HackerSpace Camp

Nuno Alves, Michael Framarin, Neeraj Magotra

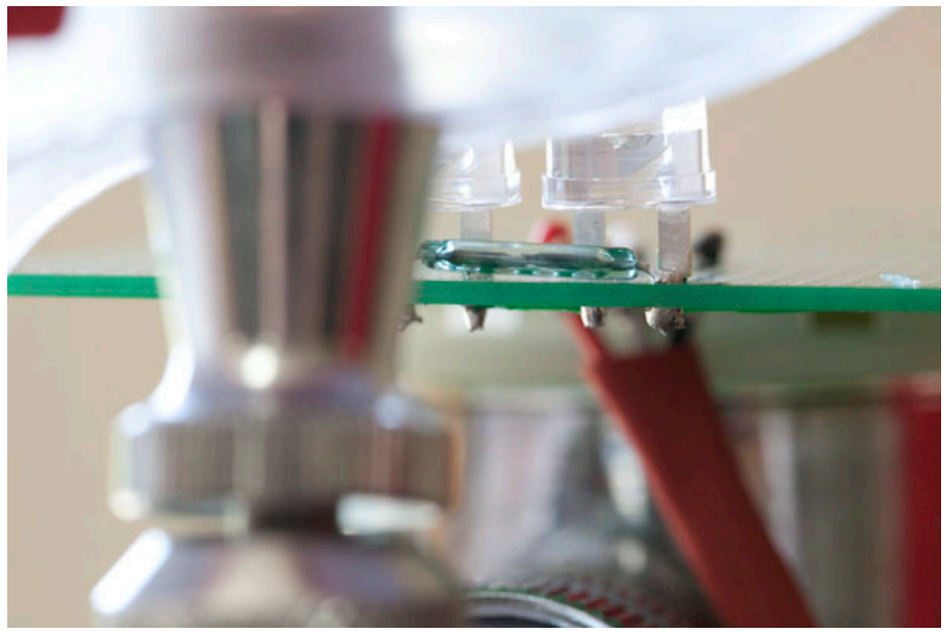
WESTERN NEW ENGLAND UNIVERSITY | WNE

What are we going to do?

- Learn how to control anything that is mechanical or electrical!
- Who cares???
- What can I do with this???

So, how much (root) beer is left in my keg?

100-lb Force sensor reports weight



Switch closes when a magnetic field is near

Keg monitor

WESTERN NEW ENGLAND UNIVERSITY | WNE

How can we make mowing the lawn fun?

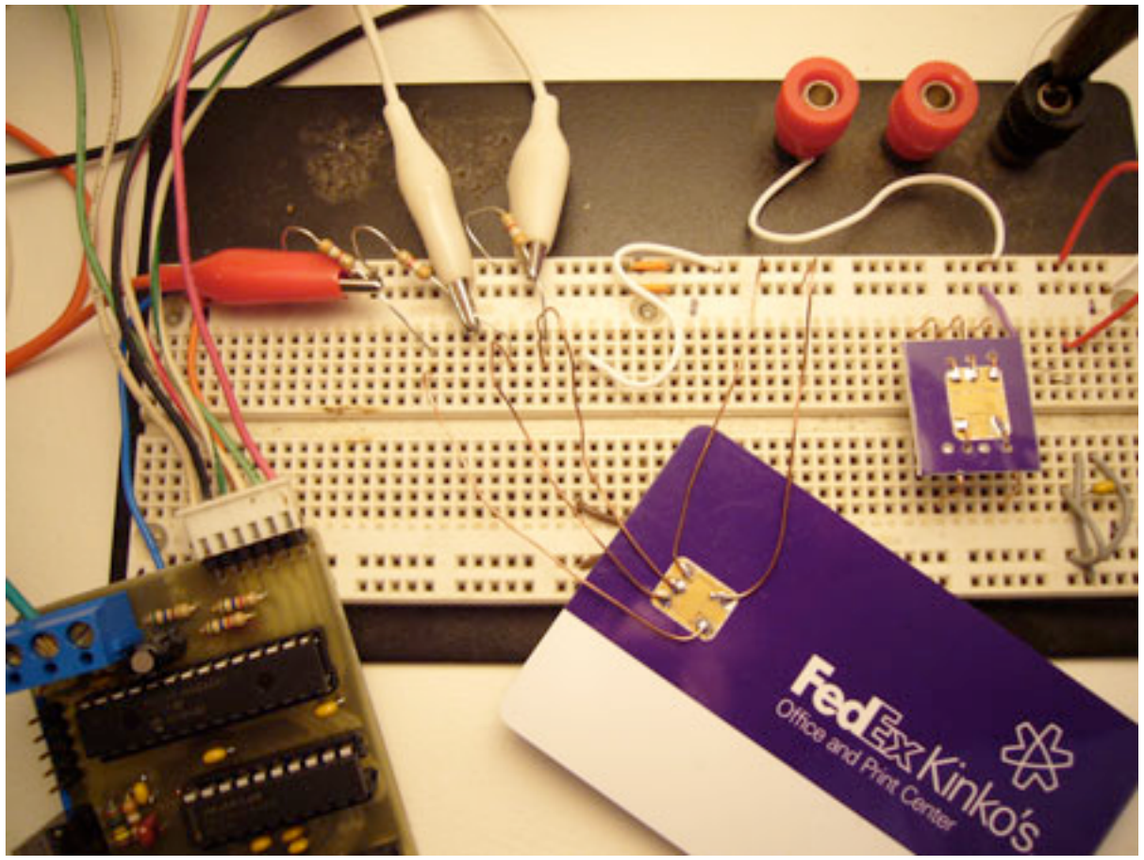
Mowing the lawn - Video

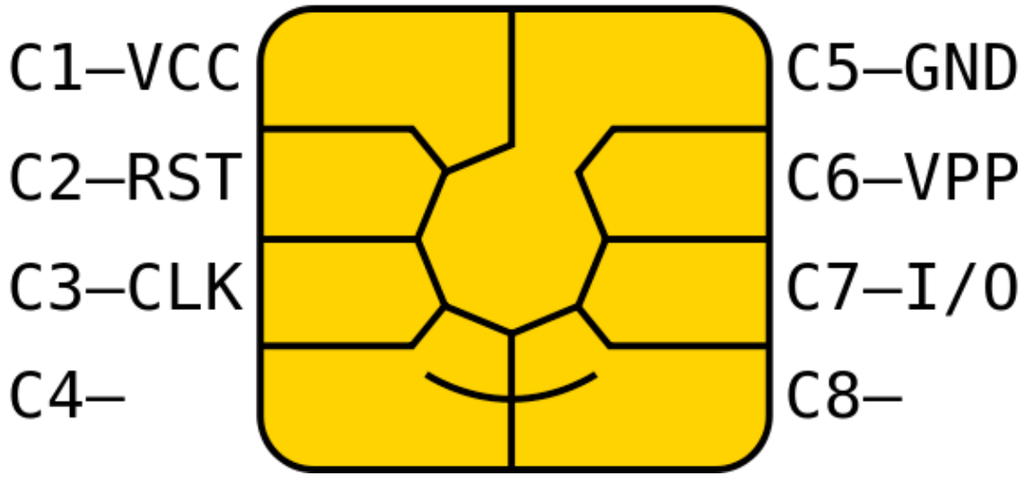
Carrying topsoil - Video

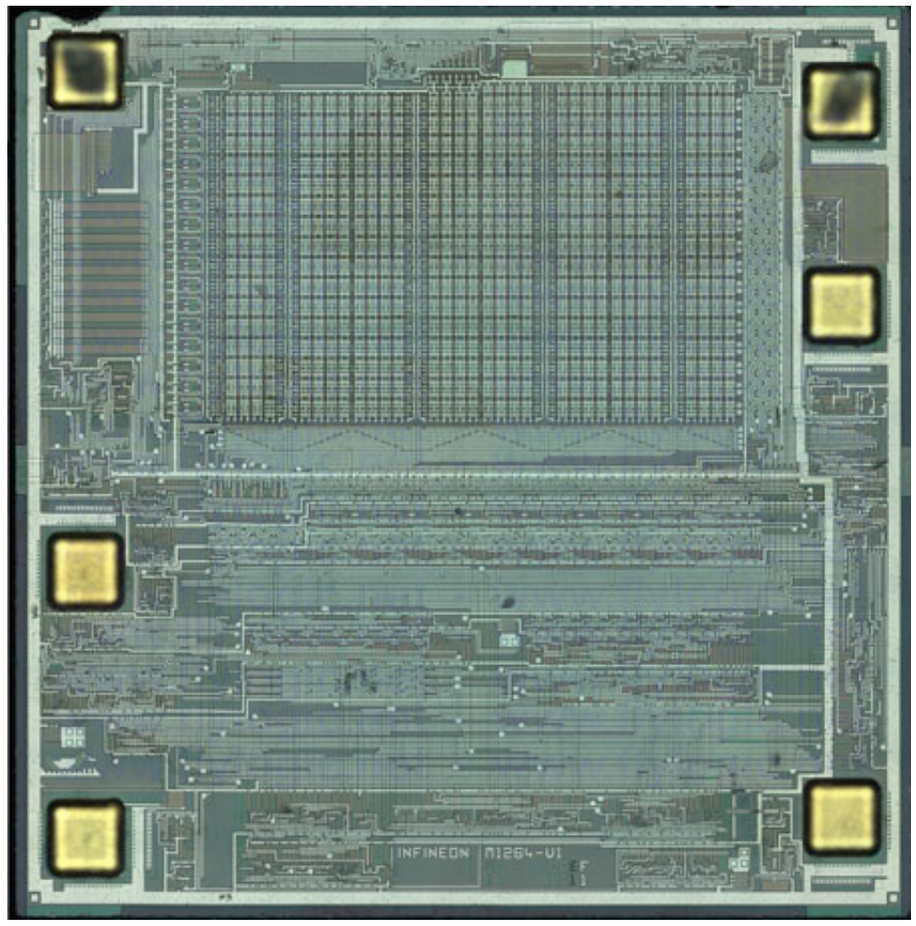
Remote controlled lawnmower

WESTERN NEW ENGLAND UNIVERSITY | WNE

How much money I have left?



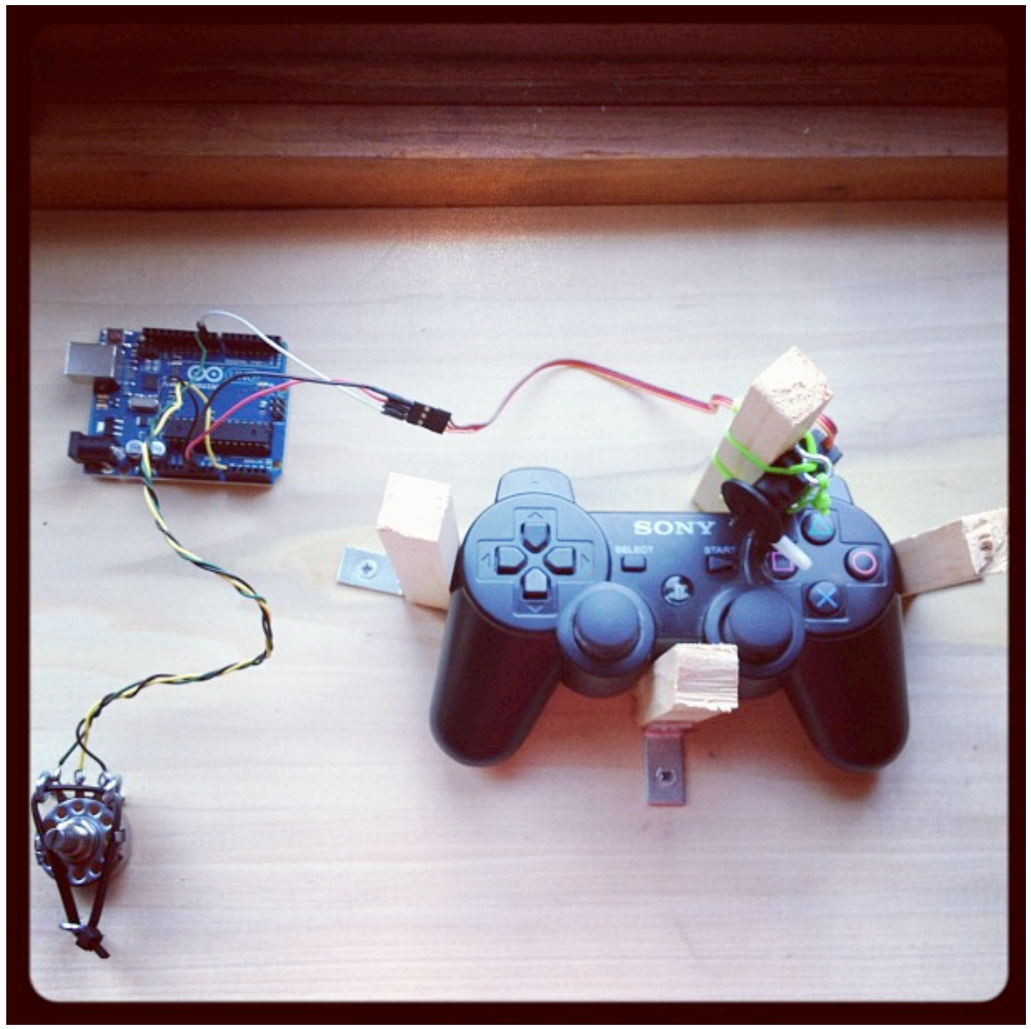




Reading smart cards

WESTERN NEW ENGLAND UNIVERSITY | WNE

What more fun that playing Castlevania?

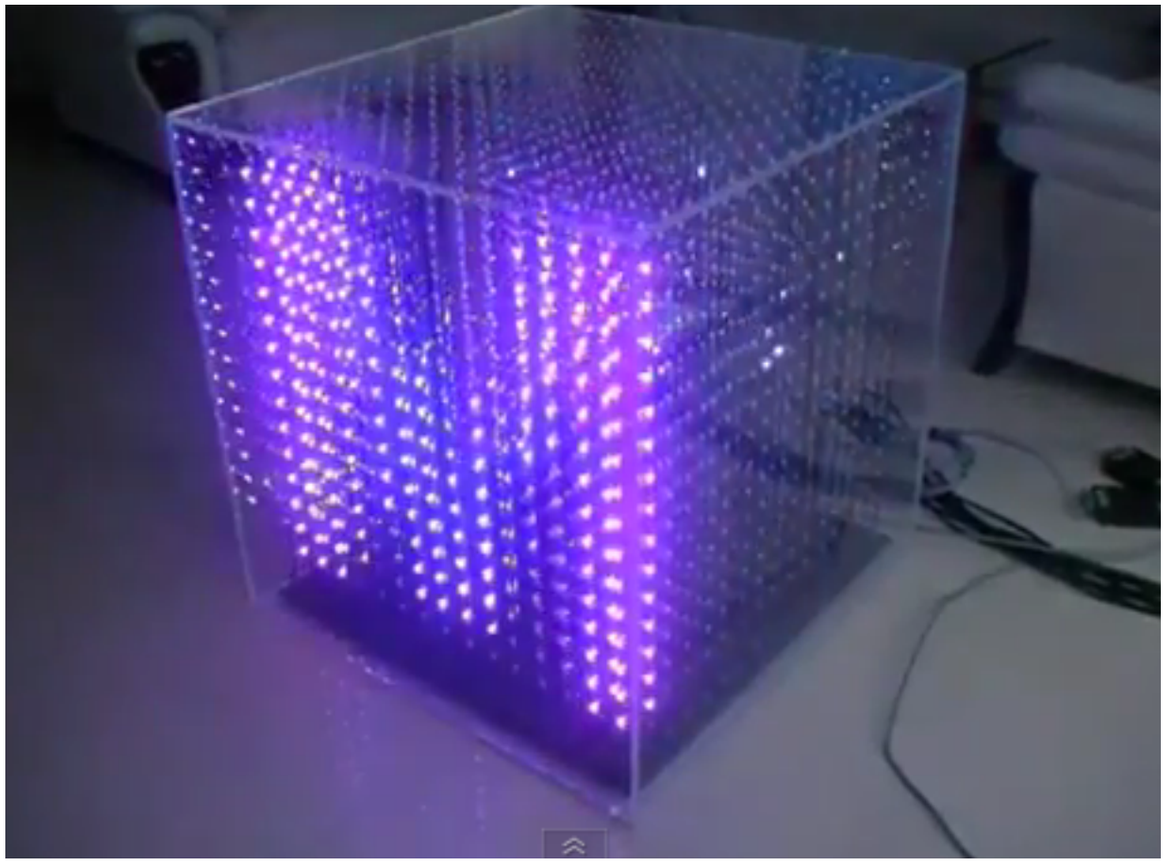


Grinding - Video

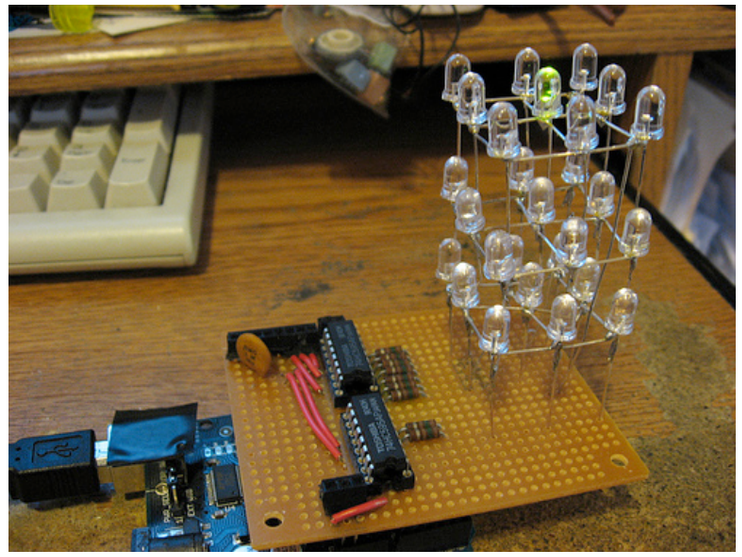
Automated gameplay

WESTERN NEW ENGLAND UNIVERSITY | WNE

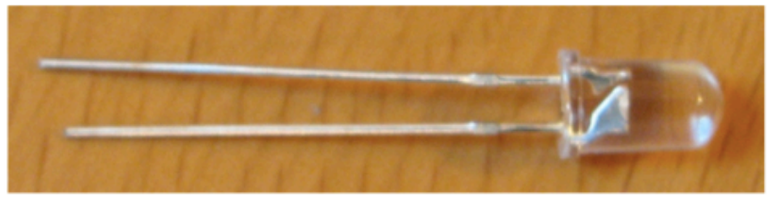
You can create your own 3D display.



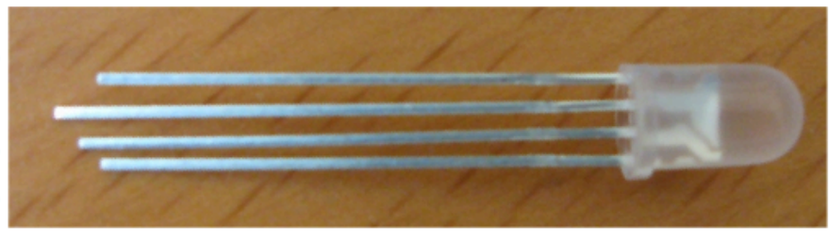
In action - Video



anode +
cathode –



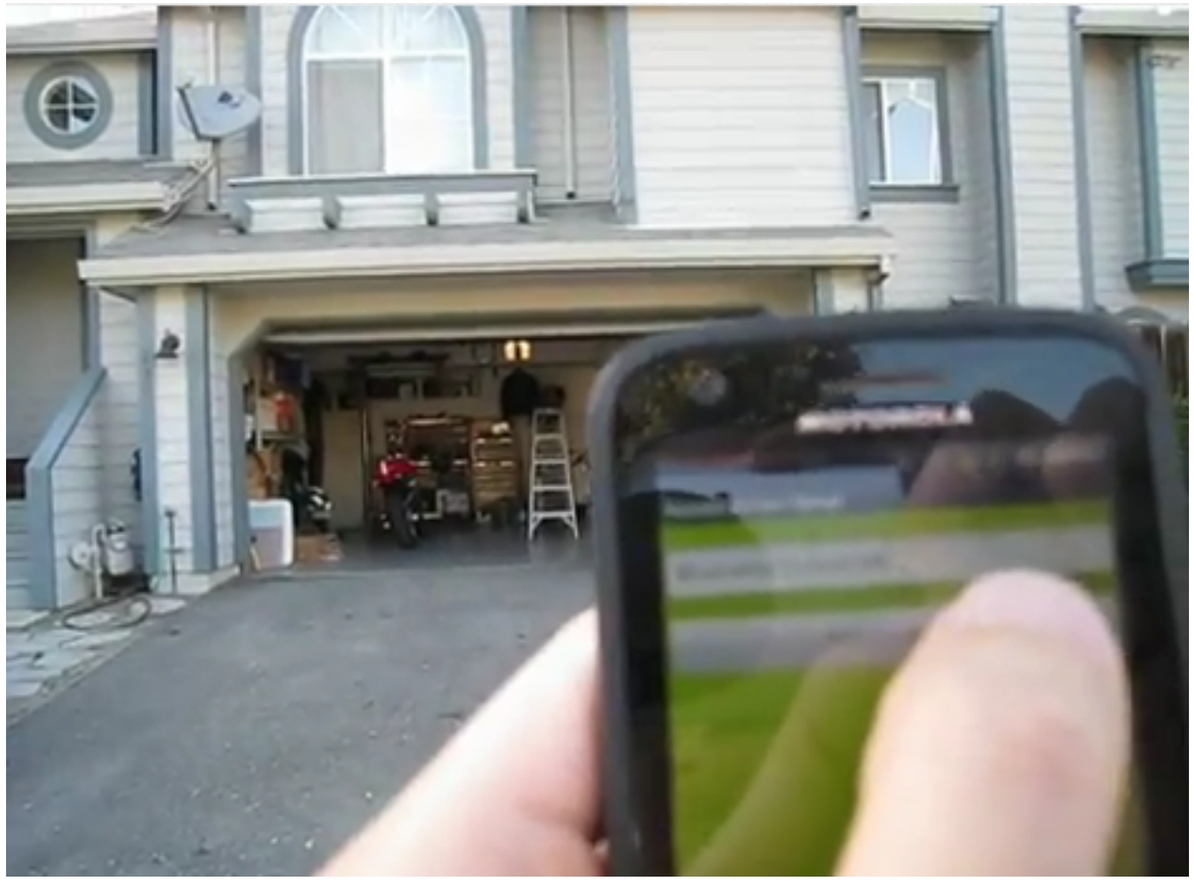
red cathode –
anode +
green cathode –
blue cathode –



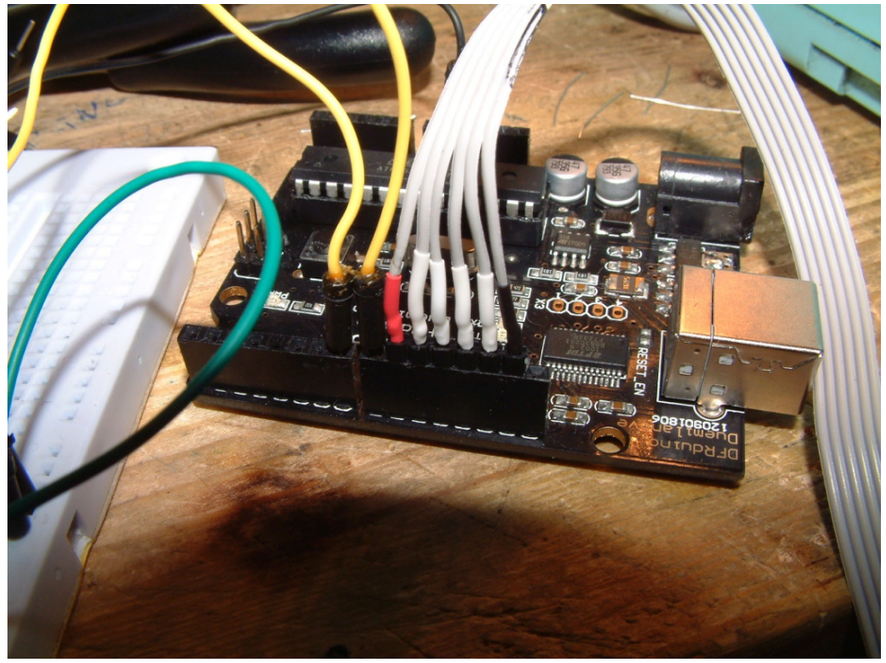
3D RGB LED Cube

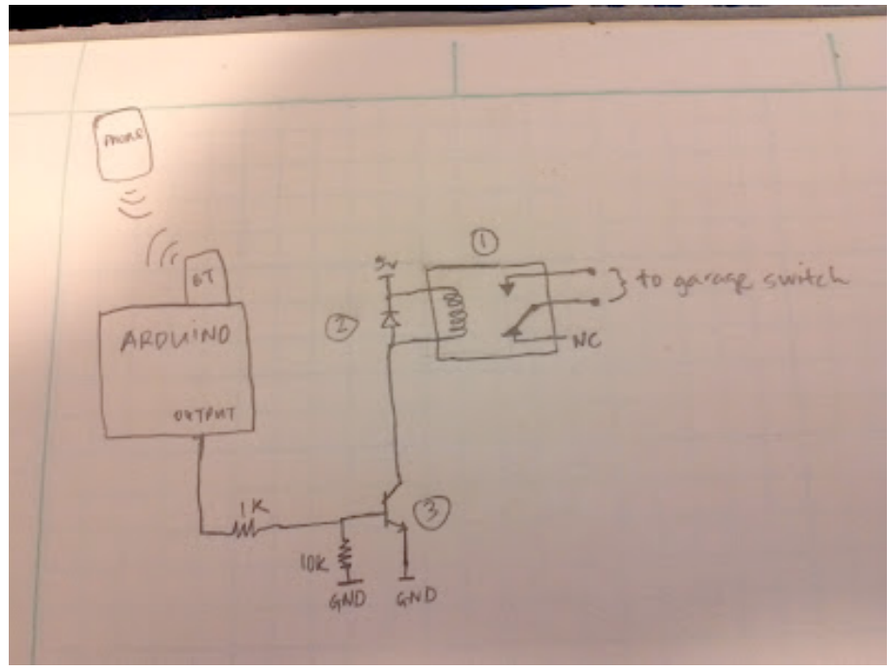
WESTERN NEW ENGLAND UNIVERSITY | WNE

I want an app to open my garage door!



In action - Video



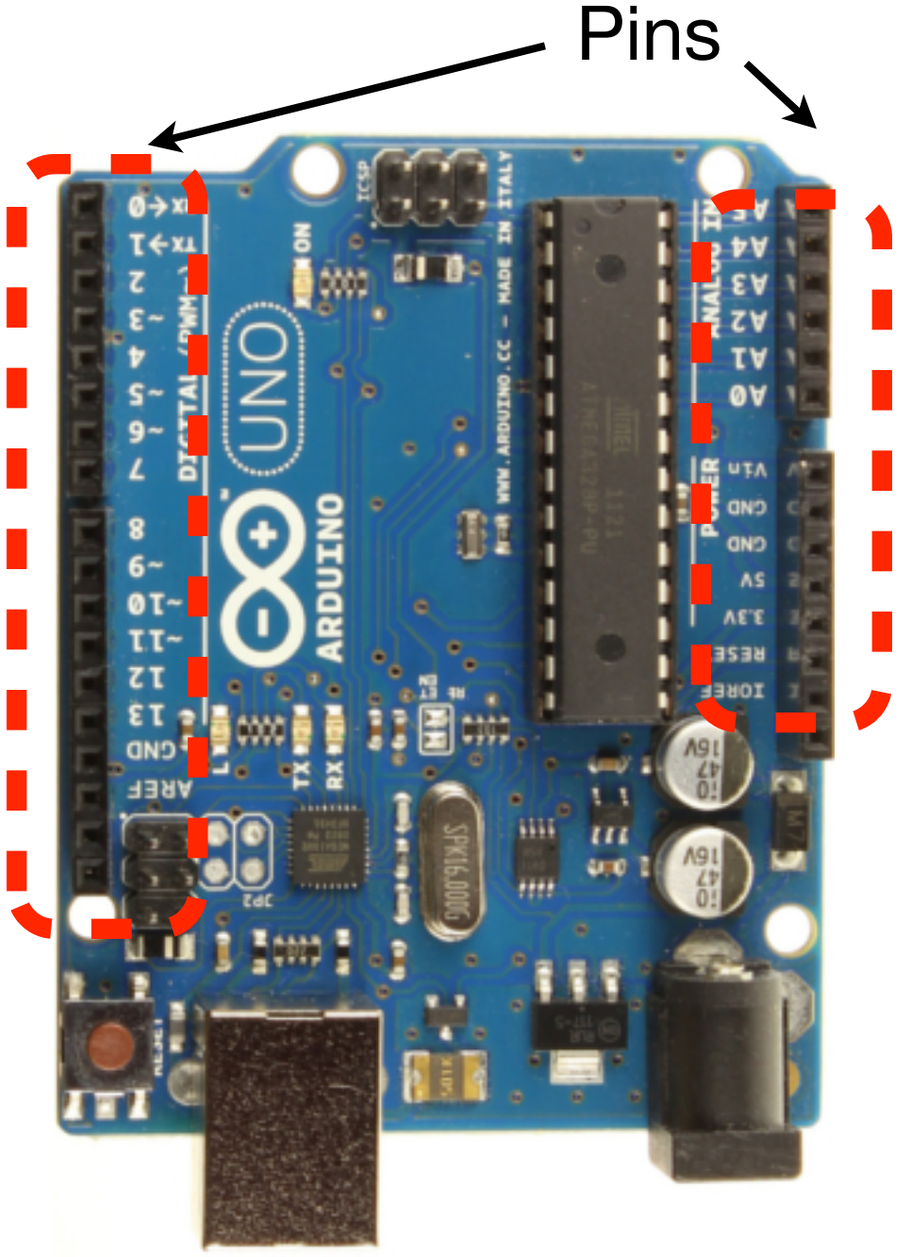


Garage door opener

WESTERN NEW ENGLAND UNIVERSITY | WNE

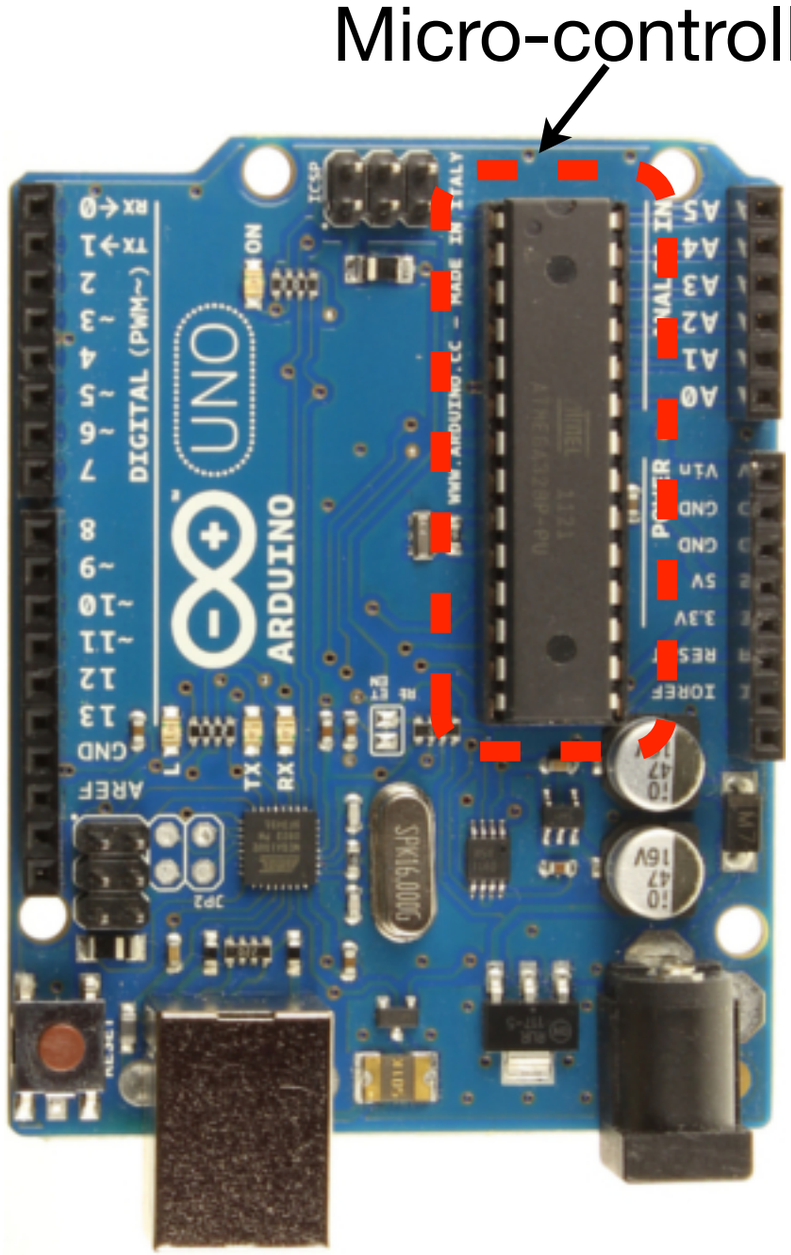
How? With a micro-controller...

- It's just a tiny computer.
- On your computer you program something (in the C language)
- You put that program into the board
- The micro-controller will run that program as long as it has power.
- Your program can:
  - Turn ON/OFF pins.
  - Read ON/OFF data from pins.

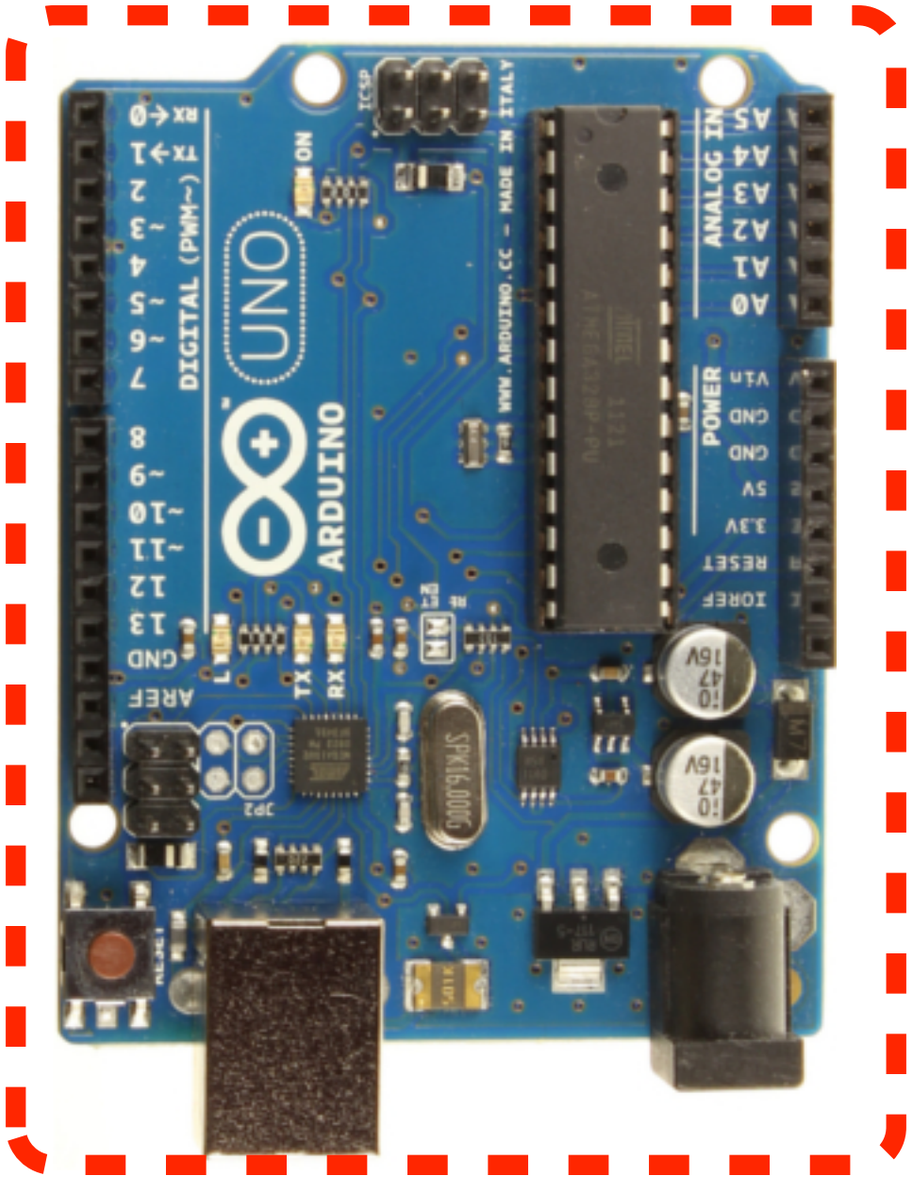


Micro-controller?

- Pretty slow (16MHz)
- Little memory.
- No operating system.
- No way you can put windows 7 in there.
- Micro-controllers are all over the place: cars, micro-waves, projectors, remote controls...
- ❖Look around... Where can you find micro-controllers in this room?



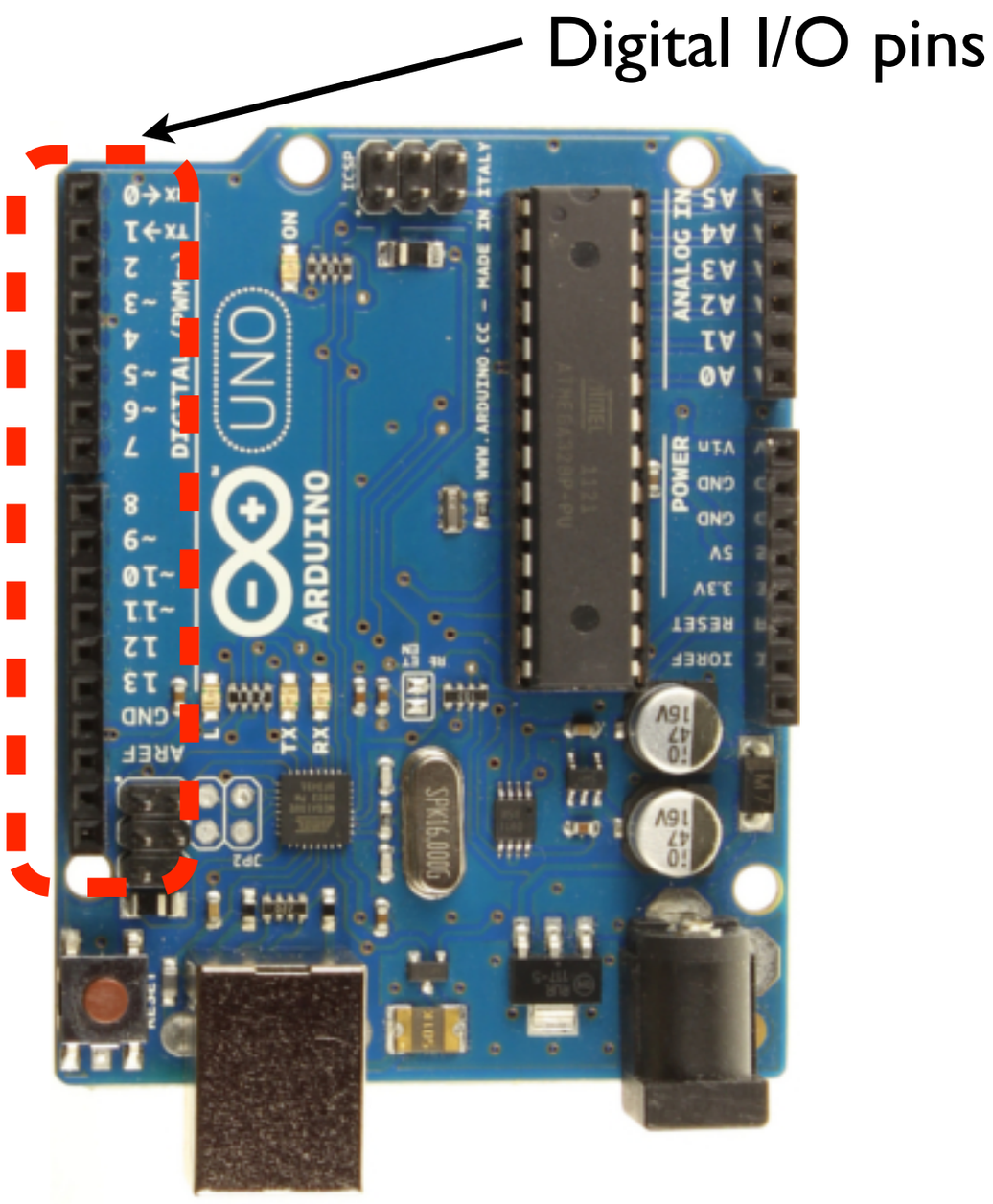
We will use the Arduino in our camp!



This board is called Arduino

Digital I/O with the Arduino

- Input/Output (I/O) is done through pins.
- Input = read stuff from the real-world.
- Output = write stuff to the real-world.
- You put a wire into a pin and connect it to something else.
- You say if a pin is **input** or **output**.
- A Digital pins can have two values: **HIGH** or **LOW**.



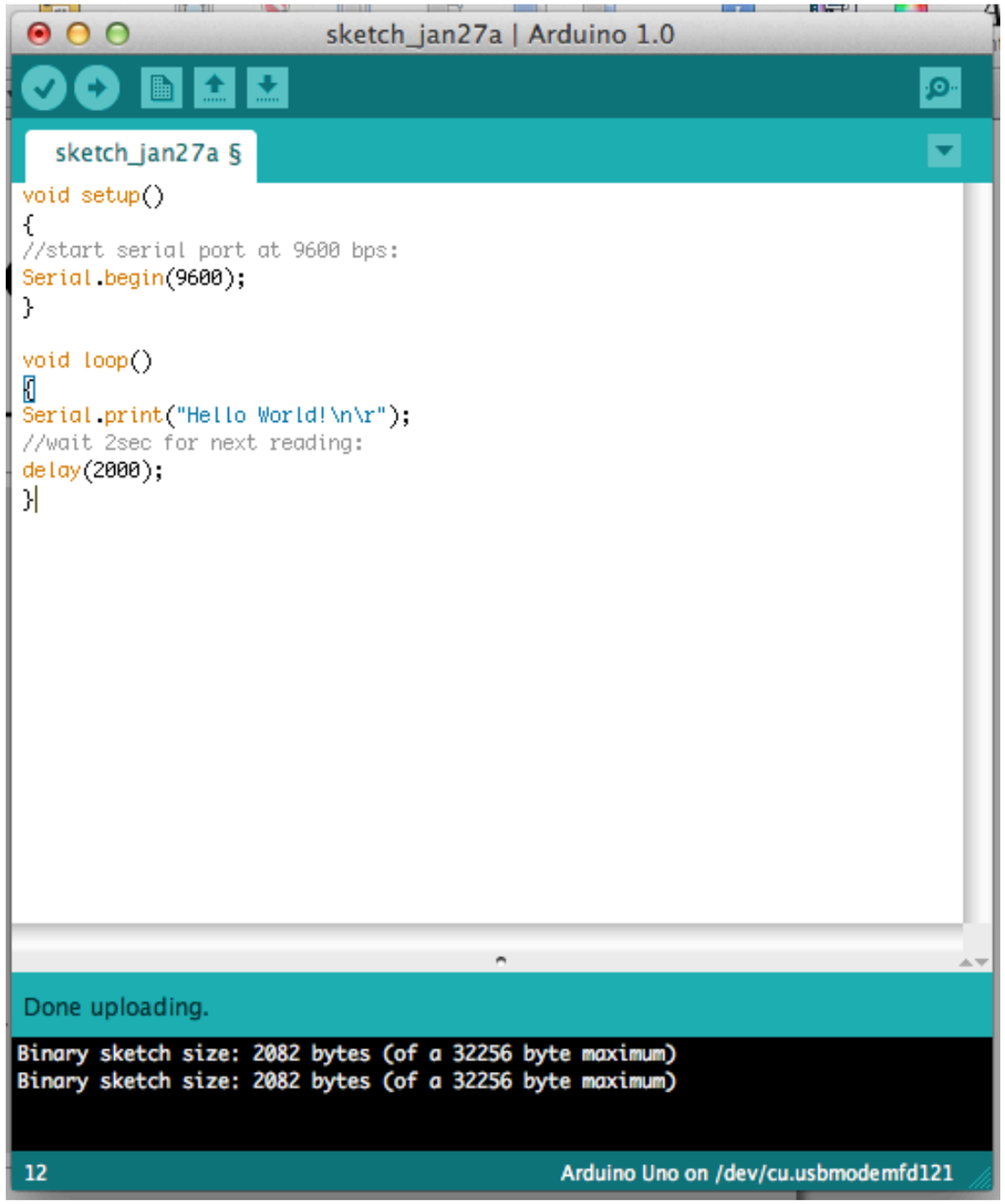
Put your hacker hat on!

- This means you are about to make something...
- Hackers help each others... Not going forward till everyone is done!
- A hacker is a person who modifies stuff.
- Building, rebuilding, modifying, and creating software, hardware.
- Other kids wish things... YOU make them!
- “I wish I could control my PS3 with my brain.” Well, YOU can do it!

Install the Arduino software

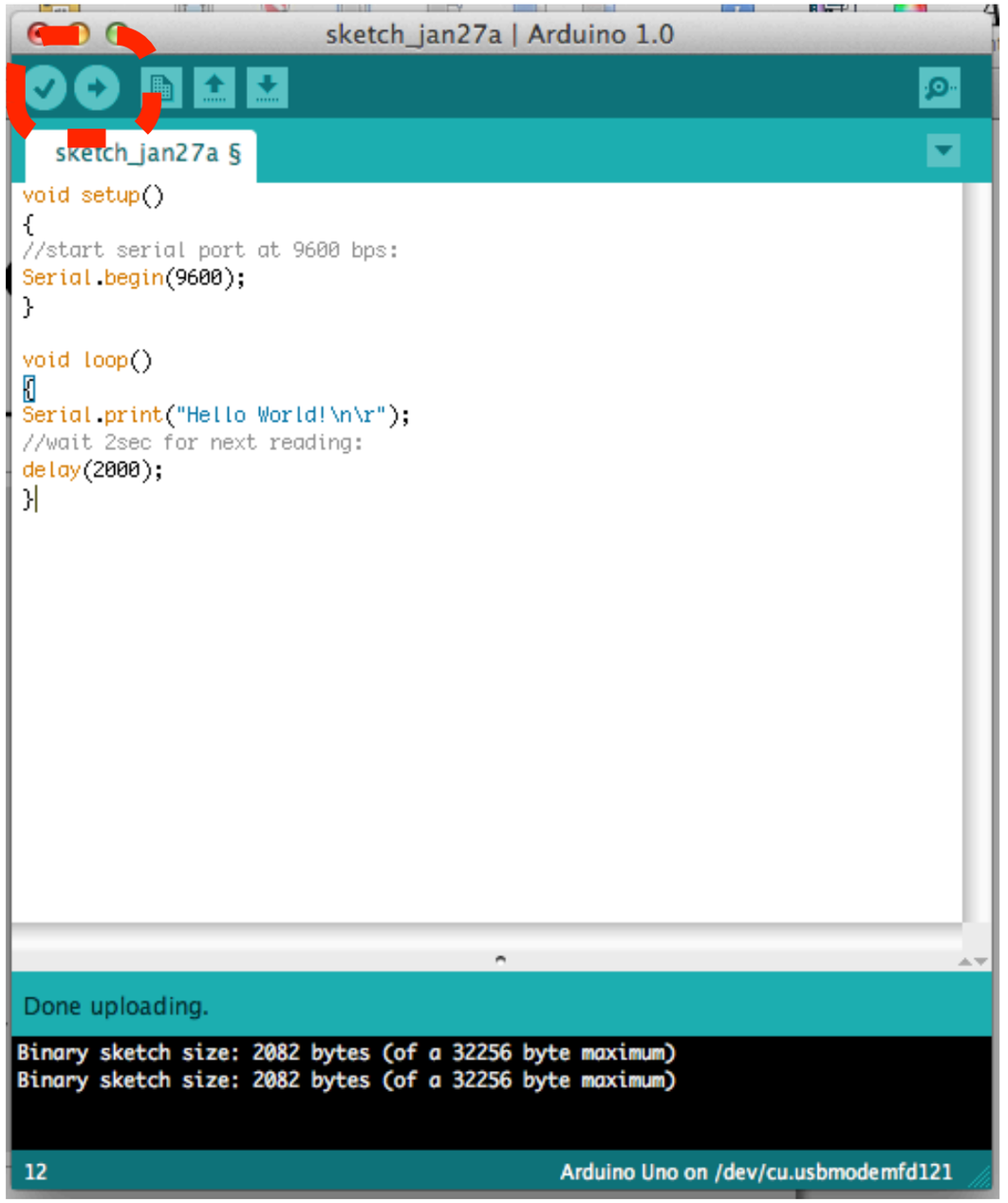
- http://arduino.cc
- http://arduino.cc/it/Guide/Windows

Yeah...read these instructions!



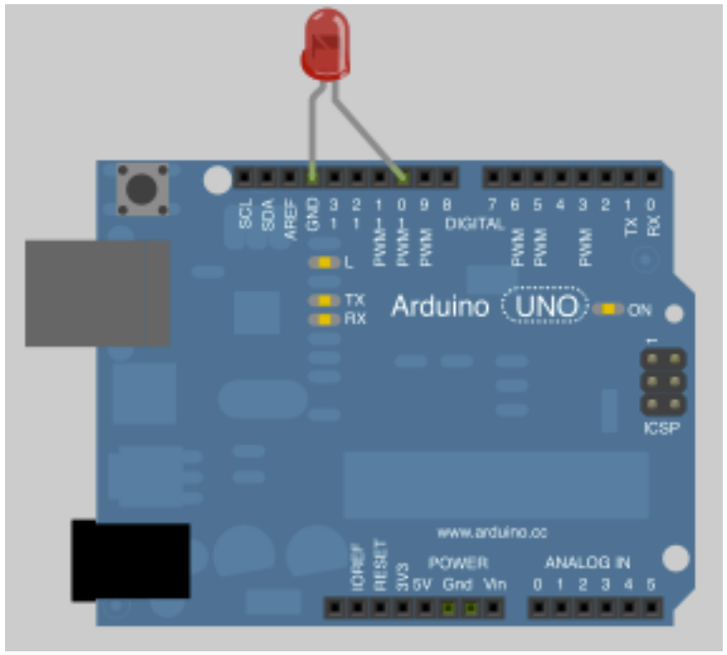
Arduino software

- You connect the Arduino into your computer with the USB cable.
- You write some stuff here.
- You press the arrow to send your stuff to the Arduino.



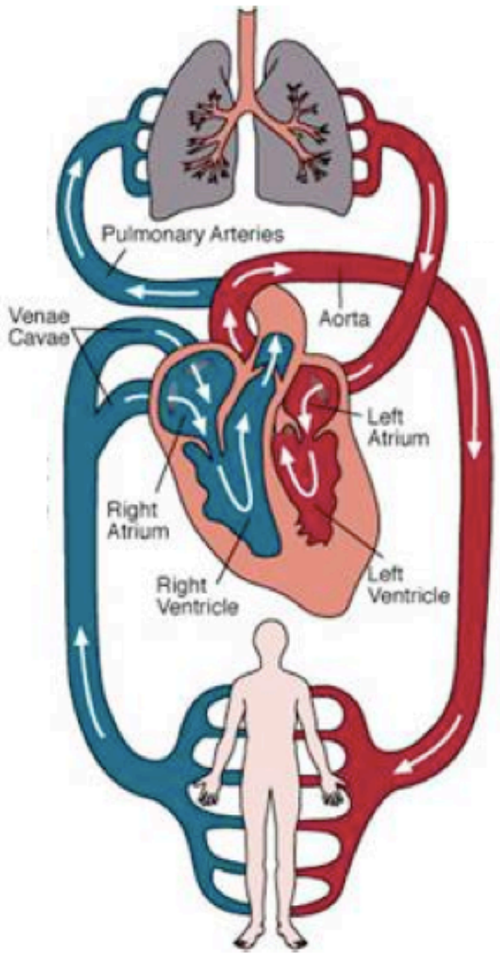
Turn ON/OFF an LED

```
int ledPin = 10;
void setup()
{
      pinMode(ledPin, OUTPUT);
}

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

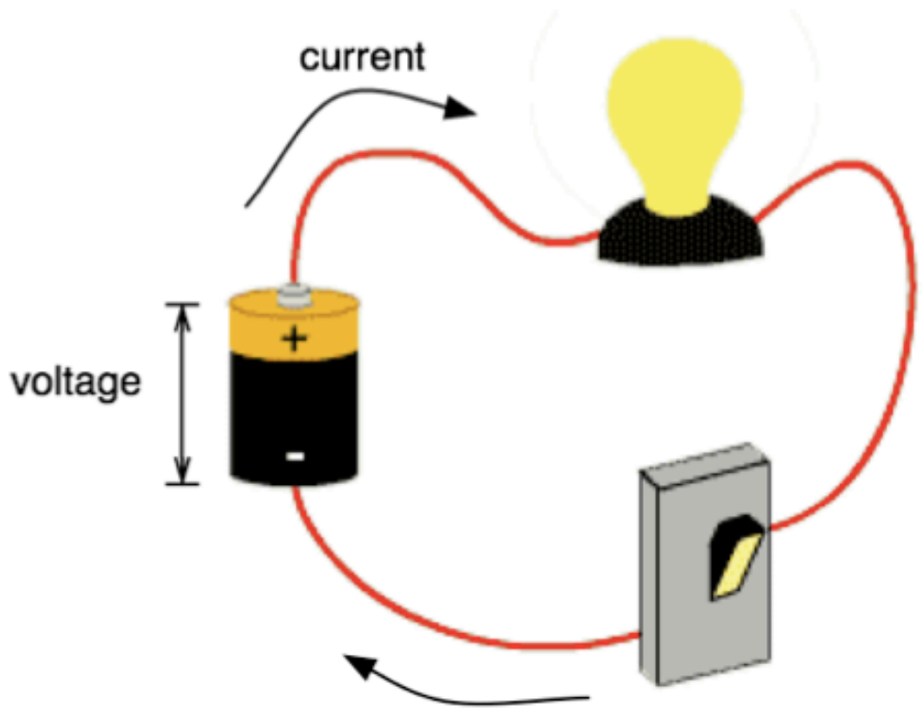


- Connect a LED to pin 10 and ground!
- Type the code.
- Make it blink faster!

What is an electrical circuit

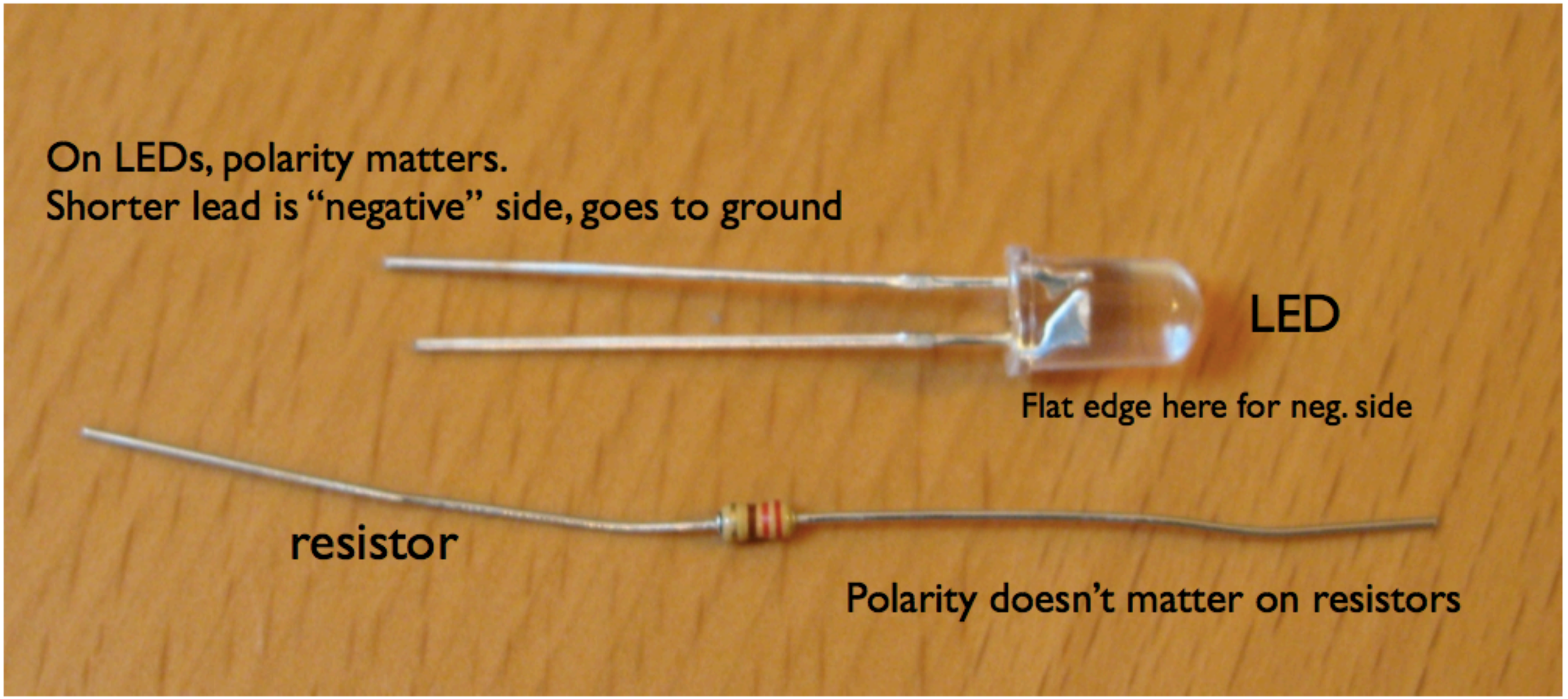


Heart pumps,
Blood flows

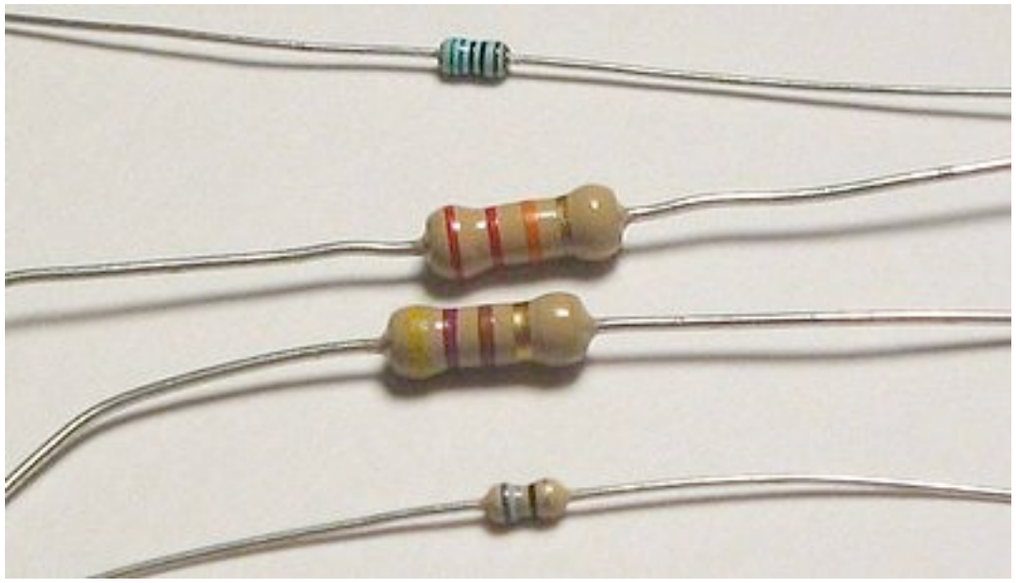


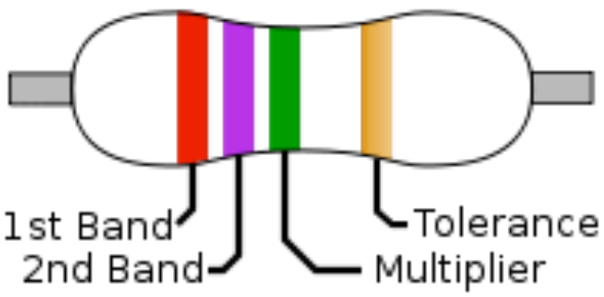
Voltage pushes,
Current flows

LEDs and resistors



Resistors?



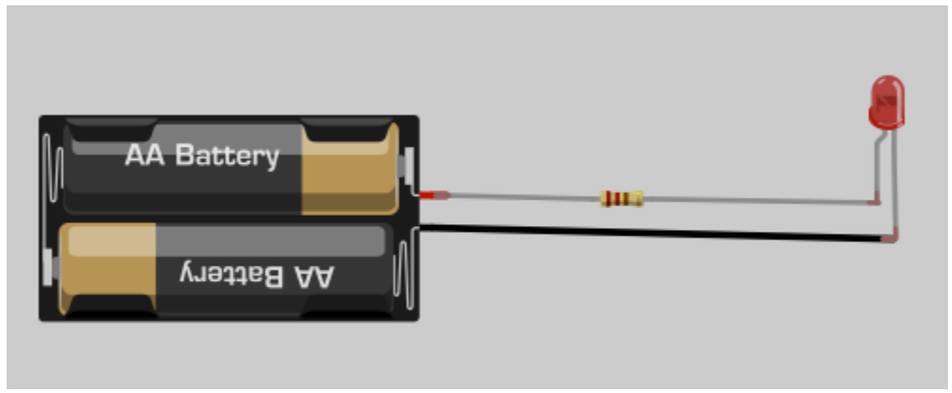


- Think of current as cars.
- Resistors are then bumps in the road.
- Big resistance, means lots of bumps.
- Lots of bumps, the cars (current) will slow down...

❖Why do we need to decrease current?

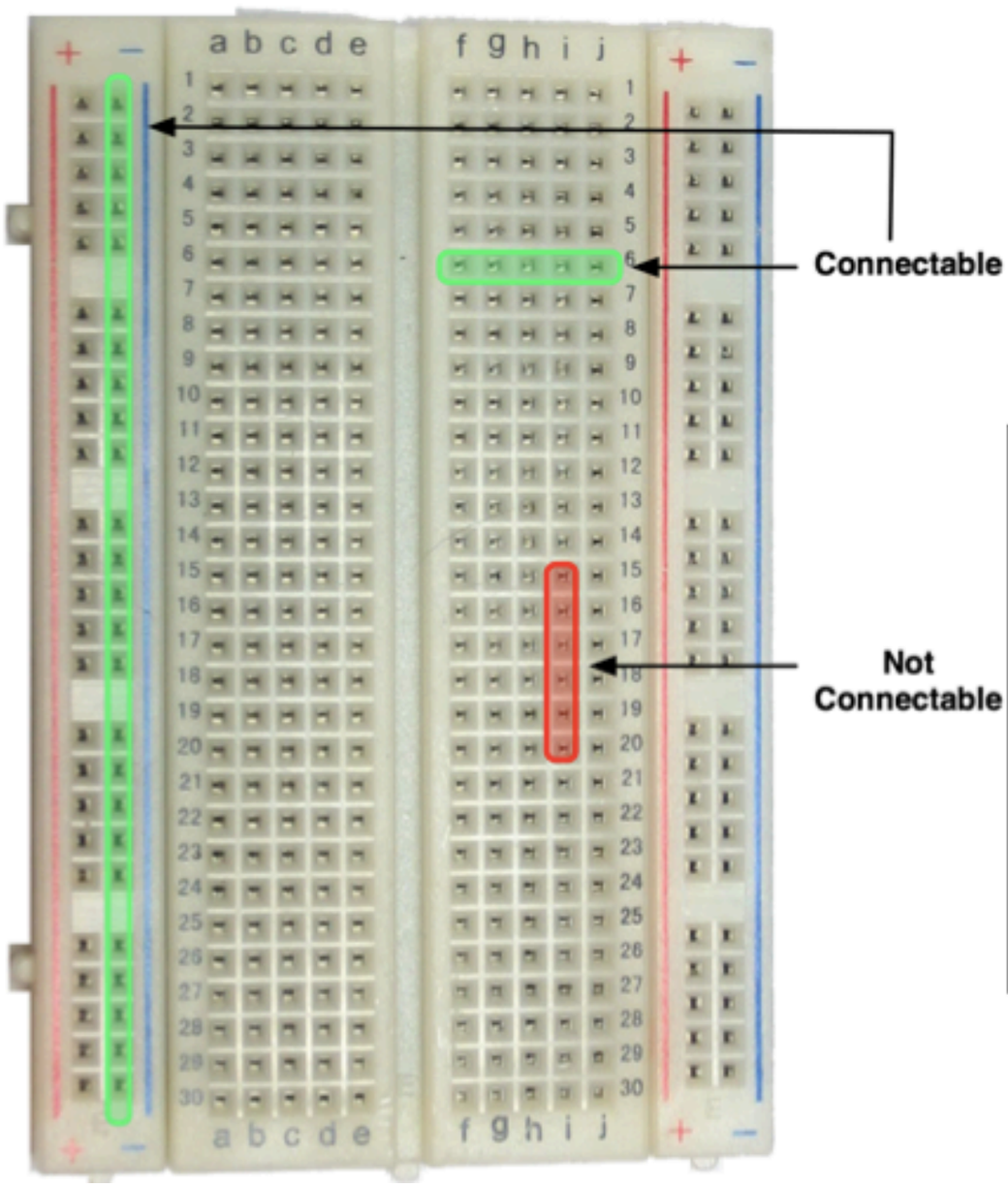
Lighting an LED

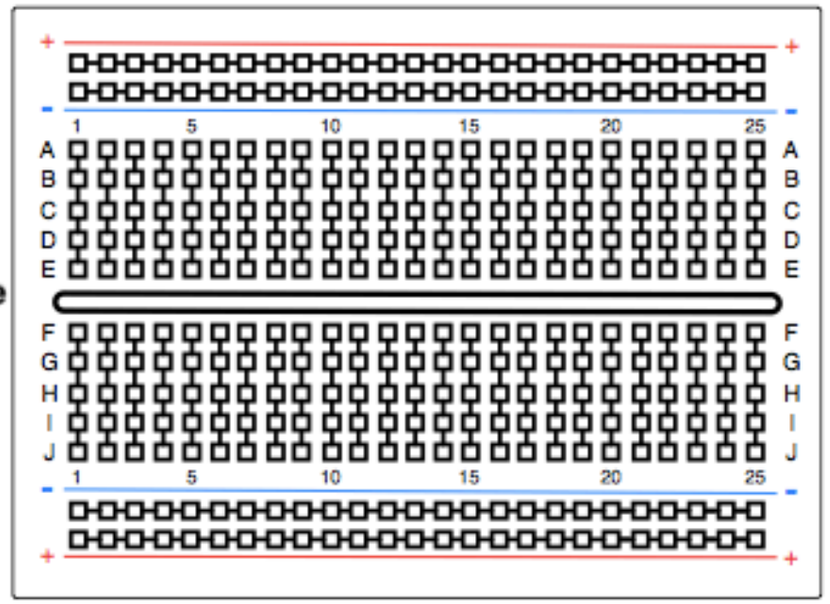
- A hacker **ALWAYS** questions everyone until he tries by himself.
- Too much current destroys.
- So... resistors protect the circuit by reducing the current.



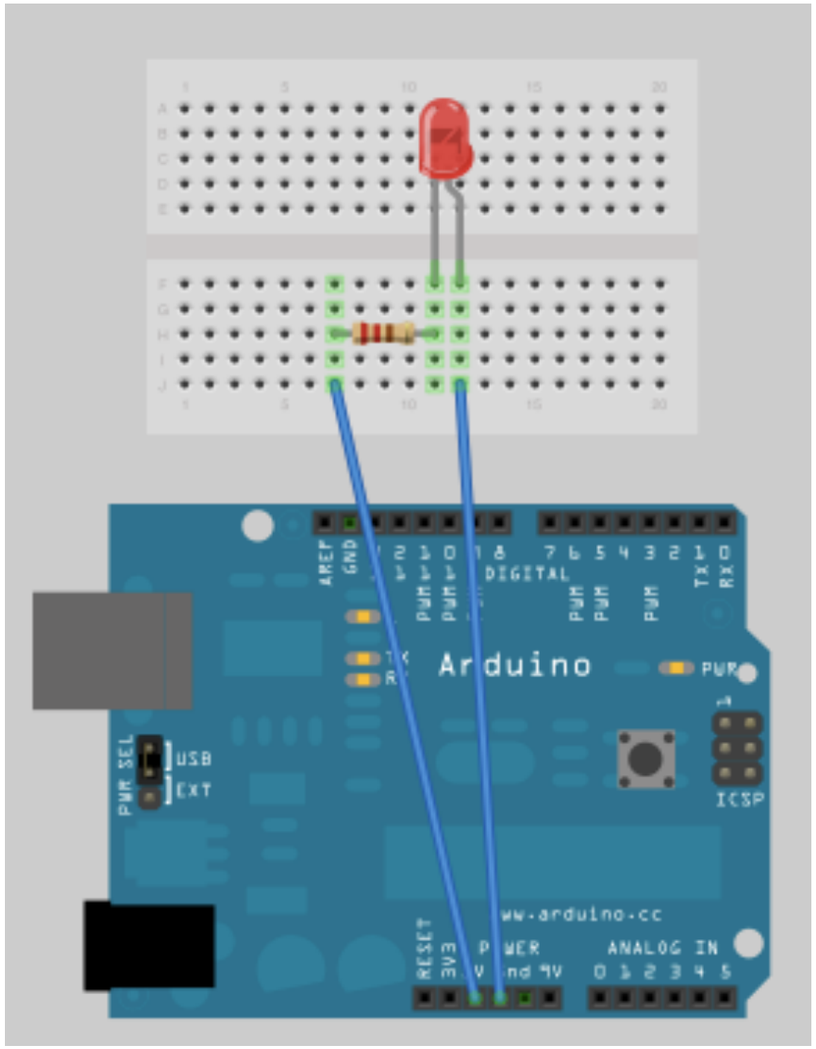
- Attach the LED into a battery without a resistor.
- Keep changing the resistors.
- Use a multi-meter to measure resistance.

We use solderless breadboards to hold circuits in place!

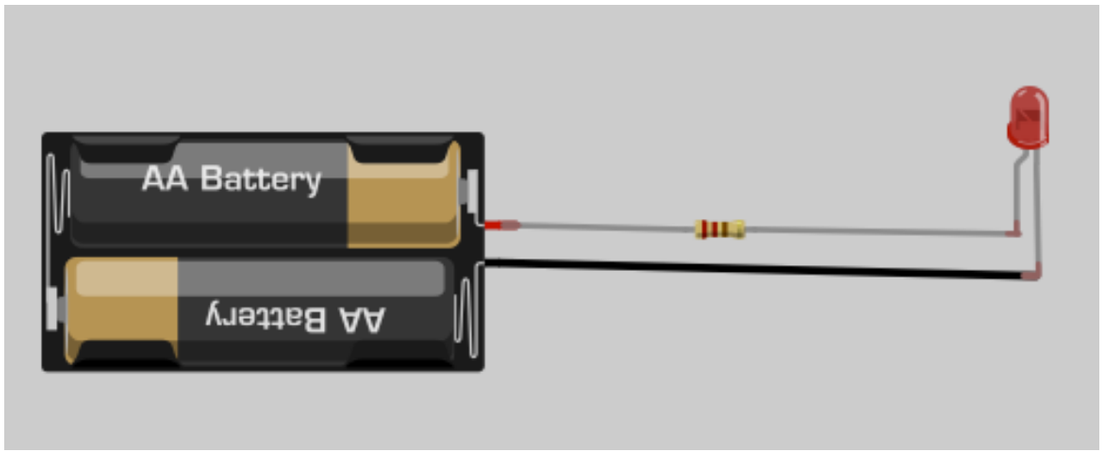




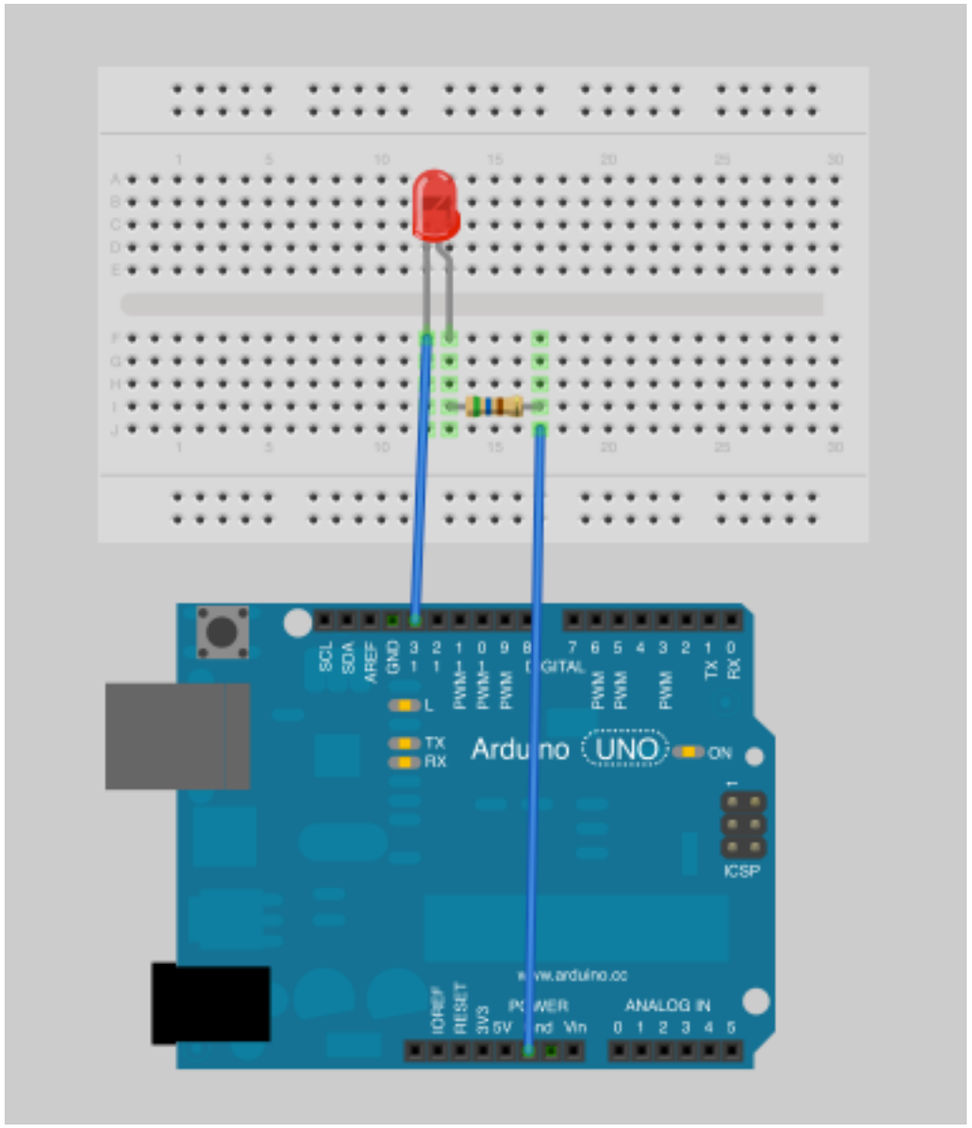
Playing with breadboards

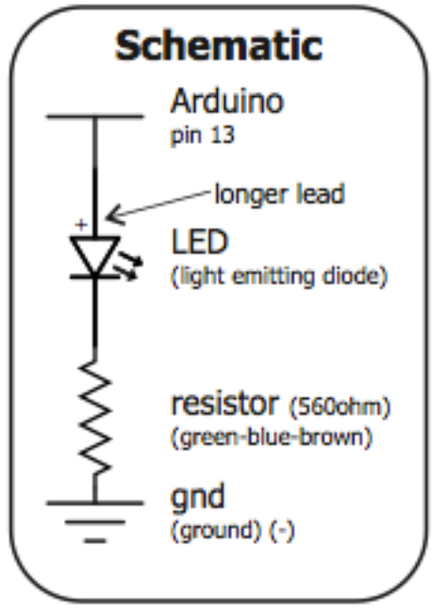


- Connect an LED and a resistor between 5V and GND.
- Pick a resistor that makes the LED really dim and blink it every 2 seconds on pin #3!



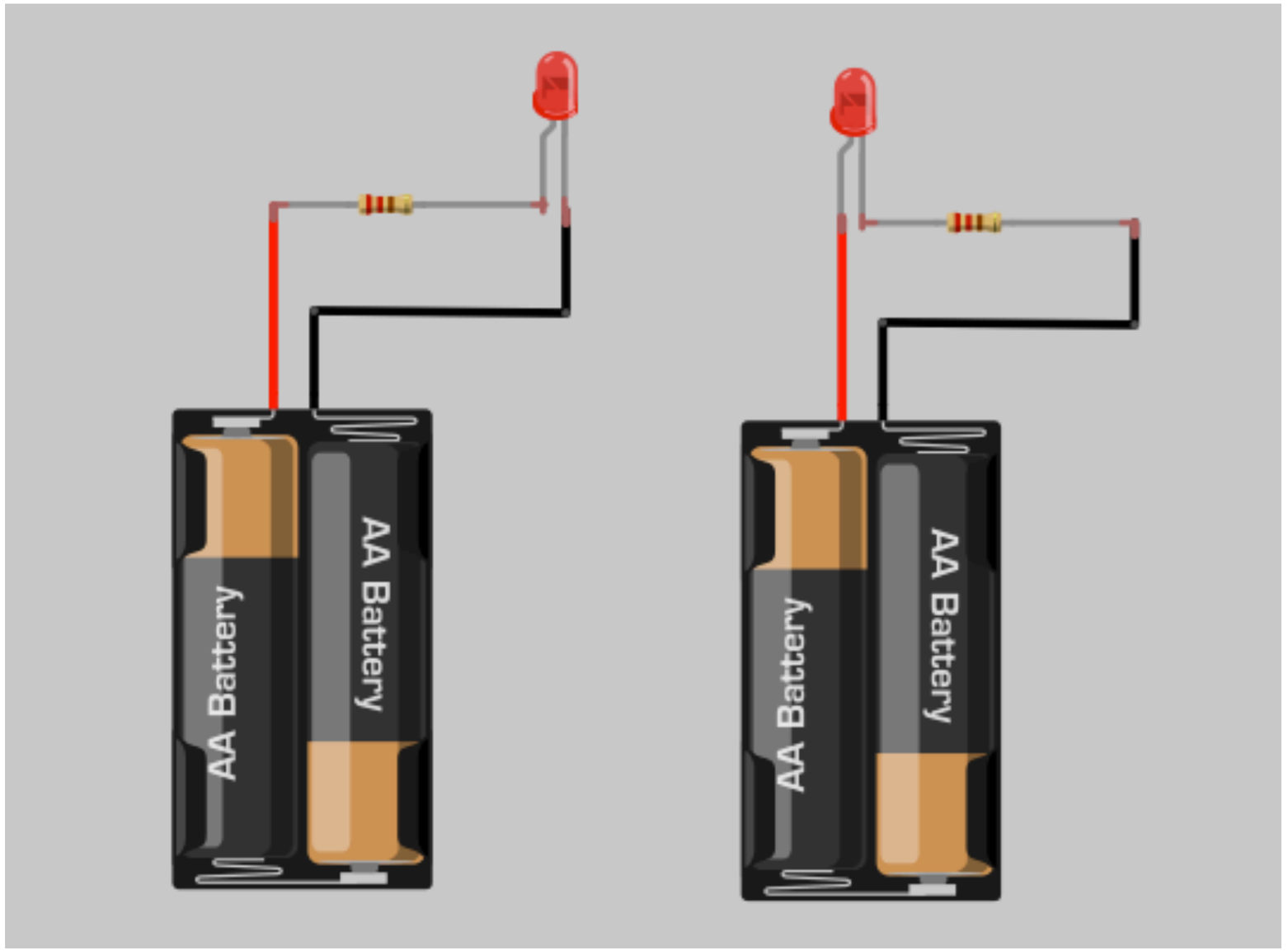
Always add resistors





- The schematic and wiring diagram show the same thing.
- Write the code that will blink the LED on pin13 every 5 seconds.

There is no difference between these two circuits



Arduino programming flow

Write code in program

↓

Connect USB to Board

↓

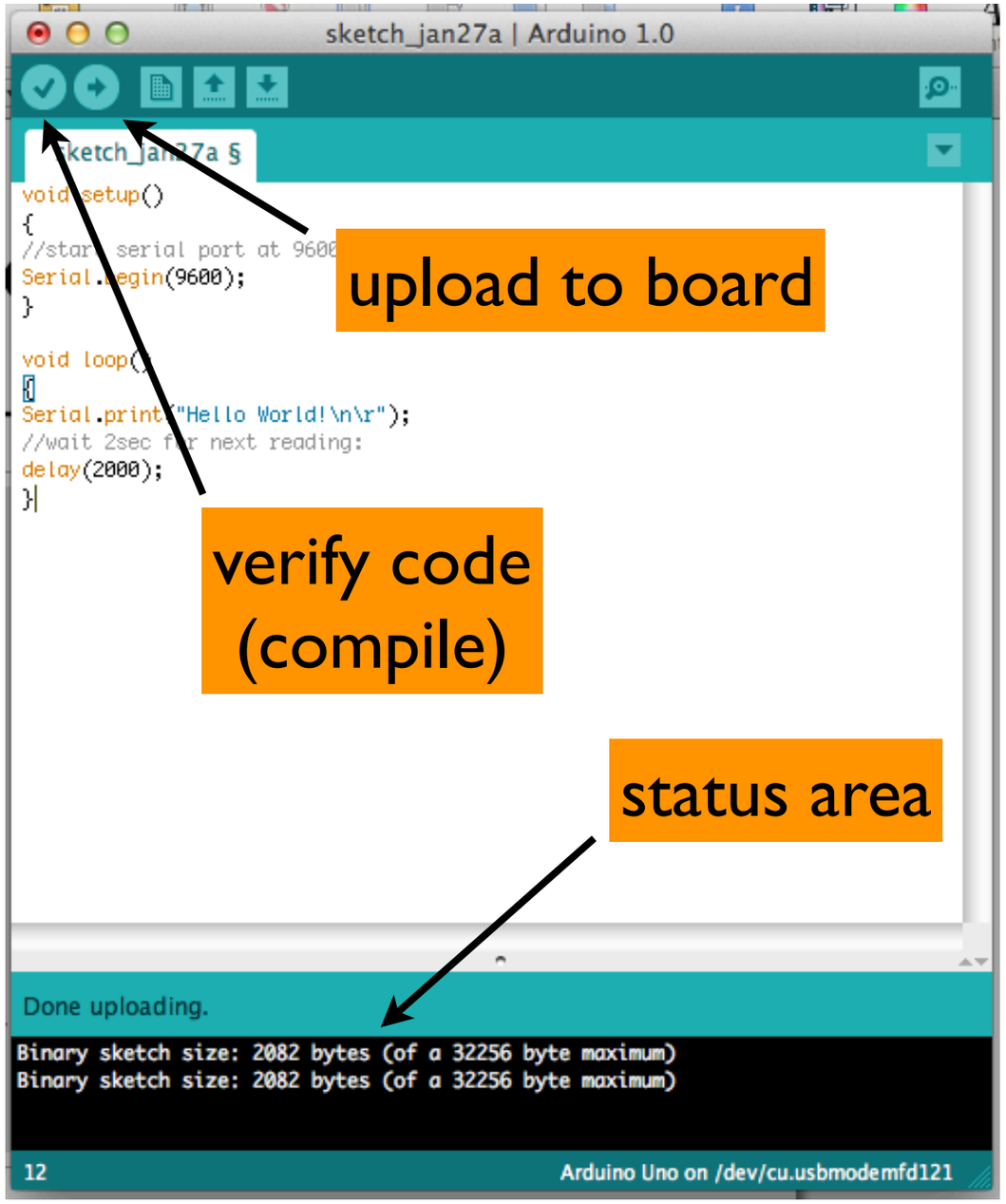
Upload Code

↓

Board LED blinks fast

↓

Your code runs



Anatomy of a “sketch”

Global Variables

Declare variables on top...
a variable is like a box of stuff.

Setup()

Stuff that is executed only once at beginning (eg: set pins)

Loop()

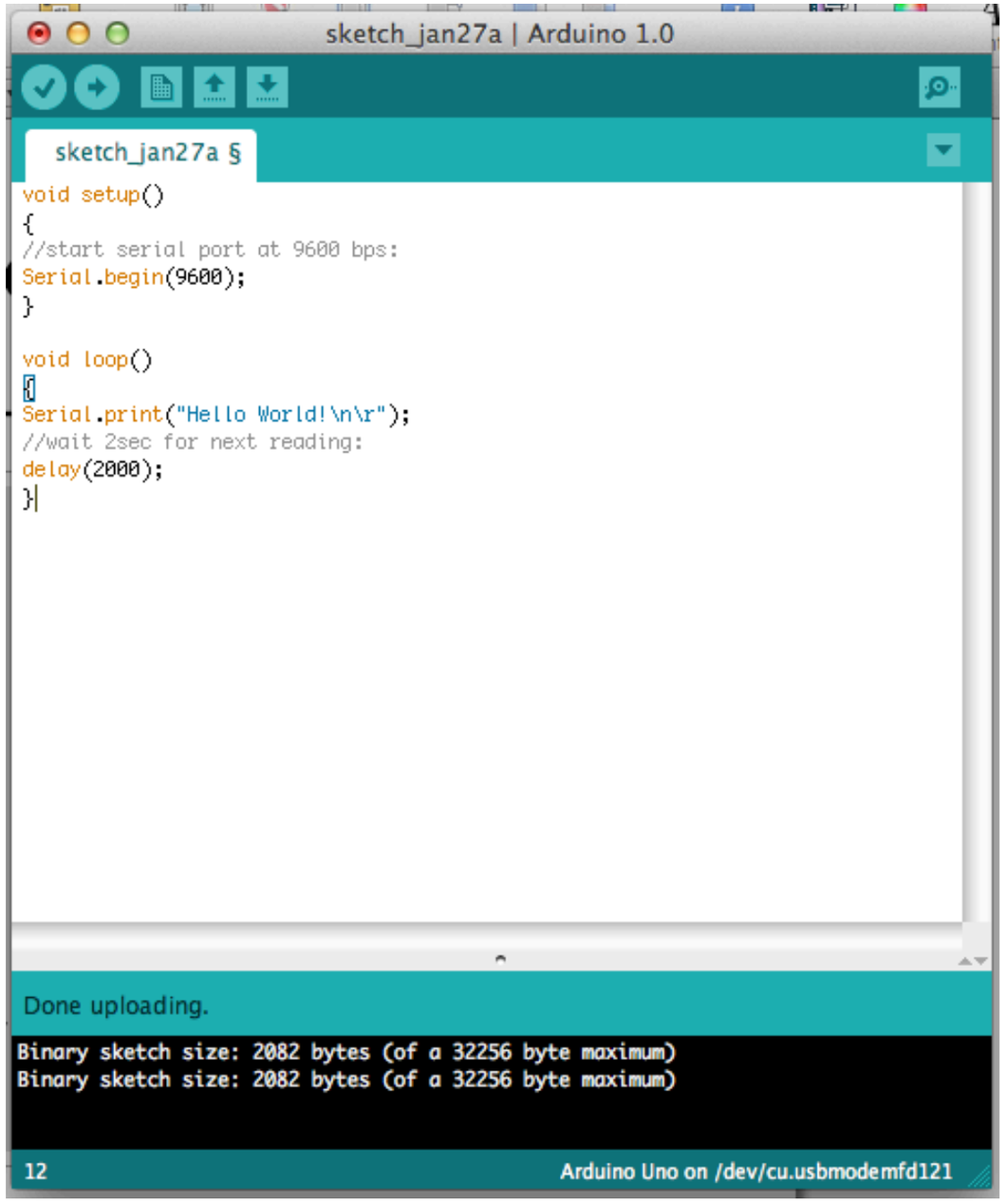
Stuff that runs repeatedly, after setup()

Arduino language

You can tell the Arduino to do lots of useful stuff.

This is done with functions:

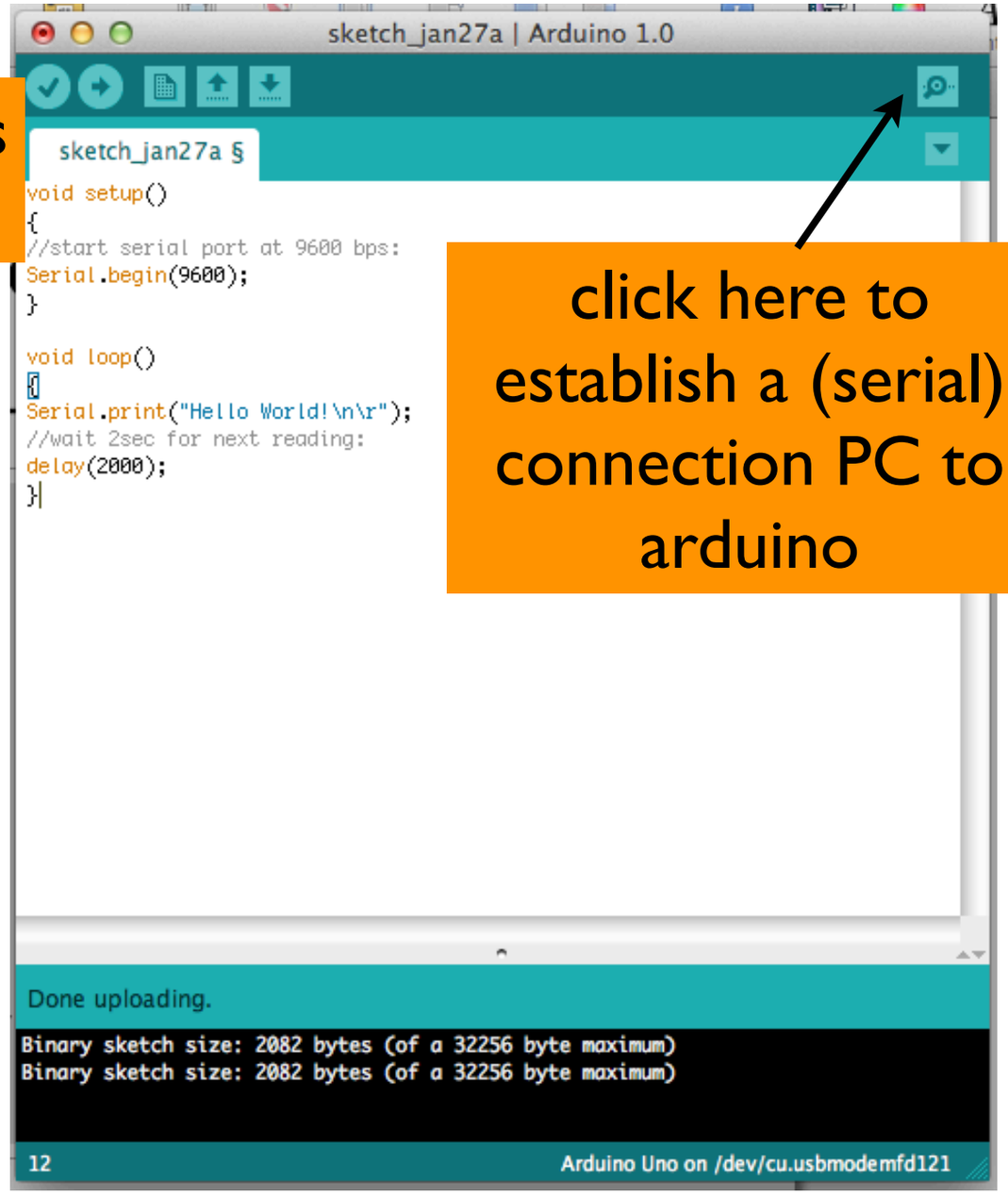
- *pinMode()* – set a pin as input or output
- *digitalWrite()* – set a digital pin high/low
- *digitalRead()* – read a digital pin's state
- *delay()* – wait an amount of time
- *millis()* – get the current time



“Hello World”

```
void setup()
{
  //start serial port at 9600 bps
  Serial.begin(9600);
}

void loop()
{
  //print something on the computer
  Serial.print("Hello World!\n");
  //wait 2sec
  delay(2000);
}
```

the Arduino ignores anything after //



click here to establish a (serial) connection PC to arduino

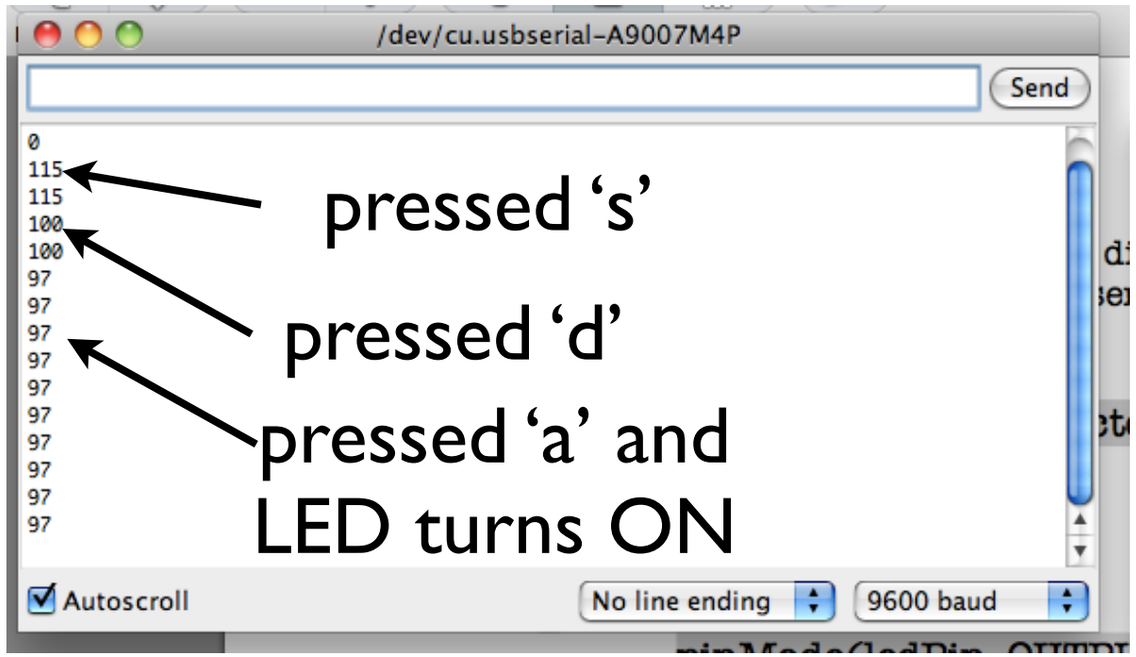
ASCII table and serial connections

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

- It is possible to control the Arduino through a serial connection
- Every time you press a key, the computer converts it into a number
- Each key has a different number and this is called ASCII.
- We can send that number to the Arduino through the cable.

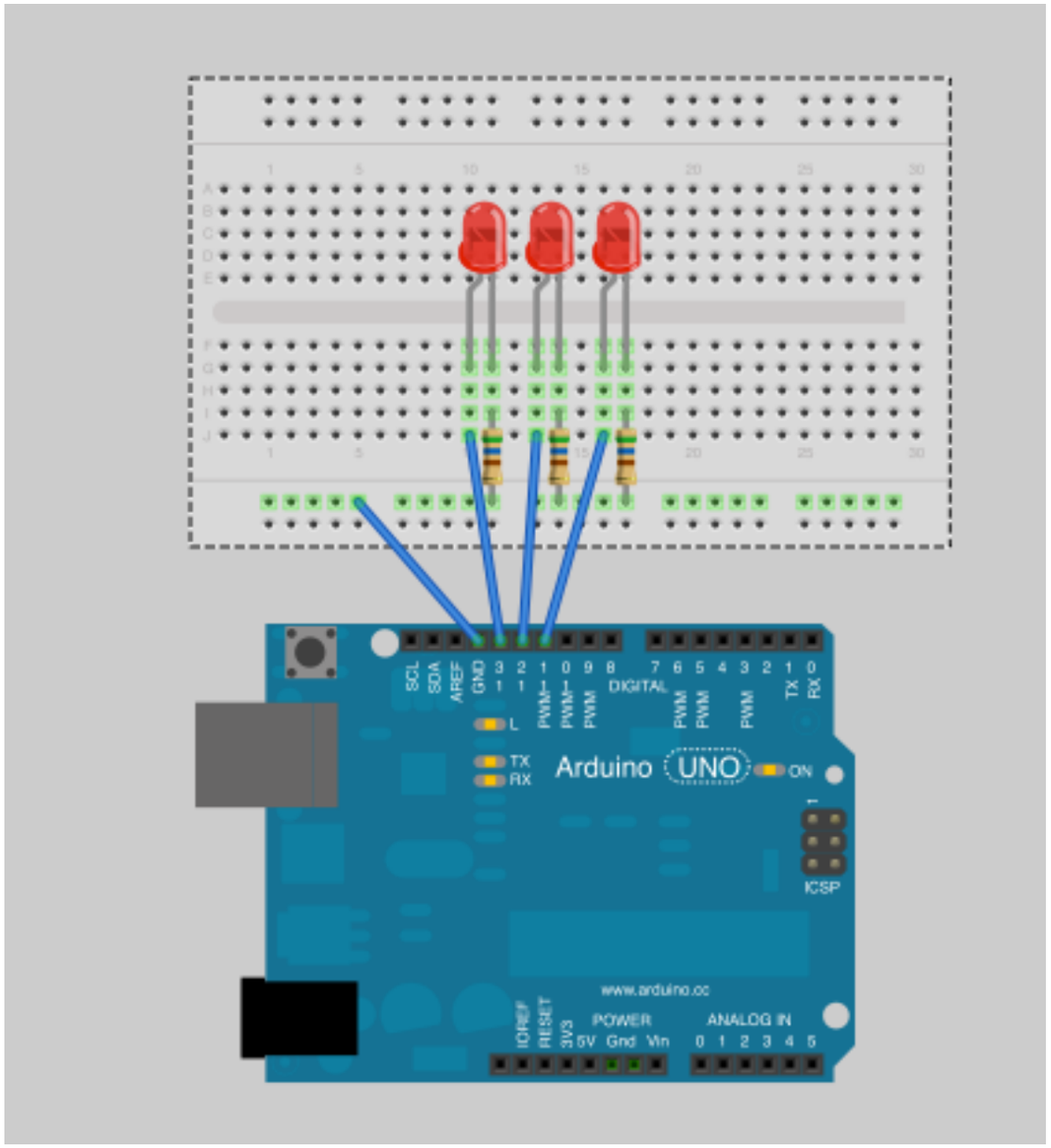
Character ‘a’ turns on LED

```
int ledPin = 10; int inByte = 0;
void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
   while (Serial.available()>0)
   { inByte = Serial.read(); }

   Serial.println(inByte);

   if (inByte==97)
   { digitalWrite(ledPin, HIGH); }
   else
   { digitalWrite(ledPin, LOW); }
   delay(1000);
}
```



- Attach a LED to pin 10 and run this code.
- Now attach another LED to pin 7 and modify the code.
- Whenever ‘b’ is pressed, this new LED should go ON.

Dealing with 3 LEDs



- Put this circuit together

  When:

  - ‘a’ is pressed, onlyLED#1 is ON
  - ‘b’ is pressed, only LED#2 is ON
  - ‘c’ is pressed, only LED #3 is ON

Dealing with 3 LEDs (possible code)

```
int ledPinA = 11, ledPinB = 12,
ledPinC = 13;
int inByte = 0;

void setup()
{
  pinMode(ledPinA, OUTPUT);
  pinMode(ledPinB, OUTPUT);
  pinMode(ledPinC, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop(){
  while (Serial.available()>0)
  { inByte = Serial.read(); }
  Serial.println(inByte);

   if (inByte==97) {
    digitalWrite(ledPinA, HIGH);
    digitalWrite(ledPinB, LOW);
    digitalWrite(ledPinC, LOW);}
  if (inByte==98) {
    digitalWrite(ledPinA, LOW);
    digitalWrite(ledPinB, HIGH);
    digitalWrite(ledPinC, LOW);}
  if (inByte==99) {
    digitalWrite(ledPinA, LOW);
    digitalWrite(ledPinB, LOW);
    digitalWrite(ledPinC, HIGH);}

  delay(1000);
}
```

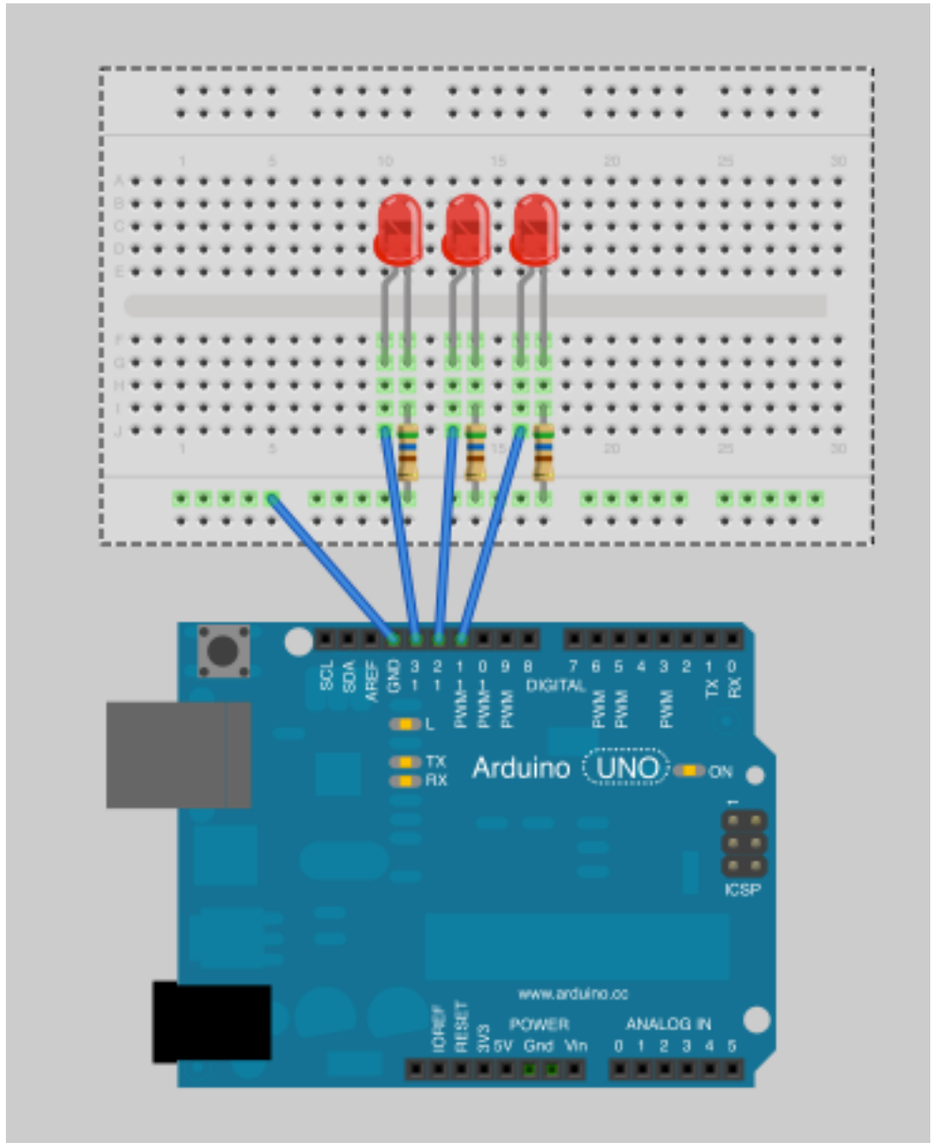
Random numbers

- The Arduino can give you random numbers.
- However, you must use a fancy function called *random*.
- What will the Arduino print out?

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
   int randNumber;
   //print a number from 0 to 299
   randNumber = random(300);
   Serial.println(randNumber);

   // print a number from 10 to 19
   randNumber = random(10, 20);
   Serial.println(randNumber);

   delay(1000);
}
```

Circuit with 3 LEDS

- Create a circuit with 3 LEDs in pins 11,12 and 13.
- If ‘1’ is pressed only one RANDOM LED will be ON.
- If ‘2’ is pressed at most two RANDOM LED’s will be ON.
- If ‘3’ is pressed ALL three LED’s will be ON.



Circuit with 3 LEDS (possible code)

```
int ledPinA = 11, ledPinB = 12;
int ledPinC = 13, inByte = 0;

void setup(){
  pinMode(ledPinA, OUTPUT);
  pinMode(ledPinB, OUTPUT);
  pinMode(ledPinC, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  while (Serial.available()>0)
  { inByte = Serial.read(); }
  Serial.println(inByte);

  if (inByte==49) {
    digitalWrite(ledPinA, LOW);
    digitalWrite(ledPinB, LOW);
    digitalWrite(ledPinC, LOW);
    int rnd = random(11,14);
    digitalWrite(rnd, HIGH);
  }
  if (inByte==50) {
    digitalWrite(ledPinA, LOW);
    digitalWrite(ledPinB, LOW);
    digitalWrite(ledPinC, LOW);
    digitalWrite(random(11,14), HIGH);
    digitalWrite(random(11,14), HIGH);
  }

  if (inByte==51) {
    digitalWrite(ledPinA, HIGH);
    digitalWrite(ledPinB, HIGH);
    digitalWrite(ledPinC, HIGH);
  }

   delay(1000);
}
```

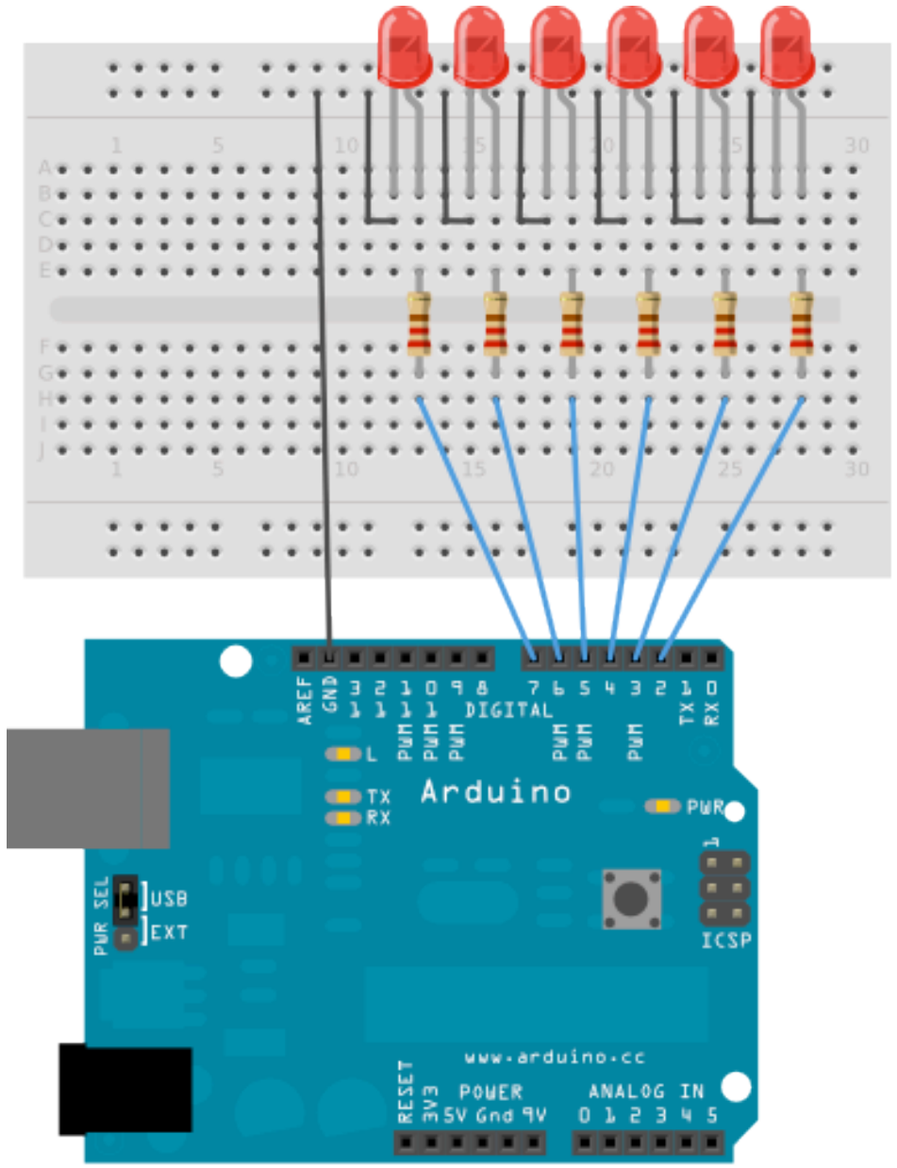
Cylon

Video - #1 #2

- BattleStar Gallatica is an awesome TV show.
- A group of robots are trying to destroy all humans.
- Their eyes are LEDS that move in a neat pattern.
- We can do that pattern too!

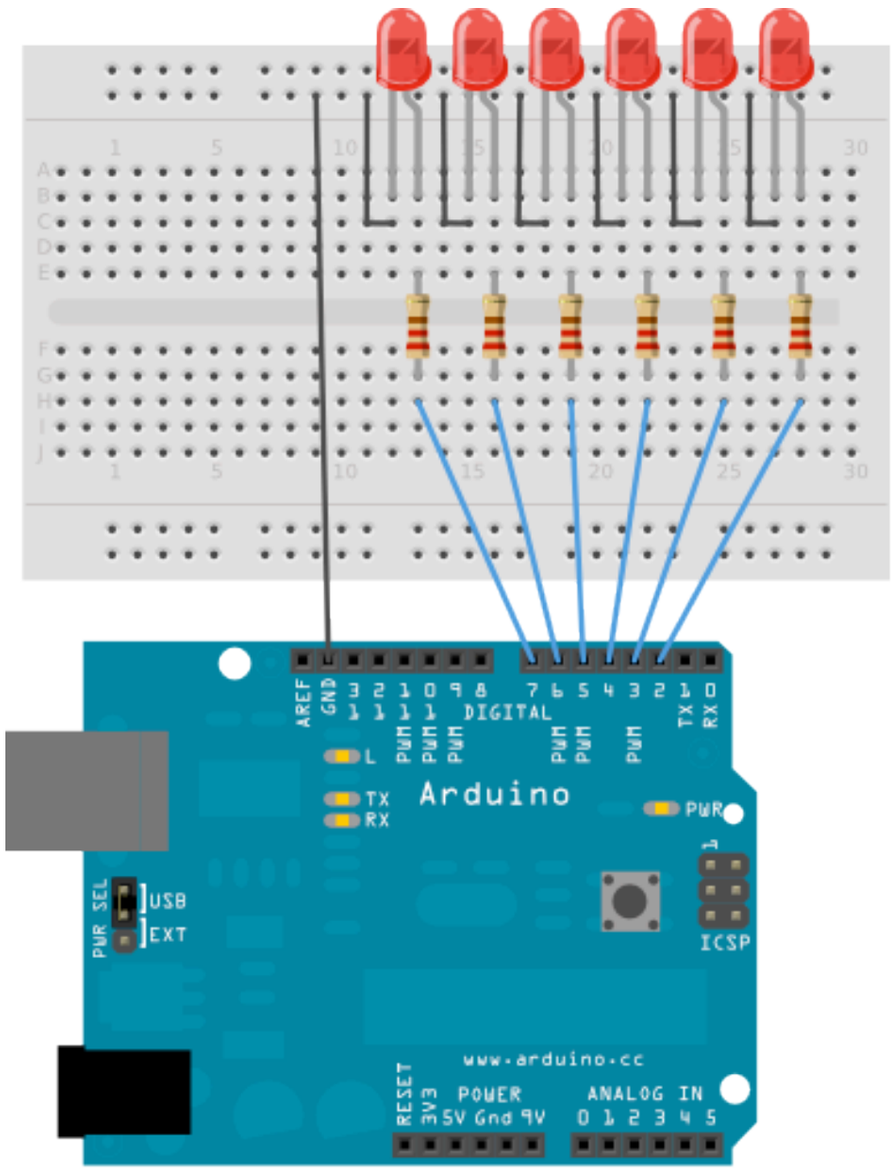
Cylon eyes

- Often you want to iterate over a series of pins and do something to each one.
- We can use a for() loop to cycle back and forth through digital pins 2-7.
- The LEDS are turned on and off, in sequence, by using both the digitalWrite() and delay() functions.

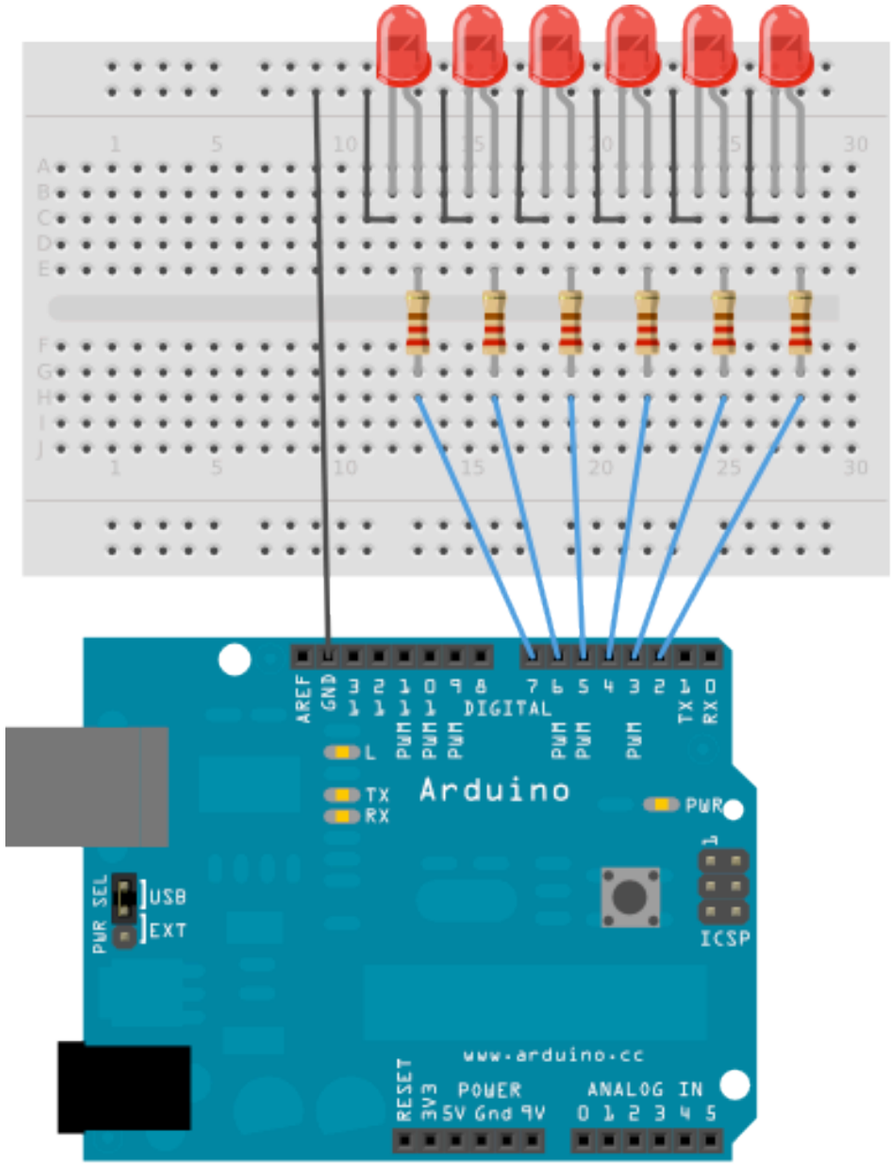


Make your own Cylon eyes

```
//Higher number, means slower timing.
int timer = 100;

void setup() {
 //for loop to make each pin an output
 for (int thisPin=2; thisPin<8; thisPin++)
 {
   pinMode(thisPin, OUTPUT);
 }
}

void loop() {
 // loop from lowest to highest pin
 for (int thisPin=2; thisPin<8; thisPin++)
 {
   digitalWrite(thisPin, HIGH);
   delay(timer);
   digitalWrite(thisPin, LOW);
 }
}
```



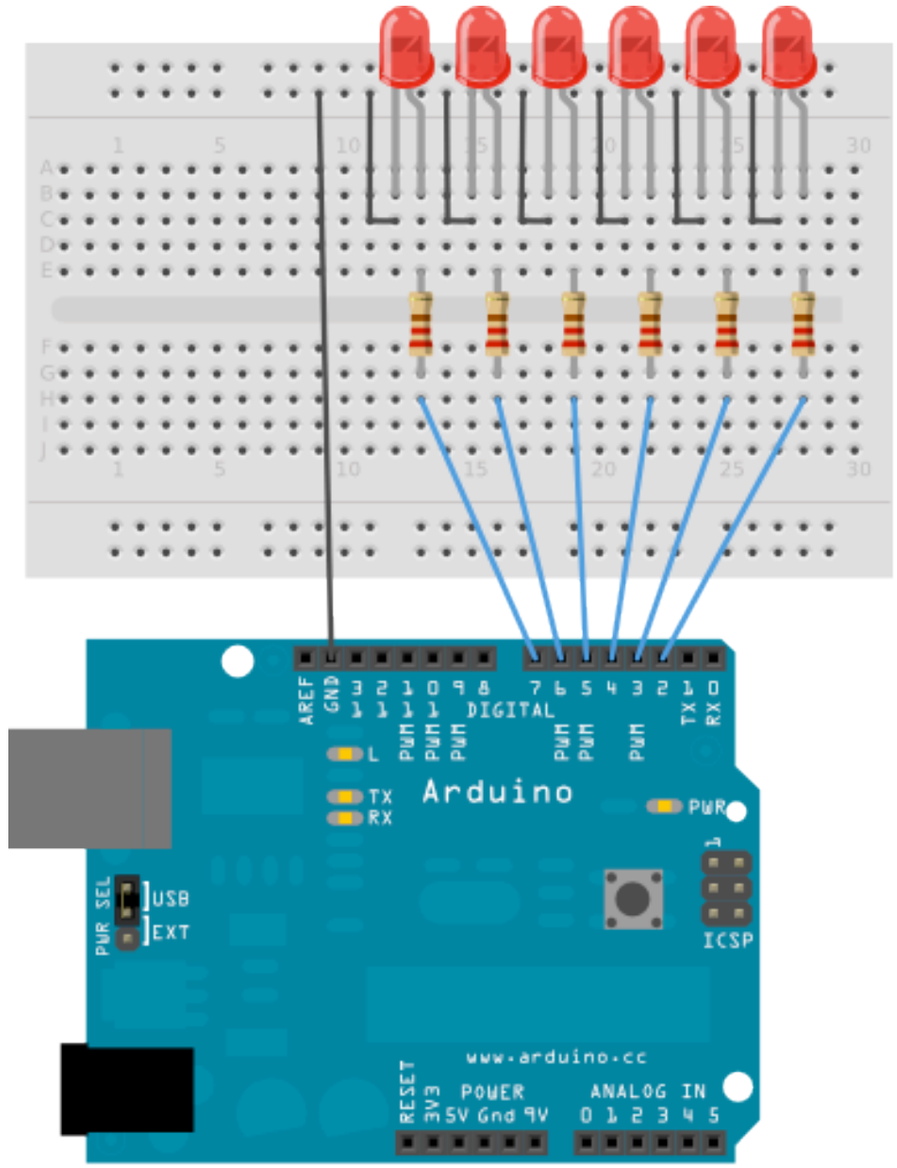
Tweak the Cylon



- Make the LEDS cycle LEFT to RIGHT and then RIGHT to LEFT.
- Change the sweeping speed.

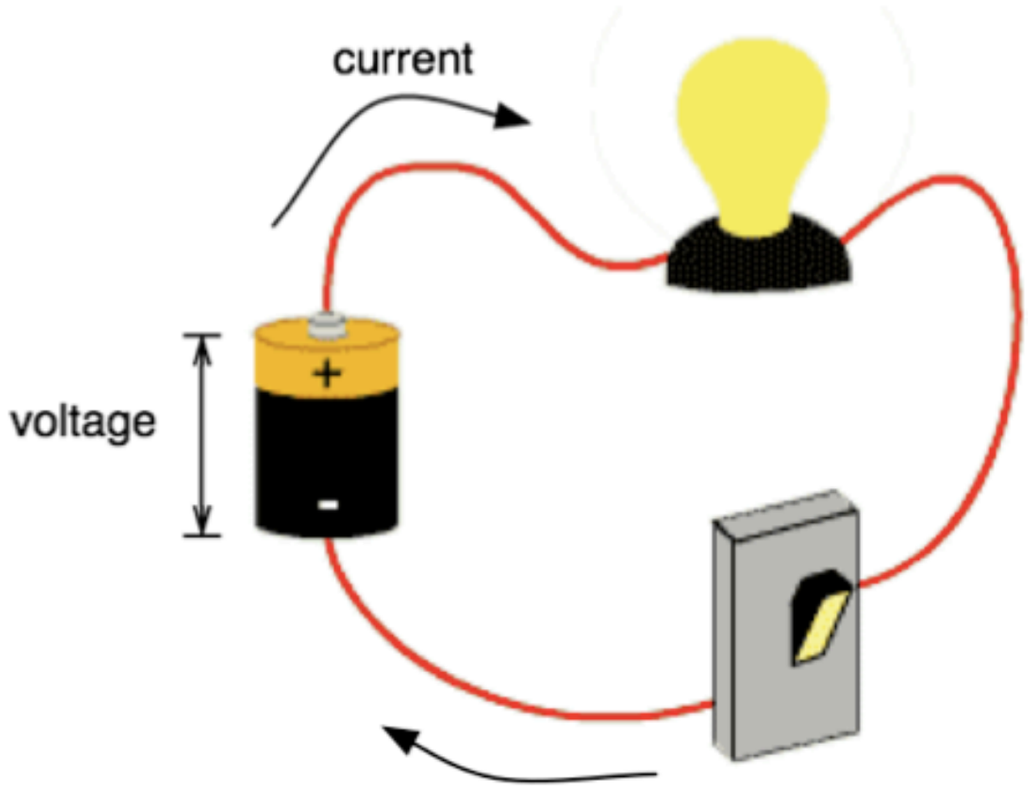
Tweak the Cylon (possible code)

```
int timer = 50;

void setup(){
  for (int thisPin=2; thisPin<=8; thisPin++)
    pinMode(thisPin, OUTPUT);
}

void loop(){
  // loop from lowest to highest pin
  for (int thisPin=2; thisPin<=8; thisPin++)
  {
    digitalWrite(thisPin, HIGH);
    delay(timer);
    digitalWrite(thisPin, LOW);
  }
  for (int thisPin=8; thisPin>=2; thisPin--){
    digitalWrite(thisPin, HIGH);
    delay(timer);
    digitalWrite(thisPin, LOW);
  }
}
```

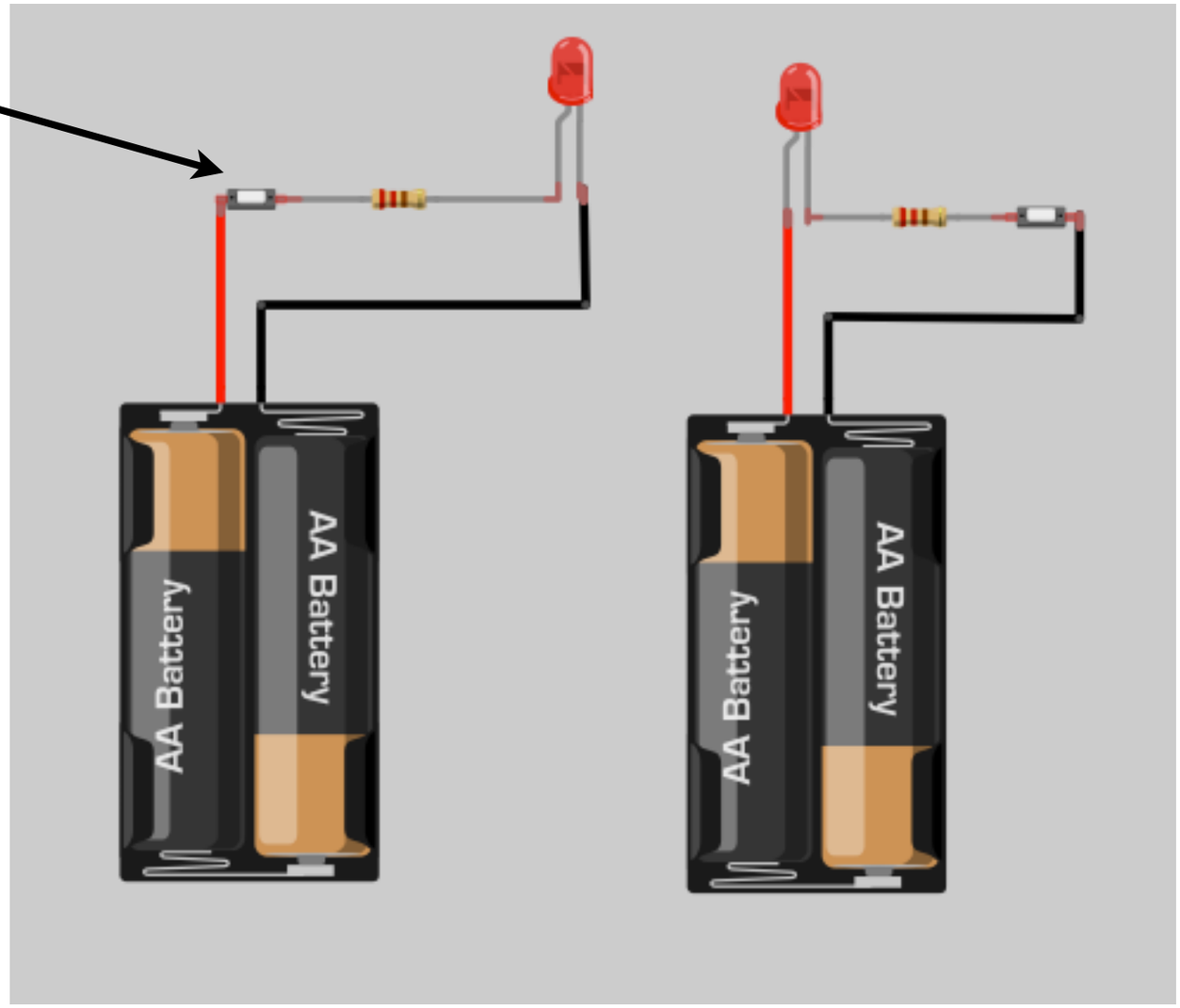


Sensors and inputs

- Many sensors are variations on switches Switches make or break a connection.
- Button ON:
  - Closed circuit, so current flows
- Button OFF:
  - Open circuit, no current



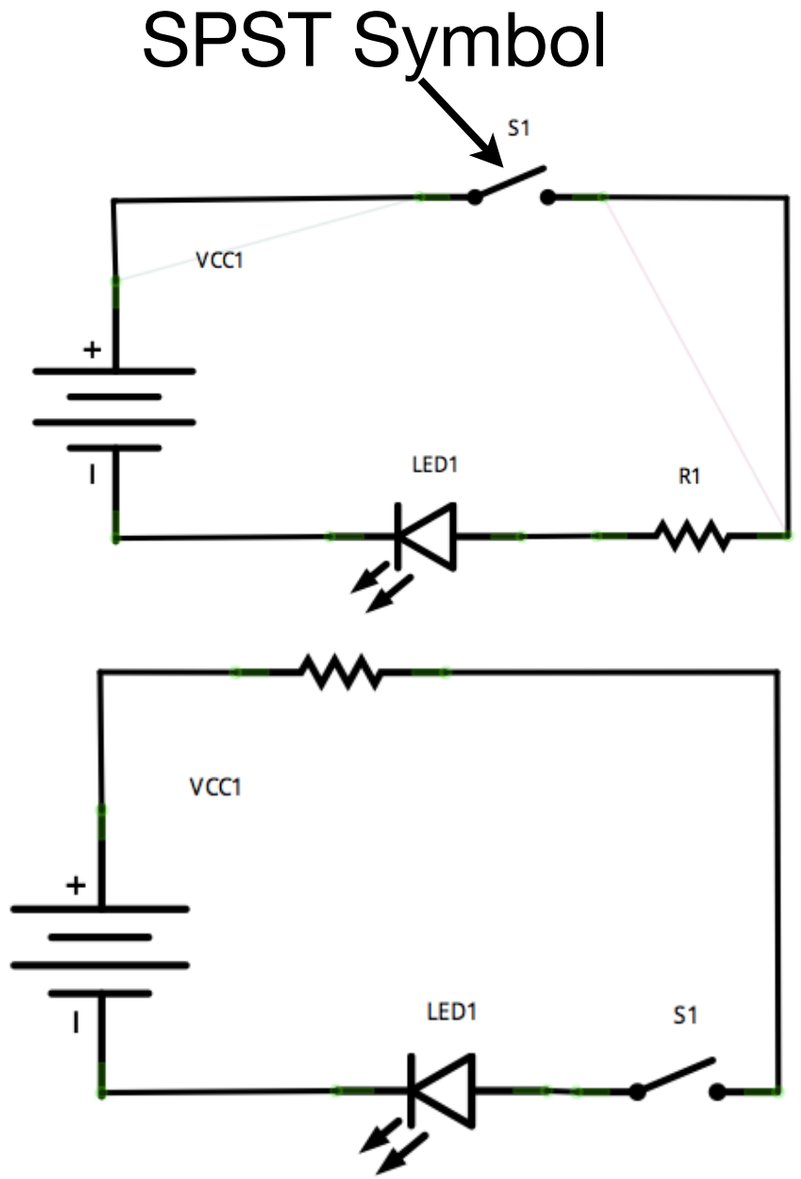
Push buttons and LEDs



- Once again, there is no difference between these two circuits.

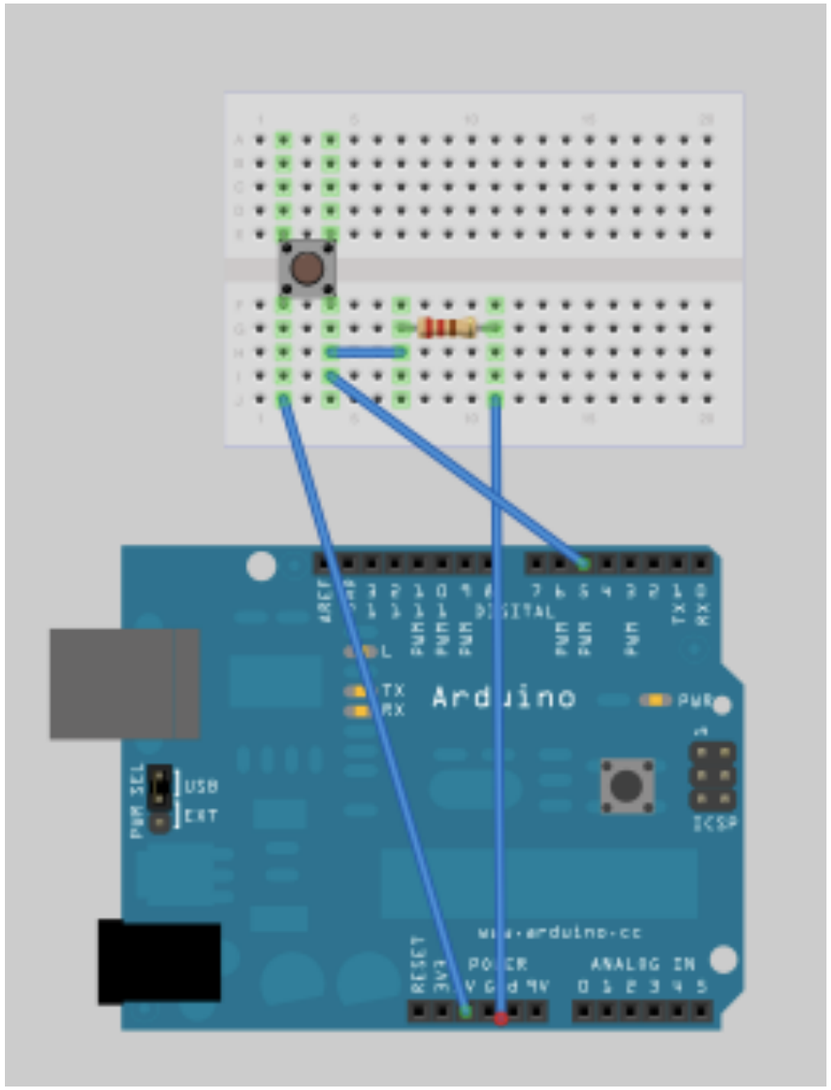
Single pole, single throw (SPST)

What is the difference between these two circuits?



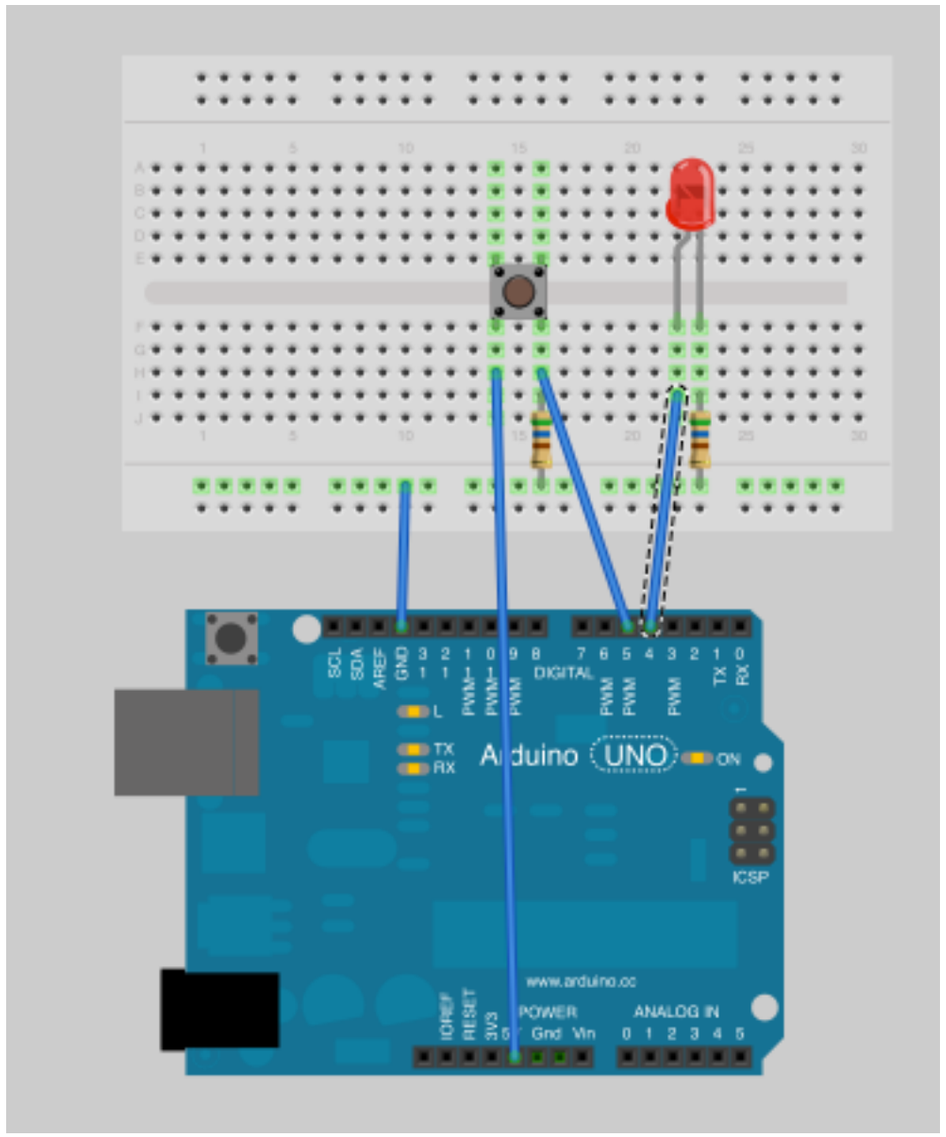
Reading input button

```
int pinUT=5;
int data=0;


void setup()
{
  //pin5 in set to receive data
  pinMode(pinUT, INPUT);
  //initiate serial communication
  Serial.begin(9600);
}

void loop() {
  data=digitalRead(pinUT);
  Serial.println(data);
  delay(1000);
}
```



Serial monitor will report “1” when button is pressed.
It will report “0” otherwise.

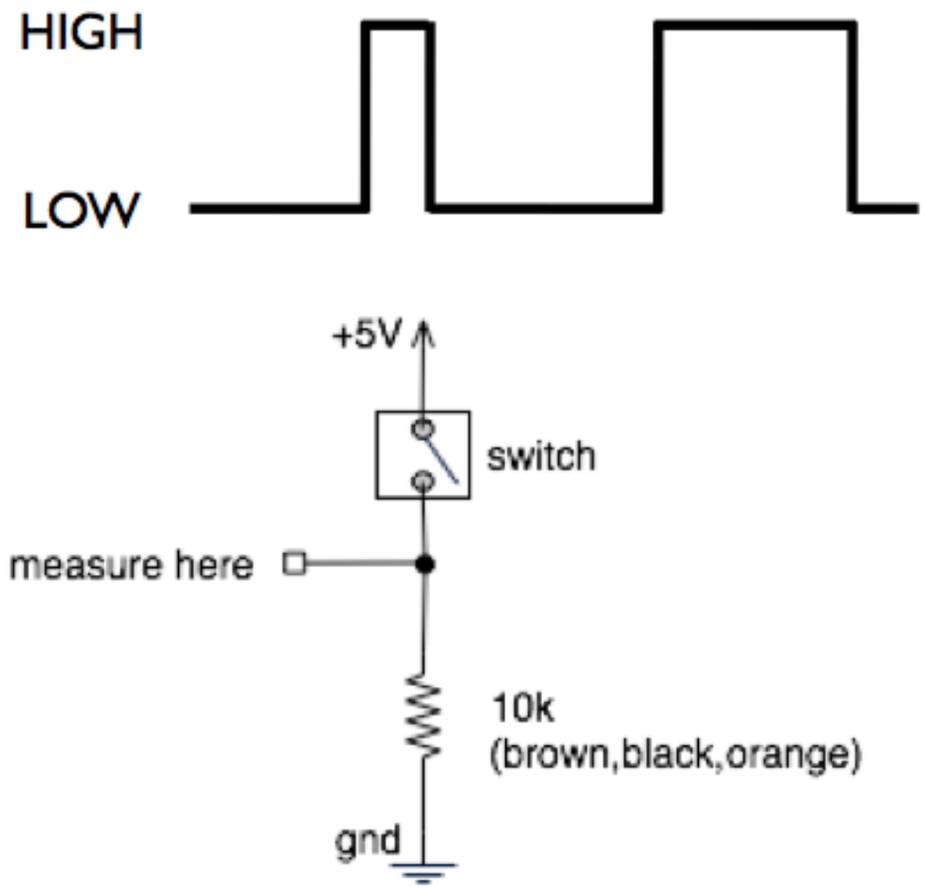
Turn ON a LED on pin #4 with a input button



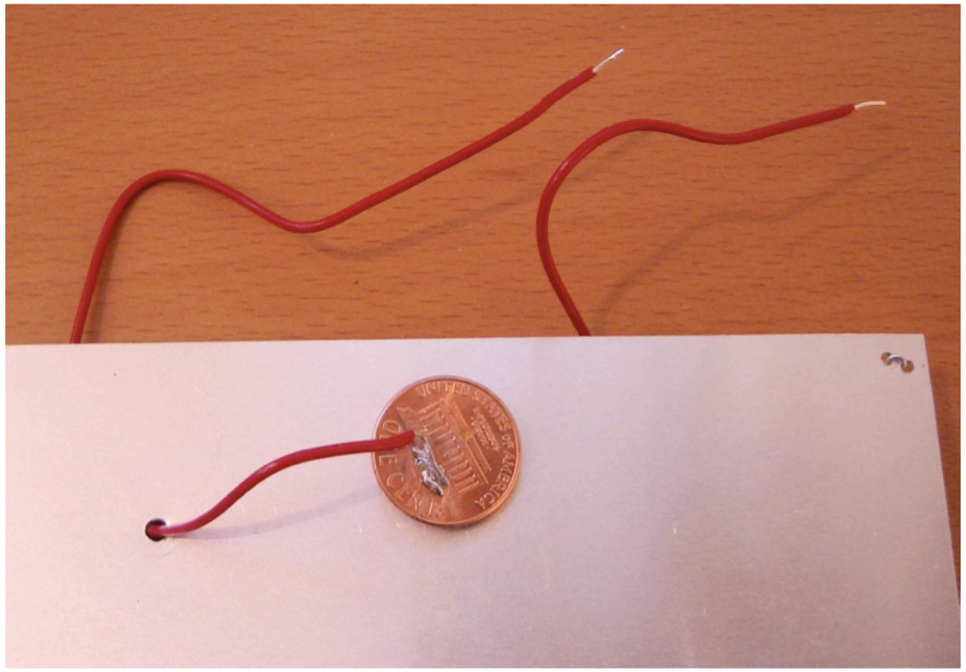

```
int pinUT=5, pinLED=4, data=0;

void setup()
{
     //pin #5 in set to receive data
     pinMode(pinUT, INPUT);
     //pin #4 has an LED
     pinMode(pinLED, OUTPUT);
}

void loop() {
     data=digitalRead(pinUT);
     if (data==1)
       digitalWrite(pinLED,HIGH);
     else
       digitalWrite(pinLED,LOW);
     delay(1000);
}
```

Why do we read button input like that?

- 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?

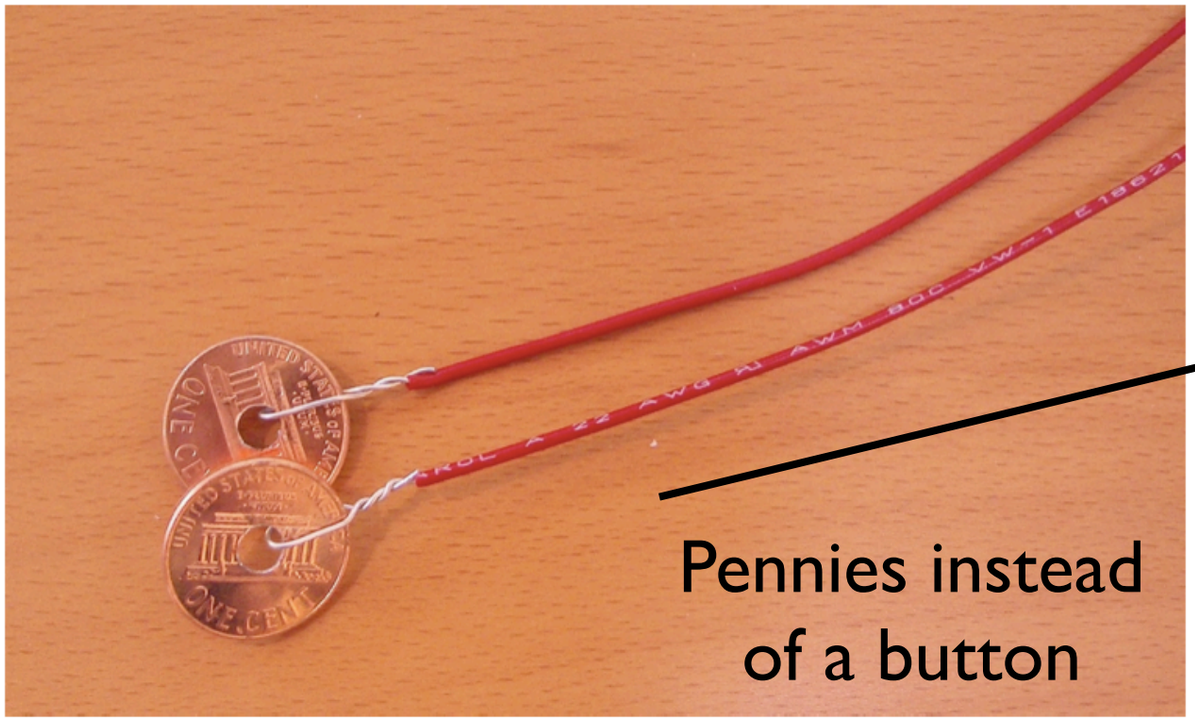


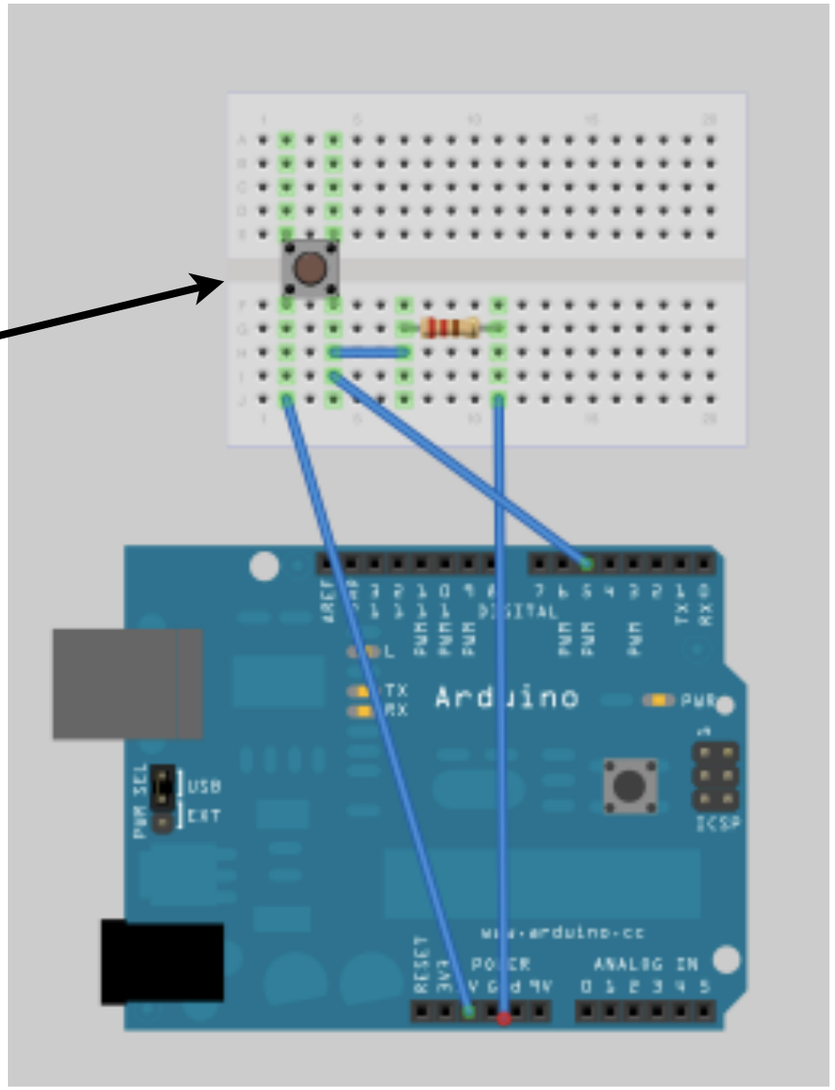
One possible switch



- Switches : “Make” or “Break” an electrical connection
- Eg: Penny on a surface. When the penny is lifted, alarms go off

Another possible switch

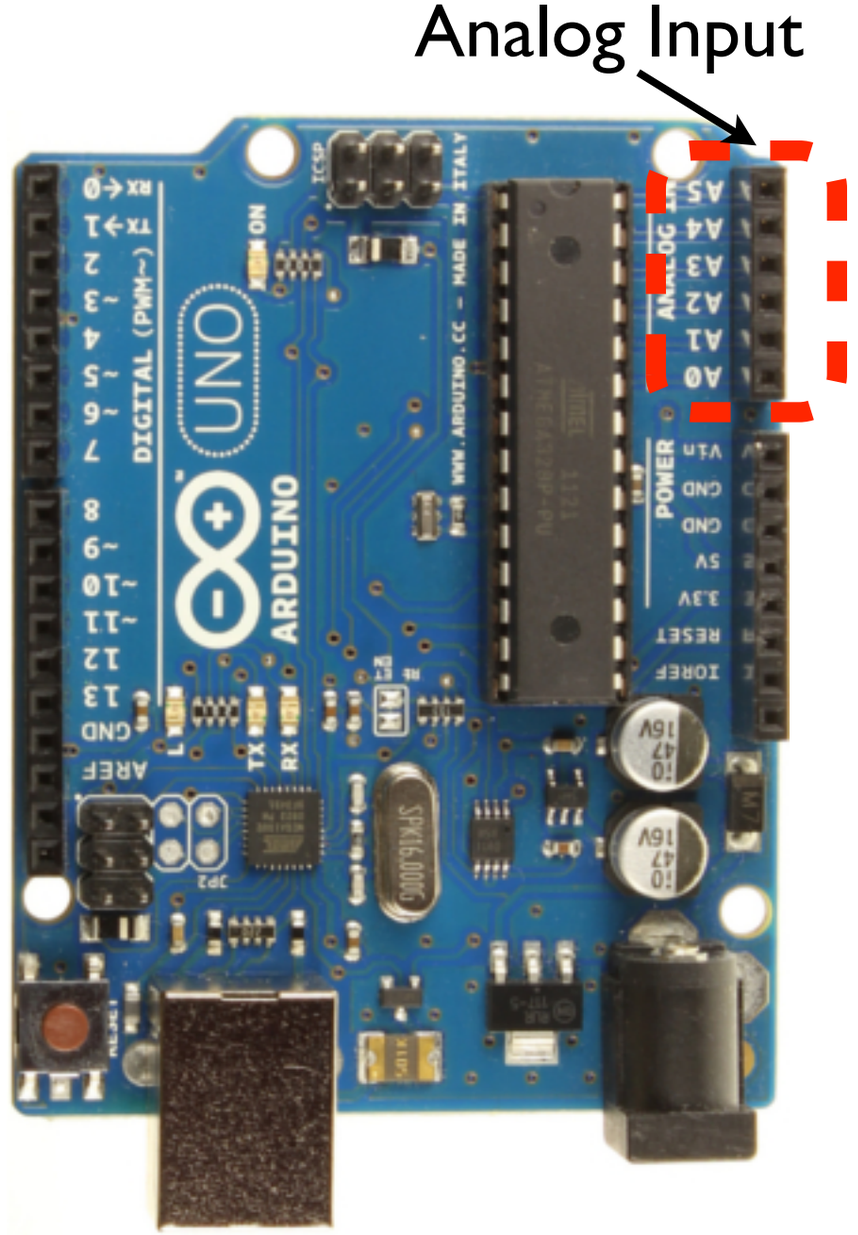




- Turn ON/OFF a LED on based on the collision of two pennies.
- Collision ensures an electrical connection.

Analog input

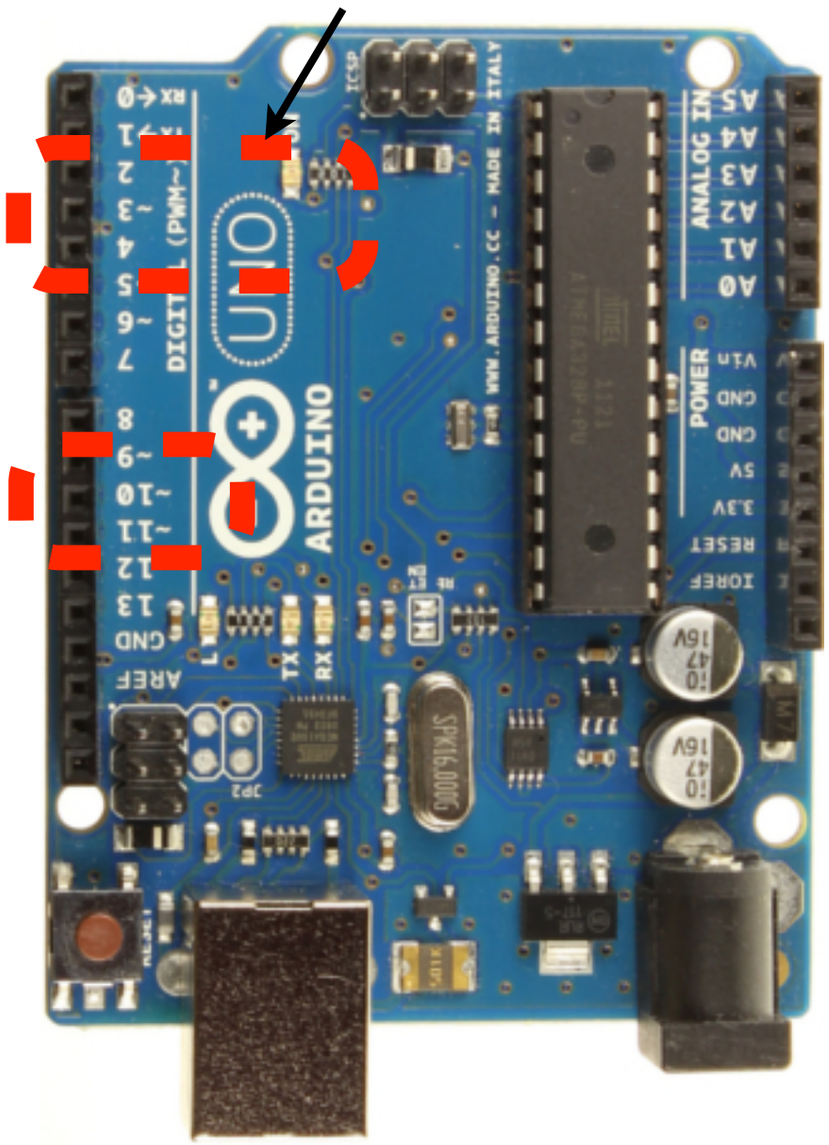
- You've seen how Arduino can output HIGH/LOW signals... what about a medium signal?
- Easy... with analog I/O!
- Output : 0 to 255
- Input: 0 to 1023
- Analog **input**: we can use it to determine the distance of an object via infra-red sensor.
- Analog **output**: we can use it to set speed of a motor or the brightness of a LED.

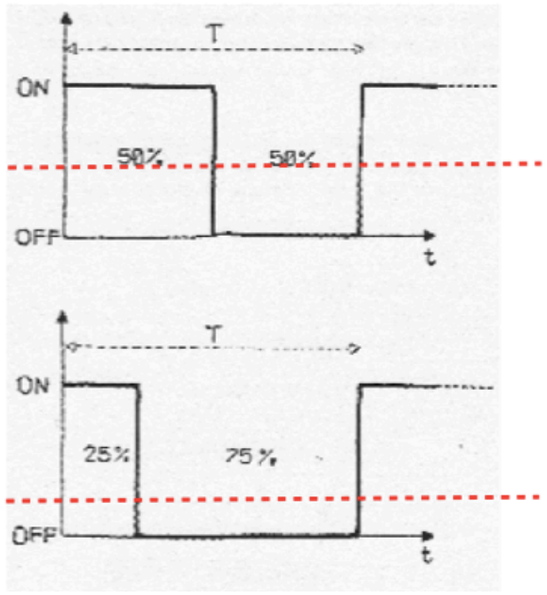


Analog output

- Some digital ports can be programmed to output analog signals.
- We enable those ports for output with pulse width modulation (PWM).
- PWM is obtained by varying between HIGH and LOW at the appropriate interval of time.

Digital pins with PWM





50% Duty Cycle

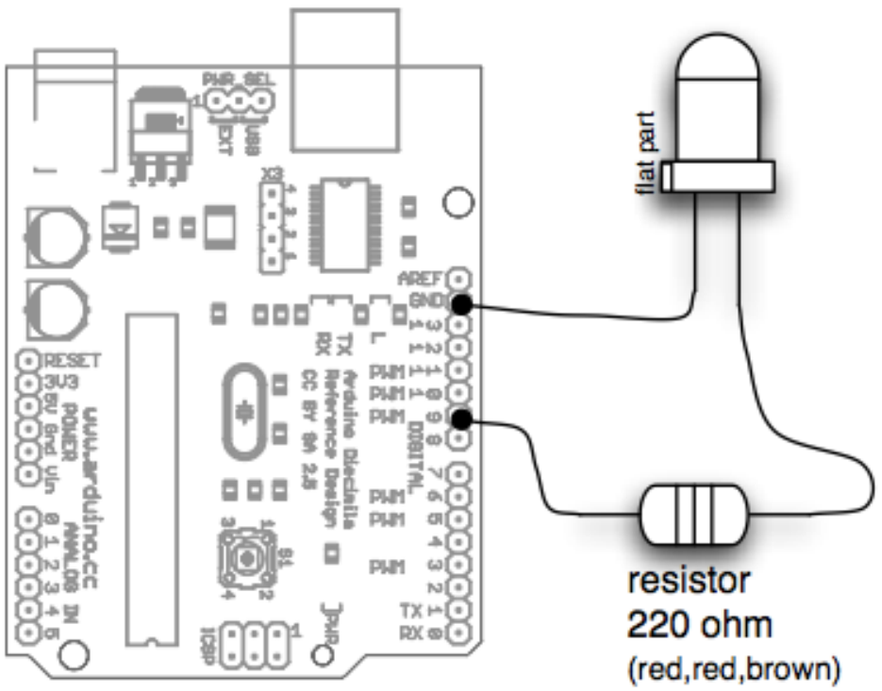
Effective Voltage

25% Duty Cycle

Effective Voltage

Dimming light

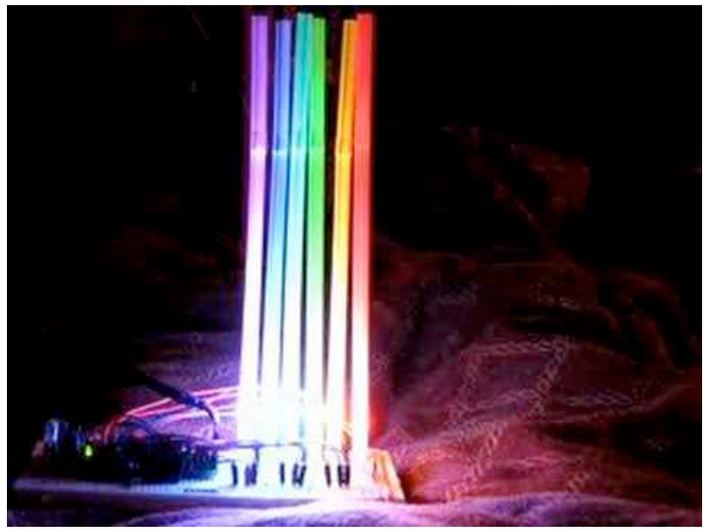
```
int ledPin=9;

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

void loop()
{
  int value=0;

  //loops value from 0 to 255
  for (value=0; value<255;value++)
  {
     //write the index of the loop to the pin
    analogWrite(ledPin,value);
    //wait 10ms
    delay(10);
  }

}
```



wiring diagram

Dimming light

- Add a couple of different LEDs at different PWM pins.
- Specify different delays to emulate candlelight flicker.
- Measure Voltage on a PWM pin. What's going on?

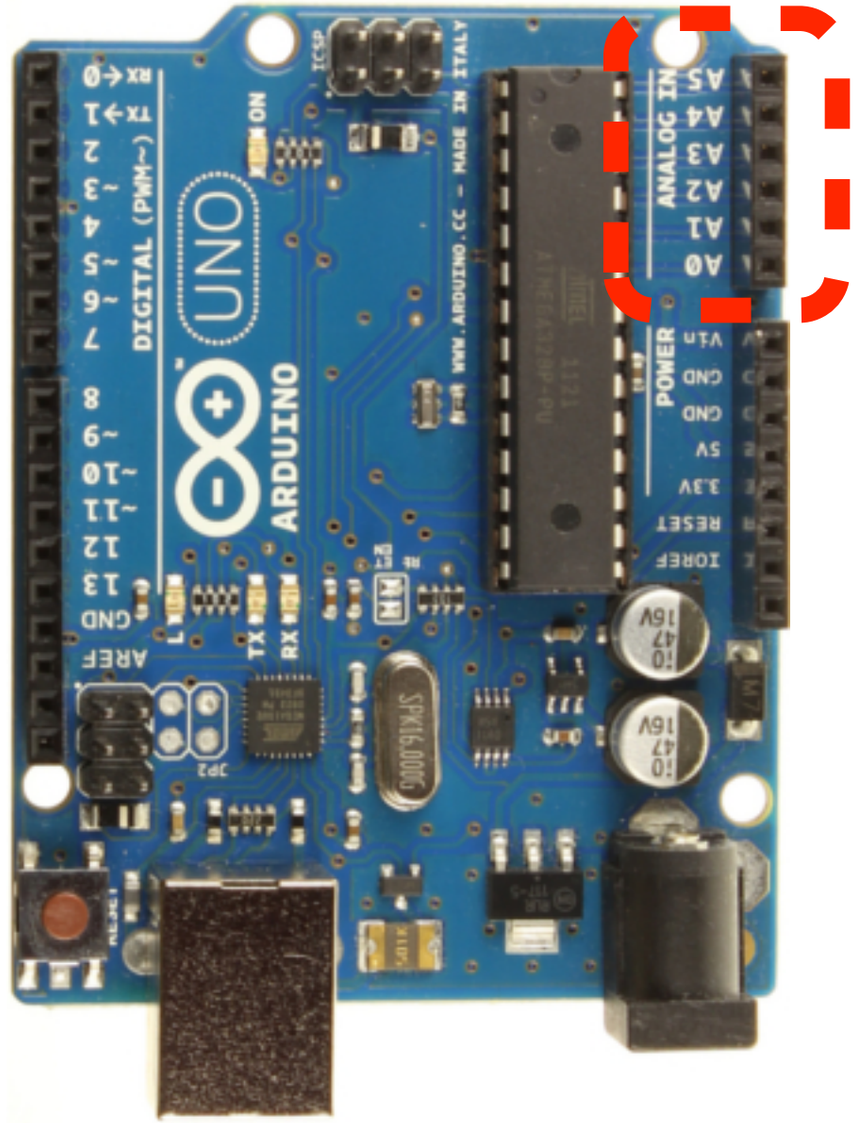


Hackerspace Camp - Summer 2012

```
int ledPin=9;

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

void loop()
{
  int value=0;

  //loops value from 0 to 255
  for (value=0; value<255;value++)
  {
     //write the index of the loop to the pin
    analogWrite(ledPin,value);
    //wait 10ms
    delay(10);
  }

}
```

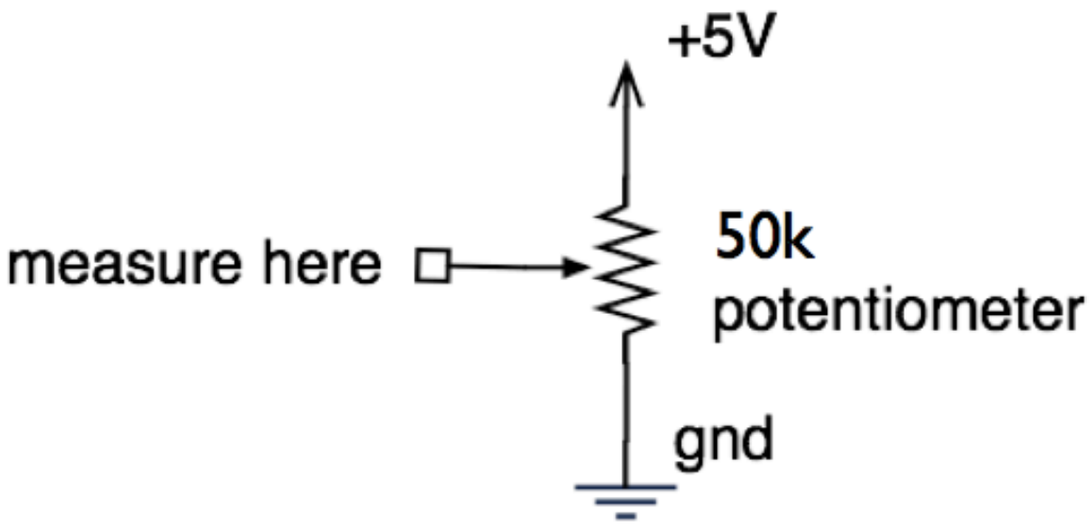
Analog input

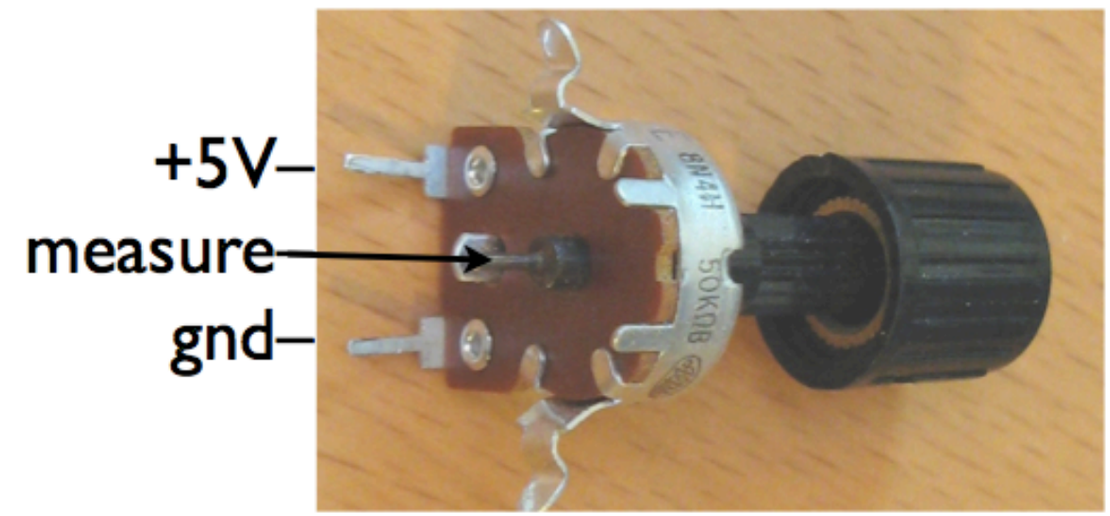
- Arduino has six analog inputs
- Reads voltage between 0 to 5 volts
- Resolution is 10-bit (1024 values)
- In other words, 5/1024 = 4.8 mV smallest voltage change you can measure
- Examples of varying voltage inputs: Temperature sensors, light sensors, ...



Potentiometers

With a potentiometer we can create an analog input.



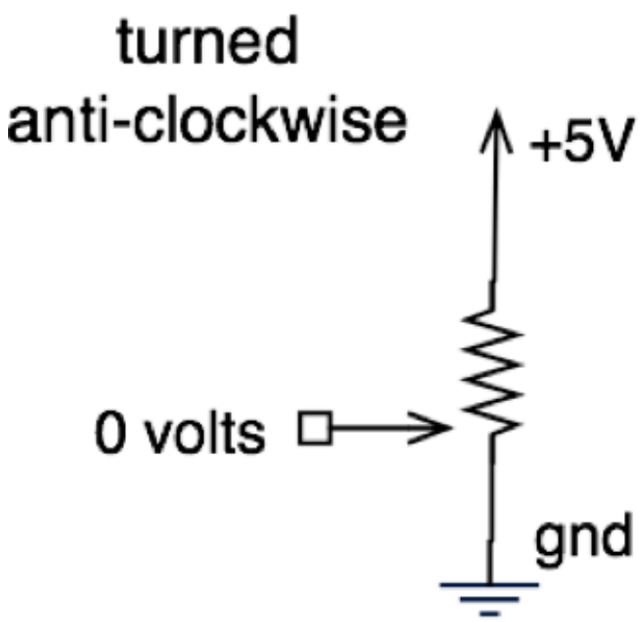


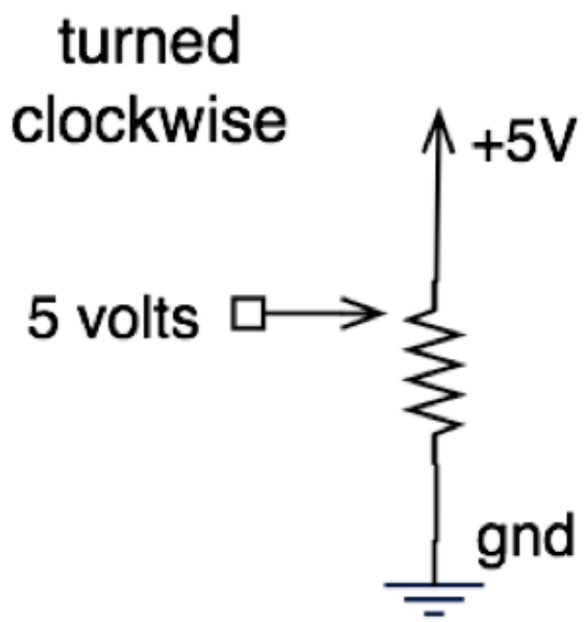
The pot you have

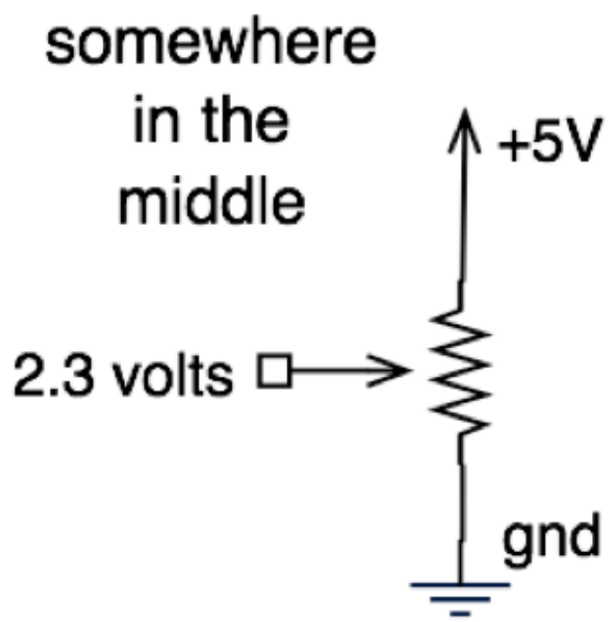
pots also look like this

Potentiometers

Moving the knob is like moving where the arrow taps the voltage on the resistor

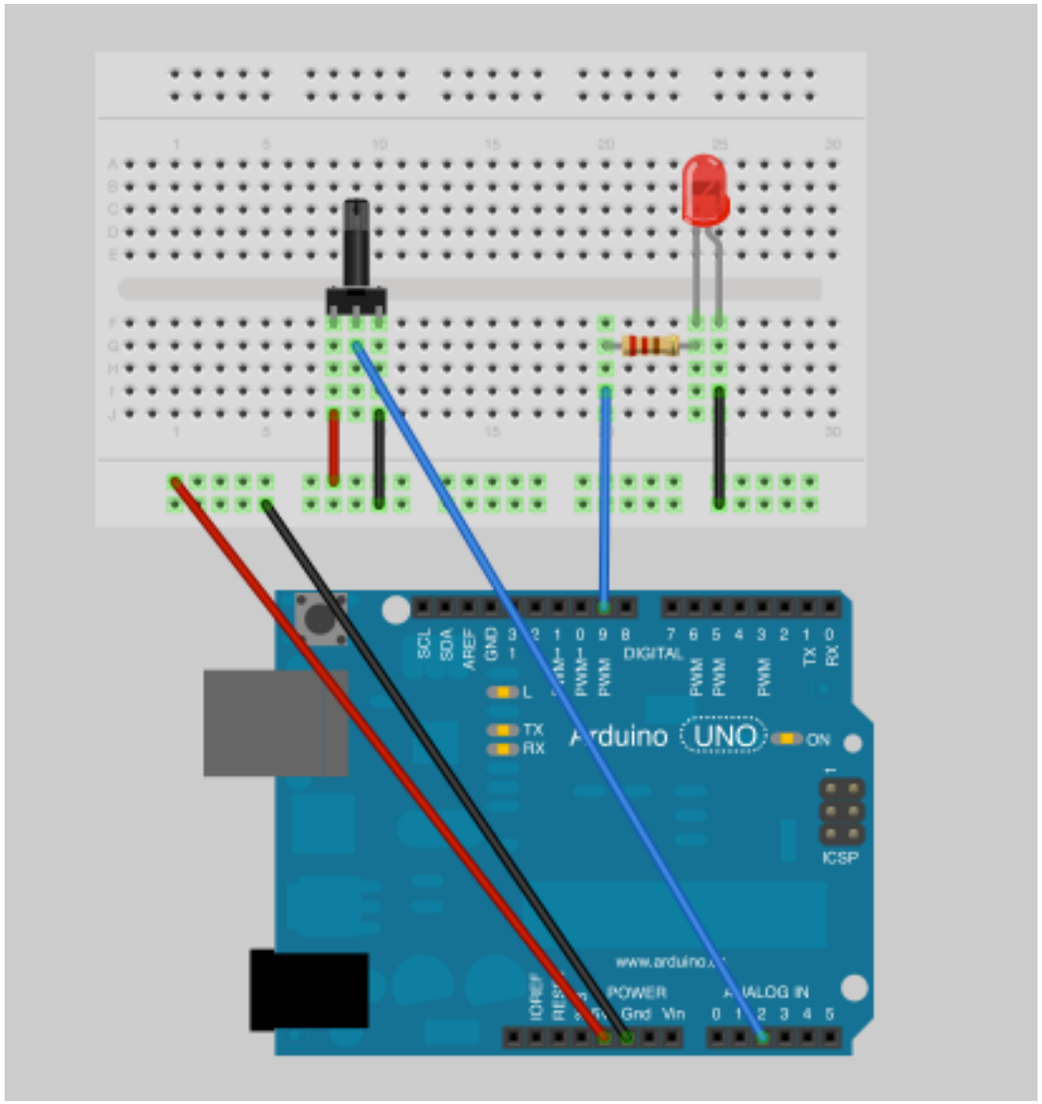






Analog read: Blink an LED at different speeds!

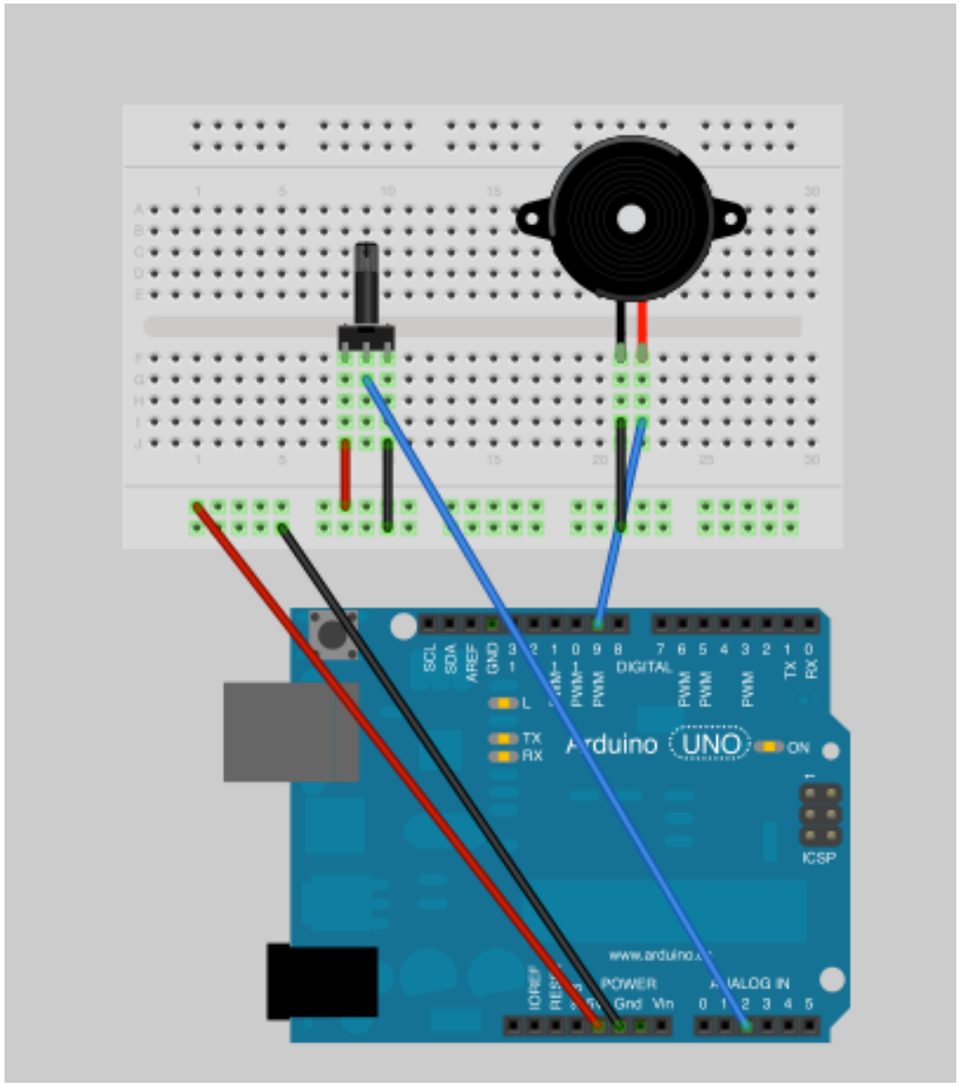
```
int ledPinA =  9; //output pin (led)
int sensorPinA = 2; //input pin for pot.

void setup()
{
  Serial.begin(9600);
  pinMode(ledPinA, OUTPUT);
}

void loop()
{
  int inputVal = analogRead(sensorPinA);
  digitalWrite(ledPinA, HIGH);
  Serial.println(inputVal);
  delay(inputVal);
  digitalWrite(ledPinA, LOW);
  delay(inputVal);
}
```



- See that sensorValueA goes from 0 to 1024.

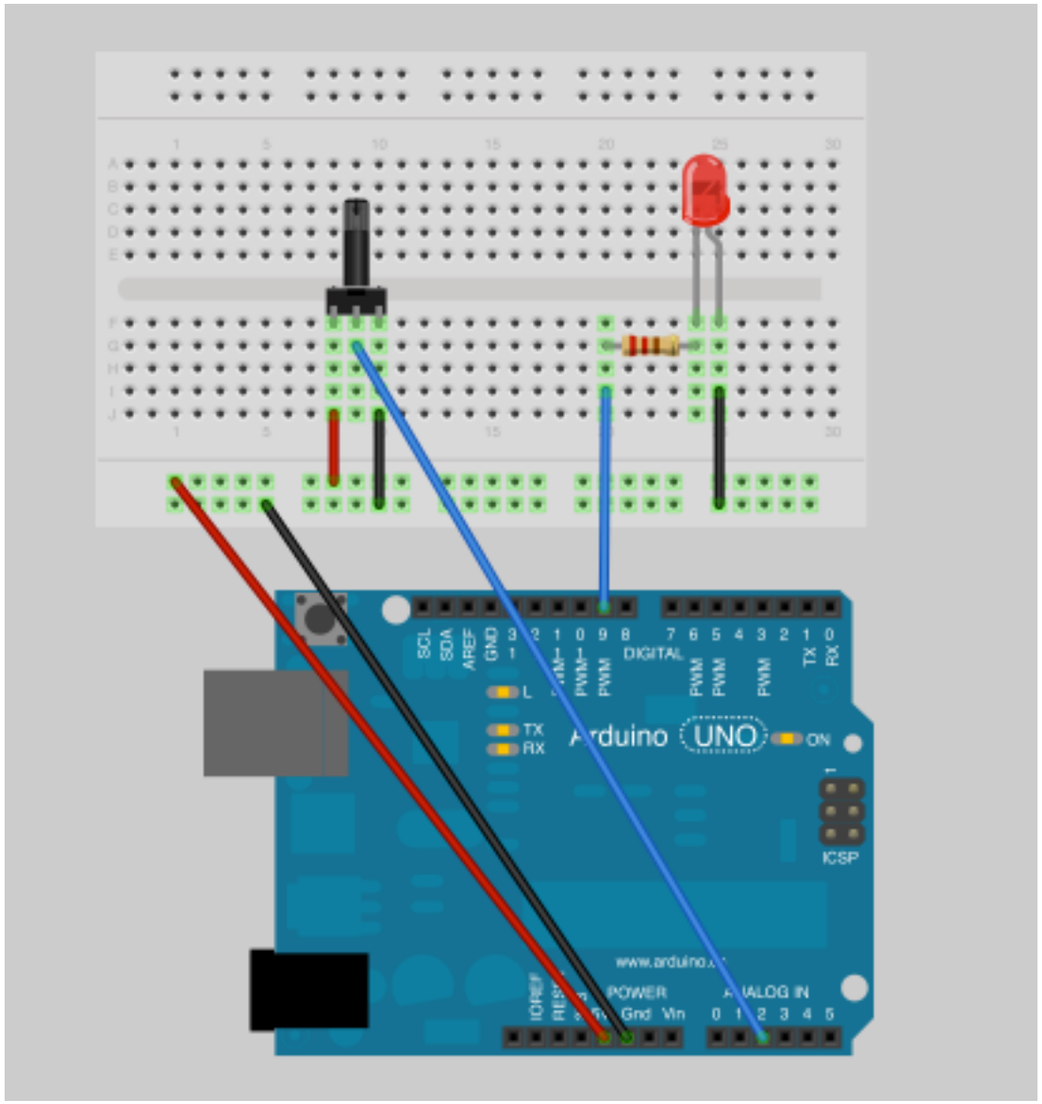
Replace the LED with a speaker

```
int ledPinA =  9; //output pin (led)
int sensorPinA = 2; //input pin for pot.

void setup()
{
  Serial.begin(9600);
  pinMode(ledPinA, OUTPUT);
}

void loop()
{
  int inputVal = analogRead(sensorPinA);
  digitalWrite(ledPinA, HIGH);
  Serial.println(inputVal);
  delay(inputVal);
  digitalWrite(ledPinA, LOW);
  delay(inputVal);
}
```



Turn one LED ... as you move potentiometer

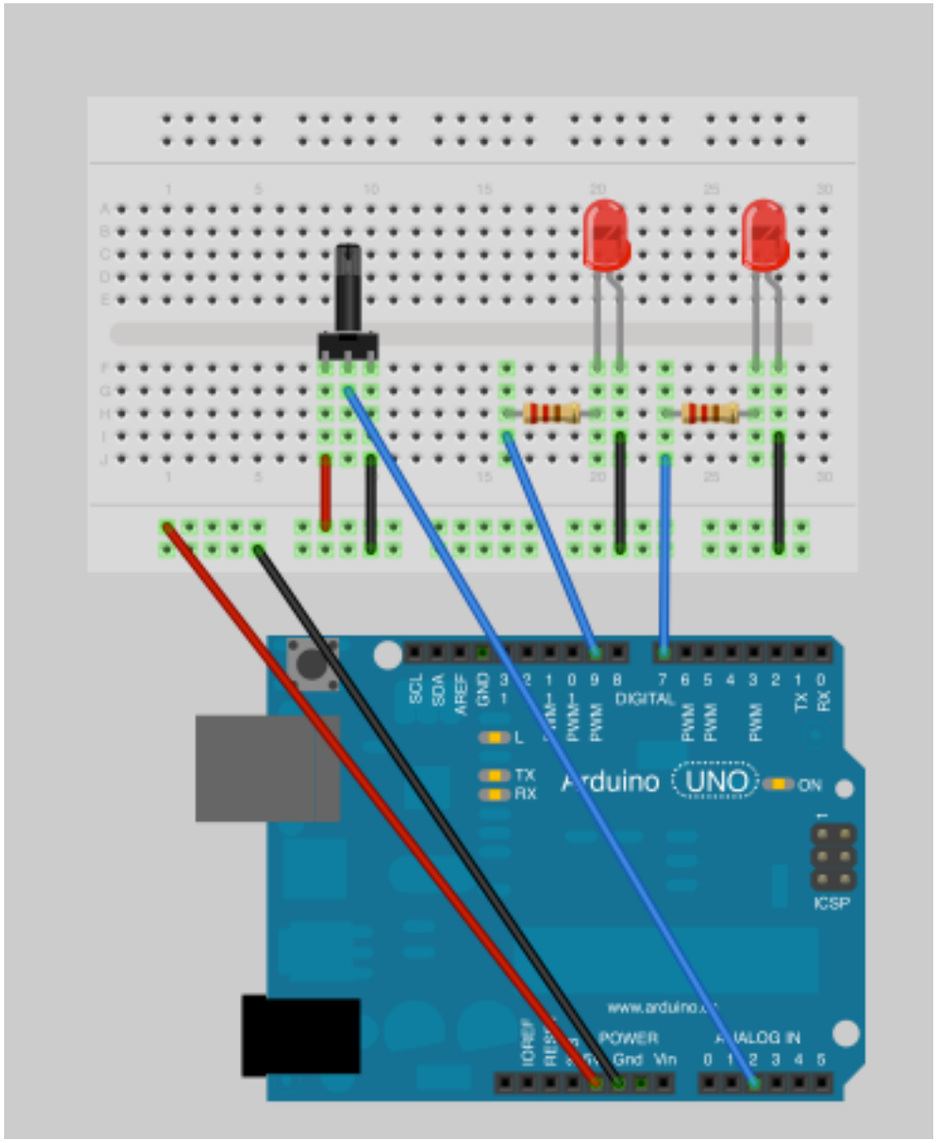
```
int ledPinA =  9; //output pin (led)
int sensorPinA = 2; //input pin for pot.

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

void loop()
{
  int inputVal = analogRead(sensorPinA);
  if (inputVal>512)
    digitalWrite(ledPinA, HIGH);
  else
    digitalWrite(ledPinA, LOW);
}
```



- See that sensorValueA goes from 0 to 1024.

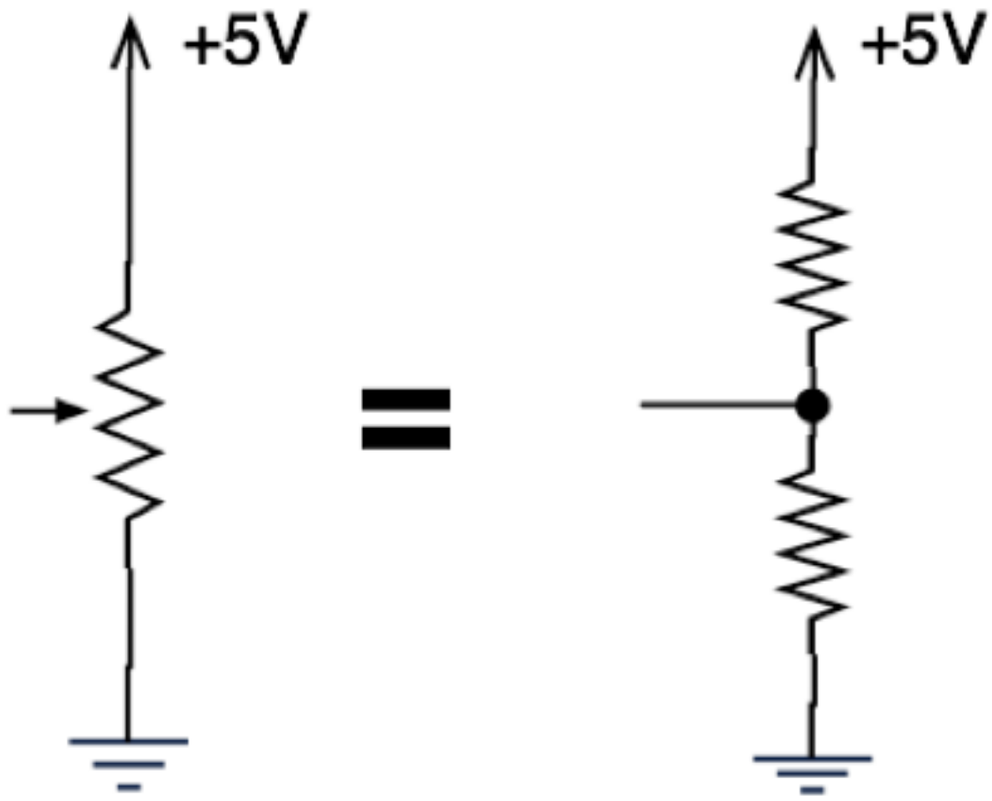
Turn one or two LEDs ... as you move potentiometer

```
int ledPinA =  9, ledPinB=7;
int sensorPinA = 2; //input pin for pot.

void setup(){
  pinMode(ledPinA, OUTPUT);
  pinMode(ledPinB, OUTPUT);
}


void loop()
{
  int inputVal = analogRead(sensorPinA);
  digitalWrite(ledPinA, LOW);
  digitalWrite(ledPinB, LOW);

  if (inputVal>512){
    digitalWrite(ledPinA, HIGH);
    digitalWrite(ledPinB, LOW);
  }
  if (inputVal>714){
    digitalWrite(ledPinA, HIGH);
    digitalWrite(ledPinB, HIGH);
   }
}
```



Voltage divider

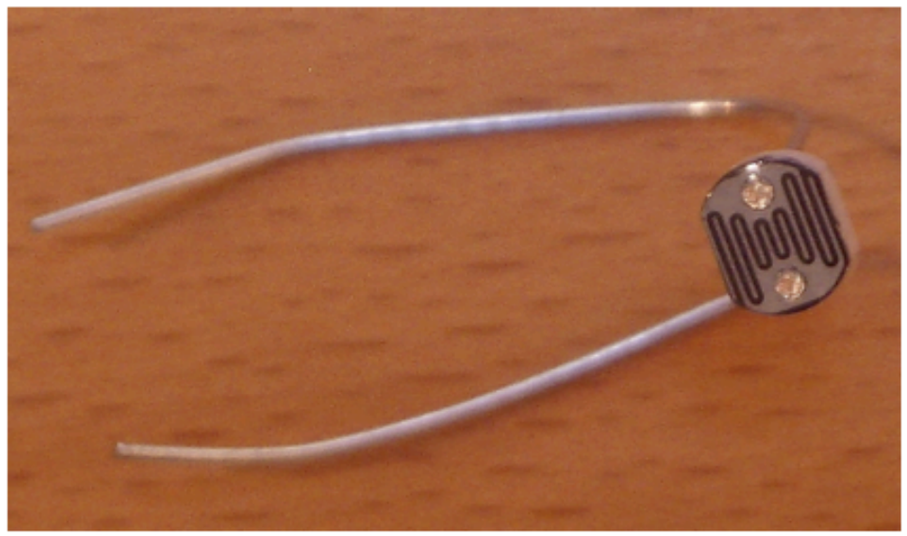
- Potentiometers are example of a voltage divider.
- Voltage divider splits a voltage in two.
- Same as two resistors, but you can vary them...



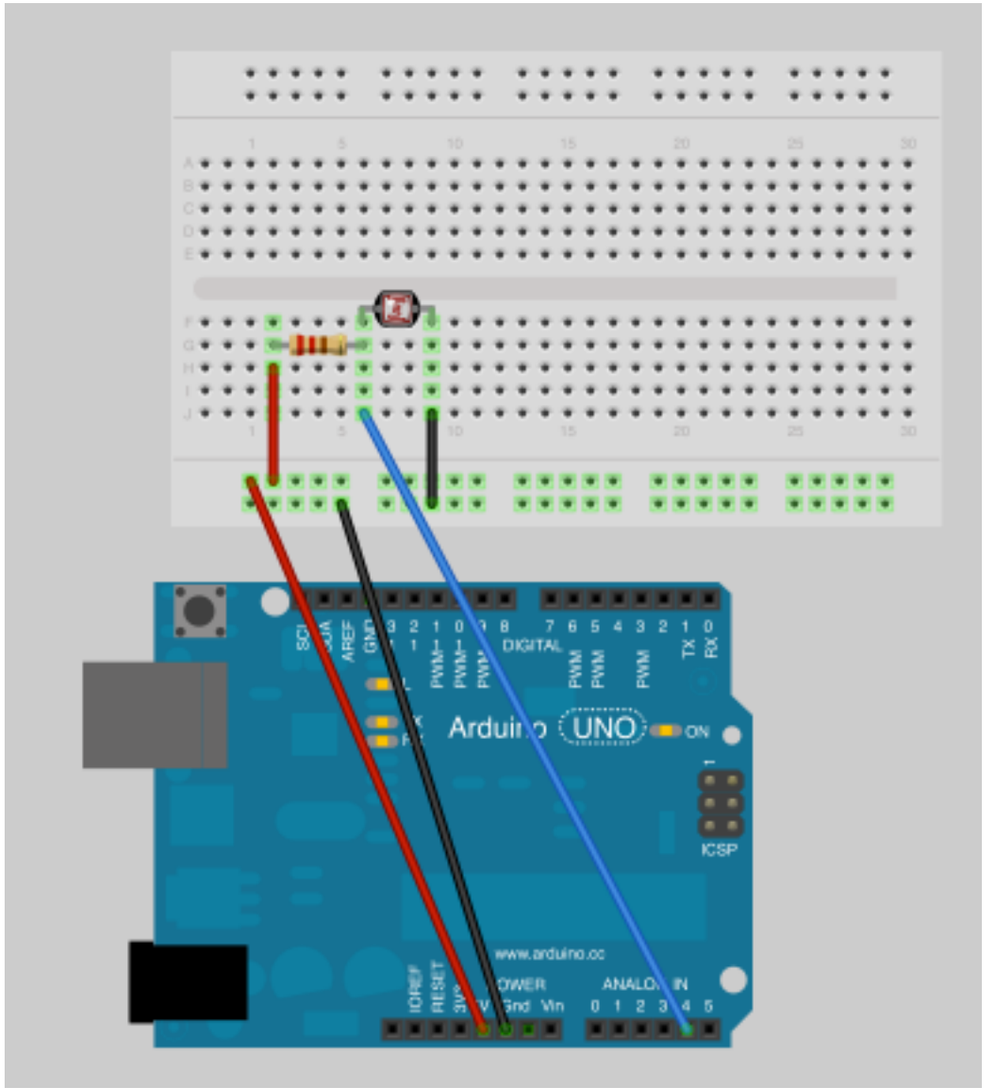
Sensing the dark: photocells

- Photoresistor, light-dependent resistor
- A variable resistor
- Brighter light == lower resistance
- Photocells you have range approx. 0-10k-1M

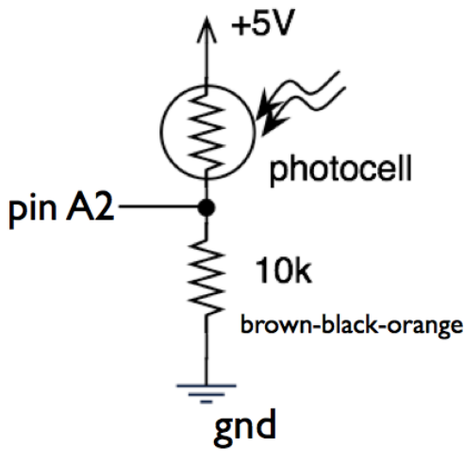
schematic symbol



PhotoCell circuit

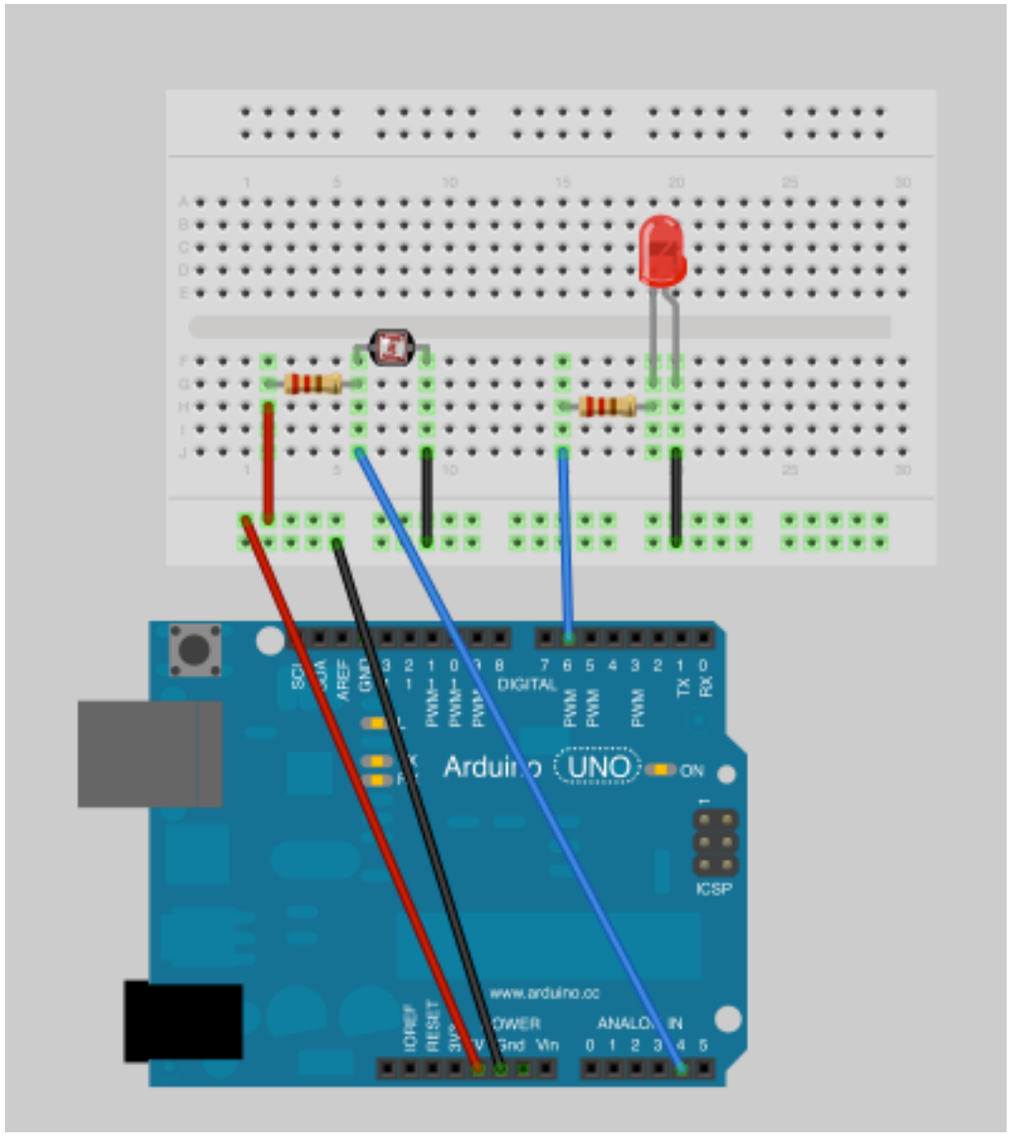



```
int sensorPinA = 4;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int inputVal = analogRead(sensorPinA);
  Serial.println(inputVal);
  delay(1000);
}
```



- How much does the input value changes when you cover the photocell?

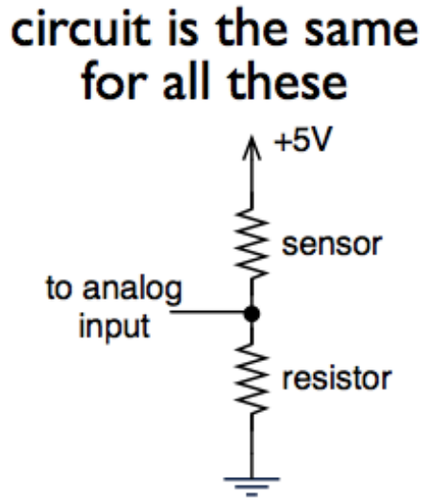
If the value > a (certain) number then turn ON LED

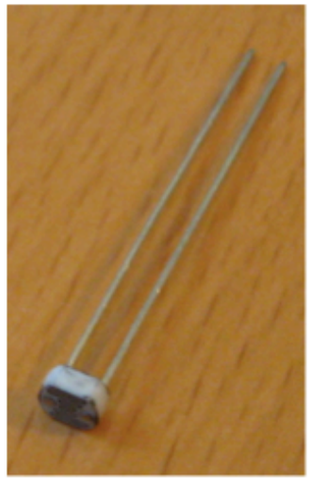



```
int sensorPinA = 4, ledPin=6;

void setup()
{
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int inputVal = analogRead(sensorPinA);
  Serial.println(inputVal);
  if (inputVal>400) <-- You pick!
    digitalWrite(ledPin,HIGH);
   else
    digitalWrite(ledPin,LOW);

  delay(1000);
}
```

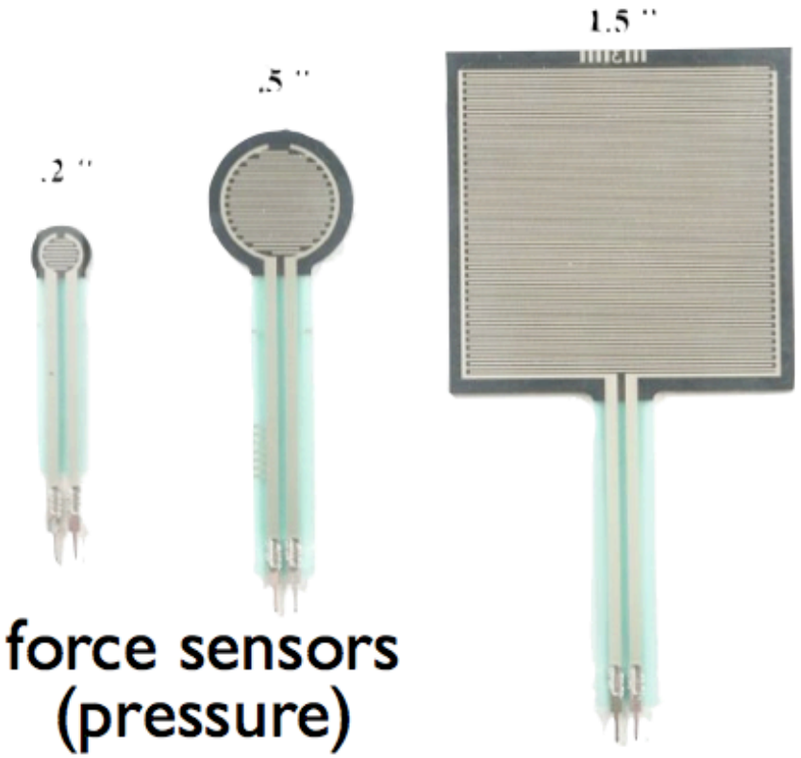
Resistive sensors

thermistor
(temperature)

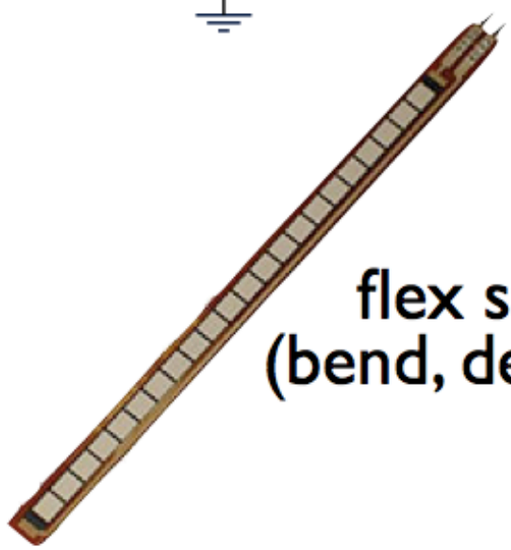




photocell
(light)



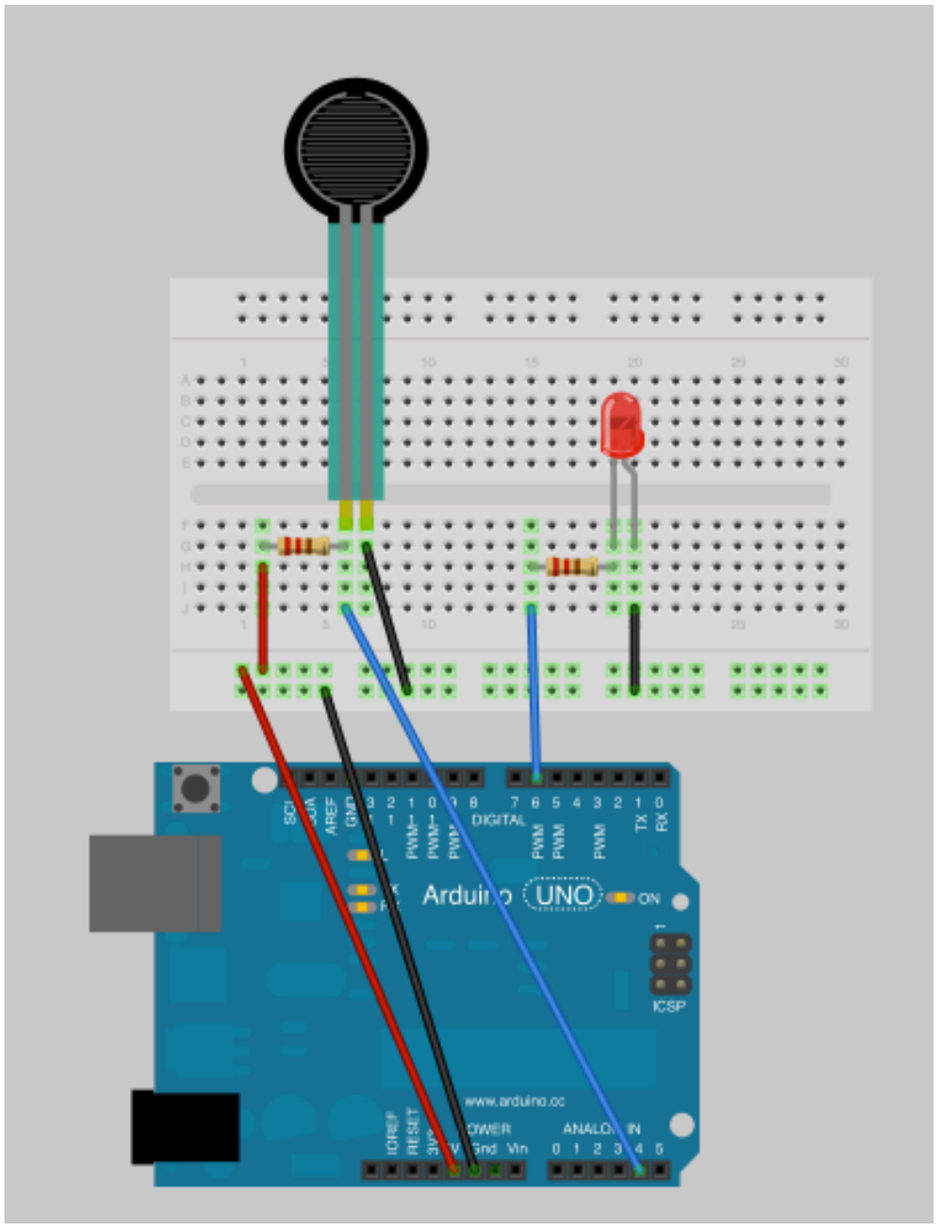
force sensors
(pressure)



flex sensor
(bend, deflection)

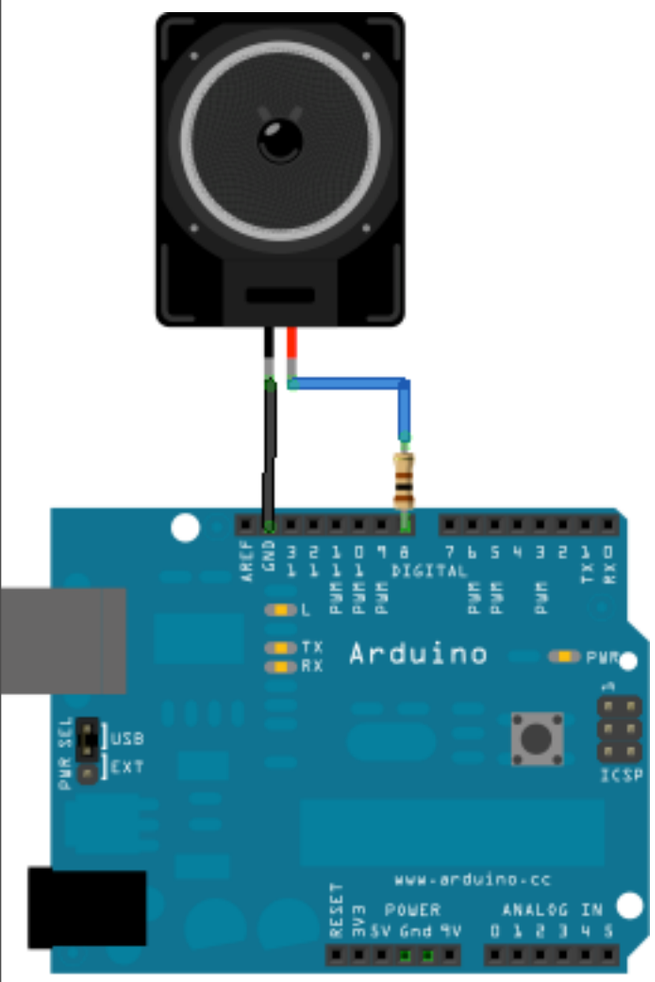
also air pressure
and others

Replace the photocell with pressure sensor!




```
int sensorPinA = 4, ledPin=6;

void setup()
{
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int inputVal = analogRead(sensorPinA);
  Serial.println(inputVal);
  if (inputVal>400) <-- You pick!
    digitalWrite(ledPin,HIGH);
   else
    digitalWrite(ledPin,LOW);

  delay(1000);
}
```

Lets play some music



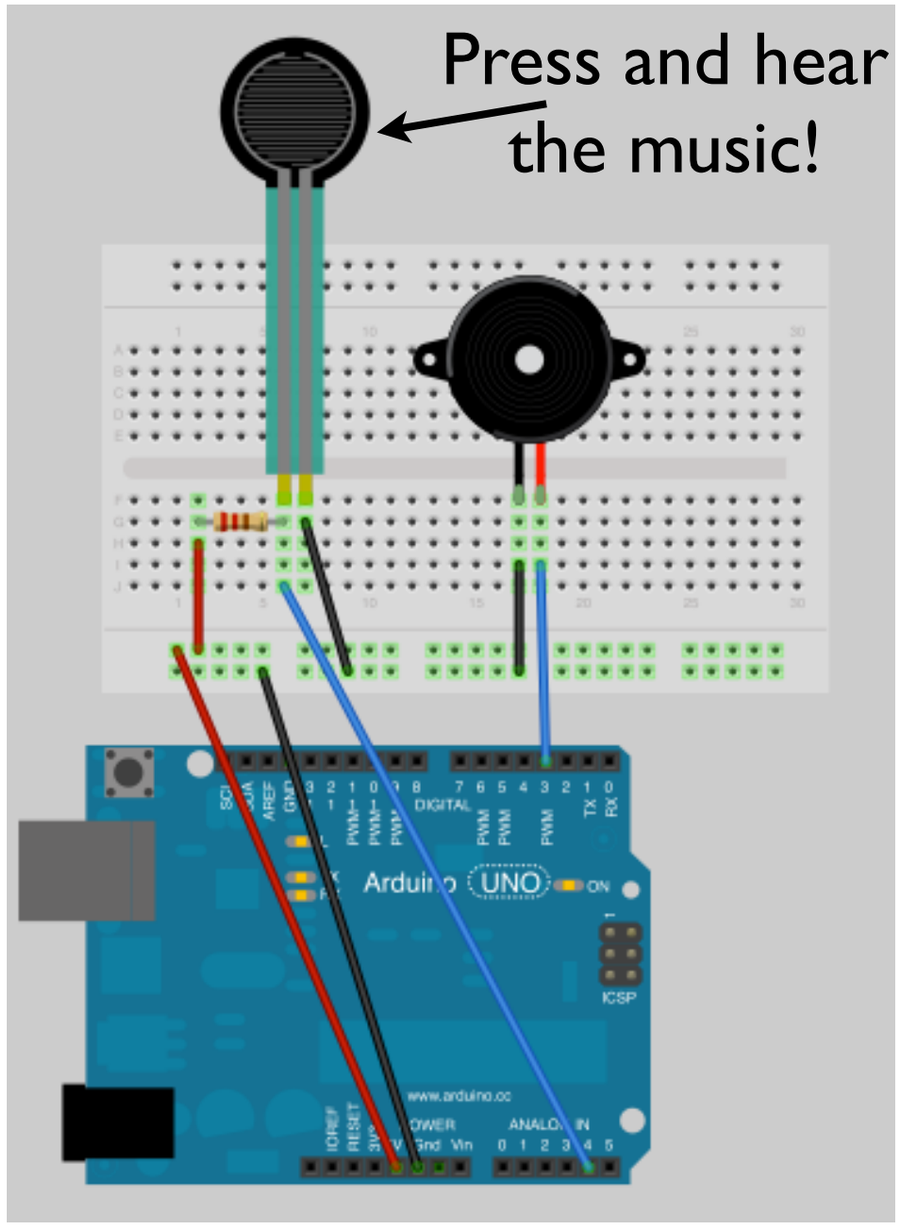

```
#define NOTE_C4  262
#define NOTE_G3  196
#define NOTE_A3  220

int melody[]={NOTE_C4,NOTE_G3,NOTE_G3,NOTE_A3,
NOTE_G3};
int noteDurations[] = {4,8,8,4,4,4};

void setup() {}
void loop()
{
  for (int thisNote = 0; thisNote < 5; thisNote++) {
    int noteDuration = 1000/noteDurations[thisNote];
    tone(8, melody[thisNote],noteDuration);

    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);

    noTone(8);
  }
}
```

Theremin

- “ooo-weee-ooooo”
- The original spooky sound machine
- Works by measuring your body’s electric field
- We’ll use a resistive sensor in lieu of RF
- Leo Theremin in his own words - video

Leon Theremin

Lets build our own theremin!

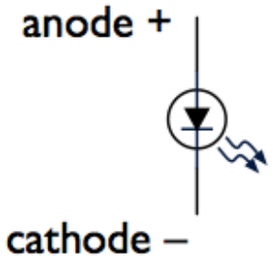



```
#define NOTE_C4  262
#define NOTE_G3  196
#define NOTE_A3  220
int sensorPinA = 4, speakerPin=3;

void setup() { Serial.begin(9600); }

void loop()
{
  int inputVal = analogRead(sensorPinA);
  Serial.println(inputVal);

  if (inputVal<=500)
      tone(speakerPin, NOTE_C4,8);
  if ((inputVal>500) && (inputVal<=900))
      tone(speakerPin, NOTE_G3,8);
  if (inputVal>900)
      tone(speakerPin, NOTE_A3,8);

   delay(10);
   noTone(speakerPin);
}
```

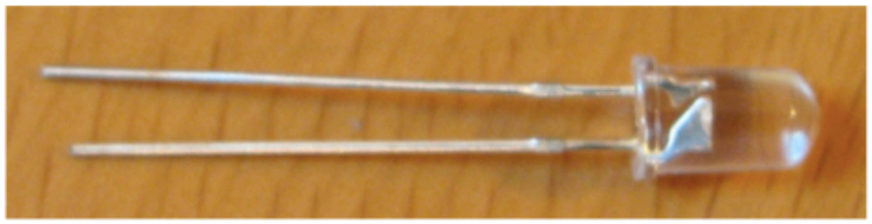
You pick!

RGB LED

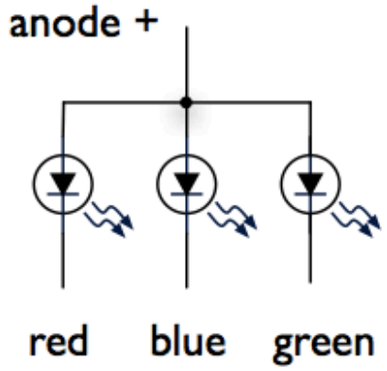
Normal LED



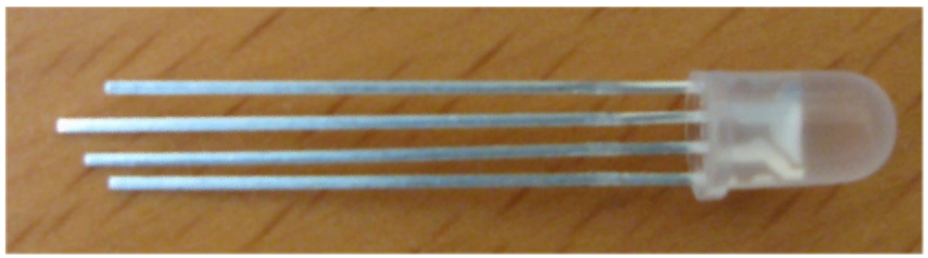
anode +
cathode –



RGB LED

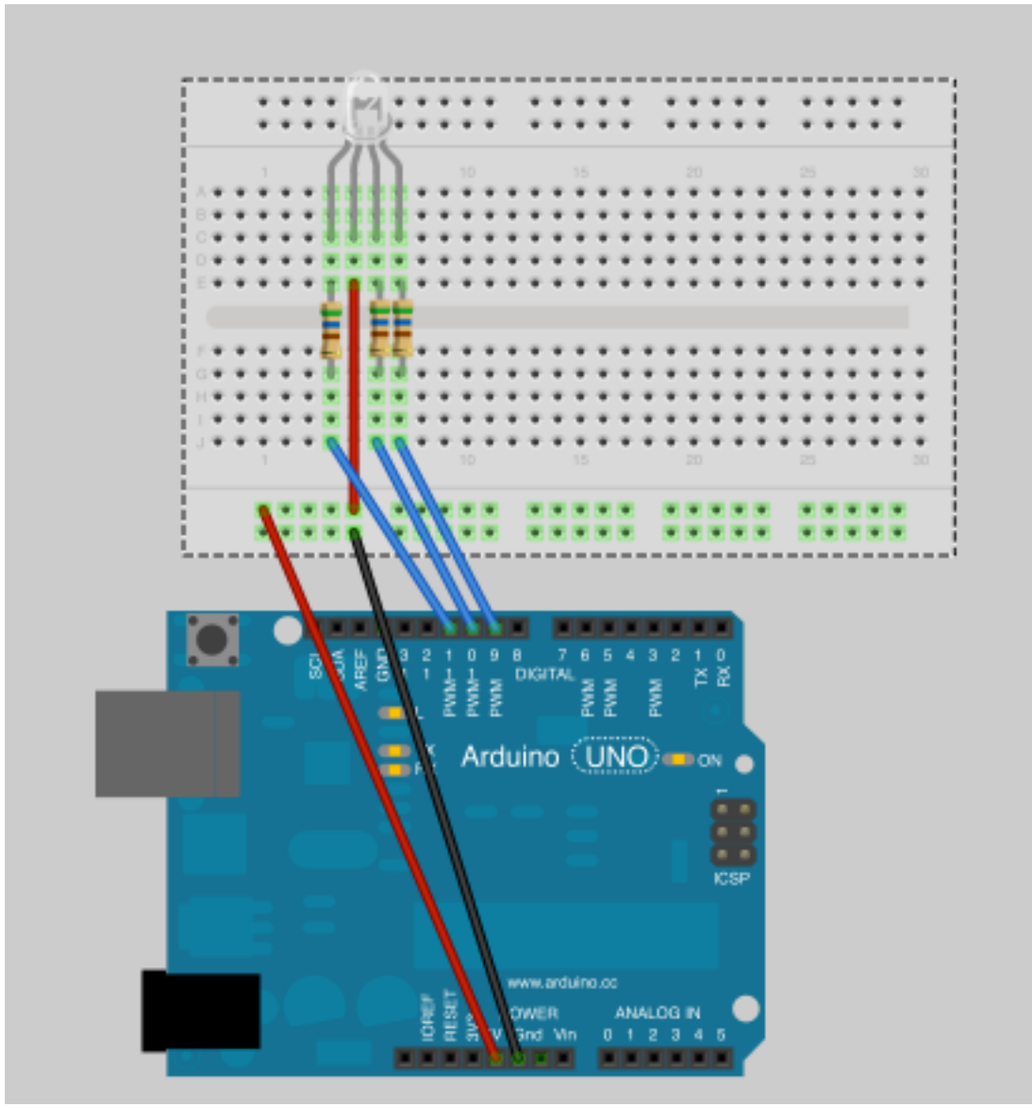


red cathode –
anode +
green cathode –
blue cathode –

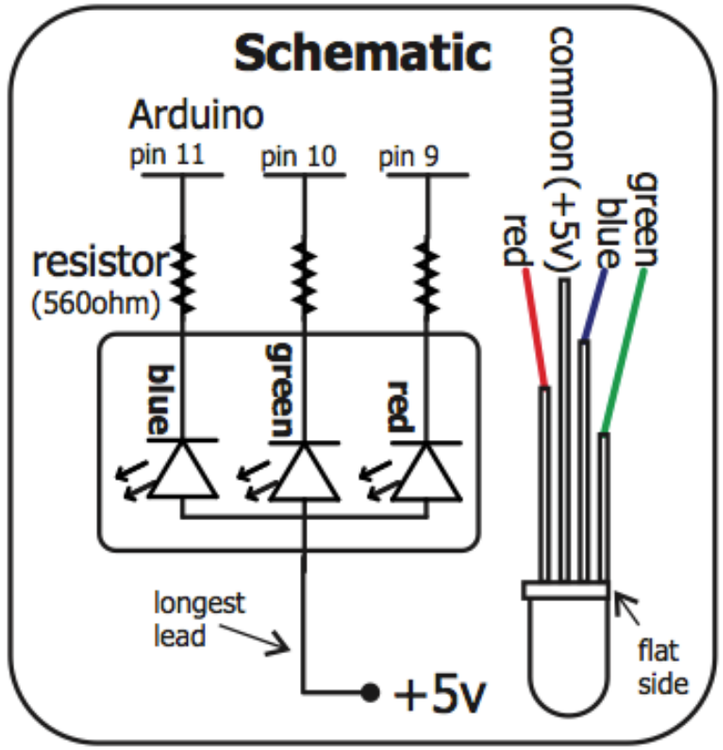


actually 3 LEDs in one package

Write up this circuit...
Then we write the code

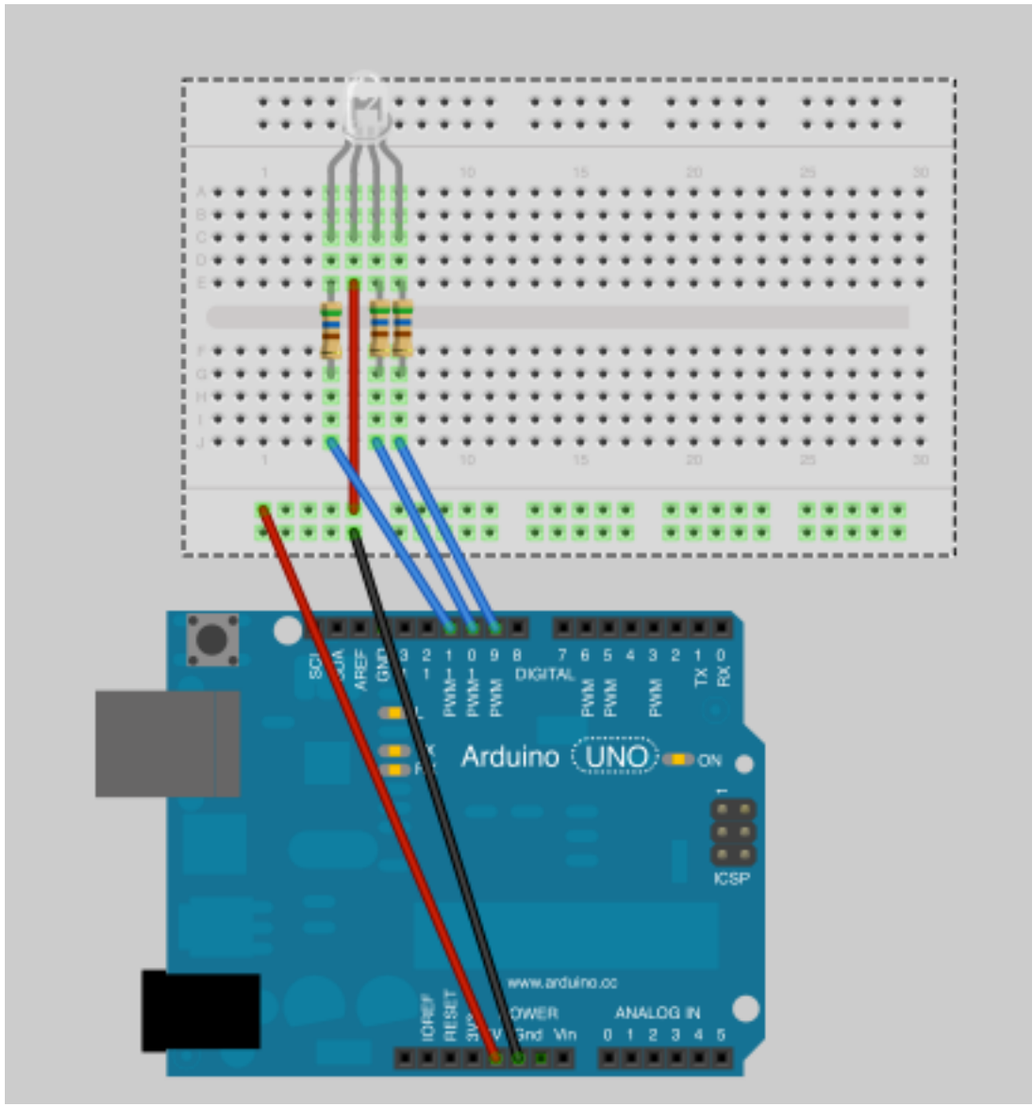


Color Truth Table			
red	green	blue	
ON	ON	OFF	yellow
OFF	ON	ON	cyan
ON	OFF	ON	magenta
ON	ON	ON	white



560ohm is Green, Blue, Brown

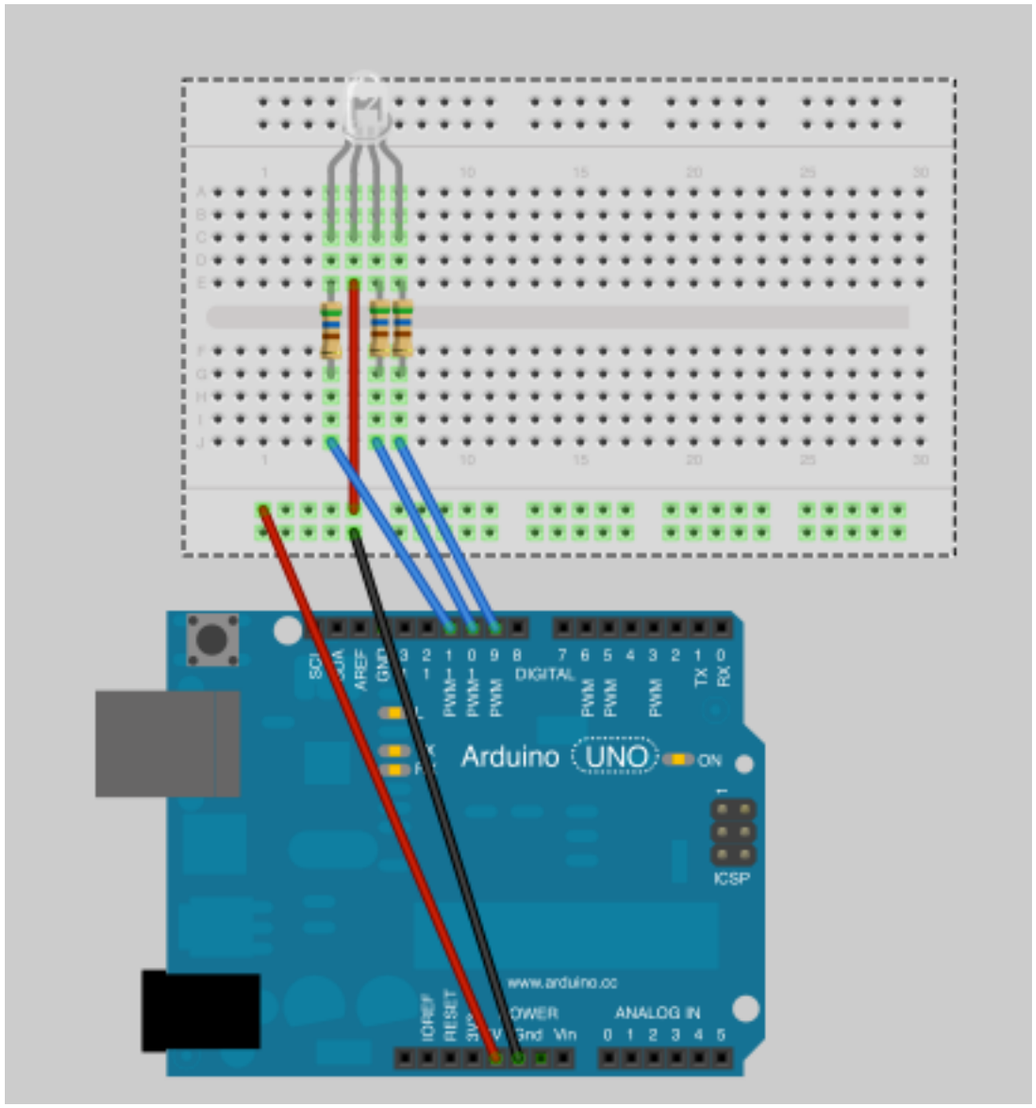
Green color



Color Truth Table			
red	green	blue	
ON	ON	OFF	yellow
OFF	ON	ON	cyan
ON	OFF	ON	magenta
ON	ON	ON	white

```
int red=9, green=10, blue=11;
void setup()
{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}

void loop()
{
  analogWrite(red,0);
  analogWrite(green,255);//255 is MAX
  analogWrite(blue,0);
}
```

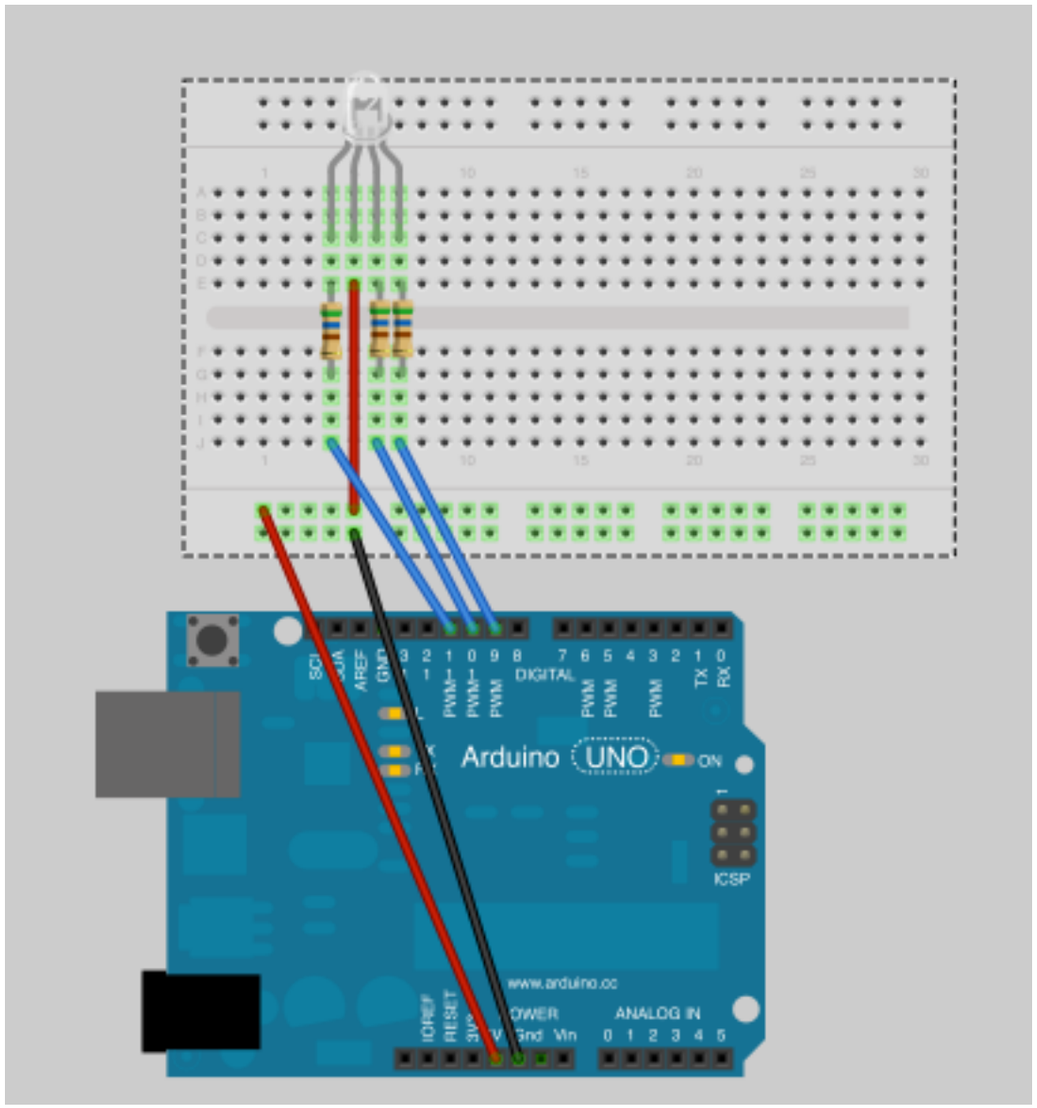
Blue color



Color Truth Table			
red	green	blue	
ON	ON	OFF	yellow
OFF	ON	ON	cyan
ON	OFF	ON	magenta
ON	ON	ON	white

```
int red=9, green=10, blue=11;
void setup()
{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}

void loop()
{
  analogWrite(red,0);
  analogWrite(green,0);
  analogWrite(blue,255);
}
```

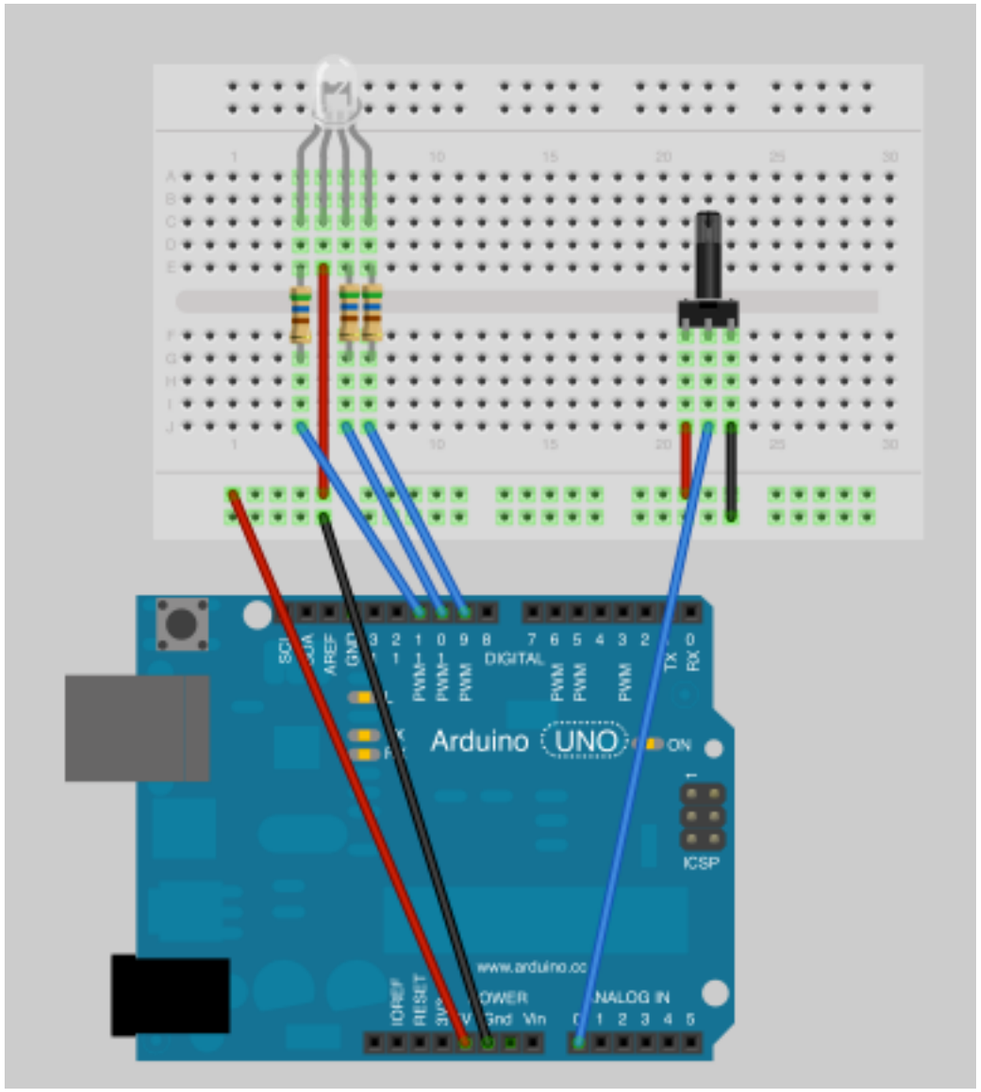
Generating random colors!




```
int red=9, green=10, blue=11;

void setup()
{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}

void loop()
{
  analogWrite(red,random(0,256));
  analogWrite(green,random(0,256));
  analogWrite(blue,random(0,256));
  delay(500);
}
```

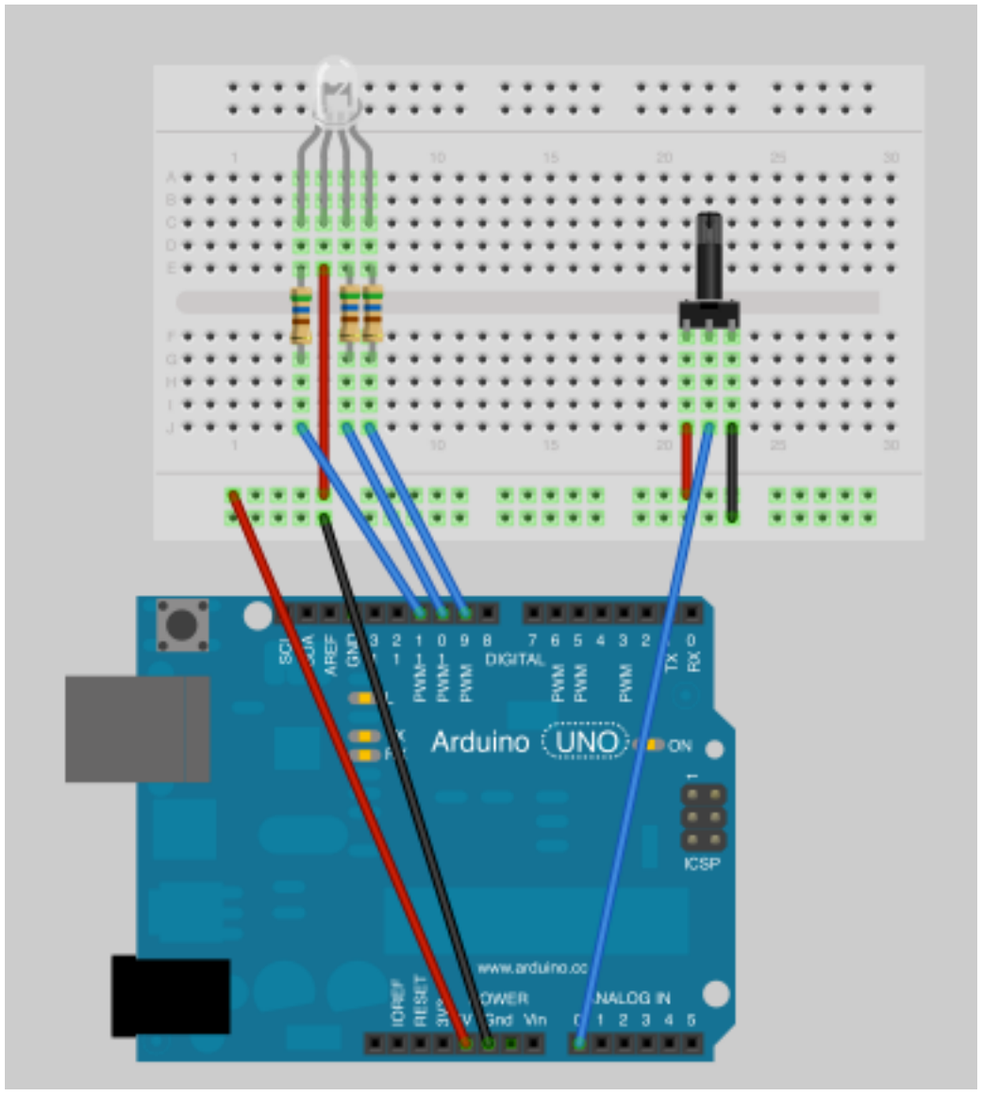
Every 0.5 seconds, a new color will be generated!

Lets use the potentiometer to generate new colors!




```
int red=9, green=10, blue=11;

void setup()
{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}

void loop()
{
  int pot=analogRead(0);
  int data = map(pot, 0, 1024, 0, 255);
  analogWrite(red,0);
  analogWrite(green,data);
  analogWrite(blue,0);
}
```

Use two(!) potentiometers to generate new colors!




```
int red=9, green=10, blue=11;

void setup()
{
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}

void loop()
{
  int pot1=analogRead(0);
  int pot2=analogRead(1);
  int data1 = map(pot, 0, 1024, 0, 255);
  int data2 = map(pot, 0, 1024, 0, 255);
  analogWrite(red, data1);
  analogWrite(green,data2);
  analogWrite(blue,0);
}
```

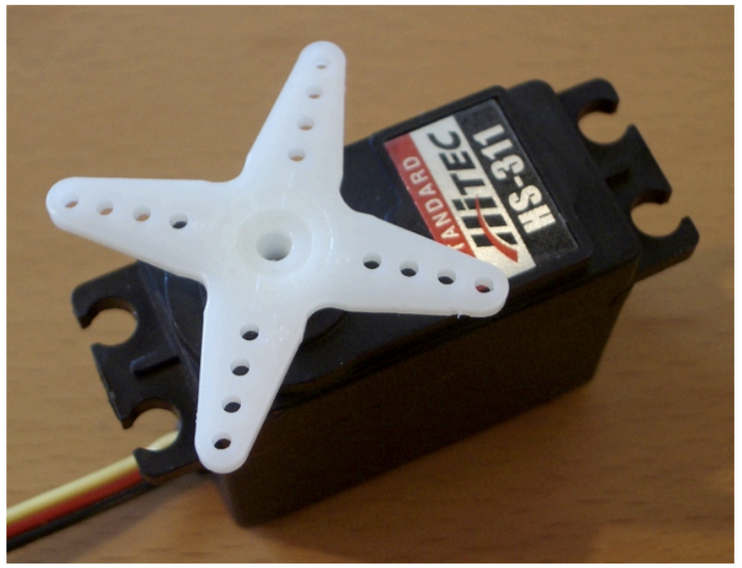
You have to put together the circuit... What about three potentiometers????

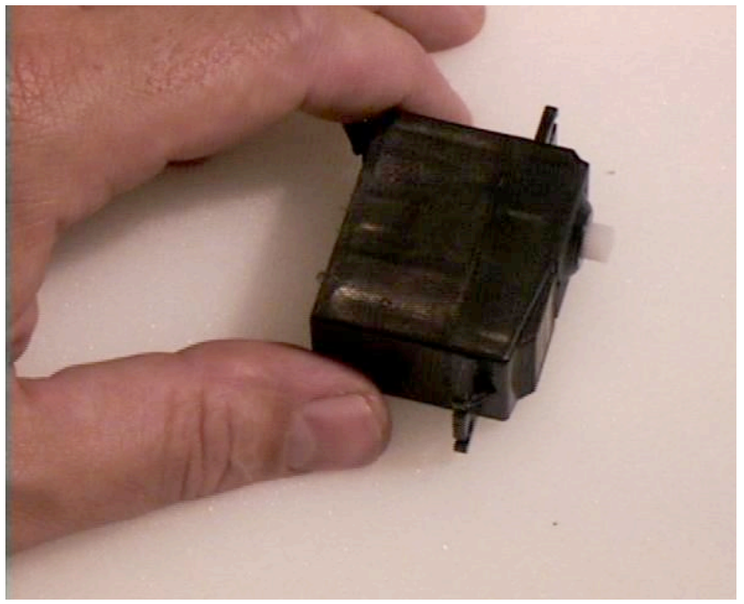
Servo motors

- A Servo is a small device that has an output shaft.

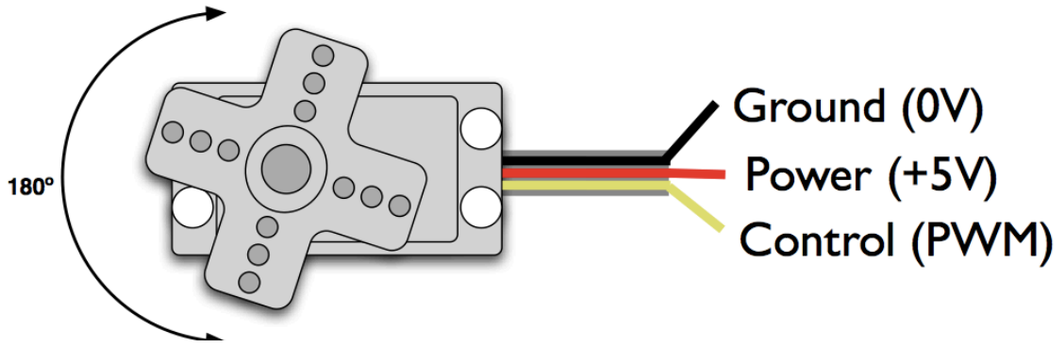
- This shaft can be positioned to specific angular positions (0-180 degrees) by sending the servo a coded signal.

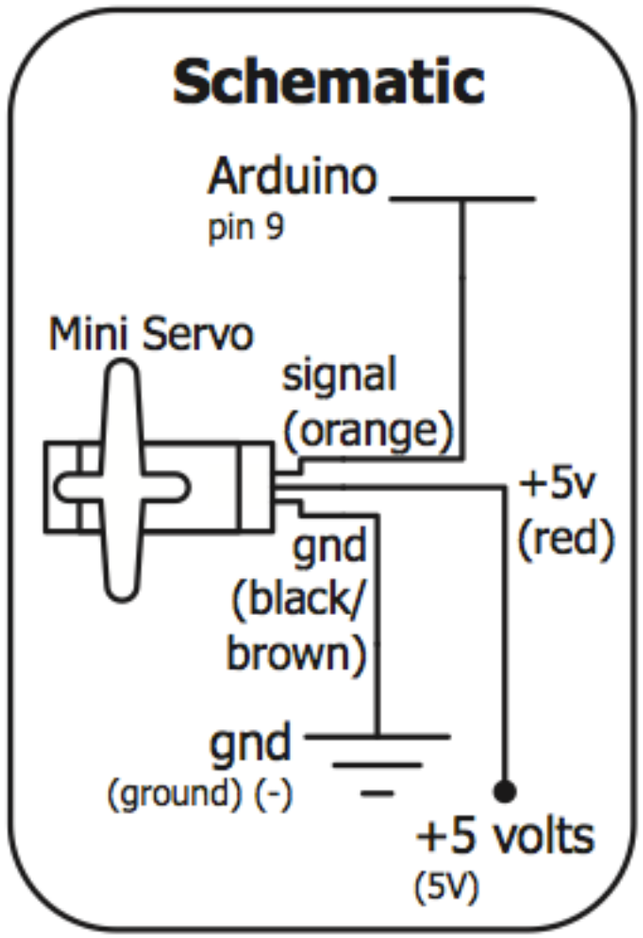
- As the coded signal changes, the angular position of the shaft changes.





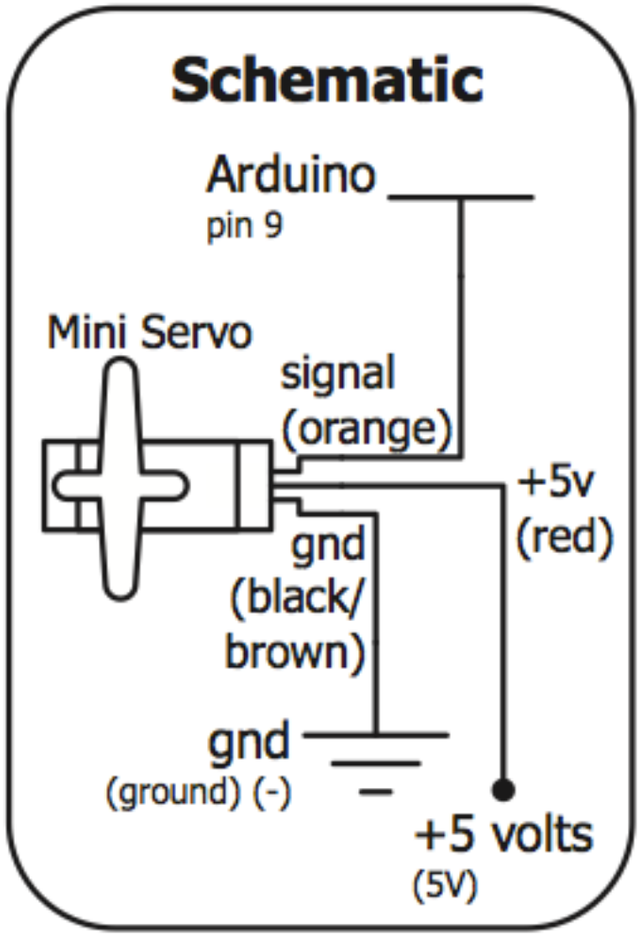
Moving a servo






```
#include <Servo.h>
Servo myservo; int pos = 0;

void setup() { myservo.attach(9); }

 void loop()
{
    //move servo to 45 degrees
    myservo.write(45);
    delay(1000); //wait 1 second

    //move servo to 90 degrees
    myservo.write(90);
    //wait 1 second
    delay(1000);
}
```

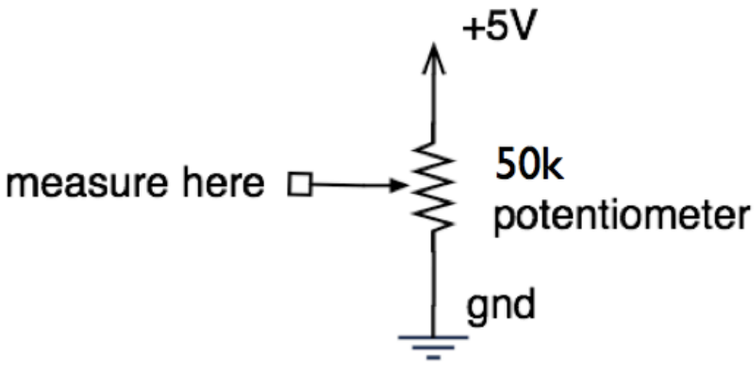
Moving a servo back and forth!




```
#include <Servo.h>
Servo myservo;
void setup() { myservo.attach(9); }

void loop()
{
  for (int i=0; i<180; i++)
  {
     myservo.write(i);
     delay(200);
  }

  for (int i=180; i>0; i--)
  {
     myservo.write(i);
     delay(200);
  }
}
```

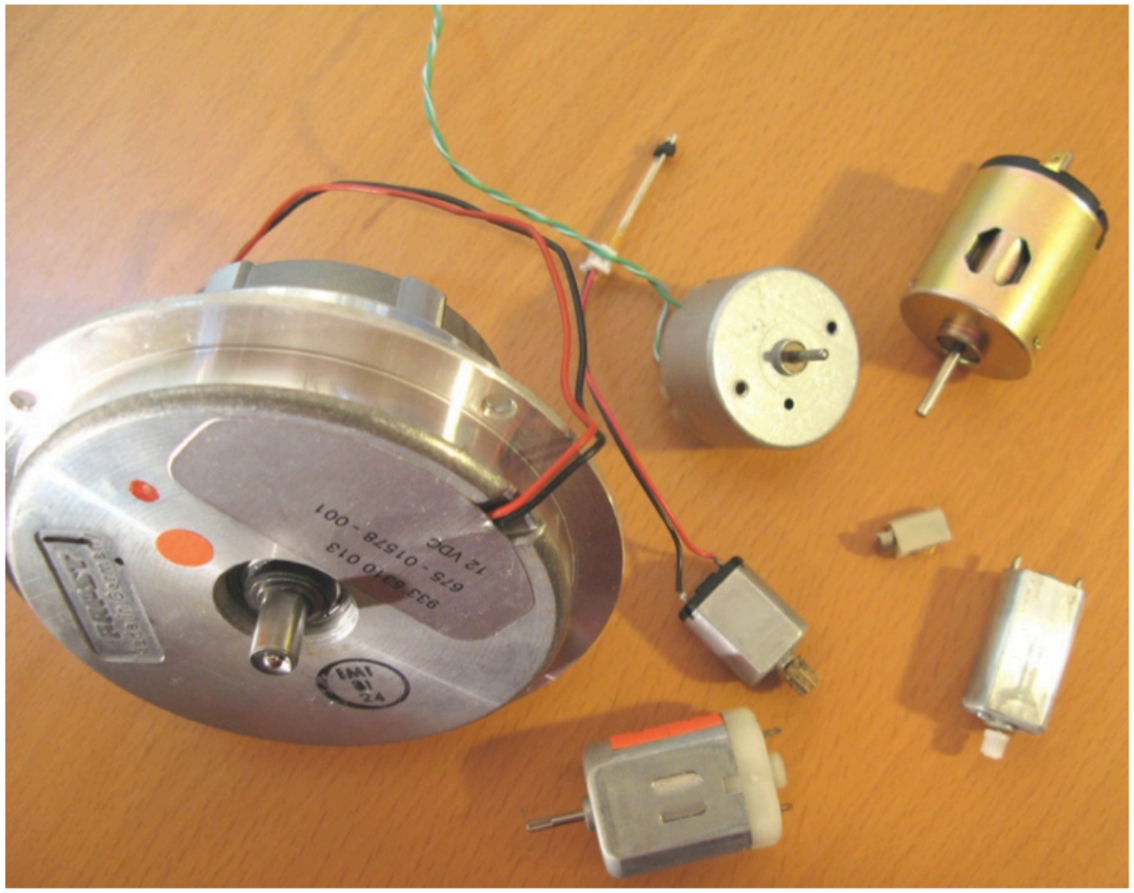
Using a potentiometer to specify the servo angle

- Instead of sweeping all angles, use a potentiometer to define an angle.
- You need to determine what are the potentiometer analog input ranges.




```
//connect potentiometer to analog pin #0
//connect servo control to digital pin#9
#include <Servo.h>
Servo myservo;

int potpin = 0;
int val;

void setup(){ myservo.attach(9); }

void loop()
{
  val = analogRead(potpin);
  // scale analog input to an angle
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
}
```

DC Motors

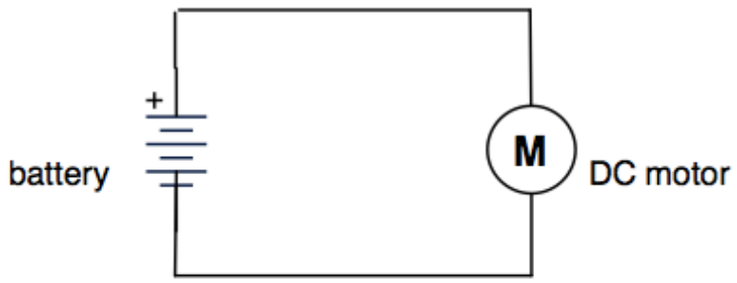


They come in all forms and sizes.

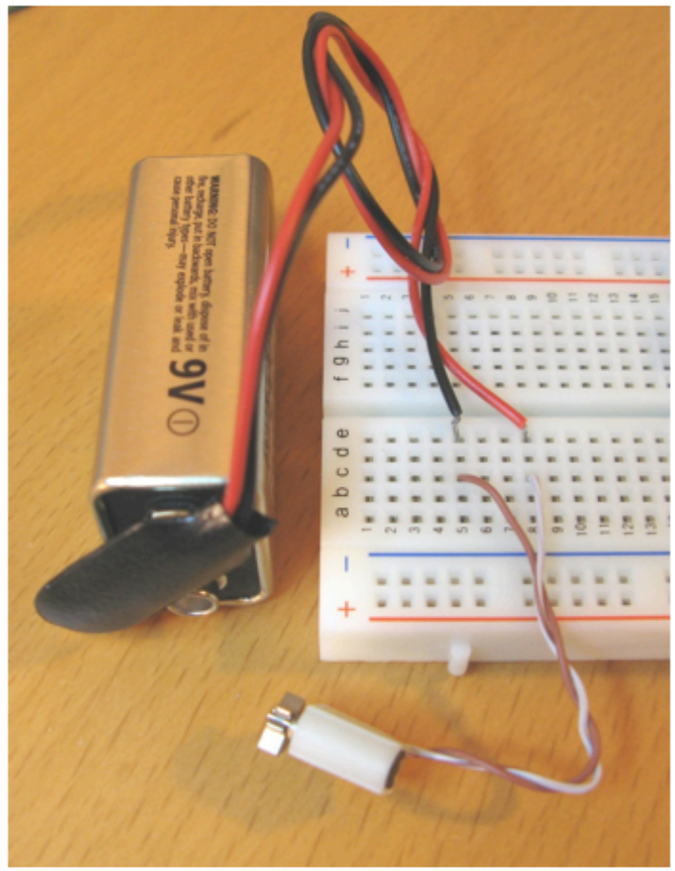
- With or without gears
- How much juice it needs to spin?
- How fast it spins
- How strong it spins
- Size, shaft diameter, shaft length,...

How to connect a DC motor?

To drive them, apply a voltage
The higher the voltage, the faster the spinning



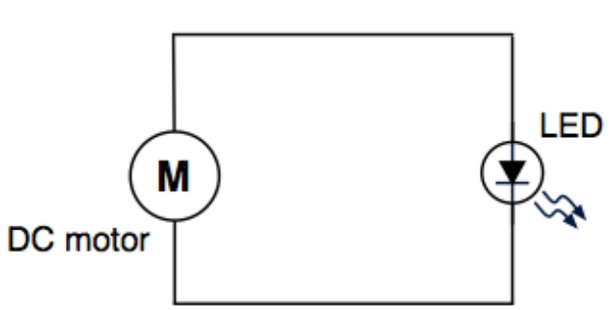
polarity determines which way it rotates



Try this out real quick.
Then swap polarity

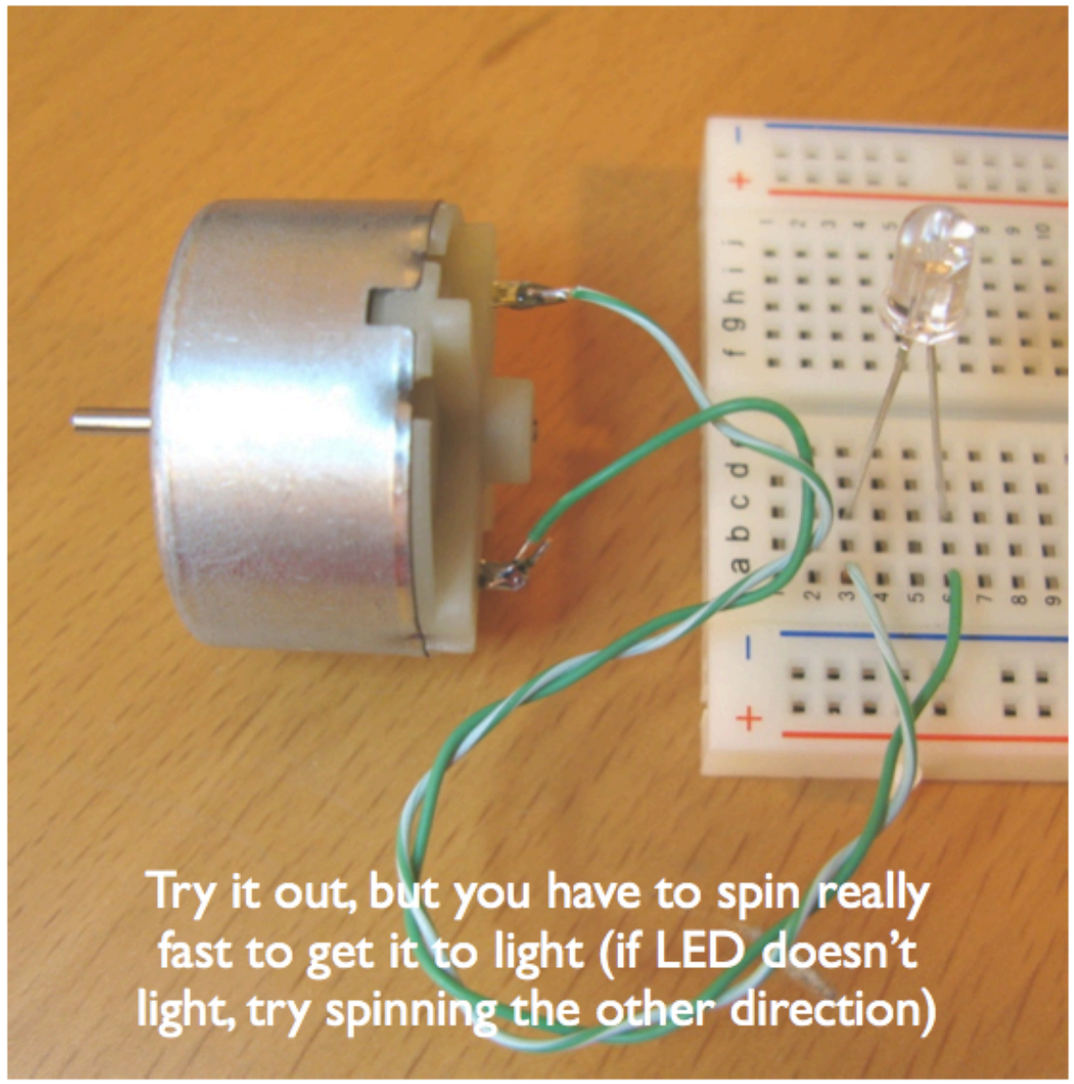
DC motors as generators

Just as voltage causes rotation...

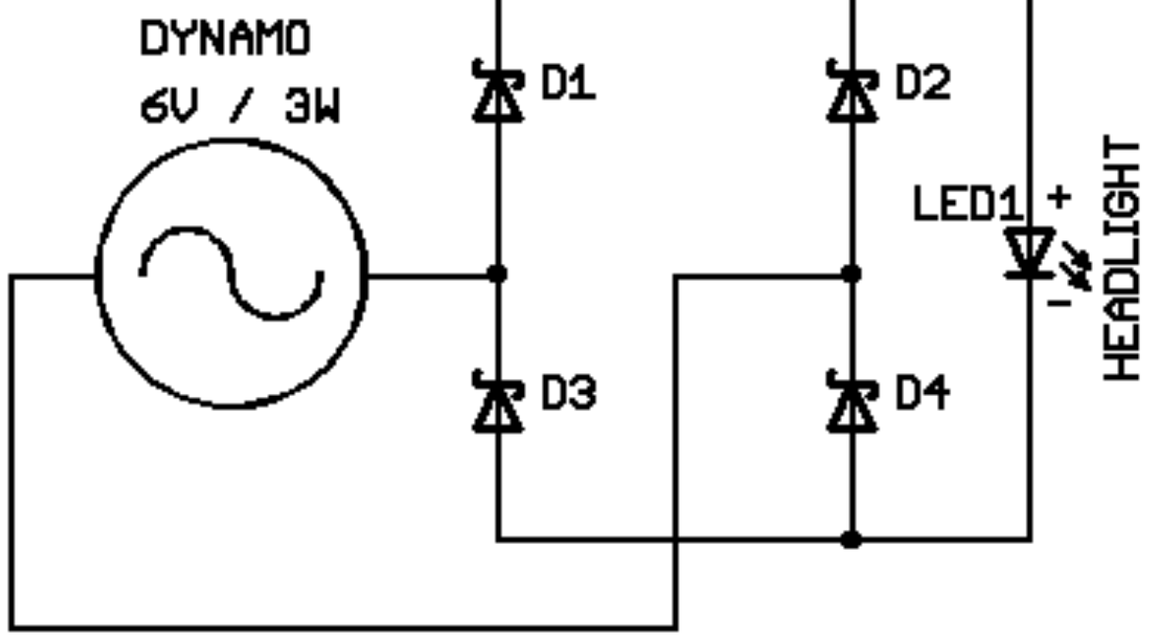


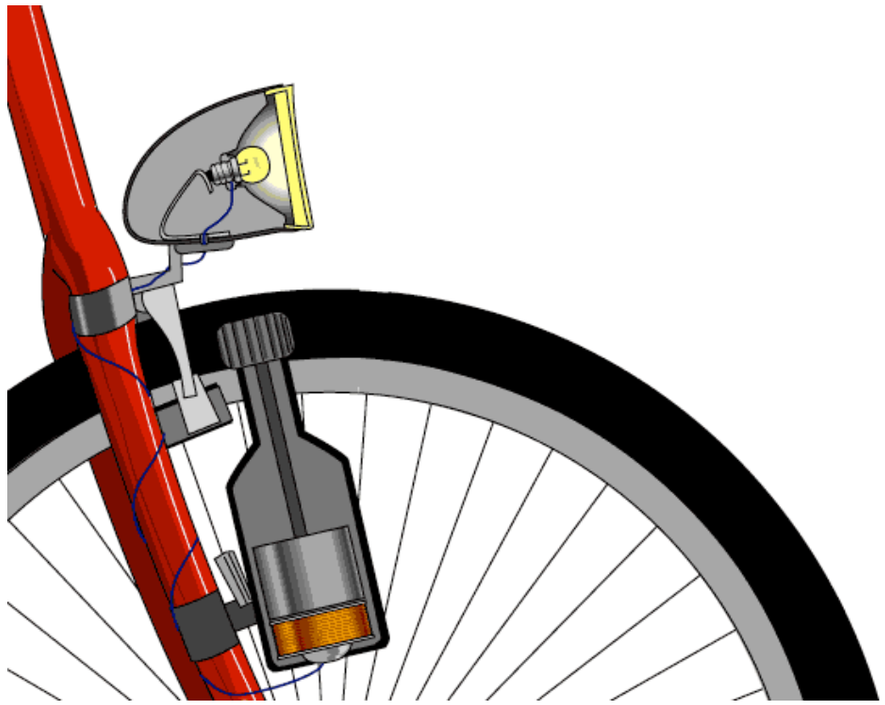
...rotation causes voltage

This is used for "regenerative braking" in electric & hybrid cars



Same idea as a dynamo

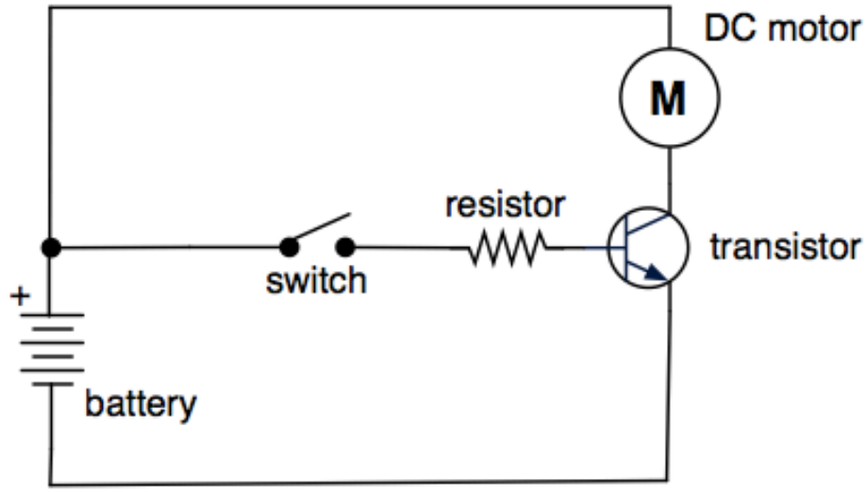




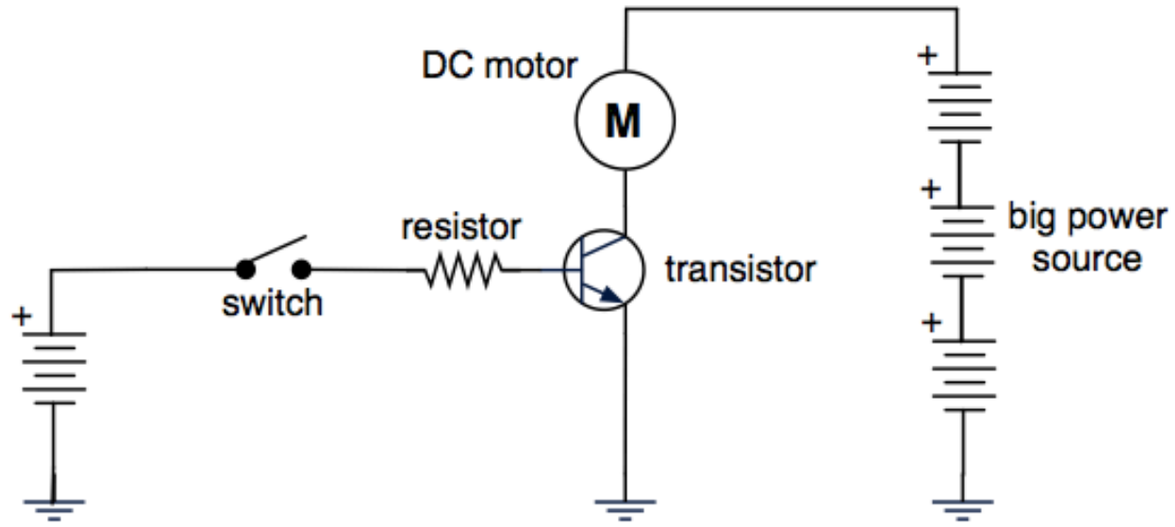
Transistors and motors

- Transistors are small switches. A little juice in once leg, will close the switch.

little motor



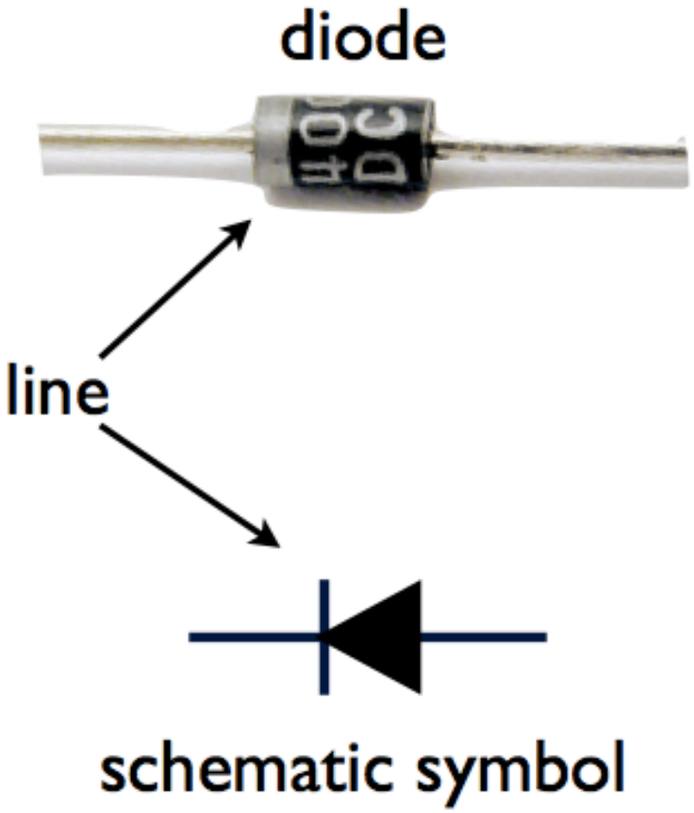
big motor

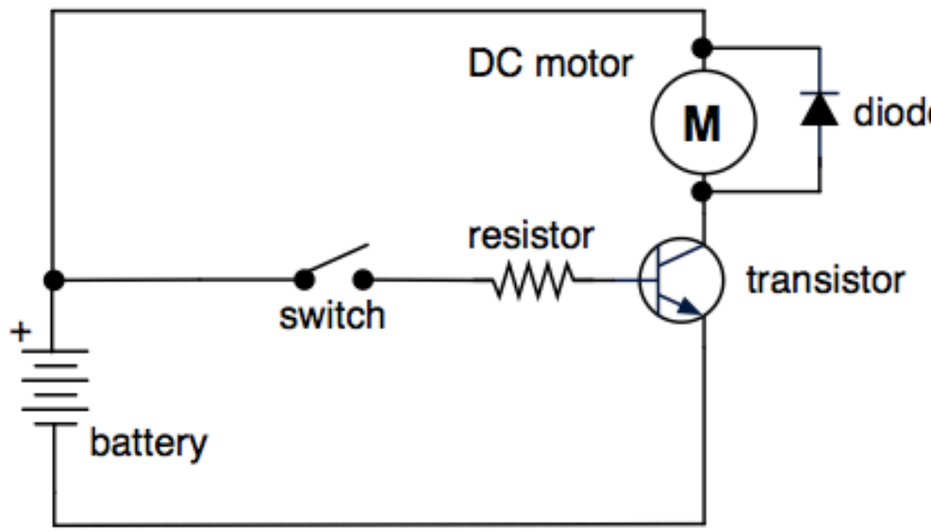


switching a different power source

- Transistors switch big signals with small signals.

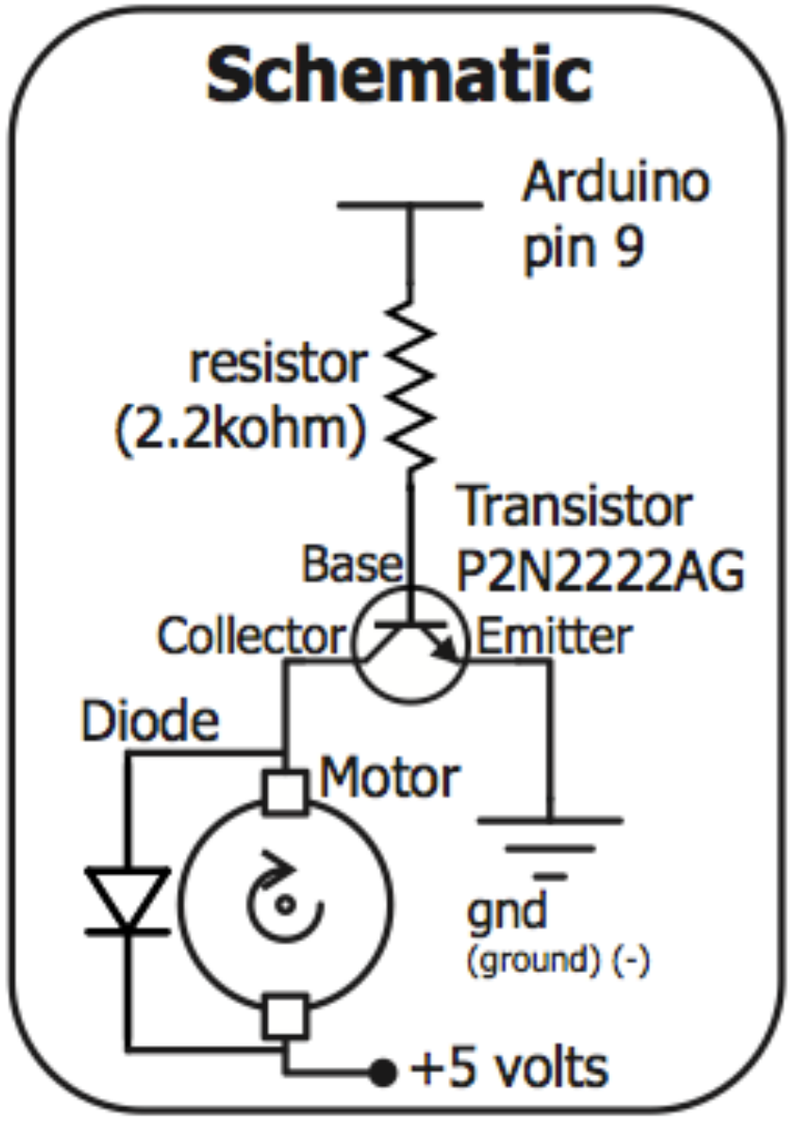
Protecting your circuit



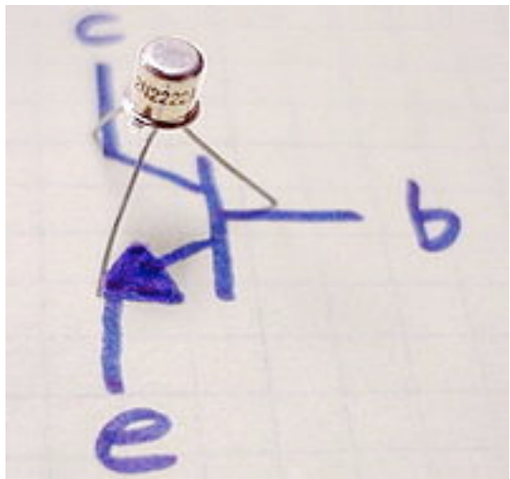


- Since motors can act like generators, need to prevent them from generating juice back into the circuit.

Run a DC motor




```
const int DCMotorpin=9;
void setup()
{
  pinMode(DCMotorpin, OUTPUT);
};

void loop() {
  analogWrite(DCMotorpin,200);
};
```

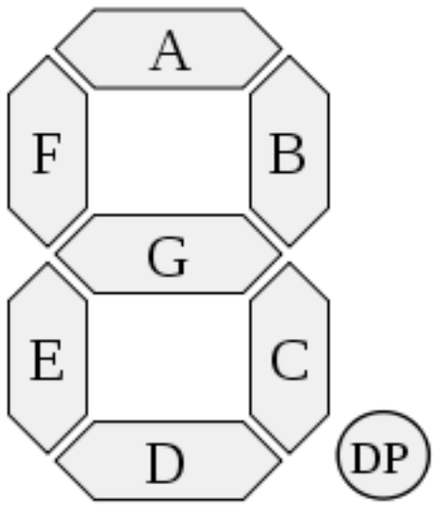


- Implement the DC motor circuit and run it. Change the analog values, for different speeds.
- Adjust the speed of the motor using a potentiometer.
- Adjust the speed of the motor using any other resistive sensor.

7-Segment LED Displays

- Old school method of displaying numbers.
- A seven segment display has seven LEDS.
- You can turn each LED ON/OFF to display a number.

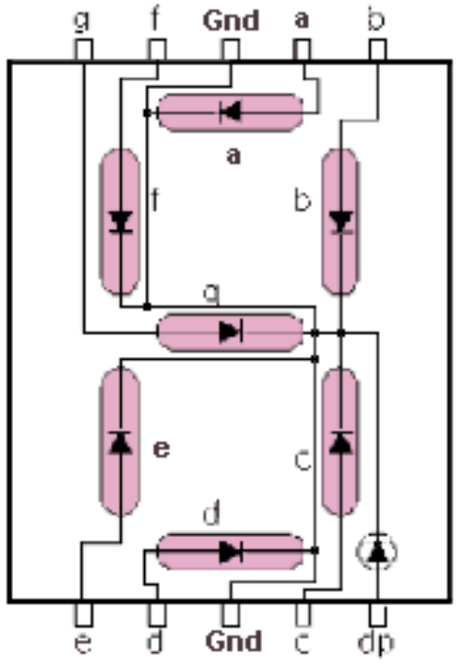
7-Segment display



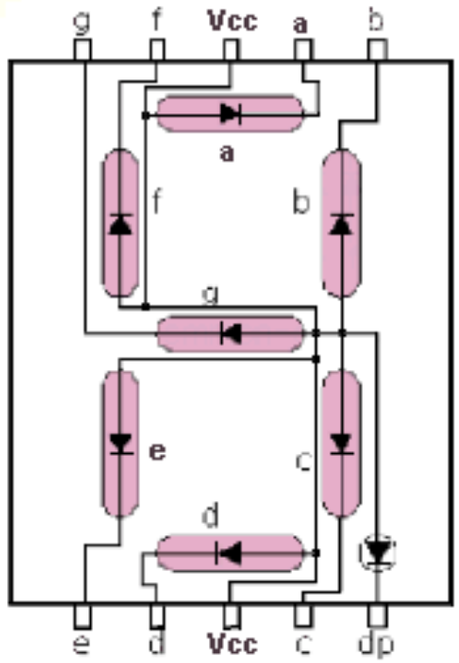
- The segments of a 7-segment display are referred to by the letters A to G.
- Some displays have a decimal point.
- Circuit connections vary from different manufacturers.

Digit Shown	Illuminated Segment (1 = illumination)						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Common Cathode

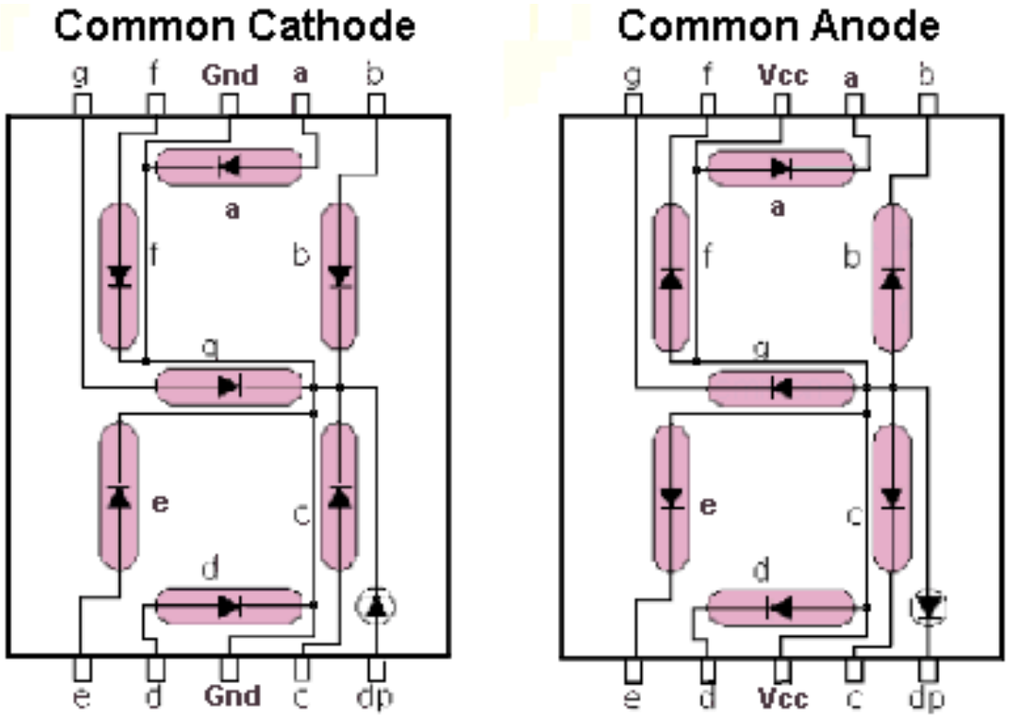


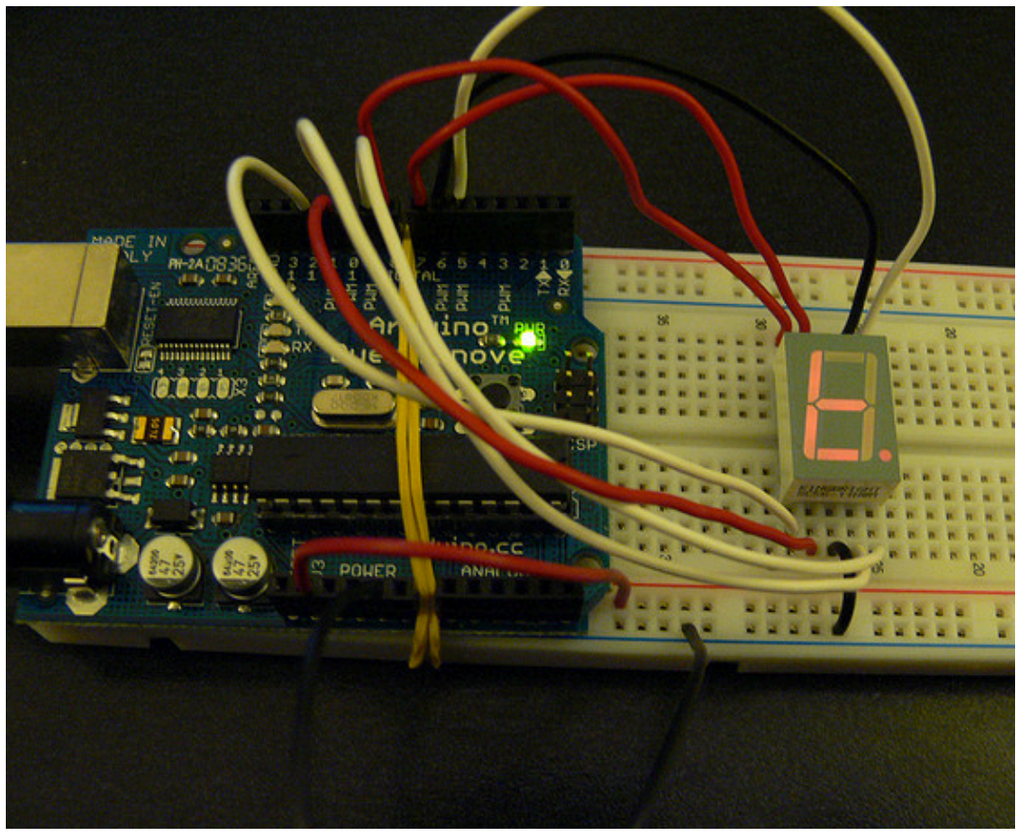
Common Anode



How does it work?

- Determine the connections of the 7-Segment display I've handed out.





Type number on keyboard ... show it on the 7-Segment!

```
int pinA=4; int pinB=7; int pinC=9; int pinD=10;
int pinE=5; int pinF=6; int pinG=8; int readKey=-1;

void setup()
{
  pinMode(pinA,OUTPUT);  pinMode(pinB,OUTPUT);
  pinMode(pinC,OUTPUT);  pinMode(pinD,OUTPUT);
  pinMode(pinE,OUTPUT);  pinMode(pinF,OUTPUT);
  pinMode(pinG,OUTPUT); Serial.begin(9600);
}

void loop()
{
  while (Serial.available()>0) readKey = Serial.read();

  if (readKey=='0'){
    digitalWrite(pinA,LOW);    digitalWrite(pinB,LOW);
    digitalWrite(pinC,LOW);    digitalWrite(pinD,LOW);
    digitalWrite(pinE,LOW);    digitalWrite(pinF,LOW);
    digitalWrite(pinG,HIGH);
  }
 }
```