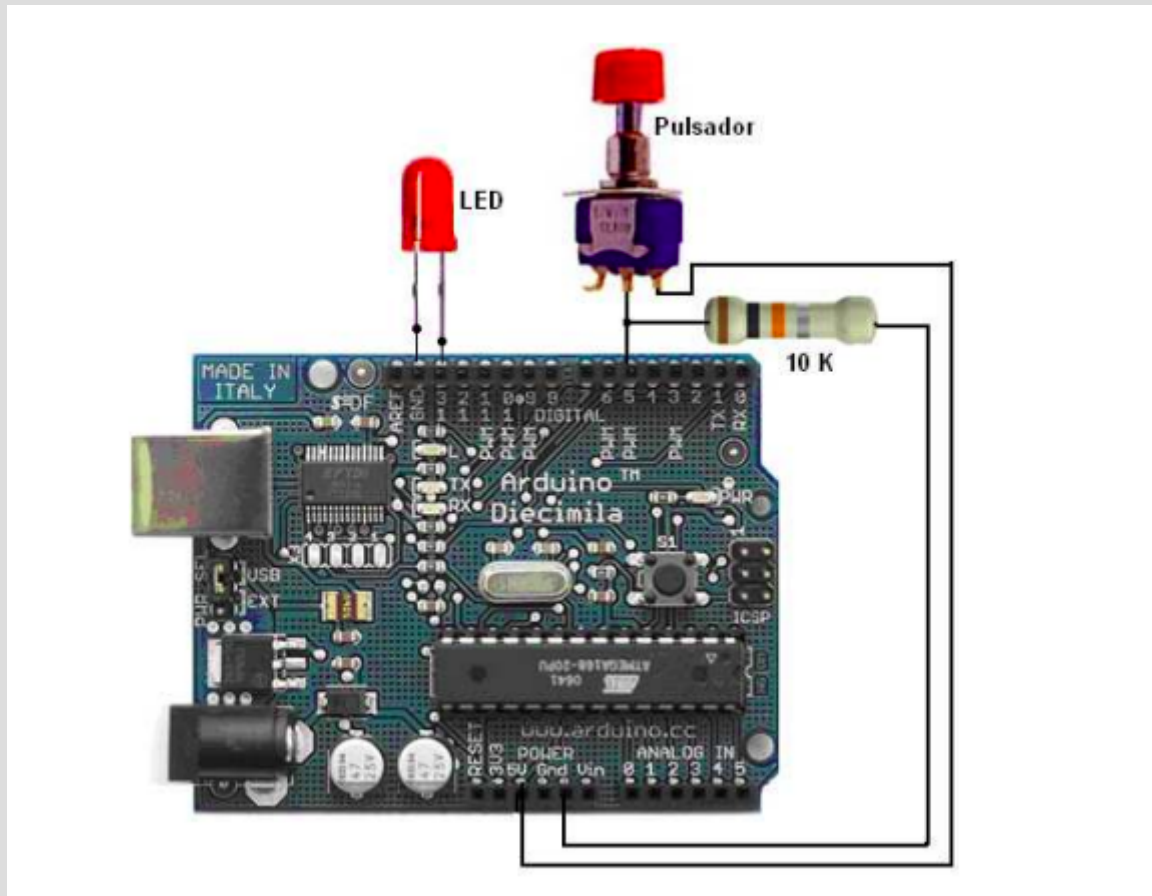


# Ejercicios de programación básica

PROF. LIZANDRO RAMIREZ

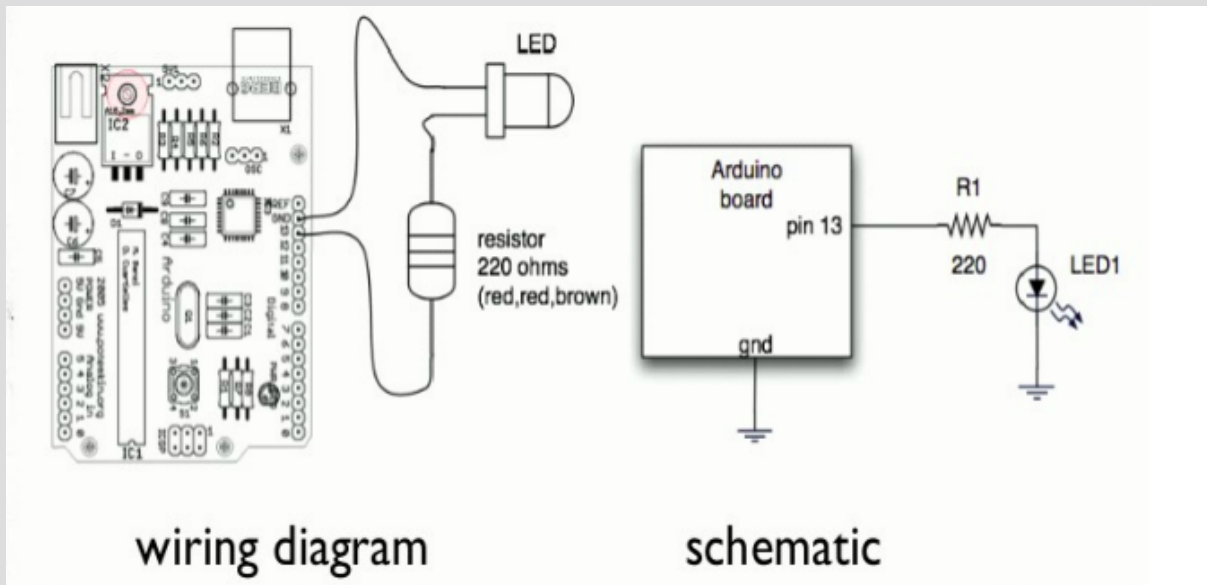
Nota: 1000 ms == 1 Segundo



1. Led parpadeante. Se trata de conectar un led al pin 13, haciendo que luzca durante 500 ms y que se apague durante 100 ms, este proceso se repetirá cíclicamente.

## Objetivos:

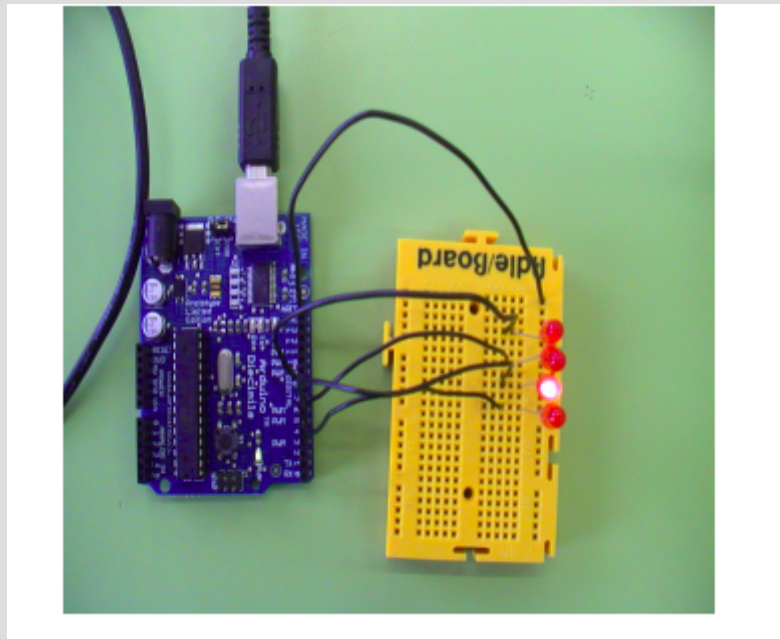
- Reconocer partes de la placa.
- Aprender a conectar leds a la placa.
- Familiarizarse con el entorno de programación.
- Reconocer las partes de un programa de arduino.
- Conocer órdenes como: pinMode, digitalWrite y delay



2. 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.


#### Objetivos:

- Familiarizarse con el entorno de programación.
- Aprender a declarar variables y variables tipo lista de valores.
- Aprender a declarar una función y llamarla cuando sea necesario.



3. En una segunda solución para las luces del ejercicio #2, la secuencia principal del programa debe estar reproducida en una función a la que llamará el programa principal. Crear distintas secuencias de encendido para el juego de luces (usar 8 Leds), una secuencia en reversa, todos apagan-encienden juntos cada un segundo (3 vcs) y una secuencia de salto con un led de diferencia.

- Video de ejemplo:

 Secuencial de leds con Arduino | 8 Secuencias, Explicacion paso a paso

4. 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.

#### **Objetivos:**

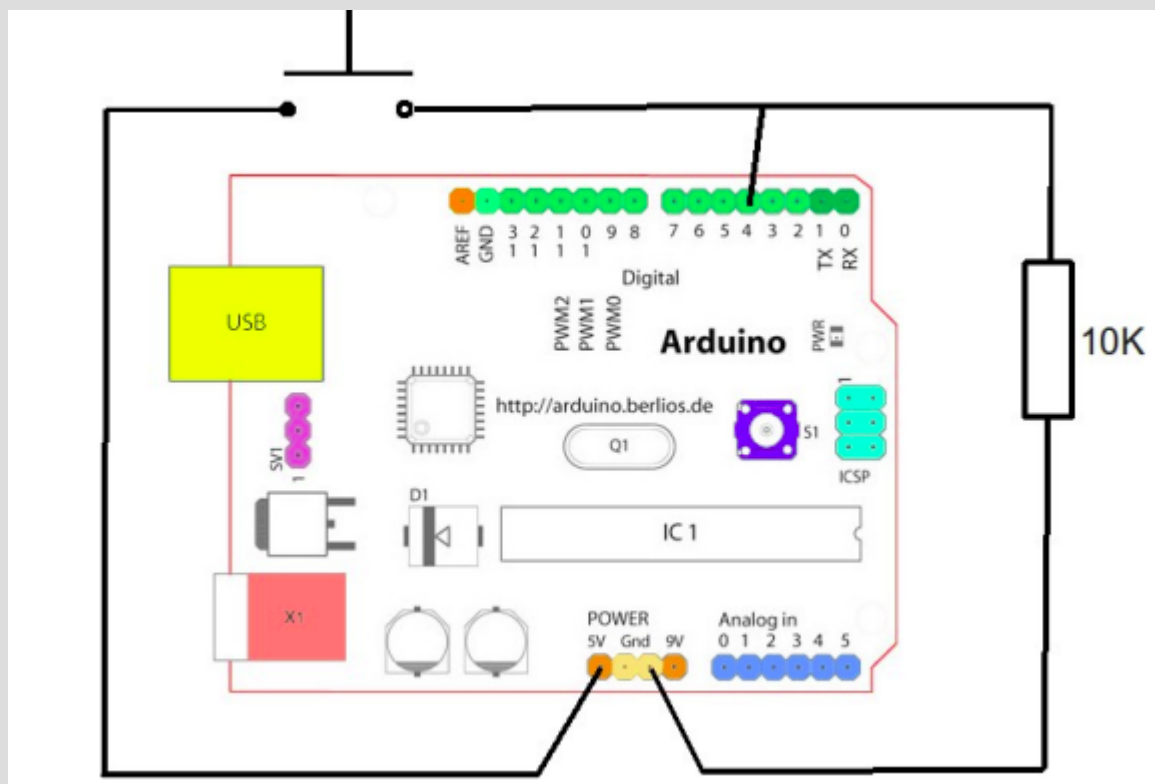
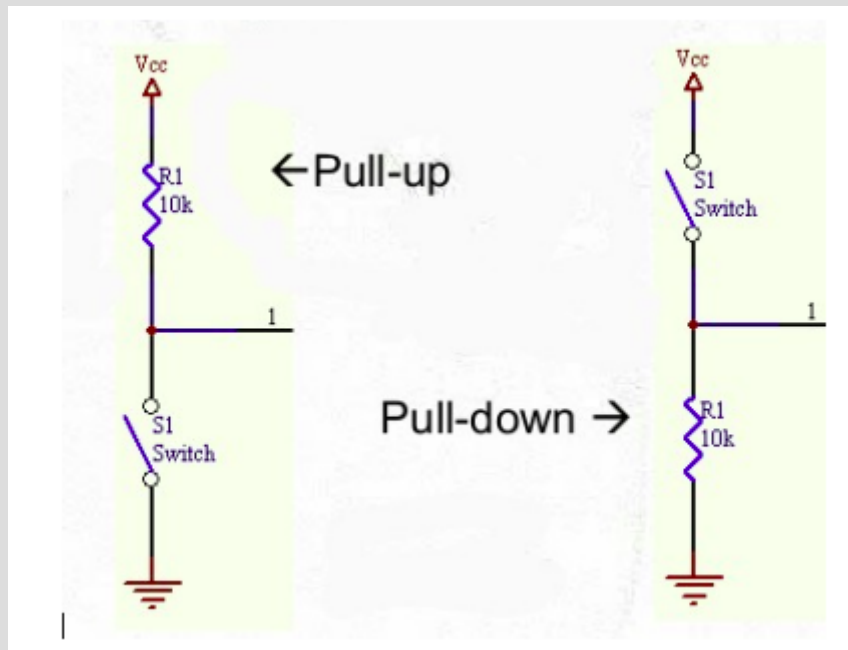
- Familiarizarse con el entorno de programación.
  - Aprender a declarar variables tipo lista de valores.
5. Se trata de encender y apagar 7 leds secuencialmente. Los leds deben estar conectados a los pines 5,6,7,8,9,10 y 11. Se deben encender y apagar los leds desde el pin 5 al 11, con un tiempo de encendido y apagado de 50 ms, más tarde se deben encender y apagar los leds desde el pin 11 al 5, con un tiempo de encendido y apagado de 50 ms. La secuencia se debe repetir indefinidamente. El efecto del programa es el de las luces delanteras de nuestro querido "Coche fantástico".

#### **Objetivos:**

- Familiarizarse con el entorno de programación.
  - Repasar declaración de variables tipo lista de valores.
  - Repasar órdenes de control de programa como: for.
6. 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.

#### **Objetivos:**

- Familiarizarse con el entorno de programación.
- Aprender a conectar una entrada digital a arduino (pulsador).
- Aprender a declarar variables tipo lista de valores
- Aprender a declarar una función y llamarla cuando sea necesario.
- Conocer órdenes como: **digitalRead**.
- Conocer órdenes de control de programa como: **If**.



7. Ruleta de la fortuna. Se trata de cinco leds que se van encendiendo y apagando formando una secuencia, el jugador debe dar al pulsador cuando el led intermedio se enciende, si acierta funciona un zumbador y la velocidad de la secuencia aumenta. Los leds deben estar conectados de los pines 5 a 9, el zumbador al pin 10, el pulsador al pin 11. El tiempo inicial entre encendido y encendido de leds debe ser

200 ms, si se acierta se decrementa el tiempo en 20 ms, si el tiempo entre encendidos llegase a 10 ms, se devuelve el tiempo a 200 ms. Objetivos:

- a. Repaso de conexión de entrada digital a arduino (pulsador).
- b. Repaso de variables tipo lista de valores.
- c. Repaso de declarar una función y llamarla cuando sea necesario.
- d. Repaso de órdenes como: `digitalRead`.
- e. Repaso de órdenes de control de programa como: `For`, `If`.

8. Se trata de un dispositivo que haga lucir un led más o menos en función del valor de un potenciómetro. Para ello conectaremos el potenciómetro a la entrada analógica 0 y un led al pin 9. Cuando el potenciómetro se encuentre entre 0 y 512 el led debe colocarse en el nivel de potencia máxima (255), si el potenciómetro se encuentra entre valores 512 y 1024 el led debe lucir al nivel de potencia 64. Además, se deberá mostrar el valor de voltaje en la entrada analógica (valor entre 0 y 1024) en una consola en el PC.

Objetivos:

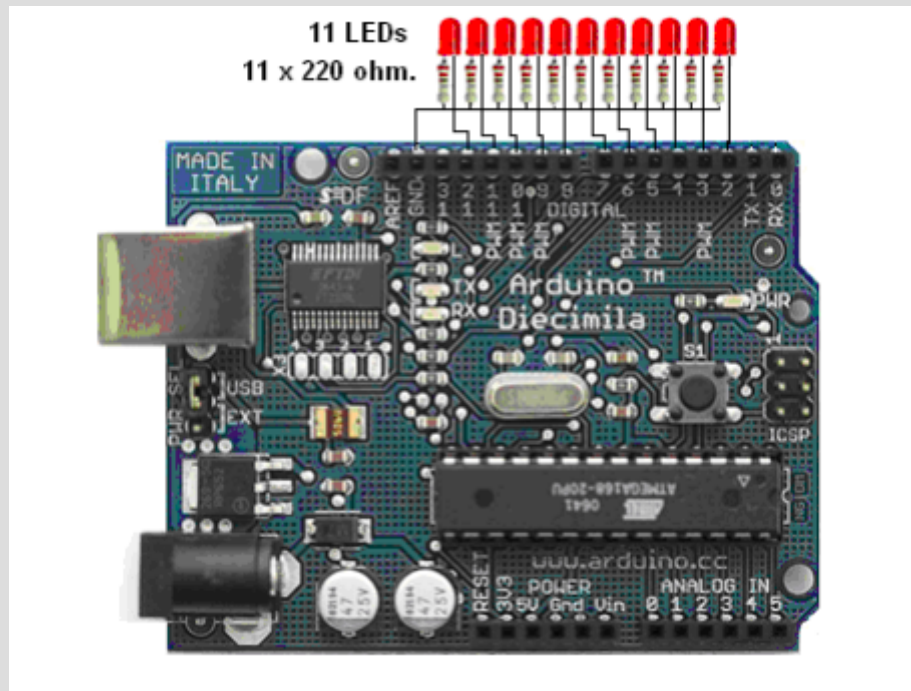
- Repaso conexión de entrada analógica a arduino (`ldr`).
- Conexión de salidas analógicas.
- Órdenes como: `analogWrite`.
- Repaso de visualizar datos en consola de puerto serie, con órdenes como: `Serial.begin`, `Serial.print`.
- Repaso de órdenes de control de programa como: `If else`.

9. Este ejercicio muestra como realizar un rayo de luz, o más poéticamente, una estrella fugaz, moviéndose a través de una línea de LED-s. Podremos configurar tanto la velocidad de la estrella, así como la longitud de la cola. No es muy elegante porque la cola brilla con la misma intensidad que la estrella, y al final, parecerá como si un rayo sólido cruzase la línea de LED-s.

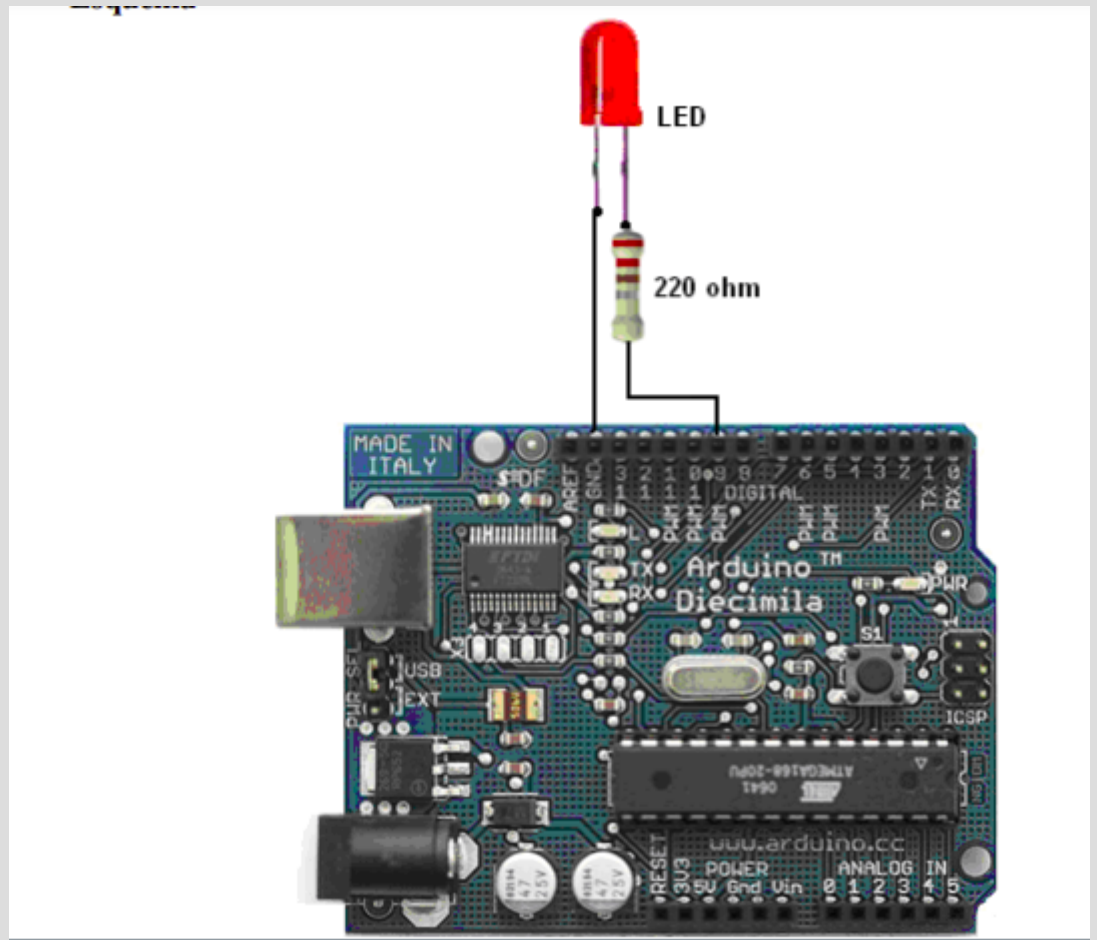
El programa comienza encendiendo LED-s hasta que llegue al número de LED-s establecido para la cola. En ese momento seguirá encendiendo LED-s hacia la izquierda (si se monta tal y como se muestra en la fotografía inferior), para mantener el movimiento de la estrella, al mismo tiempo que apaga LED-s por la derecha, para asegurarnos de que vemos la cola. De otra forma seguiría encendiendo LED-s hasta encenderlos todos. Esto ocurre cuando el tamaño de la cola es igual o mayor que el número de LED-s. El tamaño de la cola debería ser relativamente pequeño en comparación con el número de LED-s de forma que podamos ver la estrella.

Elementos necesarios:

- 11 LED-s.
- 11 resistencias de 220 Ohmios.
- Una placa protoboard.
- Cables para efectuar las conexiones.

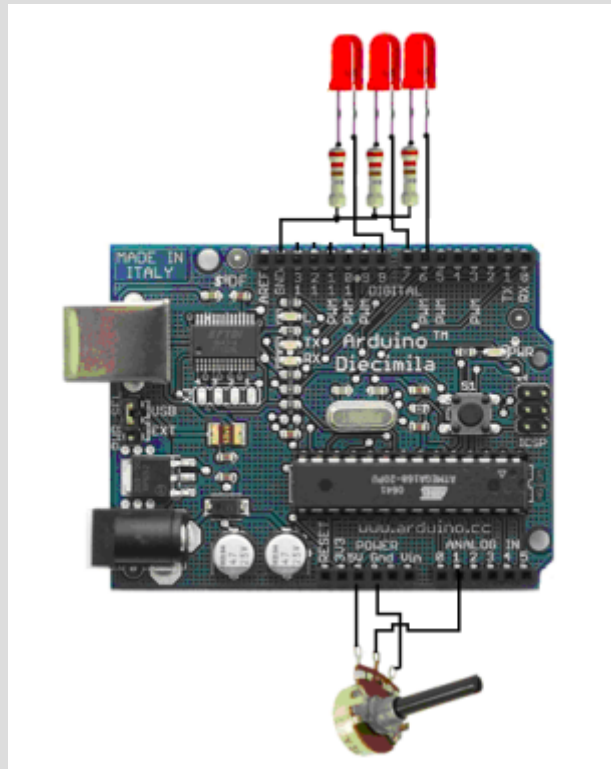


10. Se trata de simular el movimiento de la llama de una vela. Hacemos uso de la instrucción para generar un número aleatorio que lo asignamos a una salida analógica PWM y otro número que lo asociamos a la variable de temporización (tiempo que esperamos para cambiar el valor de la salida).
- a. Números aleatorios en Arduino  
<https://programarfacil.com/blog/random-en-arduino-como-usar-numeros-aleatorios/>
  - b. PWM en arduino:  
<https://programarfacil.com/blog/arduino-blog/pwm-con-arduino-analogico/>



11. Se trata de construir u indicador de nivel que sea capaz de medir el valor de una señal de entrada generada por un potenciómetro desde una entrada analógica. Se establecerán 3 diodos Led conectados a las salidas PIN6, PIN7 y PIN8. La entrada la conectaremos en la entrada analógica PIN 1 (analog A1)





12. Se trata de generar hasta 8 notas musicales por una de las salidas analógicas de Arduino –PIN10–

Se debe crear un array (vector) de datos compuesto por los valores correspondientes a las 8 notas que se pretende sacar:

```
int notas[] = {1915, 1700, 1519, 1432, 1275, 1136, 1014, 956};
```

Se deben definir también el tiempo de pausa entre nota y nota y el tiempo de pausa de fin de secuencia de notas:

```
int tnota=100; int pausa=100;
```

Las iteraciones para el recorrido de las 8 notas se realizan con una instrucción de tipo for:

```
for(n=0;n<8;n++)
```

El tiempo de activado y desactivado de la salida del zumbador también se resuelve con un bucle for:

```
for(m=0;m<=tnota;m++)
```

Arduino conectar buzzer:

<https://blog.330ohms.com/2020/06/02/como-conectar-un-buzzer-pasivo-a-arduino/>



