

MT06 - ARDUINO Y PROGRAMACIÓN

FAB
LAB
BCN

MT06* Introducción Arduino / Programación		
MASTER CLASS	7/12	8 a 930hs
LABVIRTUAL	14/12	8 a 930hs
REVIEW / WEBINAR	17/12	8 a 10hs

Clase 06/12:

- **Presentación MT06**
- **Iniciación a Arduino y electrónica**
- **Q&A - conversation**
- **Desafío**

INTRODUCCIÓN

THE MAKER'S BILL OF RIGHTS

- Meaningful and specific parts lists shall be included.
- Cases shall be easy to open. ■ Batteries shall be replaceable. ■ Special tools are allowed only for darn good reasons. ■ Profiting by selling expensive special tools is wrong, and not making special tools available is even worse. ■ Torx is OK; tamperproof is rarely OK.
- Components, not entire subassemblies, shall be replaceable. ■ Consumables, like fuses and filters, shall be easy to access. ■ Circuit boards shall be commented.
- Power from USB is good; power from proprietary power adapters is bad. ■ Standard connectors shall have pinouts defined. ■ If it snaps shut, it shall snap open. ■ Screws better than glues. ■ Docs and drivers shall have permalinks and shall reside for all eternity at archive.org. ■ Ease of repair shall be a design ideal, not an afterthought. ■ Metric or standard, not both.
- Schematics shall be included.

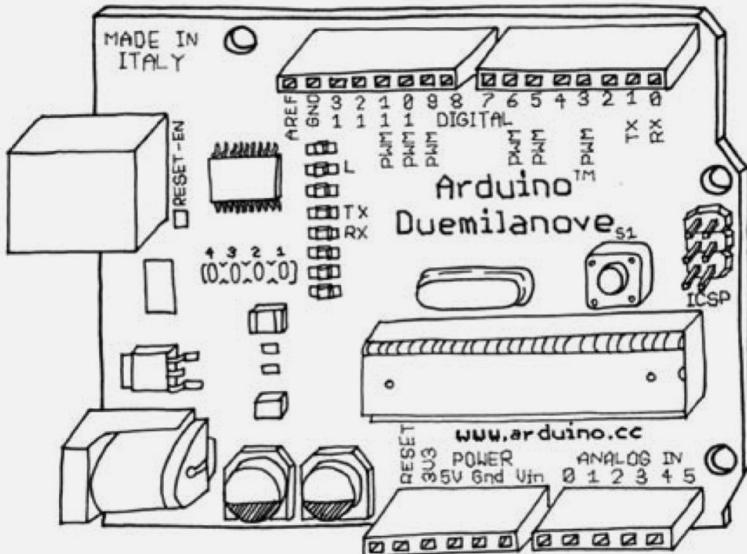
Make:
technology on your time

Drafted by Mister Jalopy, with assistance from Philip Torrone and Simon Hill.

OPEN // SOURCE

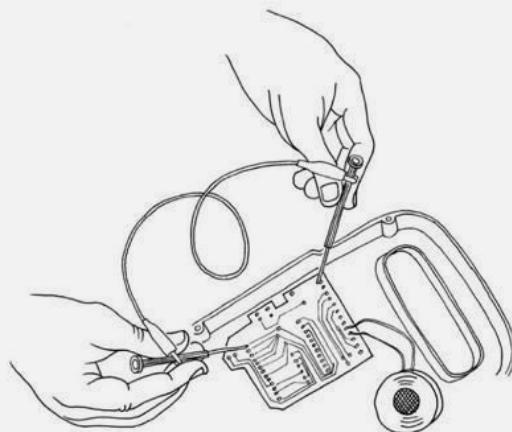
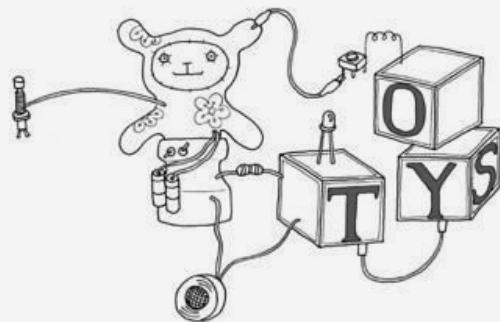
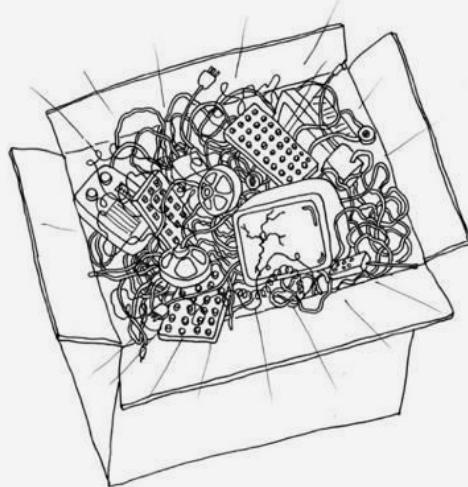
Es un manifiesto para aquellos que quieren la libertad de jugar, rehacer, reparar, recombinar y actualizar las cosas que poseen. Publicado por Make: en 2005, articula una lista de 17 mandamientos que los fabricantes deben seguir para hacer que sus productos sean reparables y pirateados.

PHYSICAL// COMPUTING



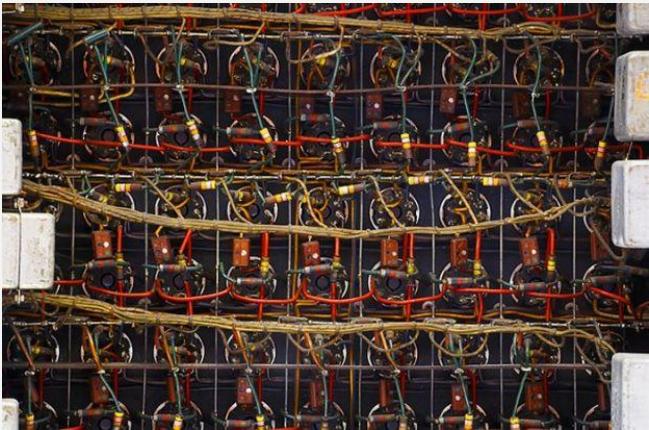
This is a compilation based on many sources, all images
are property of their respective owners.

PHYSICAL// COMPUTING



HISTORIA

COMPUTING // HISTORY



En los Laboratorios Bell, John Bardeen, Walter Houser Brattain y William Shockley inventan el transistor.

1946



Jack S. Kilby construye el primer circuito integrado.

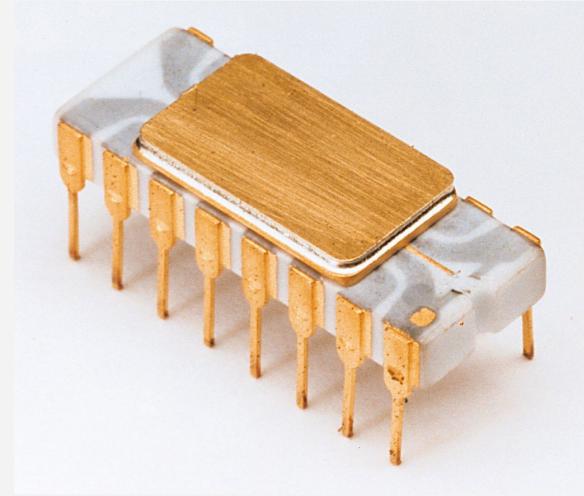
1959

COMPUTING // HISTORY



The Apollo Guidance Computer:

1968



In 1971 Intel released the first microcontroller, the Intel 4004:

1974

COMPUTING // HISTORY



[Steve Jobs](#), [Steve Wozniak](#), [Mike Markkula](#) fundan Apple.

1976



[Microsoft](#) presenta su sistema operativo [MS-DOS](#), por encargo de IBM.

1982

COMPUTING // HISTORY



[Motorola](#) lanzó, el 3 de enero de 1996, el StarTAC, era el teléfono móvil más pequeño y ligero del planeta. Lo que redefinió la idea de cómo debería ser un celular.

1996



Presentación del primer [iPhone](#), por la empresa Apple, un [teléfono inteligente](#) o *smartphone*.

2007

COMPUTING //

HISTORY



La Raspberry Pi Zero recibe una actualización, con una tarjeta inalámbrica (Wi-Fi y Bluetooth). También es la primera versión con un puerto de cámara integrado.

2017

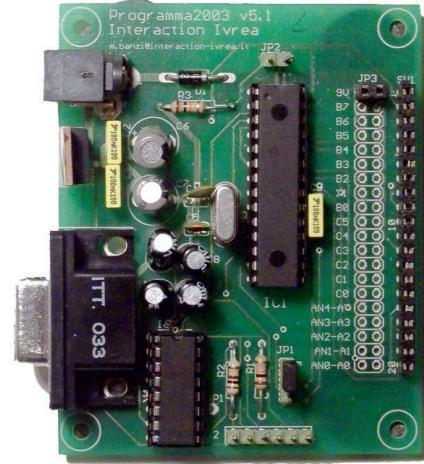
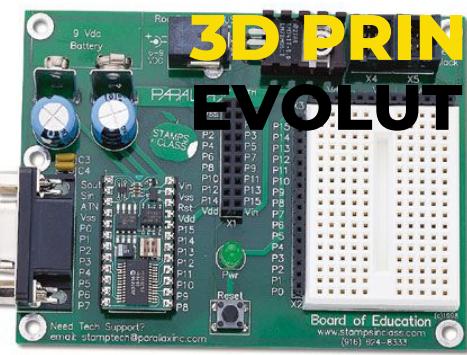
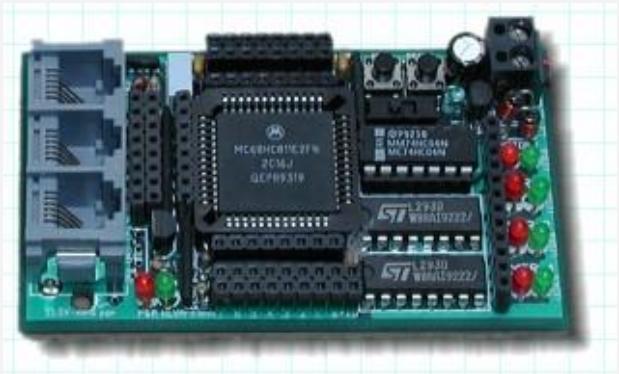
COMPUTING // HISTORY

Apollo	Arduino	Raspberry	iPhone 6
2 MHz	16 MHz	700 MHz x 2	1400 MHz x 2
4kB RAM	2kB RAM	512MB RAM	1GB RAM
72kB ROM	32kB FLASH	8GB FLASH	16GB FLASH

3D PRINTING// EVOLUTION

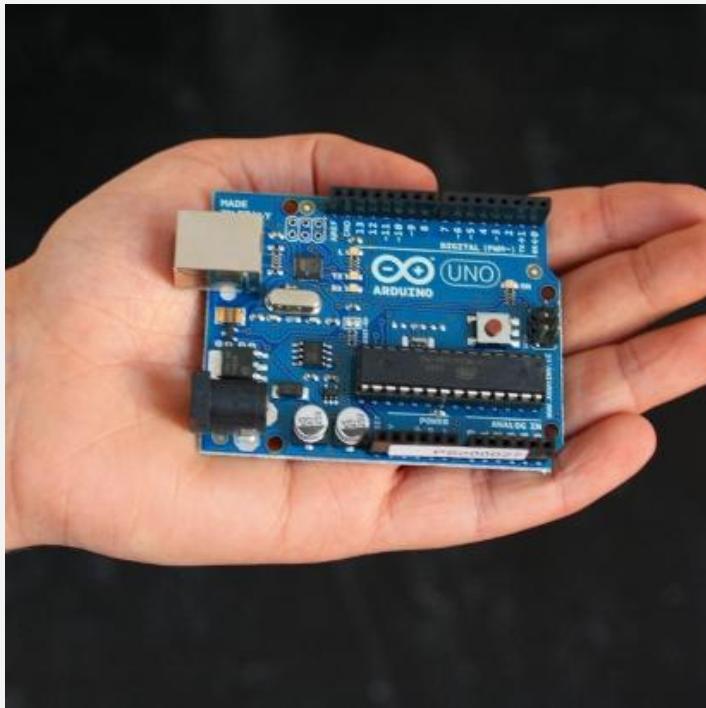


"The 3D Printing Revolution" Richard D'Aveni



ARDUINO

ARDUINO// ORIGINS

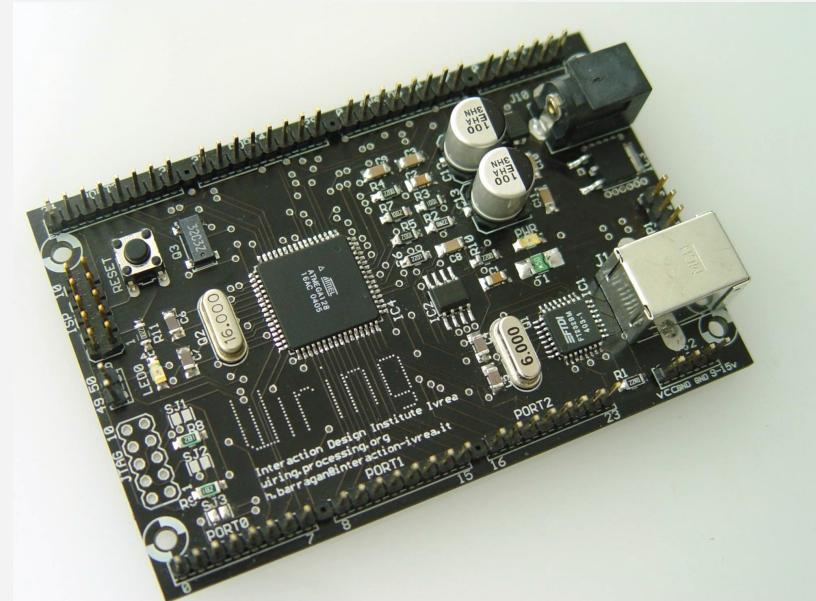
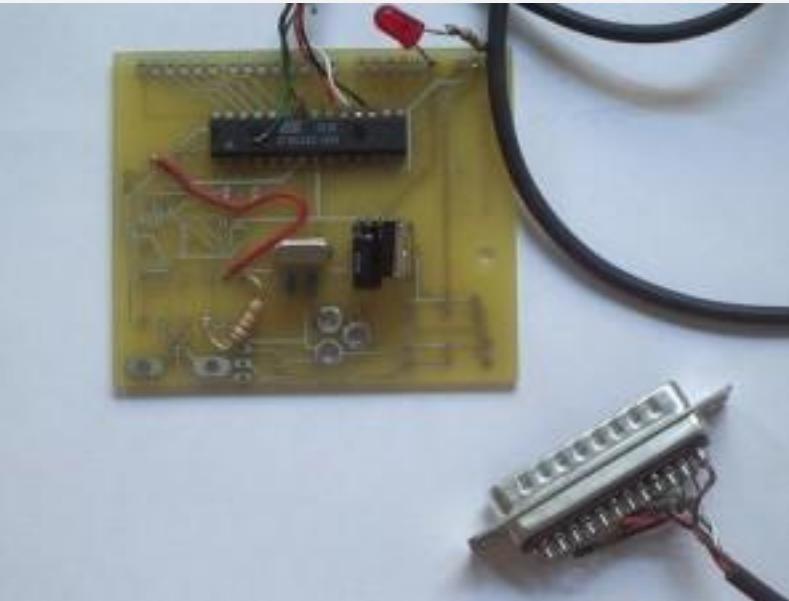


El proyecto «Arduino» se inició en el año 2005 como un proyecto enfocado a estudiantes en el Interaction Design Institute Ivrea (IDI), de Ivrea (Italia)

ARDUINO// ORIGINS



ARDUINO // EVOLUTION



"The 3D Printing Revolution" Richard D'Aveni

ARDUINO //

EVOLUTION



Arduino Uno



Arduino Leonardo



Arduino Tre



Arduino Micro



Arduino Due



Arduino Yún



Arduino Robot



Arduino Esplora

ARDUINO// EVOLUTION



Arduino Mega ADK



Arduino Ethernet



LilyPad Arduino USB



LilyPad Arduino
Simple



Arduino Nano



Arduino Pro Mini



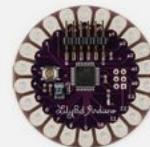
Arduino Mega 2560



Arduino Mini



LilyPad Arduino
SimpleSnap



LilyPad Arduino



Arduino Pro

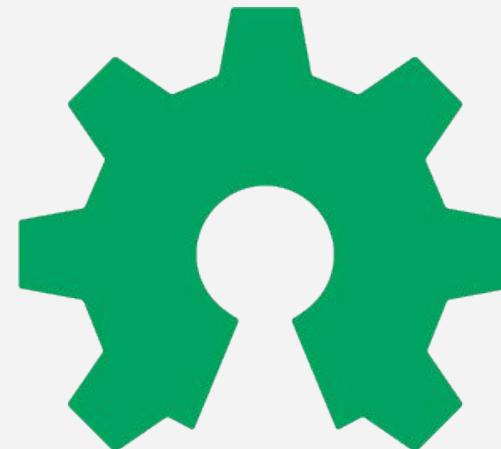


Arduino Fio

ARDUINO// ECOSYSTEM

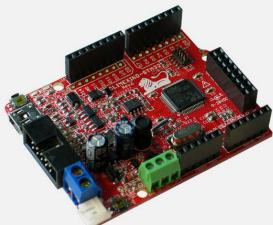


open source



**open source
hardware**

ARDUINO// ECOSYSTEM



http://www.bareconductive.com/wp-content/uploads/2015/03/BareConductive_MicrocontrollerGuide1.pdf

https://www.sparkfun.com/arduino_guide

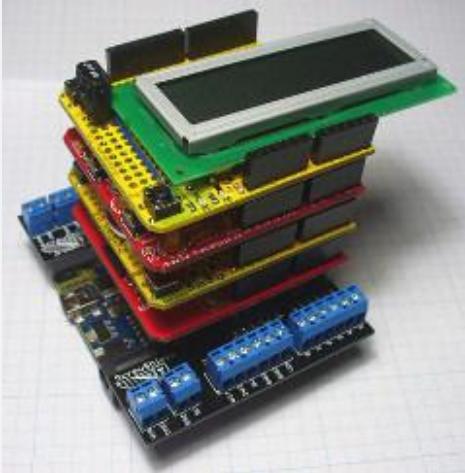
<https://learn.adafruit.com/embedded-linux-board-comparison>

<http://www.makershed.com/Articles.asp?ID=302&Click=127972>

DETAILS / TECH SPECS

	ARDUINO LEONARDO	TOUCH BOARD	LILYPAD ARDUINO	ADAFRUIT TRINKET	MAKEY MAKEY	DENTAKU OTOTO	RASPBERRY PI (MODEL B+)	INTEL EDISON (ARDUINO B/O KIT)	BEAGLEBONE BLACK	UDOO QUAD	SPARK CORE	FREESCALE FREEDOM	
ONBOARD PROCESSOR	ATmega32U4 at 16 MHz	ATmega32U4 at 16 MHz	Atmega328 at 8 MHz	Attiny85 at 8 MHz	ATmega32U4 at 16 MHz	ATmega32U4 at 16 MHz	BCM2835 at 700 MHz	Dual Core Intel Atom at 500 MHz	ARM Cortex-A8 AM335x at 1 GHz	2x 32-bit RISC processor at 200 MHz	Atmel SAM3X8E at 84 MHz	STM32F103 at 72 MHz	MKL46Z256VLL24 at 48 MHz
CO-PROCESSOR								Intel Quark microcontroller at 100 MHz					
RAM	2.5 KB	2.5 KB	2 KB	512 B	2.5 KB	2.5 KB	512 MB	1 GB	512 MB	1 GB	20 KB	32 KB	
Number of touch sensors	0	12	0	0	6	12	0	0	0	0	0	15	
Capacitive sensing (one-handed triggering)													
Proximity sensing (trigger without touch)													
Onboard MP3 / WAV / OGG / FLAC / MIDI playback													
Onboard polyphonic MIDI synthesis													
Micro SD card slot													
On / Off switch													
Lithium Polymer (LiPo) cell charging via USB													
Can run from a Lithium Polymer (LiPo) cell													
Can act as a USB serial device													
Can act as a USB keyboard or mouse													
Can act as a USB MIDI interface													
Arduino shield compatible													
Onboard headphone / line output													
Onboard video out													
Can be programmed using the Arduino IDE													
Number of USB host sockets	0	0	0	0	0	0	4	0	1	2	0	1	
Onboard LAN													
Onboard WIFI													
Onboard SATA													
Works out of the box													
Approvals	CE, FCC	CE, FCC			CE	CE, FCC	CE, FCC		CE, FCC	CE, FCC		CE	
Country of manufacture	ITALY	UK	USA	USA	CHINA	UK	UK	CHINA	USA	ITALY	CHINA	CHINA	

ARDUINO// SHIELDS



<http://shieldlist.org/>

Arduino Shield List

Pin usage details for 317 shields from 125 makers, and counting!

> Home

- [4D Systems](#) (4)
- [Adafruit Industries](#) (9)
- [AeroQuad](#) (2)
- [Andre Concalves](#) (2)
- [antrax Datentechnik](#) (1)
- [Applied Platronics](#) (1)
- [ArduCapSense](#) (1)
- [Arduino](#) (3)
- [Argent Data Systems](#) (1)
- [Asynclabs](#) (3)
- [Batsocks](#) (2)
- [Ben Combee](#) (1)
- [Bhasha Technologies](#) (2)
- [Biltronics](#) (2)
- [Blushing Boy](#) (1)
- [Carlos Neves](#) (1)
- [Chesters Garage](#) (1)
- [Chips To Bits](#) (1)
- [Circuit Ideas Design](#) (1)
- [Circuits At Home](#) (2)
- [CISECO](#) (4)
- [Collin Schulz](#) (1)
- [Conductive Resistance](#) (1)
- [Control Connection](#) (1)
- [Creathon Inc](#) (1)
- [Critical Velocity](#) (5)
- [Critter and Guitari](#) (1)
- [Excamera Labs](#) (1)
- [Faz Jaxton](#) (1)
- [FlamingoEDA](#) (1)
- [Freetronics](#) (12)
- [Futura Elettronica](#) (2)
- [Galileo 7](#) (4)
- [GeekOnFire](#) (1)
- [GfxHax](#) (1)
- [GinSing](#) (1)
- [Gravitech](#) (1)
- [Homeroasters](#) (1)
- [HW Kitchen](#) (1)
- [ITead Studio](#) (6)
- [Jee Labs](#) (1)
- [Jimmie Rodgers](#) (1)
- [John Liu](#) (3)
- [Knutsel](#) (1)
- [Lars Schumann](#) (3)
- [Libelium](#) (6)
- [LinkSprite](#) (5)
- [Liquidware](#) (12)
- [Logos Electromechanical](#) (1)
- [Low Voltage Labs](#) (1)
- [Luke Weston](#) (1)
- [macetech](#) (3)
- [Maker Shed](#) (1)
- [Mark Smouli](#) (1)
- [Ocean Controls](#) (1)
- [Open Electronics](#) (1)
- [PDK Solutions](#) (1)
- [Photoduino](#) (1)
- [Pololu](#) (1)
- [Practical Maker](#) (5)
- [Protuino](#) (2)
- [Ray's Hobby](#) (1)
- [Renbotics](#) (2)
- [RepRap Research Foundation](#) (1)
- [Ro-Bot-X Designs](#) (1)
- [Robot Power](#) (1)
- [RobotPirate](#) (1)
- [Rocket Scream](#) (1)
- [Rogue Robotics](#) (1)
- [Rugged Circuits](#) (6)
- [Samurai Circuits](#) (3)
- [Scattered Mind](#) (1)
- [Schmelle2](#) (6)
- [Seeed Studio](#) (14)
- [Shieldstudio](#) (2)
- [SK Pang Electronics](#) (3)
- [Small Room Labs](#) (1)
- [Smart Energy Groups](#) (1)
- [Snoothlab](#) (8)
- [Solarbotics](#) (2)
- [SonikTech](#) (1)

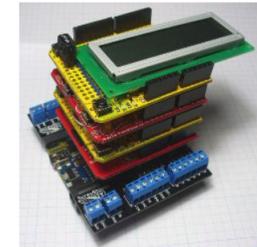


Photo: John Boxall

Tweets by @shieldlist

Arduino Shield List Retweeted

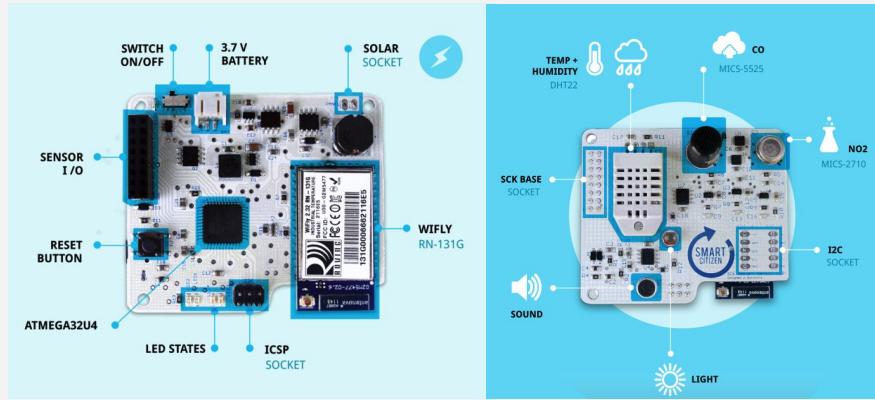
 Freetronics
@freetronics

DIY keyed headers for Arduino:
freetronics.com.au/blogs/news/ski...

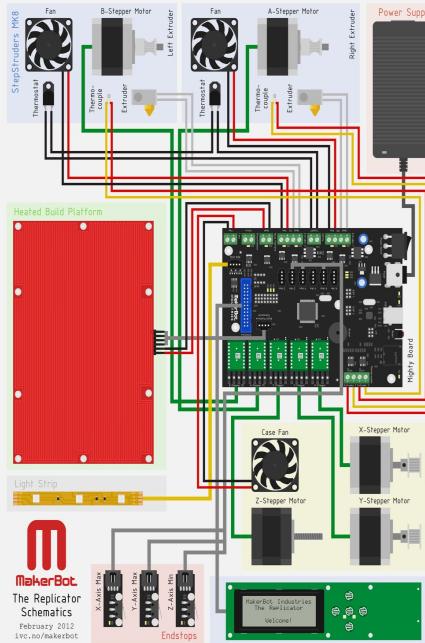


ARDUINO// SHIELDS

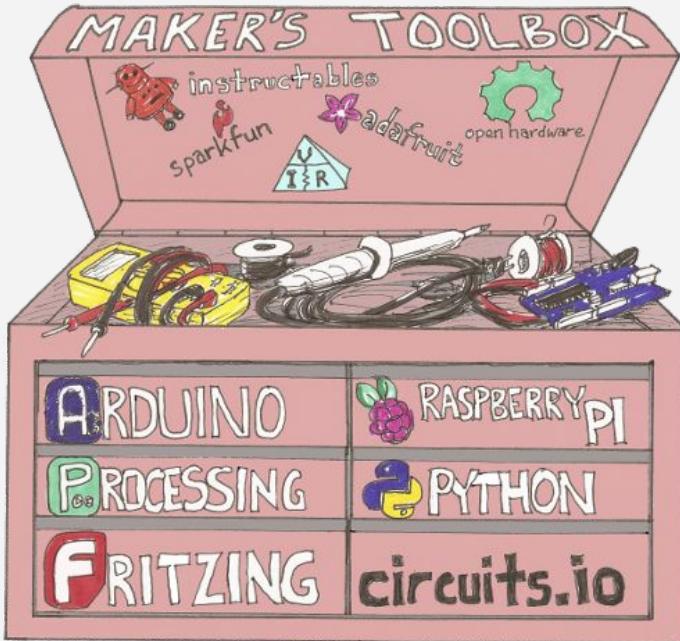




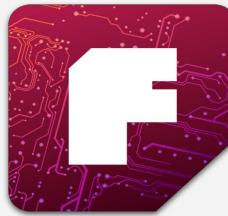
ARDUINO// SHIELDS



ARDUINO// ECOSYSTEM



arduino.cc



fritzing.org

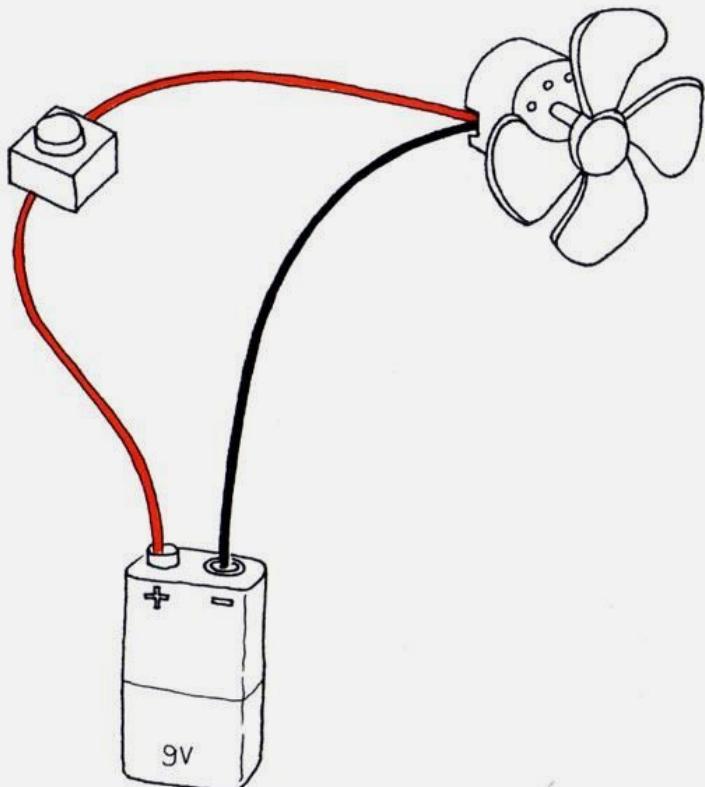


processing.org

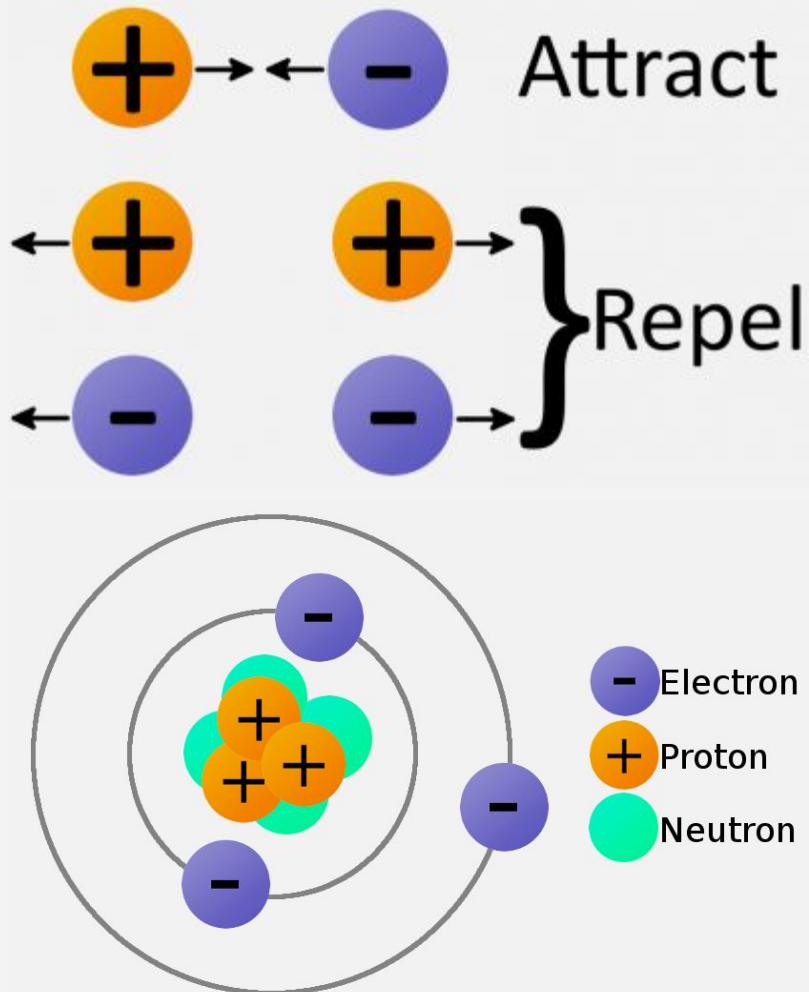
ELECTRICIDAD

ELECTRICIDAD//

101



En esta sección, comprenderemos los conceptos básicos de la electricidad y cómo puede ayudarnos a representar la información. Comenzaremos con el circuito más simple e inferiremos de él cómo se puede almacenar, administrar y enviar la información a otros componentes.



ELECTRICIDAD// CARGA

Todo se trata de carga eléctrica. Los objetos pueden contener una carga eléctrica (positiva o negativa) o no (lo que significa que son neutrales).

En las partículas subatómicas, los protones tienen carga positiva, los electrones tienen carga negativa y los neutrones son neutrales.

Estas partículas las encontramos en los átomos y definen el mundo tal y como lo vemos, siendo la estructura subyacente de lo que entendemos como elementos (Hidrógeno, Helio, Urano ...). Todos sabemos que las partículas se cargan así:

ELECTRICIDAD// VOLTAJE



AC



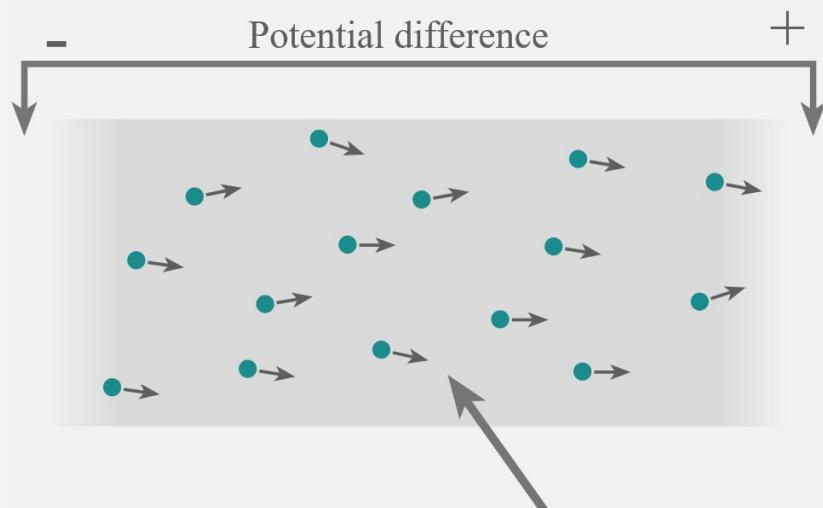
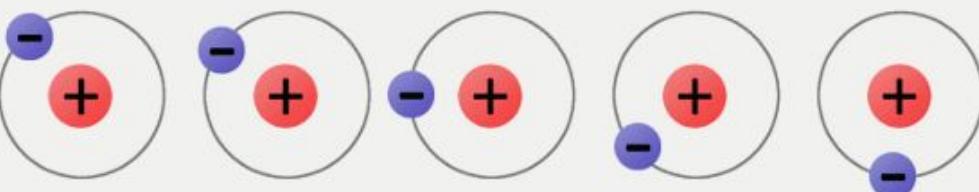
DC

Cuando forzamos a los electrones a agruparse en un área determinada, dejando otra área sin electrones, creamos una diferencia de voltaje. Este voltaje es la relación entre la energía que aplicamos y la carga eléctrica:

$$\underline{E = V \times \text{carga}}$$

Cuando dos objetos tienen una diferencia de voltaje, podemos decir que sus electrones intentarán saltar entre sí creando un flujo de corriente, para equilibrar la situación y estabilizarse. El voltaje se expresa en voltios (V). Este voltaje puede ser constante en el tiempo o alterno:

ELECTRICIDAD// CORRIENTE



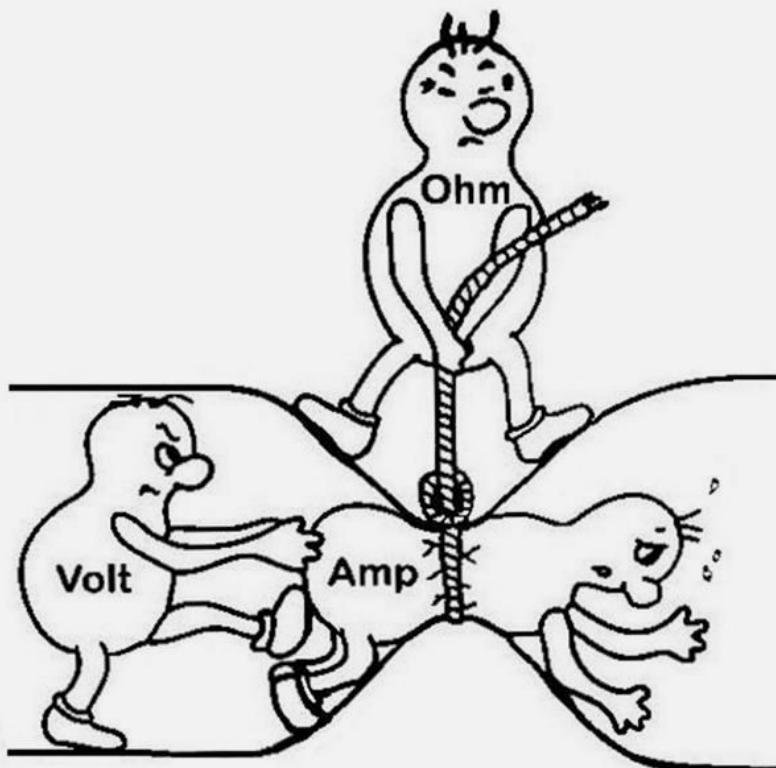
Electrical conductor
with free electrons

When two objects are subject to a difference voltage, electrons will try to come back to their position. When doing so, we say there is an *electric current* or just *current*.

This movement of electrons inside a material is measured in Amperes (A) or just Amps. If we have alternating voltage, we will have also alternating current (AC), and the same with constant voltage, in which case we will have DC.

ELECTRICIDAD//

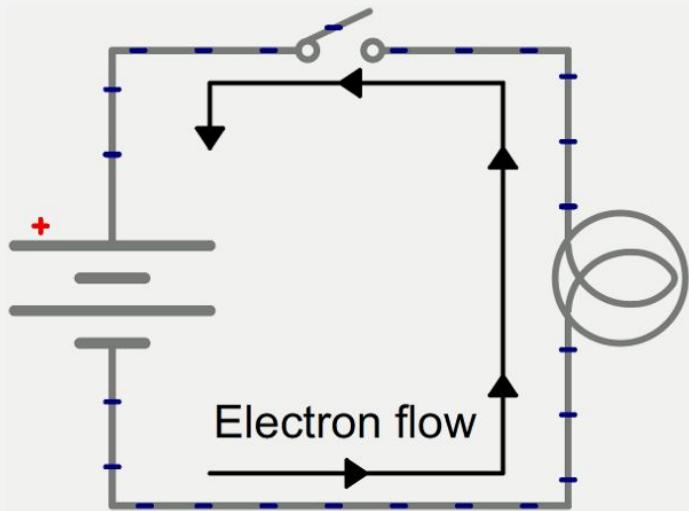
LEY DE OHM



Georg Ohm descubrió que el voltaje (V), la resistencia (R) y la corriente (I) van con la siguiente fórmula:

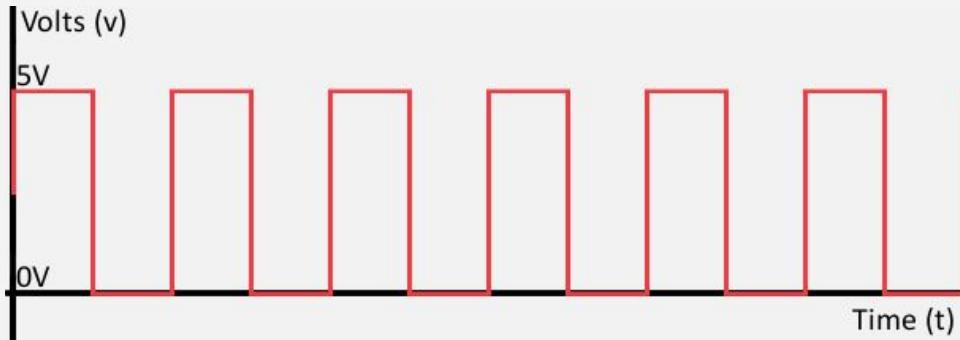
$$V = I \times R$$

Lo que significa que cuando la resistencia es alta, casi no hay corriente (y cuando la corriente es 0, significa que tenemos un circuito abierto). Cuando la resistencia es casi 0, la corriente puede ser muy grande, lo que lleva a lo que llamamos un corto.



ELECTRICIDAD// CONTROL

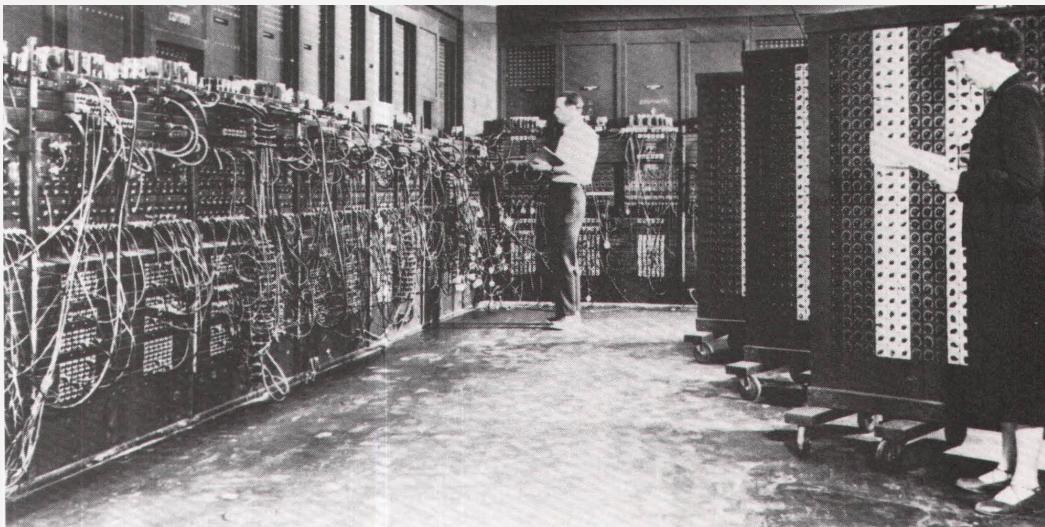
Lo bueno de esto es que podemos controlar este flujo (o la falta de él), ¡y luego podemos hacer cosas geniales!



Hemos estandarizamos lo que consideramos como 1 (alto voltaje) y 0 o bajo voltaje. A esto lo llamamos niveles lógicos:

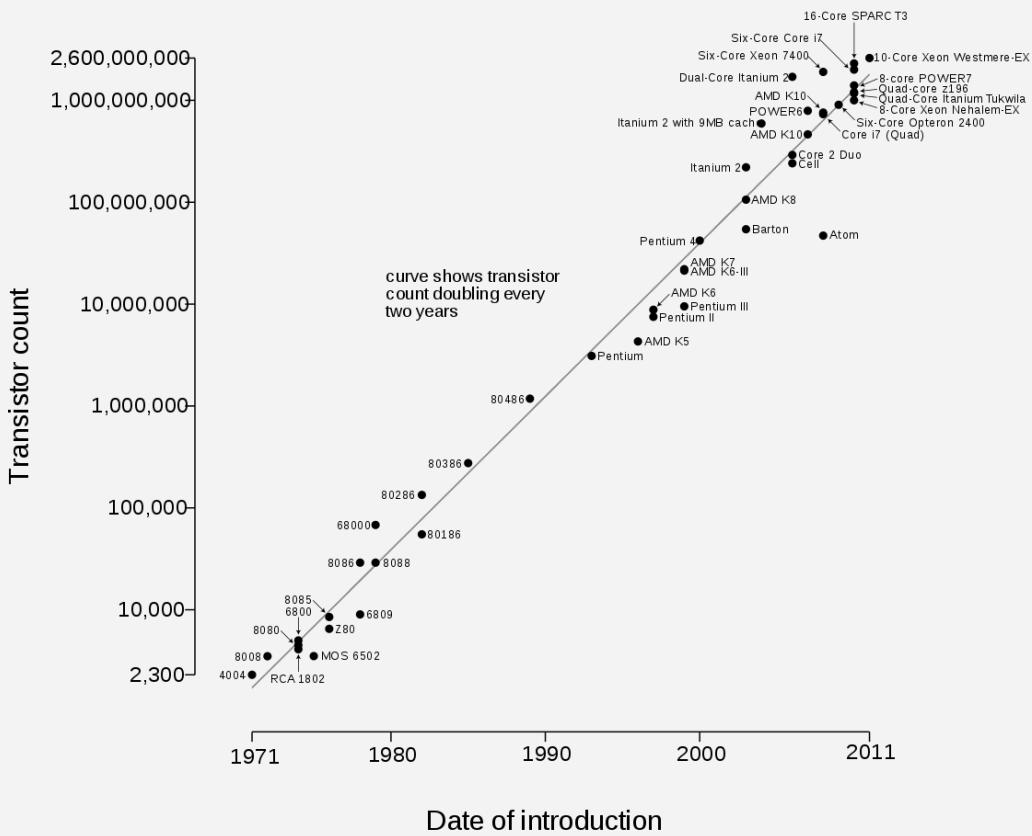
COMPUTING

COMPUTING // CONTROL



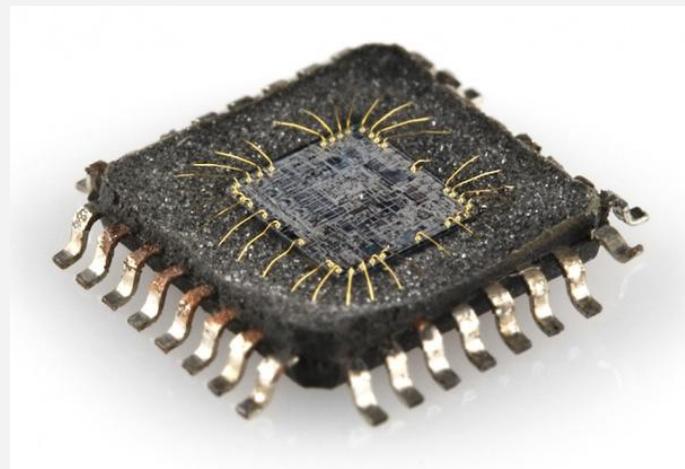
Las primeras máquinas informáticas eran programables en el sentido de que podían seguir la secuencia de pasos que se habían configurado para ejecutar, pero el "programa", o los pasos que debía ejecutar la máquina, se configuraban generalmente cambiando la forma en que se conectaban los cables, en un panel de conexiones o panel de conexiones.

Microprocessor transistor counts 1971-2011 & Moore's law



COMPUTING // CONTROL

La ley de Moore expresa que aproximadamente cada 2 años se duplica el número de transistores en un microprocesador.



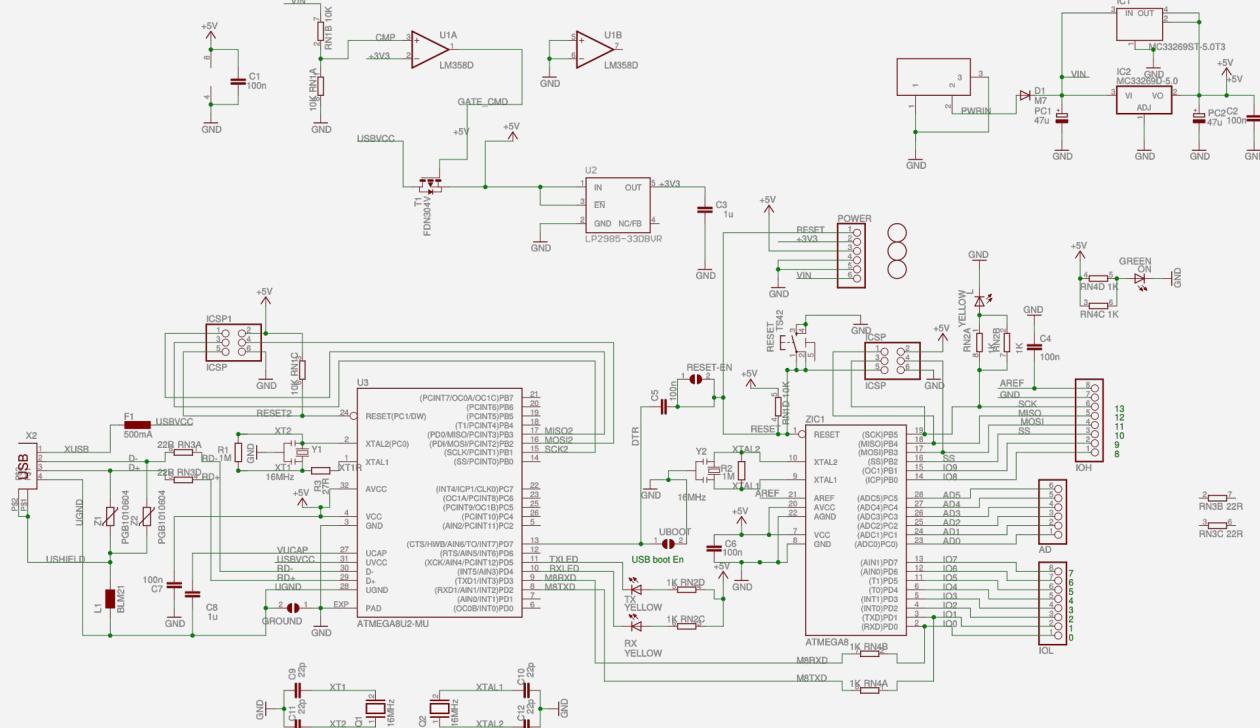
PHYSICAL COMPUTING

3D PRINTING// EVOLUTION

Arduino™ UNO Reference Design

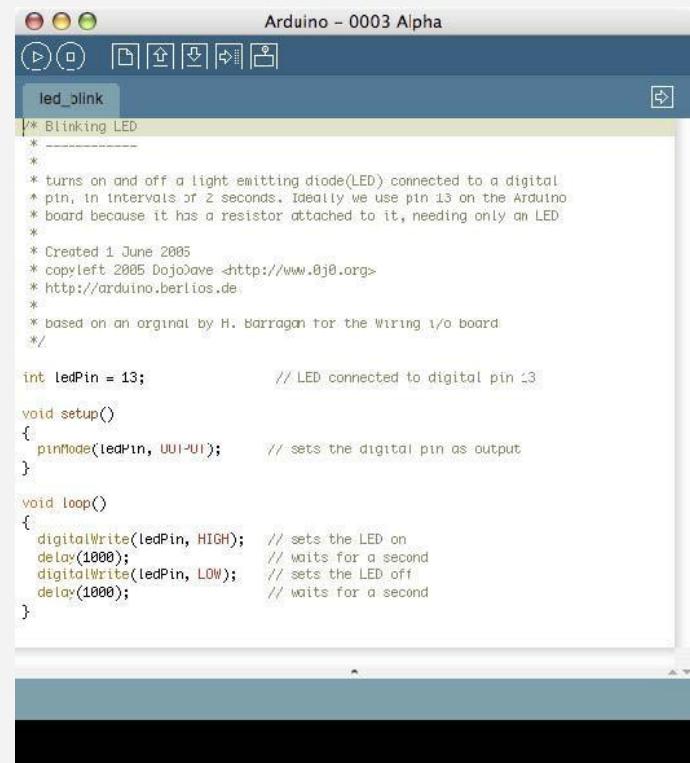
Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



"The 3D Printing Revolution" Richard D'Aveni

PHYSICAL // COMPUTING

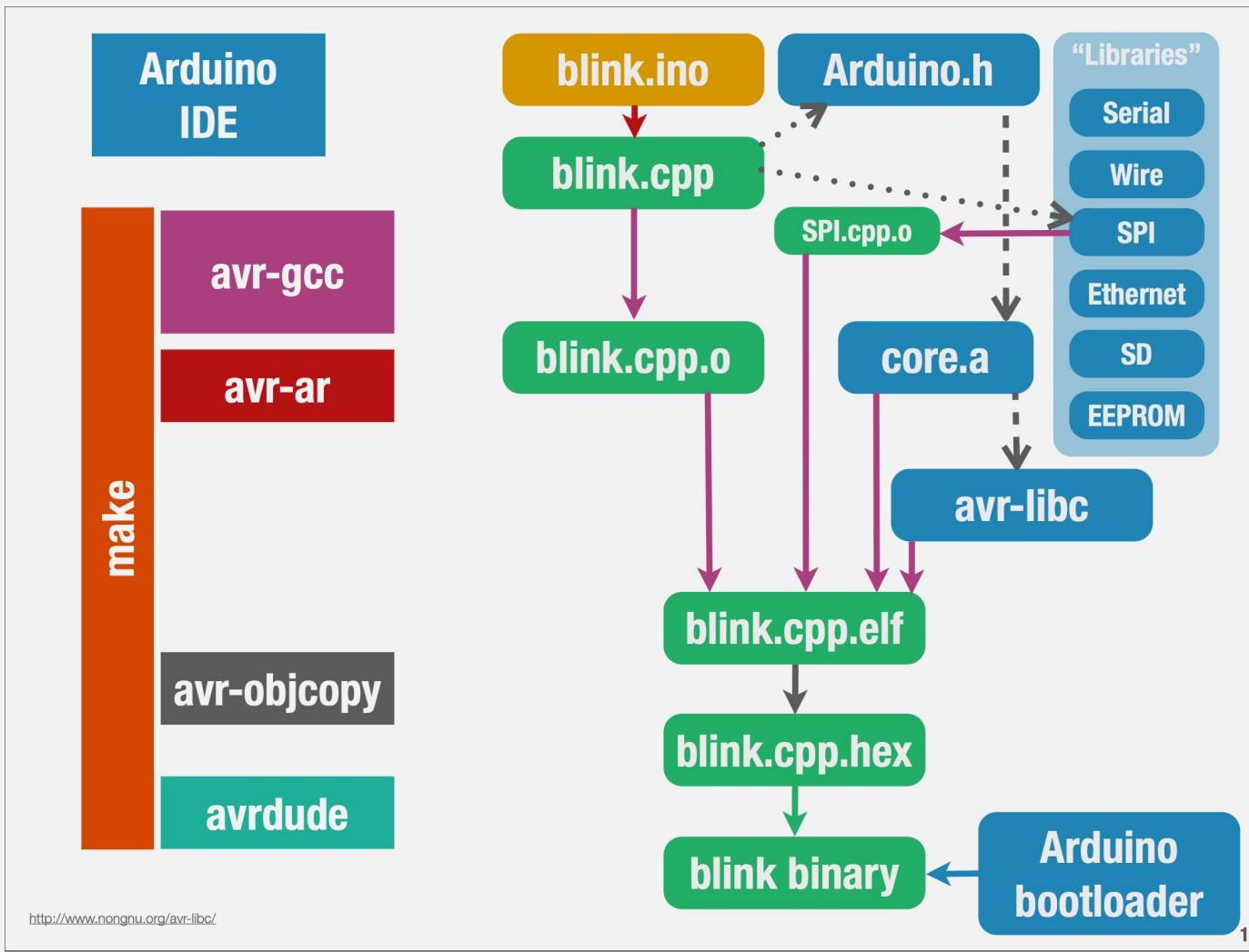


The screenshot shows the Arduino IDE interface with the title bar "Arduino - 0003 Alpha". The sketch name "led_blink" is highlighted in blue. The code editor contains the following C-like pseudocode:

```
/* Blinking LED
 *
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, in intervals of 2 seconds. Ideally we use pin 13 on the Arduino
 * board because it has a resistor attached to it, needing only an LED.
 *
 * Created 1 June 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 * based on an orginal by H. Barragon for the Wiring I/O board
 */
int ledPin = 13; // LED connected to digital pin 13

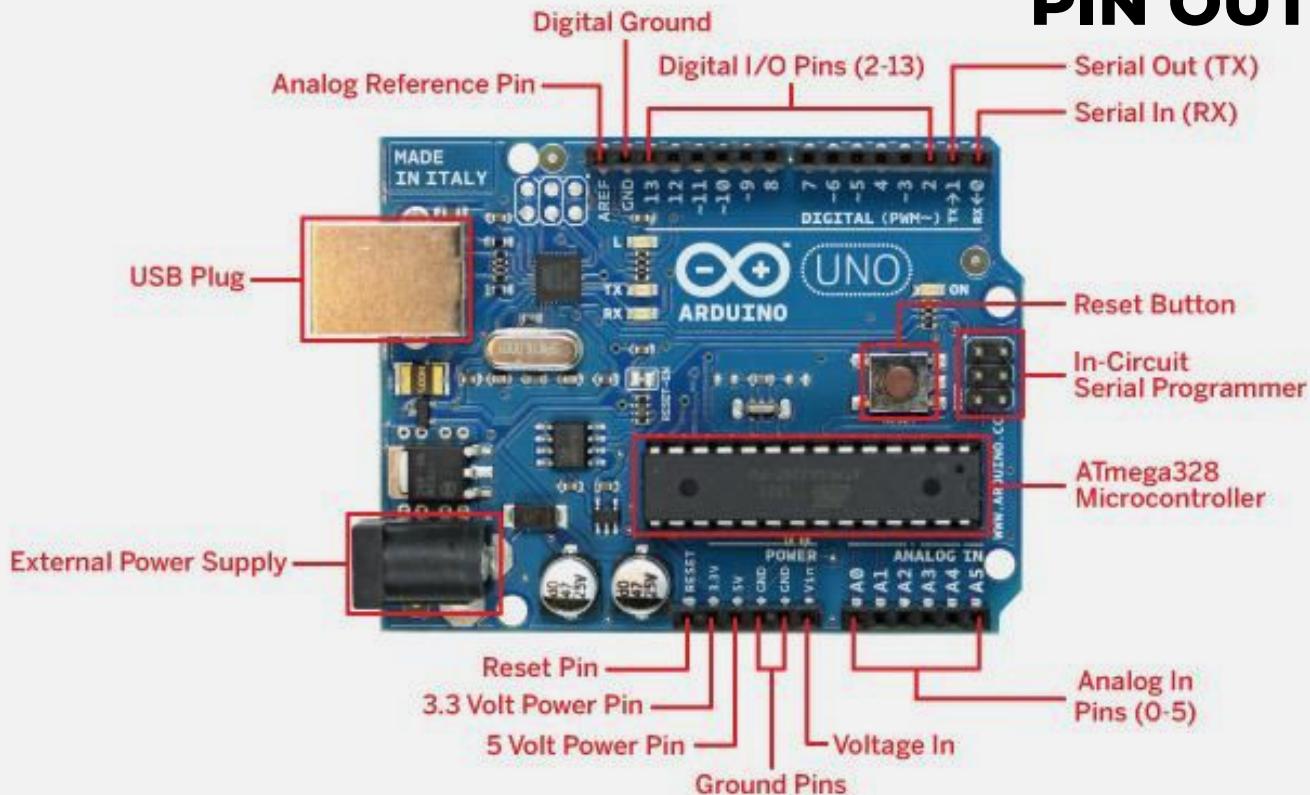
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000); // waits for a second
}
```



ARDUINO //

PIN OUT



ESP32 FEATHER // PIN OUT

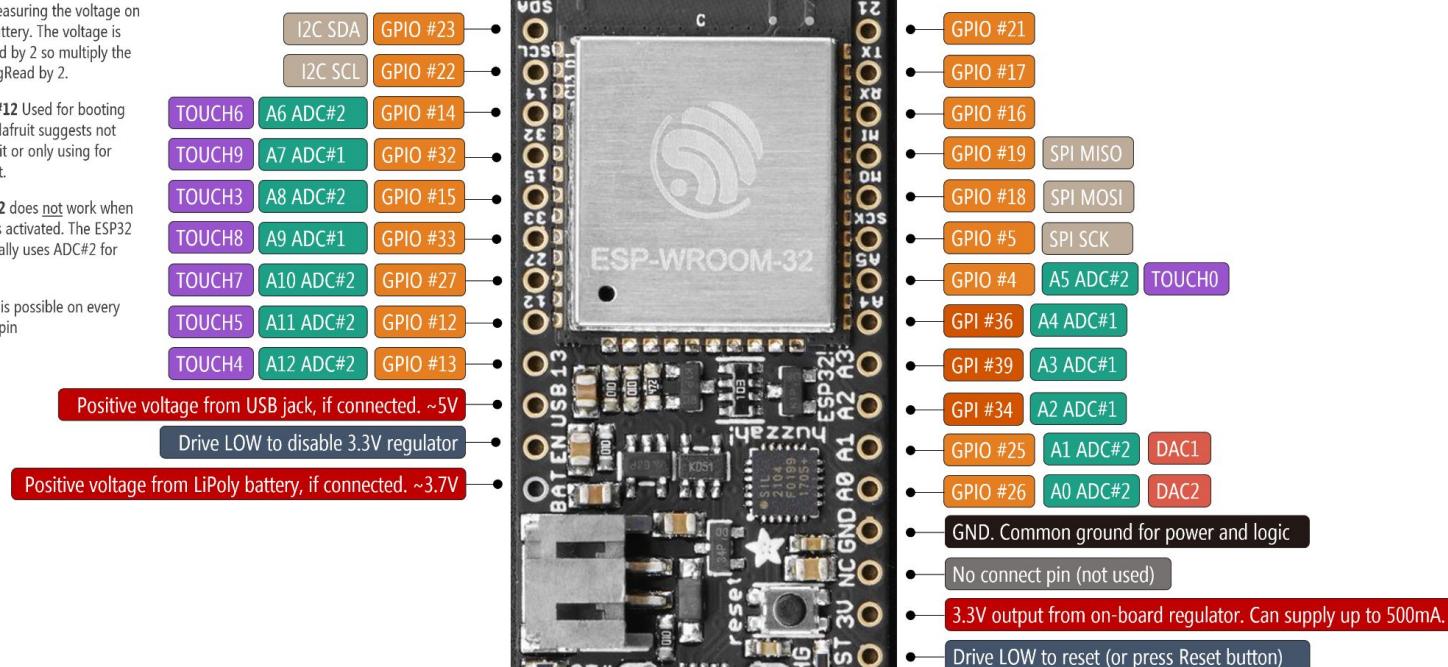
ADAFRUIT HUZZAH32 PIN DIAGRAM

A13 not exposed. It's used for measuring the voltage on the battery. The voltage is divided by 2 so multiply the analogRead by 2.

GPIO#12 Used for booting up. Adafruit suggests not using it or only using for output.

ADC#2 does not work when WiFi is activated. The ESP32 internally uses ADC#2 for WiFi

PWM is possible on every GPIO pin

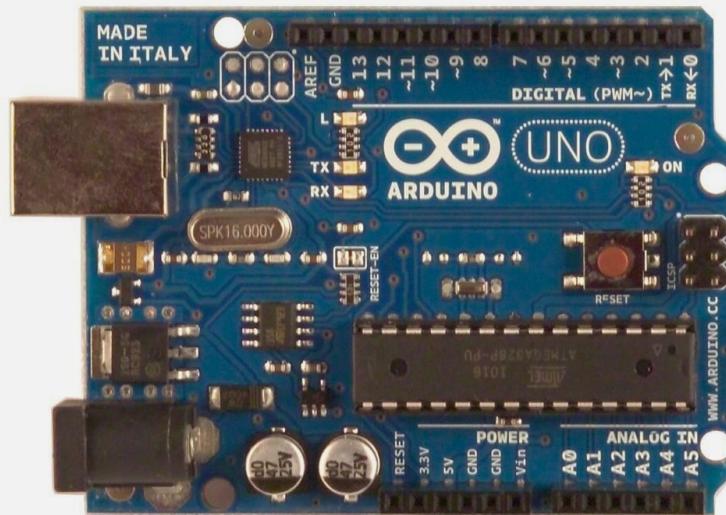


Digital IN / OUT x13

- Analog OUT (PWM) x6
 - Analog IN x6

	INPUTS	OUTPUTS
DIGITAL (on / off)	<code>digitalRead</code> HIGH / LOW 255 / 0	<code>digitalWrite</code> HIGH / LOW 255 / 0
ANALOG (less / more)	<code>analogRead</code> 0 - 1023	<code>analogWrite</code> PWM 0 - 255

ARDUINO // PIN OUT



digitalRead

```
digitalRead(pin);  
LOW/HIGH 0/255
```

SIGNALS // **INPUT**

IN ►

5 volts ——

|
HIGH 255

0 volts ——

0 LOW 0

digitalWrite

```
digitalRead(pin, value);  
LOW/HIGH 0/255
```

OUT ◀

5 volts ——

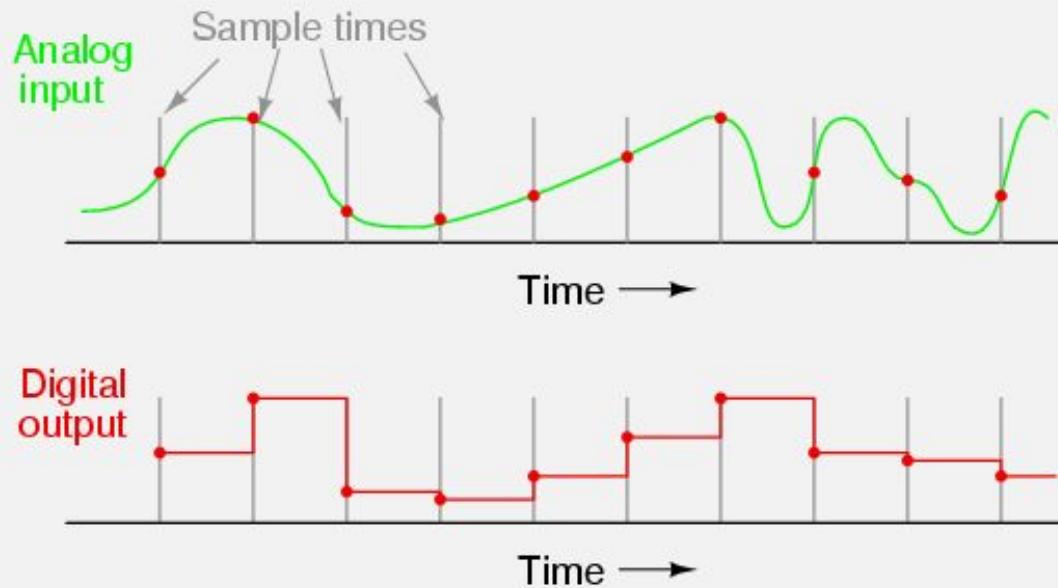
0 volts ——

0 LOW 0

SIGNALS // INPUT



ARDUINO // SIGNALS



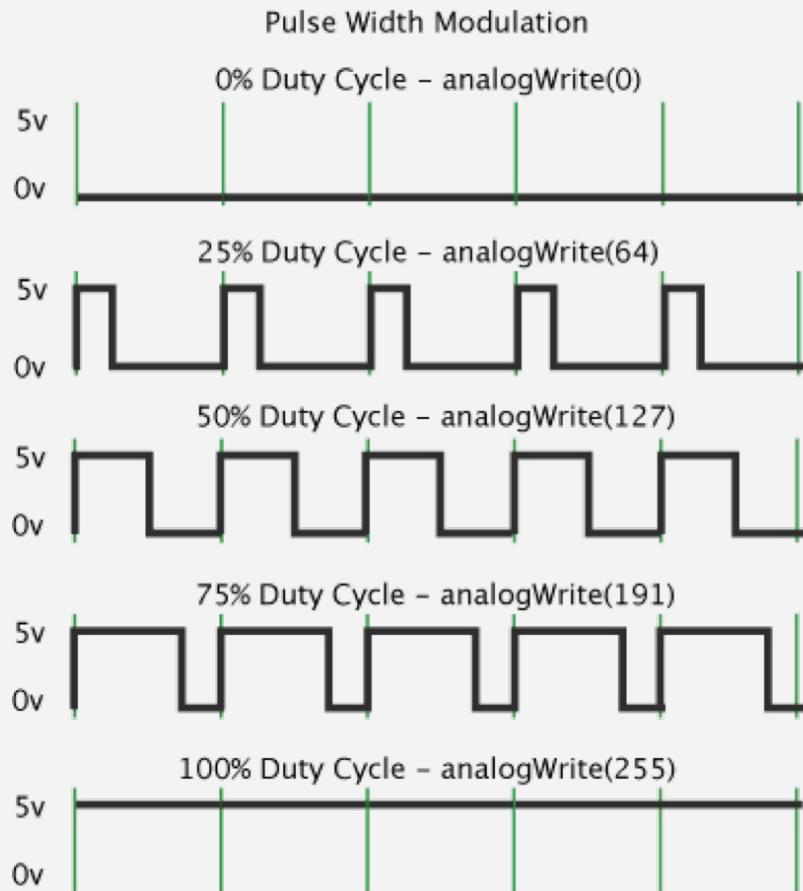
ADC (Analog to Digital Converter)
10 bits

analogRead

```
analogRead(pin);
```

0-1023

ARDUINO // SIGNALS



PWM (Pulse-width modulation)

analogWrite

`analogWrite(pin, value);`

0-255

Ejemplo 1: Blink

ENTORNO DE TRABAJO/ SOFTWARE

Arduino IDE



A screenshot of the Arduino IDE interface. The title bar says "freeServo | Arduino IDE 2.0.0-rc2". The main area shows the code for "freeServo.ino". The code is as follows:

```
freeServo.ino  arduino_secrets.h  thingProps.h
36
37 void loop() {
38   ArduinoCloud.update();
39   if(moveServo){
40     loopServo();
41   }
42 }
43
44 void loopServo(){
45   unsigned long msNow = millis();
46   if(msNow - lastServoMove > SERVO_MOVE_INTERVAL){
47     int direction = garage ? 1 : -1;
48     currentAngle += direction * degreeSteps;
49     if(currentAngle > ANGLE_MAX || currentAngle < ANGLE_MIN){
50       moveServo = false;
51       currentAngle = (direction > 0) ? ANGLE_MAX : ANGLE_MIN;
52     }
53     Serial.println(currentAngle);
54     garageDoorServo.write(currentAngle);
55   }
56 }
57
58 void onGarageChange(){
59   Serial.print("Garage switch ");
60   Serial.println(garage ? "ON" : "OFF");
61   moveServo = true;
62 }
```

The status bar at the bottom shows: Building sketch, Ln 7, Col 1, UTF-8, C++, Arduino NANO 33 IoT on /dev/cu.usbmodem101, 1, 1.

USB Drivers



Arduino Libraries

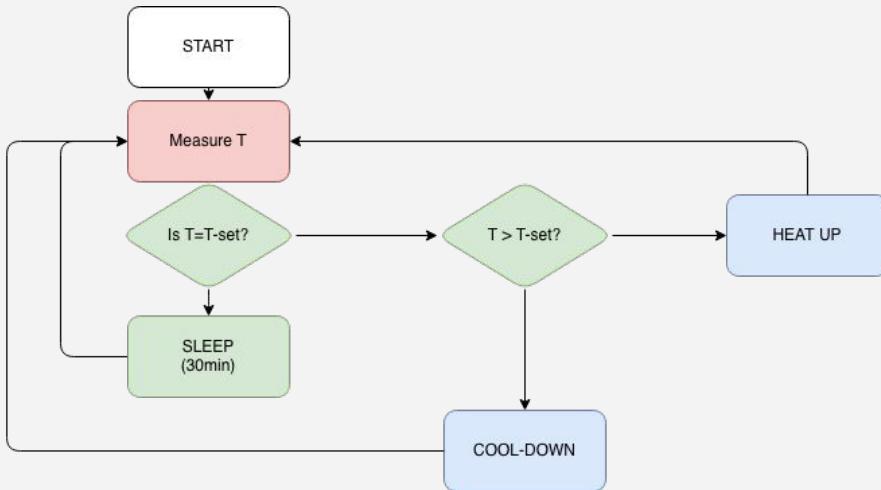


Flow Charts

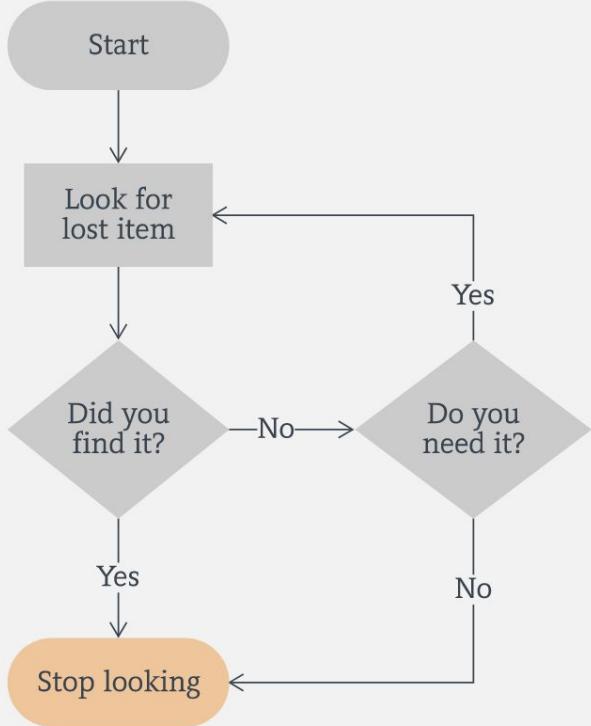
COMPUTING// LOGICS

T-set = 24degC

T = 21degC



Un algoritmo es una secuencia finita de instrucciones bien definidas que se pueden implementar por computadora, generalmente para resolver una clase de problemas o realizar un cálculo. Los algoritmos son siempre inequívocos
[...]

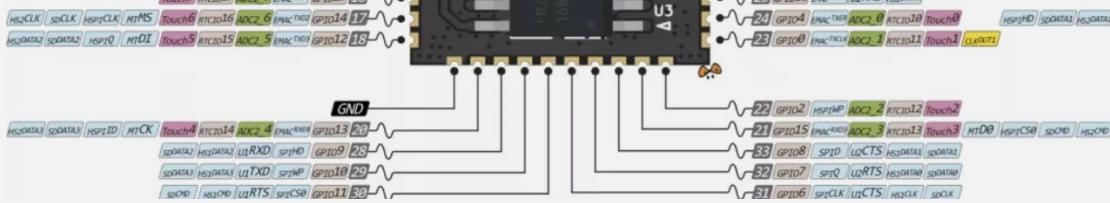


I/O Programming

WROOM32

PINOUT

Power
GND
Serial Pin
Analog Pin
Control
Physical Pin
Port Pin
Touch Pin
DAC Pin
PWM Pin



HARDWARE// PROGRAMMING

El microcontrolador interactuará con el mundo exterior con sus PIN. Los pines pueden servir para recibir información, es decir, entradas, o activar, es decir, salidas:

Podemos usar estos pines para procesar entradas y generar salidas:



Programando logicas de control

Podemos usar declaraciones de flujo de control para administrar cómo nuestro programa reacciona a diferentes entradas y actúa en las salidas.

HARDWARE// PROGRAMMING

```
while (condition) {  
  
    // Do stuff  
}
```

```
if (condition) {  
  
    // Do stuff  
} else {  
  
    // Do other stuff  
}
```

```
for (# iterations) {  
    //Do stuff  
}
```

Arduino programming



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0". The toolbar includes icons for file operations, upload, and preferences. A dropdown menu is open, showing "Blink" as the selected option. The code editor displays the "Blink" sketch, which blinks an LED connected to pin 13. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);      // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(13, LOW);       // set the LED off
  delay(1000);                // wait for a second
}
```

The status bar at the bottom indicates "Arduino Uno on /dev/tty.usbmodem1d11".

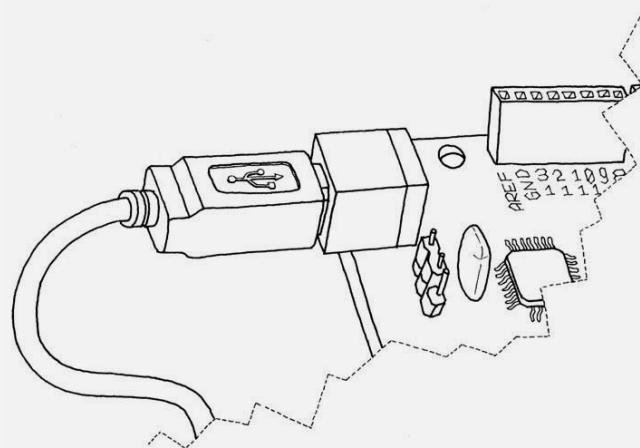
ARDUINO// PROGRAMMING

Instalación

Deberá instalar el IDE de Arduino. Después de eso, necesitamos definir la placa que vamos a utilizar.

El primer ejemplo de programación se suele utilizar un LED ON/OFF y se le llama

“hello, world!”



The screenshot shows the Arduino IDE's Help menu open. The menu items are:

- Getting Started
- Environment
- Troubleshooting
- Reference** (highlighted in blue)
- Find in Reference
- Frequently Asked Questions
- Visit Arduino.cc

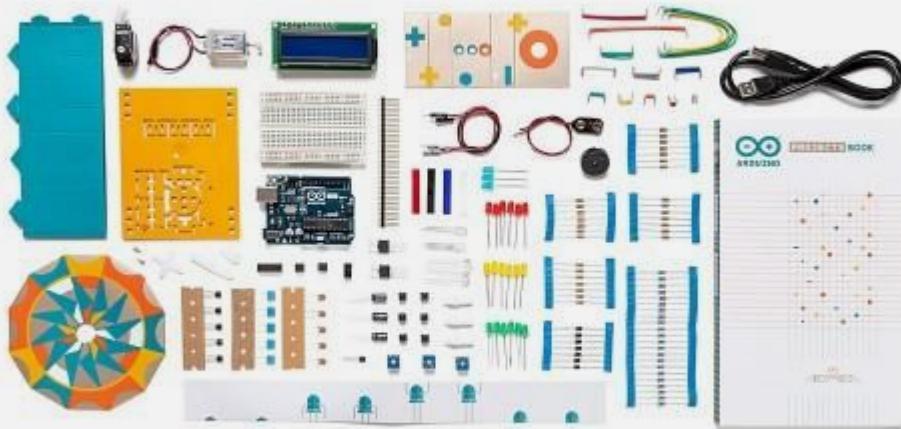
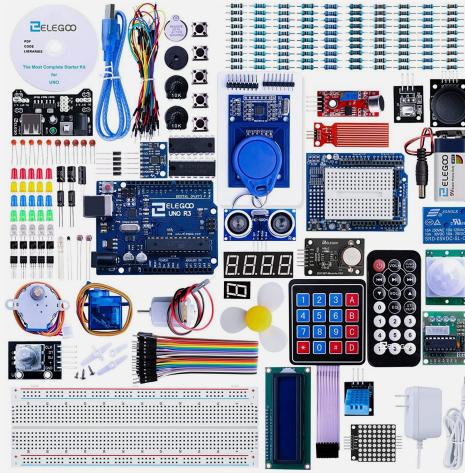
ARDUINO// PROGRAMMING

Uno de los puntos fuerte de utilizar el ecosistema de arduino es la cantidad de información disponible.

Siempre que tengamos dudas o no sepamos utilizar algo podemos recurrir a AYUDA- REFERENCIA

Donde se explica que hace cada función

<http://arduino.cc/en/Reference/HomePage>



ARDUINO// KITS

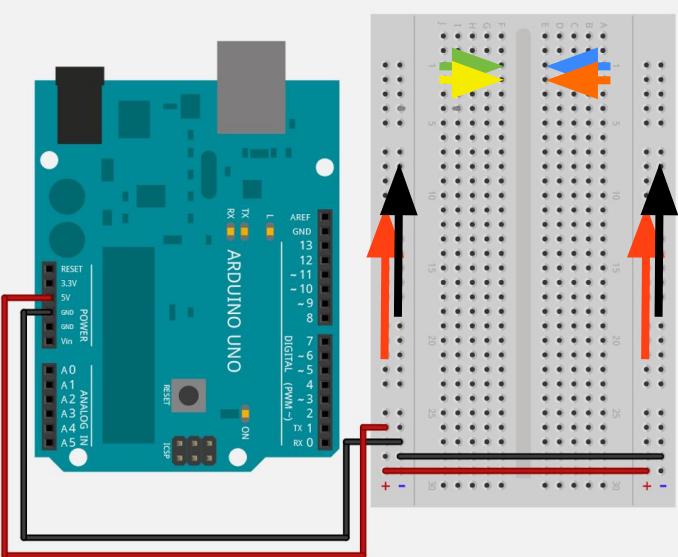
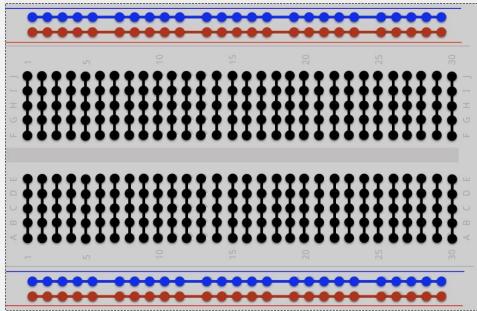
http://www.seeedstudio.com/wiki/Arduino_Sidekick_Basic_Kit

<http://www.cooking-hacks.com/arduino-starter-kit>

<https://www.arduino.cc/>

BREADBOARD

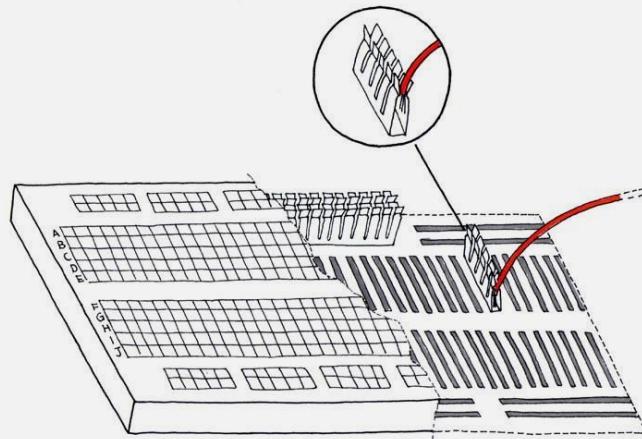
TABLERO// PROTOTIPADO



The Breadboard (aka Protoboard)

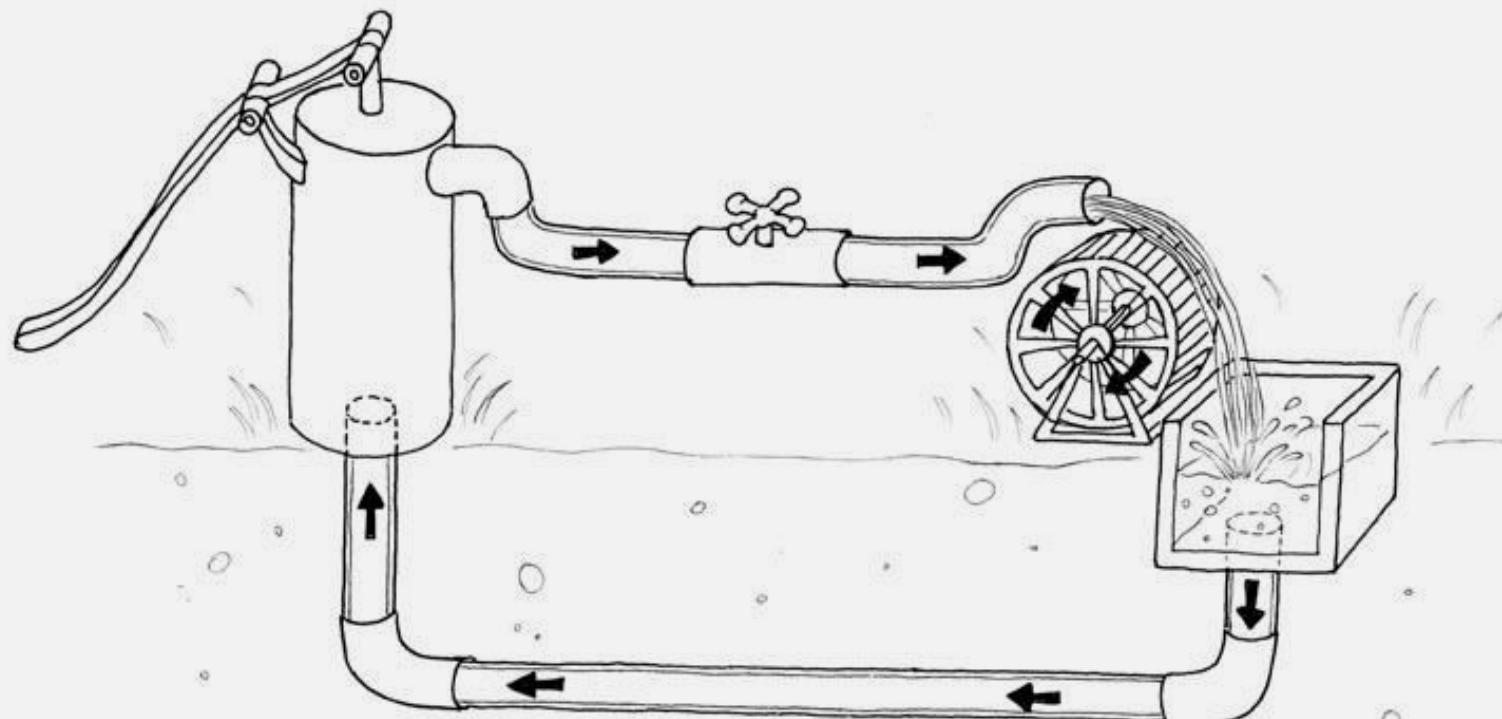
Una placa de pruebas es una herramienta de creación de prototipos que utilizamos para hacer que los circuitos sean rápidos y se repitan fácilmente:

<http://www.protostack.com/blog/2011/09/8-bread-board-hacks/>



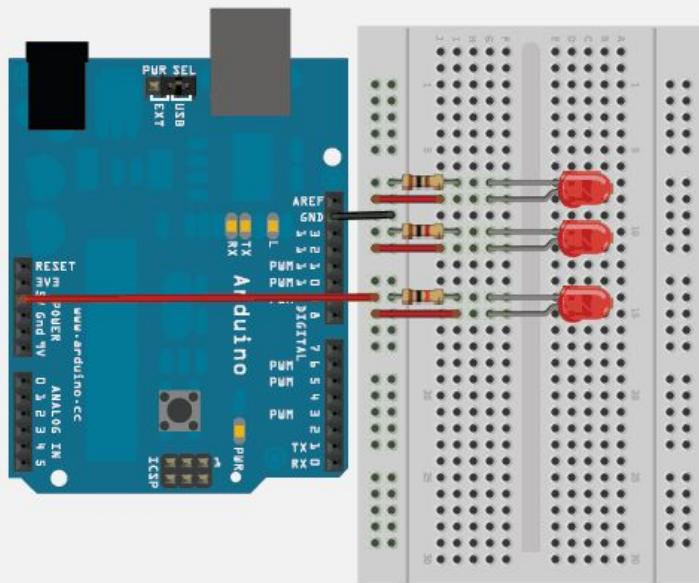
RESISTENCIAS

HARDWARE// RESISTORS



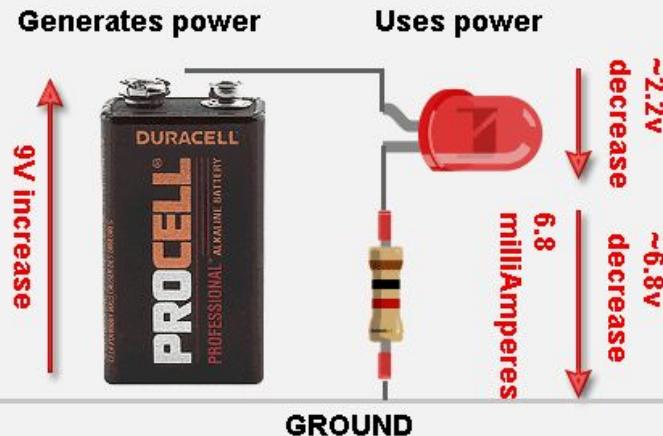
¿Por qué resistencias?

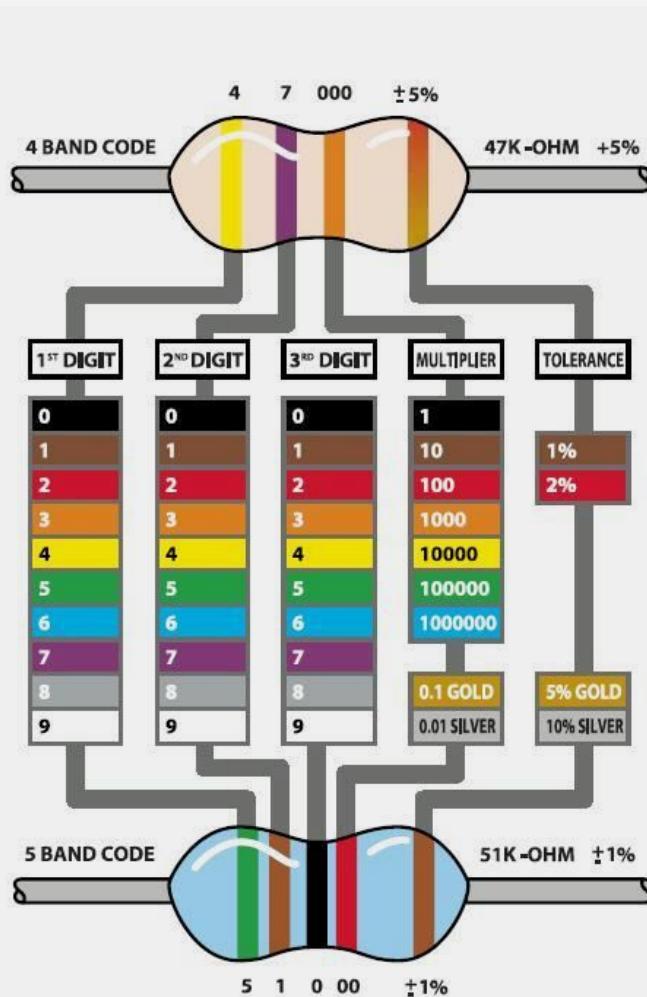
La resistencia es tu amigo.



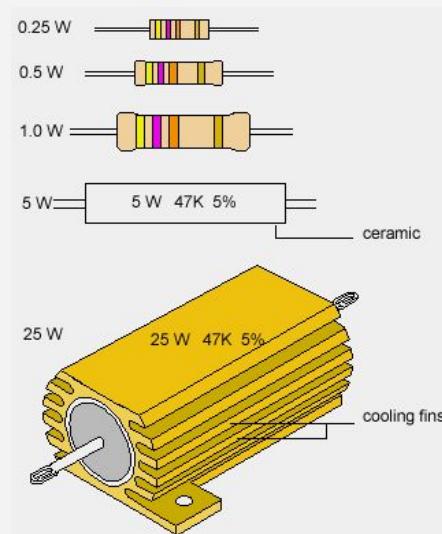
HARDWARE// RESISTORS

5V
Max 40mA x PIN





HARDWARE// RESISTORS



<http://www.dannyq.com/examples/res2/resistor.htm>
<http://www.csqnetwork.com/resistcolcalc.html>
http://www.hebeiltd.com.cn/?p=zz.led.resistor.calculat_orext

ARDUINO

“HELLO WORLD”

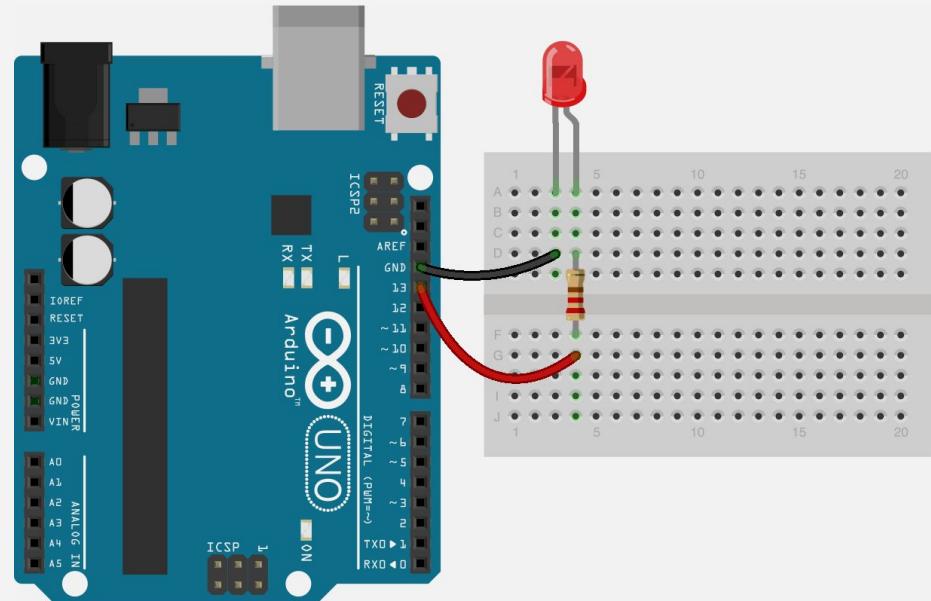
Tres cosas para recordar

1. ¿INPUT/OUTPUT? pinMode ()
2. Pin de encendido / apagado con digitalWrite ()
3. Leer el estado del botón con digitalRead ()

```
void setup() {  
    // Set pin 13 as output  
    pinMode(13, OUTPUT);  
}  
  
// Loop runs forever  
void loop() {  
    // Voltage up in pin 13  
    digitalWrite(13, HIGH);  
    // Wait 200ms with pin 12 HIGH  
    delay(200);  
    // Voltage down in pin 13  
    digitalWrite(13, LOW);  
    // Wait 200ms with pin 13 LOW  
    delay(200);  
}
```

ARDUINO //

101



```

// Our variable for checking if it's pressed or not
bool pressed = false;

// the setup function runs once when you press reset or power the board
void setup() {

    pinMode( 8 , OUTPUT);
    pinMode( 4 , INPUT);
    // Turn the LED off
    digitalWrite( 8 , LOW);
}

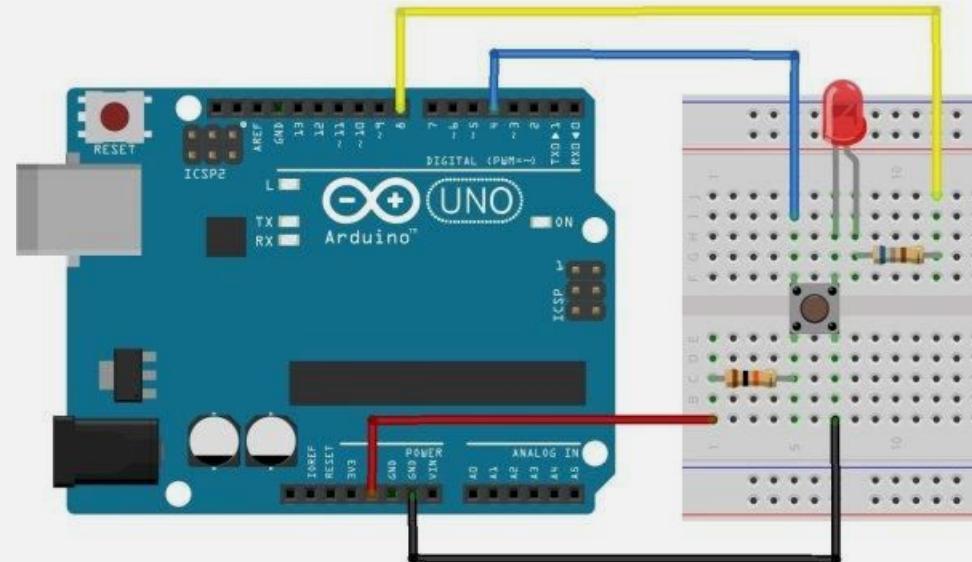
// Loop runs forever
void loop() {
    // Read the pin
    if (digitalRead( 4 )) {
        // pressed!
        pressed = false;
    } else {
        // not pressed!
        pressed = true;
    }

    if (pressed) {
        // Turn it on
        digitalWrite( 8 , HIGH);
    } else {
        // Turn it off
        digitalWrite( 8 , LOW);
    }
}

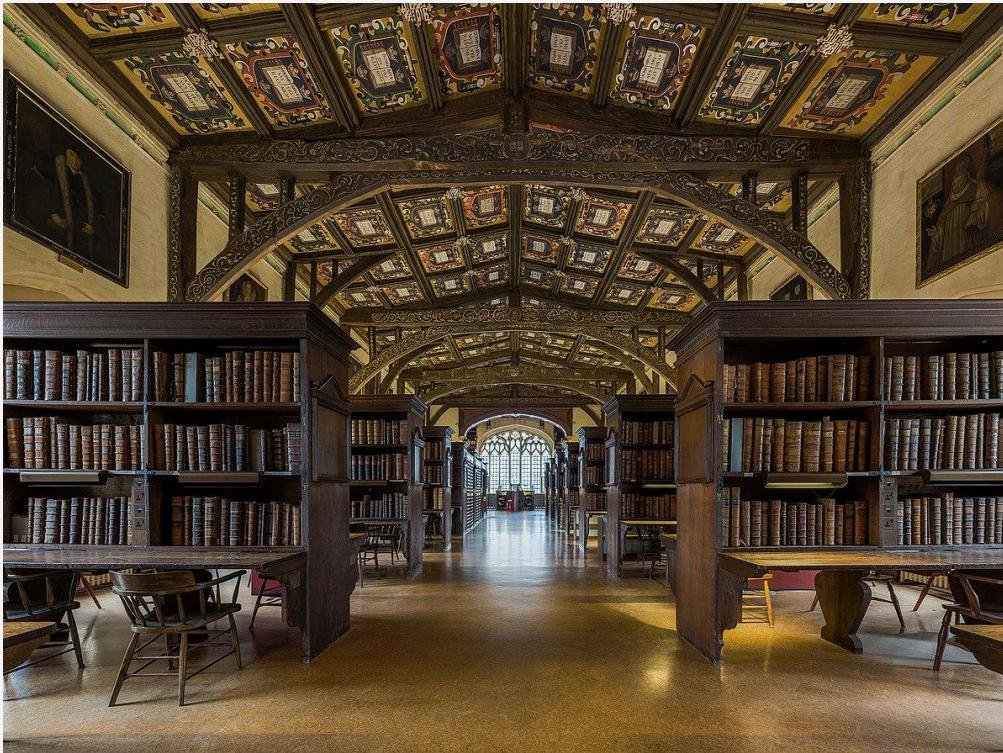
```

ARDUINO //

101



LIBRERIAS



ARDUINO //

101

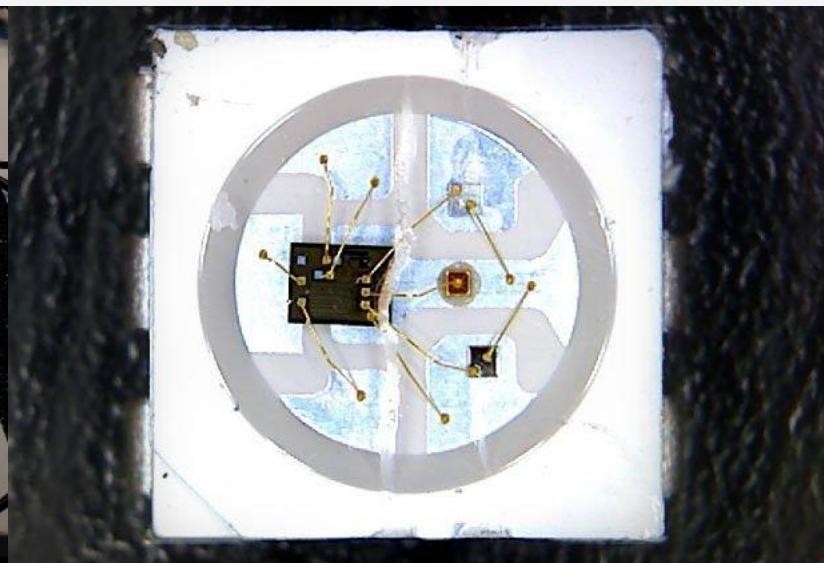
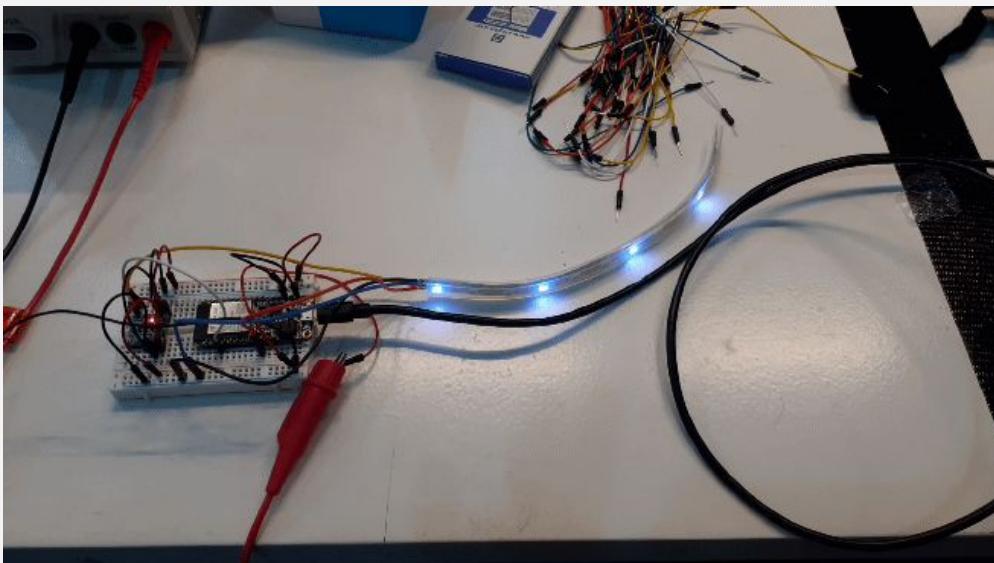
Las bibliotecas son conjuntos de herramientas que alguien ha creado para facilitar ciertas tareas, por ejemplo, leer pines en un microcontrolador o controlar los LED.

Para usar una tira de LED “NEOPIXEL”, necesitaremos instalar una biblioteca.

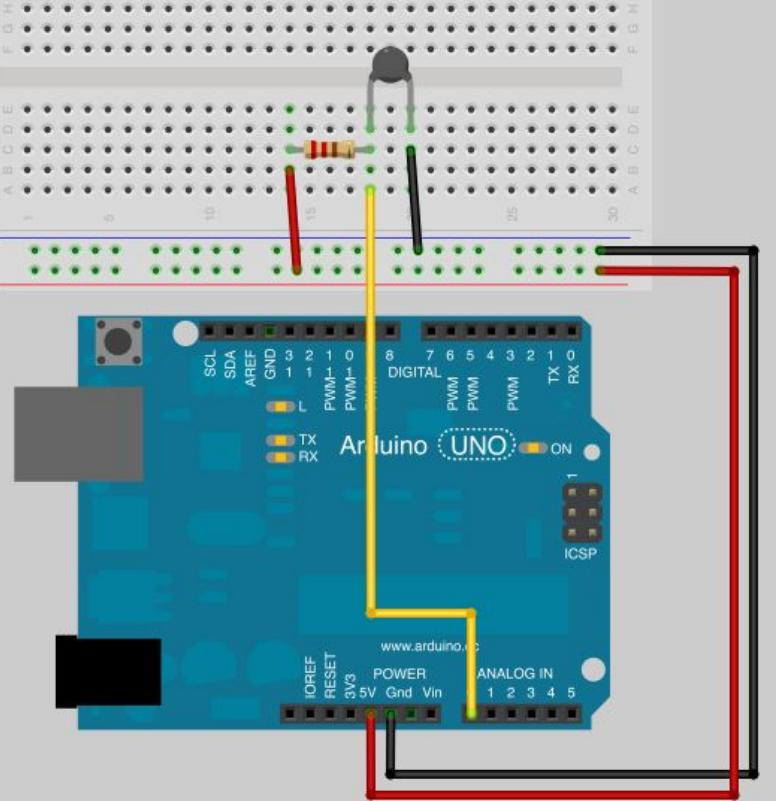
!!!Instalación de la biblioteca"

Siga esta [guía](#) para instalar la biblioteca [FastLED](#)

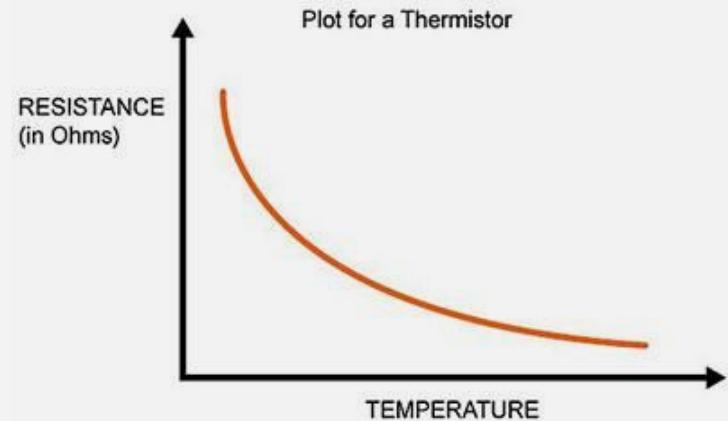
OUTPUTS// NEOPIXEL



INPUTS



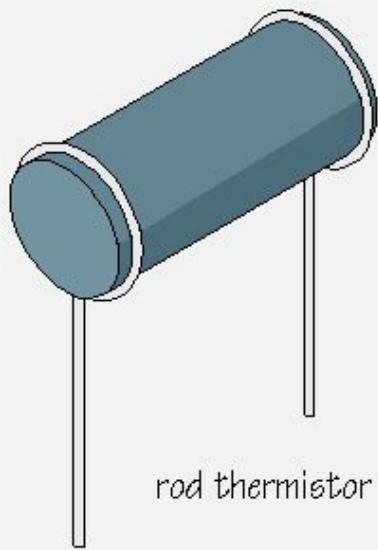
INPUTS// THERMISTOR



<http://learn.adafruit.com/thermistor>

<http://pastie.org/8467723>

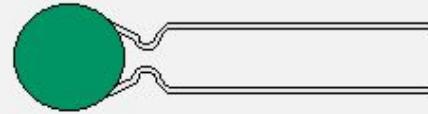
INPUTS// RESISTANCE



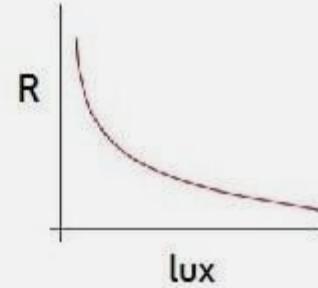
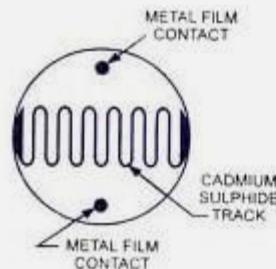
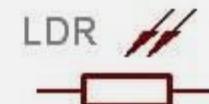
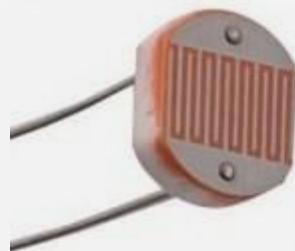
rod thermistor

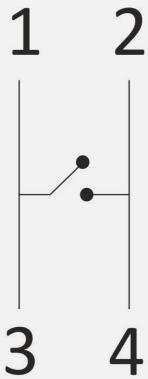


bead thermistor

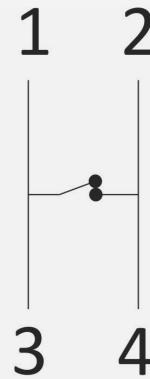


disc thermistor



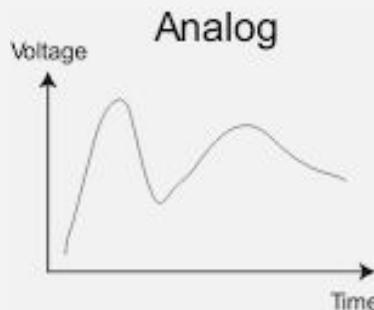


NORMAL



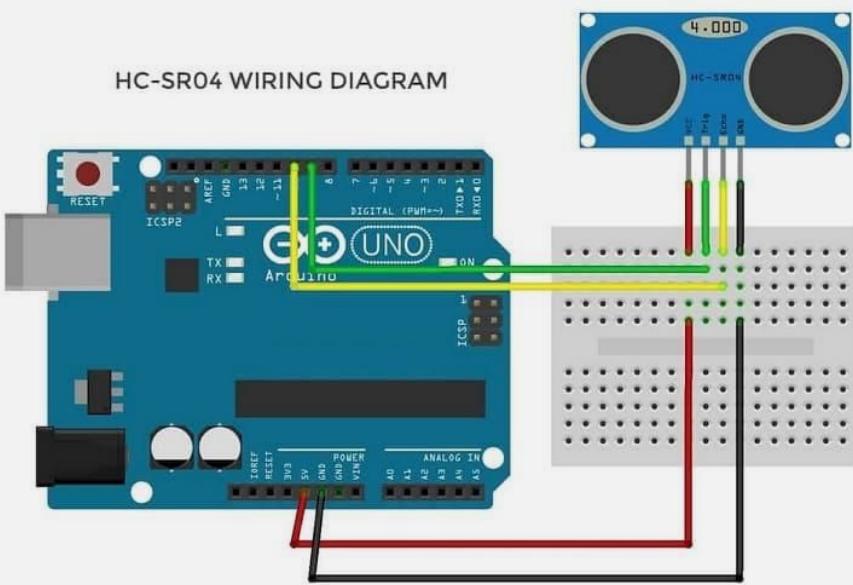
BUTTON
PRESSED

INPUTS// BUTTON

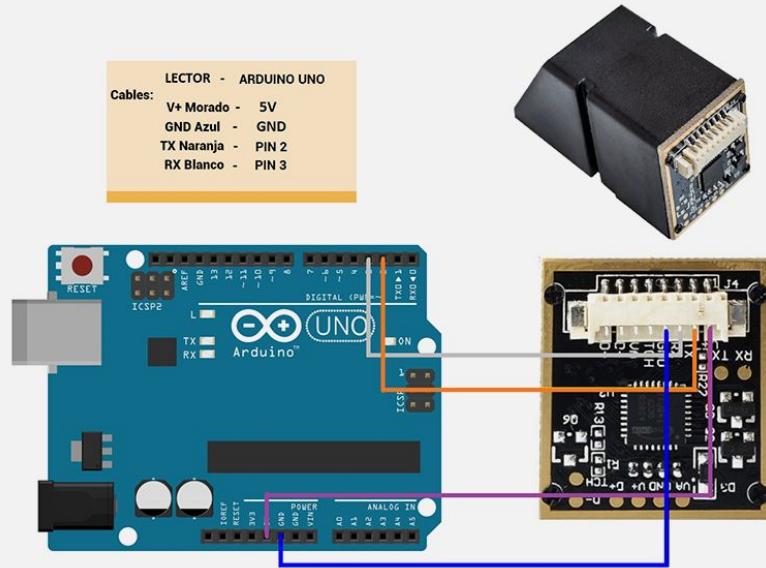


INPUTS// DIGITALS

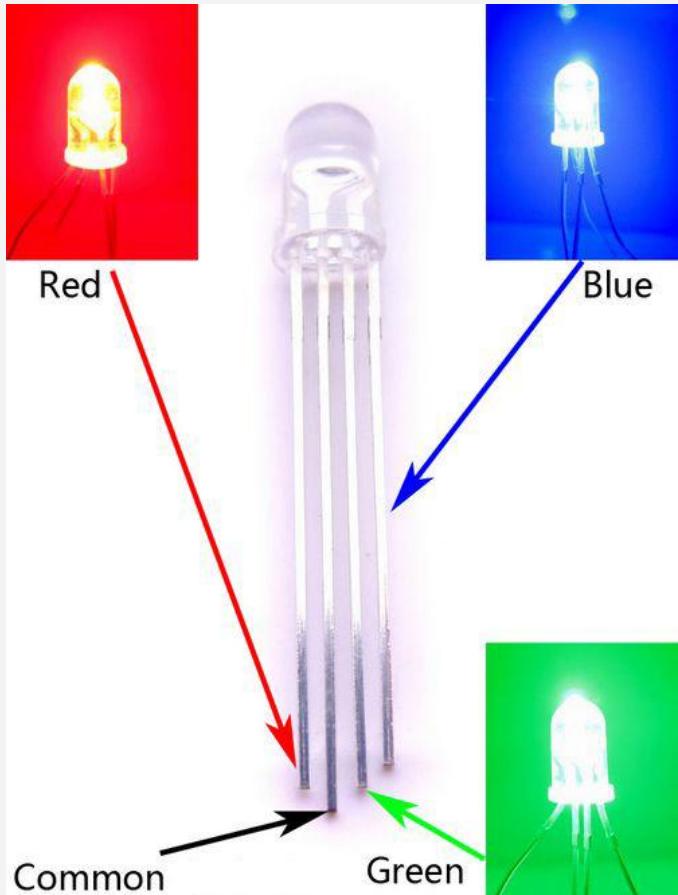
HC-SR04 WIRING DIAGRAM



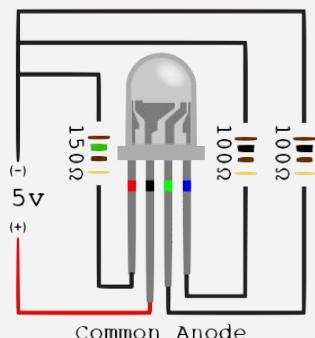
LECTOR - ARDUINO UNO
Cables:
V+ Morado - 5V
GND Azul - GND
TX Naranja - PIN 2
RX Blanco - PIN 3



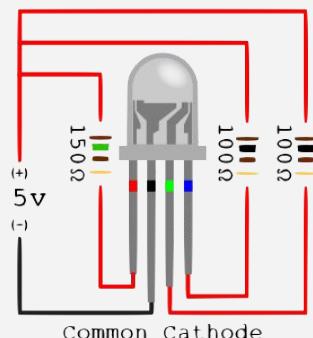
OUTPUTS



OUTPUTS// RGB LEDs

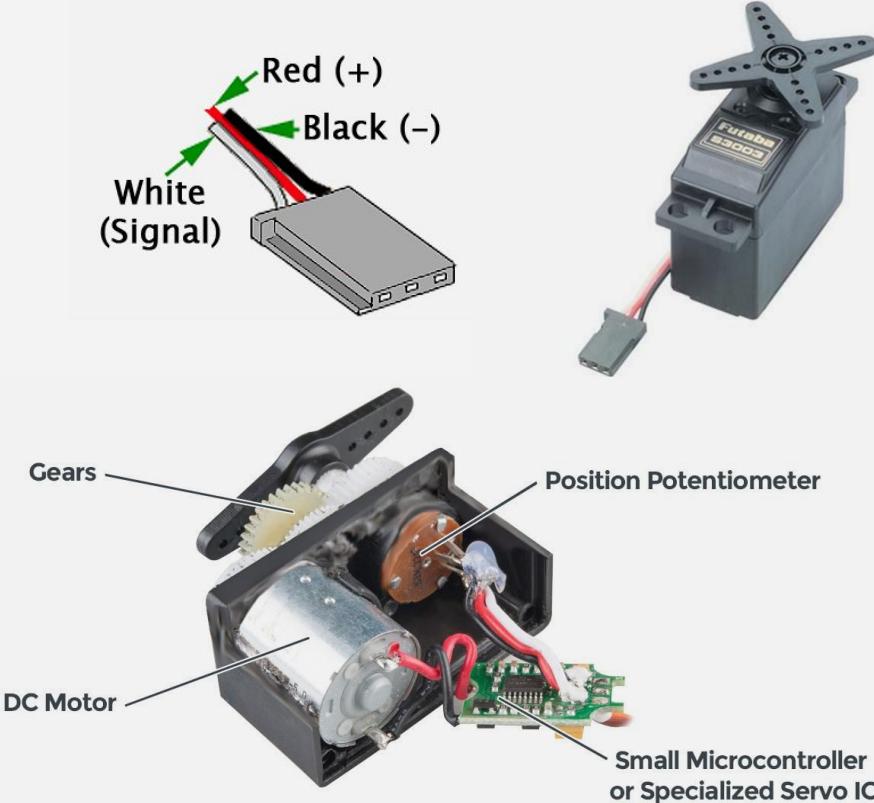


255-value



0-255

OUTPUTS // SERVOMOTOR

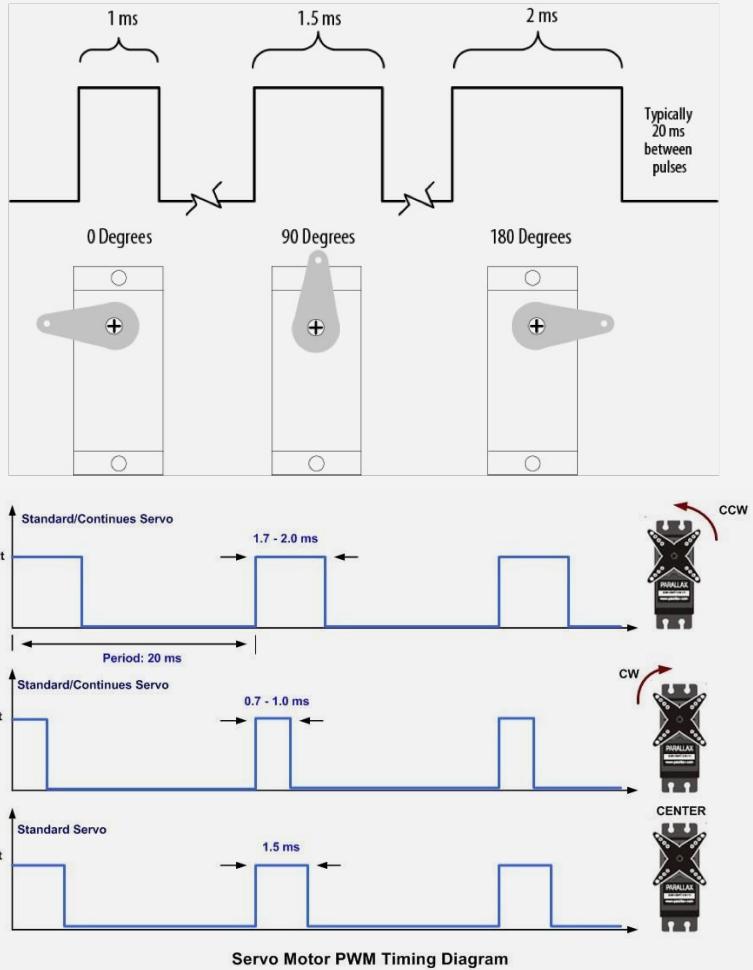


```
#include <Servo.h>
```

```
Servo myservo;
```

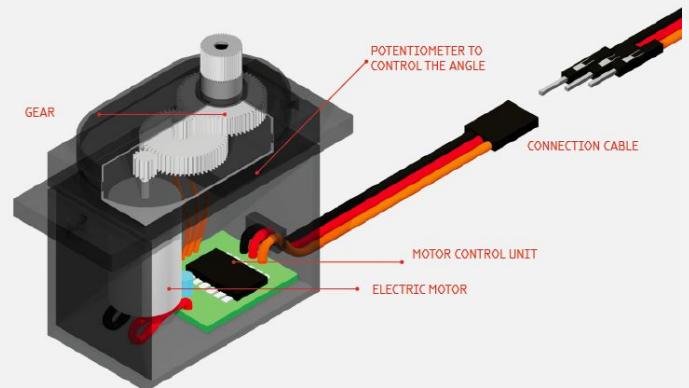
```
void setup() {  
    myservo.attach(9); // attaches the servo on pin 9  
    to the servo object  
}
```

```
void loop() {  
  
    myservo.write(pos); // tell servo to go to  
    position in variable 'pos'  
}
```

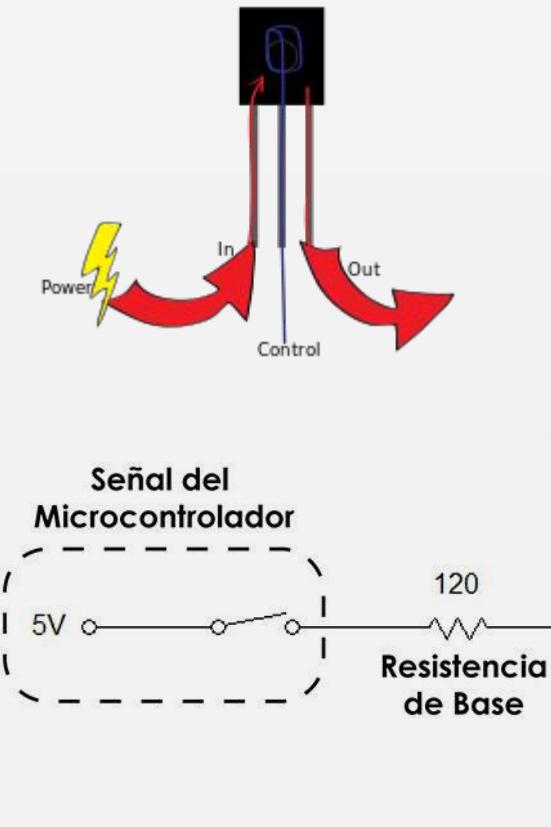


OUTPUTS // SERVOMOTOR

Close loop control

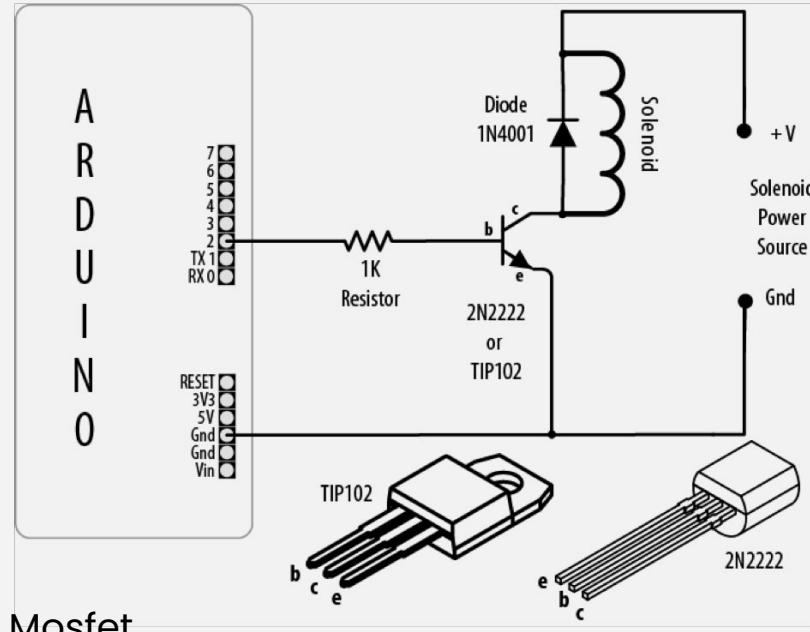


OUTPUTS // MOSFETS



Un mosfet (Metal Oxide Semiconductor Field Effect Transistor) tambien conocido Insulated Gate Field Effect Transistor (IGFET) consiste en una fina capa de oxido de silicoma entre puerta y canal.

Funciona como un interruptor digital accionado por una variación de voltaje. Permitiendo activaciones muy rápidas comparados con un relé que se basa en componentes mecánicos móviles.

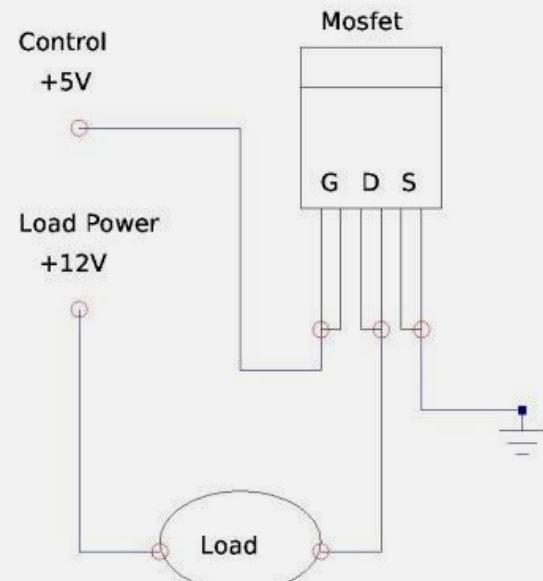


Mosfet

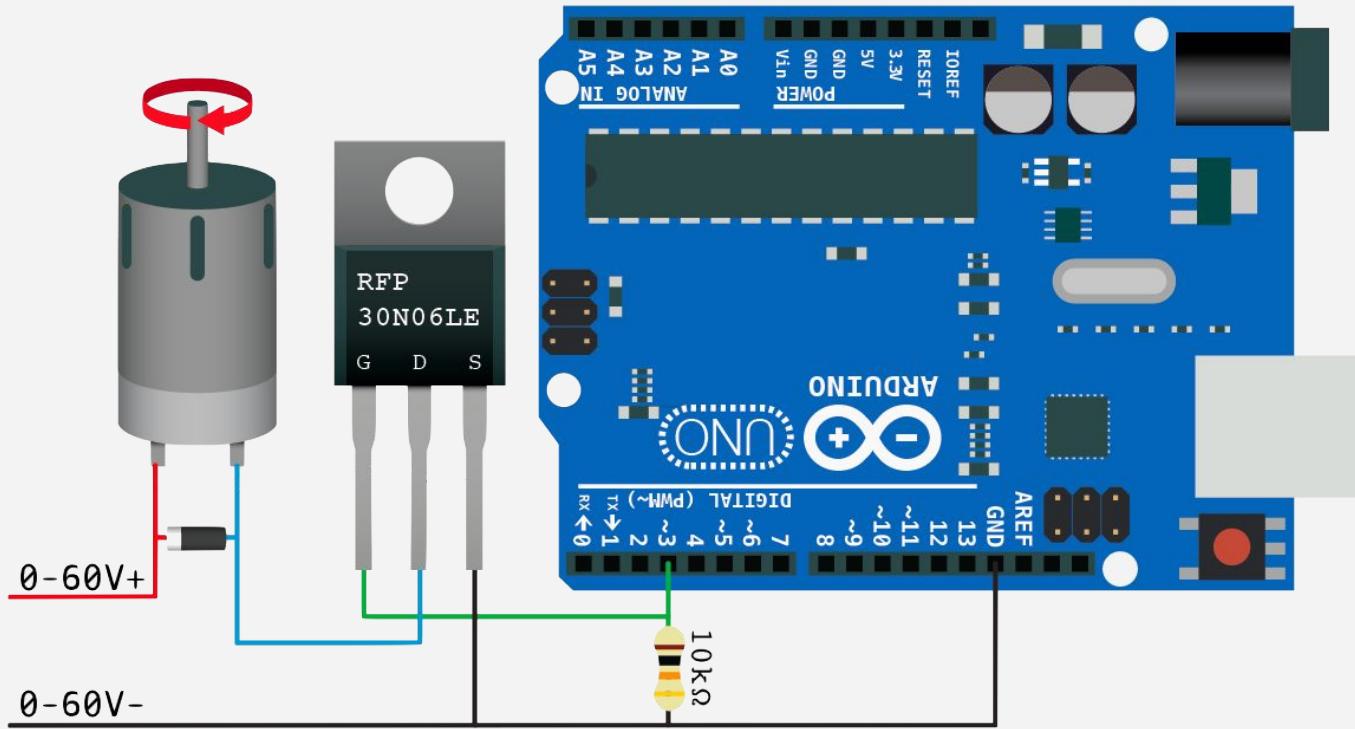
> 40mA or >
5V

on/off or PWM
(digitalWrite / analogWrite)

OUTPUTS// MOSFETS

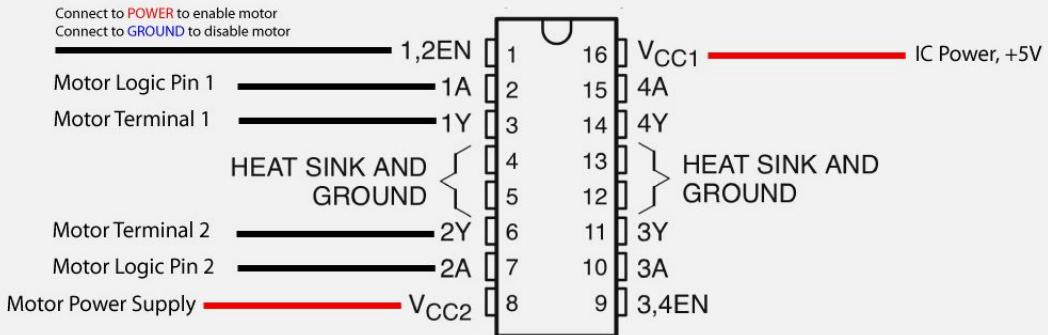


OUTPUT // MOSFET



OUTPUTS// H-BRIDGE

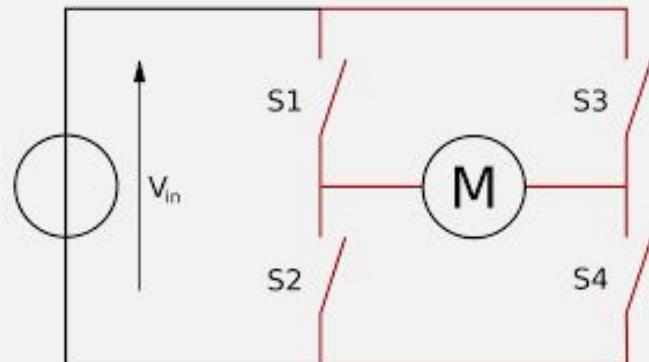
L293NE or SN754410



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

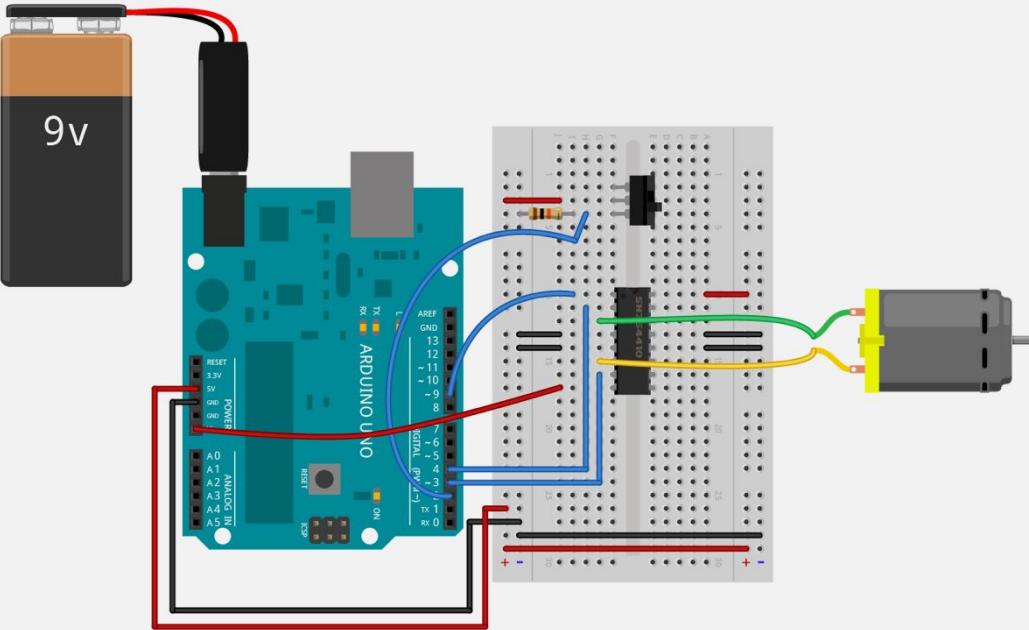
L = low, H = high, X = don't care

> 40mA or > 5V + DC motor control (+/-/+)

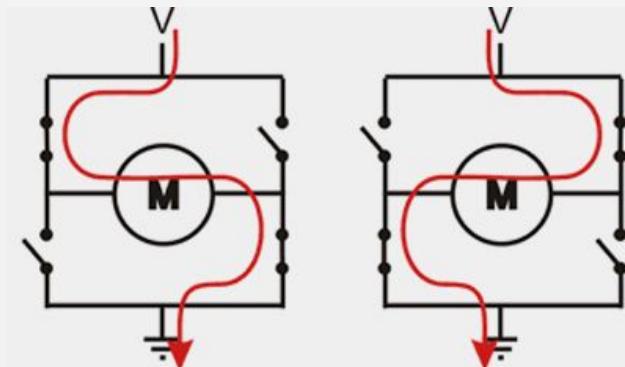


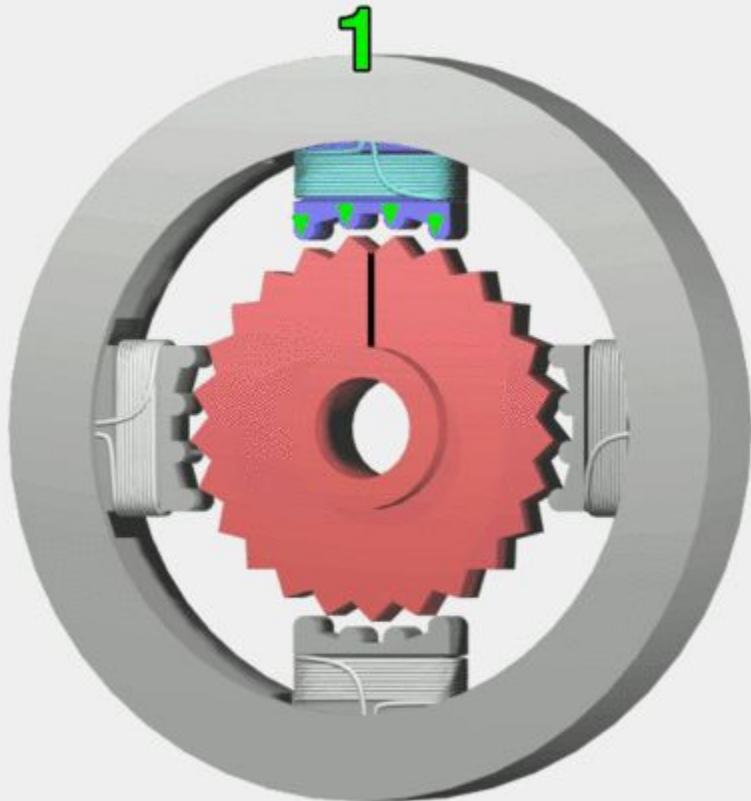
<http://learn.adafruit.com/adafruit-arduino-lesson-15-dc-motor-reversing/lm293d>

OUTPUTS// H-BRIDGE

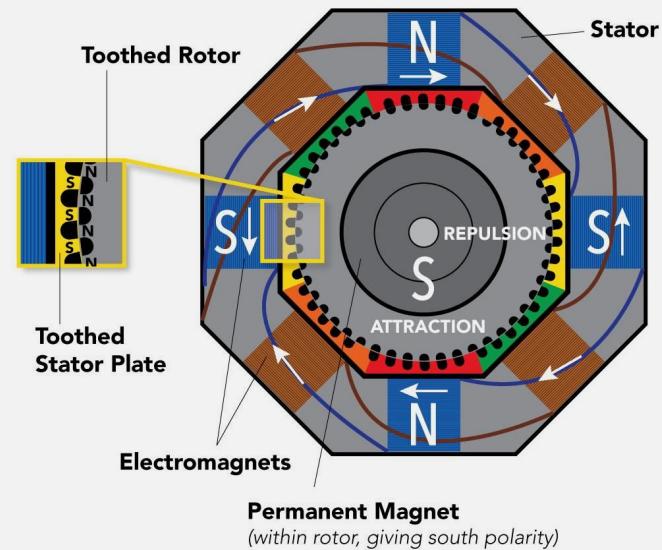


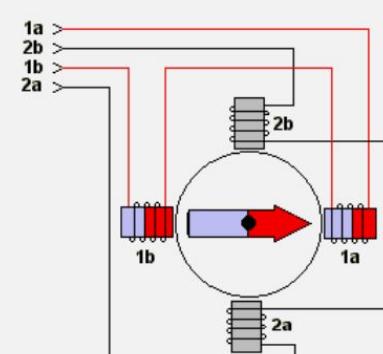
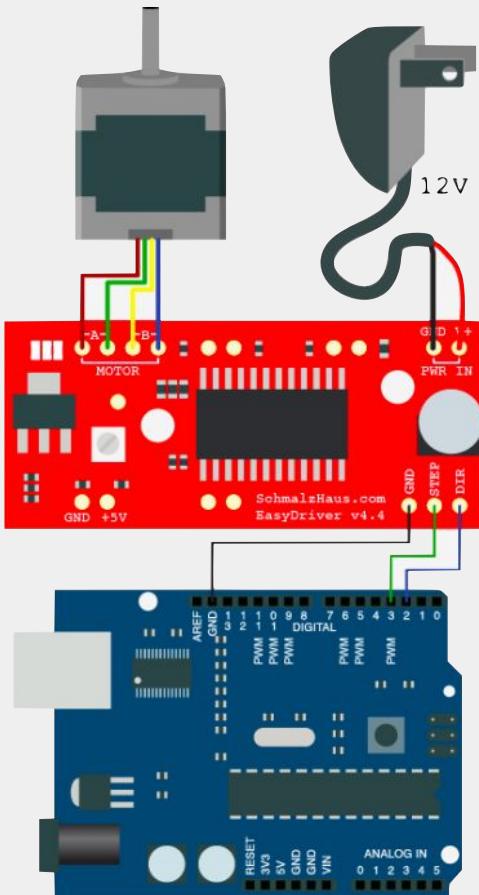
on/off or PWM
(digitalWrite
/analogWrite)





OUTPUTS// STEPPERS



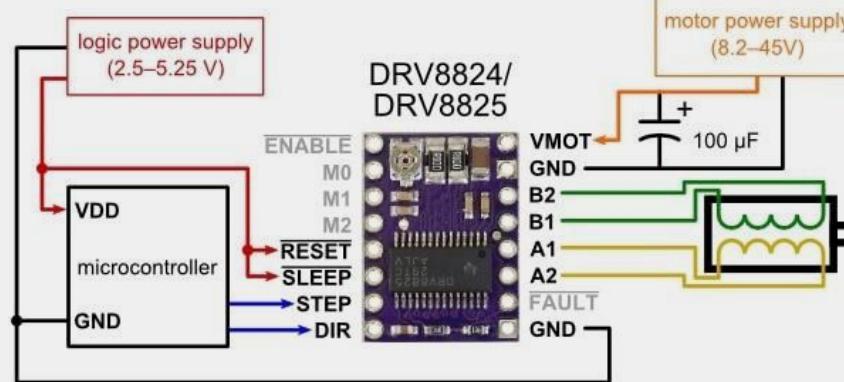


Conceptual Model of Bipolar Stepper Motor

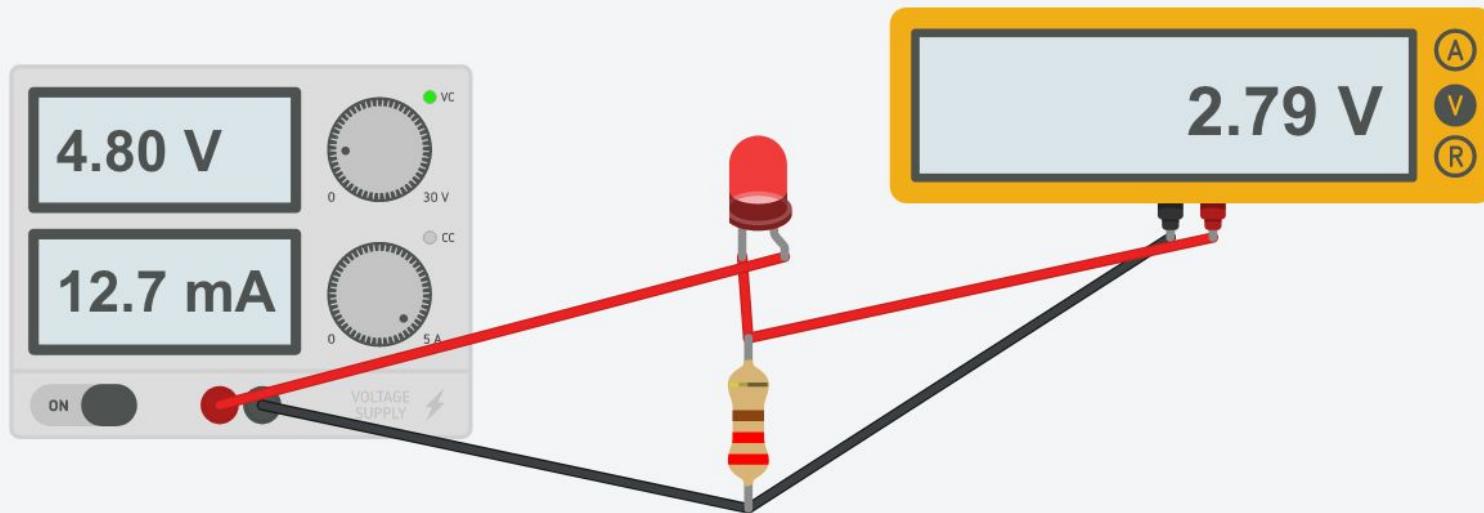
OUTPUTS// STEPPERS

<http://bildr.org/2011/06/easydriver/>
Libraries

- <https://www.arduino.cc/en/Reference/Stepper>
- <http://www.airspayce.com/mikem/arduino/AccelStepper/>

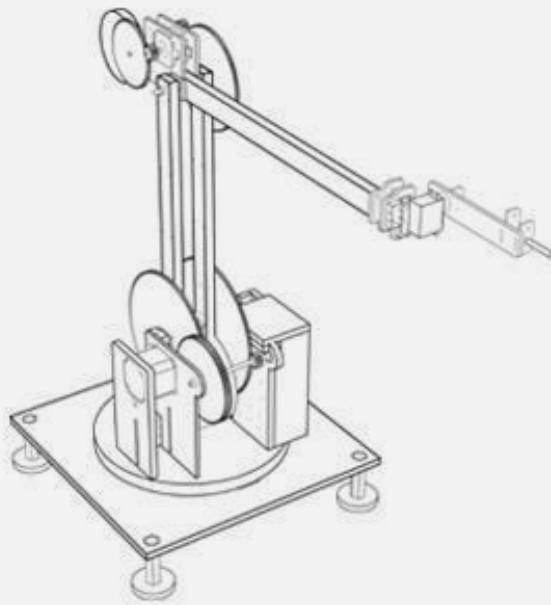
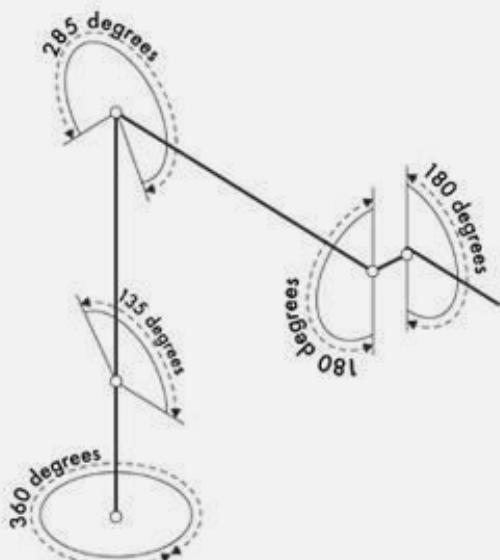


SIMULACIÓN// TINKERCAD

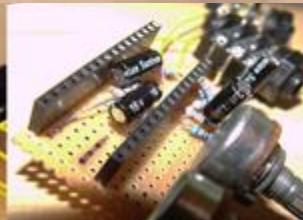
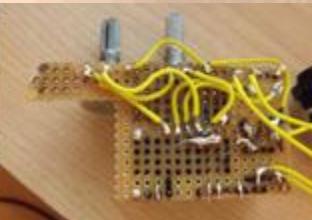
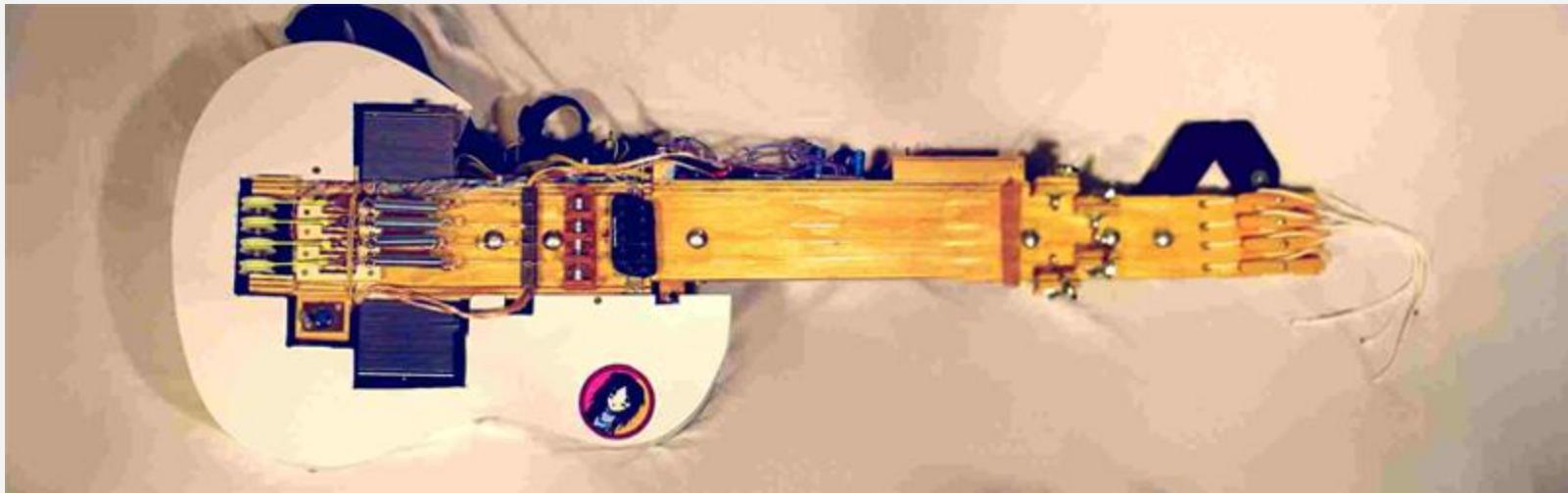


EJEMPLOS

Arduino// Ejemplos



Arduino// Ejemplos





Arduino// Ejemplos

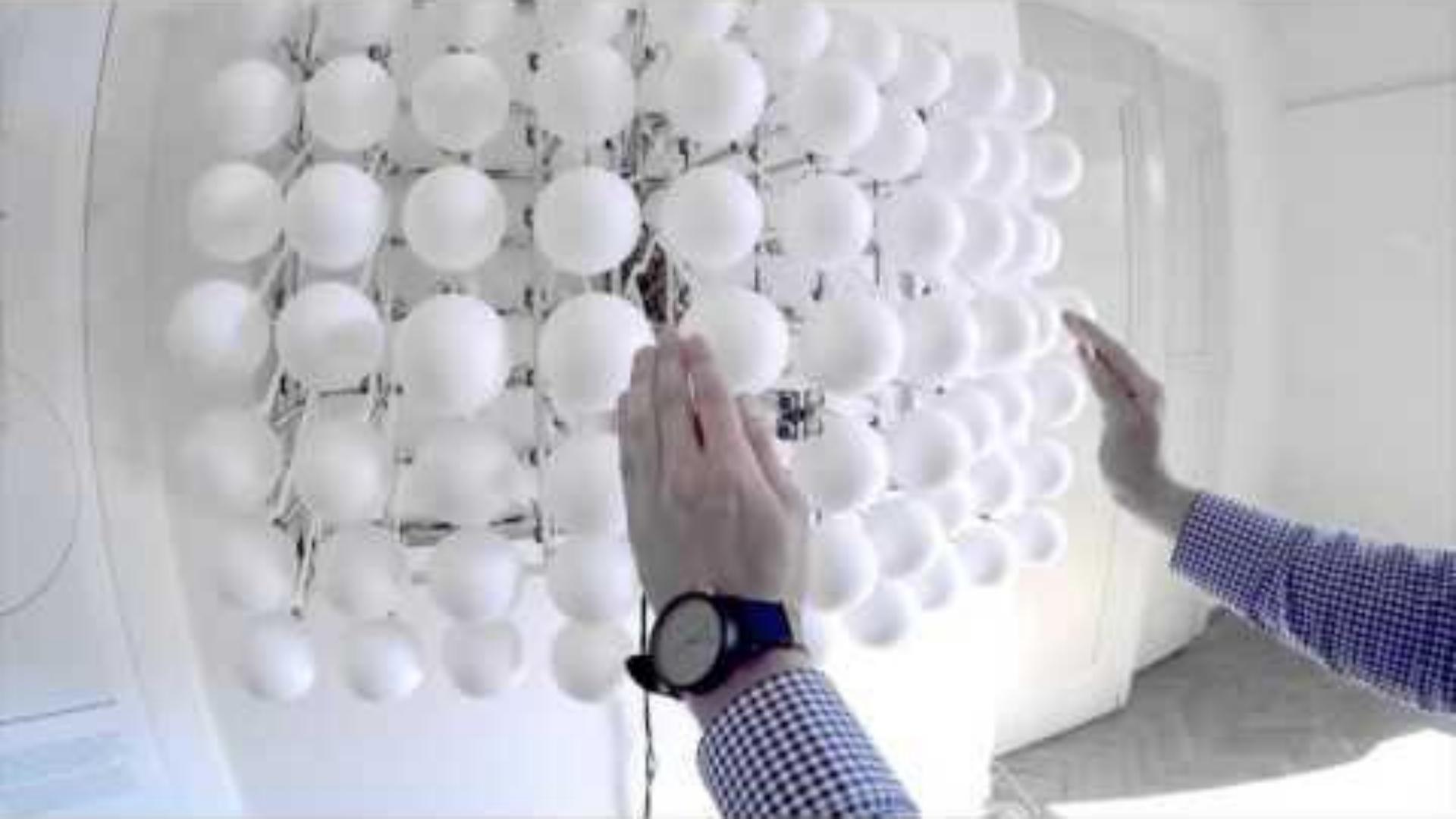


Arduino// Ejemplos



Arduino// Ejemplos





SUPPLIERS



cubie
tech

CubieBoard.org
2012-09-09

USB2

USB1

DATA CARD

USB2

USB1

ARDUINO //

STORES

Shops in Barcelona

Onda Radio <http://www.onдарadio.es/productos/buscar.aspx?v=A000066>

Location <https://plus.google.com/100101416647173159883/about?hl=en>

Diotronic http://www.diotronic.com/raspberry-pi-arduino/arduino/placas_p_1080.aspx

Location <https://plus.google.com/117114665537047889067/about?hl=en>

Ro-Botica <http://ro-botica.com/>

Location <https://plus.google.com/110636064159550584786/about?hl=en>

On-Line Shops near Barcelona (2-4 day shipping)

Cooking Hacks <http://www.cooking-hacks.com> (Zaragoza)

SnootLab <http://snootlab.com/> (Toulouse)

BricoGeek <http://www.bricogeek.com/shop/> (Carballo, Galicia)

Reflexiona <http://www.reflexiona.biz/> (Bilbao)

Electan <http://www.electan.com/> (Alicante)

International

SparkFun <http://www.sparkfun.com/>

Adafruit <http://adafruit.com/>

Liquid Ware <http://www.liquidware.com/shop>

Itead Studio <http://iteadstudio.com/store/>

Seeed Studio <http://seeedstudio.com/>

Modern Device <http://shop.moderndevice.com/>

TUTORIALS

ARDUINO //

STORES

* [Teoría de la información]

(<https://www.youtube.com/watch?v=p0ASFxKS9sg&list=PLP6PHJ8SLR6D4ytpHhZBdyIPNcazU5m7o>)

* [Más información sobre teoría y codificación (para aquellos que disfrutaron de los videos anteriores)]

(https://www.youtube.com/playlist?list=PLzH6n4zXuckpKAj1_88VS-8Z6yn9zX_P6)

* [Cómo piensan las computadoras]

(<https://www.youtube.com/watch?v=dNRDvLACg5Q>)

* [Por qué usamos binario]

(<https://www.youtube.com/watch?v=thrx3SBEpL8>)

* [Cómo funcionan las computadoras]

(<https://www.youtube.com/watch?v=nN9wNvEnn-Q>)

* [Cómo funciona la memoria de la computadora en el interior]

(<https://www.youtube.com/watch?v=XETZoRYdtkw&t=6s>)

* [Colossus, la primera computadora electrónica]

(<https://www.youtube.com/watch?v=knXWMjIA59c>)

* [Cómo diseñaron las computadoras que llevaron a los astronautas a la luna]

(https://www.youtube.com/watch?v=xQ1O0XR_cA0)

* [UNIX o lo que hace un sistema operativo]

(<https://www.youtube.com/watch?v=tc4ROCJYbm0>)

* [Observa cómo se construyen y funcionan las cosas para aprender a diseñar]

(<https://www.youtube.com/playlist?list=PLvOlSehNtuHsy89OdSxBajult8e5srVLA>)

* [Una introducción a Arduino para futuras asignaciones]

(https://www.youtube.com/watch?v=_h1m6R9YW8c)

* [La vida secreta de las máquinas]

(https://www.youtube.com/watch?v=KDpNQQqdSh8&list=PLByTa5duolYRtq45Cz_GmtzfWJyA4bik)



COMPUTING// REF



learn.adafruit.com



learn.sparkfun.com

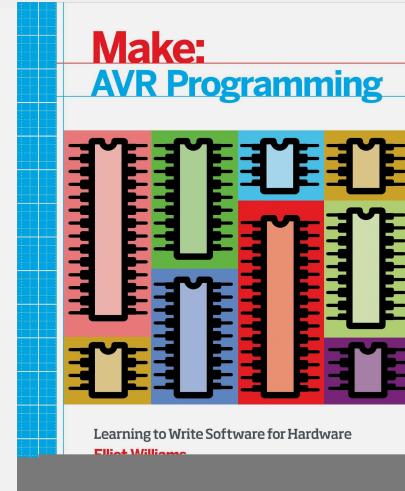
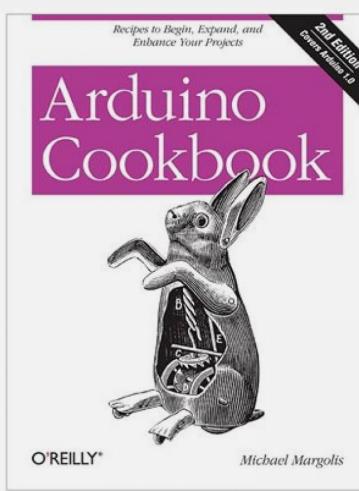
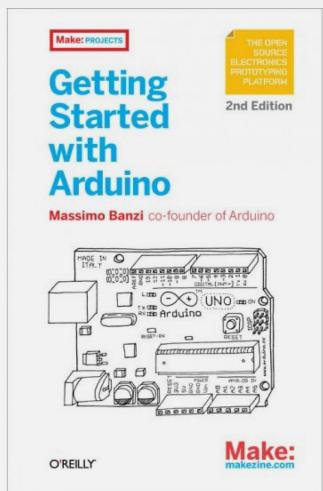


arduino.cc/en/Tutorial/HomePage



bildr.org

COMPUTING// REF



**THANK
YOU**



MT06 - Desafío

LEER MINIMO DOS INPUTS

(recomendamos Sensor de Temperatura + Fotoresistencias)

ACTUAR MINÍMO DOS OUTPUTS

(recomendamos sermotor + motor dc con transistor)

NOTA: Puntos extra por simular el circuito tambien en Thinkercad Electronics

**Documentar tu proceso y publicar en tu repositorio
de gitlab.**

Gracias!