

1. Introducción al Arduino

Tallerine Arduino/LED

Instituto de Ingeniería Eléctrica

5 de agosto de 2019

- Objetivos y características
- Equipo docente
- Sitio de cursos y foro de consultas
- Horario de clases, cronograma, horarios de consulta
- Obligatorios, parcial y proyecto final

- Actividad orientada a la generación de ingreso.
- Introducción a la Ingeniería Eléctrica, sus métodos y herramientas
- De paso, datos básicos de la carrera:
 - <http://www.fing.edu.uy/carreras/grado/ingenieriaelectrica>
 - Director de carrera: Álvaro Giusto (alvaro@fing.edu.uy)

- Motivación
- Trabajo en equipo
- Comunicación oral, escrita y digital
- Enseñanza activa: **HACER**, no sólo escuchar
- ¿Y qué vamos a hacer?

Qué vamos a hacer?



Docentes

- Sebastián Montes de Oca
- Ignacio Hounie
- Ignacio Ramírez (responsable)

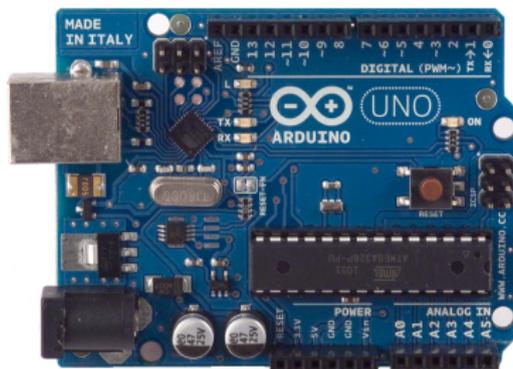
- Sitio de cursos: <https://eva.fing.edu.uy/>
- Página del Módulo:
<https://eva.fing.edu.uy/course/view.php?id=1319>
 - foros de información y consulta
 - material de apoyo
 - ejercicios a realizar

- **lunes de 9:00 a 12:00, Laboratorio de Software del IIE**
- Grupos de 3 estudiantes
- Recibirán **en préstamo** materiales para trabajar
- **Temas:** Presentación y puesta en práctica con ejercicios sencillos
- **Desafíos:** integración de temas vistos en un problema más desafiante
- **Evaluación:** informe y muestra de ejercicios y desafíos.
Los desafíos incluyen una breve defensa oral.

- 1 Introducción al Arduino
- 2 Programación en Arduino
- 3 Manejo de sensores
- 4 PRIMER DESAFÍO (2 talleres + defensa)
- 5 Electrónica digital
- 6 Generación de imágenes
- 7 Armado de circuitos
- 8 SEGUNDO DESAFÍO (3 talleres + defensa)



¿Un microcontrolador? ¿para qué sirve?



- Robótica
- Control de vehículos
- Toma de medidas: temperatura, nivel de un tanque, motores, etc.

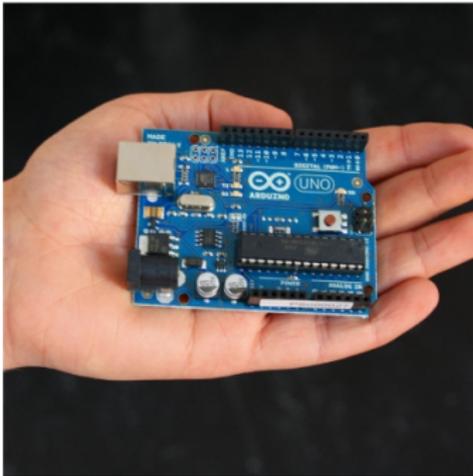


- Filosofía basada en el **software/hardware libre**.
- Estudiar el hardware para entender cómo funciona
- Hacer modificaciones al hardware
- Poder compartir esas modificaciones con la comunidad

Main Site | Blog | Playground | Forum | Labs | Store Help | Sign in or Register



Buy | Download | Getting Started | Learning | Reference | Hardware | FAQ



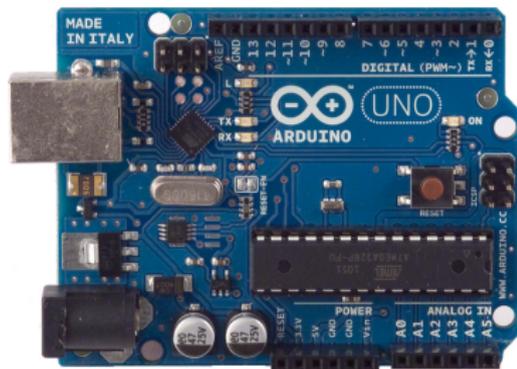
Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

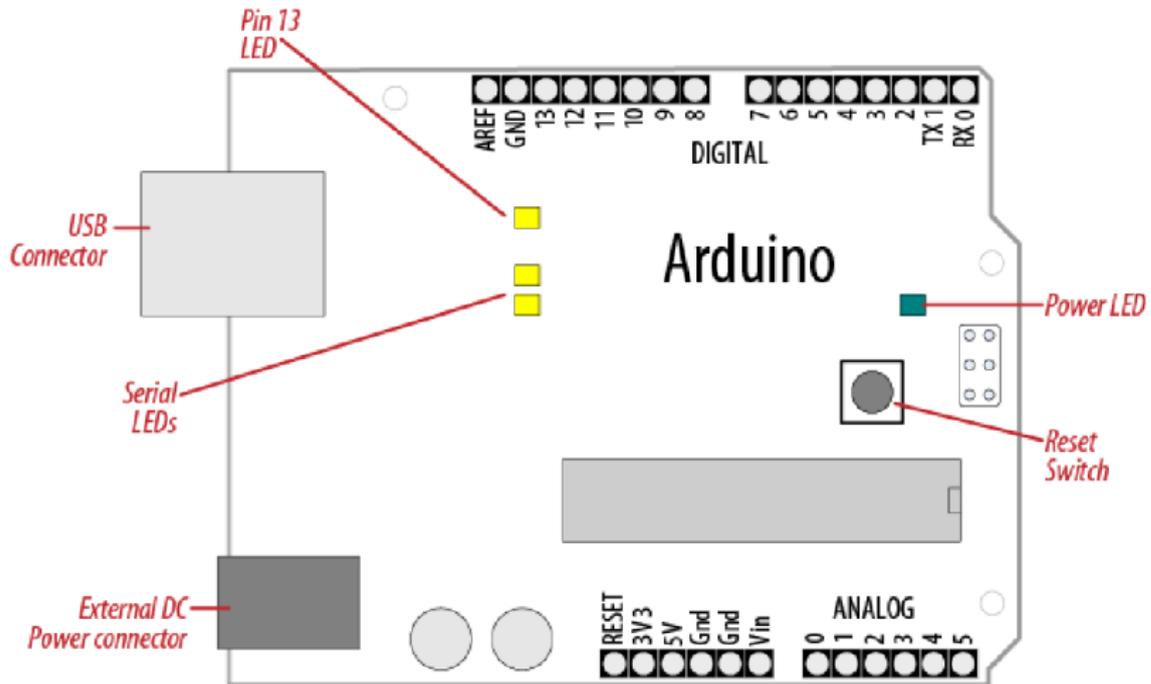
Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the **Arduino programming language** (based on **Wiring**) and the Arduino development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

Consecuencias:

- Comunidad activa de usuarios y desarrolladores
- foros, ejemplos, tutoriales, proyectos prontos, etc.
- Gran cantidad de accesorios de hardware
- Precios muy bajos
- Mucho material disponible en el web

- Microcontrolador: ATmega328 (8 bits)
- Alimentación: USB (5 V) o independiente (**7-12V**)
- 14 pines de entrada/salida (digitales + PWM)
- Corriente máxima por pin I/O: **40 mA**
- 6 pines de entrada analógica
- Comunicación serial
- Memoria flash: 32 KB
- Frecuencia del reloj: 16 MHz

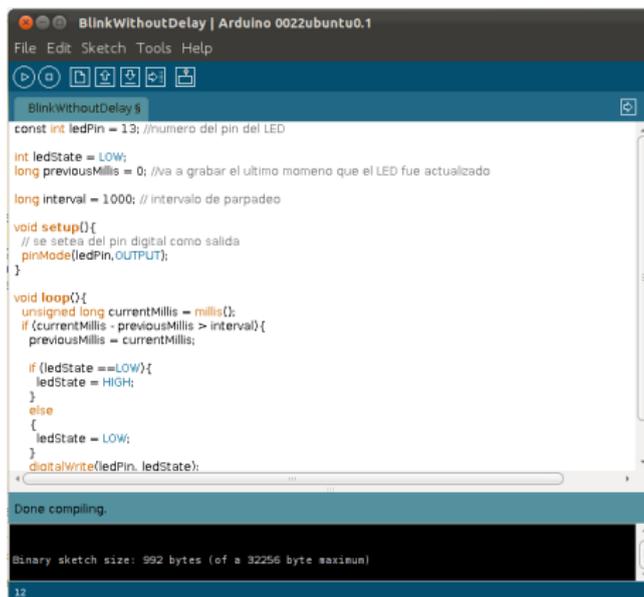




M.

Margolis, *Arduino Cookbook*, O'Reilly Media 2011.

Programación



```
BlinkWithoutDelay | Arduino 0022ubuntu0.1
File Edit Sketch Tools Help
BlinkWithoutDelay $
const int ledPin = 13; //numero del pin del LED

int ledState = LOW;
long previousMillis = 0; //va a grabar el ultimo momento que el LED fue actualizado

long interval = 1000; // intervalo de parpadeo

void setup(){
  // se setea del pin digital como salida
  pinMode(ledPin,OUTPUT);
}

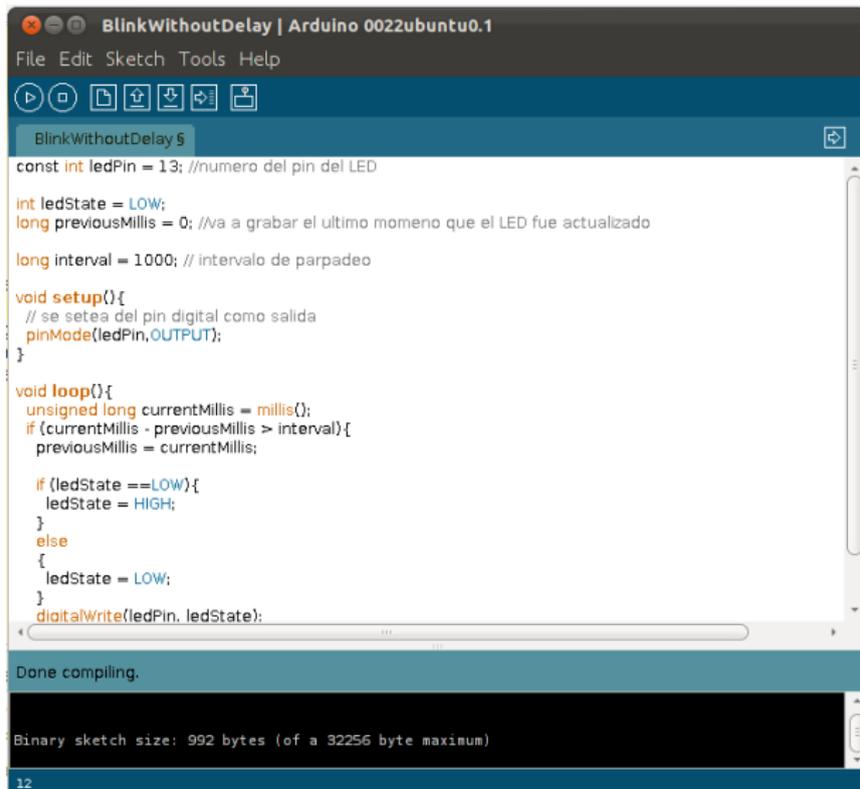
void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis > interval){
    previousMillis = currentMillis;

    if (ledState ==LOW){
      ledState = HIGH;
    }
    else
    {
      ledState = LOW;
    }
  }
  digitalWrite(ledPin, ledState);
}

Done compiling.

Binary sketch size: 992 bytes (of a 32256 byte maximum)
12
```

Entorno de desarrollo: Arduino IDE



```
const int ledPin = 13; //numero del pin del LED

int ledState = LOW;
long previousMillis = 0; //va a grabar el ultimo momento que el LED fue actualizado

long interval = 1000; // intervalo de parpadeo

void setup(){
  // se setea del pin digital como salida
  pinMode(ledPin,OUTPUT);
}

void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis > interval){
    previousMillis = currentMillis;

    if (ledState ==LOW){
      ledState = HIGH;
    }
    else
    {
      ledState = LOW;
    }
  }
  digitalWrite(ledPin, ledState);
}
```

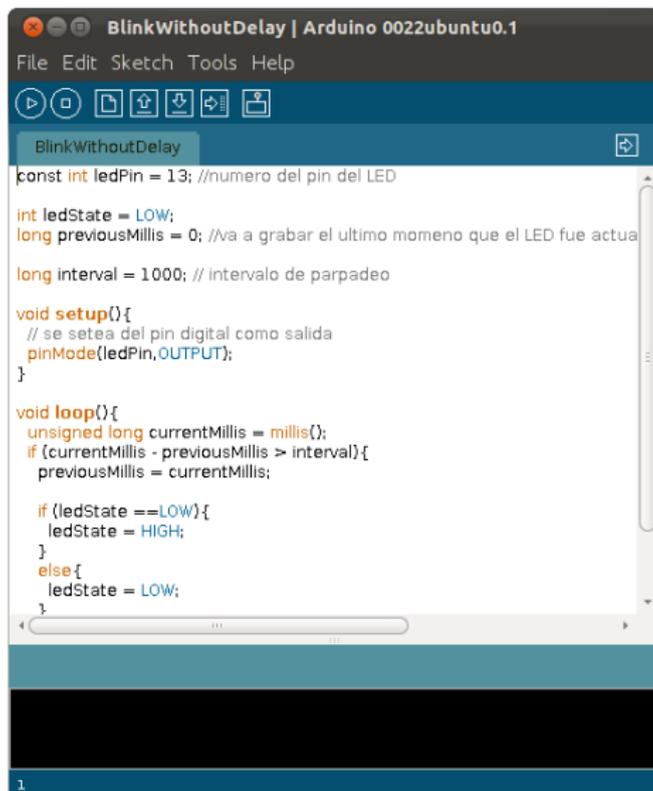
Done compiling.

Binary sketch size: 992 bytes (of a 32256 byte maximum)

12

Escritura de un programa

El "sketch"



```
const int ledPin = 13; //numero del pin del LED

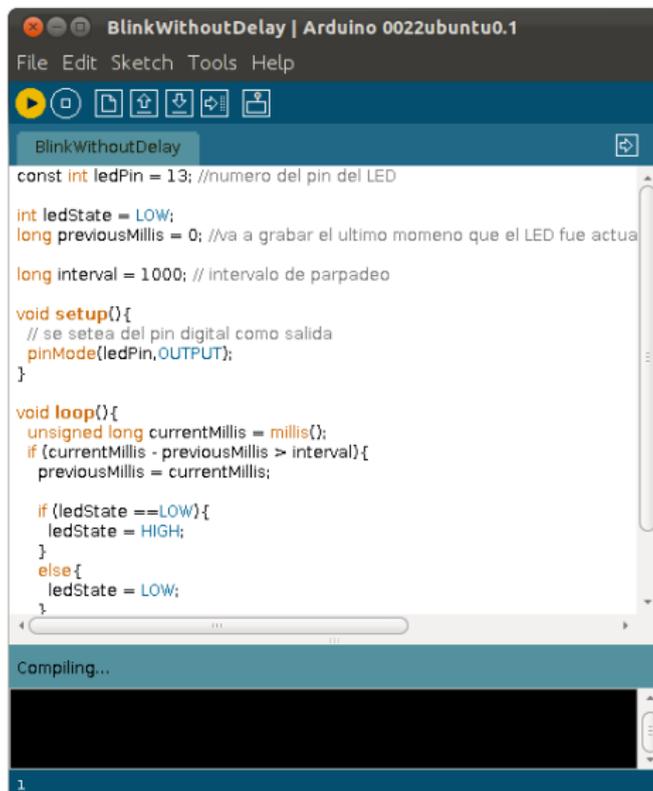
int ledState = LOW;
long previousMillis = 0; //va a grabar el ultimo momeno que el LED fue actua

long interval = 1000; // intervalo de parpadeo

void setup(){
  // se setea del pin digital como salida
  pinMode(ledPin,OUTPUT);
}

void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis > interval){
    previousMillis = currentMillis;

    if (ledState ==LOW){
      ledState = HIGH;
    }
    else{
      ledState = LOW;
    }
  }
}
```



The screenshot shows the Arduino IDE interface. The title bar reads "BlinkWithoutDelay | Arduino 0022ubuntu0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for running, stopping, saving, and other functions. The main editor window displays the following C++ code:

```
const int ledPin = 13; //numero del pin del LED

int ledState = LOW;
long previousMillis = 0; //va a grabar el ultimo momeno que el LED fue actua

long interval = 1000; // intervalo de parpadeo

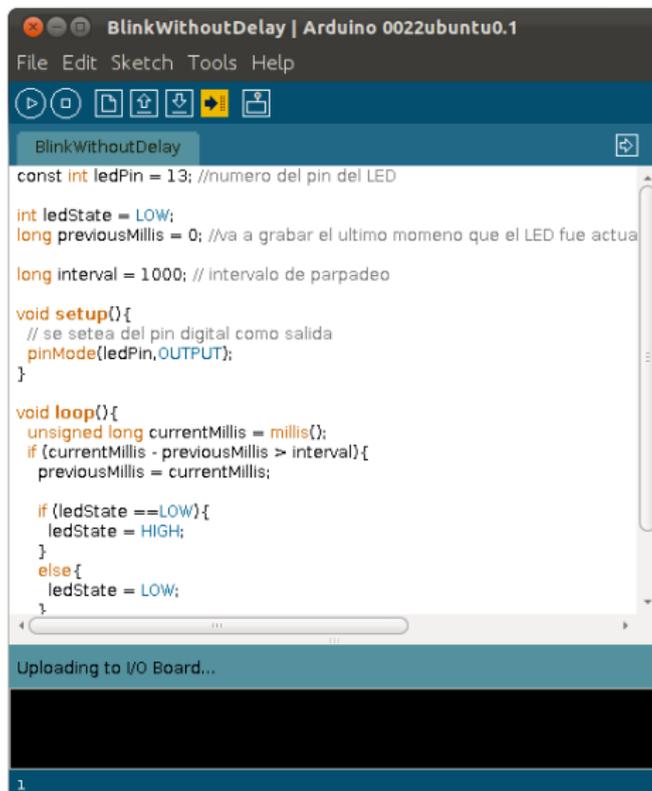
void setup(){
  // se setea del pin digital como salida
  pinMode(ledPin,OUTPUT);
}

void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis > interval){
    previousMillis = currentMillis;

    if (ledState ==LOW){
      ledState = HIGH;
    }
    else{
      ledState = LOW;
    }
  }
}
```

Below the code editor, a status bar indicates "Compiling...". At the bottom left of the IDE window, the line number "1" is visible.

Subida al arduino upload



```
const int ledPin = 13; //numero del pin del LED

int ledState = LOW;
long previousMillis = 0; //va a grabar el ultimo momeno que el LED fue actual

long interval = 1000; // intervalo de parpadeo

void setup(){
  // se setea del pin digital como salida
  pinMode(ledPin,OUTPUT);
}

void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis > interval){
    previousMillis = currentMillis;

    if (ledState ==LOW){
      ledState = HIGH;
    }
    else{
      ledState = LOW;
    }
  }
}
```

Uploading to I/O Board...

1

Estructura básica de un Sketch

```
int LED\_LUZ = 13; // declaracion de variable

void setup() // esto es una FUNCION
{
    // inicializacion de perifericos
    pinMode(LED_LUZ,OUTPUT); // pin LED_LUZ saca datos
}

void loop() // esto es otra FUNCION
{
    // todo esto se ejecuta una y otra vez
    digitalWrite(LED_LUZ,LOW); // apaga luz
    delay(1000); // espera 1000 milisegundos
    digitalWrite(LED_LUZ,HIGH); // prende luz
    delay(1000);
}
```

Variables

```
int pines = 10;
float fraccion = 0.43;
boolean logico = FALSE;
char letra = 'A';
String titulo = "Crimen";
```

Sentencias

```
a = a + 1;
f = digitalRead(PIN_OCHO);
b = (a << 3) & 0x0f;
d = !d;
m *= 3;
return a*3;
```

Control de Flujo

```
if (a < 0) {
  a = 10;
} else {
  a = 0;
}
```

Funciones

```
void setup() {
  a = 0;
}
int suma(int a, int b) {
  return a + b;
}
```

Variables

Tipos de variables

<code>byte pepe = 103;</code>	entero de 1 byte	-128 a 127
<code>int coco = -1000;</code>	entero (2 bytes)	-32.768 a 32.767
<code>long mama = 0;</code>	entero (4 bytes)	$-4e^9$ a $4e^9$
<code>float fefe = 0.123;</code>	“reales” (4 bytes)	punto flotante
<code>boolean kiki = FALSE;</code>	var. lógica	TRUE o FALSE
<code>char c= ´a´;</code>	un caracter	tabla ASCII
<code>String s="mama";</code>	texto	

Aritméticos

- $A + B$: suma
- $A - B$: resta
- $A * B$: producto
- A / B : división
- $A \% B$: resto de div. entera

Lógicos

- $A \&\& B$: **AND** B
- $A \|\| B$: **OR** B
- $!A$: **NOT** A

Comparación

- $A == B$: A igual a B
- $A != B$: A distinto de B
- $A < B$: A menor que B
- $A <= B$: A menor o igual a B

Manejo de bits

- $A \& B$: **AND**
- $A | B$: **OR**
- $A \wedge B$: **OR EXCL**
- $!A$: negación
- $A \ll k$: despl. izq.

if-else if-else

```
if (a == 3) {  
    c << 1;  
} else if (a == 2) {  
    c >> 1;  
} else {  
    a = 0;  
}
```

while

```
while (a > 10) {  
    a = a - 1;  
}
```

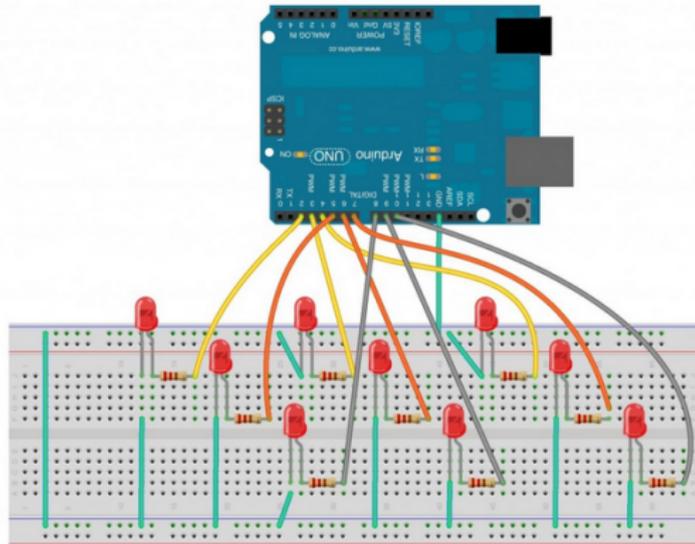
for

```
for (i = 0; i < 1000; i++) {  
    delay(i);  
    digitalWrite(LOW);  
    delay(1000-i);  
    digitalWrite(HIGH);  
}
```

switch-case

```
switch (estado) {  
    case QUIETO: arrancar();  
    case CANSADO: frenar();  
    default: acelerar();  
}
```

Entrada y salida digital



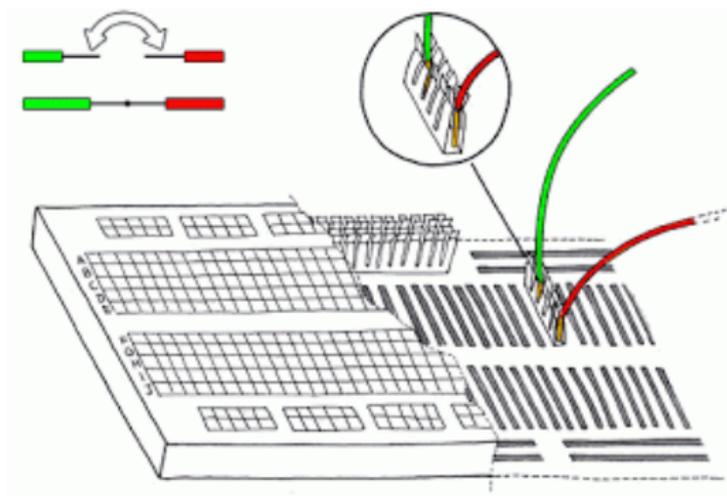
- Comunicación del Arduino con el exterior
- Pines pueden ser entrada o salida, analógicos o digitales
- Entrada: leer voltaje de un pin como un número
- Salida: definir voltaje de un pin escribiendo un número
- Digital: $0V = \text{LOW}$, $5V = \text{HIGH}$
- Analógico: $0V = 0$, $5V = 1023$

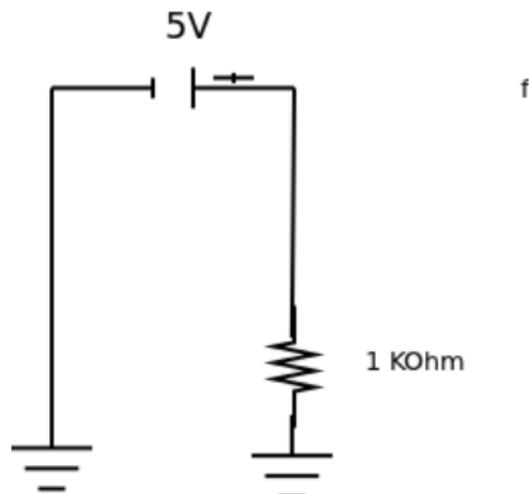
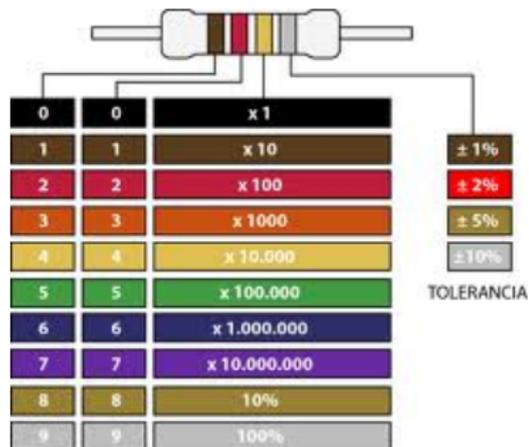
```
int pinEntrada = 10;
int pinSalida = 13;
int llave = 0;
void setup(){
    pinMode(pinEntrada , INPUT);
    pinMode(pinSalida , OUTPUT);
}

void loop(){
    llave = digitalRead(pinEntrada); //lee el valor del pin 10
    if(llave == LOW){
        digitalWrite(pinSalida , HIGH); //pone el pin 13 en +5V
    }else{
        digitalWrite(pinSalida , LOW); //pone el pin 13 en +0V
    }
}
```

Protoboard

- Placa de prototipado
- Conexión de bornes

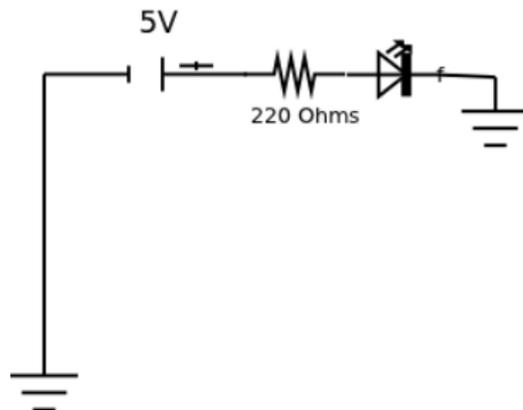
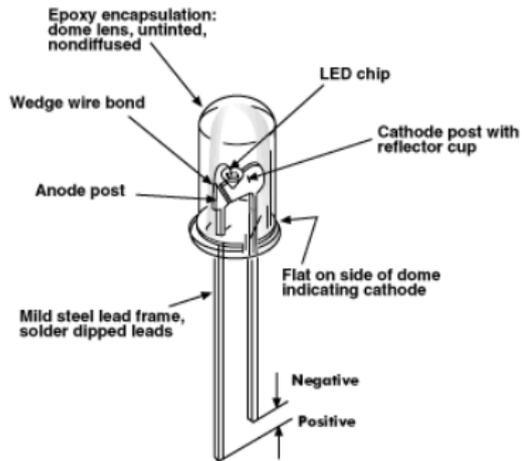




Código colores: tecnorik.blogspot.com

LED

Light Emitting Diode



Descripción LED: electroschematics.com

Ejemplo

Sketch: parpadeo.ino

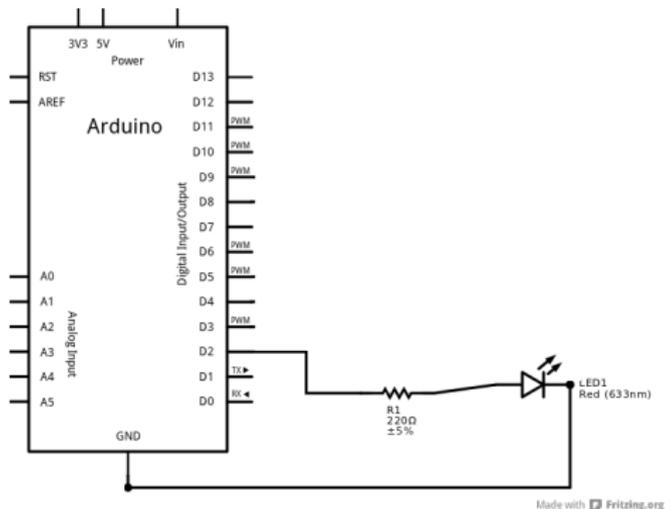
```
// el Arduino ya tiene un LED conectado a este pin
const int LED = 13;

// esto se ejecuta una vez cuando se enciende
// o resetea el Arduino
void setup() {
  // pone pin 13 como de salida
  pinMode(led, OUTPUT);
}
// despues de setup(), esto corre una y otra vez para siempre
void loop() {
  digitalWrite(led, HIGH); // prende LED
  delay(1000);             // espera 1 seg
  digitalWrite(led, LOW); // apaga LED
  delay(1000);            // espera 1 seg
}
```

Trabajo en el taller

Ejercicio 1

- Implementar el circuito de la figura
- Programar, compilar y cargar el programa parpadeo.ino
- Modificarlo para que el parpadeo sea cada 0.4 segundos



- Implementar un programa que haga parpadear el LED si el pin 5 está en 5V (HIGH), y que lo deje apagado si el pin 5 está en 0V (LOW)

Entregable 1

Escribir un programa que prenda y apague secuencialmente tres leds. Es decir se prende un led por medio segundo, luego se apaga y se prende el siguiente por un segundo, cuando llega al último vuelve al primero.

Para próxima clase

Entregable

Práctica funcionando al momento de la entrega

Informe

- Trabajo realizado en los *ejercicios de clase*
- Trabajo realizado en el entregable
- Código fuente, circuito y explicación

Pautas sobre informe

- No interesa: descripción precisa de la tarea (alcanza con título)
- Sí interesa: qué salió bien, qué salió mal, qué costó, qué aprendieron, qué no quedó claro.