

Práctica 2: Arduino UNO + LED

Llegadas a este punto, estamos listas para conectar nuestro Arduino al ordenador. Para ello, necesitaremos nuestro cable USB-AB.

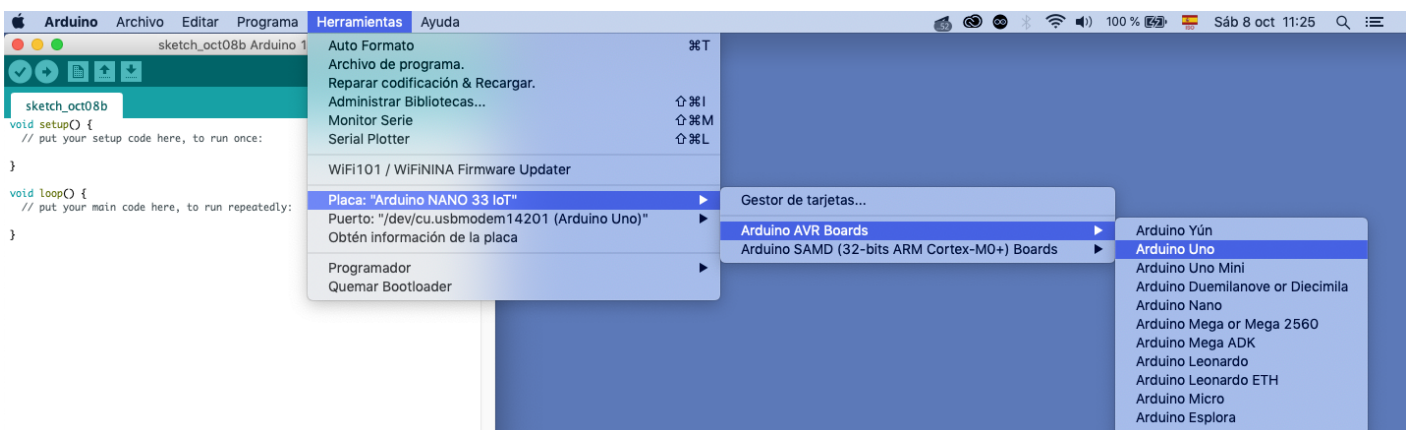
Una vez hayamos conectado un extremo a nuestro Arduino y el otro al puerto USB de nuestro ordenador, tendremos que ayudar a la IDE a que reconozca a nuestro Arduino. Lo primero que veremos será un **LED VERDE** que se enciende y junto al que pone **ON**. Eso nos indica que la placa está recibiendo corriente desde nuestro ordenador.

Los usuarios de Windows (seguramente) tendrán que instalar una serie de drivers una vez se conecte el Arduino al ordenador.

Lo siguiente que tendremos que hacer es ir a la barra de herramientas superior y precisamente en **Herramientas** tendremos que configurar dos cosas, **la placa y el puerto**.

Configuramos la placa

En **Herramientas > Placa** nos aparecerá nuestro Arduino UNO, tendremos que seleccionarlo

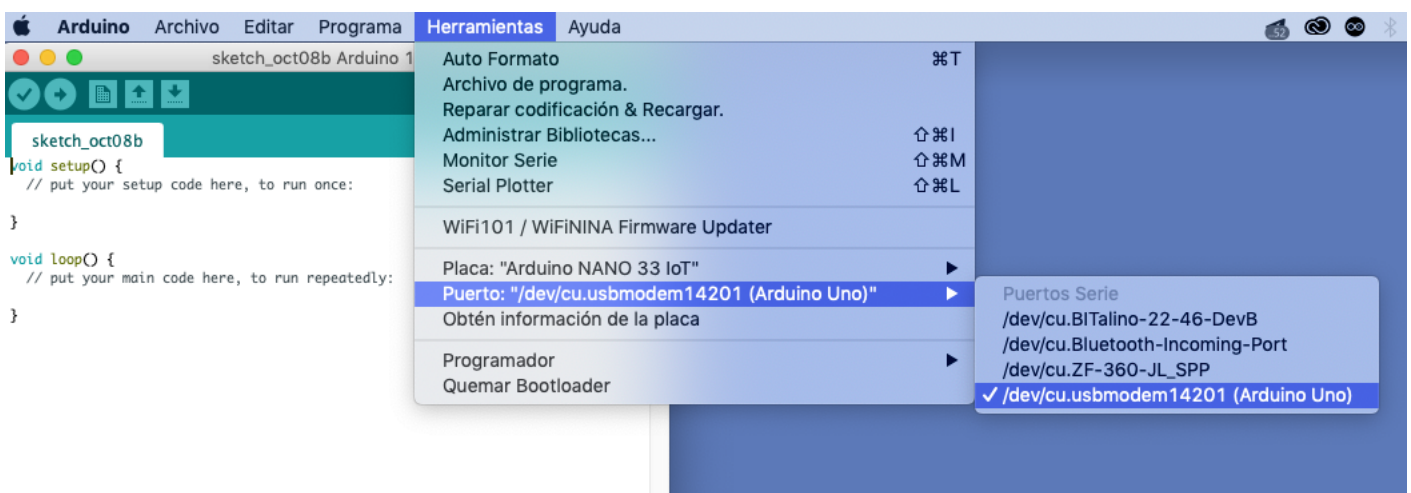


Al seleccionarla, le estamos diciendo a nuestro Arduino "¡Hey! Prepárate, voy a conectarte un Arduino UNO", por lo que si al final conectásemos otro tipo de Arduino, la IDE no la reconocería y no podríamos programarlo.

Si utilizamos otro tipo de Arduino, lo primero que tenemos que hacer es decírselo a Arduino seleccionando la placa correcta en **Placa**, como acabamos de hacer.

Configuramos el puerto

Una vez le hemos dicho la placa que vamos a conectar, nuestro IDE necesita saber en qué puerto USB la hemos conectado. Para ello, tendremos que ir a **Herramientas > Puerto**



Para los usuarios de **Windows** podría tener esta apariencia:

```
<COM29> (Arduino UN0)
```

Para los de **MAC** o **Linux**, la apariencia puede ser parecida a esta:

```
/dev/cu.usbmodem14112 (Arduino UN0)
```

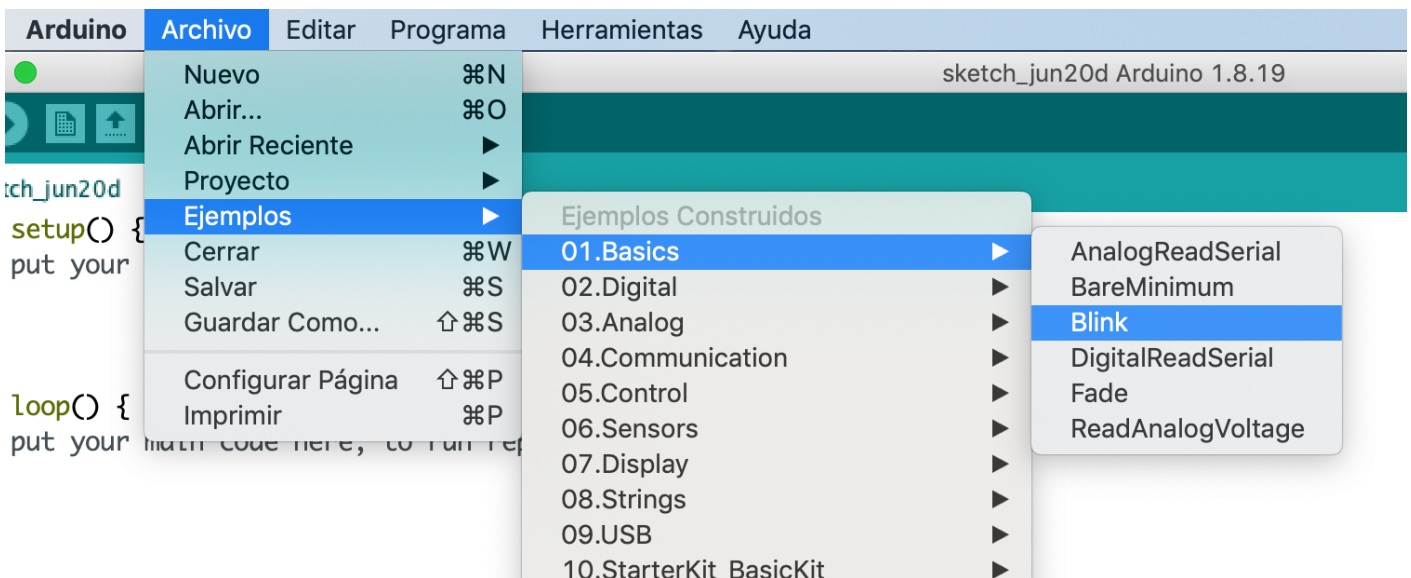
Una vez hayamos seleccionado nuestra placa y puerto, pasaremos a encender el LED.

Y se hizo la luz

Para encender nuestro LED, vamos a utilizar un ejemplo de los que se nos proporcionan desde la propia IDE de Arduino.

Estos ejemplos nos pueden ser útiles para descubrir lo que podemos hacer con estas placas, por lo que os recomiendo que les echéis un vistazo.

El ejemplo que vamos a utilizar se llama **Blink** y nos va a venir bien para asegurarnos de que la instalación y toda la configuración se han realizado correctamente. Para abrir y subir este ejemplo a nuestro Arduino tenemos que navegar a: **Archivo > Ejemplos > 01.Basics > Blink**.



Una vez hagamos click, se abrirá una nueva ventana de Arduino con el código que permitirá que nuestro LED comience a parpadear. El código que veremos será el siguiente:

```
/*  
Blink  
  
Turns an LED on for one second, then off for one second, repeatedly.  
  
Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO  
it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to  
the correct LED pin independent of which board is used.  
If you want to know what pin the on-board LED is connected to on your Arduino  
model, check the Technical Specs of your board at:  
https://www.arduino.cc/en/Main/Products
```

modified 8 May 2014

by Scott Fitzgerald

modified 2 Sep 2016

by Arturo Guadalupi

modified 8 Sep 2016

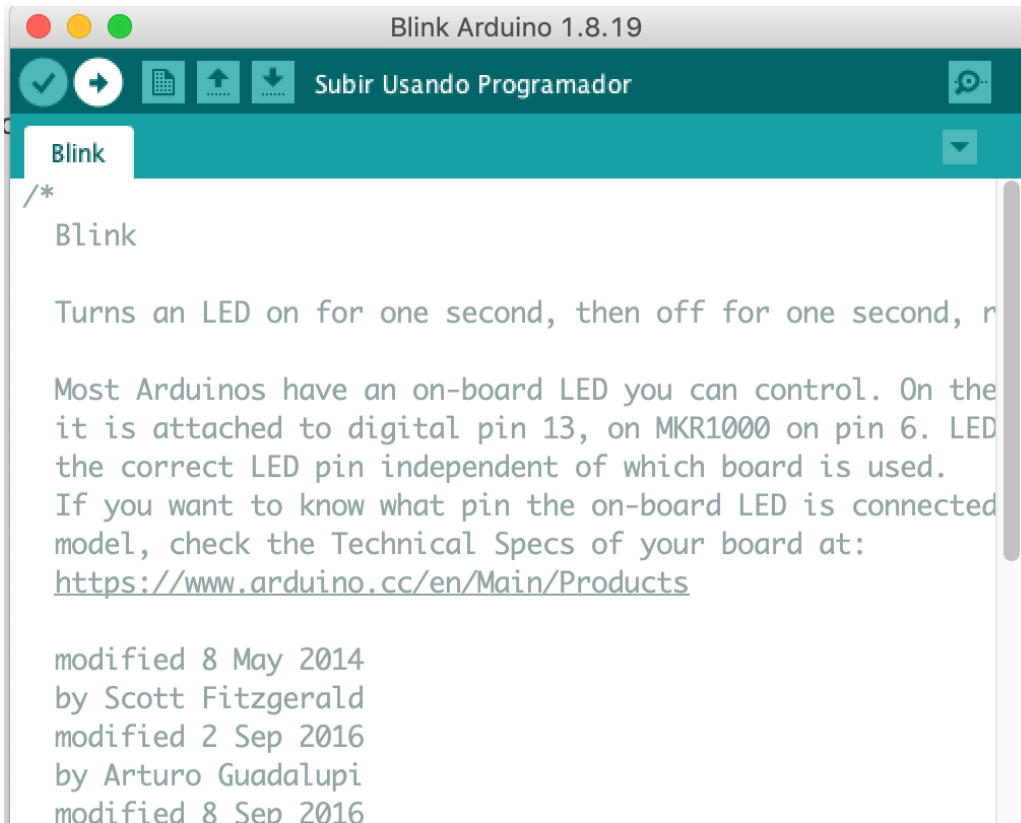
by Colby Newman

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>

```
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)  
  delay(1000);                       // wait for a second  
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW  
  delay(1000);                       // wait for a second  
}
```

Ahora, no vamos a entrar a explicar qué significa este código, eso lo haremos en la página siguiente. Lo que sí vamos a hacer es darle a la **flecha a la derecha** para subir ese código a nuestro UNO:

A screenshot of the Arduino IDE interface. The title bar reads "Blink Arduino 1.8.19". The menu bar includes "Subir Usando Programador". The "Blink" sketch is open, showing a code editor with the following text:

```
/*  
Blink  
  
Turns an LED on for one second, then off for one second, r  
  
Most Arduinos have an on-board LED you can control. On the  
it is attached to digital pin 13, on MKR1000 on pin 6. LED  
the correct LED pin independent of which board is used.  
If you want to know what pin the on-board LED is connected  
model, check the Technical Specs of your board at:  
https://www.arduino.cc/en/Main/Products  
  
modified 8 May 2014  
by Scott Fitzgerald  
modified 2 Sep 2016  
by Arturo Guadalupi  
modified 8 Sep 2016
```

Y si todo ha ido bien, lo que veremos será un LED naranja, cerca del conector USB, que permanecerá un segundo encendido, un segundo apagado.

¡Y ya lo tenemos!



Es posible que nos de algún error y que no funcione. Eso no significa que algo esté roto. Como ya avisé en la primera página del taller, no siempre funciona todo a la primera. Si en la consola vemos algún error, tendremos que asegurarnos de que hemos seleccionado el

puerto y la placa correctas y que hemos seguido todos los pasos correctamente, sin olvidar ninguno.

¿Qué tenemos que entregar?

Tendrás que subir un video o .gif en el que se vea tu Arduino conectado al ordenador con el LED parpadeando y luego la pantalla de tu ordenador con el sketch abierto.

FUENTES:

Basado en: <https://libros.catedu.es/books/microcontroladores-vestibles-y-conectados-a-internet/page/practica-11-descargar-la-ide-de-arduino-y-encender-un-led>

Revision #19

Created 2022-10-01 11:16:00 CEST by Marta P. Campos

Updated 2022-11-29 13:46:10 CET by Marta P. Campos