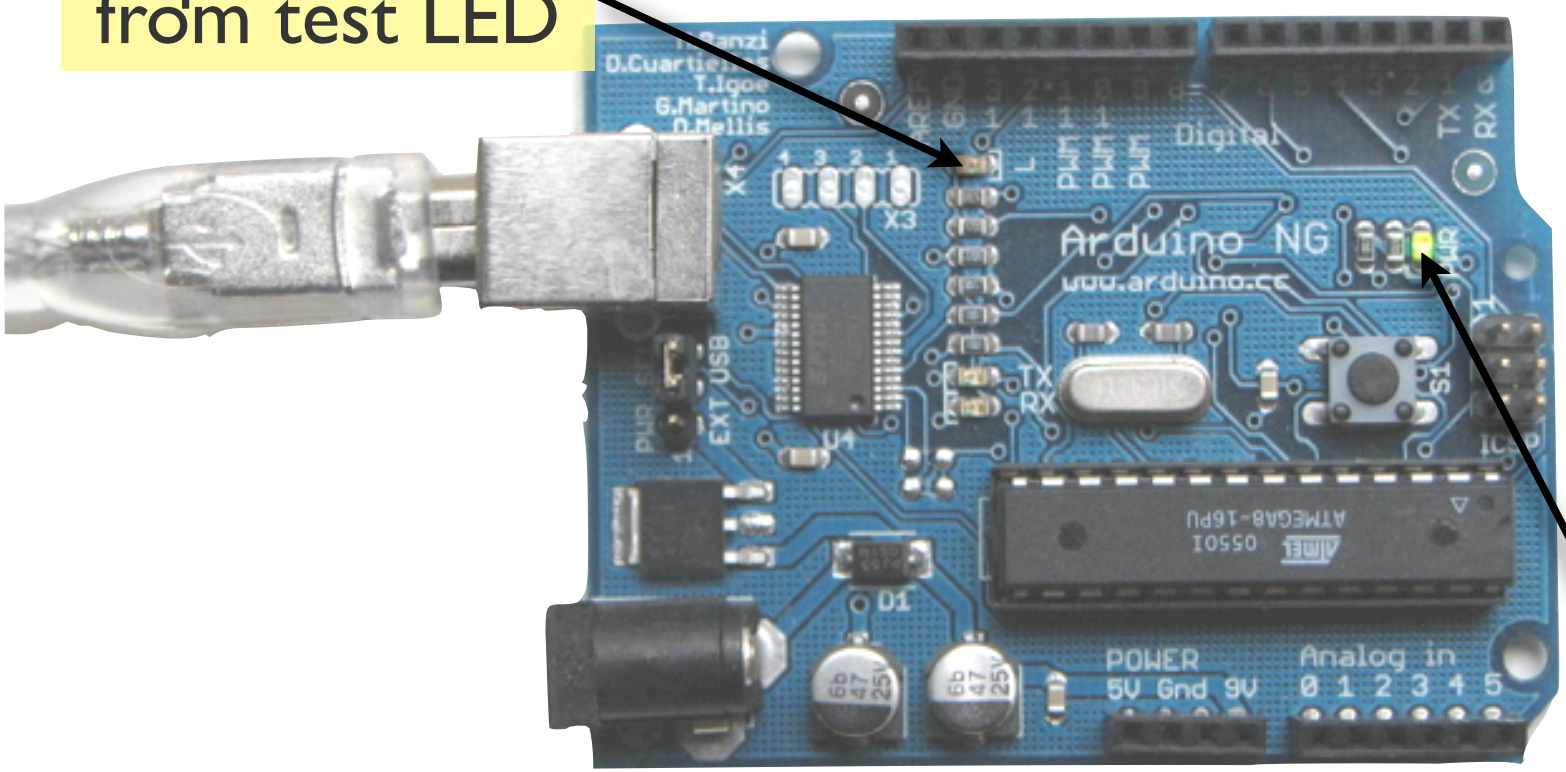
On Windows, right-click

Use "Extract All..."

- Find the "drivers" directory inside

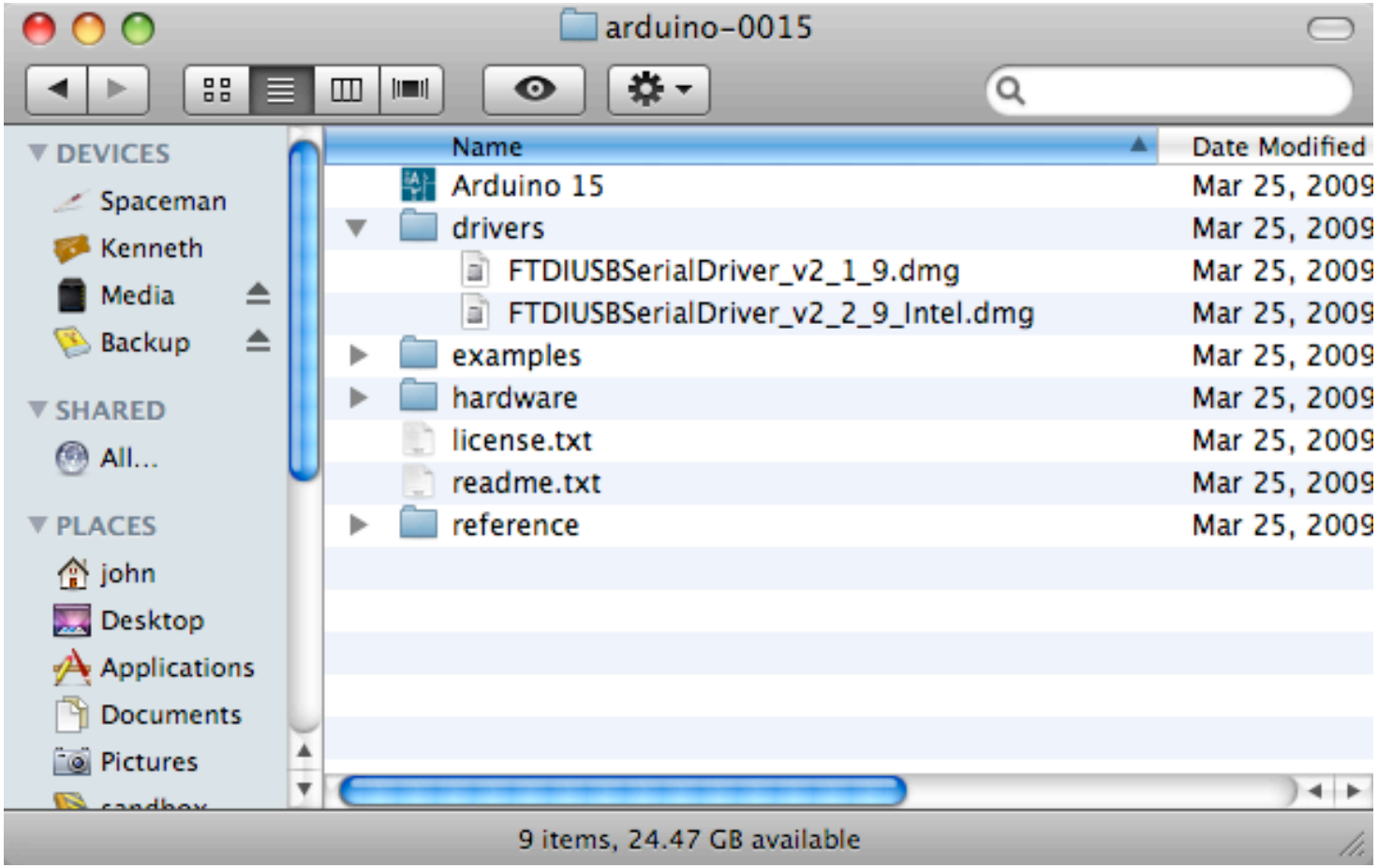
Plug in Arduino board

quick blink from test LED



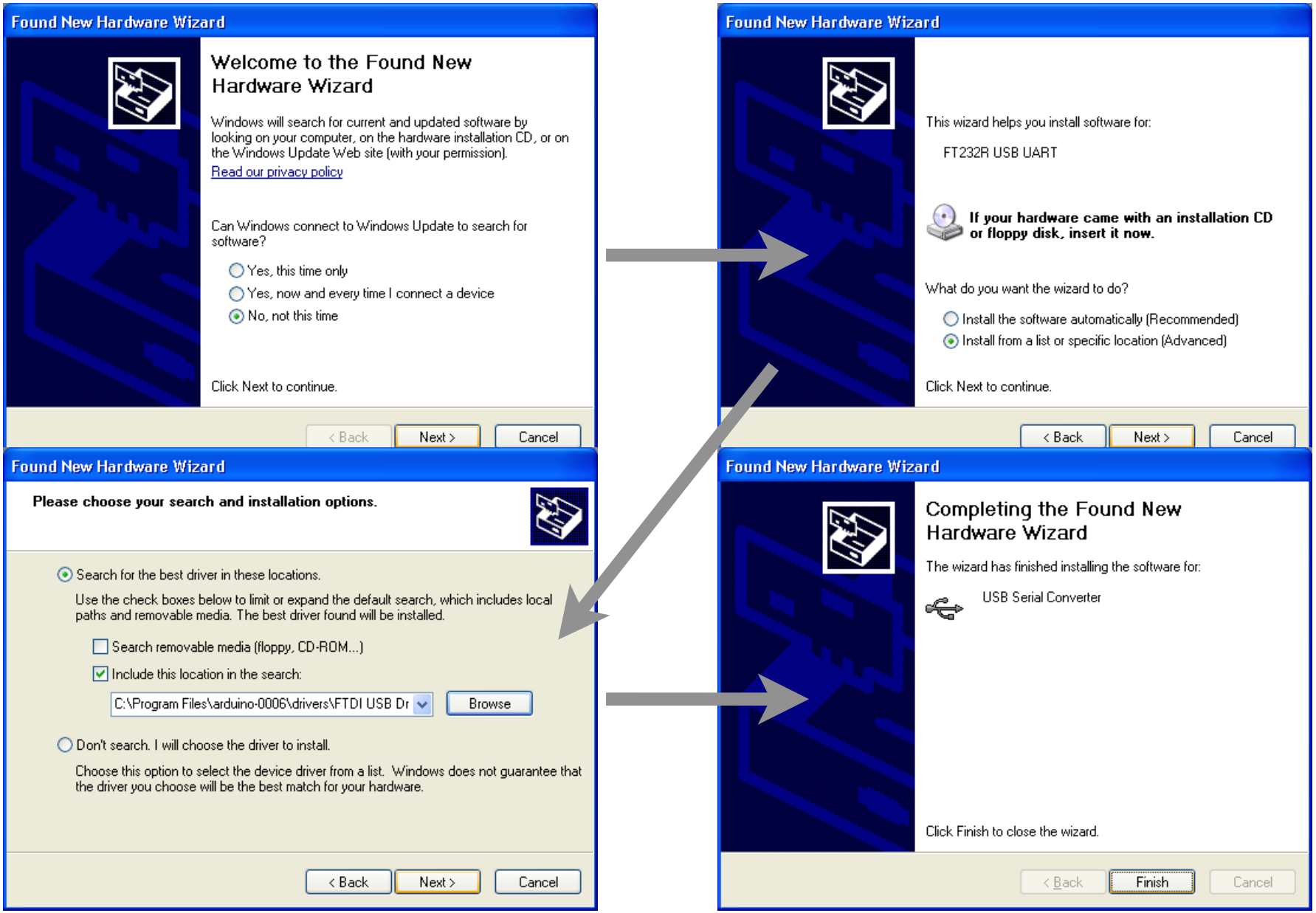
Power LED should stay on

Mac Driver Install

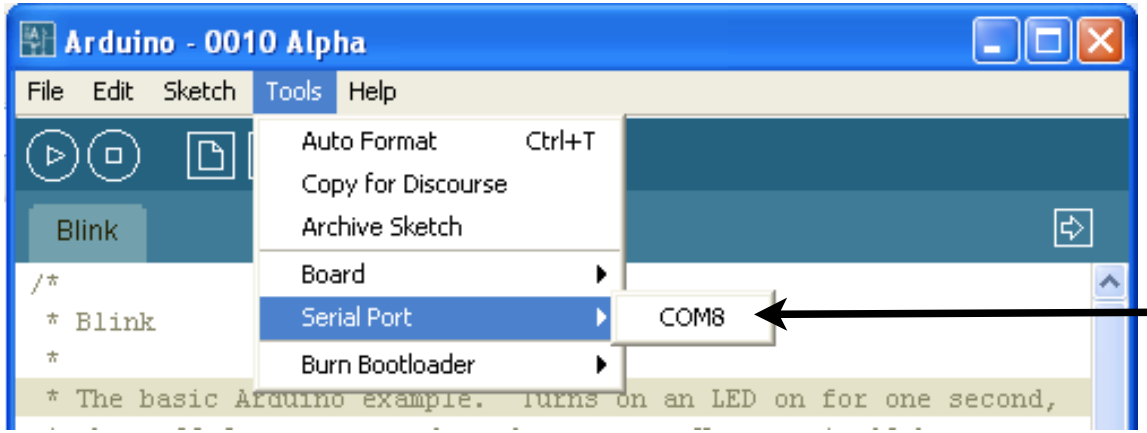


- v2_1_9 for PPC Macs
- v2_2_9 for Intel Macs

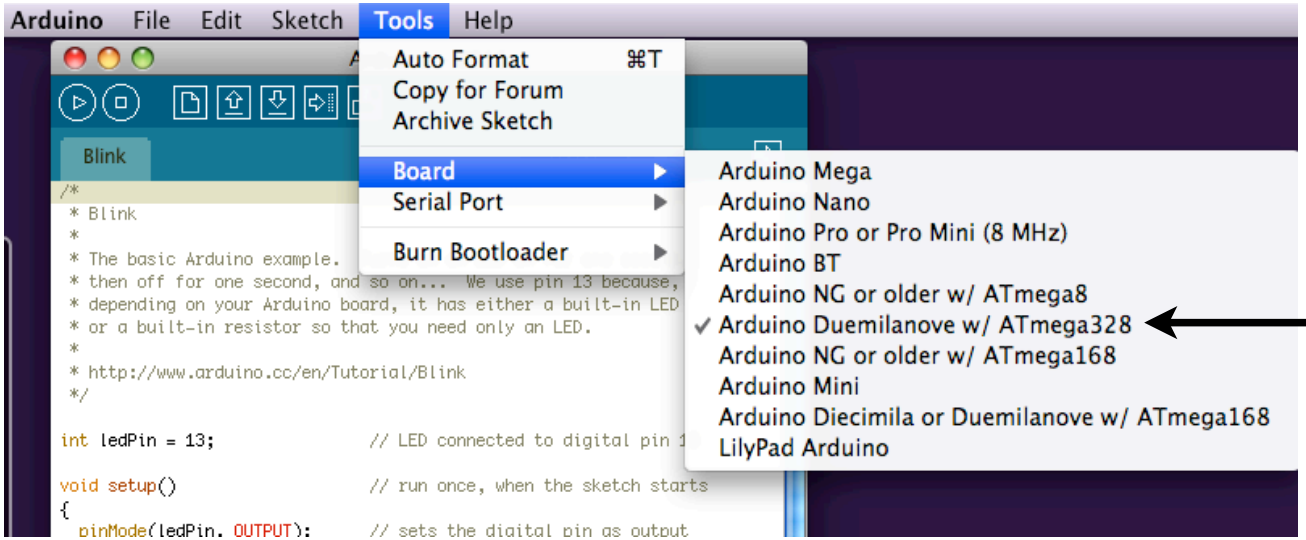
Windows Driver Install



Selecting Location & Type

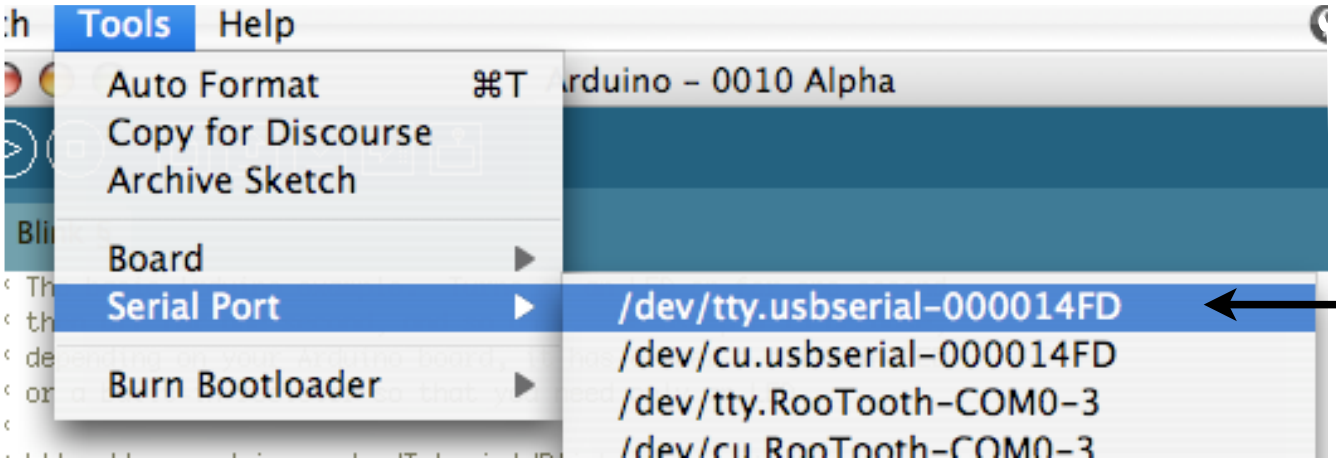


usually highest-numbered port

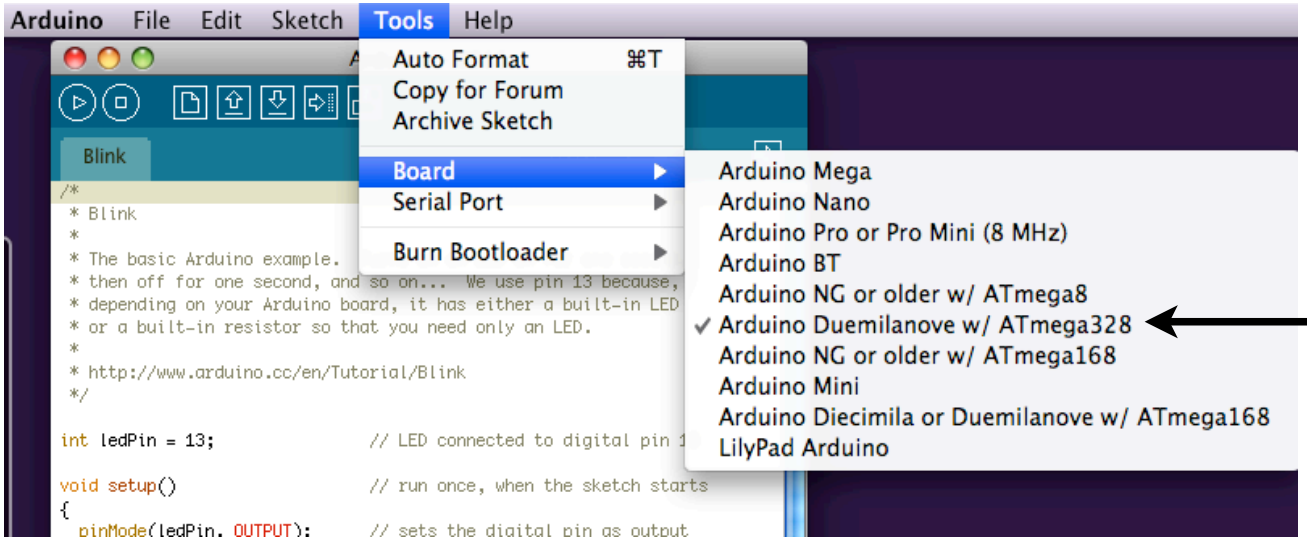


pick "Duemilanova with 328"

Selecting Location & Type



starts with *tty.usbserial-*



pick "Duemilanova with 328"

Arduino Software

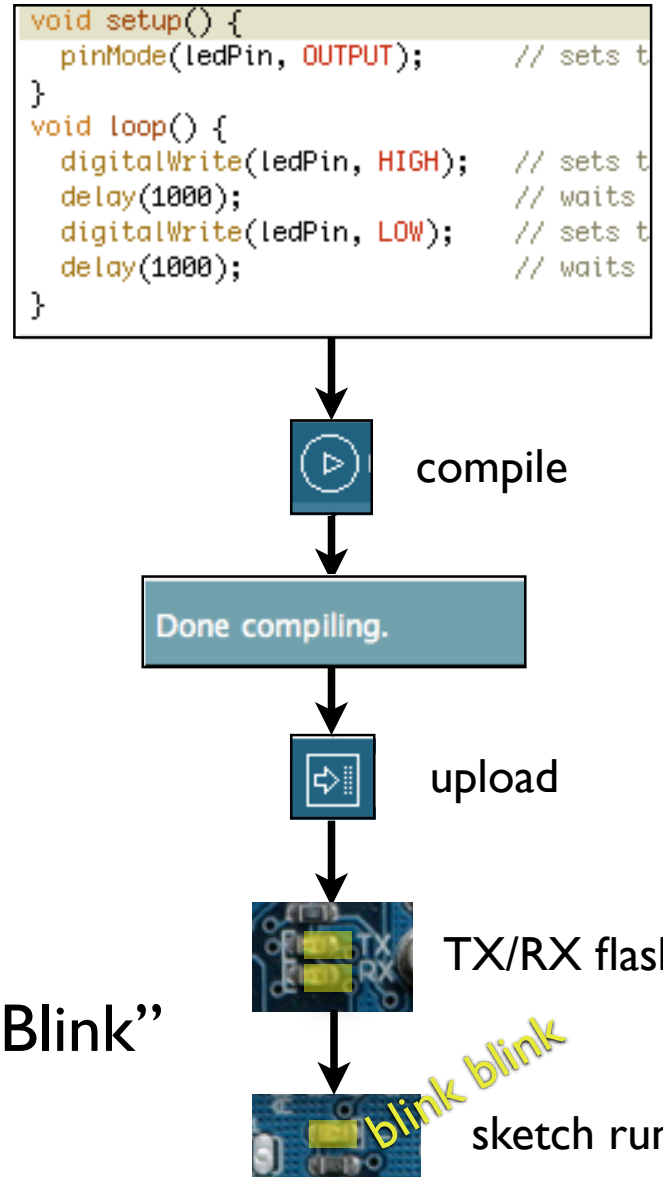
compile (*verify*)

upload to board

status area

Using Arduino

- Write your sketch
- Press Compile button
  (to check for errors)
- Press Upload button to program
  Arduino board with your sketch



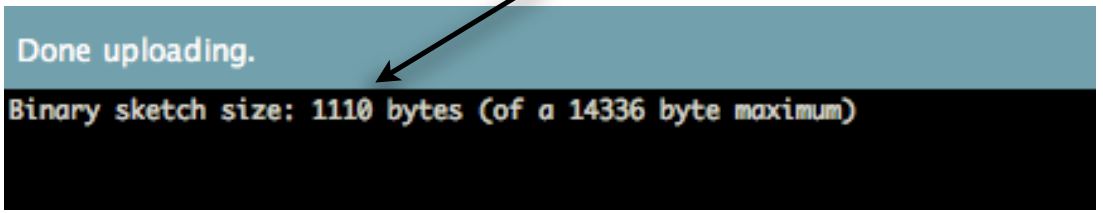
Try it out with the "Blink" sketch!

Load "File/Sketchbook/Examples/Digital/Blink"

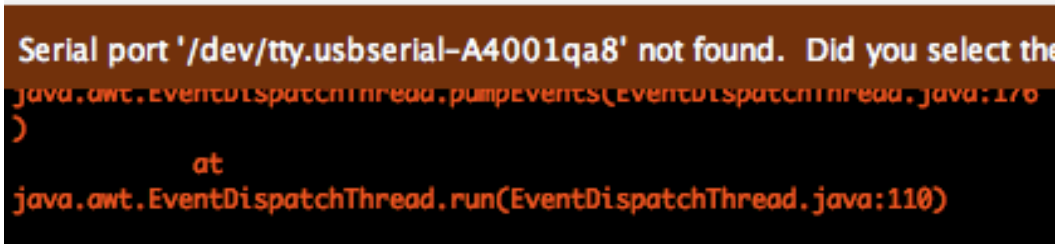
Status Messages

Uploading worked

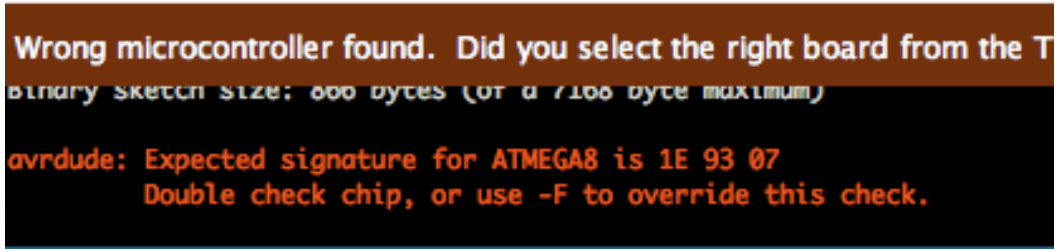
Size depends on complexity of your sketch



Wrong serial port selected



Wrong board selected



nerdy cryptic error messages

Troubleshooting

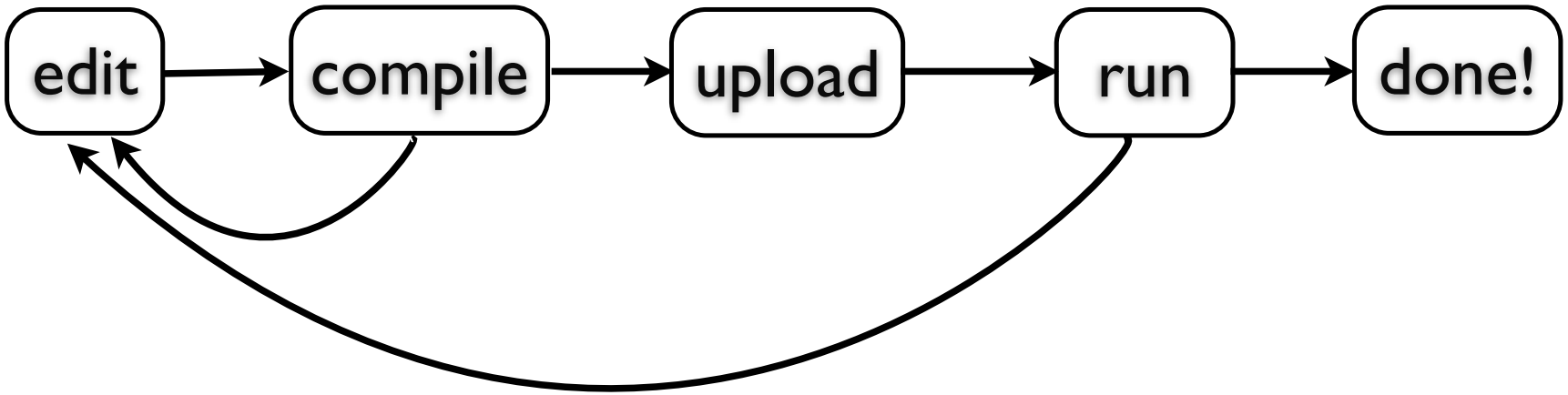
- Most common problem is incorrect serial port setting
- If you ever have any “weird” errors from the Arduino environment, just try again.
- The red text at the bottom is debugging output in case there may be a problem
- Status area shows summary of what’s wrong

I made an LED blink, so what?

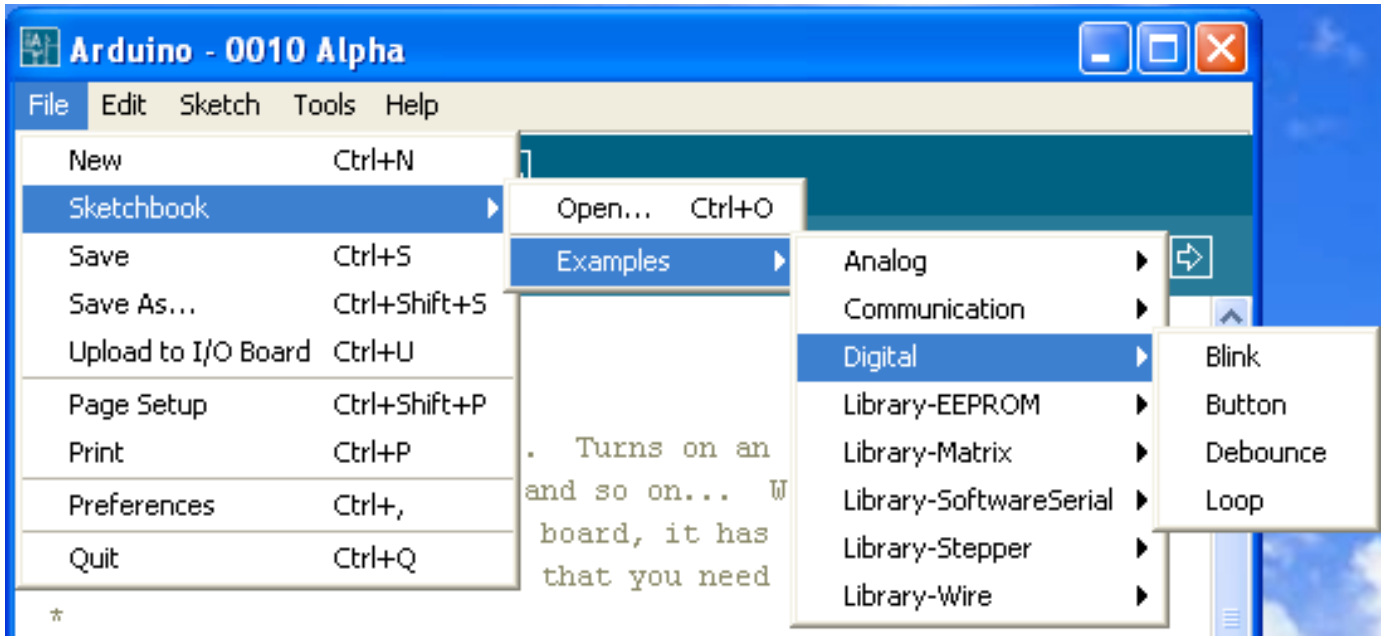
- Most actuators are switched on and off with a digital output
- The `digitalWrite()` command is the software portion of being able to control just about anything
- LEDs are easy, motors come next week
- Arduino has up to 13 digital outputs, and you easily can add more with helper chips

Development Cycle

- Make as many changes as you want
- Not like most web programming: edit → run
- Edit → compile → upload → run

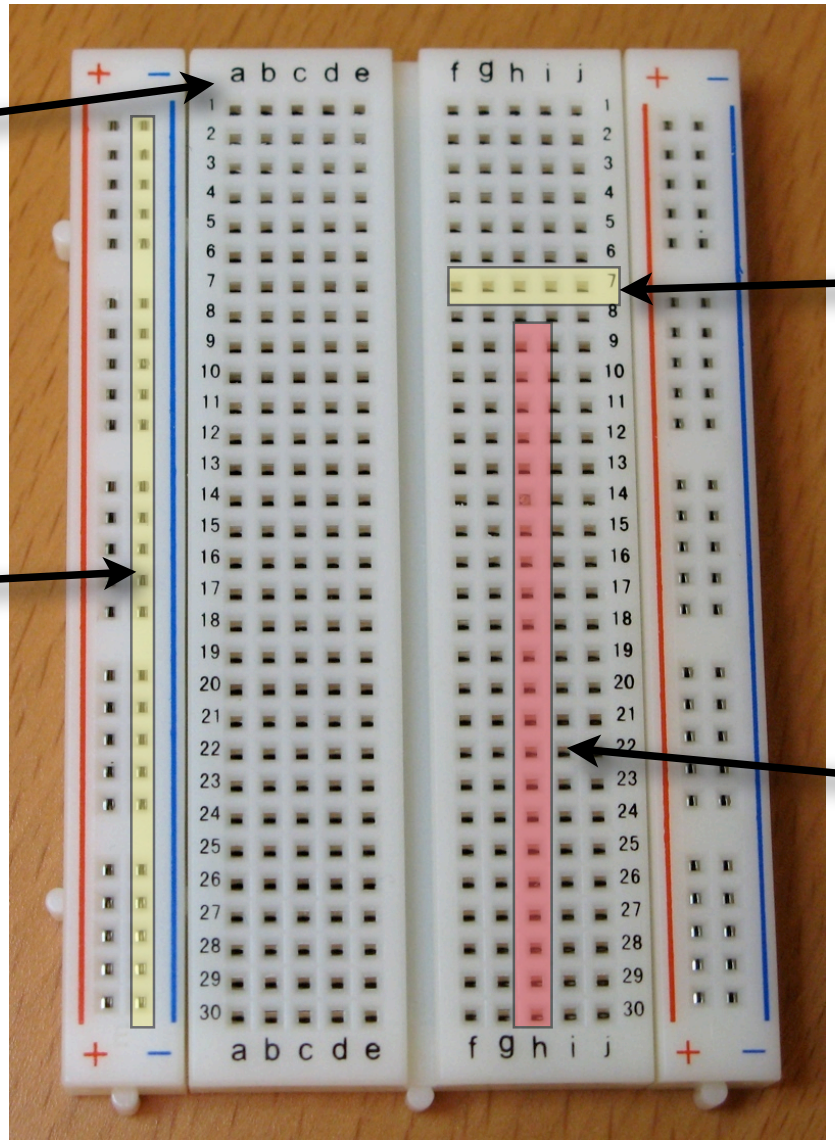


Lots of Built-in Examples



And more here:
http://www.arduino.cc/en/Tutorial/HomePage

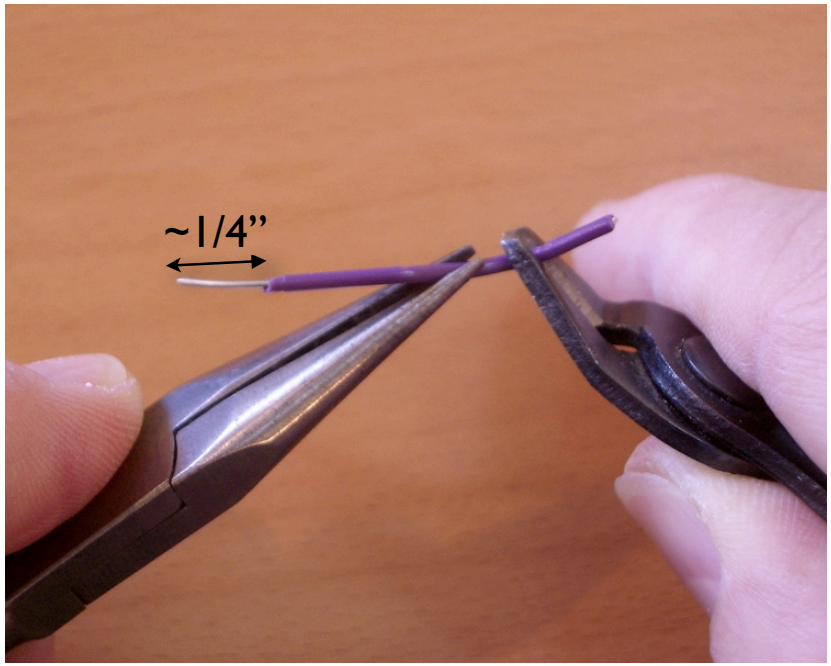
Solderless Breadboards



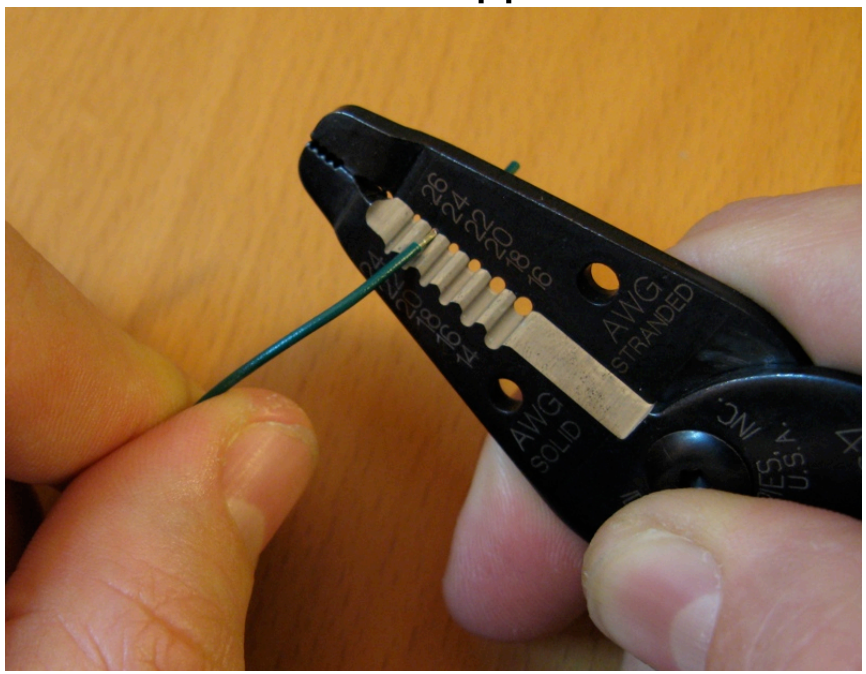
Useful Tools

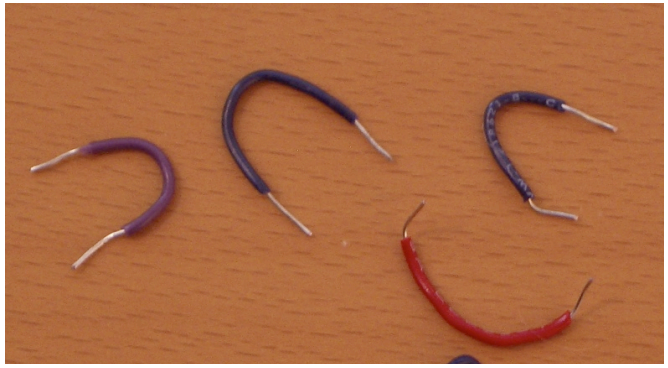
Making Jumper Wires

pliers & cutter



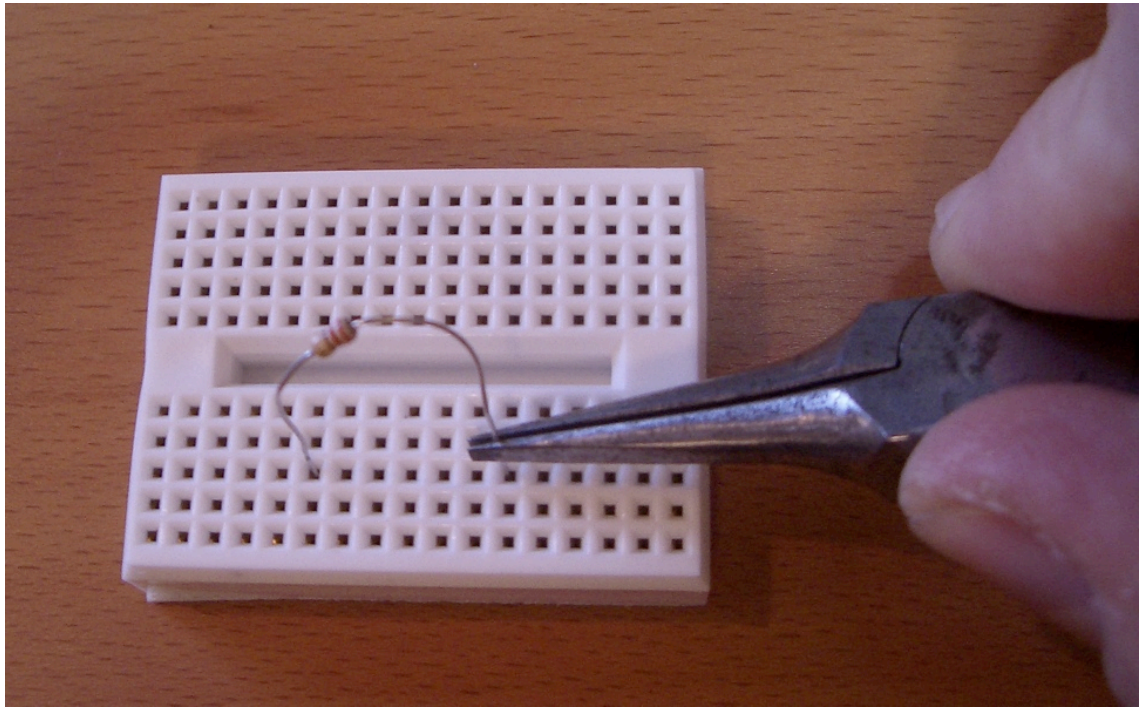
wire stripper



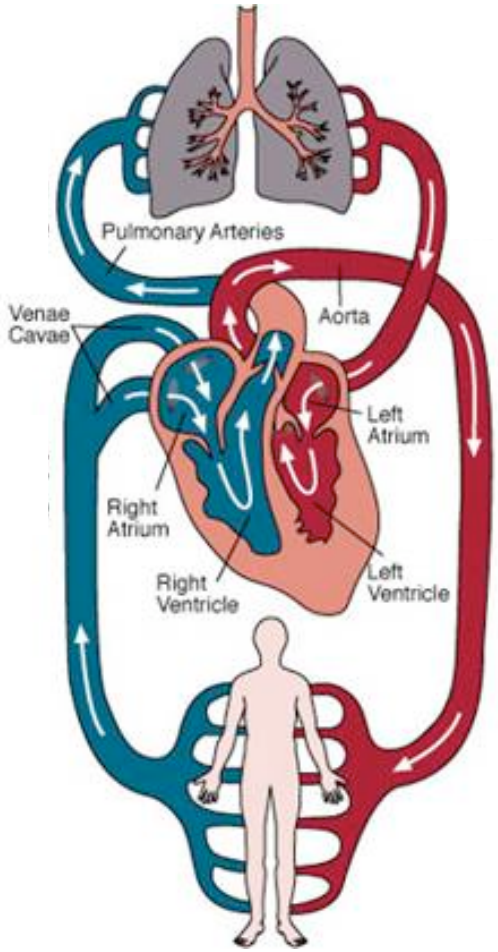


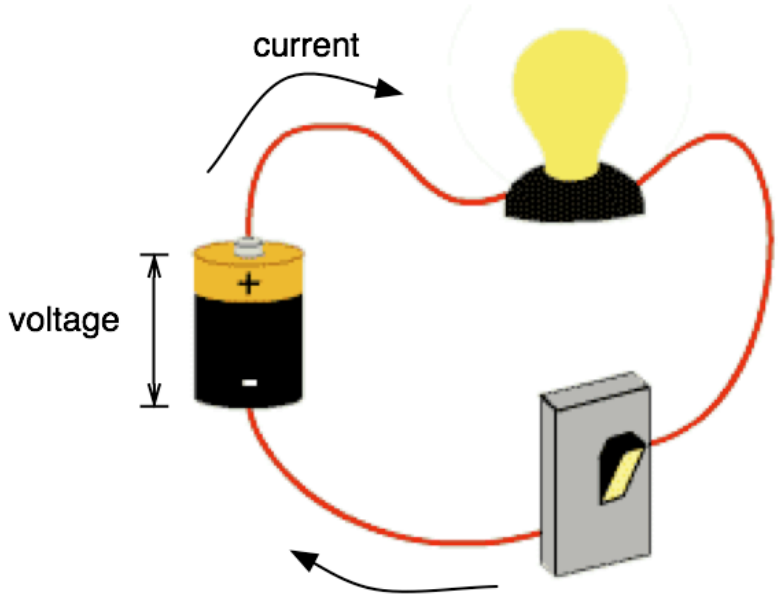
Using Solderless Breadboards

Using needle nose pliers can help push wires & components into holes



Making Circuits

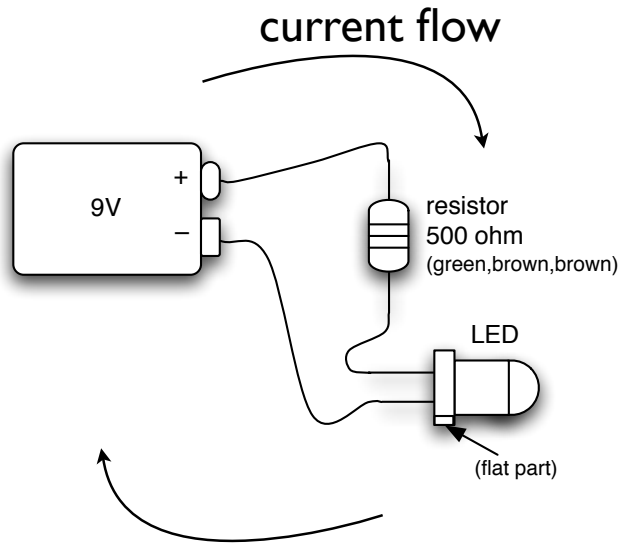


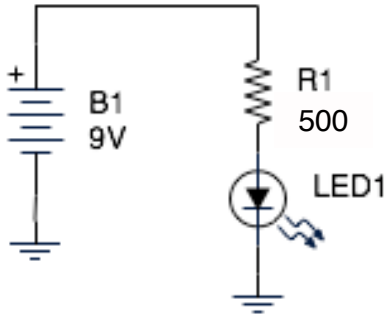


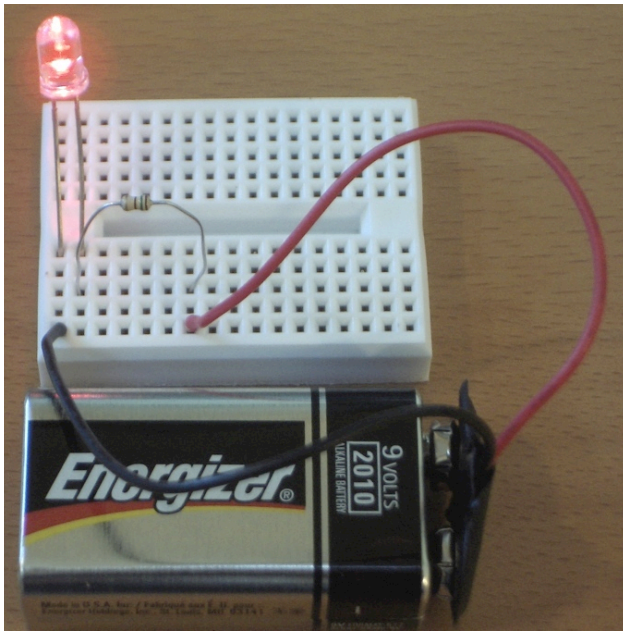
heart pumps, blood flows

voltage pushes, current flows

Example: LED flashlight



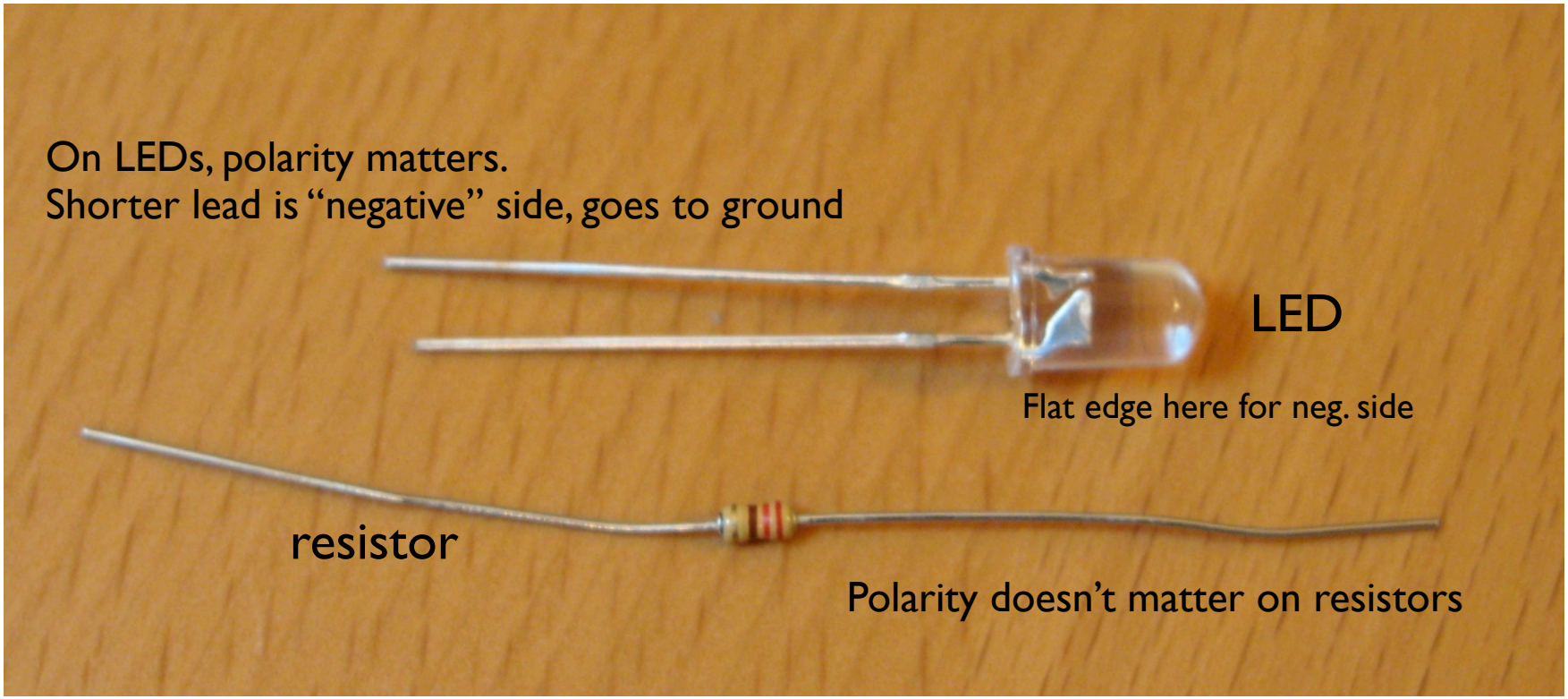




wiring diagram schematic wiring it up

Electricity flows in a loop. Can stop flow by breaking the loop

LEDs & Resistors



Arduino "Language"

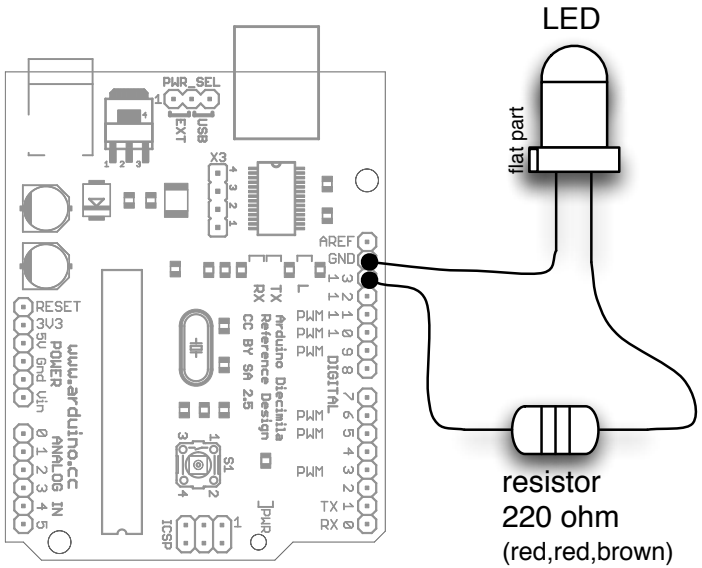
- Language is standard C (but made easy)
- Lots of useful functions
  - `pinMode()` – set a pin as input or output
  - `digitalWrite()` – set a digital pin high/low
  - `digitalRead()` – read a digital pin's state
  - `analogRead()` – read an analog pin
  - `analogWrite()` – write an "analog" value
  - `delay()` – wait an amount of time
  - `millis()` – get the current time
- And many others. And libraries add more.

Sketch structure

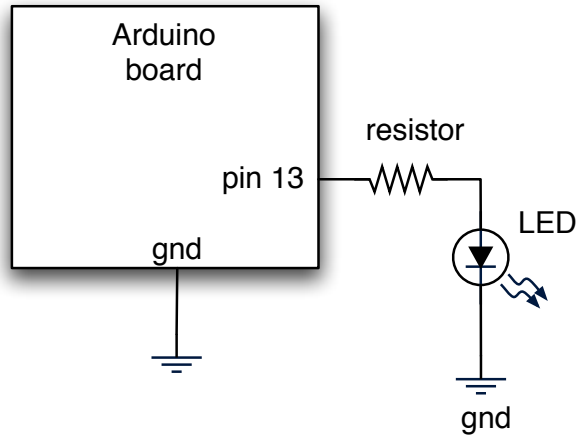
- Declare variables at top
- Initialize
  - `setup()` – run once at beginning, set pins
- Running
  - `loop()` – run repeatedly, after `setup()`

The Circuit for LED Blink

"hello world" of microcontrollers



wiring diagram



schematic

Arduino Duemilanove board has this circuit built-in

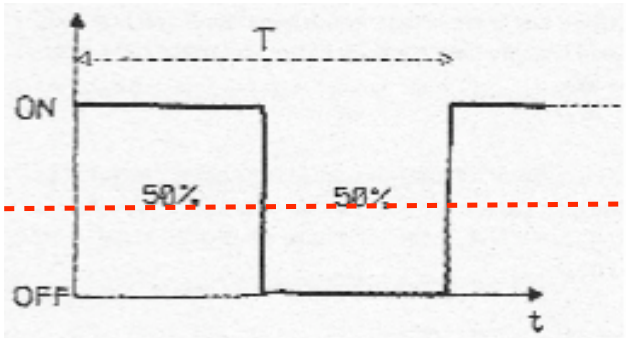
To turn on LED use `digitalWrite(13,HIGH)`

PWM Signals

- Pulse Width Modulated (PWM) Signals
- µCs cannot generate analog output, but we can fake it by creating digital signals with different "duty cycles" - signals with different pulse widths.
- To the analog world the different duty cycles create different effective voltages

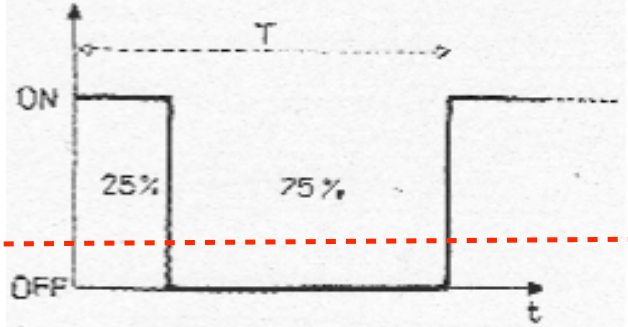
PWM Signals

50% Duty Cycle



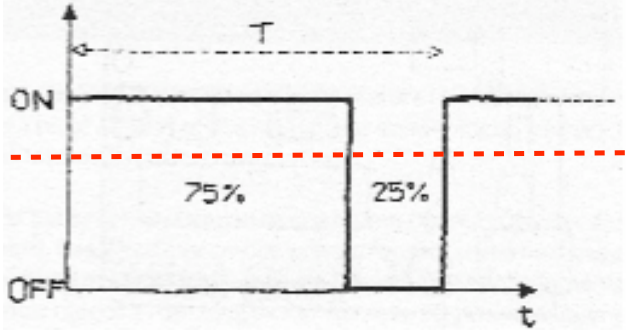
Effective Voltage

25% Duty Cycle



Effective Voltage

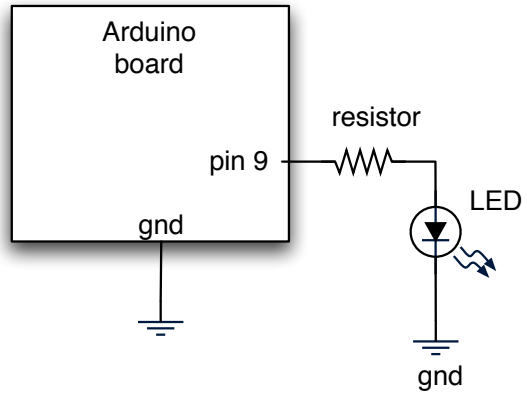
75% Duty Cycle



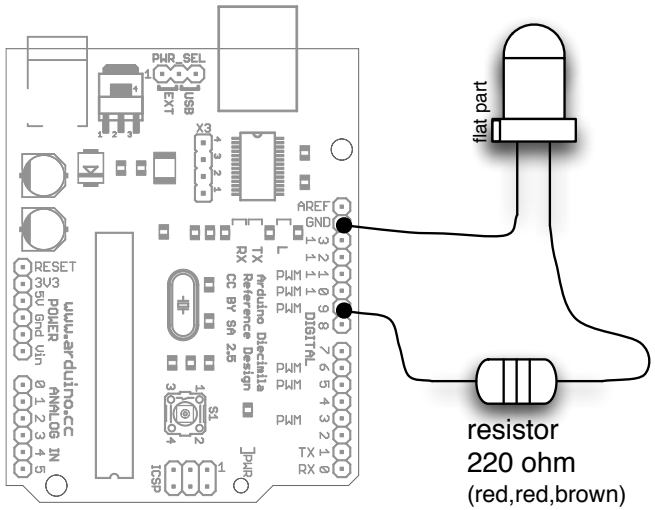
Effective Voltage

Varying LED Brightness

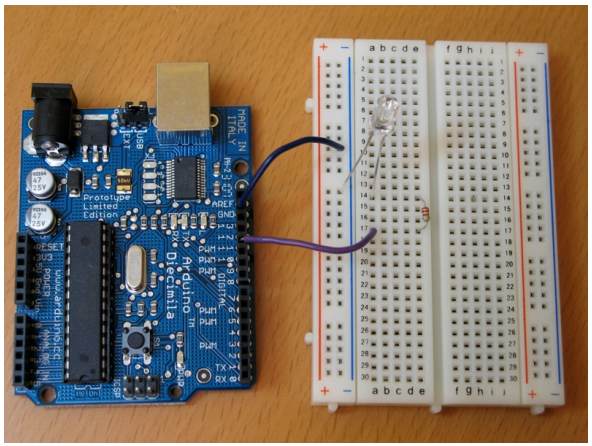
Same circuit as Blink circuit but pin 9 instead of pin 13



schematic



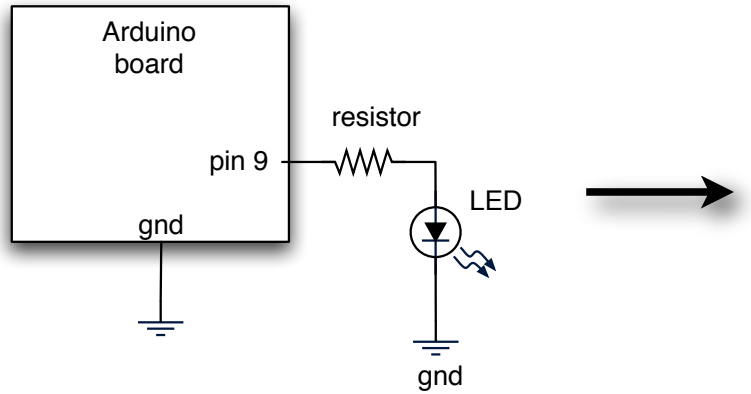
wiring diagram

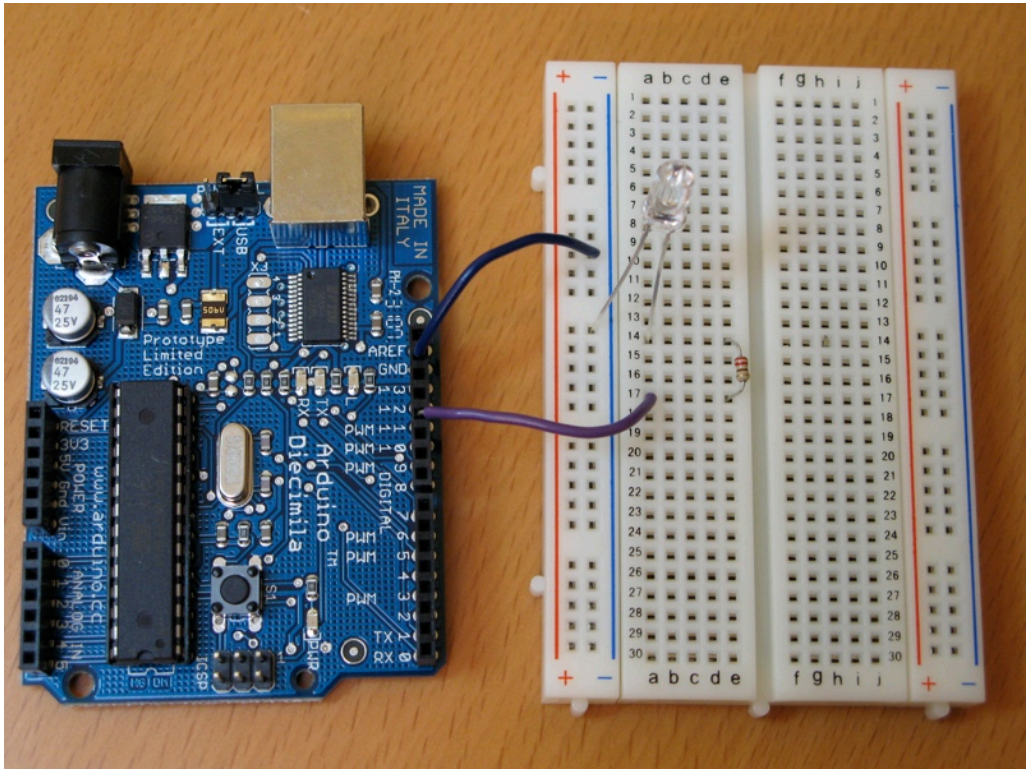


wired up

The PWM pins work with the "`analogWrite(value)`" command
where "value" ranges between 0 and 255.
To turn LED to half-bright, use `analogWrite(9,128)`

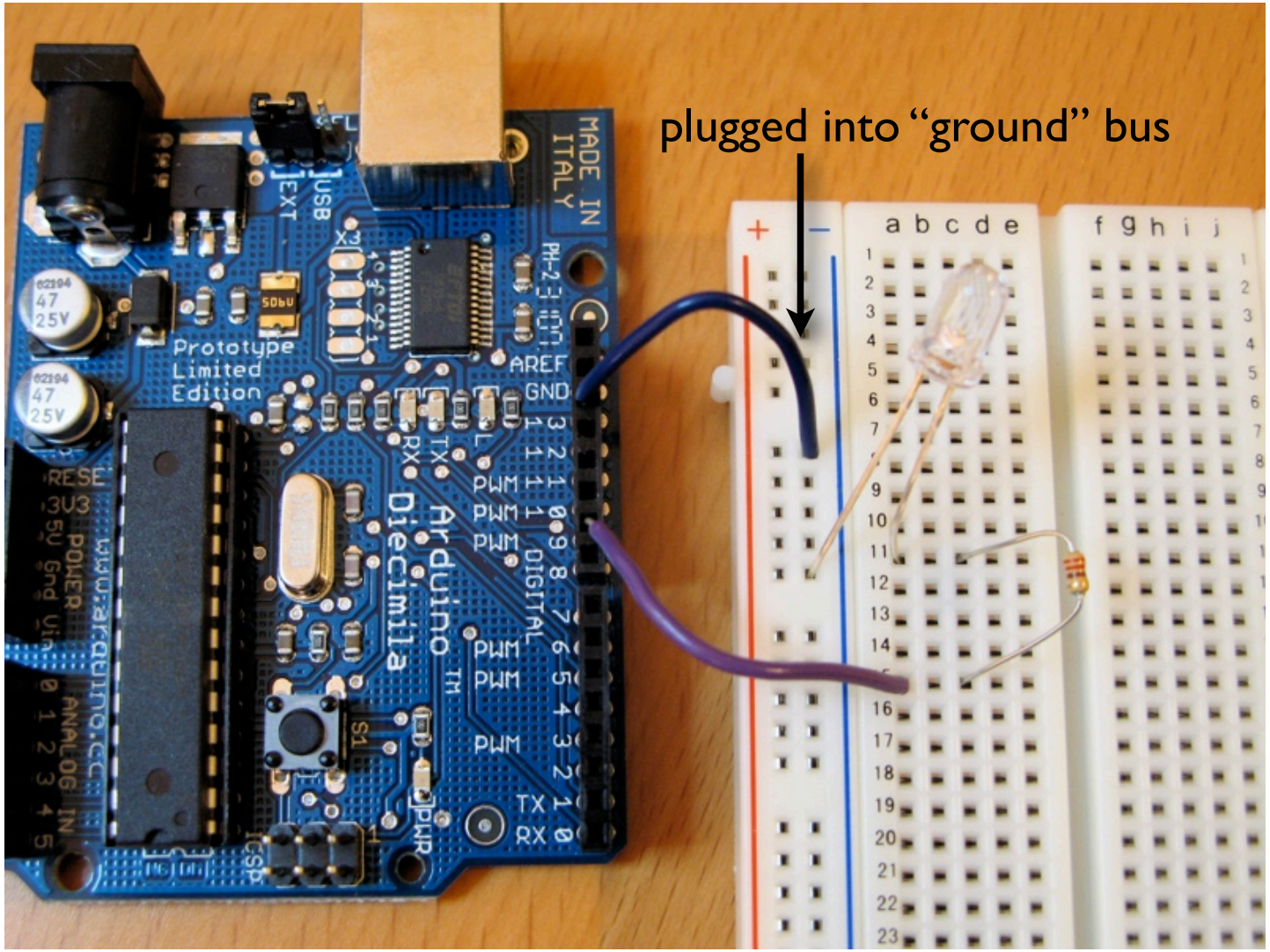
Let's Wire It Up





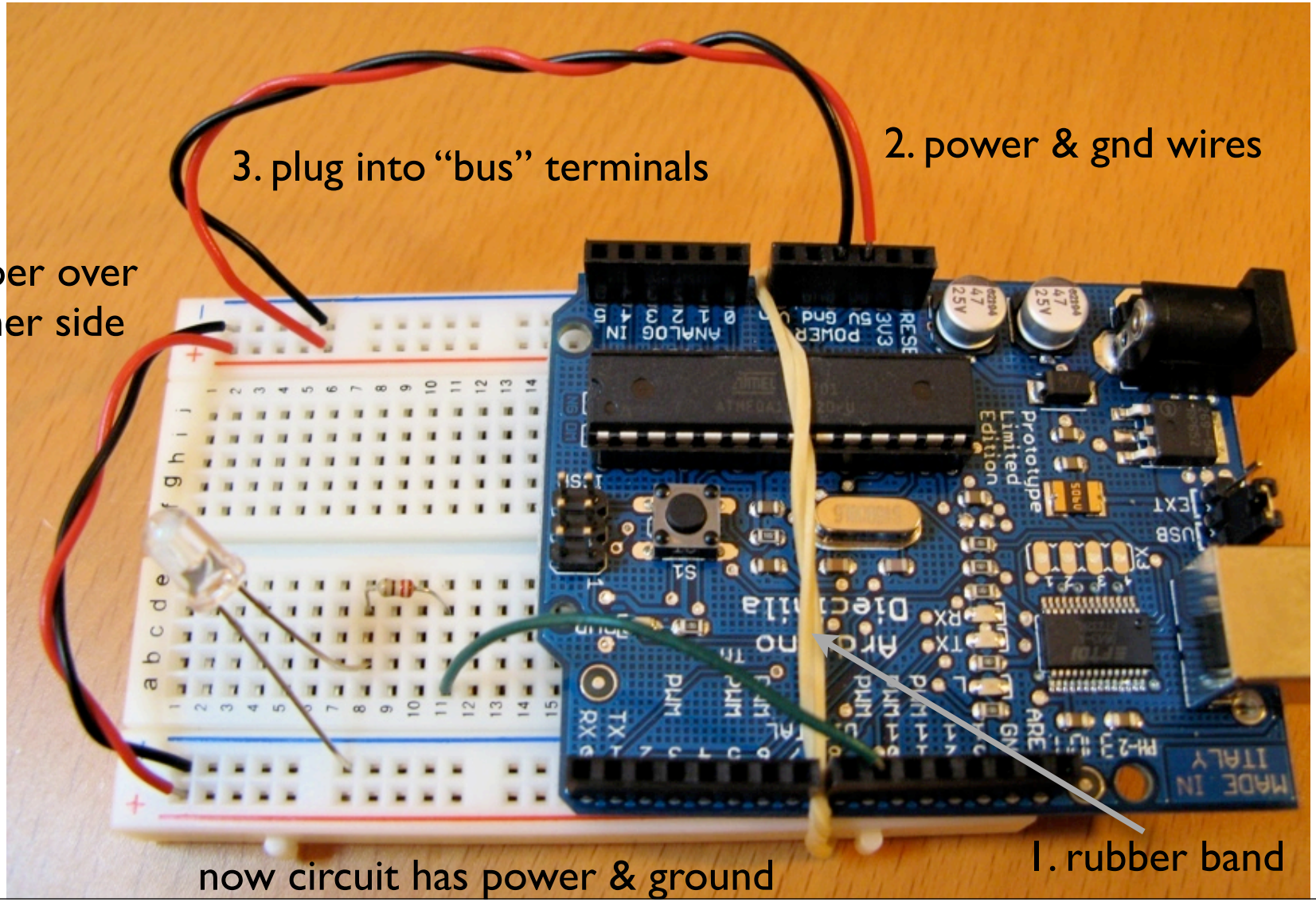
Going from schematic to physical circuit.

All Wired Up


plugged into “ground” bus
MADE IN ITALY
Prototype Limited Edition
Arduino Diecimila
a b c d e
f g h i j

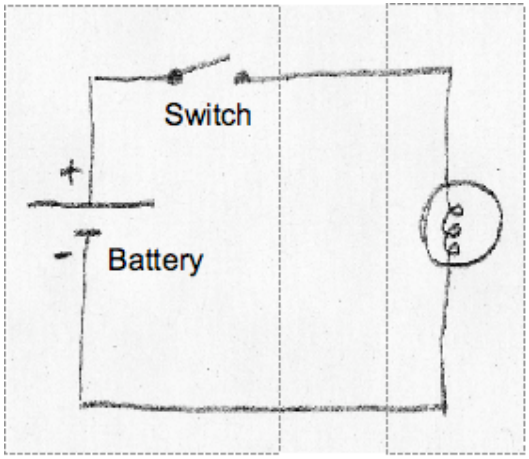
Alternate Way

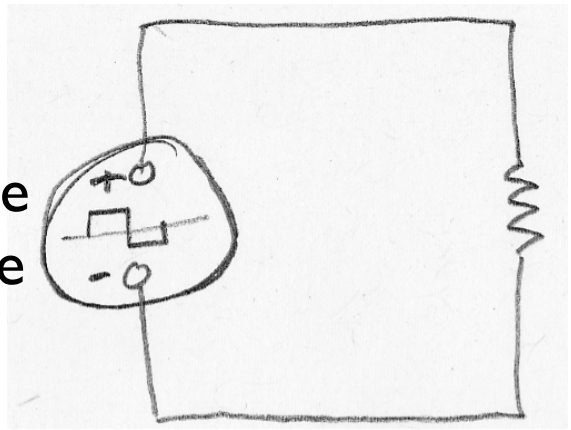
Or, adding a breadboard to Arduino for 1¢



Basic Electronics

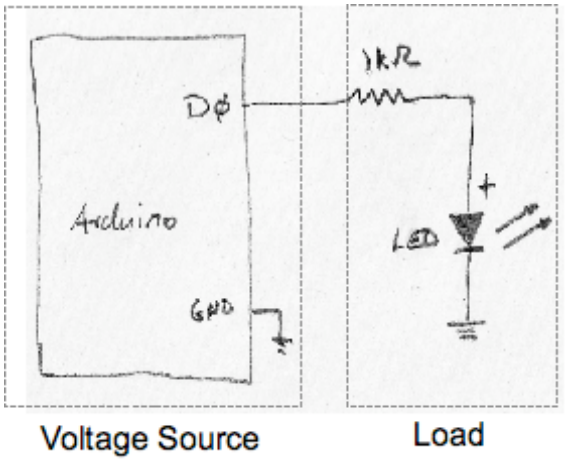
Flash Light





Voltage Source

Load

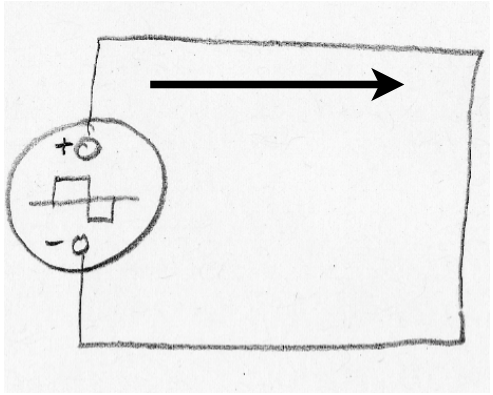


Our Blink Circuit

Basic Electronics

Current = ∞

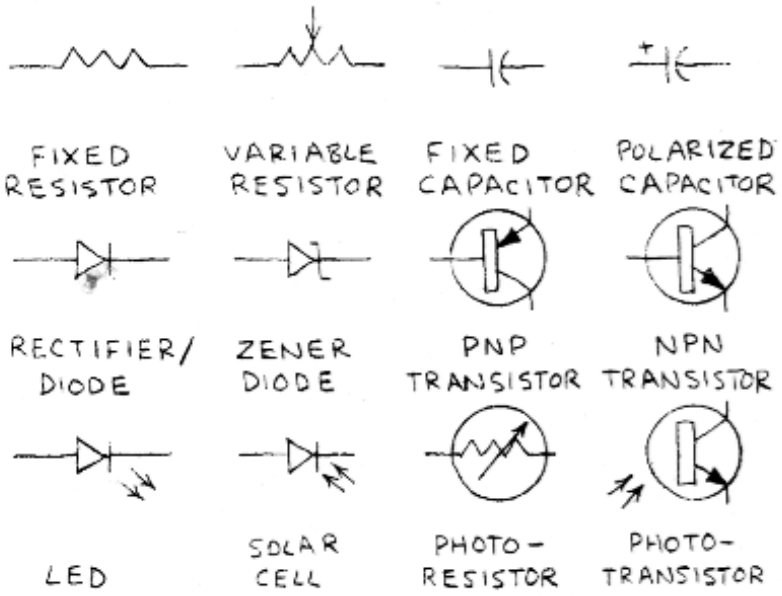
Voltage Source



Load = 0

- The dreaded *short circuit*:
- this is a circuit with a load equal to zero
- this allows "infinite" current to flow from the positive terminal of the voltage source to the negative terminal
- it will break stuff!
- Always check your circuits carefully before applying power
- Never connect an Arduino output pin directly to ground, always use a load resistor

Basic Electronics



Some Electronic Symbols

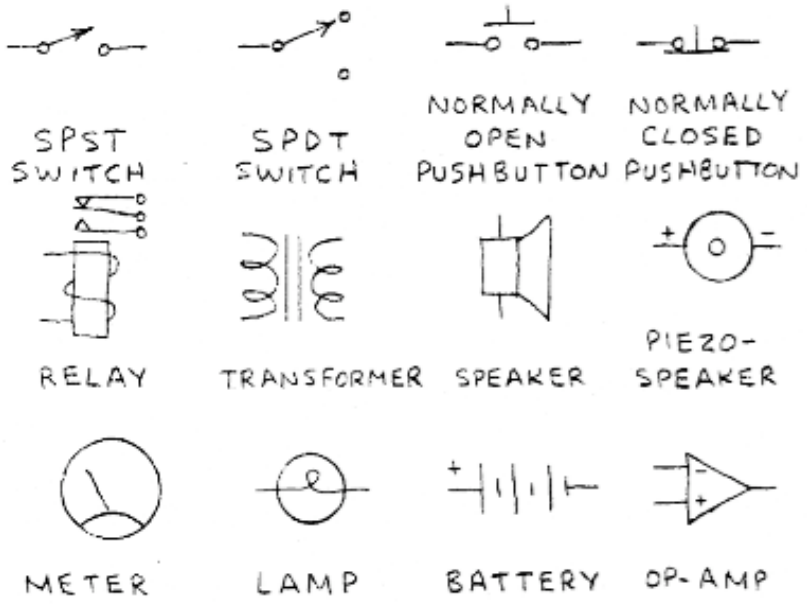


Image source: Engineer's Mini Notebook, Mims III, Master Publishining, 2007.

LED "Fading" Sketch

Load "File/Sketchbook/Examples/Analog/Fading"

note →

```
int value = 0;                              // variable to keep the actu
int ledpin = 9;                             // light connected to digito

void setup()
{
  // nothing for setup
}

void loop()
{
  for(value = 0 ; value <= 255; value+=5) // fade in (from min to max)
  {
    analogWrite(ledpin, value);           // sets the value (range fro
    delay(30);                            // waits for 30 milli second
  }
  for(value = 255; value >=0; value-=5)   // fade out (from max to min
  {
    analogWrite(ledpin, value);
    delay(30);
  }
}
```

Press "Upload". After a second, LED will "throb" on and off

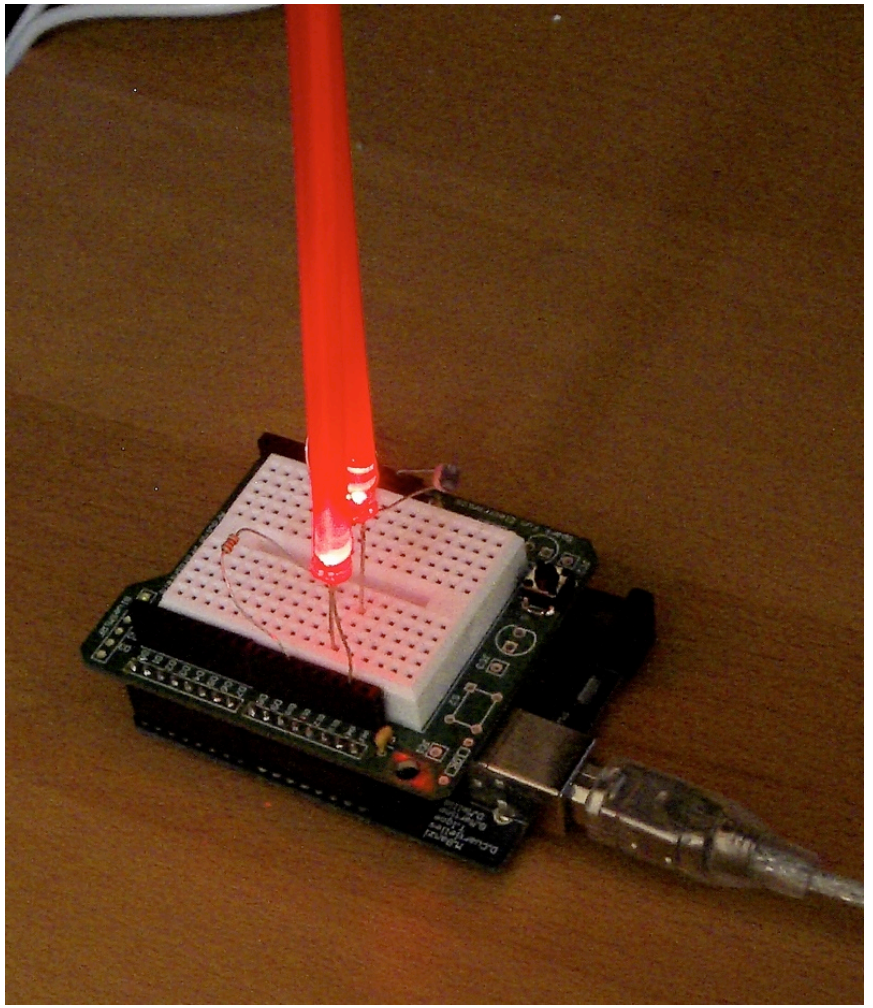
Reduce "delay()" values to make it go faster

Things to Try With "Fading"

- Make it go really fast or really slow
- Fading from half- to full-bright
- Try other PWM pins
- Multiple fading LEDs, at different rates

Aside: LED Light Tubes

Snug-fit straws on the end of your LEDs to make them glow more visibly

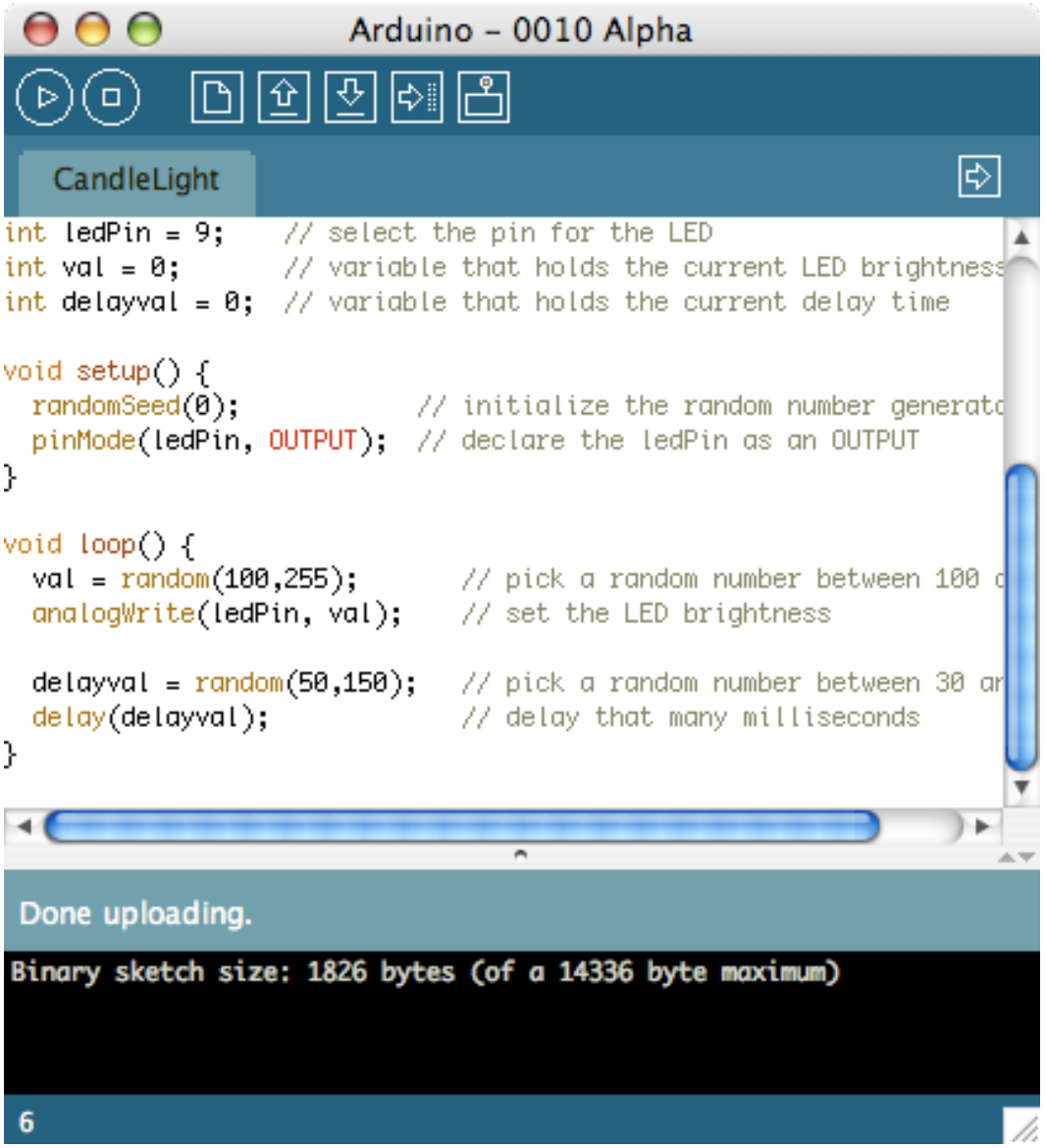


Random Behavior

"CandleLight"

Uses simple pseudo random number generator to mimic flame

Use `random(min,max)` to pick a number between min & max.

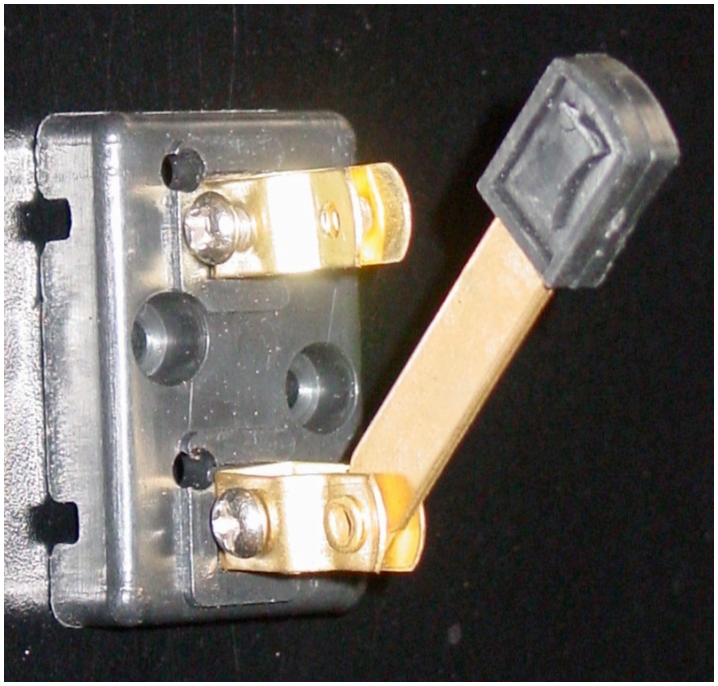


Take a Break

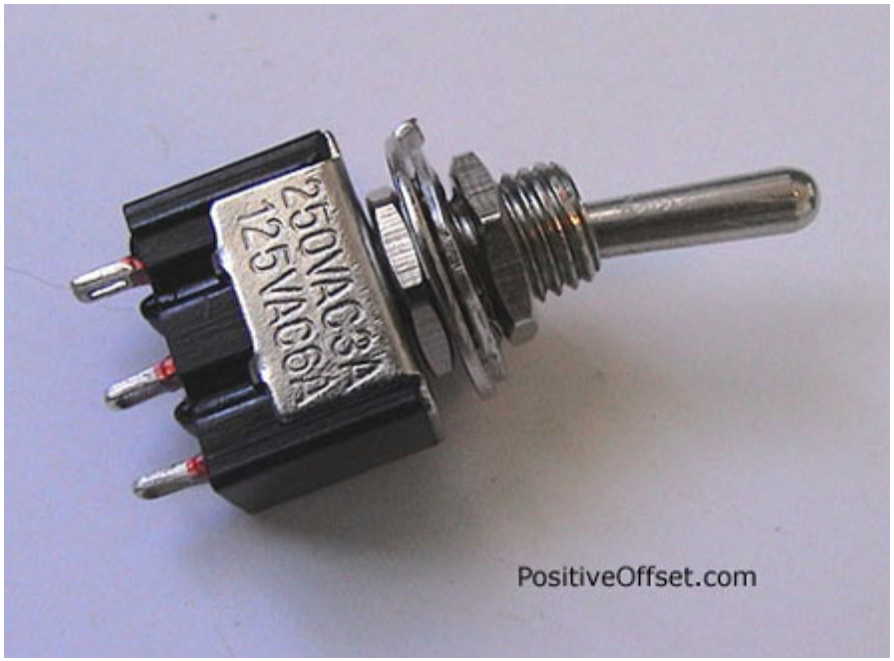
Sensors & Inputs

Many sensors are variations on switches

Switches make or break a connection

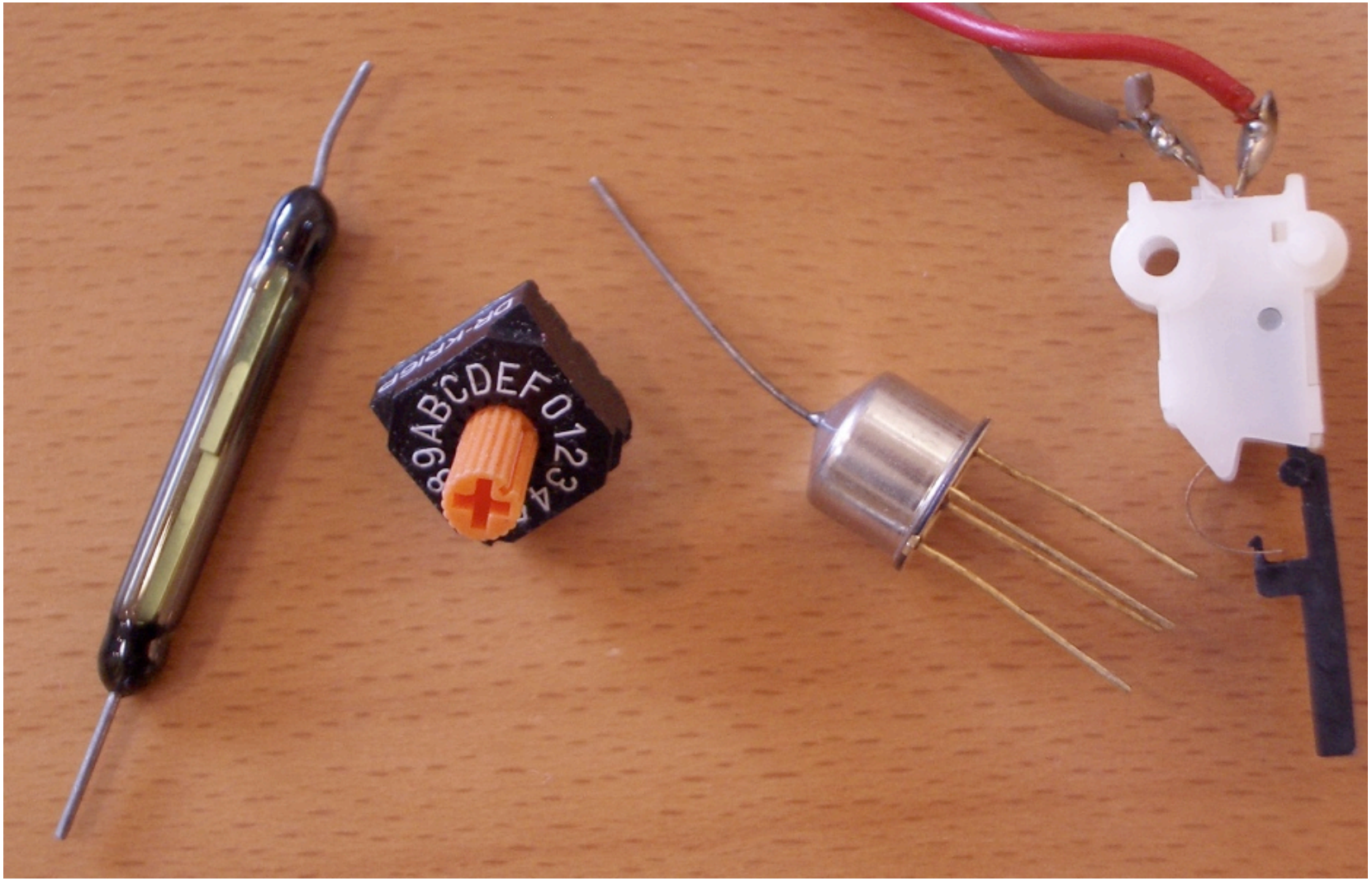


knife switch
(SPST)



toggle switch
(SPDT)

Many Kinds of Switches



magnetic hexidecimal tilt lever

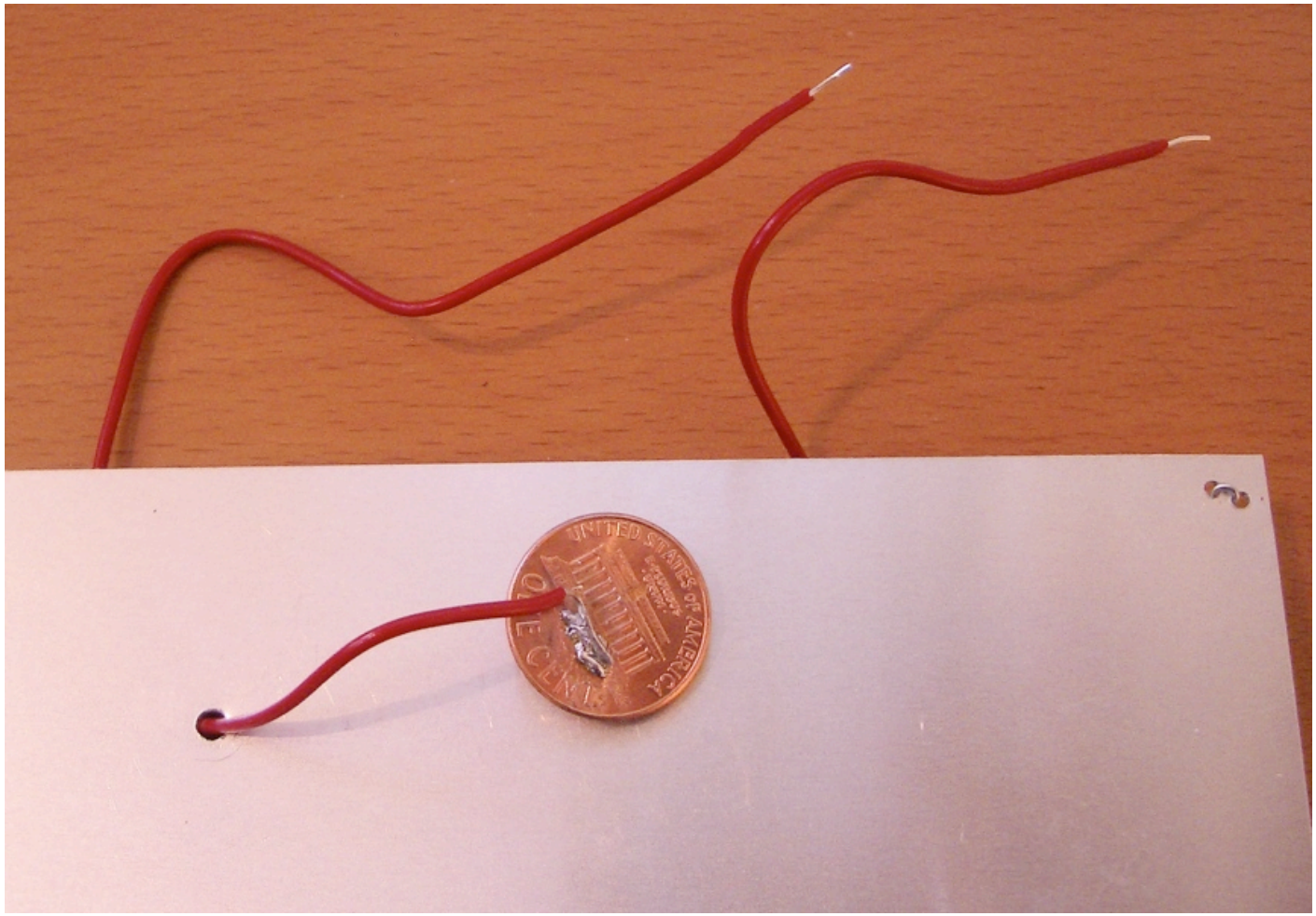
Homemade Switches

"Trick Penny"

Penny on a surface.
When the penny is lifted, alarms go off

Homemade Switches

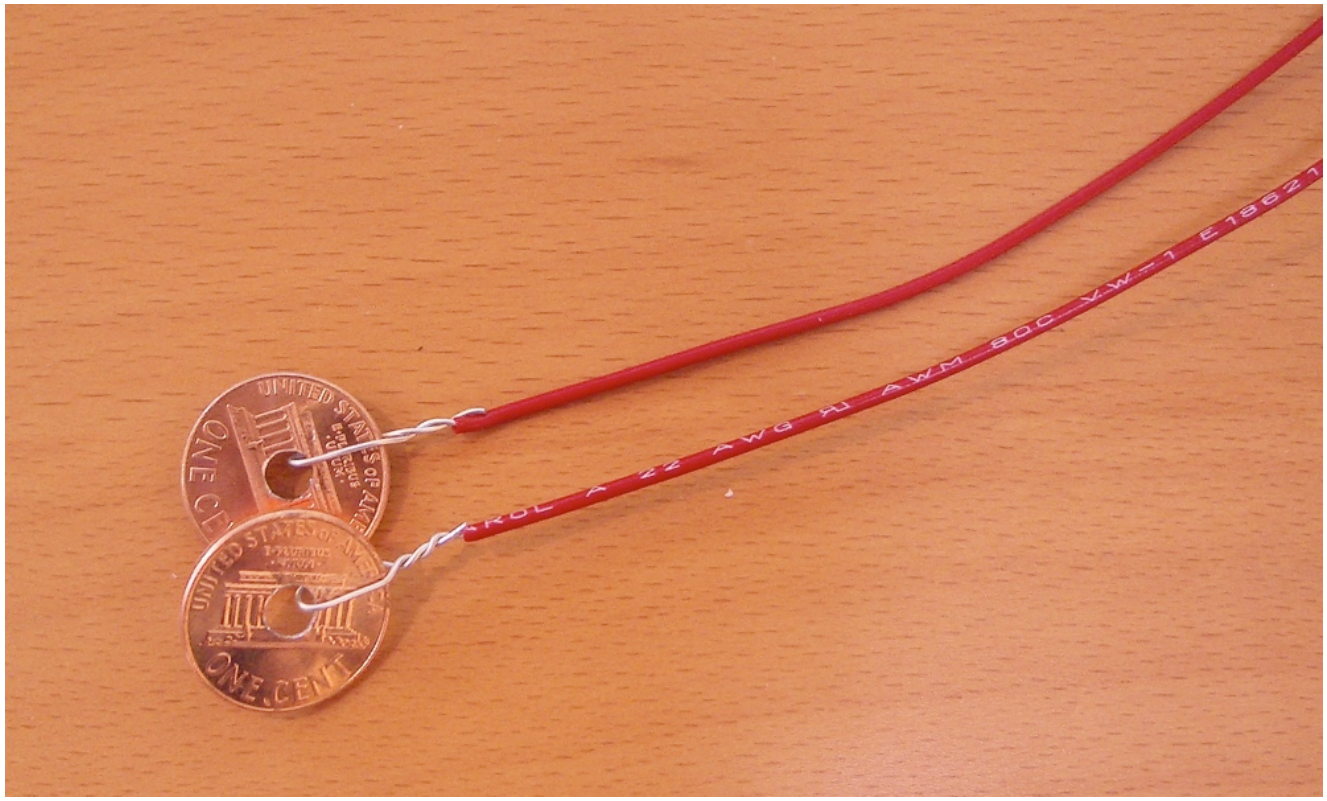
"Trick Penny"



Homemade Switches

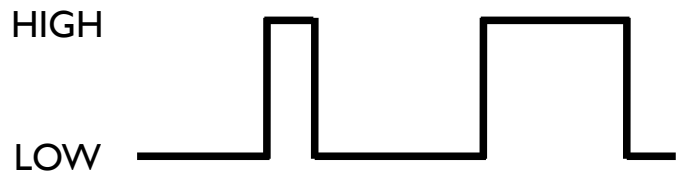
"Smart Wind Chimes"

When the wind blows hard enough, you're sent email



Digital Input

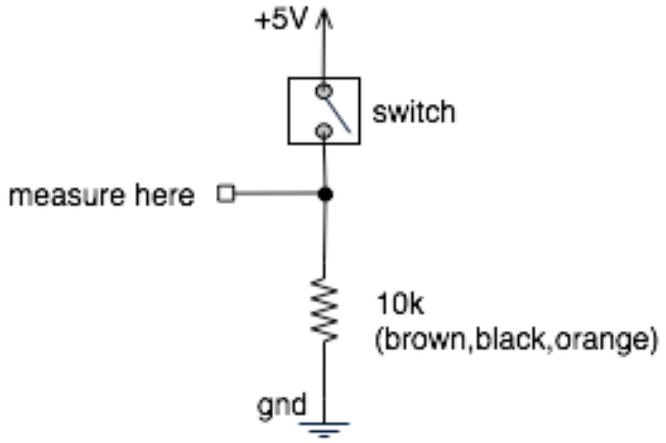
- Switches make or break a connection
- But Arduino wants to see a voltage
  - Specifically, a "HIGH" (5 volts)
  - or a "LOW" (0 volts)



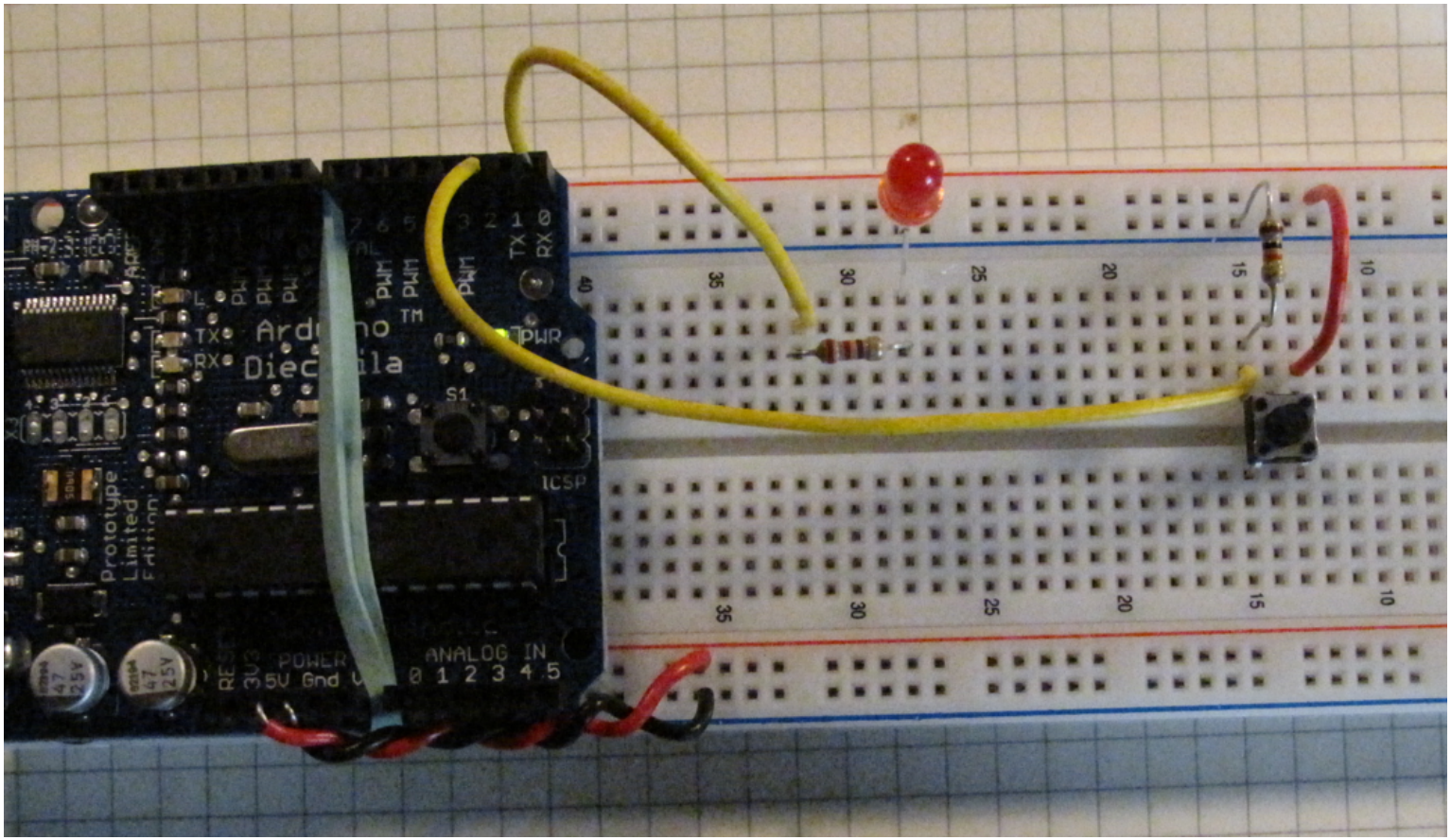
How do you go from make/break to HIGH/LOW?

From Switch to HIGH / LOW

- With no connection, digital inputs "float" between 0 & 5 volts (LOW & HIGH)
- Resistor "pulls" input to ground (0 volts)
- Pressing switch "pushes" input to 5 volts
- Press is HIGH
  Not pressed is LOW



Wiring it up



Let's plug it into pin 2

Using digitalRead()

- In `setup()`: `pinMode(myPin,INPUT)` makes a pin an input
- In `loop()`: `digitalRead(myPin)` gets switch's position
  - If doing many tests, use a variable to hold the output value of `digitalRead()`.
  - `e.g. val = digitalRead(myPin)`

Digital Input Sketch

Load "Sketchbook/Examples/Digital/Button"

```
int ledPin = 13;                // choose the pin for the LED
int inputPin = 2;               // choose the input pin (for a pushbutton)
int val = 0;                    // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    digitalWrite(ledPin, LOW);  // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

Now you control the blinking

(How would you change it to blink the external LED you wired up?)

Using Switches to Make Decisions

- Often you'll want to choose between actions, based on how a switch-like sensor
  - E.g. "If person is detected, fire super soaker"
  - E.g. "If flower pot soil is dry, turn on sprinklers"
- Define actions, choose them from sensor inputs
- Let's try that with the actions we currently know

FadeOrBlink

Load "FadeOrBlink" sketch from the handout

Schematic is same as for "Fading" sketch

Combines "Blink" & "Fading" sketches into one, selected by the button

```
int ledPin = 9;                 // choose the pin for the LED
int inputPin = 2;               // choose the input pin (for a pushbut
int val = 0;                    // variable for reading the pin status
int fadeval = 0;

void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // pushed button means do blinking
    digitalWrite(ledPin, LOW);  // turn LED OFF
    delay(50);
    digitalWrite(ledPin, HIGH); // turn LED ON
    delay(50);
  }
  else {  // else button isn't pressed so do fading
    for(fadeval = 0 ; fadeval <= 255; fadeval+=5) { // fade in (from m
      analogWrite(ledPin, fadeval);           // sets the value (range
      delay(10);
    }
    for(fadeval = 255; fadeval >=0; fadeval-=5) { // fade out (from ma
      analogWrite(ledPin, fadeval);
      delay(10);
    }
  }
}
```

Things to do for next week

- Design a concept for an interactive object
  for inspiration check out:
  http://www.arduino.cc/playground/Projects/ArduinoUsers
- individual or group projects
- Read "Getting Started with Arduino", Chapters 1 through 4, and the Appendices

END Class 1

http://duksta.org/electronics/arduinoclass

John Duksta

john@duksta.org

Resources

http://arduino.cc/
Official homepage. Also check out the Playground & forums

http://ladyada.net/learn/arduino/
Great Arduino tutorials
http://todbot.com/blog/category/arduino/
Various movies, hacks, tutorials on Arduino
http://freeduino.org/
Index of Arduino knowledge

http://adafruit.com/
Arduino starter kits, Boarduino Arduino clone, lots of cool kits
http://sparkfun.com/
Sells Arduino boards and lots of neat sensors & stuff

Books:
"Physical Computing", Dan O'Sullivan & Tom Igoe
"Making Things Talk", Tom Igoe
"Hacking Roomba", Tod E. Kurt