

LÁMPARA DE MEZCLA DE COLORES

USANDO UN DIODO LED TRI-COLOR Y TRES FOTO RESISTENCIAS, VA A CREAR UNA LÁMPARA QUE CAMBIA SUAVEMENTE LOS COLORES DEPENDIENDO DE LAS CONDICIONES DE LA ILUMINACIÓN AMBIENTE

Descubra: salida analógica, cambio de escala de rango

Tiempo: 45 MINUTOS

Nivel: **bajo-medio**

Proyectos en los que se basa: [1,2,3](#)

Hacer que los diodos LEDs parpadeen puede ser divertido, pero ¿se pueden usar para atenuar suavemente la luz que emiten o para producir una mezcla de colores? Se podría pensar que es solo cuestión de disminuir la tensión que se le aplica a un LED para conseguir que su luz se desvanezca.

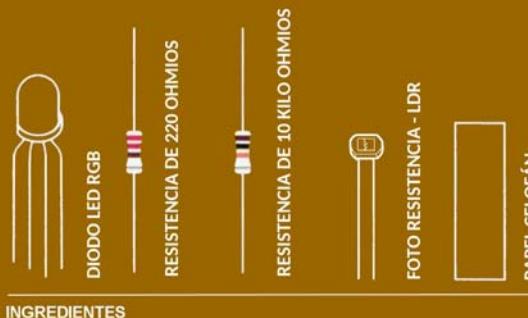
Arduino no puede variar la tensión de salida de sus pins, solo puede suministrar 5V. Por lo tanto es necesario usar una técnica llamada **Modulación por Ancho de Pulso o PWM** (en inglés) para desvanecer suavemente la luz de los LEDs. PWM consigue que un pin de salida varíe rápidamente su tensión de alto a bajo (entre 5 y 0V) durante un periodo fijo de tiempo. Este cambio de tensión se realiza tan rápido que el ojo humano no puede verlo. Es similar a la forma en las que se proyectan las películas de cine, se pasan rápidamente un número de imágenes fijas durante un segundo para crear la ilusión de movimiento.

Cuando la tensión del pin cambia rápidamente de alto (**HIGH**) a bajo (**LOW**) durante un periodo fijo de tiempo, es como si se pudiese variar el nivel de la tensión de ese pin. El porcentaje de tiempo que un pin está en estado **HIGH** con respecto al tiempo total (periodo) se conoce con el nombre de **relación cíclica**. Cuando el pin está en **HIGH** durante la mitad del periodo y en estado **LOW** durante la otra mitad, la relación cíclica es del 50%. Una relación cíclica baja hace que el diodo LED tenga una luz mucho mas tenue que con una relación cíclica más alta.



La tarjeta Arduino Uno dispone de seis pins que se pueden usar con PWM ([los pins digitales 3,5,6,9,10 y 11](#)), los cuales se pueden identificar por el símbolo ~ que aparece junto a su número sobre la tarjeta.

Para las entradas en este proyecto, se usarán foto resistencias (sensores que cambian su resistencia dependiendo de la cantidad de luz que llegue a su superficie, también se conocen con el nombre de foto células o resistencias dependientes de la luz LDR). Si conecta uno de sus pin a Arduino, se puede medir el cambio de resistencia al analizar los cambios de tensión en ese pin.



MONTANDO EL CIRCUITO

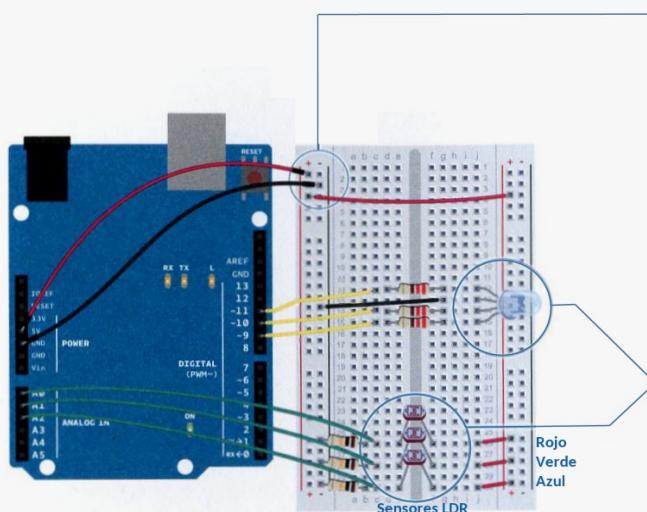


Figura 1

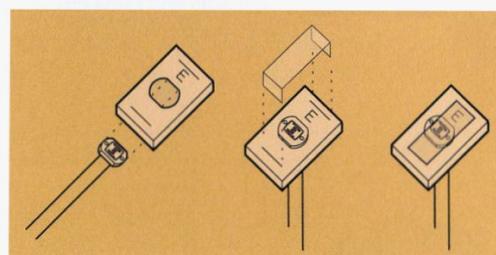


Figura 2

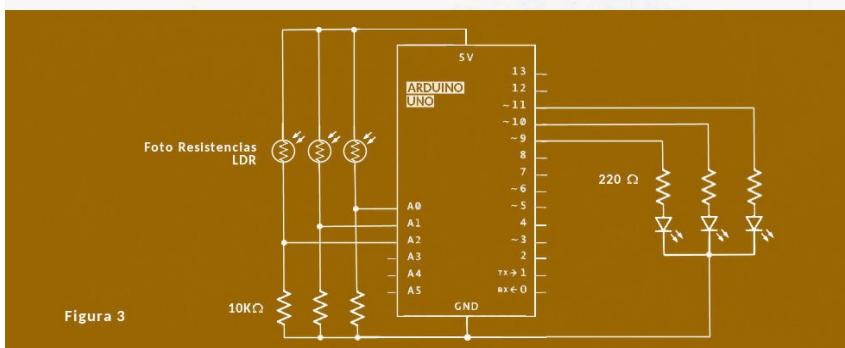


Figura 3

Tal como lo ha estado haciendo en proyectos anteriores, conectar la placa de pruebas a la alimentación y a masa.

Colocar las tres foto resistencias en el centro que divide la placa de pruebas, como se muestra en la figura 1. Conectar un terminal o pin de cada foto resistencia directamente al positivo de la alimentación. El otro extremo se conecta a una resistencia de 10 Kiloohmios la cual se conecta a masa. Esta resistencia está en serie con la foto resistencia y juntas forman un divisor de tensión. La tensión que aparece en el punto de unión entre estos dos componentes es proporcional a la relación entre sus resistencias, según la Ley de Ohm (ver el proyecto número 1 para saber más sobre la Ley de Ohm). Como el valor de la foto resistencia cambia cuando la luz incide en ella, la tensión en este punto de unión también cambia. Conectar el punto de unión entre la foto resistencia y la resistencia de 10K al pin de entrada analógico A0, A1 y A2 respectivamente usando un cable para realizar la conexión.

Coger las tres láminas de papel celofán y colocar cada una sobre cada foto resistencia. Colocar el celofán rojo sobre la foto resistencia conectada a la entrada A0, el verde sobre la que se conecta a la entrada A1, y el azul sobre la conectada a la entrada A2. Cada uno de estos papeles de colores actúan como filtros de luz y solo dejan pasar una determinada longitud de onda (color) hacia el sensor sobre el que están encima. El papel de color rojo solo deja pasar la luz roja, el papel verde solo deja pasar la luz verde, y el papel azul solo deja pasar la luz azul. Esto permite detectar los niveles relativos de color de la luz que reciben los sensores.

El LED con cuatro pins es un diodo LED RGB de cátodo común. Este diodo LED tiene los elementos rojo, verde y azul separados en su interior, y un masa común (el cátodo). Al aparecer una diferencia de tensión entre el cátodo y las tensiones de salida de los pins de Arduino en formato PWM (los cuales se conectan a los ánodos del LED a través de unas resistencias de 220 ohmios), será posible hacer que el LED varíe la iluminación de sus tres colores. Observar que el pin más largo del LED, insertado en la placa de pruebas, se conecta a masa (ver Figura 1). Conectar los otros tres pins del LED a los pins digitales 9, 10 y 11 de Arduino y en serie con cada uno de ellos una resistencia de 220 ohmios. Asegurarse que cada cable del LED se conecta correctamente a su pin PWM correspondiente, tal y como se puede ver en la Figura 1 de la izquierda.



EL CÓDIGO

Un par de constantes útiles

Definir como variables de tipo constante y entero los pins que se van a usar para las entradas y las salidas, y así saber que LED se enciende según el color de la luz que incide en el sensor correspondiente. Para el sensor que detecta la luz roja el LED de color rojo, el LED verde para el sensor de luz de color verde y el LED azul para el sensor de luz azul.

Variables para guardar las lecturas de los sensores así como el nivel de intensidad de cada luz

Añadir variables para almacenar los valores leídos de los sensores así como otras variables en donde almacenar estos valores después de realizar una conversión para poder variar la luminosidad de los LEDs. Se pueda usar el tipo de variable entero para todas estas variables.

Establecer como funcionan los pins digitales y configurar el puerto serie

Dentro del apartado de configuración del programa “**setup()**” establecer la comunicación serie a 9600 baudios por segundo. Como en el ejemplo anterior, se usa esta opción para ver los valores que se leen de los sensores de luz y que se mostrarán en el monitor serie. Además será posible ver un mapa de valores, los cuales se usarán para variar la luminosidad de los diodos LEDs. Aquí también se define los pins de los LEDs como salidas mediante la instrucción **pinMode()**.

Leyendo el valor de cada sensor de luz

En la parte de ejecución del bloque del programa en sí “**loop()**” se leen los valores de los sensores en las entradas analógicas A0, A1 y A2 mediante la instrucción **analogRead()** y a continuación se almacenan estos valores en las variables que comienzan con el nombre “ValorSensor...”. Se coloca una instrucción de retraso “**delay()**” entre cada instrucción de lectura analógica “**analogRead()**” para que el convertidor analógico-digital (ADC) pueda realizar su trabajo.

Informe de las lecturas del sensor de luz al ordenador

Imprimir los valores de los sensores en una sola línea. El texto “\t” equivale a presionar la tecla “tab” del teclado para realizar una tabulación al comienzo de la línea.

```
1 const int PinLedVerde = 9;  
2 const int PinLedRojo = 11;  
3 const int PinLedAzul = 10;  
  
4 const int PinEntradaLDR_Rojo = A0;  
5 const int PinEntradaLDR_Verde = A1;  
6 const int PinEntradaLDR_Azul = A2;
```

```
7 int ValorSensorRojo = 0;  
8 int ValorSensorVerde = 0;  
9 int ValorSensorAzul = 0;  
  
10 int ValorRojo = 0;  
11 int ValorVerde = 0;  
12 int ValorAzul = 0;
```

```
13 void setup() {  
14     Serial.begin(9600);  
  
15     pinMode(PinLedVerde,OUTPUT);  
16     pinMode(PinLedRojo,OUTPUT);  
17     pinMode(PinLedAzul,OUTPUT);  
18 }
```

```
19 void loop() {  
20     ValorSensorRojo = analogRead(PinEntradaLDR_Rojo);  
21     delay(5);  
22     ValorSensorVerde = analogRead(PinEntradaLDR_Verde);  
23     delay(5);  
24     ValorSensorAzul = analogRead(PinEntradaLDR_Azul);
```

```
25     Serial.print("Mapa de valores sensores \t Rojo: ");  
26     Serial.print(ValorSensorRojo);  
27     Serial.print("\t Verde: ");  
28     Serial.print(ValorSensorVerde);  
29     Serial.print("\t Azul: ");  
30     Serial.print(ValorSensorAzul);
```

Convertir las lecturas de los sensores

La instrucción que cambia el brillo del diodo LED a través de PWM se llama `analogWrite()`. Necesita dos argumentos, el pin sobre el que se escribe y un valor comprendido entre 0 y 255. Este segundo número representa la relación cíclica que aparecerá en el pin que se especifique como salida en Arduino. Un valor de 255 colocará en estado alto (**HIGH**) el pin de salida durante todo el tiempo de esta señal, haciendo que el diodo LED que está conectado a este pin brille con su máxima intensidad lumínosa. Para un valor de 127 colocará el pin a la mitad del tiempo que dura la señal (periodo), haciendo que el diodo LED brille menos que con 255. O se puede fijar el pin en estado bajo (**LOW**) durante todo el periodo, de manera que el diodo LED no alumbrará. Los valores de lectura del sensor de 0 a 1023 hay que convertirlos a valores de 0 a 255 para trabajar dentro del rango de PWM, como se menciona al comienzo de este párrafo. Por tanto los valores de lectura del sensor hay que dividirlos por 4 y los valores convertidos entre 0 y 255 se utilizarán con la instrucción `analogWrite()` para variar el brillo de los diodos LEDs.

Informe de los niveles de iluminación calculados para los diodos LEDs

Mostrar en el monitor serie un mapa de valores después de dividir por 4 las lecturas de los sensores

Establecer los niveles de iluminación para los LEDs

COMO SE UTILIZA

Una vez que Arduino ha sido programado y conectado, abrir el monitor serie. El diodo LED probablemente mostrará un color blanco apagado, dependiendo del color de la luz predominante en la habitación donde este montado este proyecto. Fijarse en los valores de los sensores que aparecen en el monitor serie, si se encuentra en un entorno con una iluminación que no varía, los números que aparecen apenas van a variar, se mantendrán en valores bastante constantes.

Apagar la luz de la habitación en donde está el circuito con Arduino montado y ver como varían ahora los valores de los sensores en el monitor serie. Usando una linterna, iluminar cada sensor (LDR) individualmente y observar como los valores cambian en el monitor serie, además de ver como cambia el color que emite el diodo LED. Cuando se cubre las foto resistencias con papel celofán de color, estos sensores solo reaccionarán a una luz de una determinada longitud de onda, es decir, que solo son sensibles a un tipo de color en consonancia con el color del papel celofán que tienen encima. De esta forma se consigue cambiar cada uno de los colores que emite el diodo LED independientemente.

```
31 ValorRojo = ValorSensorRojo/4;  
32 ValorVerde = ValorSensorVerde/4;  
33 ValorAzul = ValorSensorAzul/4;
```

```
34 Serial.print("Mapa de valores de los sensores \t Rojo: ");  
35 Serial.print(ValorRojo);  
36 Serial.print("\t Verde: ");  
37 Serial.print(ValorVerde);  
38 Serial.print("\t Azul: ");  
39 Serial.print(ValorAzul);
```

```
40 analogWrite(PinLedRojo, ValorRojo);  
41 analogWrite(PinLedVerde, ValorVerde);  
42 analogWrite(PinLedAzul, ValorAzul);  
43 }
```

Puede fijarse que la salida de la foto resistencia no trabaja dentro de todo el rango de 0 a 1023. Esto esta bien en este proyecto, pero para obtener una explicación más detallada sobre como calibrar el rango de lectura del sensor ver el proyecto número 6.



Probablemente se habrá fijado que la variación del brillo del diodo LED no es lineal. Cuando el LED está a la mitad del valor de PWM a 127 (mitad del brillo), aparece mucho menos brillante. Esto es debido a que nuestros ojos no perciben las variaciones del brillo linealmente. El brillo del LED no solo depende del valor de la instrucción `analogWrite()`, sino también de la distancia de la luz desde el difusor, la distancia de los ojos a la luz, y el brillo del LED en relación con otra luz que pueda haber en la habitación.

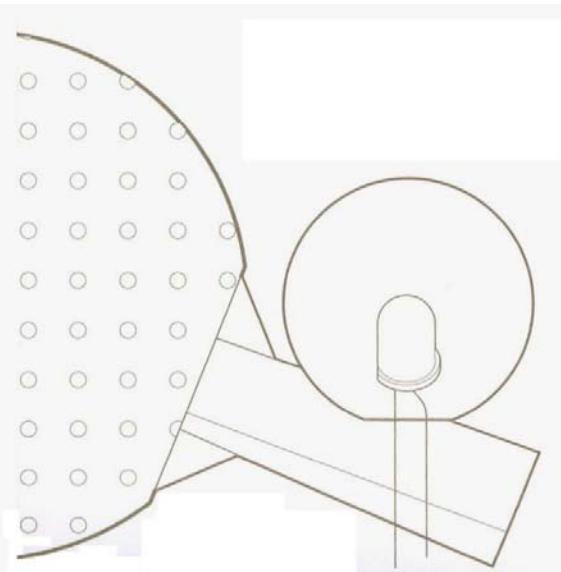


¿Como podría usar este proyecto para saber si en el exterior hace un buen día mientras usted está trabajando en el interior? ¿Que otros tipos de sensores, en lugar de las resistencias LDR, se pueden usar para controlar los colores del diodo LED?



Aunque el diodo LED no deja de ser un lámpara, la luz que produce nada tiene que ver con la luz de una lámpara. Existen varias formas de difuminar la luz del diodo LED para que se parezca a la luz que produce una bombilla de incandescencia. Realizar un agujero en una pelota de ping pong (figura 4) para colocarla encima del diodo LED y de esta forma difuminar la luz que produce. Otra forma de hacerlo es colocando pegamento transparente sobre el LED o lijando su superficie. No importa que procedimiento utilice, se va a perder un poco de brillo cuando se difumine la luz, pero al final probablemente obtendrá una luz mucho más agradable.

Ya no solo se limita a encender o apagar luces, sino que ahora tiene el control sobre como variar la iluminación de una luz. `analogWrite()` es la función que permite a los componentes conectados a los pines PWM 3, 5, 6, 9, 10 o 11, variar la relación cíclica de la señal aplicados a ellos.



Cortar una bola de ping pong para colocarla encima del diodo led

Figura 4