

Workshop: Wearable E-Textiles



FASHION DESIGN and TECHNOLOGY LAB

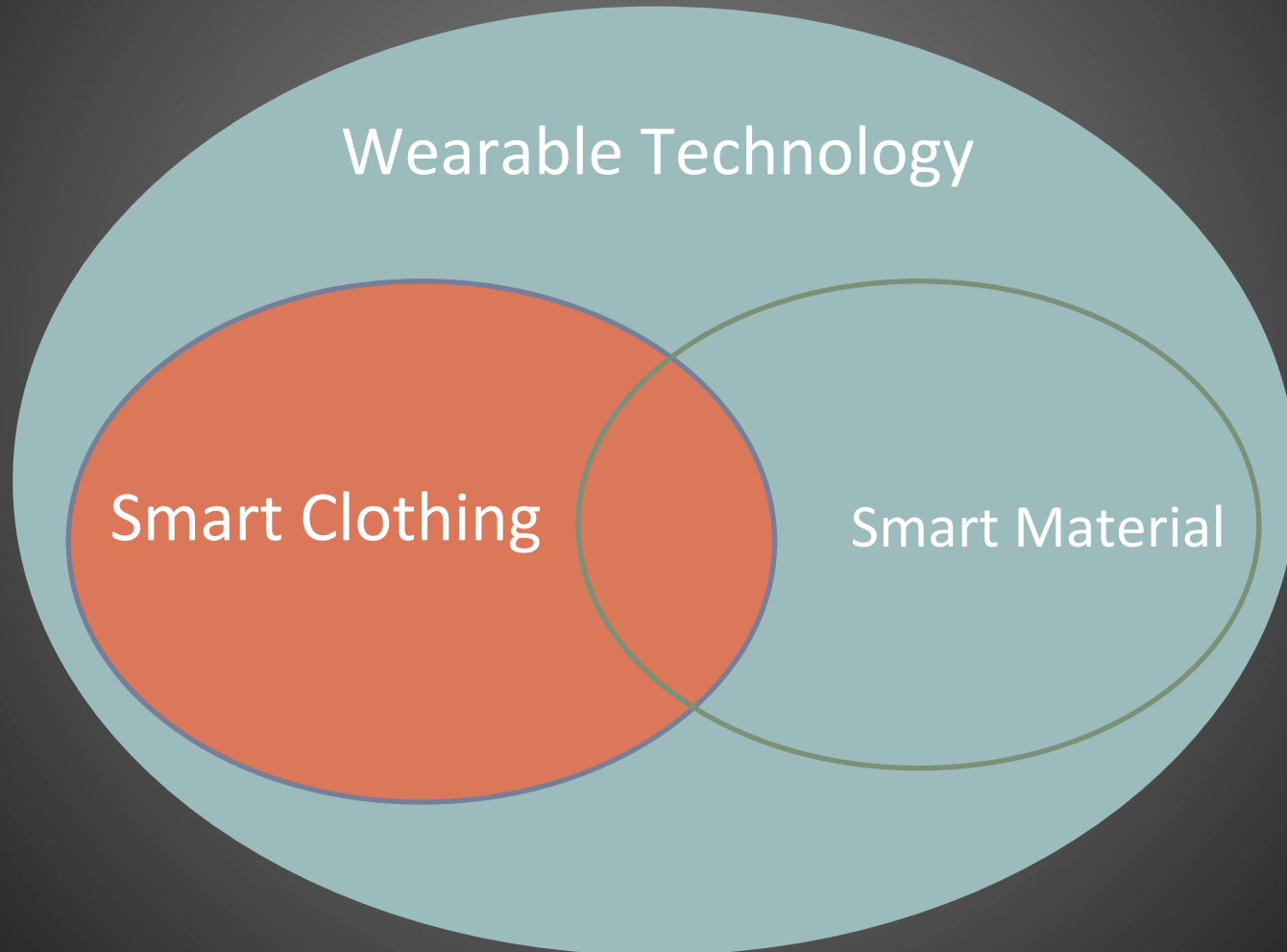
**MEDIALAB
PRADO**

Dr. Helen S. Koo

University of California, Davis
Department of Design
Fashion Design and Technology Lab (FT Lab)

hskoo@ucdavis.edu
www.helenskoo.com

1. What is Wearable Technology?

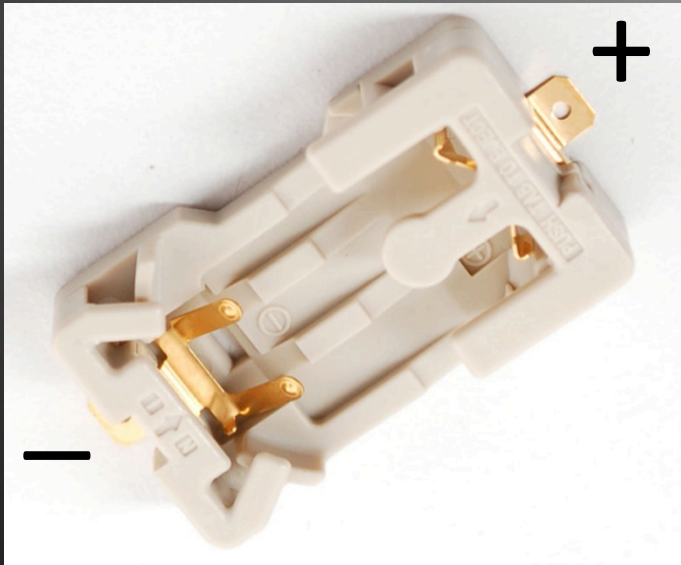


DESIGN 1

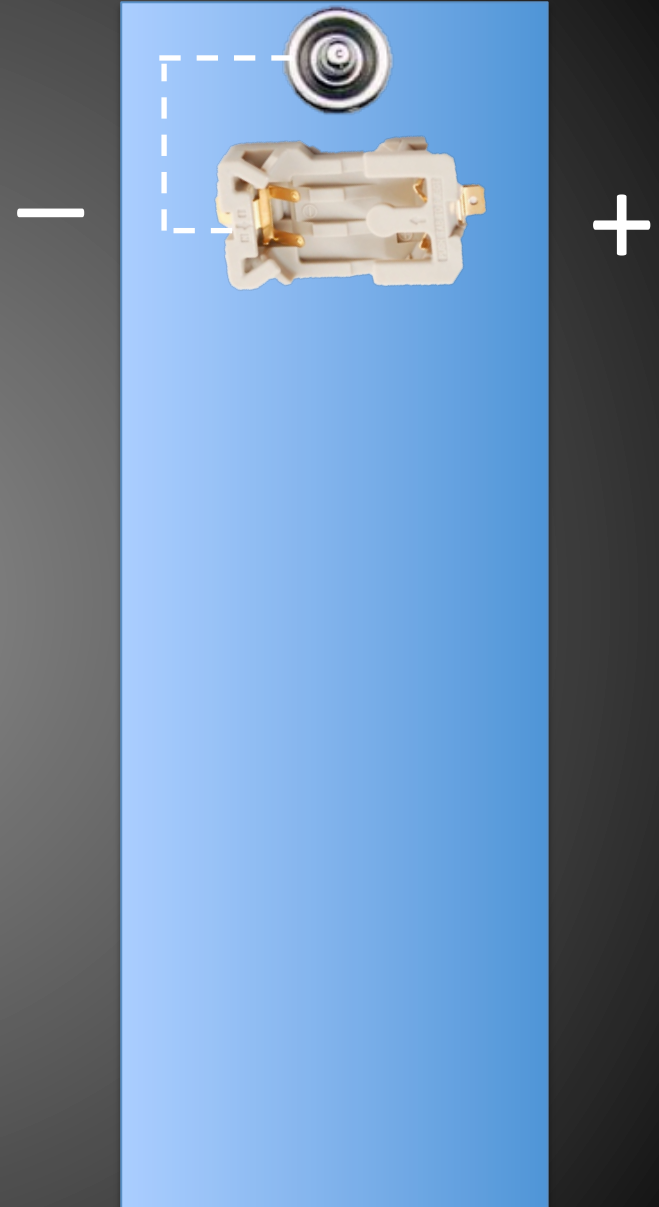
SOFT E-CIRCUIT: BRACELET

LED Bracelet

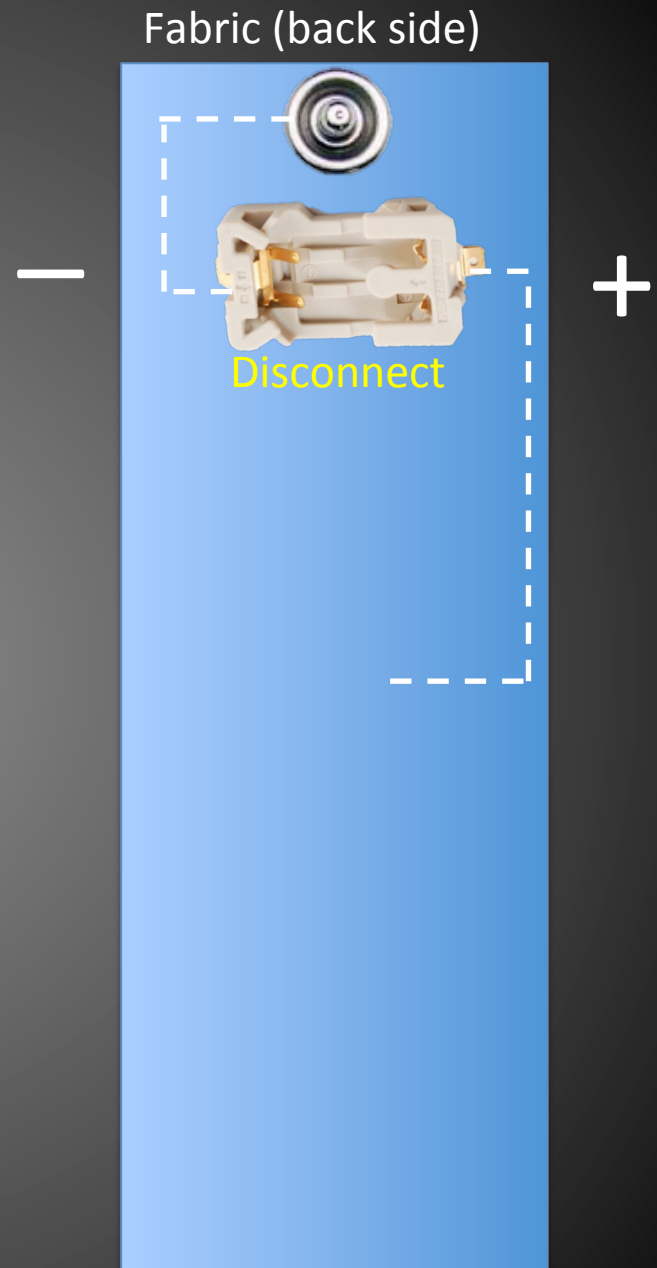
1. On the back side of the fabric, attach a battery cap. Sew the negative side of the battery holder to a male snap. A few stitches through the holes. Use conductive threads.



Fabric (back side)



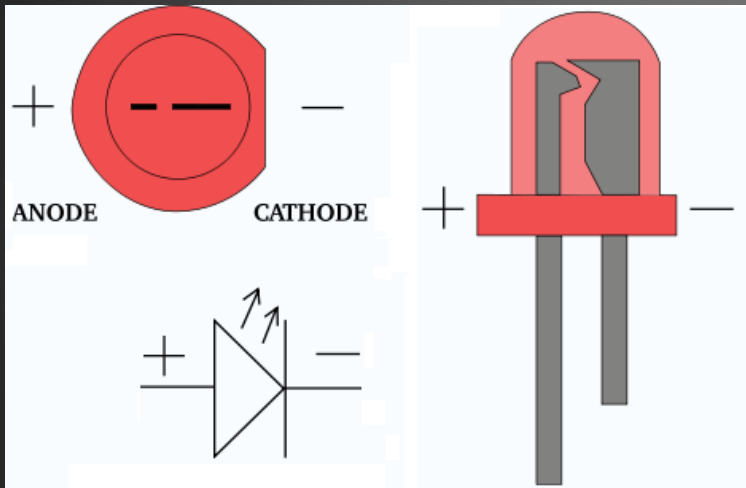
2) Stitch negative of the battery holder to the felt and stitch to the center.



3) Use tweezers or pliers, bend the LED leads and spiral it around until you are almost all the way to the bulb.



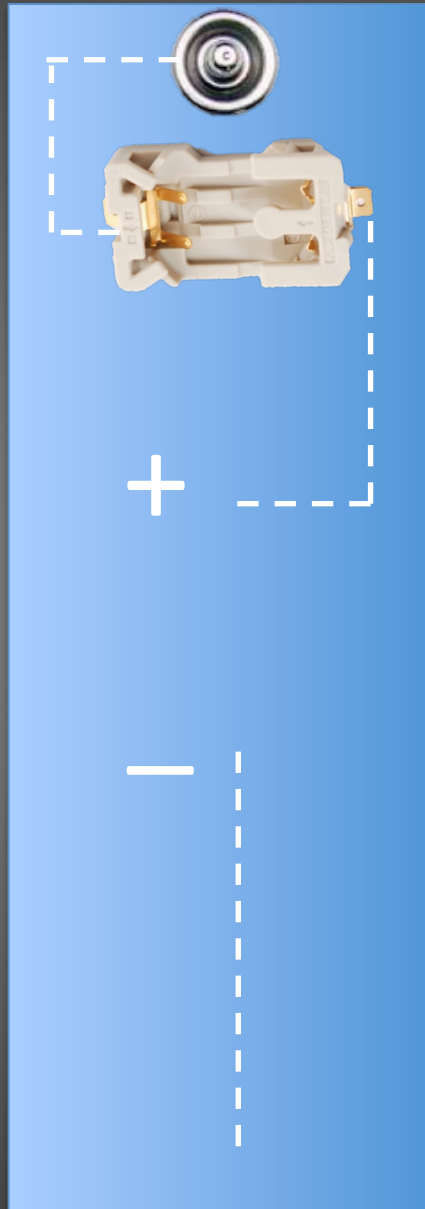
- LEDs are polarized: The current from the battery can only run through them in one direction
- **(+) Positive side → long lead**
(-) Negative side → short lead



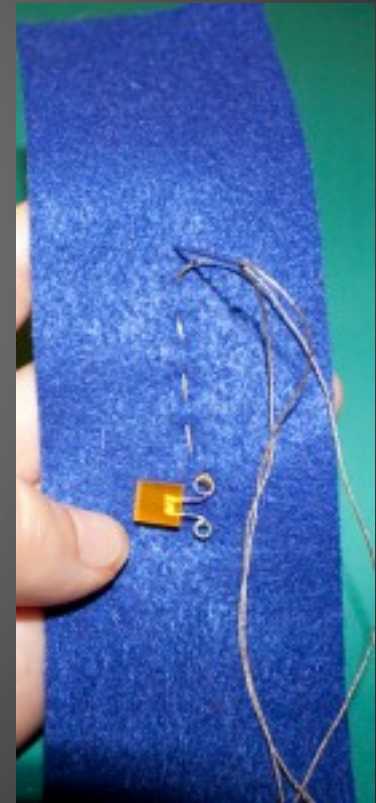
Fabric (front side)



Fabric (back side)



4) Connect positive side of battery holder to the positive side of the LED.



5) Connect the negative side of the LED and stitch until you reach the end of the fabric. Attach the other side of the snap on the front side of the fabric.

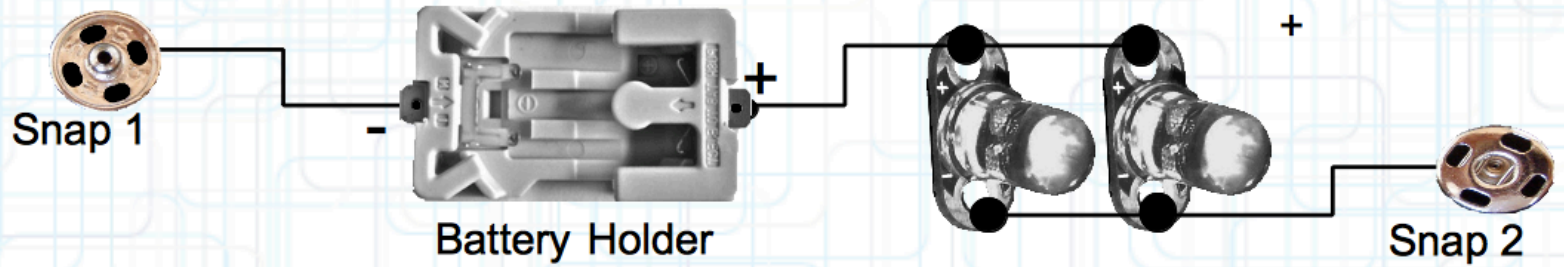
Fabric (front side)



6) Snap the battery into the holder and cover the battery using electrical tape or fabrics.



Adding more LEDs:



Two halves of snap go on opposite sides of felt

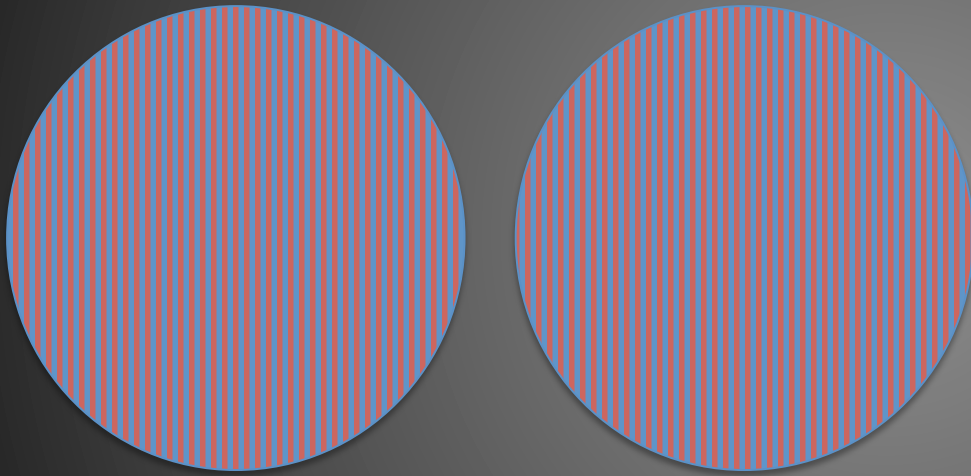




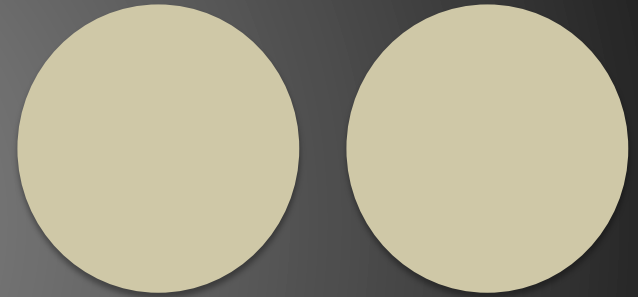
Fabric Button

<https://www.youtube.com/watch?v=ze5Wn3Bgshs>

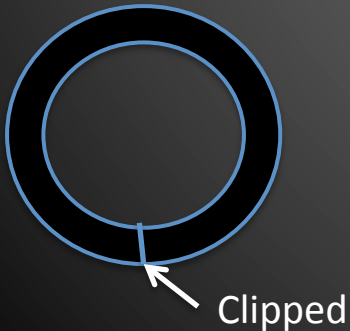
1. Prepare: 2 pieces of regular fabric circles



2 pieces of conductive fabric



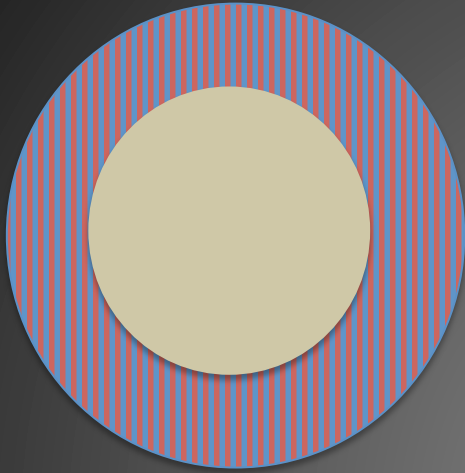
1 foam piece



1 pair of snap button

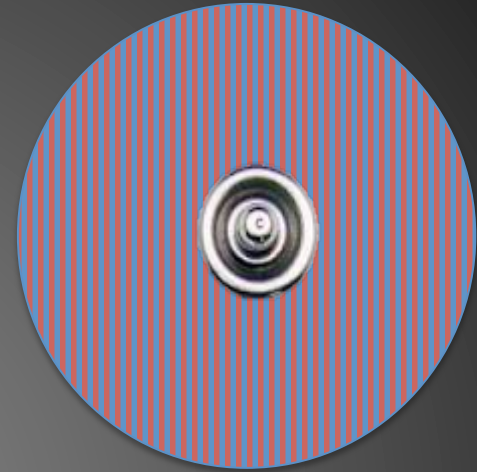


Inside



2. Place conductive fabric over the regular fabric

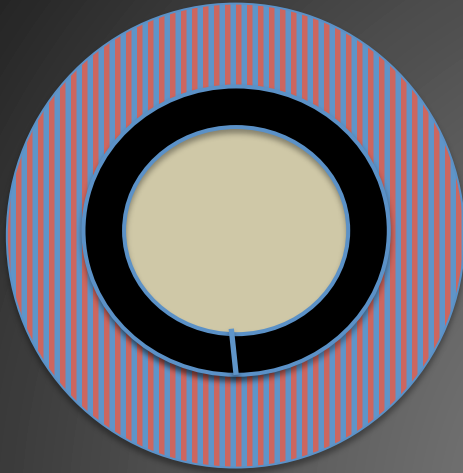
Outside



3. Sew snap button, conductive fabric, and regular fabric altogether with conductive threads

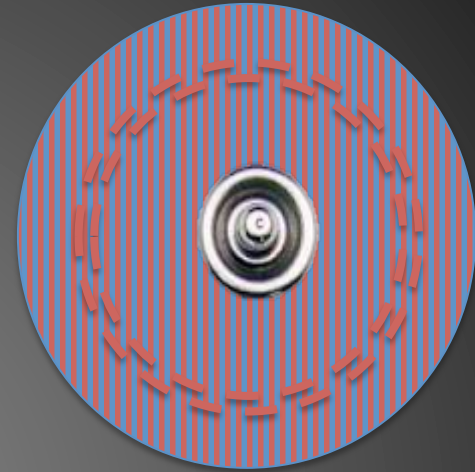
4. Do the same thing to the other piece to prepare cover and bottom pieces

Inside



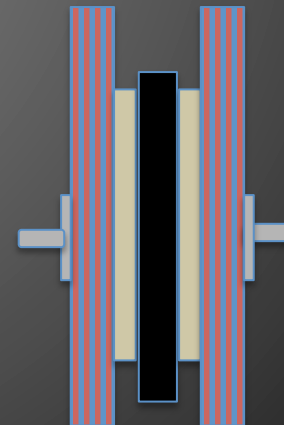
5. Place the foam piece over the conductive fabric

Outside



6. Use a regular thread and the cover and bottom pieces together following the foam edges

7. Attach the completed button between the bracelet opening where the snap buttons are



Side view



Lighting Technology

 [Buy](#) [Download](#) [Getting Started](#) [Learning](#) [Reference](#) [Hardware](#) [FAQ](#)

Official Arduino boards



Arduino Uno



Arduino Leonardo



Arduino WiFi Shield



Arduino Mega 2560



Arduino LilyPad



Arduino Ethernet Shield



Arduino Mega ADK



Arduino Fio



Arduino Wireless SD Shield

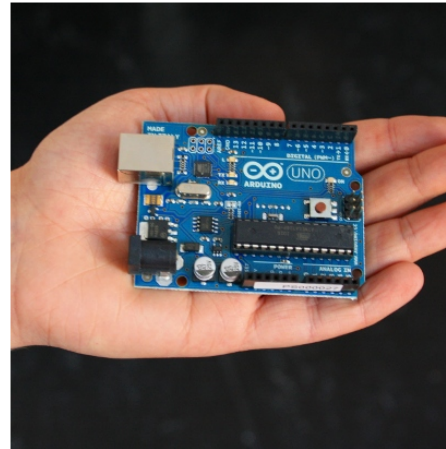


Photo by the Arduino Team

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-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 **Arduino programming language** (based on **Wiring**) and the Arduino development environment (based on **Processing**). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, MaxMSP).

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

Arduino received an Honorary Mention in the Digital Communities section of the 2006 Ars Electronica Prix. The Arduino team is: **Massimo Banzi**, **David Cuartielles**, **Tom Igoe**, **Gianluca Martino**, and **David Mellis**. [Credits](#)

Arduino on Twitter ([more](#))

[about 7 hours ago](#)

RT @mbanzi: @florianoroma arduino is not in the business of selling books, Make does that very effectively :)

[about 7 hours ago](#)

@FMMT666 OSHW and Trademark are two different things...

www.arduino.cc ➔ Download the program

Readings: <https://www.arduino.cc/en/Guide/Introduction>

- Open-source
- Microcontroller (Processor + Memory)
- Prototyping
- C-language

Building Circuits/ Gadgets

Power

Provide electricity

Battery
(e.g., coin, AA, 3V...)



Input

Interpret data

Button/Switch
(e.g., push buttons,
on/off switch...)



Sensor
(e.g., light, motion,
temperature...)



Output

Make changes
(e.g., visual, physical)

Light (e.g., LED)



Movement
(e.g., motor, fan)



Audio/speaker



Wire

Route of power and
communication
between
components

Solid wire
(e.g., copper)



Textile-based
conductive
material
(e.g., conductive
thread)



Power

4.5 volts

D

C

AA

AAA

AAAA

A23

9-volts



LR44 (1.5volts)

Common in
Europe and
Russian
Federation

(D~AAAA) 1.5 volts

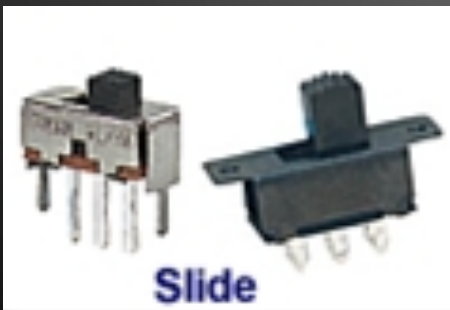
RF products

CR2032
(3.0 volts)

Input

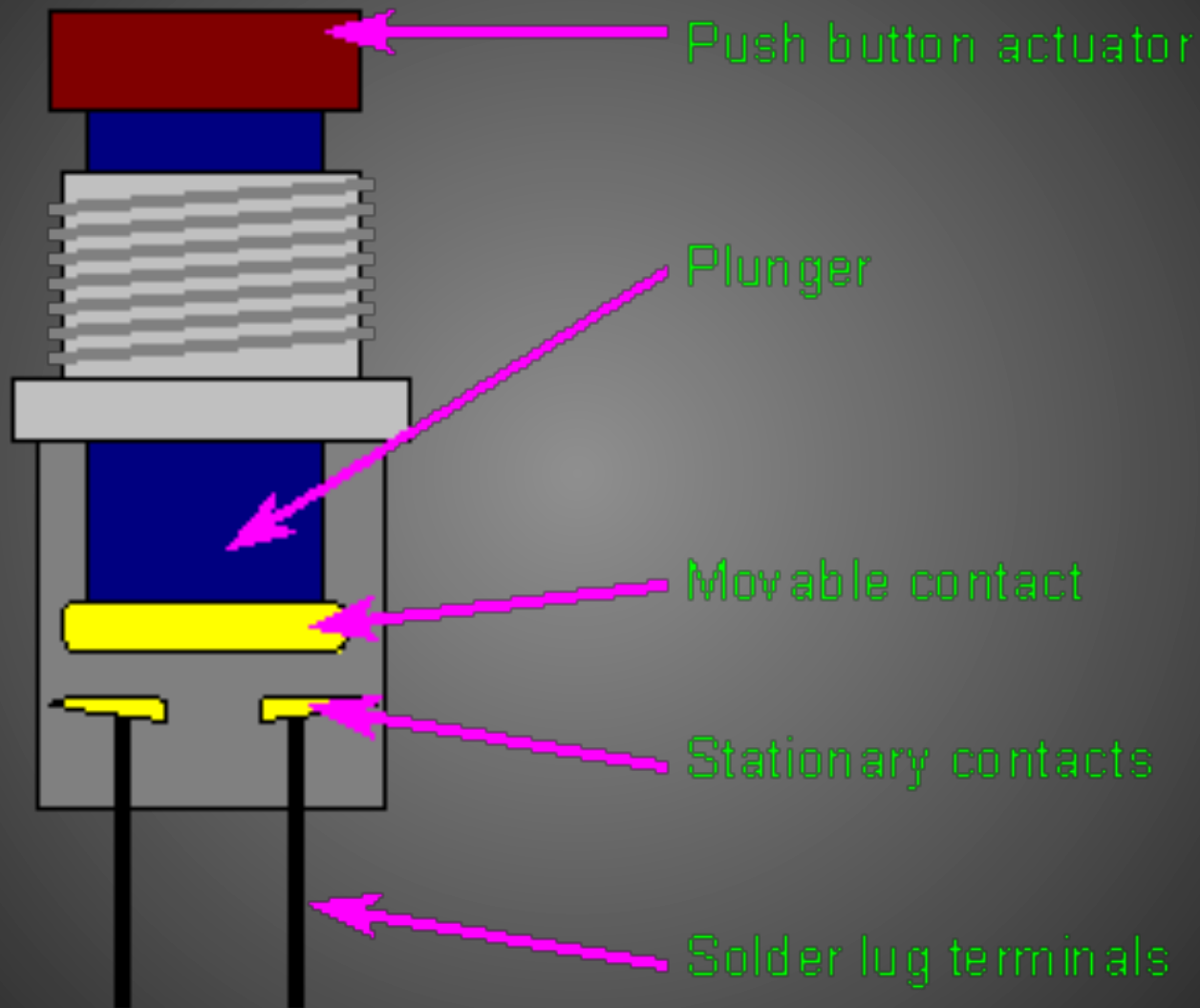
- **Switch**

: Electrical component that can connect and disconnect an electrical circuit



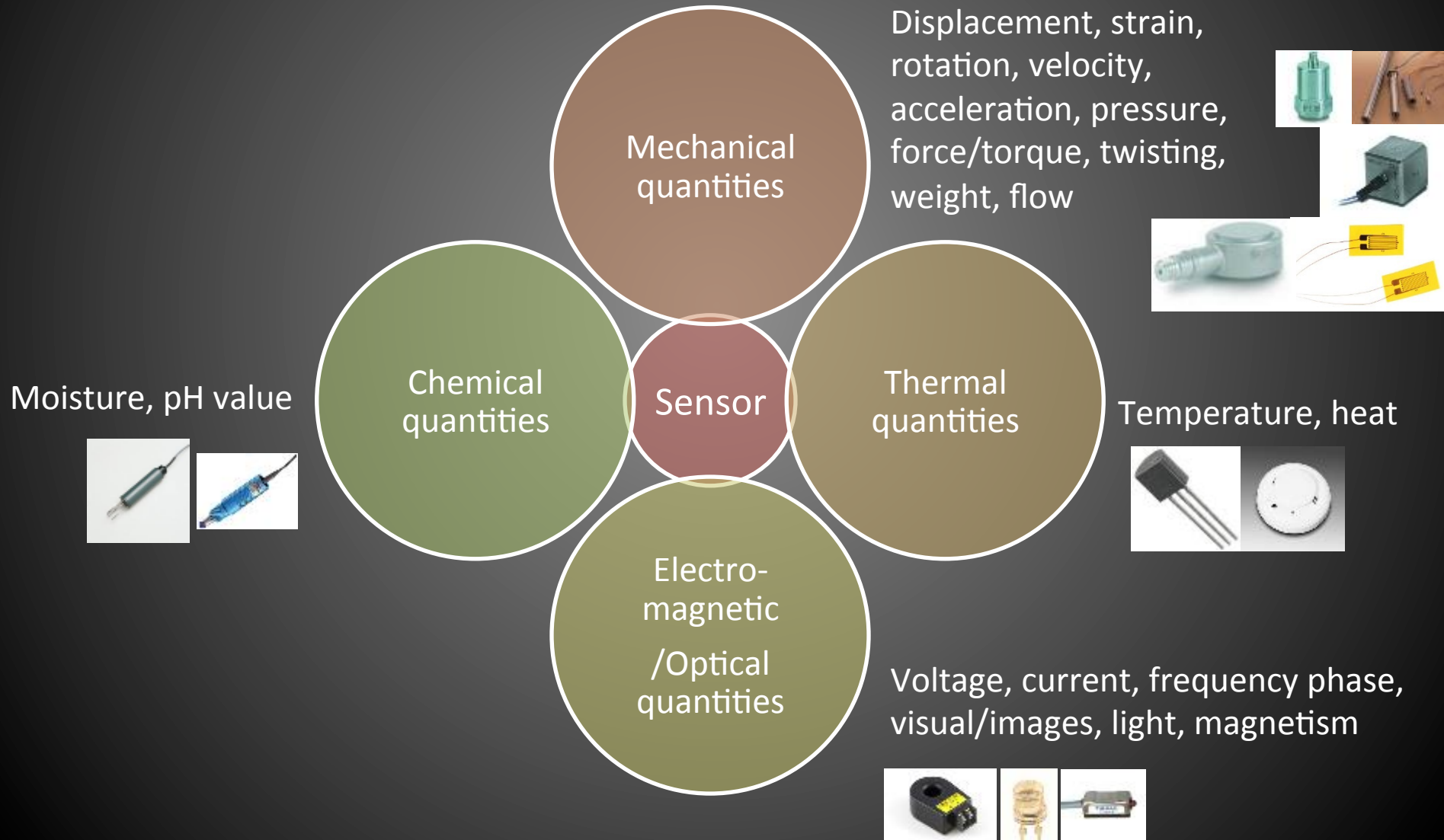
Rocker switch

Push Button Switch (Normally Open)



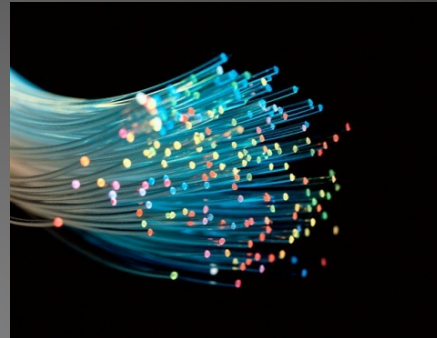
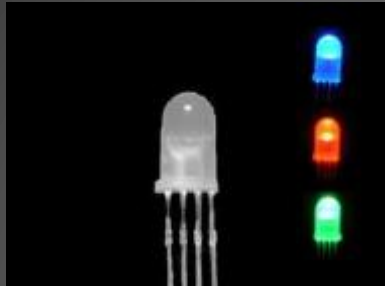
- **Sensor**

: A device for sensing a physical variable of a physical system or an environment



Output

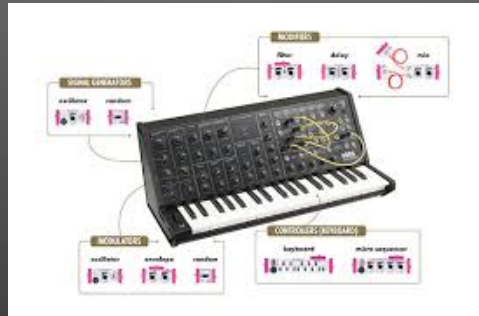
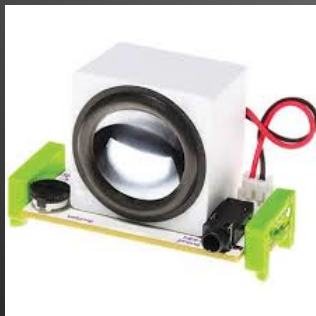
Light



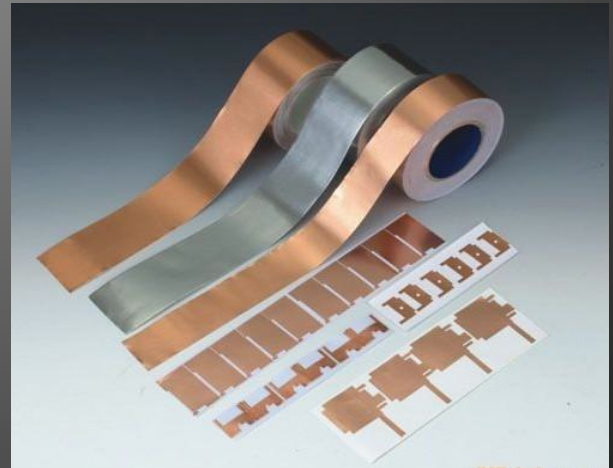
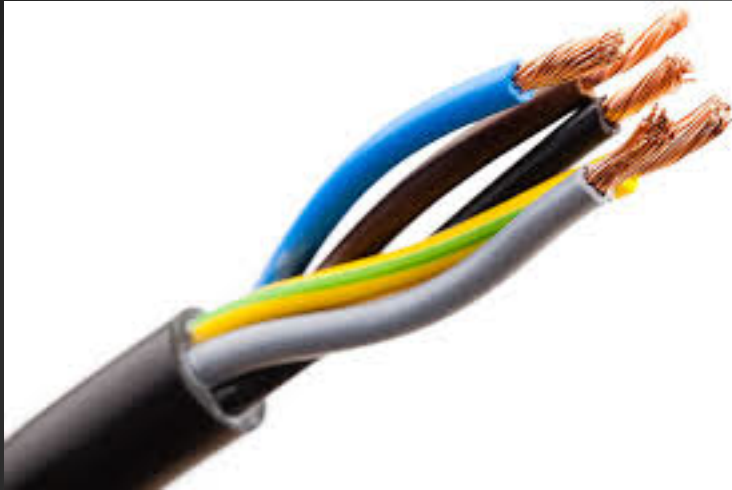
Movement



Audio



Wire



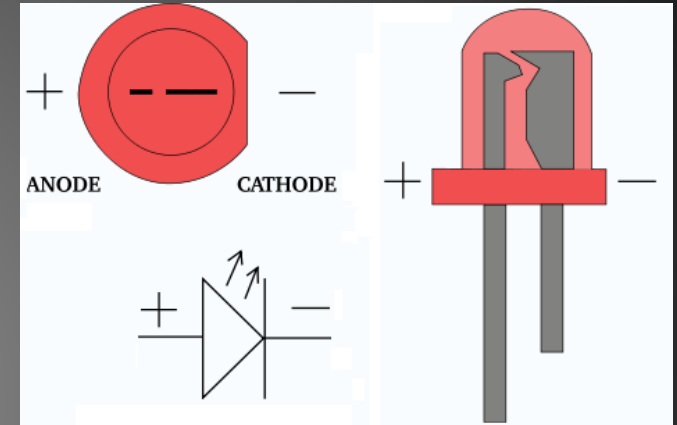
1. Light-Emitting Diode (LED)



http://www.societyofrobots.com/electronics_led_tutorial.shtml

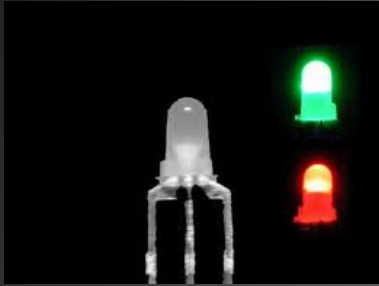


http://www.societyofrobots.com/electronics_led_tutorial.shtml

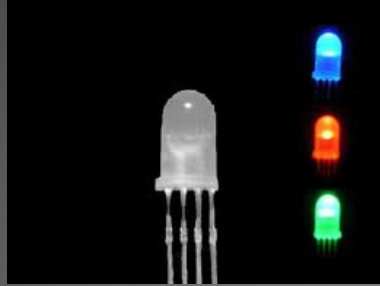


- Composition: Gallium (Ga) compounds
- Properties:
 - Small light source capable of emitting bright light
 - Consume very little power, do not emit much heat, and programmable
 - Come in a variety of colors, shapes, and sizes

Different types of LEDs



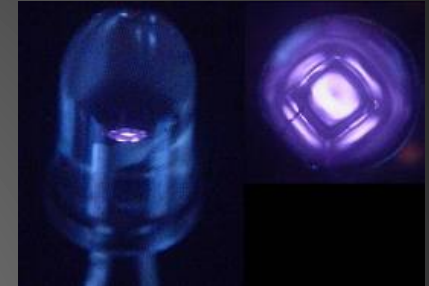
Bicolor LEDs



Tricolor LEDs



Blinking/Flashing LEDs



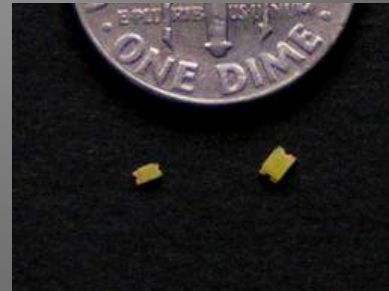
Infrared(IR) LEDs



Piranha/ High-Flux LEDs



Standard LEDs



Surface Mount Device (SMD) LEDs



Ultraviolet (UV) LEDs

LED wires



http://www.christmas-light-source.com/50-Blue-Wide-Angle-LED-White-Wire_p_2282.html



<http://mengengusainc.com/catalog/i37.html>



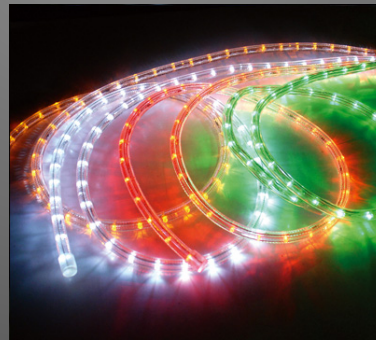
<http://www.oogalights.com/LED-Micro-Light-Strand-36-Inch-Silver-Wire-Battery-Operated-with-Timer-18-Cool-White-LED-Lights.aspx>



<http://www.made-in-china.com/showroom/hottywin/product-detailSbLQfpjYkx/China-LED-Wire-Cold-Light-Cable-LED-3-2B-.html>



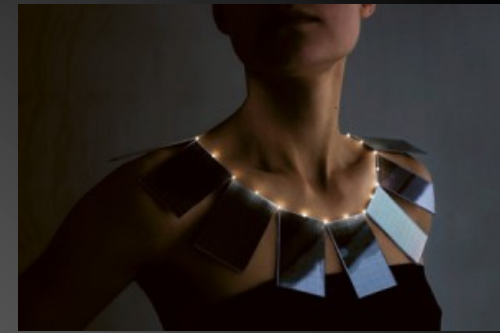
<http://www.deconeon-product.com/LED-2-Wire-Chasing-Rope-Light-187.html>



<http://www.birddogdistributing.com/>



Miya Masaoka's LED Kimono, 444 LEDs embroidered, respond to sound and movement
<http://www.ecouterre.com/part-dress-part-instrument-the-led-kimono-reacts-to-music-motion/>



<http://www.ledhut.co.uk/blog/bright-sparks-fashion-of-the-future>

Aurora Dress, 2009, 10,000 LEDs

<http://inhabitat.com/10000-leds-adorn-cutecircuit%E2%80%99s-dazzling-%E2%80%99Caurora%E2%80%99D-dress/>



<http://foolishgadgets.com/2009/07/page/11/>

<http://media.treehugger.com>



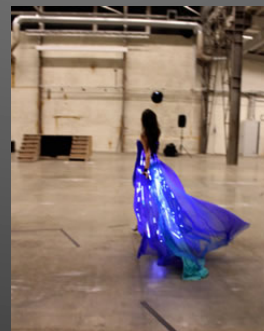
<http://www.ubergizmo.com/2009/09/breath-monitoring-dress/>



<http://foolishgadgets.com/2009/07/page/11/>



<http://worldstrangest.com>



http://www.gizmodiva.com/fashion/safuras_illuminated_dress_lights_up_the_stage.php





<http://mariakotsari.blogspot.com/2010/09/led-fashion.html>



<http://mariakotsari.blogspot.com/2010/09/led-fashion.html>



<http://mariakotsari.blogspot.com/2010/09/led-fashion.html>



<http://blog.annettehyder.com/2009/11/18/wed.aspx>



Hussein-Chalayan, Airborne, 15000 flickering LED lights
<http://trendsupdates.com/wearable-and-portable-architecture/>



<http://mariakotsari.blogspot.com/2010/09/led-fashion.html>



Hussein-Chalayan
<http://trendsupdates.com/wearable-and-portable-architecture/>



http://product.madeinchina.com/30pair-arrival-Fashion-LED-Earrings-light-up_13604169.shtml



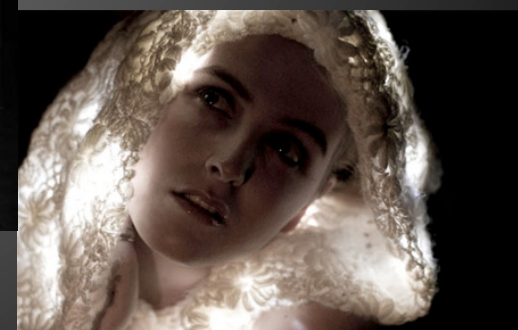
<http://www.dezeen.com/2007/11/28/ski-suits-with-solar-powered-lights-by-willy-bogner/>



<http://www.geekologie.com/2007/09/beat-dress-lights-up-to-music.php>



<http://www.techpin.com/how-would-you-look-like-in-a-led-decorated-coat/>



<http://www.ledhut.co.uk/blog/bright-sparks-fashion-of-the-future>



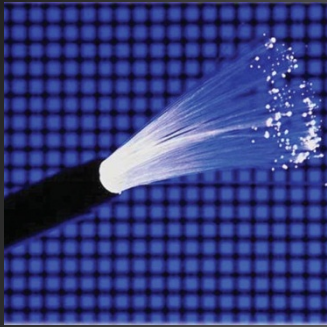
<https://vimeo.com/102654093>

Koo (2010)

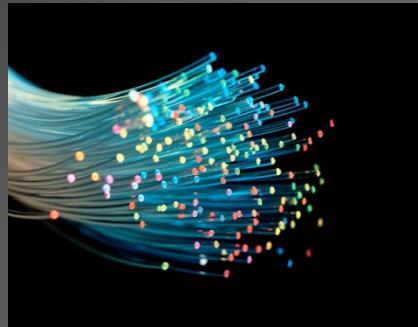


2. Fiber-Optics

: Strands of optically pure glass will carry digital information as pulses of light.

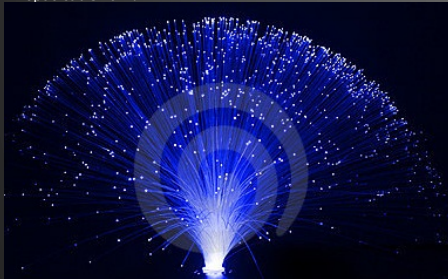
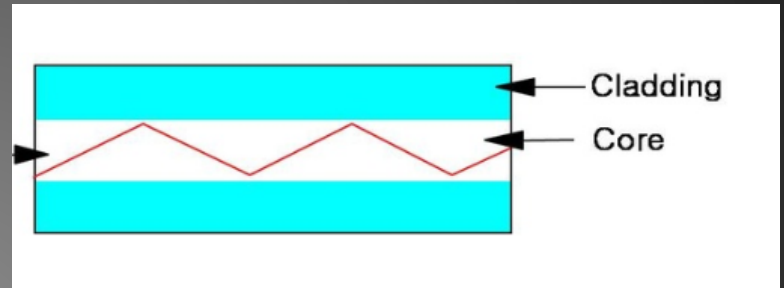


<http://commgear.com/index.php/cable-work-materials/cable-components-1/cable/fiber-optic-cable.html?>

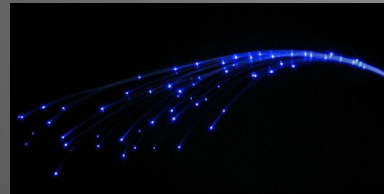


<http://bturn.com/4440/eastern-european-cities-have-the-fastest-internet/fiber-optic-tips>

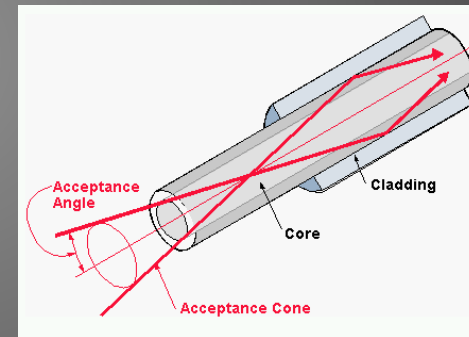
Light



<http://www.dreamstime.com/stock-image-fiber-optic-lamp-image10306391>



<http://followfrisby.wordpress.com/2011/06/09/fiber-optic-fashion-illuminate-your-soul/>



<http://images.yourdictionary.com/fiber-optics-glossary/>

- **Composition:** Thin plastic or glass cable capable of transmitting light
- **Properties:** Transmit light from one end of the cable to the other
Can be bundled together, flexible
- **Applications:** Woven, couching



<http://starsweare.com/2010/11/09/illuminated-clothing/>



<http://starsweare.com/2010/11/09/illuminated-clothing/>



<http://thecoolgadgets.com/enchanting-led-mini-dresses-hussein-chalayan/>



http://www.alibaba.com/product-gs/423963588/fiber_optics_costume.html



<http://followfrisby.wordpress.com/2011/06/09/fiber-optic-fashion-illuminate-your-soul/>



<http://charlottemcmanus.wordpress.com/2012/03/15/fashiontechnology-future-threadz-supersuper-vol-2-004/>



<http://gizmodo.com/5270184/we-ponder-the-social-bandwidth-of-this-fiber-optic-dress>

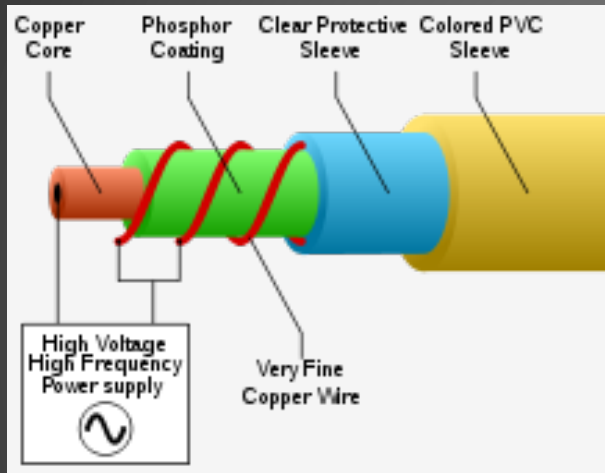


<http://followfrisby.wordpress.com/2011/06/09/fiber-optic-fashion-illuminate-your-soul/>

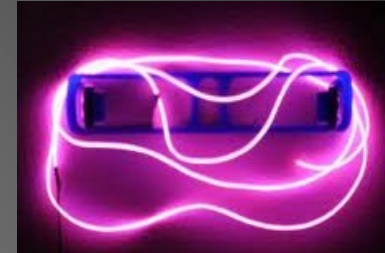
<http://www.youtube.com/watch?v=HVykymab7IM>

3. Electroluminescent Wire (EL)

: Made of thin copper wire coated in a phosphor that glows when alternating current is applied



<http://www.instructables.com/id/LED-lit-Tron-v20-suit/>



- Properties: A thin and flexible film or wire coated in phosphor
Can be bundled together, flexible
- Applications: Woven, couching, piping



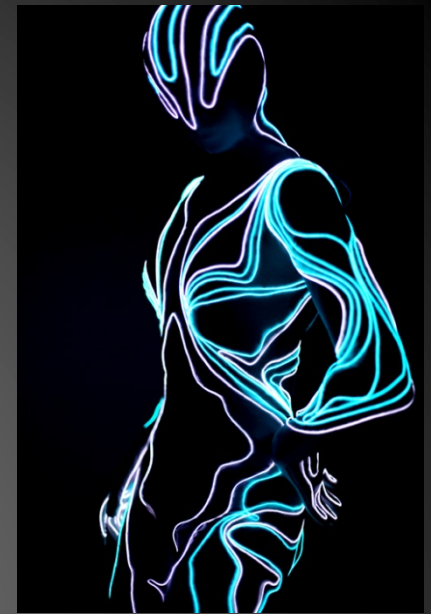
<http://www.ellightmade.com/el-design>



http://stdl.se/technology_textiles/?p=98



<http://www.made-in-china.com/showroom/elproducts88/product-detailqVmrEYOaMKU/>
China-Hot-EL-Flashing-Shoelaces-
HNR-004-.html



<http://myriadwhimsies.wordpress.com/tag/bodysuit/>



<http://intergalacticprimate.tumblr.com/post/29054342443/gentlegentry-geekyglamorous-tron-party>



<http://www.ellightmade.com/el-design>



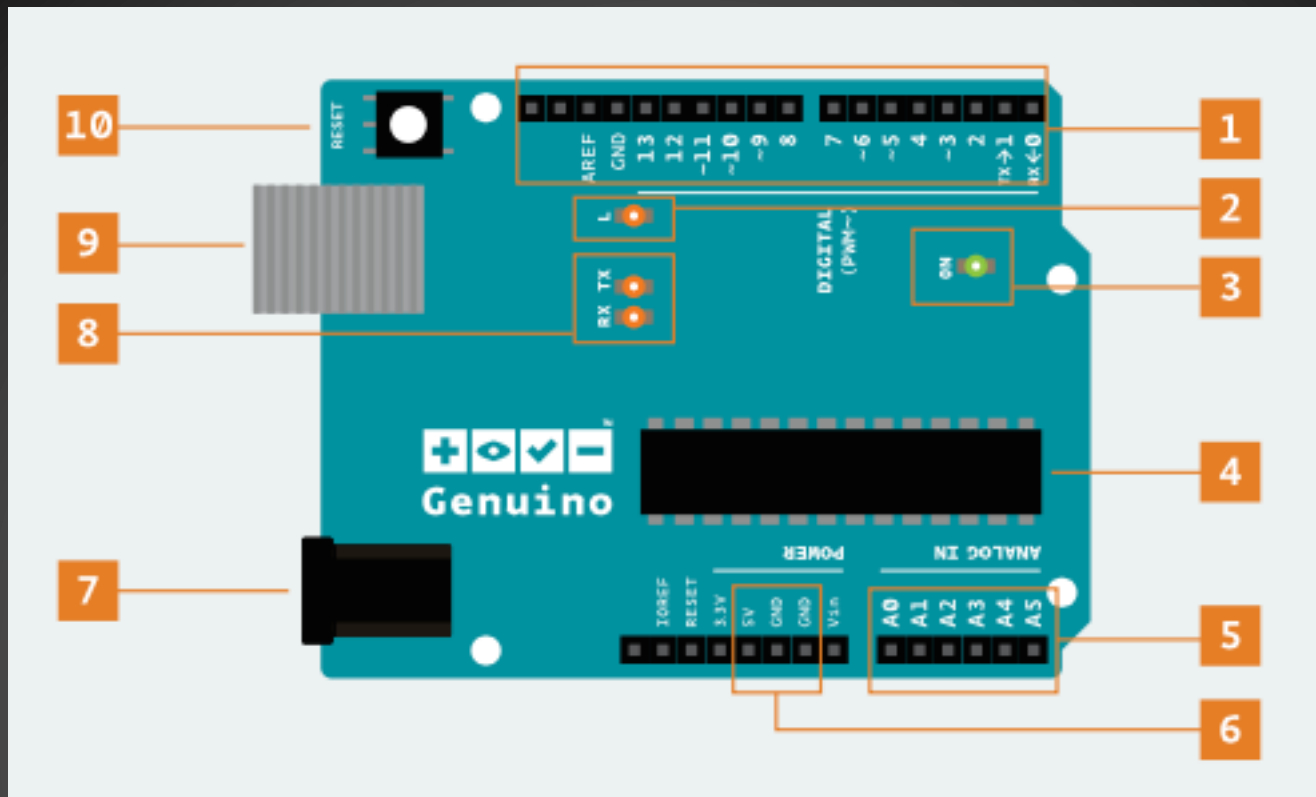
<http://www.youtube.com/watch?feature=endscreen&v=Rot9uaVO8s&NR=1>



<http://www.shockblast.net/neon-clothing/>



<http://www.instructables.com/id/LED-lit-Tron-v20-suit/>



1. **Digital pins:** use these pins with `digitalRead()` and `digitalWrite()`.
2. **Pin 13 LED:** built-in LED
3. **Power LED:** indicates the Arduino Board is receiving power
4. **ATmega microcontroller:** heart of the board
5. **Analog:** pins with `analogRead()`
6. **GND and 5V pins:** ground-negative side and 5V power
7. **Power connector:** plug into a USB for power, can accept voltages between 7-12V
8. **TX and RX LEDs:** indicate communication between the Arduino board and computer
9. **USB port:** powering the Arduino board and uploading the code
10. **Reset button:** reset the ATmega microcontroller

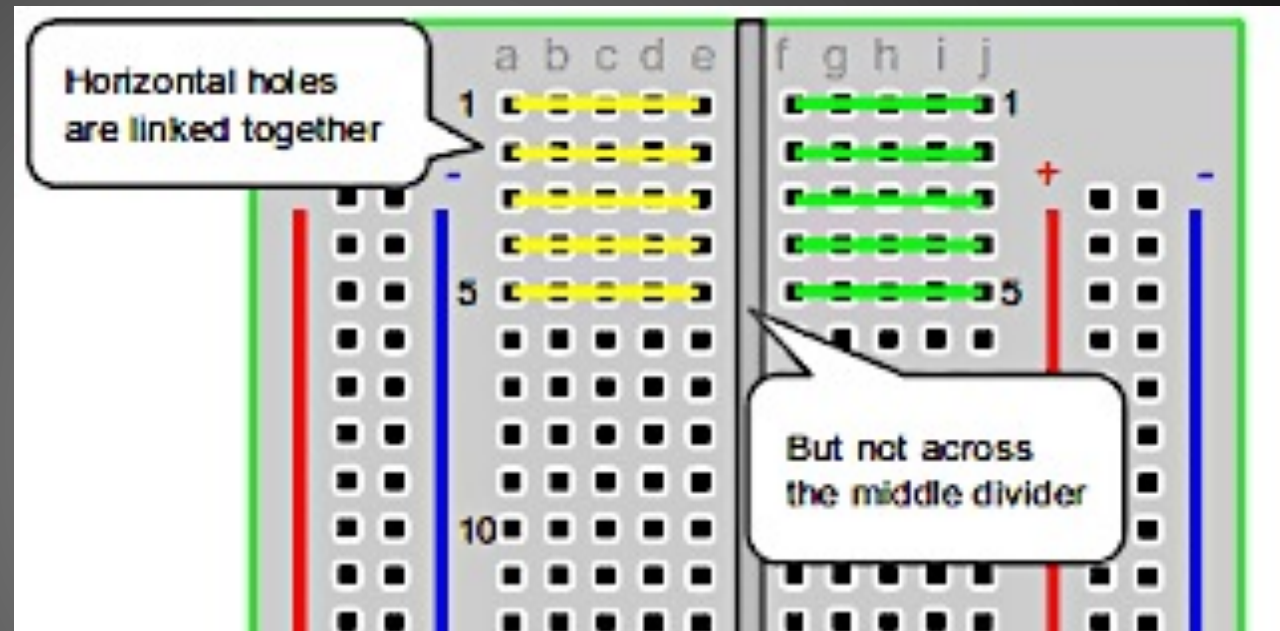
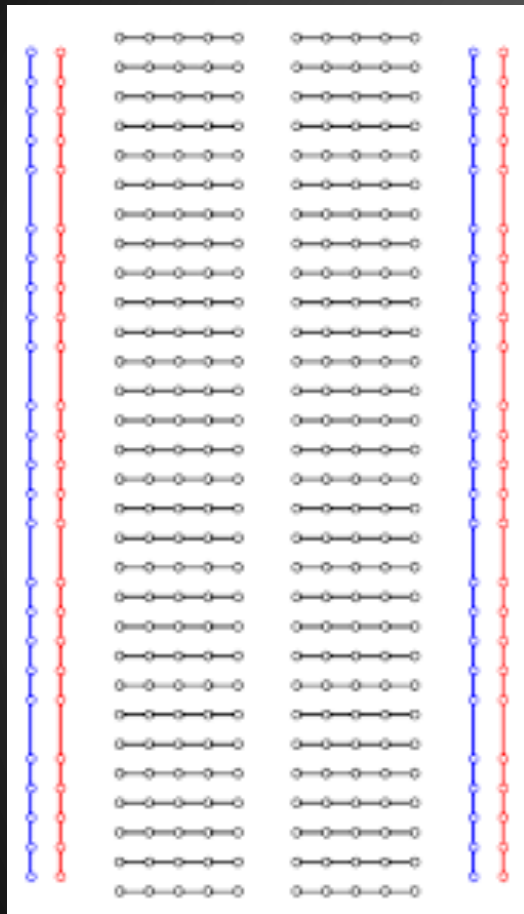
Arduino Language reference

<https://www.arduino.cc/en/Reference/HomePage>

Arduino code lighting tech.

<https://drive.google.com/a/ucdavis.edu/folderview?id=0BwzaBH8bg-dxR0MtaFQ4eEYzUjg&usp=sharing>

Breadboard



Design 1. Blink ver. 1: Basic

- Download the Arduino software-> Open the program
- File -> Examples -> 01. Basics -> Blink
- Connect Arduino and the laptop using a cable-> Verify-> Upload
- Check: Tools-> Port: /dev/cu.usbmodem1411 (Arduino/Gnuino Uno)
- LED on the pin 13 (Digital) will blink
- Change the number: delay (1000)
-> Verify -> Upload
(e.g., 500, 100, 5000)



```
int led = 13;
void setup() {
    pinMode(led, OUTPUT);
}
void loop() {
    digitalWrite(led, HIGH);
    delay(1000);
    digitalWrite(led, LOW);
    delay(1000);
}
```


<div> <div> <div>0 1 2 3 4 5 6 7 8 9</div> <div> <div>0 Black</div> <div>1 Brown</div> <div>2 Red</div> <div>3 Orange</div> <div>4 Yellow</div> <div>5 Green</div> <div>6 Blue</div> <div>7 Purple</div> <div>8 Grey</div> <div>9 White</div> </div> </div> <div> <div>±1% Brown</div> <div>±2% Red</div> <div>±5% Gold</div> <div>±10% Silver</div> </div> </div> <div>Color Codes</div>	<div> <div>±1%</div> <div>±2%</div> <div>±5%</div> <div>±10%</div> </div> <div> <div>27K</div> <div>EXAMPLE</div> </div> <div> <div>0 X1</div> <div>1 1 X10</div> <div>2 2 X100</div> <div>3 3 X1000</div> <div>4 4 X10000</div> <div>5 5 X100000</div> <div>6 6 X1000000</div> <div>7 7 ÷10</div> <div>8 8 ÷100</div> <div>9 9</div> </div> <div>4 Band Resistors</div>	<div> <div>±1%</div> <div>±2%</div> <div>±5%</div> <div>±10%</div> </div> <div> <div>15K</div> <div>EXAMPLE</div> </div> <div> <div>0 0 X1</div> <div>1 1 1 X10</div> <div>2 2 2 X100</div> <div>3 3 3 X1000</div> <div>4 4 4 X10000</div> <div>5 5 5 ÷10</div> <div>6 6 6 ÷100</div> <div>7 7 7</div> <div>8 8 8</div> <div>9 9 9</div> </div> <div>5 Band Resistors</div>	<div> <div>±1%</div> <div>±2%</div> <div>±5%</div> <div>±10%</div> </div> <div> <div>100</div> <div>25</div> <div>10</div> <div>1</div> <div>50</div> <div>15</div> <div>5</div> </div> <div> <div>620K</div> <div>EXAMPLE</div> </div> <div> <div>0 0 X1</div> <div>1 1 1 X10</div> <div>2 2 2 X100</div> <div>3 3 3 X1000</div> <div>4 4 4 X10000</div> <div>5 5 5 ÷10</div> <div>6 6 6 ÷100</div> <div>7 7 7</div> <div>8 8 8</div> <div>9 9 9</div> </div> <div>6 Band Resistors</div>
---	--	---	--

- Resistor calculator

<http://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band>

<http://ledcalc.com>

LED MIXED PACK

 sparkfun.com

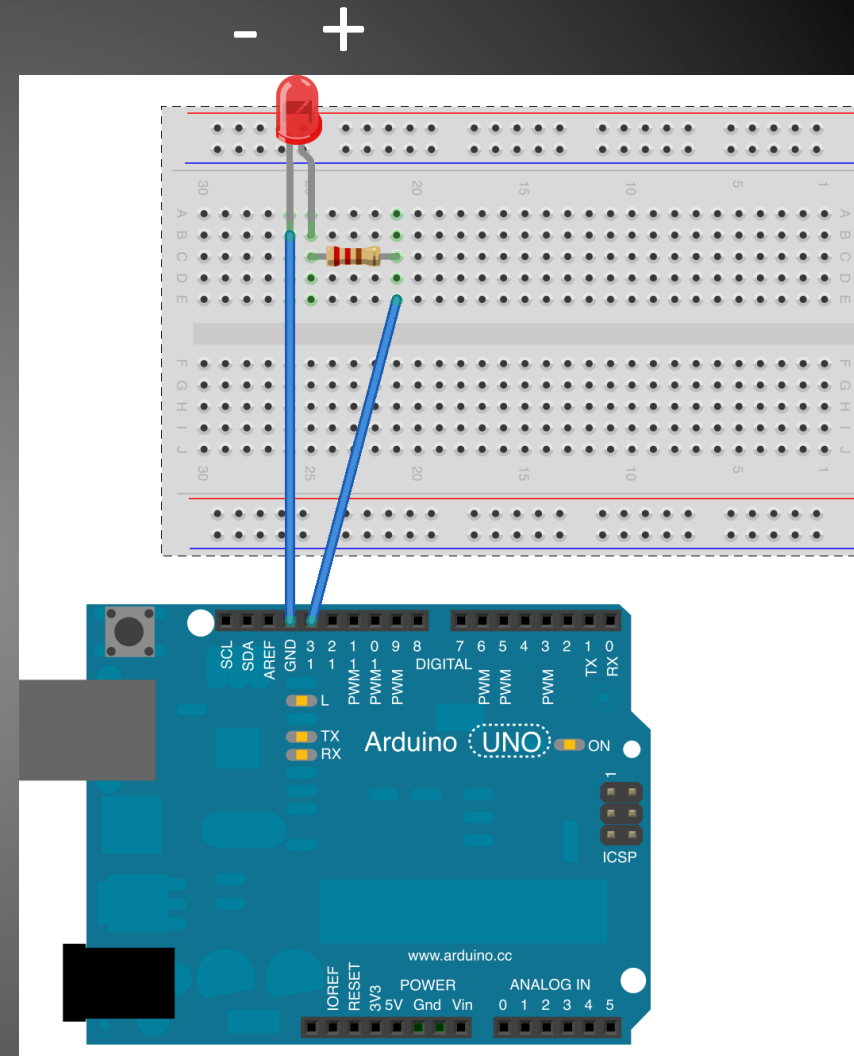
COLOR	BRIGHTNESS	WAVELENGTH	VDROP
BASIC RED	150 - 200mcd	620 - 625nm	2.0 - 2.4 V
BASIC YELLOW	150 - 200mcd	587 - 591nm	2.0 - 2.4 V
BASIC BLUE	300 - 400mcd	465 - 467.5nm	3.0 - 3.4 V
BASIC GREEN	150 - 200mcd	570 - 575nm	2.0 - 2.4 V

* 3mm 5mm LED: 20mA

<https://www.sparkfun.com/products/12062>

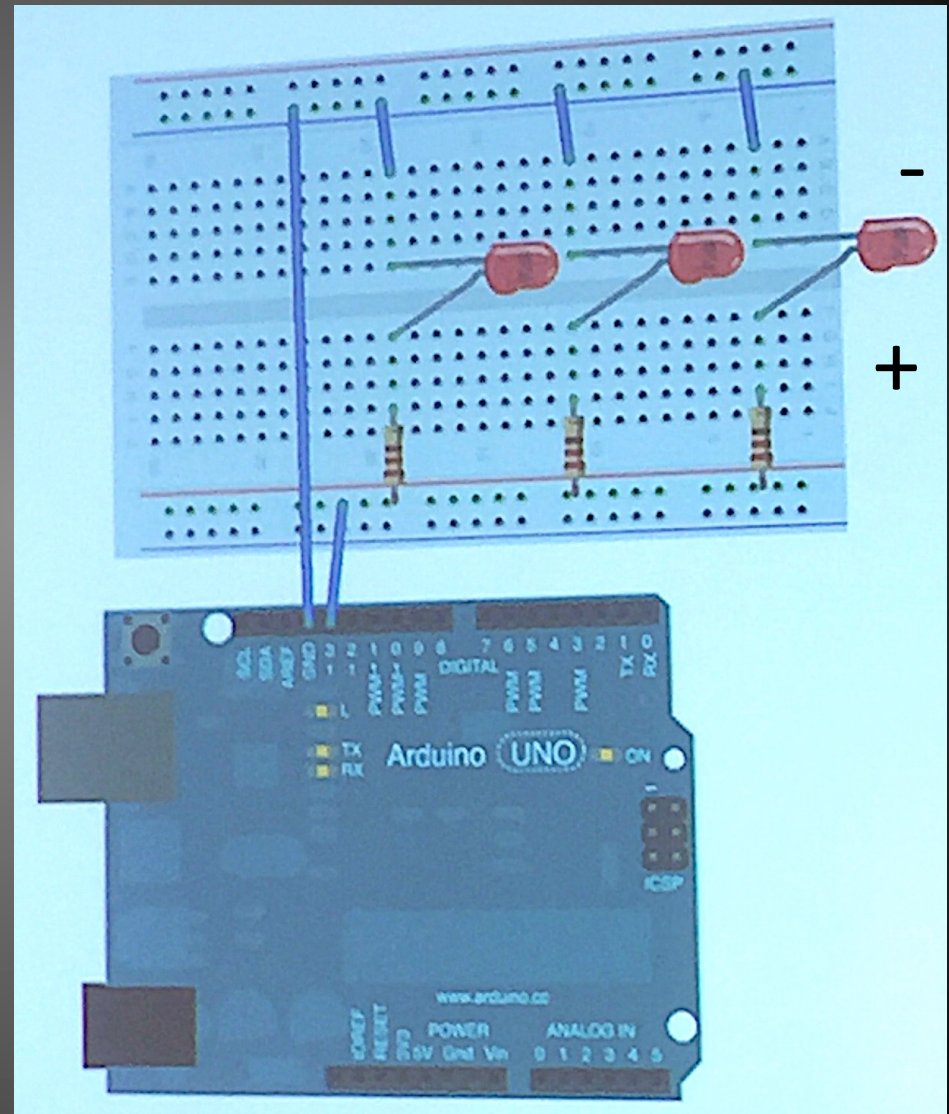
Design 1. Blink ver. 2: One LED

- Set back to delay (1000)→ Verify-> Upload
- Use a LED and 330 or 220 ohm resistor (orange, orange, brown, gold)
- Connect LED + to Pin 13 (Digital) and connect LED – to Ground (Digital)



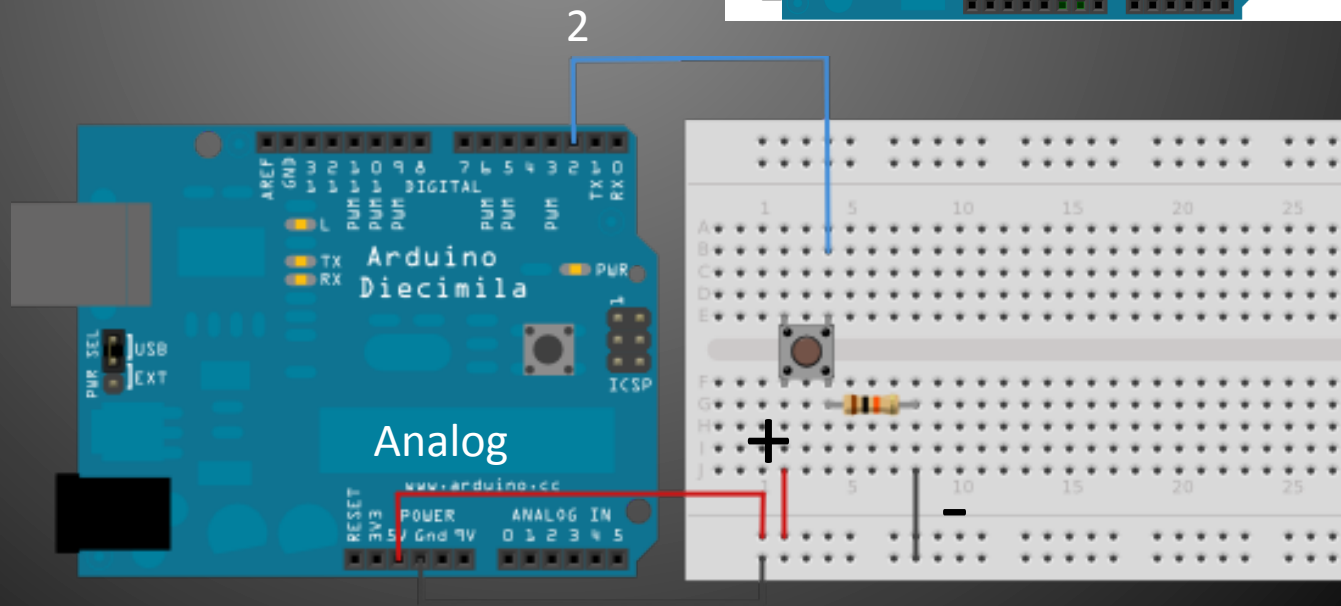
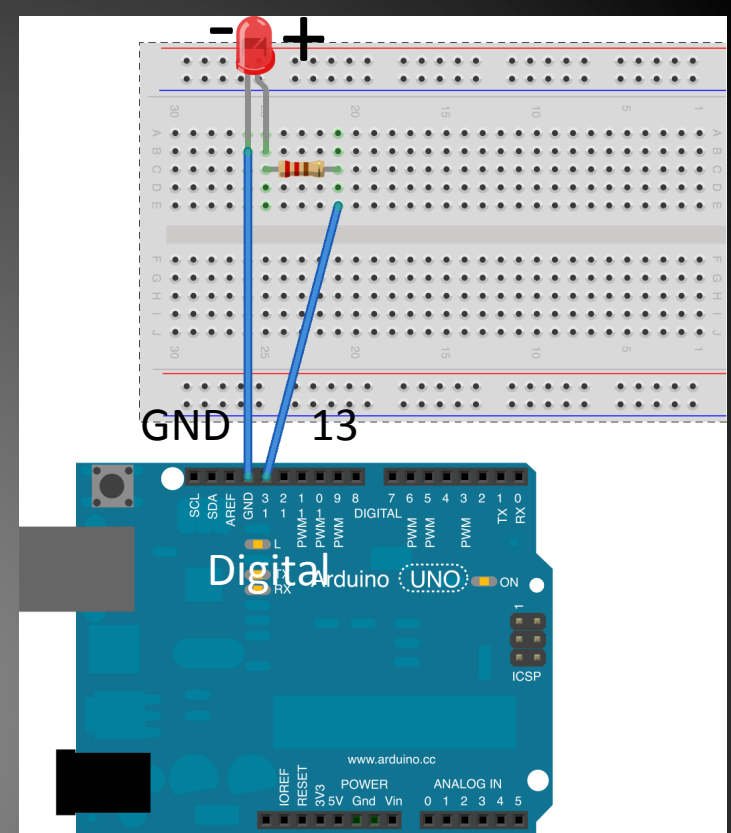
Design 1. Blink ver. 3: Parallel circuit

- Set back to delay (1000)→ Verify-> Upload
- Use a LED and three 330 or 220 ohm resistor (orange, orange, brown, gold)
- Connect LED + to Pin 13 (Digital) and connect LED – to Ground (Digital)
- Connect 9V battery



Design 2. LED & Button

- File-> Examples-> 02. Digital-> Button
- Use a LED and a 330 or 220 ohm resistor. Connect LED + to Pin 13 (Digital) and connect LED - to Ground (Digital)
- Use a push button and a 10K ohm resistor (brown, black, orange, gold)




```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;  // the number of the pushbutton pin
const int ledPin = 13;    // the number of the LED pin

// variables will change:
int buttonState = 0;      // variable for reading the pushbutton status

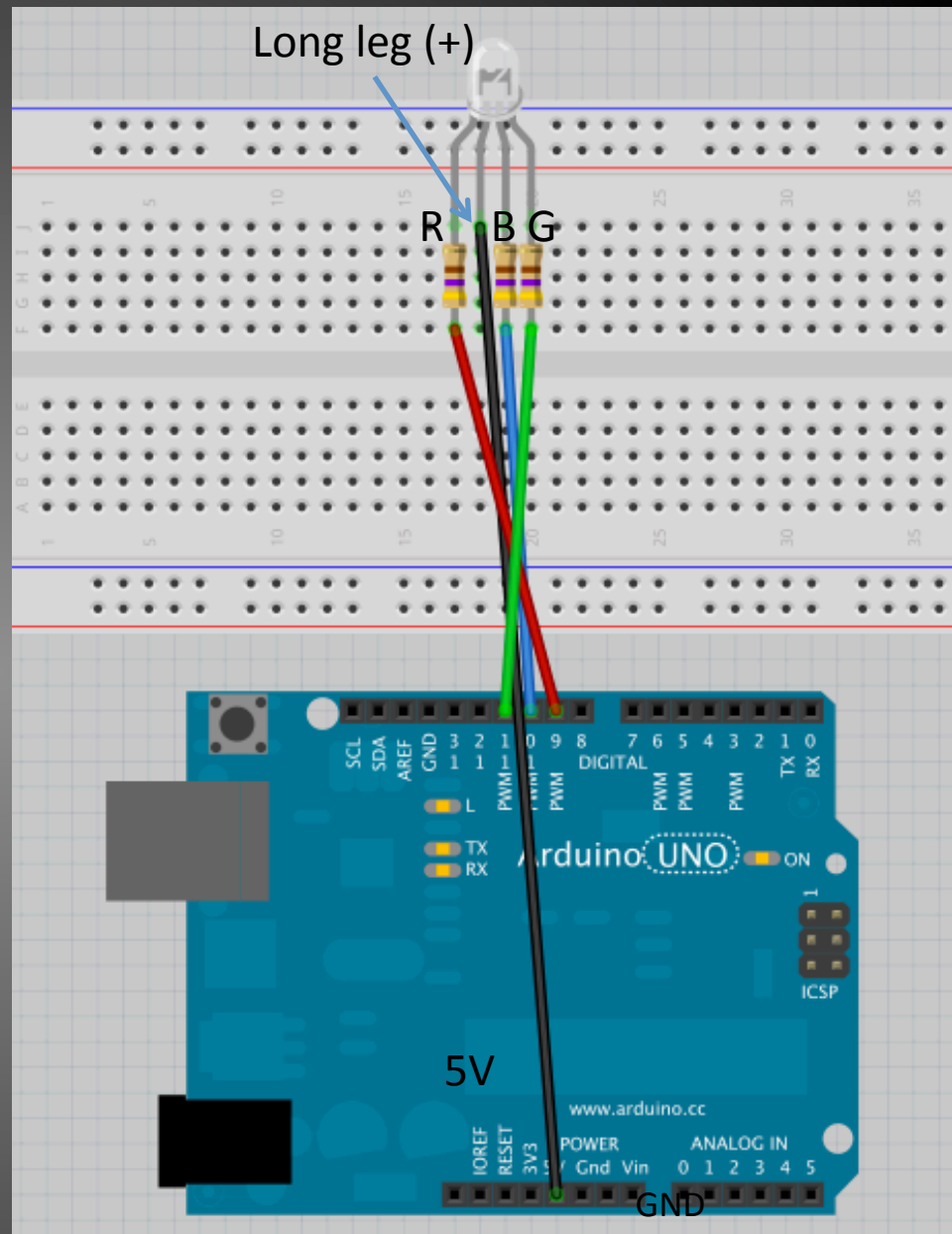
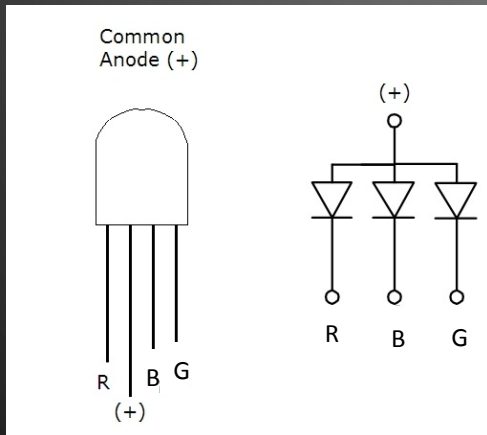
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Design 2. RGB LED

- Open the Arduino code "RGB LED"
- Use a photocell and a 470 ohm resistor (yellow, violet, brown, gold)
- Adjust the numbers to change the color (e.g, R-100, G-250, B-250)

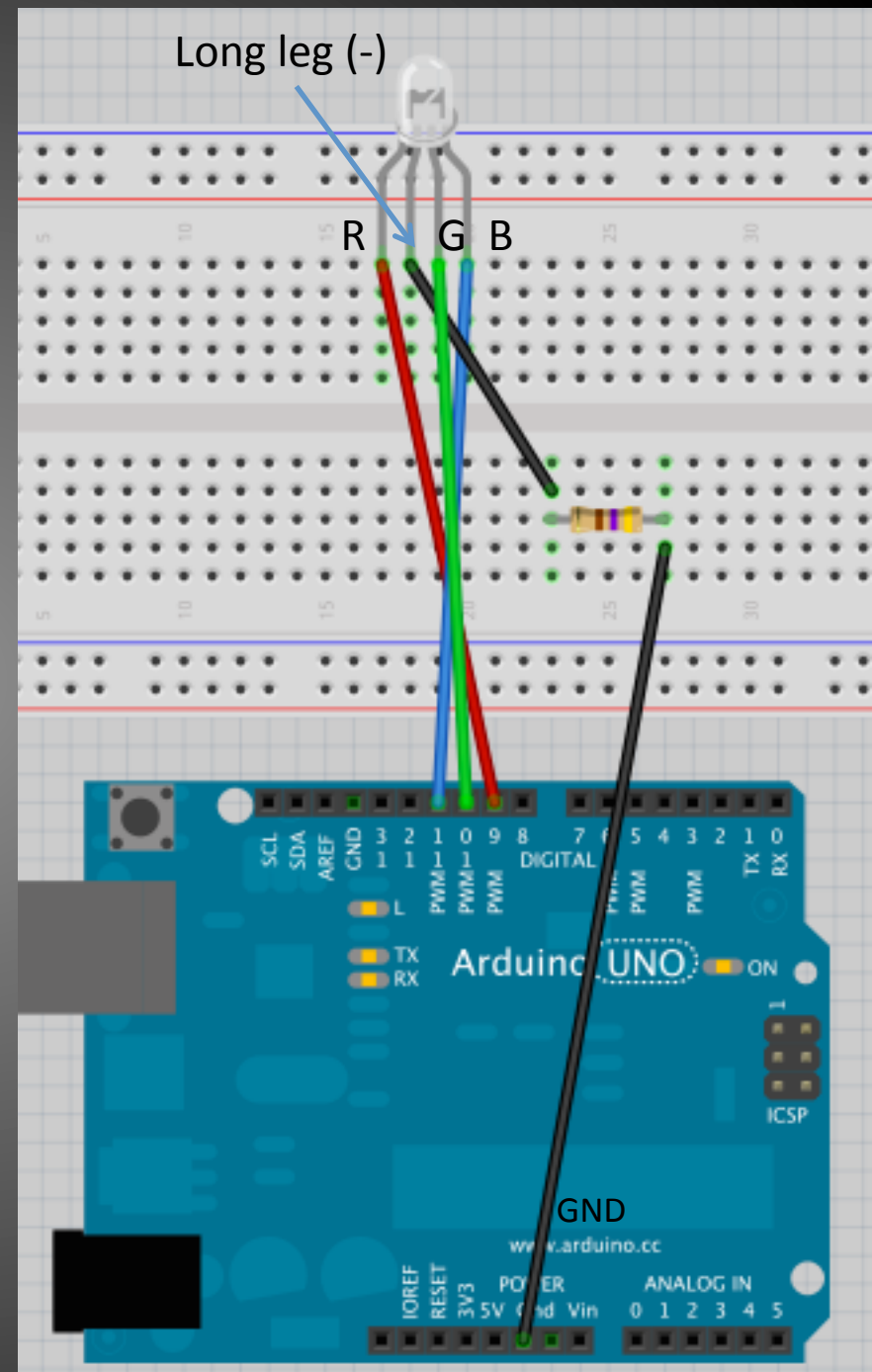
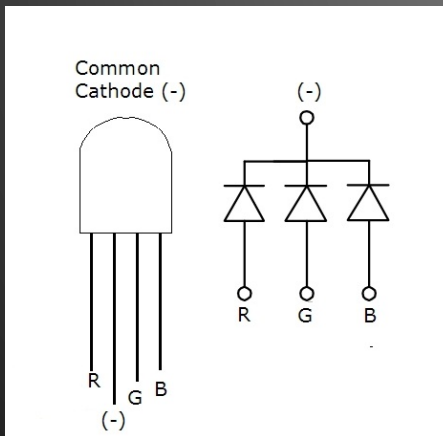


```
int r=9;  
int b=10;  
int g=11;
```

```
void setup() {  
  pinMode(r, OUTPUT);  
  pinMode(b, OUTPUT);  
  pinMode(g, OUTPUT);  
  // put your setup code here, to run once:  
  
}
```

```
void loop() {  
  analogWrite(r,30);  
  analogWrite(b, 30);  
  analogWrite(g, 50);  
  // put your main code here, to run repeatedly:  
  
}
```

- Open the Arduino code “RGB LED”
- Use a photocell and a 470 ohm resistor (yellow, violet, brown, gold)
- Adjust the numbers to change the color (e.g, R-100, G-250, B-250)



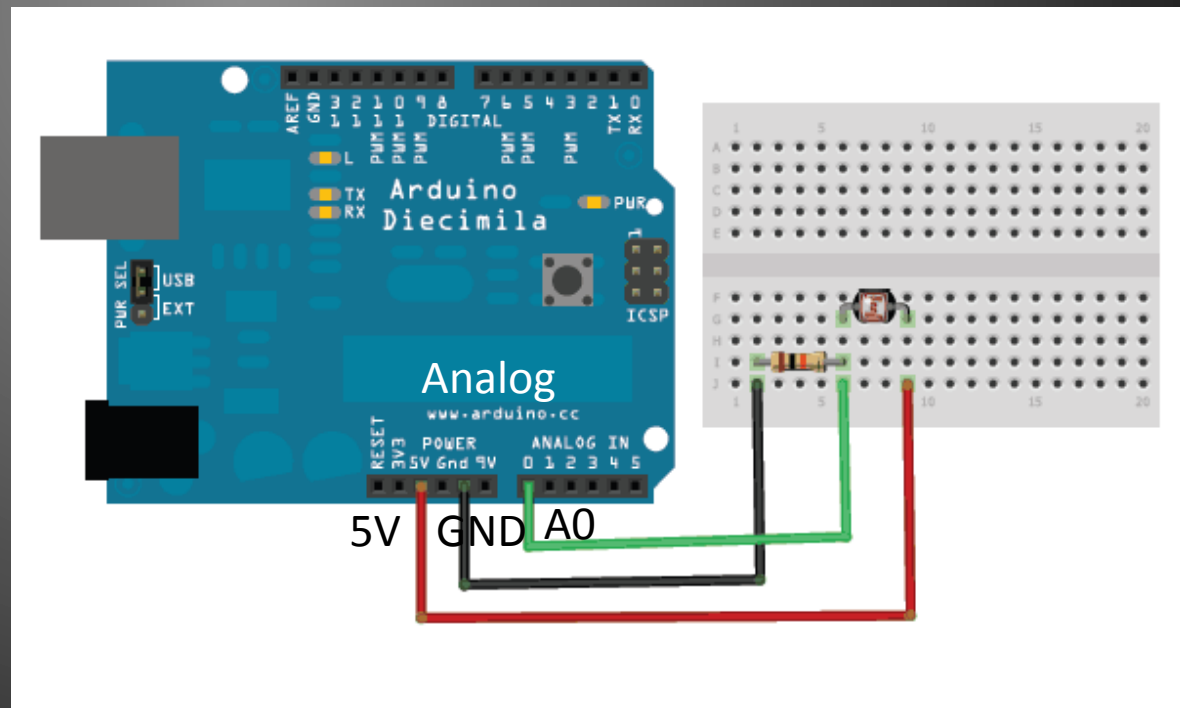
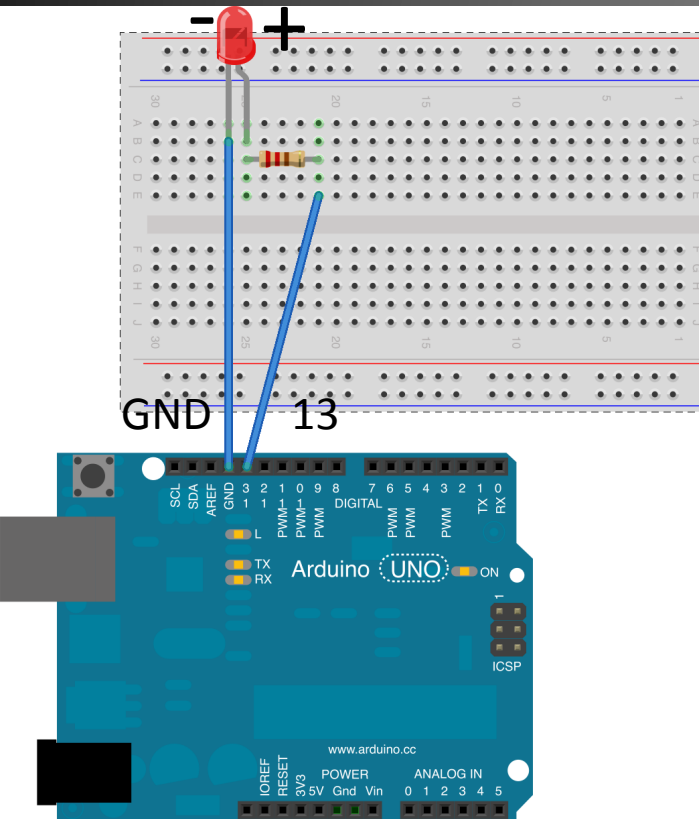
```
int r=9;  
int g=10;  
int b=11;
```

```
void setup() {  
  pinMode(r, OUTPUT);  
  pinMode(g, OUTPUT);  
  pinMode(b, OUTPUT);  
  // put your setup code here, to run once:  
  
}
```

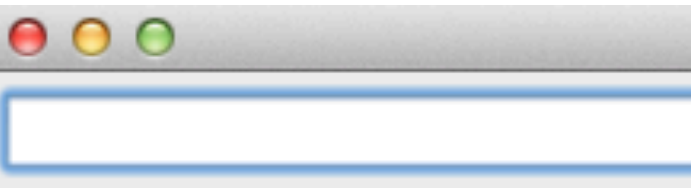
```
void loop() {  
  analogWrite(r,30);  
  analogWrite(g, 30);  
  analogWrite(b, 50);  
  // put your main code here, to run repeatedly:  
  
}
```


Design 3. Photocell switch-LED

- Smartsite-> Announcement-> Adruino code-lighting tech.-> download and open “Photocell-check”
- Use a photocell and a 10K ohm resistor (brown, black, orange, gold)
- Verify-> Upload-> open “Serial Monitor” by clicking the top right icon



```
Analog = 578 - Bright
Analog reading = 588 - Bright
Analog reading = 588 - Bright
Analog reading = 588 - Bright
Analog reading = 261 - Light
Analog reading = 272 - Light
Analog reading = 253 - Light
Analog reading = 216 - Light
Analog reading = 219 - Light
Analog reading = 192 - Dim
Analog reading = 215 - Light
Analog reading = 220 - Light
Analog reading = 188 - Dim
Analog reading = 192 - Dim
Analog reading = 192 - Dim
Analog reading = 584 - Bright
Analog reading = 602 - Bright
Analog reading = 602 - Bright
Analog reading = 602 - Bright
Analog reading = 508 - Bright
```



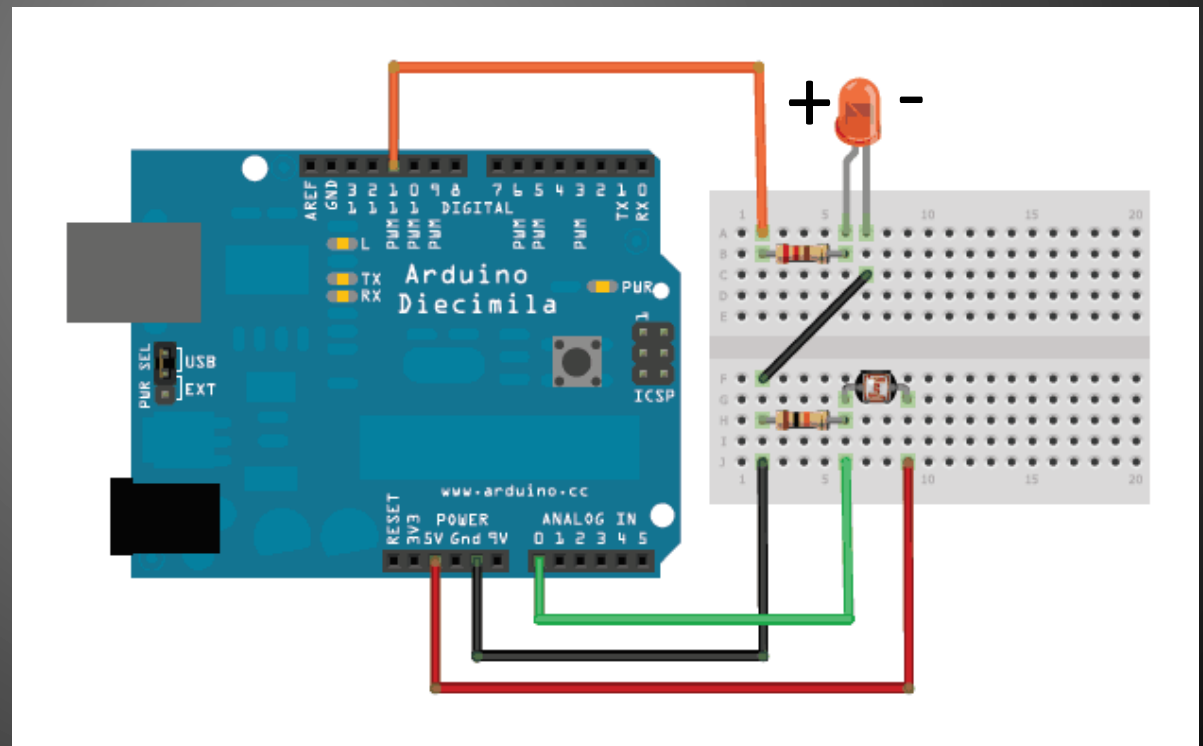
```
Analog = 578 - Bright
Analog reading = 588 - Bright
Analog reading = 588 - Bright
Analog reading = 588 - Bright
Analog reading = 261 - Light
Analog reading = 272 - Light
Analog reading = 253 - Light
Analog reading = 216 - Light
Analog reading = 219 - Light
Analog reading = 192 - Dim
Analog reading = 215 - Light
Analog reading = 220 - Light
Analog reading = 188 - Dim
Analog reading = 192 - Dim
Analog reading = 192 - Dim
Analog reading = 584 - Bright
Analog reading = 602 - Bright
Analog reading = 602 - Bright
Analog reading = 602 - Bright
Analog reading = 508 - Bright
```

- Open the Arduino code “Photocell-switch.”
- Use a photocell and a 330 ohm resistor (orange, orange, brown, gold).
- Change the number < 600 to smaller number than “Bright” state bigger than dim state (e.g., 500).
- Cover the photocell then the LED will light up.

```

Analog reading = 578 - Bright
Analog reading = 588 - Bright
Analog reading = 588 - Bright
Analog reading = 588 - Bright
Analog reading = 261 - Light
Analog reading = 272 - Light
Analog reading = 253 - Light
Analog reading = 216 - Light
Analog reading = 219 - Light
Analog reading = 192 - Dim
Analog reading = 215 - Light
Analog reading = 220 - Light
Analog reading = 188 - Dim
Analog reading = 192 - Dim
Analog reading = 192 - Dim
Analog reading = 584 - Bright
Analog reading = 602 - Bright
Analog reading = 602 - Bright
Analog reading = 602 - Bright
Analog reading = 508 - Bright

```



```
int sensorPin = 0; // select the input pin for the photocell
int ledPin = 13;   // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the sensor
```

```
void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.print("Digital reading = ");
  Serial.println(sensorValue); // the raw analog reading
  // turn the ledPin on
  if (sensorValue < 700){
    digitalWrite(ledPin, HIGH);
  }else{
    digitalWrite(ledPin,LOW);
  }
  // stop the program for <sensorValue> milliseconds:
  delay(10);
}
```




Kinetic Technology

http://www.intel.com/press_kit/docs/technology/kinetic_headline.htm

I. Motor

: An electromechanical device that converts electrical energy into mechanical energy

1) Servo



<http://www.seattlerobotics.org/guide/servos.html>

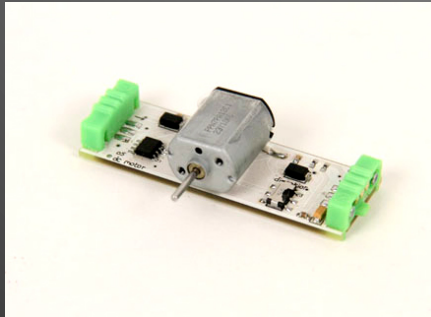


<http://www.seattlerobotics.org/guide/servos.html> 15



<http://www.ladyada.net/make/mshield/>

2) DC motor



<https://community.littlebits.cc/bits>



http://www.optms.net/China_DC_motor.html

VIDEO-TRANSFORMABLE DRESS-CHALAYAN:

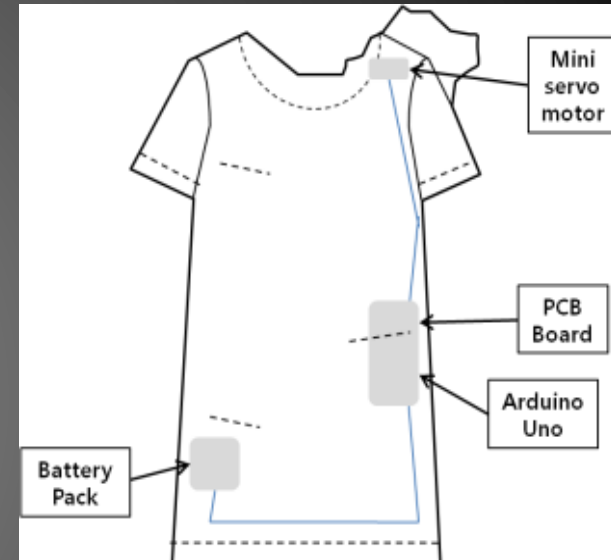
<https://www.youtube.com/watch?v=XEGWarCzerY>

<https://www.youtube.com/watch?v=P6SPJTKMbjk>
(from 2 min 28 sec)



2. Push the button

1. Slide the button
(off→on)



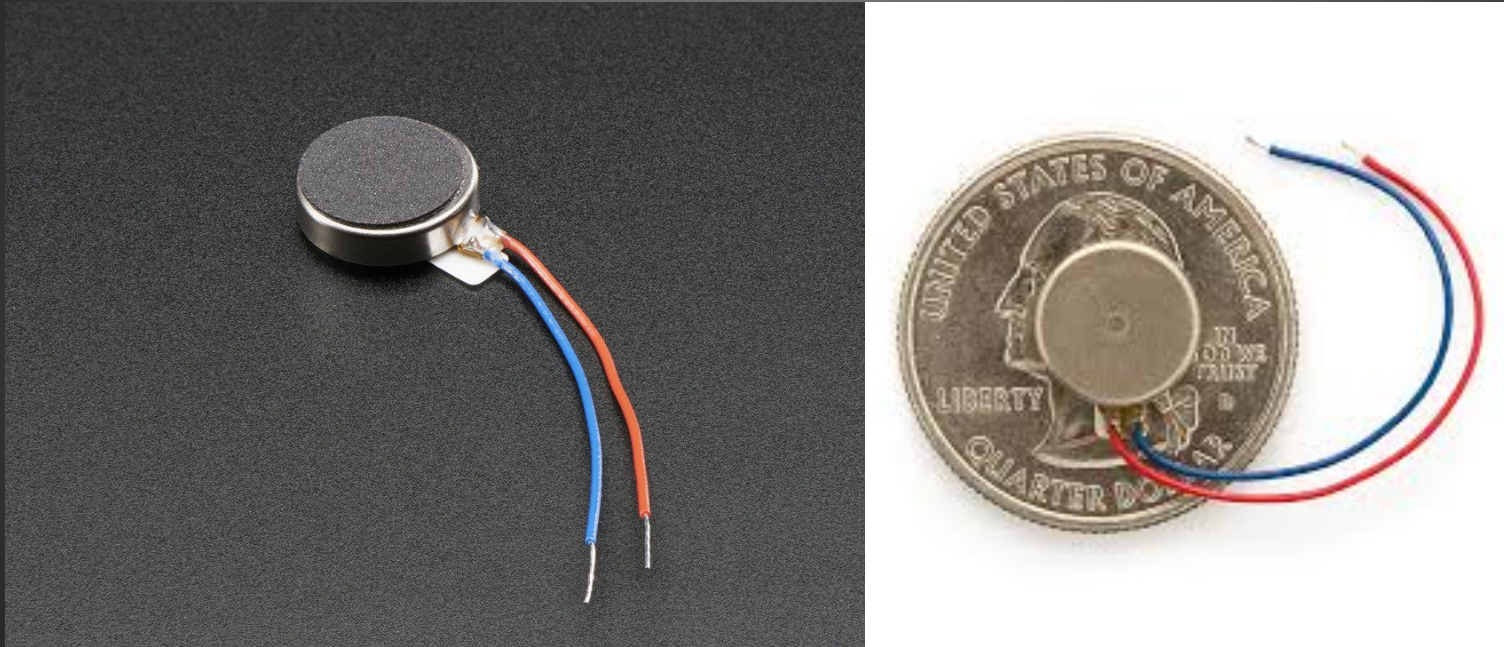
Fairy Tale Kinetic Dress (Koo, 2012)

: Preschool teacher's dress, interactive dress,
Arduino, Servo motor, LEDs



3) Vibration motor

: A mechanical device that generate vibrations



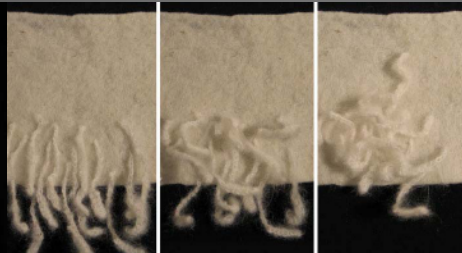
Two wires are used to control/power the vibe. Simply provide power from a battery or microcontroller pin (red is positive, blue is negative)

VIDEO-"TellMe"-DC motor & vibration motor

: <https://vimeo.com/105722771>

Shape Memory Alloy

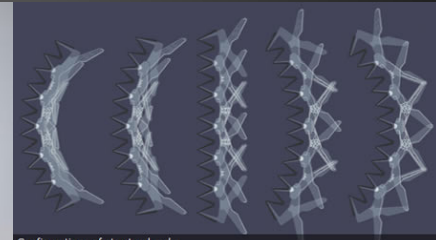
: An alloy that remembers its original shape
(e.g., robot, medical devices, aerospace)



Kinetic electronic garments
(Berzowska & Coelho, 2005)



<http://www.notcot.org/page/114/#5279>



Configurations of structural racks

<http://www.interactivearchitecture.org/category/audio/page/2>



<http://www.kobakant.at/DIY/?p=3396>

VIDEOS-SMA:

<http://www.youtube.com/watch?v=HdRRy7hltgl>

http://www.youtube.com/watch?v=Hp5Q_Dadz6Y&NR=1&feature=endscreen

<http://www.youtube.com/watch?v=EikQOrLyc-A&feature=relmfu>

http://www.youtube.com/watch?v=4JQl8_D-370&feature=endscreen&NR=1

<http://www.youtube.com/watch?v=mVbcVMwl44M>

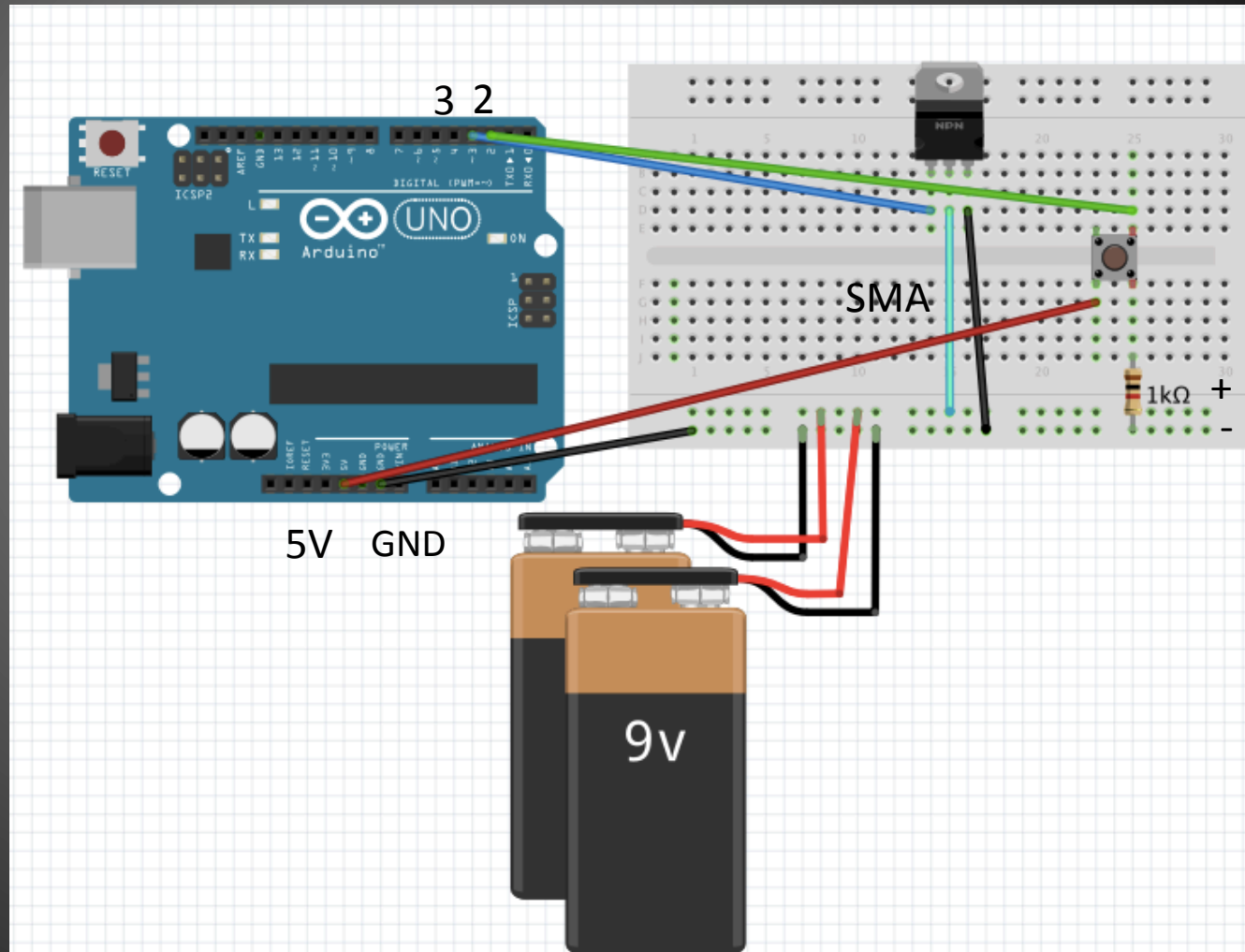
FT Lab

<https://vimeo.com/131447146>

https://drive.google.com/a/ucdavis.edu/file/d/0B8_RZC9fwUTjU0JpRFBIY2NIYUE/view

Design 1: Shape Memory Alloy (SMA)

1. Connect the buttons with 1K ohm resistor. Use code [NitinolActivationCode](#).
2. Cyan colored wire is the SMA wire. Use alligator wires.
3. Use alligator wires to connect the 9V batteries as parallel.
4. Press the button and check the SMA movement.



```
int button =2;
int val=0;
int gate=3;

// the setup routine runs once when you press reset:
void setup() {
  //Assign pin modes
  pinMode(button, INPUT);
  pinMode(gate,OUTPUT);
}

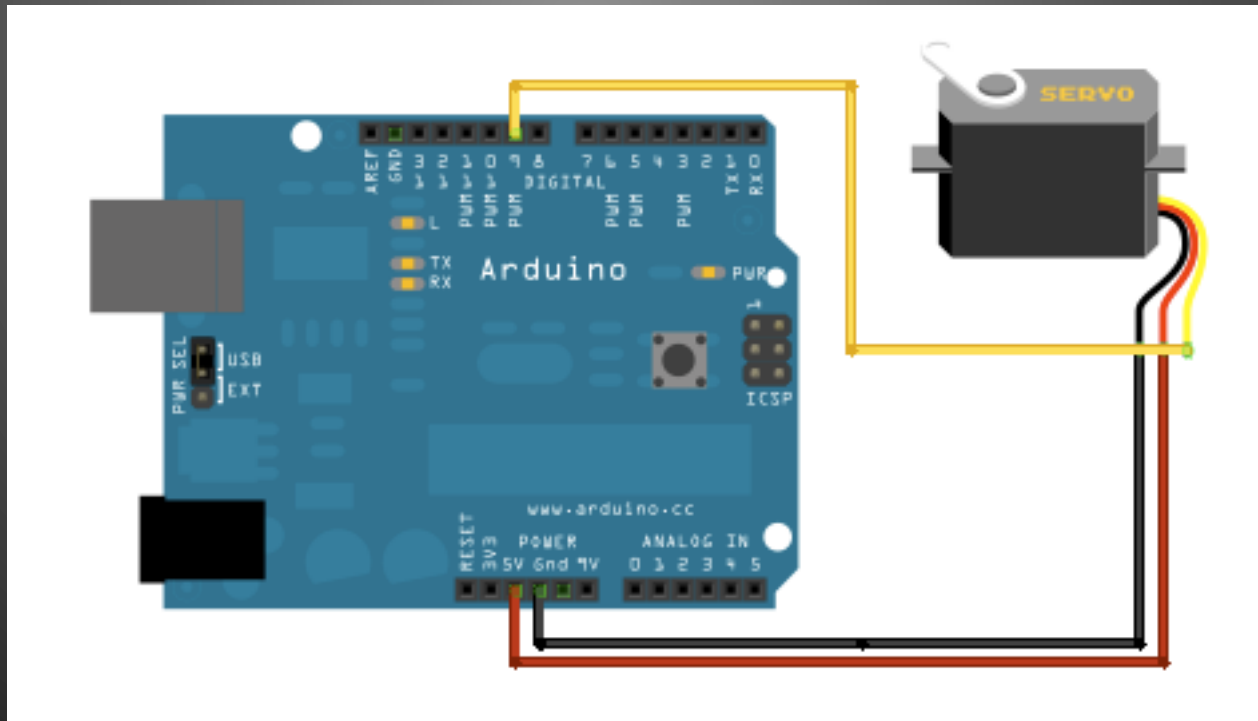
// the loop routine runs over and over again forever:
void loop() {
  val=digitalRead(button); //Read input from switch and store to val
  if(val==HIGH){           //If switch is pressed
    analogWrite(gate,253); //Start outputting PWM signal from pin 3 at 99% Duty Cycle
  }
  if(val==LOW){            //If switch is released
    analogWrite(gate,0);   //Stop outputting PWM signal
  }
}

/* PWM Compare is set to 253 to prevent sending a DC signal and directly
 * shorting the battery, preserving battery life and preventing fires */
```

Design 2: Simple Servo Motor

1. Connect the servo motor directly to the Arduino board
2. Open the code from the library: File-> Examples-> Servo-> **Sweep**
3. Verify and download the code to the Arduino board
4. Change the time and degrees in the coding file and see how the servo moves
e.g., `delay(30)` or `delay(10)` ; from 180 to 90 or 120
5. Add more codes in the “void loop()”

```
e.g., for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 180 degrees
// in steps of 1 degree
myservo.write(pos);          // tell servo to go to position in variable 'pos'
delay(10);                   // waits 15ms for the servo to reach the position
}
```



```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo  
// twelve servo objects can be created on most boards
```

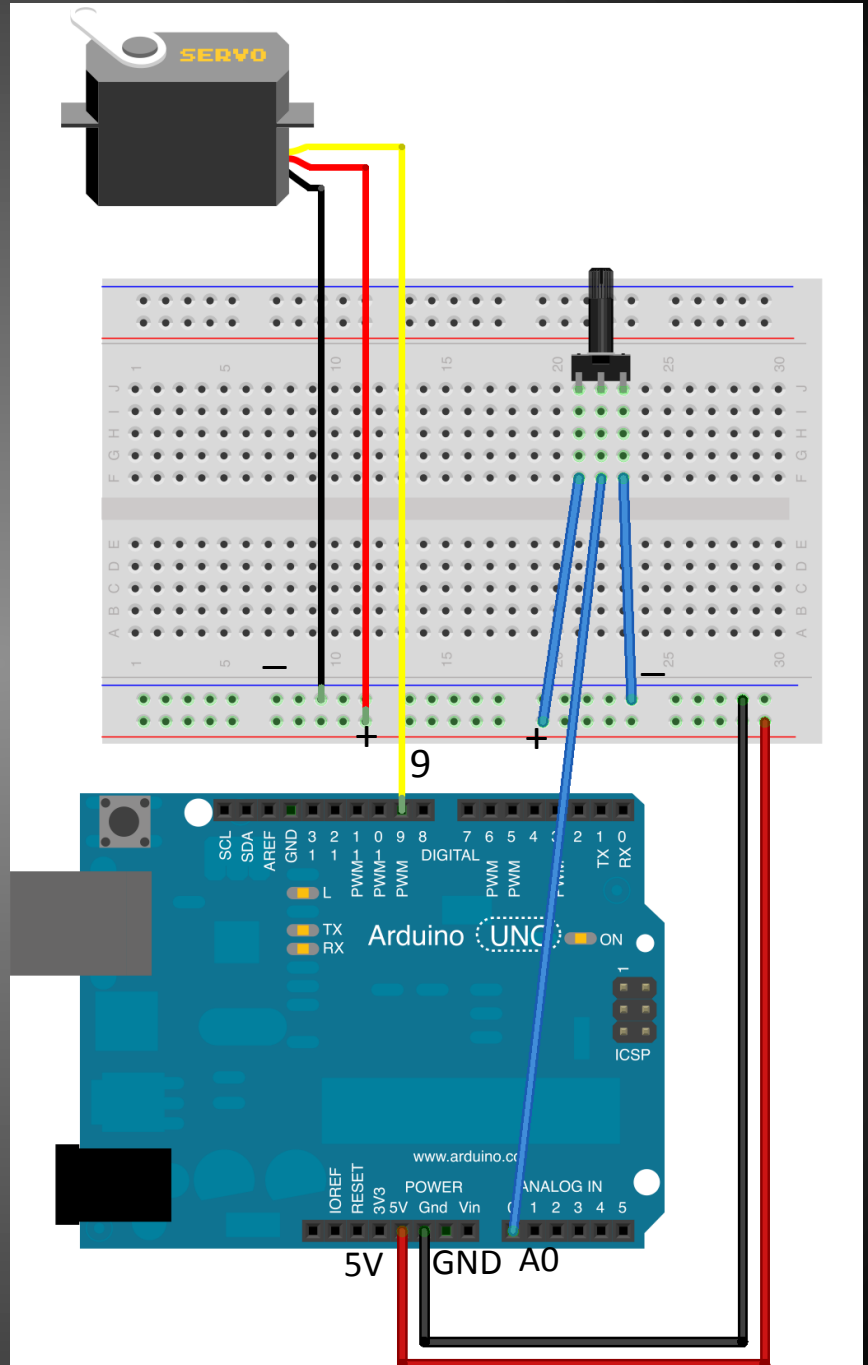
```
int pos = 0; // variable to store the servo position
```

```
void setup() {  
  myservo.attach(9); // attaches the servo on pin 9 to the servo object  
}
```

```
void loop() {  
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees  
    // in steps of 1 degree  
    myservo.write(pos); // tell servo to go to position in variable 'pos'  
    delay(15); // waits 15ms for the servo to reach the position  
  }  
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees  
    myservo.write(pos); // tell servo to go to position in variable 'pos'  
    delay(15); // waits 15ms for the servo to reach the position  
  }  
}
```


Design 3: Servo Motor & Potentiometer

1. Connect the servo motor and the potentiometer to the Arduino board.
2. Use alligator wires to connect the potentiometer
3. Open the code from the library: File-> Examples-> Servo-> **Knob**
4. Verify and download the code to the Arduino board
5. Rotate the potentiometer



```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo
```

```
int potpin = 0; // analog pin used to connect the potentiometer
```

```
int val; // variable to read the value from the analog pin
```

```
void setup() {
```

```
  myservo.attach(9); // attaches the servo on pin 9 to the servo object  
}
```

```
void loop() {
```

```
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)  
  val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)  
  myservo.write(val); // sets the servo position according to the scaled value  
  delay(15); // waits for the servo to get there  
}
```

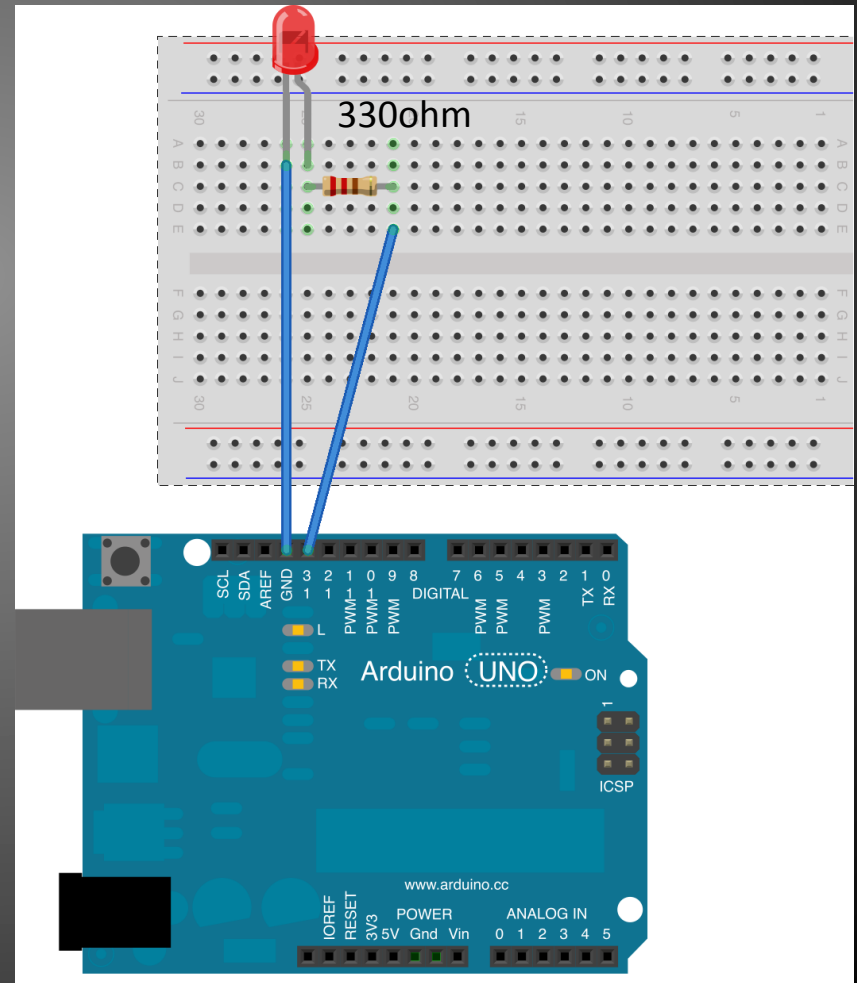
Potentiometer-LED

1. Connect the potentiometer directly to the Arduino board
2. Open the code: File-> Example->Analog->**AnalogInOutSerial**
3. Change the code pin no. from 9 to 13 “const int analogOutPin = 13”
4. Connect LED to the digital Pin no. 13
5. Rotate the potentiometer -> LED light on and off - +

5V

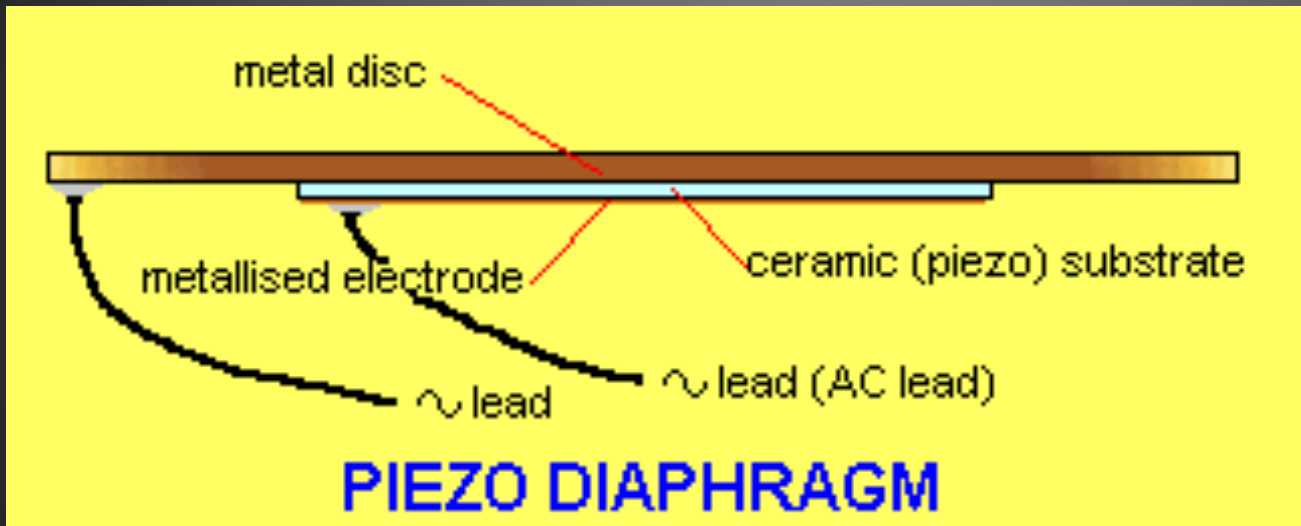


A0 GND



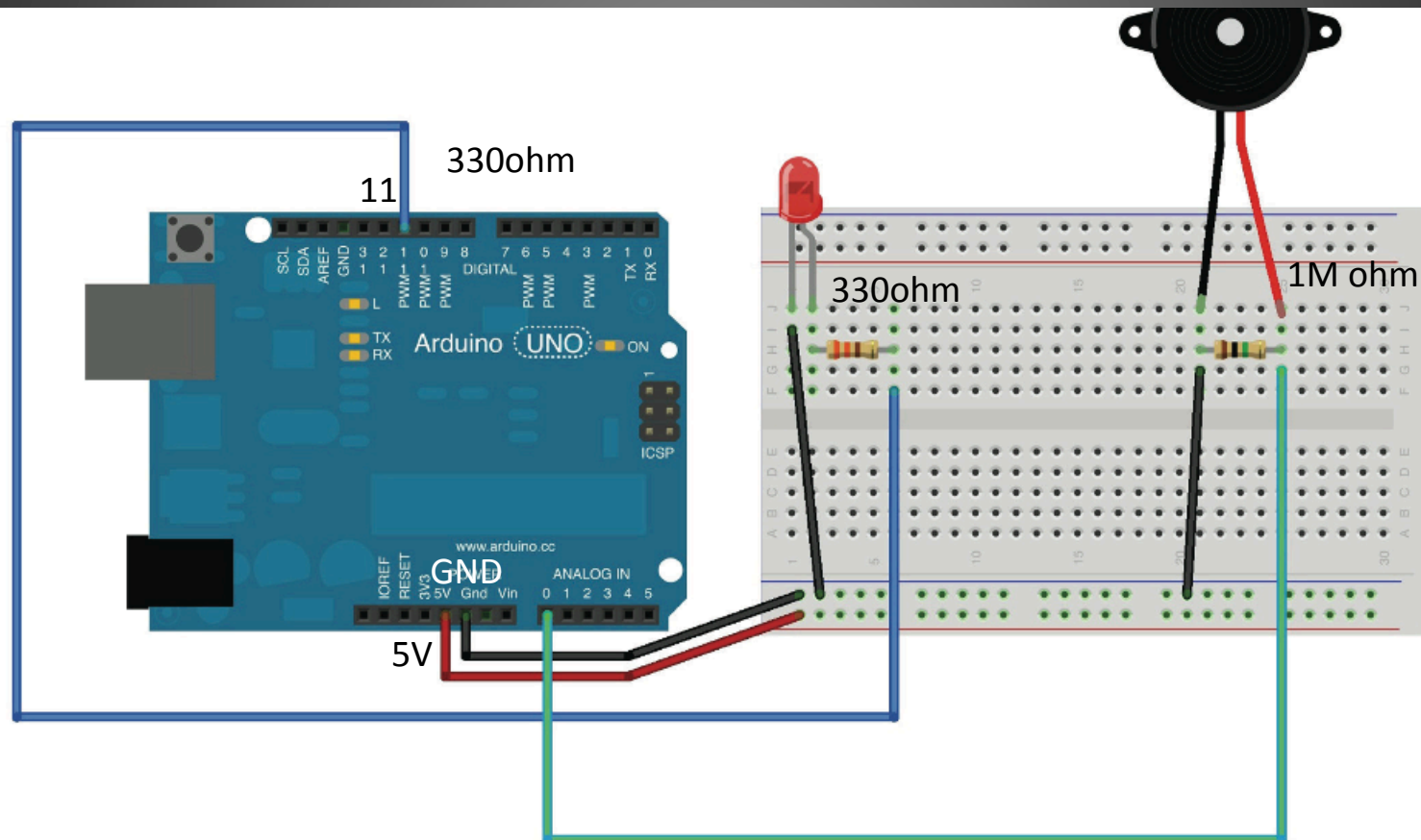
Optional

* **A Piezo buzzer** has a Piezo disc and an oscillator inside. When the buzzer is powered, the oscillator generates a frequency and the Piezo element vibrates accordingly to produce the sound.



Design 1: Piezo Sensor-Knock Knock

1. Connect the piezo element directly to the Arduino board.
2. Open the code from the library: File-> Examples-> 06 Sensors-> **Knock**
3. Change the code: LED pin from 13 to 11
4. Verify and download the code to the Arduino board -> open the Serial Monitor
5. Change the work "Knock" into other words and upload the code to the Arduino board-> open and check the Serial Monitor



```
// these constants won't change:
const int ledPin = 11;    // led connected to digital pin 13
const int knockSensor = A0; // the piezo is connected to analog pin 0
const int threshold = 100; // threshold value to decide when the detected sound is a knock or not
```

```
// these variables will change:
int sensorReading = 0;    // variable to store the value read from the sensor pin
int ledState = LOW;       // variable used to store the last LED status, to toggle the light
```

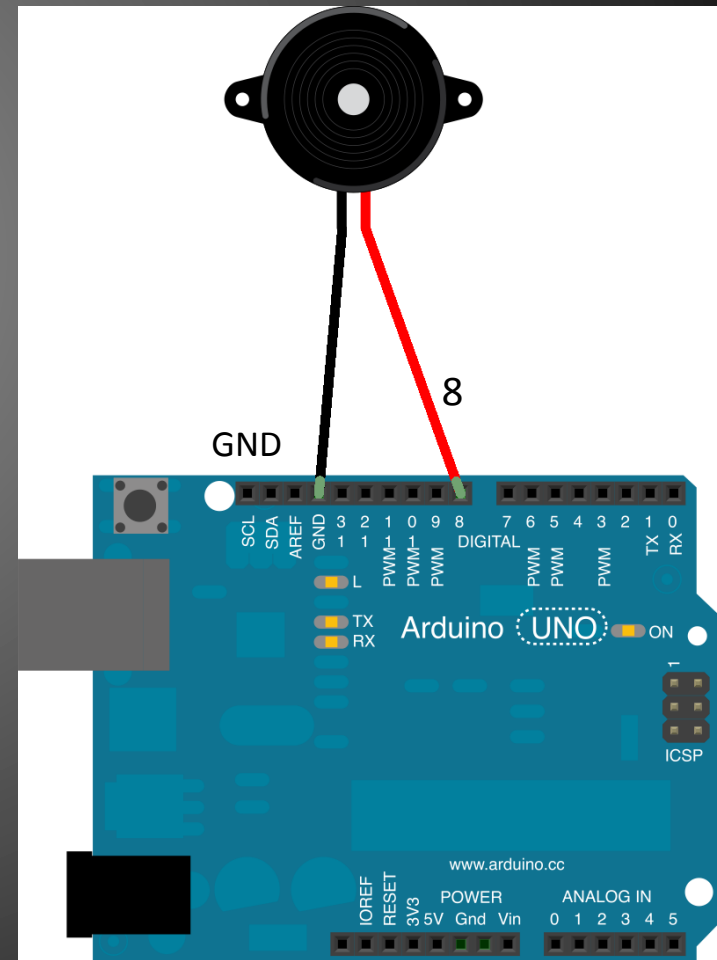
```
void setup() {
  pinMode(ledPin, OUTPUT); // declare the ledPin as as OUTPUT
  Serial.begin(9600);      // use the serial port
}
```

```
void loop() {
  // read the sensor and store it in the variable sensorReading:
  sensorReading = analogRead(knockSensor);

  // if the sensor reading is greater than the threshold:
  if (sensorReading >= threshold) {
    // toggle the status of the ledPin:
    ledState = !ledState;
    // update the LED pin itself:
    digitalWrite(ledPin, ledState);
    // send the string "Knock!" back to the computer, followed by newline
    Serial.println("Knock!");
  }
  delay(100); // delay to avoid overloading the serial port buffer
}
```

Design 2: Piezo Buzzer-Melody

1. Connect the piezo element directly to the Arduino board: Red wire (+) to the digital pin 8 and black wire (-) to GND.
2. Open the code from the library: File-> Examples-> Digital-> **ToneMelody**
3. Verify and download the code to the Arduino board
4. Modify the Red notes and numbers to change the melody



```

*/
#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

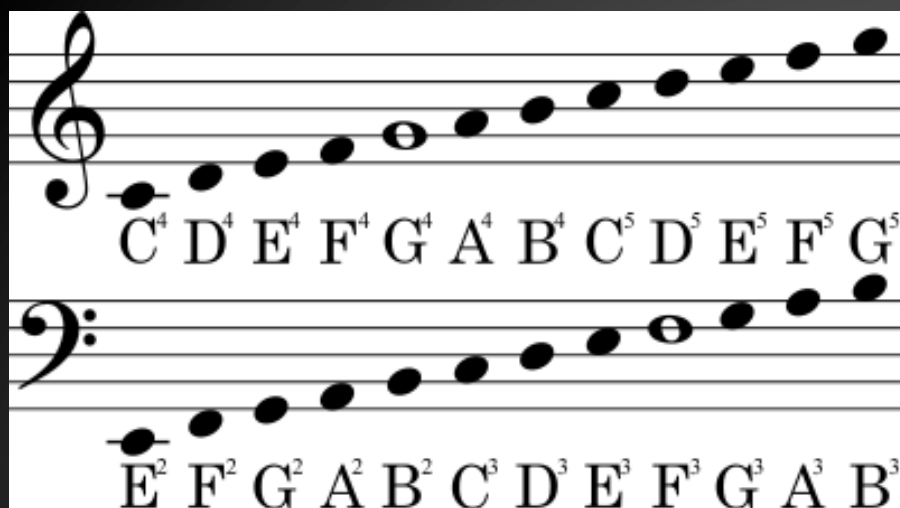
void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {










    // to calculate the note duration, take one second
    // divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // stop the tone playing:
    noTone(8);
  }
}

void loop() {
  // no need to repeat the melody.
}

```

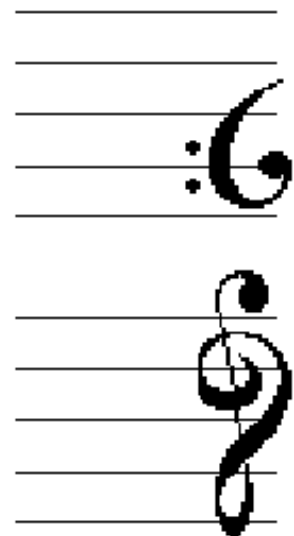


Note	Rest	American name	British name
		whole note	semibreve
		half note	minim
	 or 	quarter note	crotchet
		eighth note	quaver
		sixteenth note	semiquaver

Symbol-Value in array

whole note 1
half note 2
quarter note 4
eighth note 8
sixteenth note 16
 thirty-second note 32
 sixty-fourth note 64

		MIDI number	Note name	Keyboard	Frequency	
21			A0		27.500	
23	22		B0		30.868	29.135
24			C1		32.703	
26	25		D1		36.708	34.648
28	27		E1		41.203	38.891
29			F1		43.654	
31	30		G1		48.999	46.249
33	32		A1		55.000	51.913
35	34		B1		61.735	58.270
36			C2		65.406	
38	37		D2		73.416	69.296
40	39		E2		82.407	77.782
41			F2		87.307	
43	42		G2		97.999	92.499
45	44		A2		110.00	103.83
47	46		B2		123.47	116.54
48			C3		130.81	
50	49		D3		146.83	138.59
52	51		E3		164.81	155.56
53			F3		174.61	
55	54		G3		196.00	185.00
57	56		A3		220.00	207.65
59	58		B3		246.94	233.08
60			C4		261.63	
62	61		D4		293.67	277.18
64	63		E4		329.63	311.13
65			F4		349.23	
67	66		G4		392.00	369.99
69	68		A4		440.00	415.30
71	70		B4		493.88	466.16
72			C5		523.25	
74	73		D5		587.33	554.37
76	75		E5		659.26	622.25
77			F5		698.46	
79	78		G5		783.99	739.99
81	80		A5		880.00	830.61
83	82		B5		987.77	932.33
84			C6		1046.5	
86	85		D6		1174.7	1108.7
88	87		E6		1318.5	1244.5
89			F6		1396.9	
91	90		G6		1568.0	1480.0
93	92		A6		1760.0	1661.2
95	94		B6		1975.5	1864.7
96			C7		2093.0	
98	97		D7		2349.3	2217.5
100	99		E7		2637.0	2489.0
101			F7		2793.0	
103	102		G7		3136.0	2960.0
105	104		A7		3520.0	3322.4
107	106		B7		3951.1	3729.3
108			C8	J. Wolfe, UNSW	4186.0	



Design 3: Piezo Buzzer-Songs

1. Connect the piezo element directly to the Arduino board: Red wire (+) to the digital pin 3 and black wire (-) to GND.
2. Download the code “**Melody-super_mario**” and upload to the Arduino board

