

PROGRAMACION EN ARDUINO

ING. JULIO CESAR SORIA QUISPE

ARDUINO



Introducción

- Arduino (<http://www.arduino.cc/>) es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.



ARDUINO



Arduino UNO

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14
 - of which 6 provide PWM output
- Analog Input Pins 6
- Flash Memory 32KB
- SRAM 2KB
- EEPROM 1KB
- Clock Speed 16 MHz

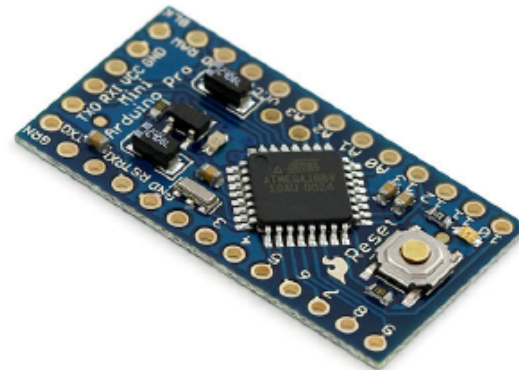
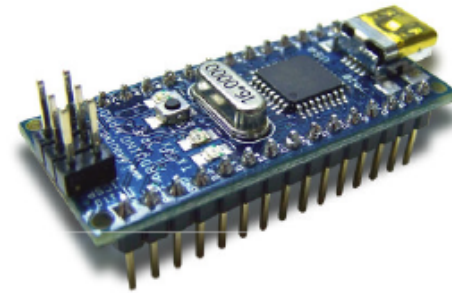
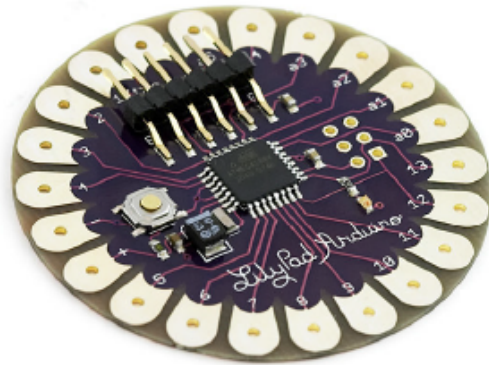


22 €

ARDUINO



Otras placas Arduino



ARDUINO



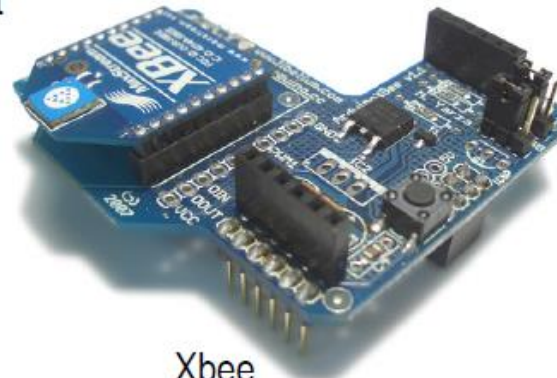
Algunos Shields



Ethernet



Motor



Xbee



LCD

ARDUINO



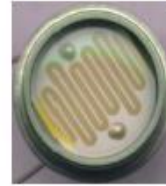
Algunos Sensores



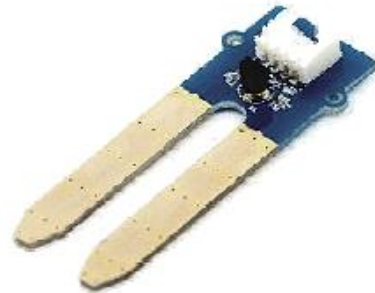
Fotodiodos



Láser



Fotoresistencias



Humedad



Ultrasonido

ARDUINO



Programación

- Lenguaje de programación Arduino (basado en Wiring, a su vez basado en C/C++)
 - El entorno Arduino (basado en Processing, a su vez desarrollado en Java) es abierto.
 - Se descarga de <http://arduino.cc/es/Main/Software>
 - Otros entornos:
 - Autónomos
 - Minibloq, Ardubloq, Amici, ModKit, VirtualBreadBoard, Matlab
 - Esclavos
 - Etoys (Squeak), S4A (Scratch), Labview, Firefly, MyOpenLab
-

ARDUINO



Entorno de programación

Programas = Sketches

The screenshot shows the Arduino IDE 1.0.2 window. The title bar reads "Blink | Arduino 1.0.2". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for opening, saving, and running sketches. The main text area displays the code for the "Blink" sketch. The code is as follows:

```
*/  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage  
  delay(1000);             // wait for a second  
}
```

At the bottom of the window, a status bar indicates "Arduino Uno on COM1".

ARDUINO

Configuración



- 1 FILE>**
 - EXAMPLES>**
 - BASICS>**
 - BLINK**
- 2 TOOLS>**
 - BOARD>**
 - ARDUINO UNO**
- 3 TOOLS>**
 - SERIAL PORT**
 - Windows OS **COM####**
 - Mac OS **/dev/ttyusbmodem####**
 - Linux OS **/dev/ttyACM####**
- 4 FILE>**
 - UPLOAD**

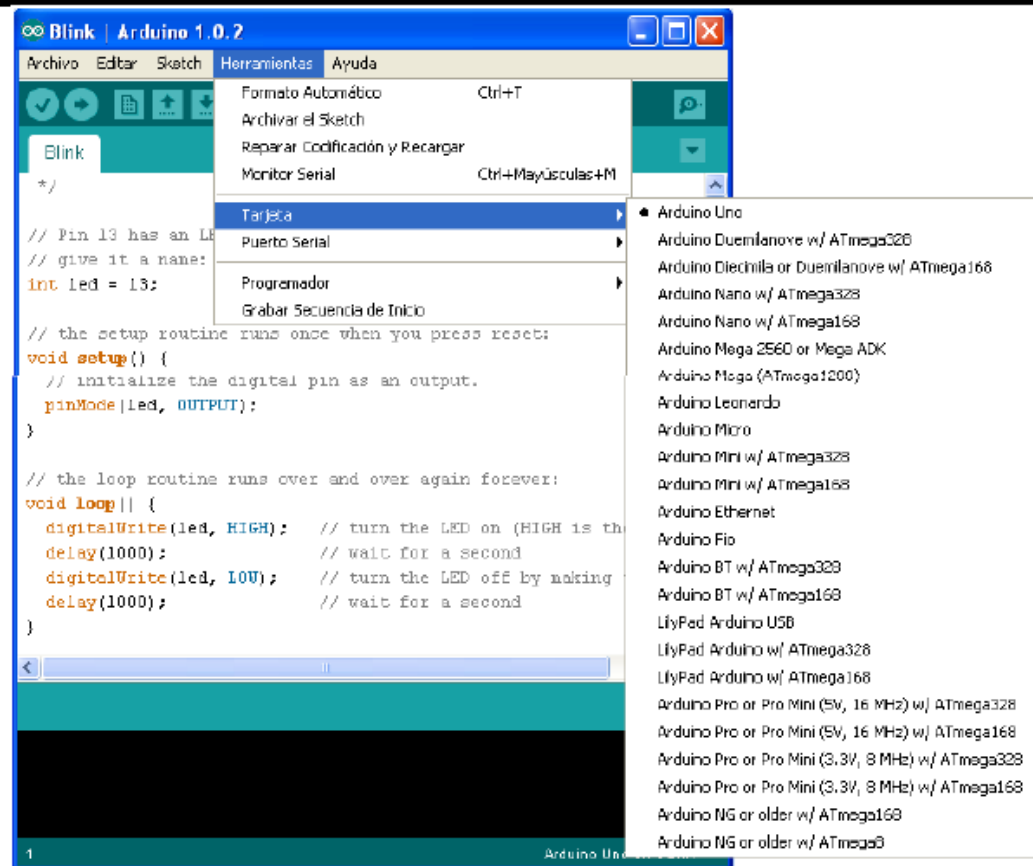


[Imagen CTC Arduino Verktad]

ARDUINO



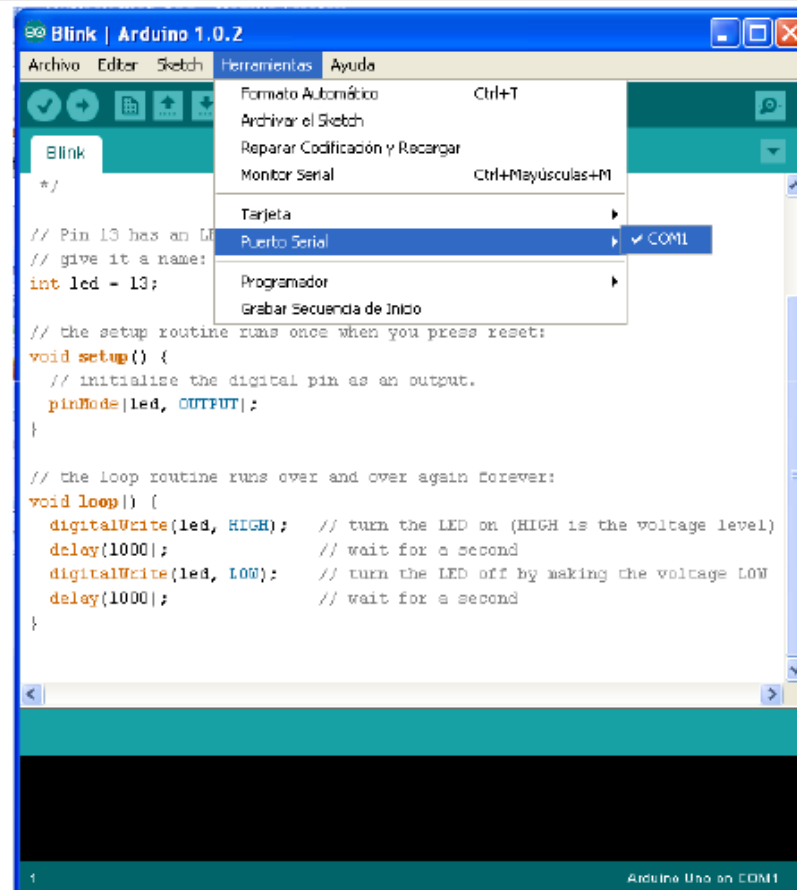
Tipo de tarjeta (board)



ARDUINO



Puerto serie / COM



ARDUINO



Programación: estructura

```
void setup() {  
    ....se ejecuta una vez cuando se ejecuta  
    ....un programa Arduino  
}  
void loop() {  
    .... se ejecuta después de setup. Se ejecuta  
    .... de manera repetida hasta que se quita la  
    .... tensión  
}
```

ARDUINO



Programación: Comentarios y sintaxis

- Igual que en C
 - Multilínea `/* Comentario */`
 - Línea `// Línea`
 - `{ }` para encerrar código
 - `;` para terminar instrucciones
-

ARDUINO



Programación: Operadores de comparación

- == (equal to)
 - != (not equal to)
 - < (less than)
 - > (greater than)
-

ARDUINO



Programación: Operadores matemáticos

- = (assignment)
 - % (módulo)
 - +
 - -
 - *
 - /
-

ARDUINO



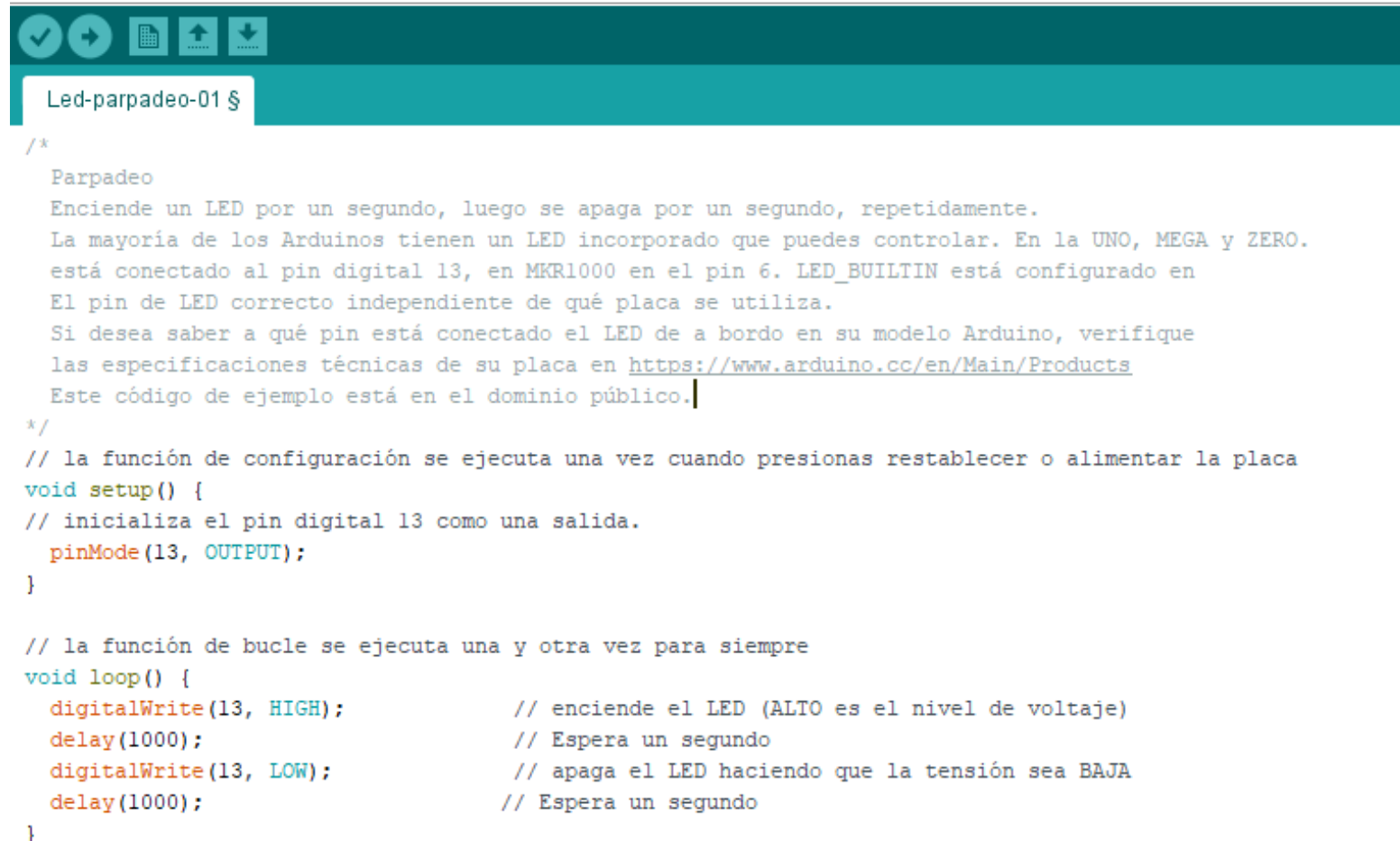
Programación: Variables

- Tipos básicos:
 - int
 - long
 - boolean
 - float
 - char
-

ARDUINO

☺ Led-parpadeo-01 Arduino 1.8.1

Archivo Editar Programa Herramientas Ayuda



```
/*
 * Parpadeo
 * Enciende un LED por un segundo, luego se apaga por un segundo, repetidamente.
 * La mayoría de los Arduinos tienen un LED incorporado que puedes controlar. En la UNO, MEGA y ZERO.
 * está conectado al pin digital 13, en MKR1000 en el pin 6. LED_BUILTIN está configurado en
 * El pin de LED correcto independiente de qué placa se utiliza.
 * Si desea saber a qué pin está conectado el LED de a bordo en su modelo Arduino, verifique
 * las especificaciones técnicas de su placa en https://www.arduino.cc/en/Main/Products
 * Este código de ejemplo está en el dominio público.
 */
// la función de configuración se ejecuta una vez cuando presionas restablecer o alimentar la placa
void setup() {
  // inicializa el pin digital 13 como una salida.
  pinMode(13, OUTPUT);
}

// la función de bucle se ejecuta una y otra vez para siempre
void loop() {
  digitalWrite(13, HIGH);      // enciende el LED (ALTO es el nivel de voltaje)
  delay(1000);                 // Espera un segundo
  digitalWrite(13, LOW);       // apaga el LED haciendo que la tensión sea BAJA
  delay(1000);                 // Espera un segundo
}
```

ARDUINO

Código

```
// Declaración de variables:

int led = 12;           // Led conectado en el pin 12

// Configuración:

void setup() {
  pinMode(led, OUTPUT); // Configuramos el pin como salida
}

// Programa:

void loop() {           // El programa se ejecuta repetidamente

  digitalWrite(led, HIGH); // Envía 5V al pin del led (12) (enciende)
  delay(500);              // espera 500ms = 0,5s
  digitalWrite(led, LOW);  // Envía 0V (apaga)
  delay(100);              // espera 100ms = 0,1s

}
```

ARDUINO

Práctica #1: Encender y apagar un Led:

Objetivo: Prender y apagar un led ubicado en el PIN9 de manera automática que se encienda por 3 segundos y se apague por 0.5 segundos de manera indefinida.

En esta práctica se recomienda usar los comandos:

- **pinMode:** Realiza asignación de puerto de entrada o salida, debido a que el microcontrolador no sabe cómo se va a usar cada puerto, antes de empezar a usarlos debemos declararlos ya sea como entrada o como salida
- **digitalWrite:** Este comando permite poner un uno o un cero lógico en el puerto
- **delay:** Genera un retardo de 1ms, esto quiere decir que si entre paréntesis colocamos el número 1000 el retardo será de un segundo.



En los sistemas de hardware digitales una de las funciones más importantes es colocar un uno digital o un cero digital en un puerto, esto implica que el puerto tenga una tensión equivalente al voltaje alto digital (5v, 3.3v, 1.8v) con un uno digital y 0V con un cero digital



No coloques más de 5V en una entrada analógica o digital de la Arduino UNO ya que puedes dañar el puerto, tampoco conectes directamente a los puertos elementos como motores que requieran más de 20mA debido a que este se puede dañar.

ARDUINO

Programación

```
/*  
Este programa enciende un led por 3segundos y lo apaga  
durante 0.5segundo y se repite de manera indefinida.  
para esto se conecta un led entre el pin9 y tierra  
*/  
int led = 9;           // constante led con valor de 9  
void setup() {  
  pinMode(led, OUTPUT); // puerto 13 como salida  
}  
void loop() {  
  digitalWrite(led, HIGH); // Puerto en Alto  
  delay(3000);             // retardo de 3seg  
  digitalWrite(led, LOW);  // Puerto en bajo  
  delay(500);              // retardo de 0.5seg  
}
```

Montaje

