

Seminario. Uso del simulador de Arduino

Duración: 1 sesión

1. Objetivos del seminario

Los objetivos concretos de este seminario son:

- conocer la plataforma Arduino: arquitectura, modo de funcionamiento, programación y conexión de periféricos
- conocer diferentes opciones para simular prototipos de Arduino
- realizar sencillos programas y prototipos en varios simuladores de Arduino

2. Introducción

Arduino 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.

La plataforma está formada por **tres componentes** principales que la hacen funcionar y ser seguida a nivel mundial por miles de usuarios:

- **Hardware**: placa electrónica básica para hacer desarrollos rápidos y económicos. Basado en hardware libre: los códigos y esquemas de conexión están disponibles para todo el mundo. Se permite una forma simple de aprender electrónica básica analizando los componentes que forman sus circuitos. También permite ampliar el hardware mediante el uso de otras placas (shields) que se conectan a la del microprocesador.
- **Entorno de programación**: Processing permite de forma muy sencilla la programación del hardware escogido. Además, este entorno es multiplataforma (Windows, Mac y Linux) y únicamente necesita un cable USB para conectarse a la placa.
- **Comunidad de usuarios**: existen miles de usuarios que participan activamente en nuevos desarrollos, ideas y evaluación. Se puede consultar cualquier duda tanto del hardware como del software de forma libre, tanto en la web oficial de Arduino como en múltiples foros y webs creados por la comunidad.

El éxito y repercusión que ha tenido se debe sobre todo a varios factores: **muy bajo coste del material a utilizar, facilidad de entendimiento y manejo** para crear aplicaciones concretas y finalmente se trata de un software y hardware abierto, con una licencia que permite su estudio, reproducción y modificación.

Arduino permite **implementar sistemas integrados de una forma muy sencilla**. Utilizándolo se dispone de un hardware genérico, el diseño del hardware se reduce a conectar placas de expansión o shields a la placa base de Arduino. De esta forma se elimina una de las partes que requiere conocimiento más específico y que supone fabricación. Así se permite que el desarrollador se concentre en la programación del firmware y en la funcionalidad del sistema.

Habitualmente se usa una placa física, de entre la gran variedad de placas de Arduino, como por ejemplo el **Arduino UNO o el Arduino MEGA**. Sin embargo, hay ocasiones en que se hace **necesario utilizar un simulador** para probar las conexiones y el programa antes de pasar a trabajar con los componentes en el circuito real.

Existen muchos simuladores de Arduino que no precisan una placa física Arduino conectada, como Simuino, ArduSim, Emulino, Virtronics, etc. Dada la popularidad en la actualidad de Arduino y su filosofía de software libre, surgen continuamente nuevos recursos disponibles en la Red relacionados con esta plataforma.

En este seminario vamos a estudiar los siguientes simuladores muy populares:

- UnoArduSim
- TinkerCad Circuits

3. Simulador UnoArduSim

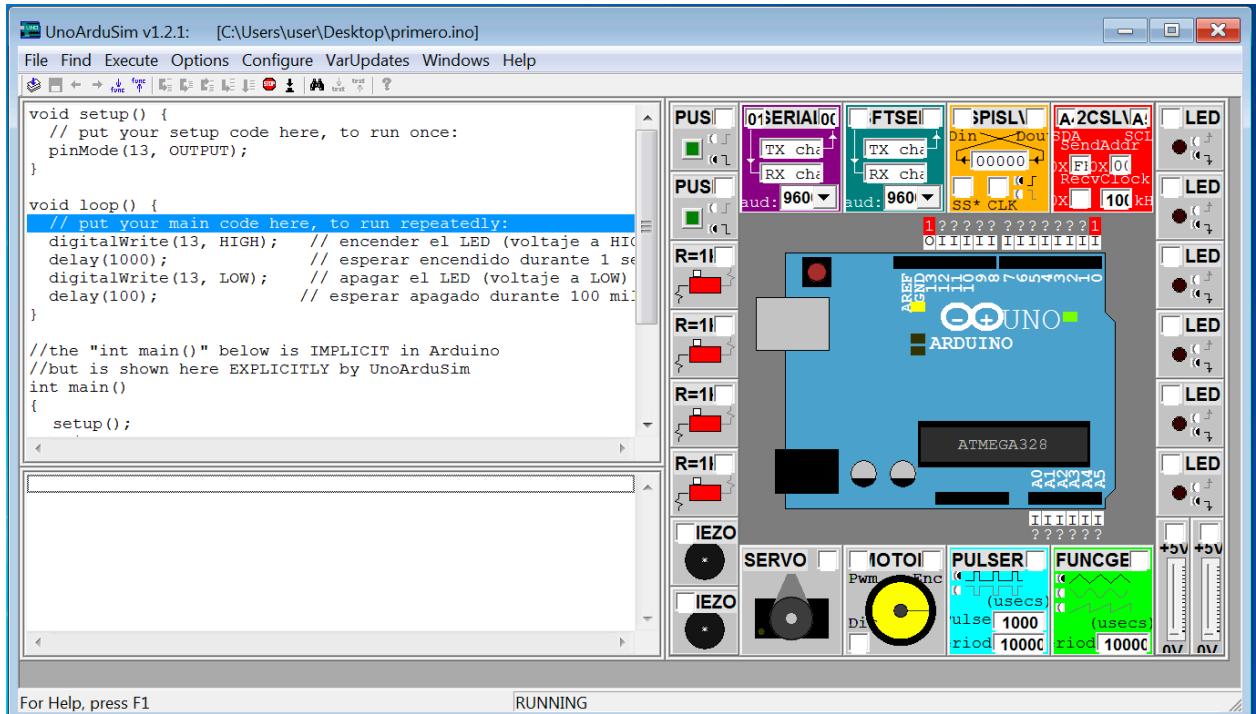
UnoArduSim es un programa que simula en la pantalla de nuestro ordenador una placa Arduino Uno. Nos permite utilizar LED, pulsadores, resistencias, potenciómetros, servomotores, buses de comunicación, generador de ondas y altavoz piezoelectrónico, entre otros elementos. Se trata de una herramienta educativa libre y gratuita, creada por Stan Simmons, de la Universidad de Queen's.

Este simulador **permite trabajar con prototipos de Arduino** sin necesidad de tener el dispositivo hardware para probar o depurar el funcionamiento de los proyectos/programas desarrollados para Arduino. Además, permite la ejecución paso a paso del programa, establecer puntos de parada (*breakpoints*), ejecución continua, etc, y analiza errores sintácticos o de escritura.

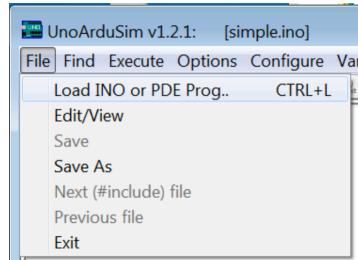
Podemos **descargar la versión más actual** de la página web del proyecto UnoArduSim:

<https://www.sites.google.com/site/unoardusim/services>

Una vez descargado, ejecutamos el programa y veremos una pantalla similar a la siguiente:



Veamos a continuación cómo hacer un programa sencillo que haga parpadear cada segundo el LED 8. Para ello, tenemos dos opciones, o bien cargamos el ejemplo “primero.ino” (con la opción File->Open), o bien creamos un ejemplo nuevo desde cero a partir del código básico que se nos muestra al abrir el programa (usar la opción del menú File->Edit/View):

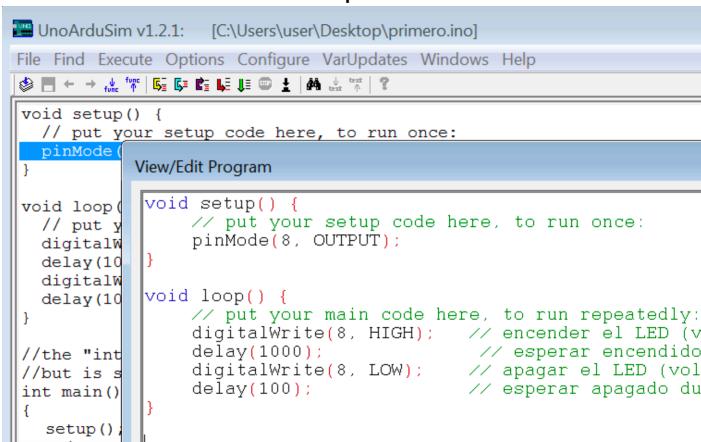


Como ejemplo, cargaremos el ejemplo del proyecto que hace parpadear el LED 8:

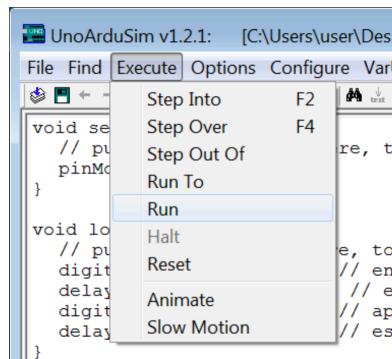
```
void setup() { // código de inicialización
    pinMode(8, OUTPUT); // usaremos el pin 8 como salida
}

void loop() { // código del programa principal
    digitalWrite(8, HIGH); // encender el LED (voltaje a HIGH)
    delay(1000); // esperar encendido durante 1seg
    digitalWrite(8, LOW); // apagar el LED (voltaje a LOW)
    delay(100); // esperar apagado durante 100mseg
}
```

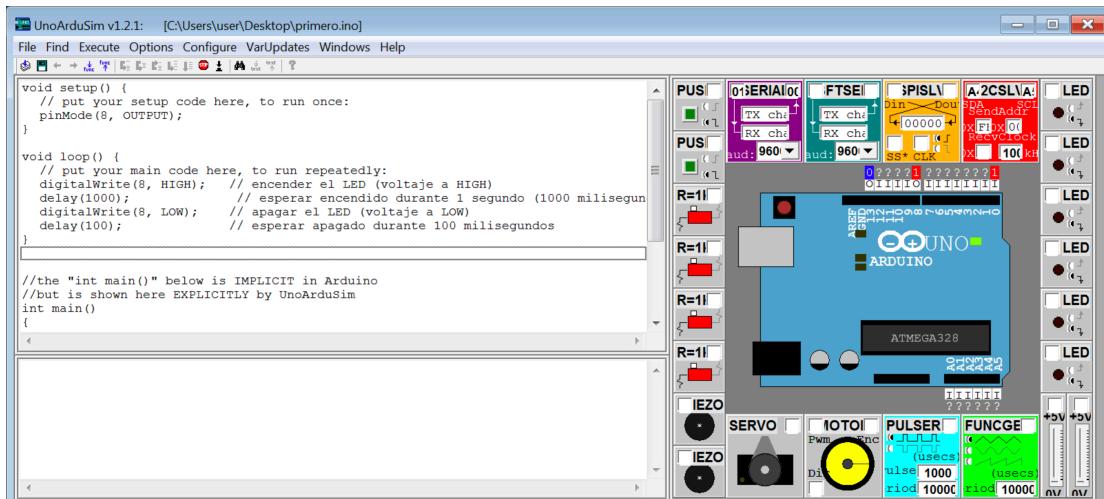
Si queremos modificar algo en el programa, haremos doble clic en la ventana del programa y se nos abrirá el editor para hacer modificaciones. Una vez terminadas, pincharemos a la derecha en el botón “Adopt”:



Para ejecutar el programa y comprobar su funcionamiento, usamos la opción del menú Execute->Run:

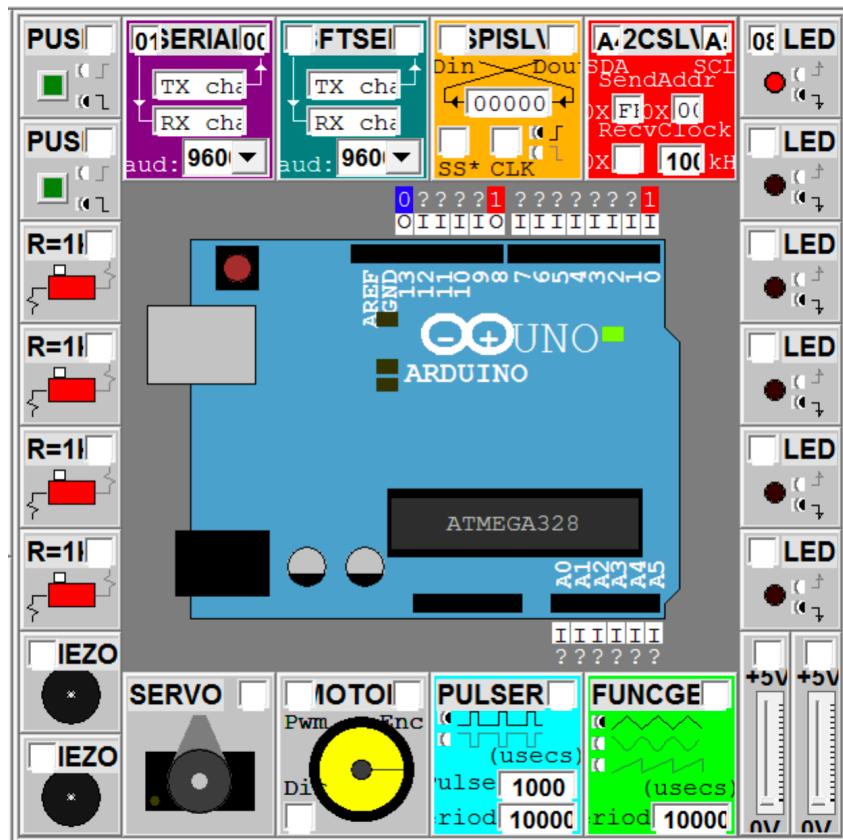


Veremos el LED asociado al pin 8 parpadear cada segundo:



Para parar la ejecución usaremos la opción Execute->Halt o el botón rojo de la barra de herramientas.

Finalmente, si queremos “unir” la salida del pin 8 con el LED que hay arriba del todo en la barra de herramientas de la derecha de la ventana, debemos escribir el número de pin (8 en este caso) en la pequeña ventana que tiene asociado ese LED. Veremos que al ejecutarlo, no sólo veremos cómo parpadea la marca del pin 8, sino que el LED también parpadea al mismo tiempo:



Adicionalmente, podemos encontrar en Internet muchos recursos para aprender a usar este simulador (ver los enlaces en la sección de referencias de este guión).

4. Simulador TinkerCad

TinkerCad Circuits (<https://www.tinkercad.com/circuits>) permite diseñar circuitos electrónicos a través de un navegador en línea. Se pueden crear circuitos propios desde cero o modificar plantillas y otros proyectos compartidos públicamente.

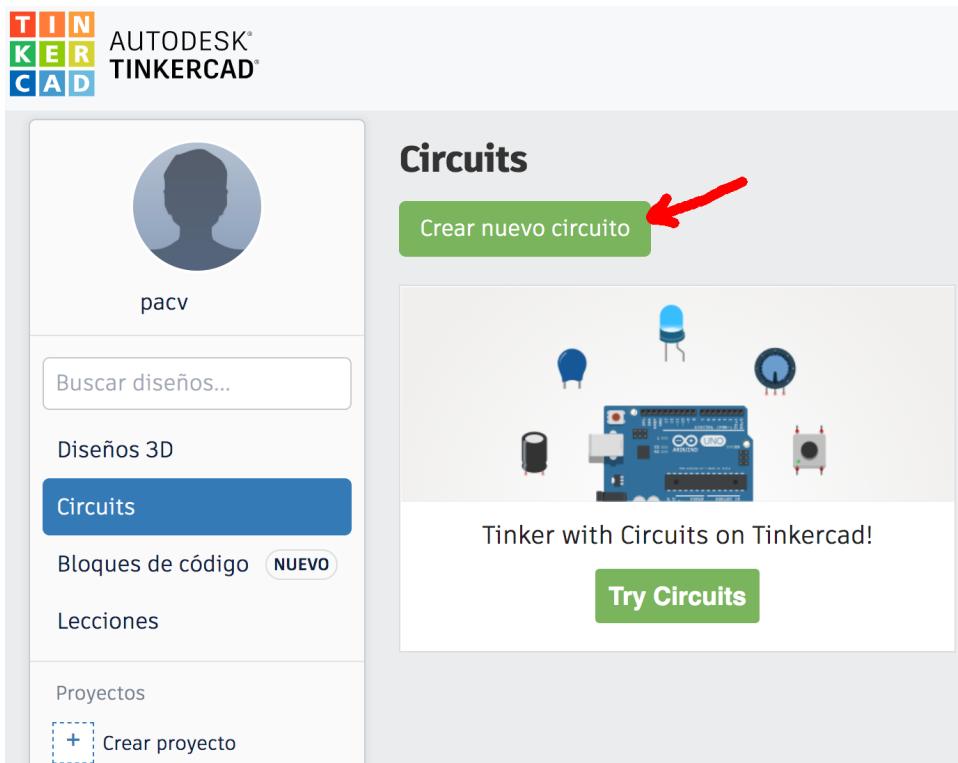
Su uso es gratuito, aunque para usarlo debemos registrarnos en:

<https://www.tinkercad.com>

y a continuación acceder al panel de control de los circuitos o las lecciones:

<https://www.tinkercad.com/dashboard?type=circuits&collection=designs>

<https://www.tinkercad.com/dashboard?type=all&collection=lessons>



Para comenzar a trabajar con un nuevo circuito, entraremos en:

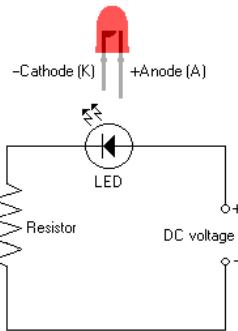
[Crear nuevo circuito](#)

Proyecto sencillo: Hacer parpadear un LED

Para crear el circuito más básico, en el que un LED conectado a un pin de salida parpadea, debemos seleccionar en la parte derecha los siguientes componentes y arrastrarlos a la zona de trabajo:

- Arduino Uno R3
- LED (color rojo)
- Resistencia (220 ohmios)

Debemos conectarlos como se muestra en el siguiente circuito eléctrico, y como se ve más abajo en la captura de pantalla del simulador:



Una vez conectados los componentes necesarios, debemos ir a la sección del “Código” para introducir las instrucciones del programa (no usaremos programación en bloques, sino el código escrito), tal y como vemos a continuación. Una vez escrito el programa podemos probarlo con el botón “Iniciar simulación” y deberíamos ver cómo parpadea el LED:

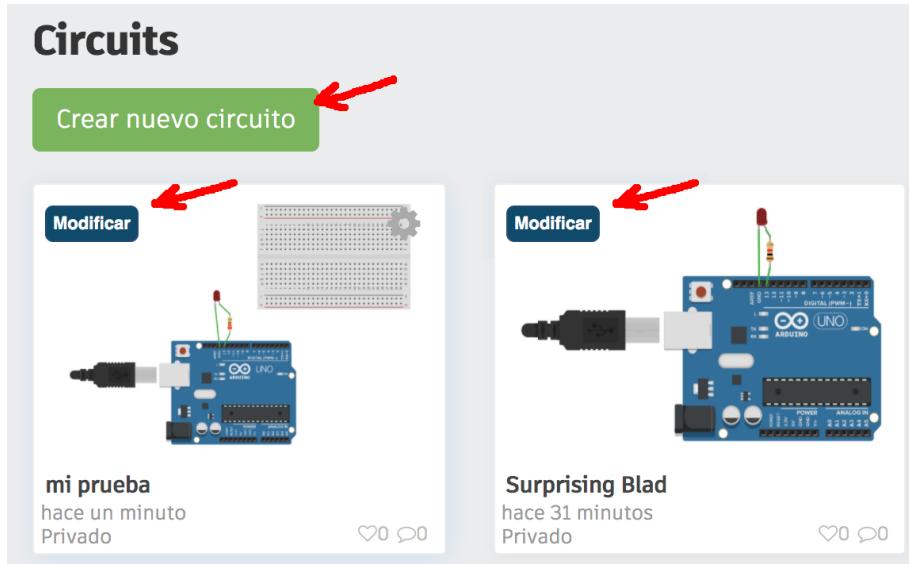
The screenshot shows the Fritzing software interface with the code editor open. The code is as follows:

```

Hora de simulador: 00:00:28
Código Detener simulación
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }

```

Una vez tengamos nuestro proyecto terminado, podemos volver al panel de control de los circuitos, donde veremos todos los que hemos construido. En cualquier momento podremos retomar cualquiera de nuestros circuitos y modificarlo mediante el botón azul correspondiente:



Tutoriales de uso de TinkerCad

Dentro de la plataforma se facilitan varios tutoriales, de diferente nivel de dificultad, para aprender a usar el simulador y construir todo tipo de prototipos. Para seguirlos debemos desplegar “Circuits” en el menú principal y elegir “Circuits” de entre las tres opciones:

The screenshot shows the Tinkercad website with the "Circuits" section selected. The main heading is "Aprende a usar Tinkercad" with the subtext "Mejora tus habilidades de diseño y creación". Below this, there are four tutorial projects listed: "Blink an LED With Digital circuits", "Multiple LEDs & Breadboard circuits", "Fading LED With Analog circuits", and "RGB LED Color Mixing circuits". Each project has a thumbnail image and a brief description.

Conviene comenzar con el tutorial “Project 0: Introduction” y seguir todos los pasos guiados que nos ofrecen:

tinkercad.com/learn/project-gallery;collectionId=OMOZACHJ9IR8LRE

Project 0: Introduction

Descripción

In this introduction to the Arduino Basic Kit projects, you will learn about the Arduino platform, how to set up your kit, and how to interact with the circuit simulator. These are the tools you need to create successful Arduino projects!

6 lecciones

- 1 - Welcome to Arduino
- 2 - Parts in the Kit
- 3 - Programming the Arduino (Simulator) Completada ✓
- 4 - Programming the Arduino (Kit) Completada ✓
- 5 - Build a Simple Circuit (Simulator) Completada ✓
- 6 - Prerequisites

Reiniciar proyecto

Project 0: Introduction
Build a Simple Circuit (Simulator)

Build a Simple Circuit (Simulator) Se han guardado todo

Código Inicia

Resistencia

Nombre: 1
Resistencia: 1 kΩ

Siguiente >

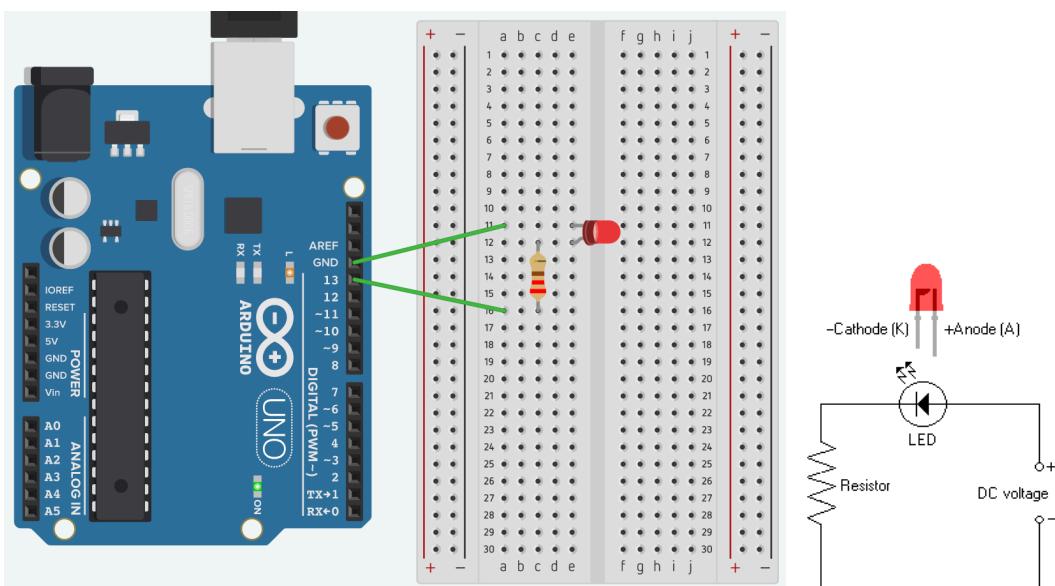
Adicionalmente, TinkerCad nos ofrece un conjunto de prototipos a modo de lecciones que podemos cargar, inspeccionar y modificar según nuestras necesidades:
<https://www.tinkercad.com/dashboard?type=all&collection=lessons>

Finalmente, y por supuesto, en Internet encontraremos muchos recursos, como videotutoriales detallados, que nos ayudarán a construir los prototipos que deseemos con este simulador (ver los enlaces en la sección de referencias de este guión).

Hacer parpadear un LED usando una placa de prototipado

En lugar de conectar los componentes como hemos visto anteriormente, los circuitos se suelen construir usando una placa de prototipado (*breadboard*). Puesto que en este tipo de placas es importante poder colocar los componentes en cualquier orientación, debemos recordar que pinchando sobre un elemento cualquiera y pulsando en la letra "R" repetidas veces podremos aplicar giros hasta colocar el elemento en la posición deseada.

A continuación se muestra el esquema para el proyecto que hace parpadear un LED, usando una *breadboard*:



Código del proyecto que hace parpadear un LED:

```

void setup() {
    pinMode(13, OUTPUT);           // código de inicialización
}                                     // usaremos el pin 13 como salida

void loop() {
    digitalWrite(13, HIGH);        // código del programa principal
    delay(1000);                 // encender el LED (voltaje a HIGH)
    digitalWrite(13, LOW);         // esperar encendido durante 1seg
    delay(100);                  // apagar el LED (voltaje a LOW)
}                                     // esperar apagado durante 100mseg
}

```

Cuestiones a resolver

El objetivo principal es conocer cómo utilizar varios simuladores de Arduino para crear varios programas y prototipos sencillos.

El estudiante debe llevar a cabo los siguientes proyectos de Arduino:

- Implementar en los simuladores el programa de parpadeo de LED, ampliándolo para que encienda y apague alternativamente dos LED (uno rojo y otro verde), conectados a las salidas digitales 12 y 13 del Arduino, a un intervalo de 1.5 segundos.
- Implementar en los simuladores el programa de parpadeo de LED, ampliándolo con las modificaciones necesarias para que se encienda el LED solo cuando se pulse un interruptor conectado a la entrada digital 7.

Como resultado se mostrará al profesor el funcionamiento de cada archivo propuesto.

En el documento a entregar se describirá cómo se han creado los diferentes ejemplos y se mostrará el correcto funcionamiento con capturas de pantalla.

Normas de entrega

La práctica o seminario podrá realizarse de manera individual o por grupos de hasta 2 personas.

Se entregará como un archivo de texto en el que se muestre la información requerida. También se puede utilizar la sintaxis de Markdown para conseguir una mejor presentación e incluso integrar imágenes o capturas de pantalla. La entrega se realizará subiendo los archivos necesarios al repositorio “**PDIH**” en la cuenta de GitHub del estudiante, a una carpeta llamada “**S-arduino**”.

Toda la documentación y material exigidos se entregarán en la fecha indicada por el profesor. No se recogerá ni admitirá la entrega posterior de las prácticas/seminarios ni de parte de los mismos.

La detección de copias implicará el suspenso inmediato de todos los implicados en la copia (tanto de quien realizó el trabajo como de quien lo copió).

Las faltas de ortografía se penalizarán con hasta 1 punto de la nota de la práctica o seminario.

Referencias

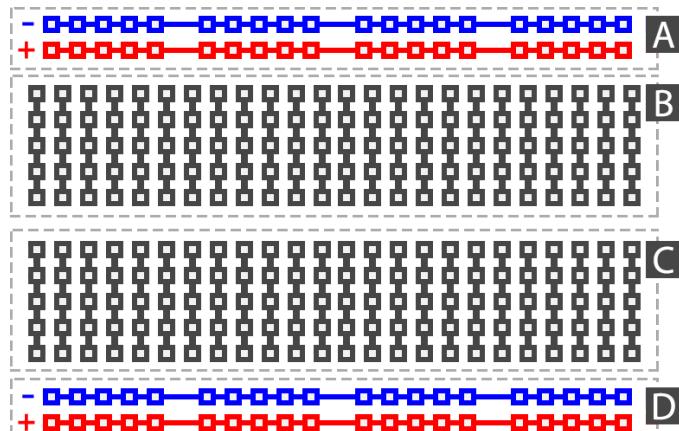
- Arduino Website. Arduino - HomePage. Disponible en: <http://arduino.cc>
J. R. Lajara Vizcaíno and J. Pelegrí Sebastià, Sistemas integrados con Arduino. Barcelona: Marcombo, 2014.
Rugged circuits. 10 Great Ways to Destroy an Arduino. Disponible en: <http://www.ruggedcircuits.com/10-ways-to-destroy-an-arduino>
A. G. González. (2014). Un documental sobre la Historia de Arduino. Disponible en: <http://panamahitek.com/un-documental-sobre-la-historia-de-arduino/>
Cooking Hacks. (2014). Cooking Hacks. Packs Kits. Lab Kit. Arduino Kit. Disponible en: <http://www.cooking-hacks.com/arduino-lab-kit>
<http://www.maelabs.ucsd.edu/mae156alib/electronics/DestroyArduino.pdf>
<http://www.trastejant.es/blog/?p=192>
<http://mecabot-ula.org/tutoriales/arduino/pratica-2-encender-y-apagar-un-led-utilizando-un-boton-pulsador/>
<http://mecabot-ula.org/tutoriales/arduino/practica-10-encender-un-led-con-boton-pulsador-y-luego-apagarlo-con-el-mismo-boton/amp/>
<https://www.arduino.cc/en/Tutorial/Button>

- <https://www.arduino.cc/en/Tutorial/StateChangeDetection>
- <https://www.tinkercad.com/circuits>
- <https://youtu.be/6uz1sCA9joc>
- <http://soloelectronicos.com/2018/04/21/disene-y-simule-circuitos-electronicos-facilmente-con-tinkercad/>
- <https://www.sites.google.com/site/unoardusim/services>
- <https://tallerelectronica.com/2015/04/11/unoardusim-software-de-simulacion-de-arduino-uno/>
- <https://felixmaocho.wordpress.com/2016/06/12/curso-de-arduino-aprender-a-utilizar-el-simulador-unoardusim/>
- <https://youtu.be/j9cMCjZjMo>
- https://www.arduinando.com/tutoriales_arduino/
- <https://www.instructables.com/Arduino-Serial-Monitor-in-Tinkercad/>

Apéndice

Para construir prototipos se suele usar una placa de prototipado (*breadboard*) en la que se pueden montar todos los componentes del circuito. Es importante conocer cómo están interconectados los conectores en la placa.

En la siguiente figura se muestran las bandas de alimentación (A y D) y las columnas de interconexión (B y C):



Además, en todo circuito tendremos que usar varios componentes, entre ellos, resistencias. El valor en ohmios es crítico, ya que se podría dañar la placa de usar otro valor de resistencia. A continuación se muestra la codificación de las resistencias mediante cuatro bandas de color:

