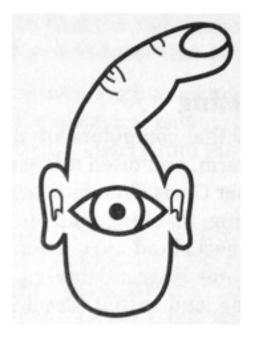
# week 01

# **Physical Computing**

#### Bridging the gap between the physical and virtual

# How the Computer Sees Us

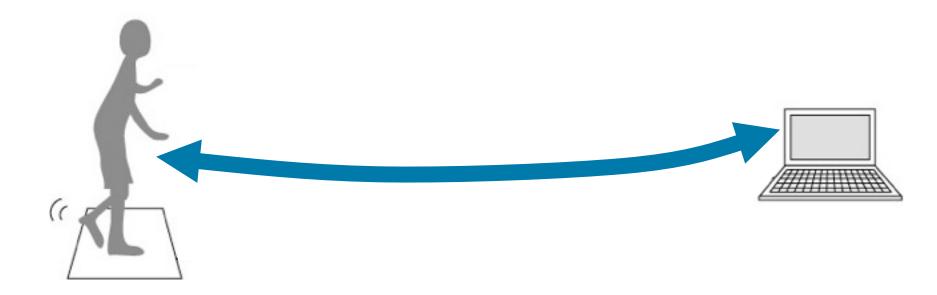
Shall we take a better look at ourselves to see our full range of expression?



2

# **Physical Computing**

A conversation between the physical world and the virtual world of the computer.



# **Physical Computing**

A conversation between the physical world and the virtual world of the computer.



4

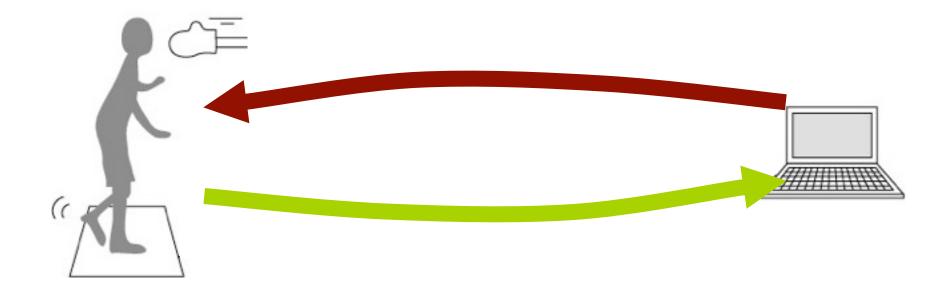
# Input and Output

Ways of sensing your physical energy/expressions. Input is usually easier than output because it takes less energy to sense activity than to move things.

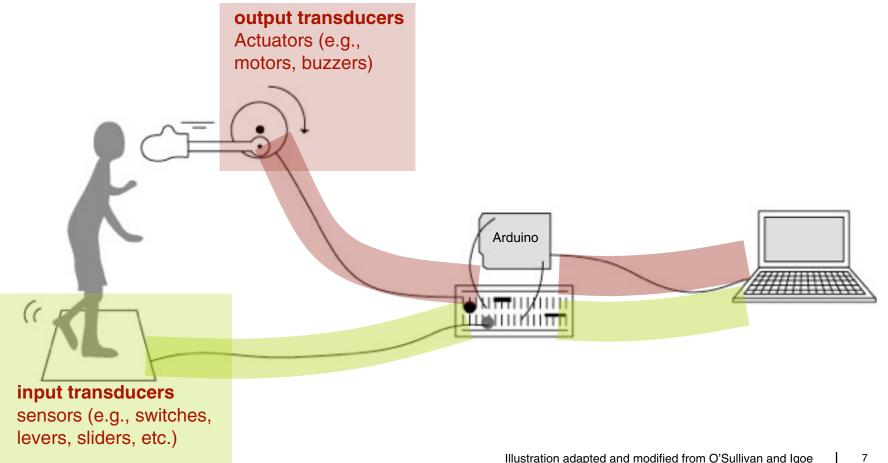


# Input and Output Output

Physical computing is not just about sensing the world, but also about changing it. But moving things are hard (you need electrical and mechanical skills).



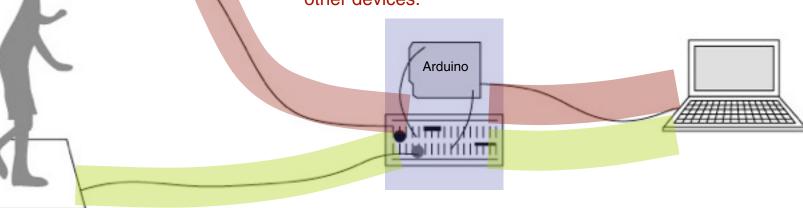
#### **Transducers**



#### **Microcontrollers**

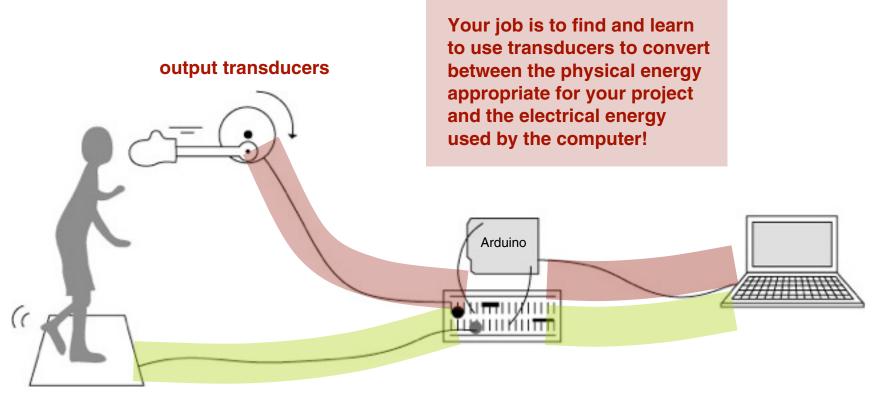
#### Gateway between the physical and the virtual

 Receiving information from sensors,
 controlling basic motors and other devices that create physical change, and
 sending information to computers and other devices.



8

#### **Transduction**



input transducers

9

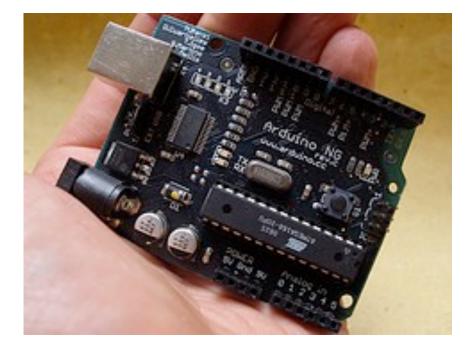
# **Working with Arduino**

#### An open-source electronics prototyping

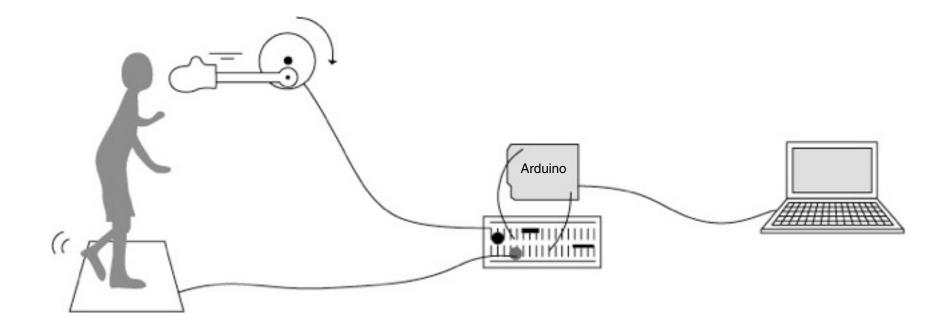
Create interactive objects and environments

### What is Arduino

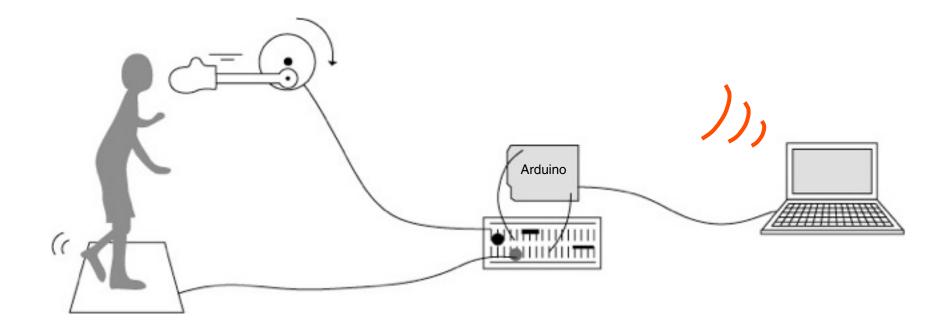
#### A tiny computer you can program, for rapid prototyping



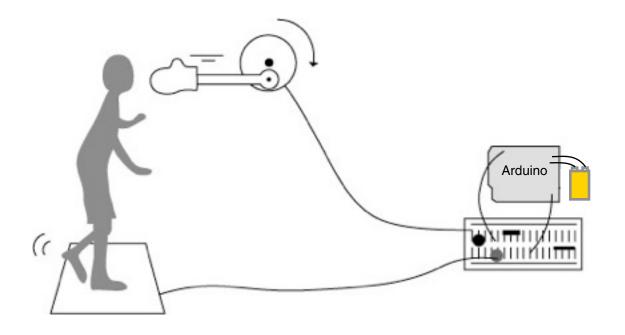
#### **Arduino as an Interface Board**



#### **Arduino as an Interface Board**

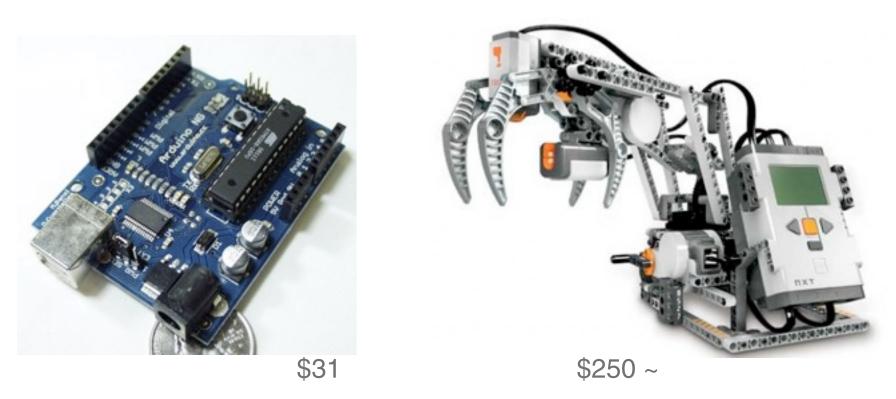


#### Arduino as an Embedded Computing Device



### What is Arduino

#### Relatively cheap (compare with LEGO Mindstorm)



### What is Arduino

It used to be a bit more complicated...

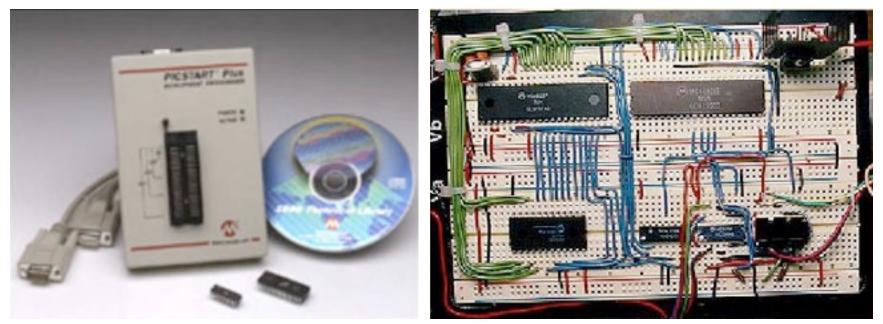
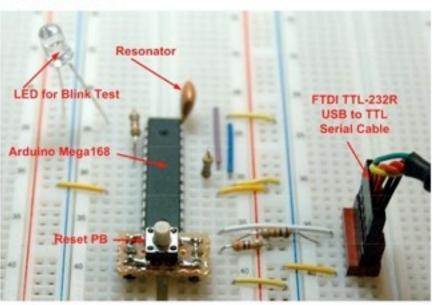


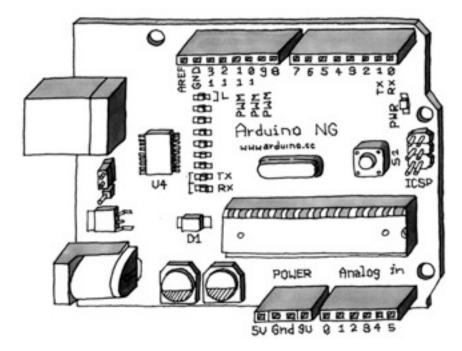
image from www.tangentsoft.net/elec/breadboard.html



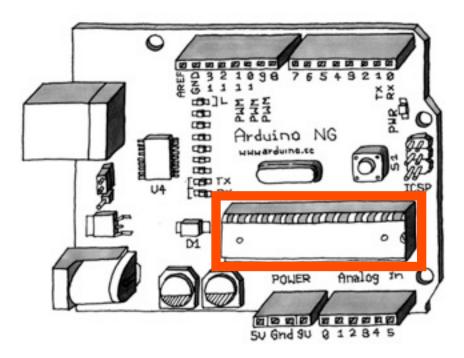
# Open source, so you can build one yourself!



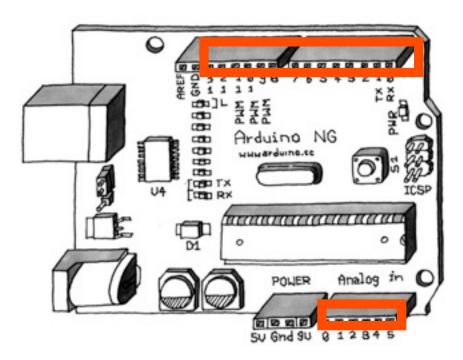
Here is a truly barebones Arduino setup. Just the Arduino chip and a few support parts. This has to be close to the simplest and lowest cost way to play with microcontrollers. The only special parts are the resonator and the Mega168 programmed with the Arduino boot loader. Everything else you should already have as an electronics hobbyist.

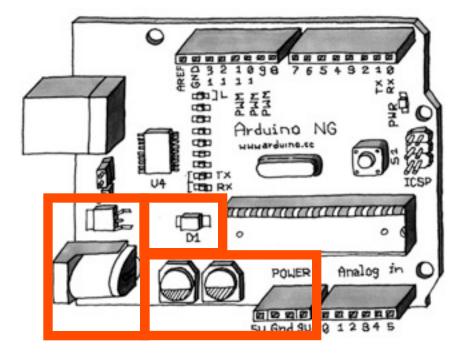


- 32 kBytes of Flash program memory (your program stays in Arduino when powered off)
- 2 kByte of RAM
- 16 MHz processor speed (c.f., Apple II: 1 MHz)

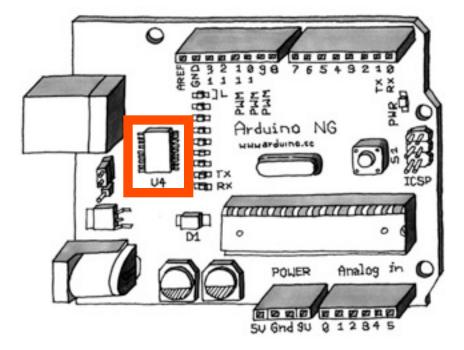


- 14 digital input/output pins
- 6 analog input pins





• USB to serial chip (converts simple serial signal to USB)



#### Arduino Software www.arduino.cc

Reference Blog Playground Forum	Help   Sign in or Register
	(waid)

Buy Download Cetting Started Learning Reference Hardware FAQ



Photo by the Arduino Team

#### Arduino on Twitter (more)

2 days ago @apfelslice it's now in the hands of 20 beta testers.. more news Arduino is an open-source electronics prototyping platform based on flexible, easyto-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the <u>Arduino programming</u> <u>language</u> (based on <u>Wiring</u>) and the Arduino development environment (based on <u>Processing</u>). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

The boards can be built by hand or purchased preasembled; the software can be <u>downloaded</u> for free. The hardware reference designs (CAD files) are <u>available</u> under an open-source license, you are free to <u>adapt them to your needs</u>.

Arduino received an Honorary Mention in the Digital Communities section of the 2006 Ars Electronica Prix. The Arduino team is: Massimo

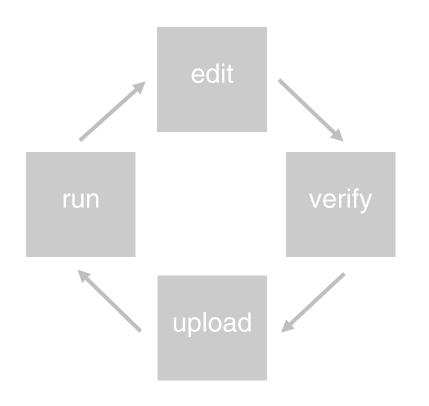
### **Arduino Software**

Arduine - 0003 Alpha Ne Edt Sketch Tools Help		
Disk		Ē
/*		
* Blink		
* then off for one second, an	Turns on an LED on for one second, d so on We use pin 13 because, serd, it has either a built-in LED hat you need only an LED.	
<pre>http://www.arduino.cc/en/Tu */</pre>	torial/Blink	
int ledFin = 13;	// LED connected to digital pin 13	
void setup []	// run once, when the sketch starts	
pinHode(ledFin, OUTFUT):	$\ensuremath{{\prime\prime}}\xspace$ )/ sets the digital pin as output	
() quoi houp ()	// run over and over again	
digitalWrite(ledPin, HIGH);	// sets the LED on	
deley(1000);	// waits for a second	
digitalWrite(ledPin, LOW);		
deley(1000);	// waits for a second	
All and a second se		
Dane compiling.		

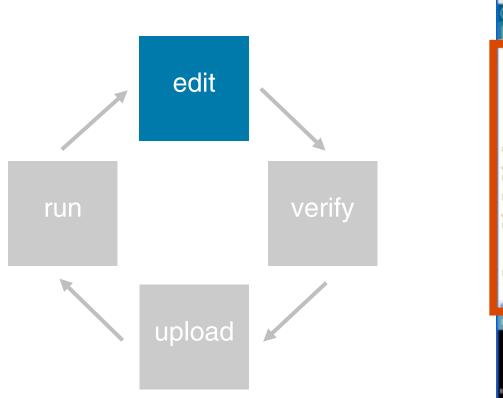
Binary sketch size: 1108 bytes (of a 14336 byte maximum)

### **Arduino Sketches**

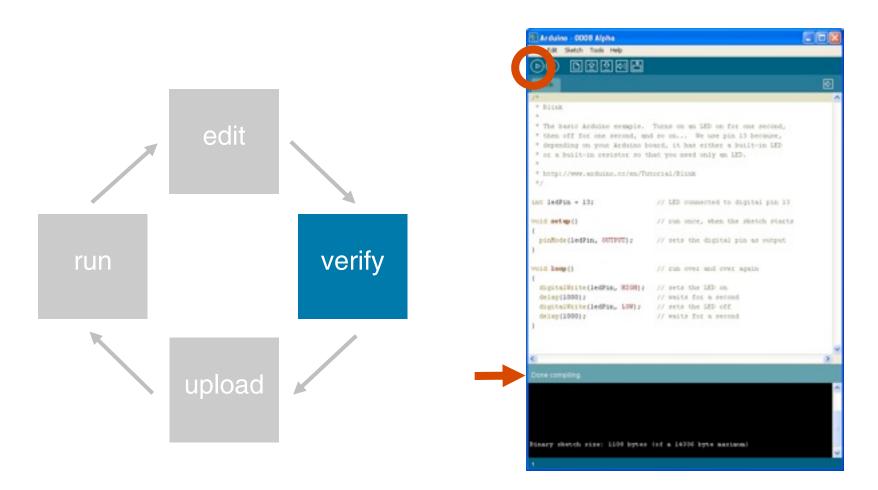
Edit Sketch To	ols Help					
New	Ctrl+N					
Sketchbook	•	Open	Ctrl+O			
Save	Ctrl+S	arduino-0008-win	۲			¢
Save As	Ctrl+Shift+S	Kimiko Sketching				
Upload to I/O Board	Ctrl+U	processing-0124				
Page Setup	Ctrl+Shift+P	sketch_070811a				
Print	Ctrl+P	sketch_070811b		for one second,		
Preferences	Ctrl+,	sketch_070813a		in 13 because,		
Quit	Ctrl+Q	sketch_070819a		a built-in LED	- 24	
		Examples	•	Analog		
http://www.ard	iuino.cc/en/T	utorial/Blink		Communication	•	
1				Digital	•	Blink
				Library-EEPR.OM		Button
t ledPin = 13;		// LED conne	ected	Library-Matrix		Loop
			1000	Library-SoftwareSerial	•	
oid setup()		// run once,	, when	Library-Stepper	•	
ninMode (ledDin	OUTPUT) -	11 sars the	diate.	Library-Wire	•	
primode (redrin,	,	// Secs che	argros	a have as eachers		
id loop()		// run over	and ou	/er again		
digitalWrite(1	edPin, HIGH);			A second s		
delay(1000);		// waits for a second				
digitalWrite(1						

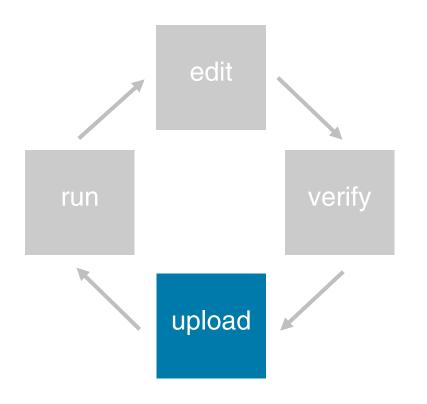


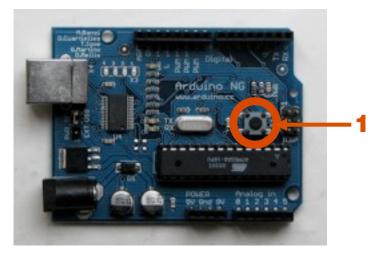
uns un an 18D on for one perond. 10 mm., Ne ure pin 13 because,	Ø
	Ø
to sm We use pin 13 because.	
rd, it has either a built-in LED	
t you need only an LED.	
rial/Blink	
I HE AND A REAL PROPERTY AND A REAL PROPERTY.	
the connected on arditury buy an	
// run once, when the shetch starts	
// sets the digital pin as output	
// run over and over again	
// waits for a second	
	2
	rial/Blink // LED comnected to digital pin 13 // run once, when the sketch starts // sets the digital pin as output



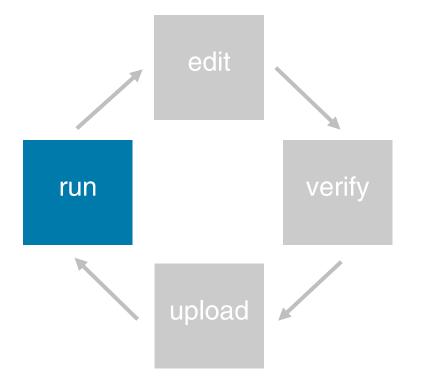
<pre>Phe Edd Shetch Took Peeb</pre>	🗄 Arduino - 0008 Alpha		
<pre> * Slink * * * * * * * * * * * * * * * * * * *</pre>	File Edit Sketch Tools Help		
<pre>* Blink * * * * * * * * * * * * * * * * * * *</pre>			
<pre>* Blink * * * * * * * * * * * * * * * * * * *</pre>	Birk		তা
<pre>* * * The Baste Acduino example. Tunns us an LED on fix one second. * Uses off Dox one second, and so us We use pin 13 because, * depending on your Acduino beard, it has either a built-in LED * or a built-in resister so that you need only an LED. * thetp://www.arduino.cc/em/Turcerial/Blink */ int ledPin = 13; // LED connected to digital pin 13 void setup() // run onde, when the sketch starts {     pinEnded(ledPin, OUTPUT); // sets the digital pin as output } void leop() // run over and over again {     digitalWrite(ledPin, 1001); // sets the LED on     delag(1000); // waits Dor a second } Core compoint {     Core company     Core company</pre>			
<pre>* The Basid Acduine example. Tunns us an LED on fix one second, * then off Dox one second, and so us We use pin 13 because, * depending on your Acduine board, it has either a built-in LED * or a built-in resister so that you meed only an LED. * hetp://www.ardwine.cc/wm/Tutorial/Blink */ int ledFim = 13; // LED commetted to digital pin 13 void setup() // run once, when the sketch starts { pinBode(LedFim, OUTPUT); // sets the digital pin as output } void leeg() // run over and over sgain { digitalWrite(LedFim, LDW); // sets the LED on digitalWrite(LedFim, LDW); // sets the LED off digitalWrite(LedFim, LDW); // waits for a second digitalWrite(LedFim, LDW); // waits for a second } } Core corecine</pre>			
<pre>* then off for one second, and so sm We use pin 13 because, * depending on your Arduino board, it has either a built-in LED * or a built-in resistor so that you need only an LED. * http://www.arduino.cc/em/Tutorial/Blink */ int ledFus = 13; // LED connected to digital pin 13 void setup() // run once, when the sketch starts { pinBode(ledFin, OUTPUT); // sets the digital pin as output } void leeg() // run over and over again { digitalWrite(ledFin, EDW); // sets the LED on deleg(1000); // waits for a second digitalWrite(ledFin, LDW); // sets the LED on deleg(1000); // waits for a second } } Core corrected </pre>		Torona on an 187 on for one particul	
<pre>* depending on your Ardsimo board, it has either a built-in LED * de a built-in remistor so that you meed only an LED. * * hetp://www.ardwine.cc/em/Tutorial/dlink */ int ledFin = 13; // LED connected to digital pin 13 void setup() // run onde, when the sketch starts {     pinNode(ledFin, GUTPUT); // sets the digital pin as output } void heap() // run over and over sgain {     digitalWrite(ledFin, MDNM); // sets the LED on     delay(1000); // waits for a second     digitalWrite(ledFin, LNM); // sets the LED of     delay(1000); // waits for a second } Core correctes </pre>			
<pre>* * http://www.ardwine.cc/em/Turocial/dlink */ int ledFim = 13; // LED commetted to digital pin 13 void setup() // cm once, when the sketch starts {     pinHode(ledFin, OUTPUT); // sets the digital pin as output } void beep() // cm over and over spain {     digitalWitte(ledFin, MDW); // sets the LED on     delegitaDWitte(ledFin, LDW); // sets the LED off     delegitaDWitte(ledFin, LDW); // sets</pre>			
<pre>* http://www.ardwins.cc/em/Tutorial/Blink */ int ledFin = 13; // LED connected to digital pin 13 void setup() // run onde, when the sketch starts {     pinEnde(ledFin, OUTPWT); // run over and over again {     digitalWrite(ledFin, WDW); // sets the LED on     deleg(1000); // waits Dor a second     digitalWrite(ledFin, LOW); // sets the LED on     deleg(1000); // waits Dor a second } Corescoreding </pre>		hat you need only an LED.	
<pre>*/ int ledFin = 13; // LED connected to digital pin 13 void setup() // run oute, when the sketch starts {     pinHode(LedFin, OWIPUT); // run over and over again } void leep() // run over and over again {     digitalWrite(LedFin, MDW); // sets the LED on     deleg(1000); // waits for a second     digitalWrite(LedFin, LOW); // sets the LED of     deleg(1000); // waits for a second } Concernent </pre>		and all different	
<pre>int ledPin = 13; // LED connected to digital pin 13 void setup() // run unde, when the shotch starts {     pinHode(LedPin, OWIPHT); // sets the digital pin as output } void leep() // run over and over spain {     delep(1000); // sets the LED on     delep(1000); // sets the LED off     delep(1000); // waits for a second } Core corpling </pre>		a na ya a dila.	
<pre>void setup() // run oude, when the sketch starts {     pinHode(LedPin, OWIPUT); // sets the digital pin as output } void leep() // run over and over again {     digitalWrite(LedPin, MDW); // sets the MDD on     deleg(1000); // waits for a second     digitalWrite(LedPin, 10W); // sets the MDD of     deleg(1000); // waits for a second } Contenangle Contenangle</pre>			
<pre>{     pinEcde(LedFin, OUTPUT); // sets the digital pin as output } void Leop() // rum over and over spain {     digitalWrite(LedFin, MIDH); // sets the LED on     delep(1000); // waits for a second     digitalWrite(LedFin, LOH); // sets the LED off     delep(1000); // waits for a second } Core correlate </pre>	int ledFim = 10;	// LED connected to digital pin 13	
<pre>{     pinEcde(LedFin, OUTPUT); // sets the digital pin as output } void Leop() // rum over and over spain {     digitalWrite(LedFin, MIDH); // sets the LED on     delep(1000); // waits for a second     digitalWrite(LedFin, LOH); // sets the LED off     delep(1000); // waits for a second } Core correlate </pre>	unid anten ()	// run once, when the shatch starts	
<pre>} void heep() // run over and over spain {     digitalWrite[14:0%is, WIDW]; // sets the LED on     delag(10001) // waits Dor a second     digitalWrite[14:0%is, 10W]; // sets Widts Dor a second     delag(10001); // waits Dor a second } Core company </pre>			
<pre>void Deep() // run over and over spain {     digitalWitte(l+dFin, HDW); // sets the LED on     deleg(1000); // waits for a second     digitalWitte(l+dFin, LOW); // sets the LED off     deleg(1000); // waits for a second } Core complete Core complete</pre>	pinHode (LedPin, OUTPUT) :	// sets the digital pin as output	
I digitalWrite(ledWim, WIDW); // sets the LED on deleg(1990); // waits Dox a second digitalWrite(ledWim, LOW); // sets the LED off deleg(1990); // sets the LED off deleg(1990); // waits Dox a second } Econe compling	1		
I digitalWrite(ledWim, WIDW); // sets the LED on deleg(1990); // waits Dox a second digitalWrite(ledWim, LOW); // sets the LED off deleg(1990); // sets the LED off deleg(1990); // waits Dox a second } Econe compling	world Lama ()	// run over and over again	
deleg(1000); // waits Dor a second digitalWrite(LedWim, 100); // sets the LED off deleg(1000); // waits Dor a second } Constant of the second		The same state and state a second	
digitalWrite(LedWin, LOW); // sets the IED off delay(1990); // white for a second } C Done compling			
deleg(1990): // waits for a second } Constant Done compling			
) Cone compling			
	č.		15
Dinary sketch size: 1100 bytes (of a 14304 byte maximum)			
Dinary sketch size: 1100 bytes (of a 14304 byte maximum)			
Binary sketch size: 1100 bytes (of a 14004 byte maximum)			
Binary sketch size: 1100 bytes (of a 14304 byte maximum)			
Dinary sketch size: 1100 bytes (of a 14706 byte maximum)			
	Dinary sketch size: 1100 bytes	(of a 14776 byte maximum)	
			4











Watch your LED blink!

# **Arduino Program**

Program consists of 3 parts:

- 1. Declare variables at top
- 2. Initialize

setup() - run once at beginning, set pins

#### 3. **Run**

loop() - run repeatedly, after setup()

# **Arduino Language**

Like C but easier

**Example 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" PWM value
- delay() wait an amount of time

```
* Blink
 .
 * The basic Arduino example. Turns on an LED on for one second,
* then off for one second, and so on... We use pin 13 because,
* depending on your Arduino board, it has either a built-in LED
* or a built-in resistor so that you need only an LED.
* http://www.arduino.cc/en/Tutorial/Blink
 #1
int ledPin = 13;
                               // LED connected to digital pin 13
                               // run once, when the sketch starts
void setup()
 pinMode(ledPin, OUTPUT);
                              // sets the digital pin as output
void loop()
                               // run over and over again
 digitalWrite(ledPin, HIGH);
                              // sets the LED on
  delay(1000);
                               // waits for a second
 digitalWrite(ledPin, LOW);
                              // sets the LED off
 delay(1000);
                               // waits for a second
```

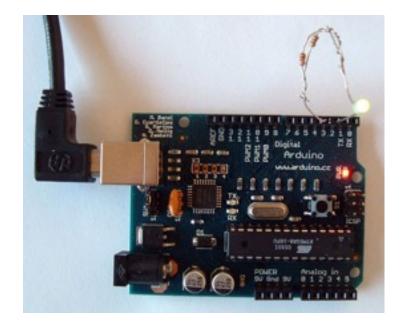
/*		
* Blink		
*		
* The basic Arduino example.	Turns on an LED on for one second,	
* then off for one second, an	nd so on We use pin 13 because,	
	board, it has either a built-in LED	
* or a built-in resistor so	that you need only an LED.	
*		
* http://www.arduino.cc/en/To	utorial/Blink	
*/		
int ledPin = 13;	// LED connected to digital pin 13	Declare variables
<pre>roid setup()</pre>	// run once, when the sketch starts	
(		
pinMode(ledPin, OUTPUT);	// sets the digital pin as output	
)		
void loop()	// run over and over again	
vold roop()	// Ital over and over again	
dista Weite Medbin UTCU.	11 man the TPD on	
digitalWrite(ledPin, HIGH);		
delay(1000);	// waits for a second	
digitalWrite(ledPin, LOW);	// sets the LED off	
delay(1000);	// waits for a second	
)		

```
* Blink
 .
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 .
 * http://www.arduino.cc/en/Tutorial/Blink
 #1
int ledPin = 13;
                               // LED connected to digital pin 13
                                                                                 Initialize
                                // run once, when the sketch starts
void setup()
Ł
 pinMode(ledPin, OUTPUT);
                               // sets the digital pin as output
void loop()
                               // run over and over again
  digitalWrite(ledPin, HIGH);
                              // sets the LED on
  delay(1000);
                               // waits for a second
  digitalWrite(ledPin, LOW);
                               // sets the LED off
  delay(1000);
                               // waits for a second
```

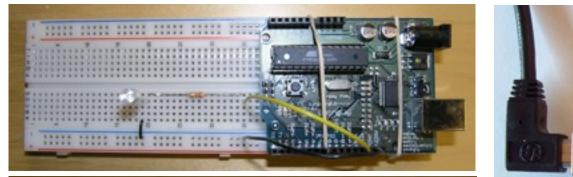
Run

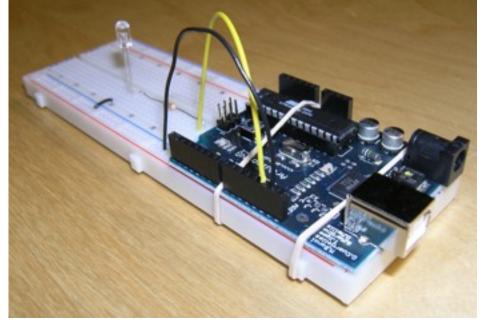
```
* Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
* then off for one second, and so on... We use pin 13 because,
* depending on your Arduino board, it has either a built-in LED
* or a built-in resistor so that you need only an LED.
 *
* http://www.arduino.cc/en/Tutorial/Blink
 #1
                               // LED connected to digital pin 13
int ledPin = 13;
                               // run once, when the sketch starts
void setup()
£
 pinMode(ledPin, OUTPUT);
                              // sets the digital pin as output
void loop()
                               // run over and over again
 digitalWrite(ledPin, HIGH);
                              // sets the LED on
 delay(1000);
                               // waits for a second
 digitalWrite(ledPin, LOW);
                               // sets the LED off
 delay(1000);
                               // waits for a second
```

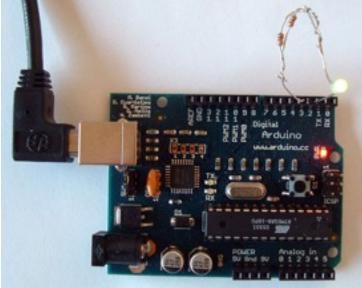
#### **Arduino and Breadboard**



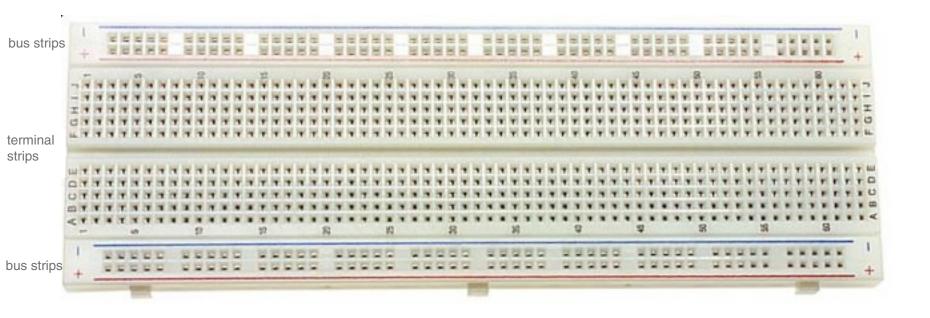
#### **Arduino and Breadboard**



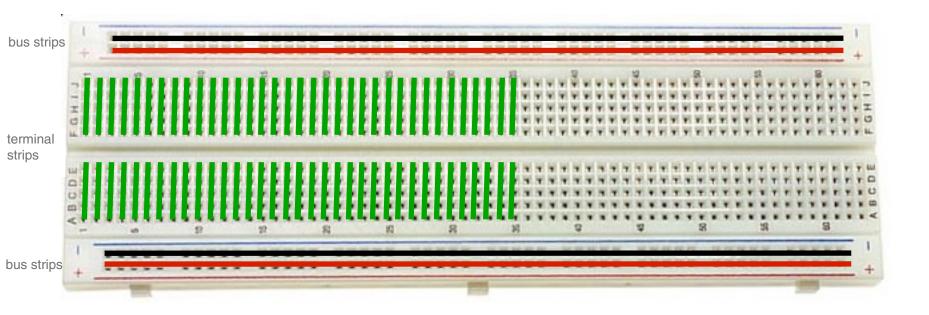




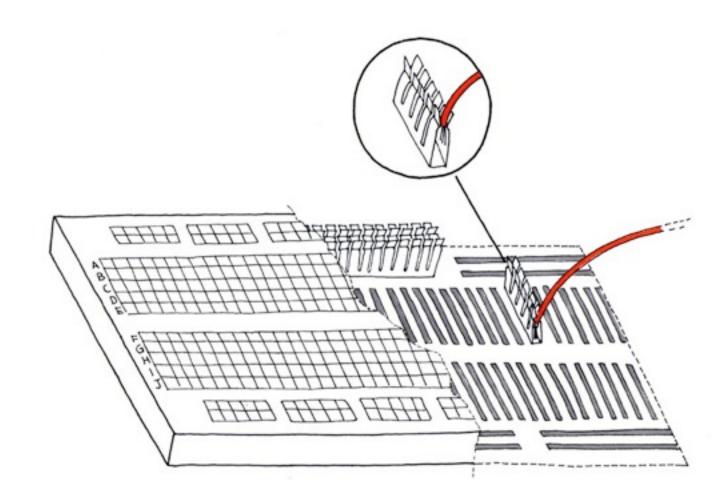
#### **Solderless Breadboard**



#### **Solderless Breadboard**

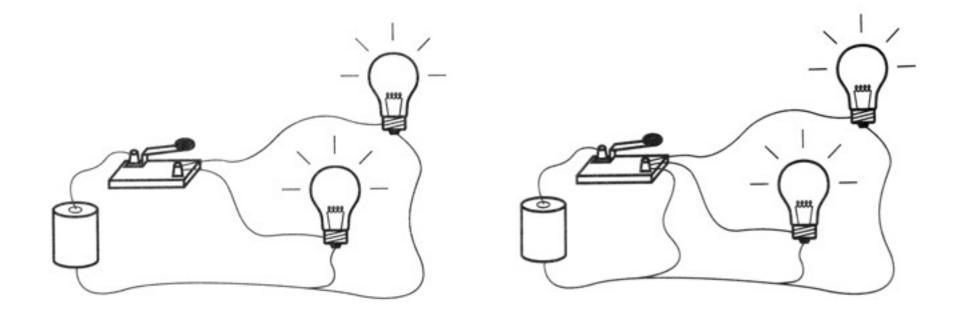


#### **Solderless Breadboard**



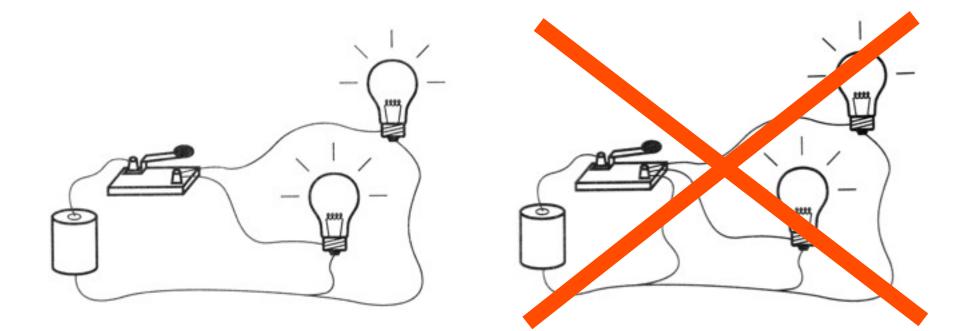
### **Circuits: Avoid Shortcuts**

Electricity always favors the path of least resistance to ground



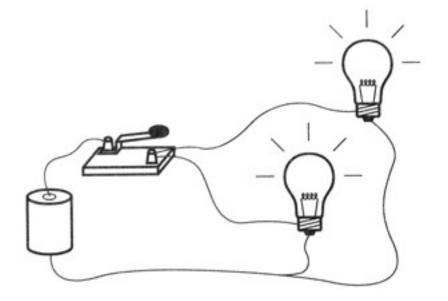
### **Circuits: Avoid Shortcuts**

Electricity always favors the path of least resistance to ground



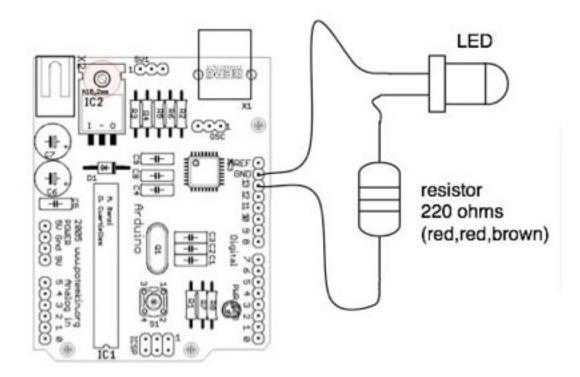
### **Circuits**

#### All the electrical energy in a circuit must be used



#### LED

- LED = Light-Emitting Diode
- Needs a "current limiting" resistor, or burns out



### **Circuits Summary**

Avoid deadly shortcuts

- Flows to the lowest resistance
- All the electrical energy in a circuit must be used

When in doubt, talk to us, we can help

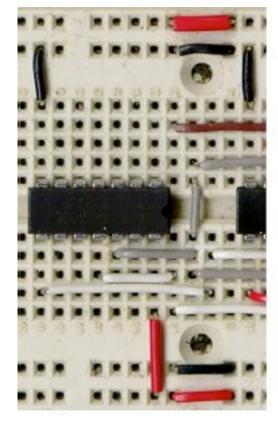
(Your Arduino has some fail safe, but in the worst case, you could fry your board [\$29])

## Try to be Neat

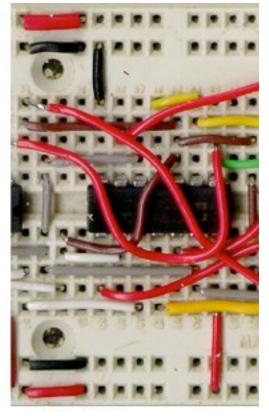
Color code:

- Red: power
- Black: ground

Be consistent

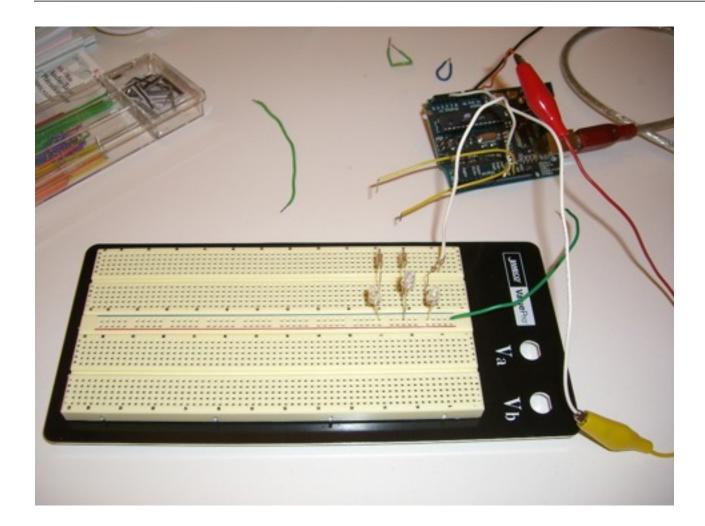


Good



Bad

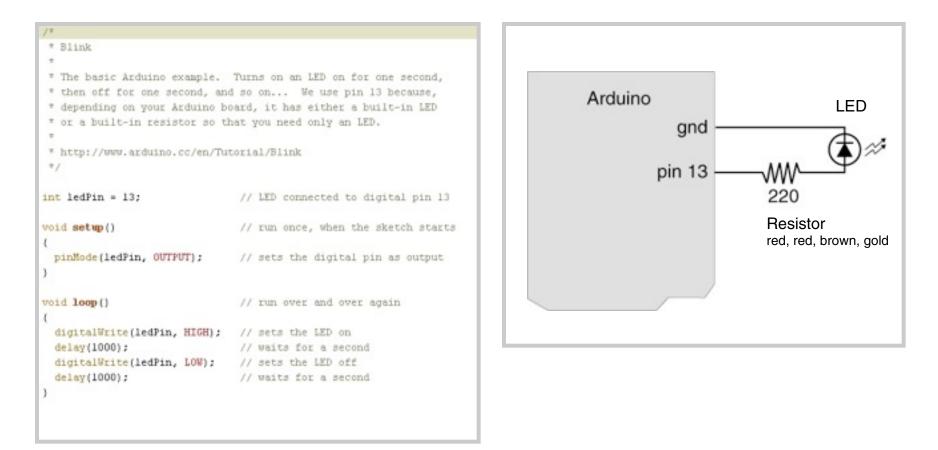
#### **Be Careful**



### Lab Assignment This Week

- Get the course lab kit
- Download & install Arduino software (www.arduino.cc)
- Make an LED blink
  - Make it blink at different rate
- Create your course web account
- Post the photo of your board and an optional comment on the course website

#### Lab Assignment: Blinking LED



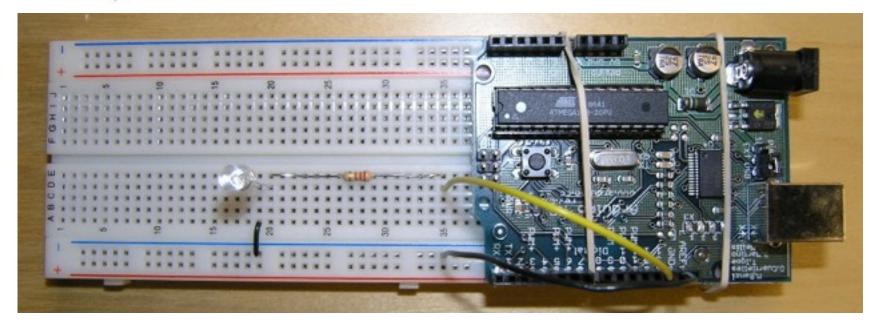
#### Your assignment looks like this



LED Shorter leg → ground Polarity matters



220 ohm resistor red, red, brown, gold Polarity does not matter



### **Next Wednesday**

Don't forget to bring your lab kit in Do use the lab hour to catch up Create your course account Post your assignment on the course website

# **Thanks!**