

Que es arduino?

Arduino es un microcontrolador de código abierto que viene en una sola placa. Podríamos decir que es una pequeña computadora en un solo circuito integrado.

Arduino viene con un procesador de la familia ATmega, una pequeña memoria, pines digitales de E/S, pines de entrada analógicos, conector USB y una entrada de energía.

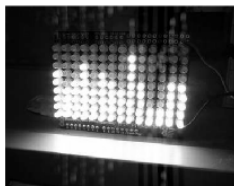
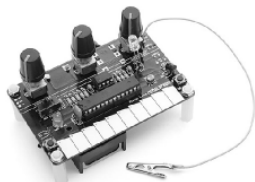
Para quienes está pensado arduino?

Está pensado para los artistas, diseñadores, aficionados y cualquier persona interesada en la creación de objetos o entornos interactivos.

Con arduino no es necesario ser un experto en electrónica ni tampoco saber mucho de programación! Todo lo que importa es tu creatividad y tus ideas!

Qué software utiliza?

La IDE de arduino está escrita en Java y por lo tanto está pensada para ser multiplataforma, funciona tanto en equipos con Windows como con GNU/Linux. El lenguaje de programación que utiliza arduino es una implementación de un lenguaje llamado Wiring, que utiliza una versión simplificada de C++. La IDE y el lenguaje que arduino utiliza permite de una manera sencilla y rápida de programar y entrar al mundo de la electrónica!



Por donde empezar?

Sitio web oficial de arduino en español:
<http://arduino.cc/es/>

Documental de arduino en español:
<http://vimeo.com/18390711>

No hay nada mejor como aprender con otras personas!
Visita nuestro hackspace!
<http://www.mateslab.com.ar>

Donde conseguir un arduino?

En Argentina:
<http://openhacks.com>, <http://www.ditentec.com.ar/>, <http://www.mercadolibre.com.ar/>

En el mundo y con mejores precios:
<http://www.dealextreme.com/>
(Los envíos tardan un mínimo de 2 semanas)



MatesLab
Hackspace

<http://www.mateslab.com.ar>
<https://twitter.com/#!/mateslab>
<http://www.facebook.com/MatesLab>

GNU/Linux

Hoja de referencia o *Cheat sheet*



Trabajando con archivos

pwd	Muestra en que directorio estamos ubicados
ls	Lista el contenido de un directorio. Ej.: <code>ls /home/</code>
ls -l	Lista el contenido de un directorio y muestra información adicional de los archivos
ls -a	Lista el contenido de un directorio y también los archivos ocultos
cd	Ir al directorio <i>home</i>
cd ..	Ir al directorio padre
cd dir/	Ir la directorio <i>dir</i>
mkdir dir	Crea el directorio <i>dir</i>
rm file	Elimina el archivo <i>file</i>
rm -r dir/	Elimina el directorio <i>dir</i>
cp file1 file2	Copia el archivo <i>file1</i> en <i>file2</i> (Si no existe <i>file2</i> lo crea)
cp -r dir1 dir2	Copia el directorio <i>dir1</i> en <i>dir2</i> (Si no existe lo crea)
mv file1 file2	Renombra <i>file1</i> como <i>file2</i> . Si <i>file2</i> es un directorio entonces mueve <i>file1</i> dentro de <i>file2</i>

Información del sistema

df	Muestra el espacio libre en los discos
df -h	Muestra el tamaño del espacio libre en un formato entendible por humanos
du	Muestra el tamaño usado por los directorios
free	Muestra el uso actual de memoria y swap
date	Muestra la fecha y hora actual
cal	Muestra el calendario del mes actual
uptime	Muestra el tiempo que lleva encendida la computadora
w	Muestra los usuarios conectados a la computadora e información de cada uno
finger <usuario>	Muestra información sobre el <i><usuario></i>
whoami	Retorna el nombre de usuario del usuario actual
uname -a	Muestra información sobre el <i>kernel</i> o <i>núcleo</i>
cat /proc/cpuinfo	Muestra información detallada sobre la CPU
cat /proc/meminfo	Muestra información detallada sobre la memoria
man <comando>	Abre el manual de referencia del <i><comando></i> . Ej.: <code>man ls</code>

Gestión de procesos

ps	Muestra información sobre una selección de los procesos activos
ps ax	Muestra información sobre todos los procesos activos
top	Muestra dinámicamente y en tiempo real una vista de todos los procesos activos
kill <pid>	Manda una señal para matar al proceso indicado por <i><pid></i>
killall <proceso>	Manda una señal para matar a todos los procesos corriendo el comando <i><proceso></i>
bg	Lista los procesos parados o en background. Permite reanudar un proceso en background
fg	Trae el trabajo más reciente al primer plano

Red

ping <host>	Envía paquetes ICMP ECHO-REQUEST al <i>host</i> y muestra los resultados. Ej: <code>ping 8.8.8.8</code>
whois <dominio>	Muestra información whois del <i>dominio</i> . Ej.: <code>whois google.com</code>
dig <dominio>	Muestra información sobre los DNS del <i>dominio</i> . Ej.: <code>dig google.com</code>
wget <URL>	Descarga el contenido referenciado por <i>URL</i>
ifconfig	Muestra información sobre las interfaces de red activas
iwconfig	Muestra información sobre las interfaces wireless existentes
iptables -vnL	iptables es un firewall, con estas opciones se lista todas las reglas del firewall existentes.
route -n	Muestra el contenido actual de las tablas de enrutamiento

Basado en el cheat sheet hecho por fosswire.com



Arduino

Hoja de referencia o *Cheat sheet*

Estructura básica del programa

<code>void setup() { //código }</code>	Se ejecuta una vez cuando el programa inicia
<code>void loop() { //código }</code>	Se ejecuta repetidamente una vez iniciado el programa

www.mateslab.com.ar

Operadores generales

=	Asignación	!	Negación
+	Adición	-	Sustracción
*	Multiplicación	/	División
==	Igual a	!=	Distinto que
>	Mayor que	<	Menor que
>=	Mayor o igual que	<=	Menor o igual que
&&	(y)	 	(ó)
%	Módulo		

Operadores compuestos

++	Incremento
--	Decremento
+=	Suma compuesta
-=	Resta compuesta
*=	Multiplicación compuesta
/=	División compuesta
&=	AND binario compuesto
 =	OR binario compuesto

Tipos de datos (Lista reducida)

void	Vacío
boolean	(0, 1, true, false)
char	(Ej.: a., -128 a 127)
int	(-32768 a 32767)
long	(-2147483648 a 2147483647)
byte	(0 a 255)
word	(0 a 65535)
float	(-3,4028235E+38 a 3,4028235E+38)

Basado en el cheat sheet oficial de arduino.cc

Estructuras de control de flujo

<code>if(condición) { //código a ejecutar si condición es verdadera } else { //código a ejecutar si condición es falsa }</code>
<code>while (condición) { //código a ejecutar mientras la condición //sea verdadera }</code>
<code>do { //código a ejecutar hasta que la condición //sea verdadera } while (condición);</code>
<code>for (int i = 0; i < 10; i++) { //código a ejecutar para los valores de i entre 0 y 9 }</code>
<code>switch (variable) { case 1: //código a ejecutar cuando variable es igual a 1 break; case 2: //código a ejecutar cuando variable es igual a 2 break; default: //código a ejecutar si variable no toma ninguno //de los valores anteriores break; }</code>
<code>break</code> Para inmediatamente la ejecución del bucle
<code>continue</code> Salta inmediatamente a la siguiente iteración

Como hacer parpadear un LED!

<code>int ledPin = 13; //LED conectado al pin digital 13 void setup() { //Seteamos el pin digital como salida pinMode(ledPin, OUTPUT); } void loop() { digitalWrite(ledPin, HIGH); //Encendemos el LED delay(1000); //Esperamos 1 segundo digitalWrite(ledPin, LOW); //Apagamos el LED delay(1000); //Esperamos 1 segundo }</code>
--

www.arduino.cc/es/