MÓDULO 4: PROJECTO 4 "Lámpara mezcla de colores"

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS

Introducción

- Arduino no puede variar el valor de la tensión de salida en sus pins, tan solo puede suministrar un voltaje de 5V.
- Por consiguiente, necesitaremos utilizar una técnica llamada **Modulación del Ancho de Pulso (MAP)** (Pulse Width Modulation en inglés) para poder disminuir el brillo de un led.
- La técnica MAP cambia rápidamente el valor del pin de salida entre High y Low en un lapso de tiempo determinado.
- Dicho cambio se produce más rápido de lo que el ojo humano puede detectar, tal y como funcionan las películas: mostrando al ojo una serie de imágenes fijas de forma tan rápida que da la sensación de haber movimiento.

Montando el circuito

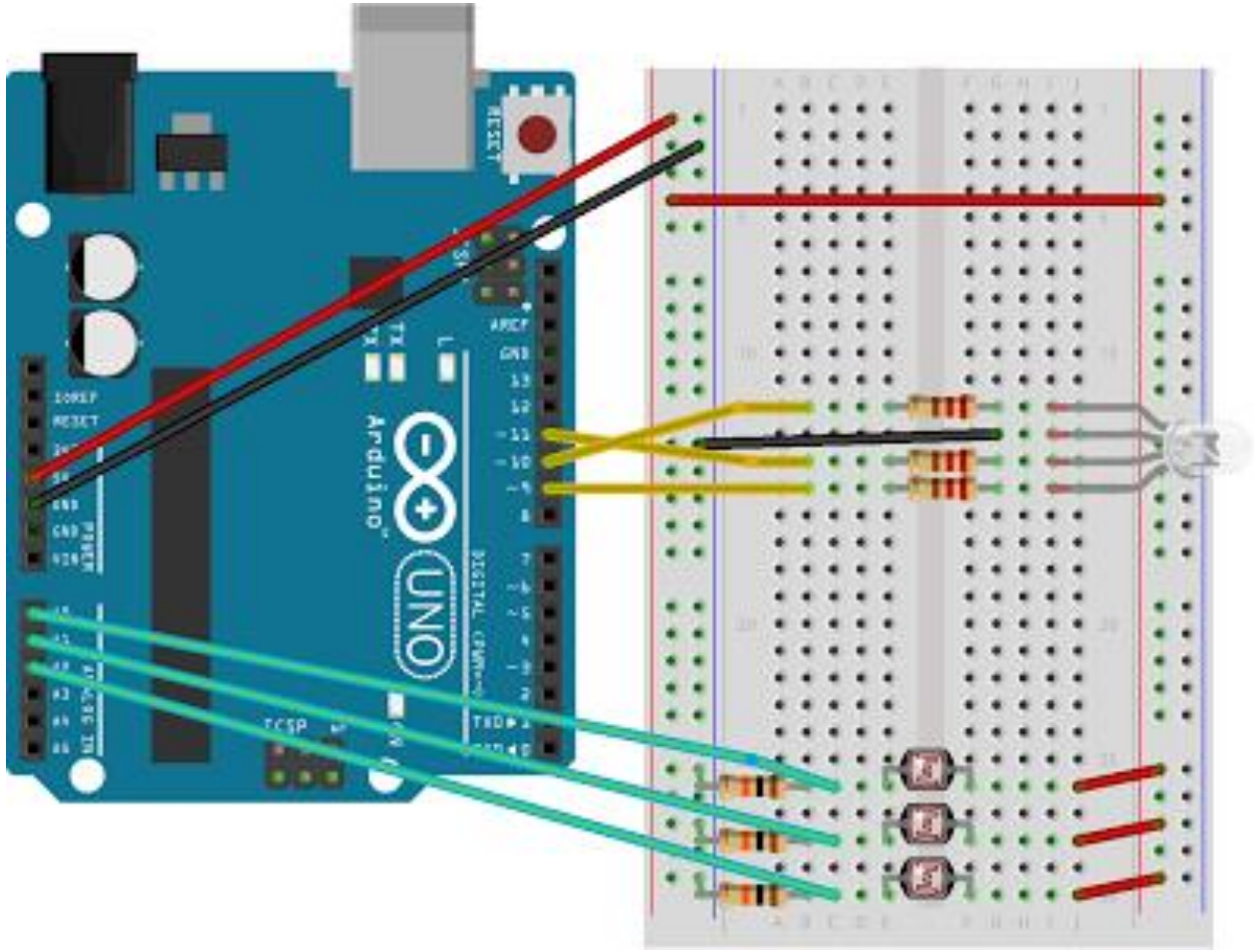
1. Tal y como hemos hecho en los proyectos anteriores, conectamos la alimentación y la tierra en nuestra protoboard.
2. Colocamos las tres fotorresistencias en la protoboard a lo largo de la línea central de la misma, de manera que una patilla quede a un lado de dicha línea y la otra al otro lado.
3. De las dos patillas que tiene cada fotorresistencia, una la conectamos a la alimentación y la otra, a una resistencia de 10KΩ que a su vez irá conectada a tierra. Como dicha resistencia de 10KΩ se encuentra en serie con la fotorresistencia, en conjunto forman un divisor de tensión. Cuando el valor de la fotorresistencia varíe a causa de la incidencia de la luz, así lo hará también el valor de la tensión entre ambas resistencias.
4. A continuación, la patilla de cada fotorresistencia que se encuentra conectada a la resistencia de 10KΩ, la uniremos con un cable a los pins de entrada analógica 0, 1 y 2 (una fotorresistencia por cada pin).

Montando el circuito

5. Seguidamente, cogemos los tres trozos de papel o plástico de colores y colocamos uno sobre cada fotorresistencia. Ponemos el de color rojo sobre la fotorresistencia conectada al pin A0, el verde sobre la que está conectada al pin A1 y el azul, sobre la fotorresistencia conectada al pin A2. Cada uno de estos plásticos solo deja pasar la luz de una determinada longitud de onda. El rojo solo deja pasar el rojo, el de color verde la luz verde y el azul solo la luz azul. De esta manera, cada fotorresistencia solo detectará un determinado tipo de luz.
6. El siguiente componente que vamos a utilizar es el led RGB. Este tipo de led tiene cuatro patillas: una para el color rojo, una para el color verde, una para el color azul y otra que sirve de tierra para las otras tres (el cátodo). Cada una de las patillas estará conectada a una resistencia de 220Ω (a excepción del cátodo).

Montando el circuito

Componentes	
LED RGB	1
Fotorresisténcias LDR	3
Resistencias 220 Ω	3
Resistencias 10 K Ω	3



Comentario del código

- Establecemos las constantes const para los pines que se utilizarán para las salidas del LED y los pines para las entradas de las fotorresistencias. Utilizamos el tipo de dato int.
- Añadimos las variables de los valores entrantes de los sensores y los valores de salida que utilizarán para apagar gradualmente el LED.
- En la función **setup**() comienza la comunicación con el monitor serie a *9600 bps* donde se verán los valores de los sensores. También, se definen los pines del LED como salidas con pinMode().
- En la función **loop**() se leen los valores de los sensores en A0, A1 y A2 con analogRead() y se almacenan en las variables apropiadas. Hay que poner un pequeño delay() entre cada lectura ya que el *ADC* toma unas milésimas de segundo para funcionar correctamente.

Comentario del código

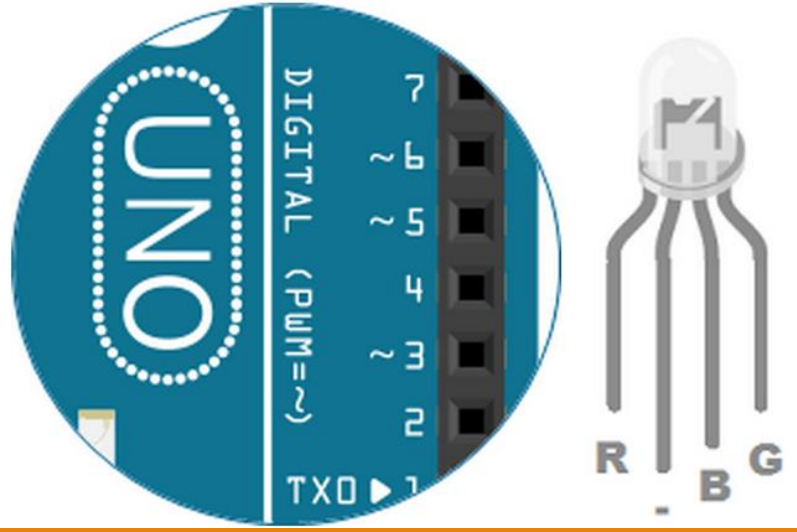
- Establecemos las constantes const para los pines que se utilizarán para las salidas del LED y los pines para las entradas de las fotorresistencias. Utilizamos el tipo de dato int.
- Añadimos las variables de los valores entrantes de los sensores y los valores de salida que utilizarán para apagar gradualmente el LED.
- En la función **setup**() comienza la comunicación con el monitor serie a *9600 bps* donde se verán los valores de los sensores. También, se definen los pines del LED como salidas con pinMode().
- En la función **loop**() se leen los valores de los sensores en A0, A1 y A2 con analogRead() y se almacenan en las variables apropiadas. Hay que poner un pequeño delay() entre cada lectura ya que el *ADC* toma unas milésimas de segundo para funcionar correctamente.

Comentario del codigo

- Ahora, imprimimos los valores de los sensores **Serial**.print() en una línea. \t es el equivalente a presionar el tabulador en el teclado.

- Arduino no puede variar la tensión de salida en sus pines, sólo puede dar 5V de salida (completos). Por lo tanto, deberás utilizar una técnica llamada ***Modulación por ancho de pulsos (PWM)*** para apagar gradualmente los LEDs con analogWrite().

- *PWM* convierte rápidamente el pin de salida a alta y baja tensión durante un período de tiempo fijo. El cambio se produce más rápido que lo que el ojo humano puede ver.

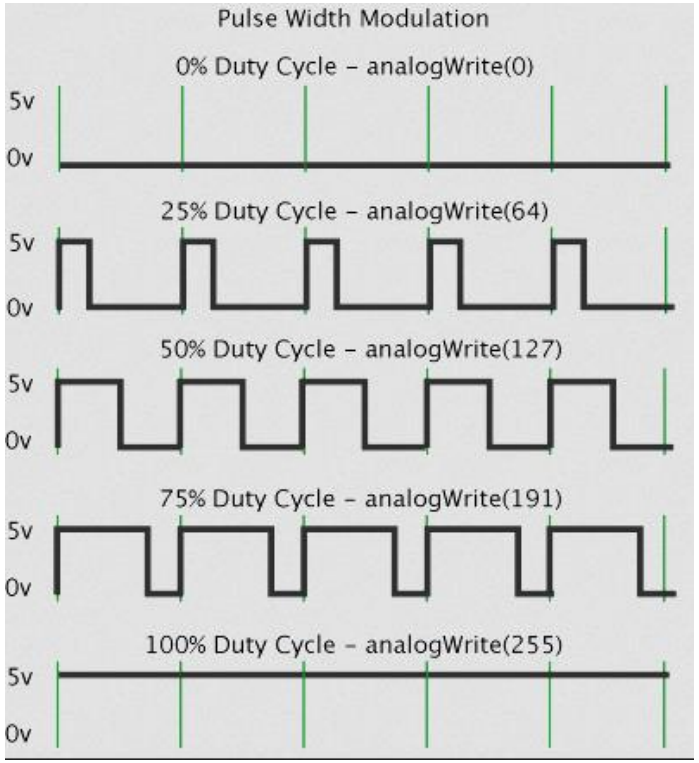


Comentario del codigo

- La función para cambiar el brillo del LED vía PWM es analogWrite(). Tiene dos argumentos: el pin y el valor entre 0-255 a escribir. Este segundo numero representa el ciclo de trabajo de Arduino que dará salida al pin específico.

- Un valor de 255 establecerá el pin HIGH todo el tiempo (LED encendido en su máximo esplendor), un valor de 127 establecerá el pin HIGH la mitad del periodo (regulando la intensidad) y un valor de 0 establecerá el pin LOW todo el tiempo (LED apagado). Para convertir la lectura del sensor (0-1023) a valores entre 0-255, necesarios para analogWrite(), dividimos la lectura del sensor entre 4.

Comentario del codigo



- Por ultimo, se imprimen los nuevos valores asignados en su propia línea del monitor serie.