



24/10/2013

---

---

# Práctica 0: Arduino

*Tecnologías Emergentes*

---

---

Guión de prácticas

# Práctica 0: Arduino

## *Introducción al uso elemental de la placa Arduino UNO (R3)*

### Introducción

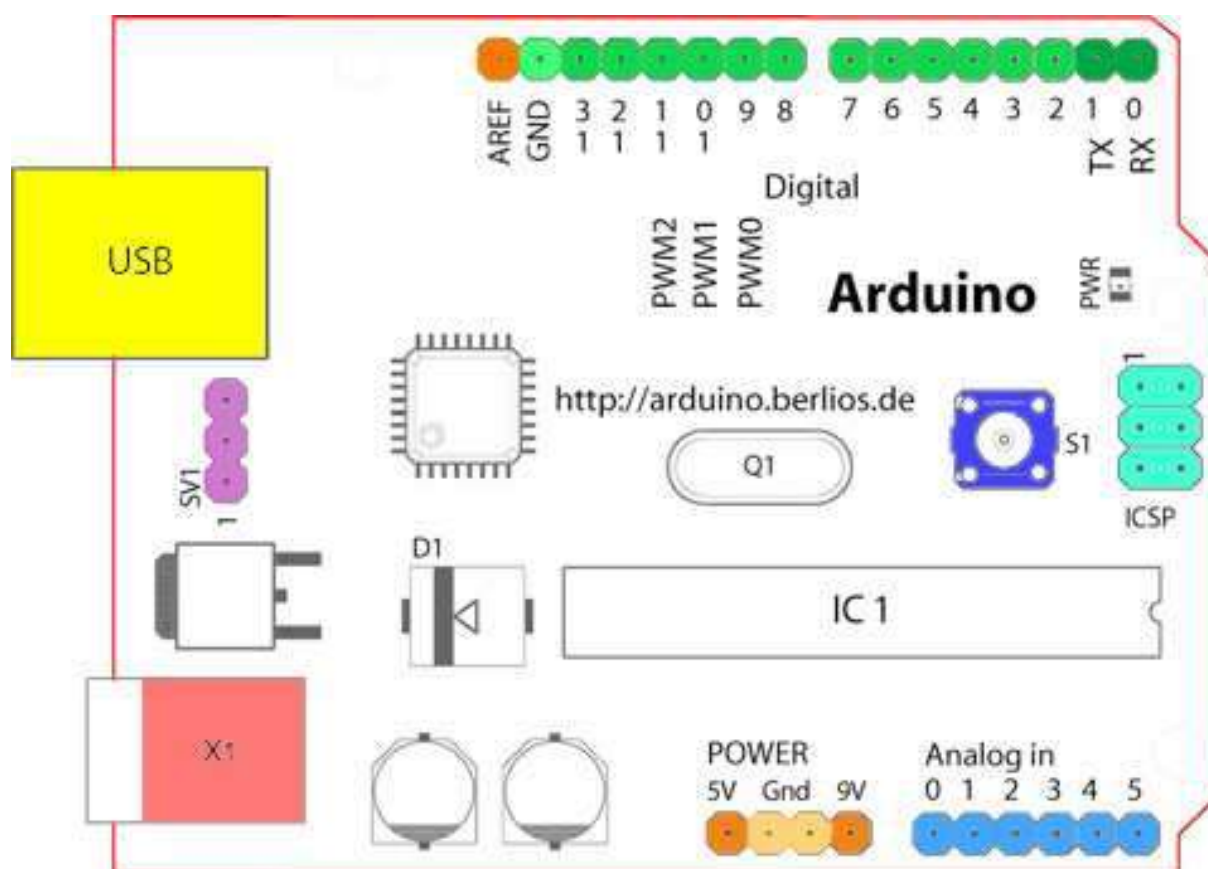
La placa Arduino UNO es uno de los modelos de placas de hardware libre (es decir, cuyo diseño, esquemáticos, programación, etc. están abiertos y disponibles a todos) más populares para el desarrollo de proyectos de hardware hoy en día.

La placa, que se puede alimentar bien por USB, por baterías o por un alimentador externo, se puede conectar al PC para programarla e incluso para comunicarse con el mismo durante su actividad. Pero también, una vez programada, puede usarse sin depender del PC.

Como se observa en el gráfico anterior, la placa dispone de una serie de E/S digitales (13) y entradas analógicas (5), algunas de ellas especializadas en realizar ciertas funciones (como la 0 y 1 para comunicación en serie, las 9 a 11 para generar señales cuadradas PWM, etc.).

La placa se programa en un entorno con lenguaje de alto nivel muy similar a C.

Recomendamos revisar los apuntes de teoría del seminario de Arduino antes de comenzar con la práctica.



## Entorno de trabajo

---

Para trabajar con la placa en el aula de prácticas, entraremos al cargador de imágenes REMBO (el que aparece al encender la máquina) con nuestro usuario y contraseña y con código “tec\_em”.

Hay que tener en cuenta que esta imagen tiene instalado el software y drivers para la placa Arduino modelo UNO R3. Si se desea usar otra placa, no contamos con privilegios de administrador para instalar los drivers, y por tanto, habrá que usar un portátil.

En caso de usar nuestro portátil, descargaremos el entorno de programación y seguiremos las instrucciones en:

<http://arduino.cc/en/Guide/Windows> (para Windows).

## Consideraciones previas.

---

Lo primero es la seguridad... así que antes de usar la placa, y para evitar daños en la misma, es conveniente leer la siguiente información:

<http://www.maelabs.ucsd.edu/mae156alib/electronics/DestroyArduino.pdf>

(“10 great ways to destroy an Arduino”, diez formas de cargarse una Arduino).

A modo de repaso de las funciones básicas de la placa, se recomienda dar un repaso al tutorial:

<http://www.me.umn.edu/courses/me2011/arduino/arduinoGuide.pdf>

(la misma web tiene gran cantidad de información, como ejemplos de proyectos, guía de instalación, etc.)

<http://www.me.umn.edu/courses/me2011/arduino/>

## Ejercicios propuestos

---

Para ir aprendiendo de forma progresiva el uso y programación de la placa, se propone realizar los siguientes ejercicios:

- Realizar el montaje necesario (usando LED, resistencia y placa de prototipado) para lanzar el programa básico de ejemplo “Blink” de parpadeo de un LED.

- Secuencia de leds. Se trata de encender y apagar 4 leds secuencialmente. Los leds deben estar conectados a los pines 5, 6, 7 y 8. Se deben encender y posteriormente apagar los leds desde el pin 5 al 8, con un tiempo de duración de encendido y apagado de 200 milisegundos. Nota: la secuencia principal del programa debe estar reproducida en una función a la que llamará el programa principal
- Cruce de semáforos. Se trata de un cruce de semáforos controlado por arduino, para ello utilizaremos en el primer semáforo los pines 3 (led rojo), 4 (led ambar), 5 (led verde), en el segundo semáforo utilizaremos los pines 6 (led rojo), 7 (led ambar) y 8 (led verde). La secuencia de funcionamiento debe ser : rojo 1 – verde 2 durante 3 segundos, rojo 1 – ambar 2 durante 500 ms, verde 1 – rojo 2 durante 3 segundos, ambar 1 - , rojo 2 durante 500 ms.
- SOS con zumbador. Se trata de un zumbador que en código morse (pitidos largos/cortos) especifica una palabra, en nuestro caso SOS. Para el que no lo sepa, la S son tres señales acústicas de corta duración y la O tres señales acústica de larga duración. El zumbador debe estar conectado al pin 13, los pitidos cortos tendrán una duración de 100 ms y los largos 300 ms. Entre letra y letra debe pasar un tiempo de 300 ms y entre SOSs debe haber un tiempo de 1000 ms. Nota: Debes usar variables para guardar los tiempos que vas a usar.
- Secuencia de leds con pulsador. Se trata de encender y apagar 4 leds secuencialmente al accionar un pulsador. El pulsador debe estar conectado al pin 4, y los leds a los pines 5,6,7 y 8. Se deben encender y posteriormente apagar los leds desde el pin 5 al 8, con un tiempo de duración de encendido y apagado de 200 milisegundos. Nota: la secuencia principal del programa debe estar reproducida en una función a la que llamará el programa principal.
- Termostato. Se trata de un dispositivo que haga funcionar un motor y un led cuando la temperatura supera cierto umbral. Para ello conectaremos una ntc (sensor térmico) a la entrada analógica 0, un led al pin 5 y un motor de corriente continua al pin 10. Cuando la temperatura llegue a cierto umbral de voltaje (entre 0 y 1024) que nosotros decidamos, se conectarán a la vez el diodo led y el motor que puede tener unas aspas de ventilador en su eje para enfriar la ntc. Además se deberá visionar el valor de voltaje en la entrada analógica (valor entre 0 y 1024) en una consola en el PC

## Para saber más

- Ejercicios resueltos de Arduino, con éstos y otros más completos  
[http://dl.dropboxusercontent.com/u/1894196/ejercicios\\_de\\_arduino\\_resueltos.pdf](http://dl.dropboxusercontent.com/u/1894196/ejercicios_de_arduino_resueltos.pdf)
- Electrodroid (para calcular resistencias, LEDS, etc), para Android  
<https://play.google.com/store/apps/details?id=it.android.demi.elettronica&hl=es>
- Arduino Boards, para Android  
[https://play.google.com/store/apps/details?id=com.coderz.arduino\\_boards&hl=es](https://play.google.com/store/apps/details?id=com.coderz.arduino_boards&hl=es)
- Arduino Examples Free (Android),  
[https://play.google.com/store/apps/details?id=com.coderzheaven.arduino\\_examples&hl=es](https://play.google.com/store/apps/details?id=com.coderzheaven.arduino_examples&hl=es)
- Arduino Companion (Android)  
<https://play.google.com/store/apps/details?id=air.com.flashgamer.ArduinoCompanion&hl=es>