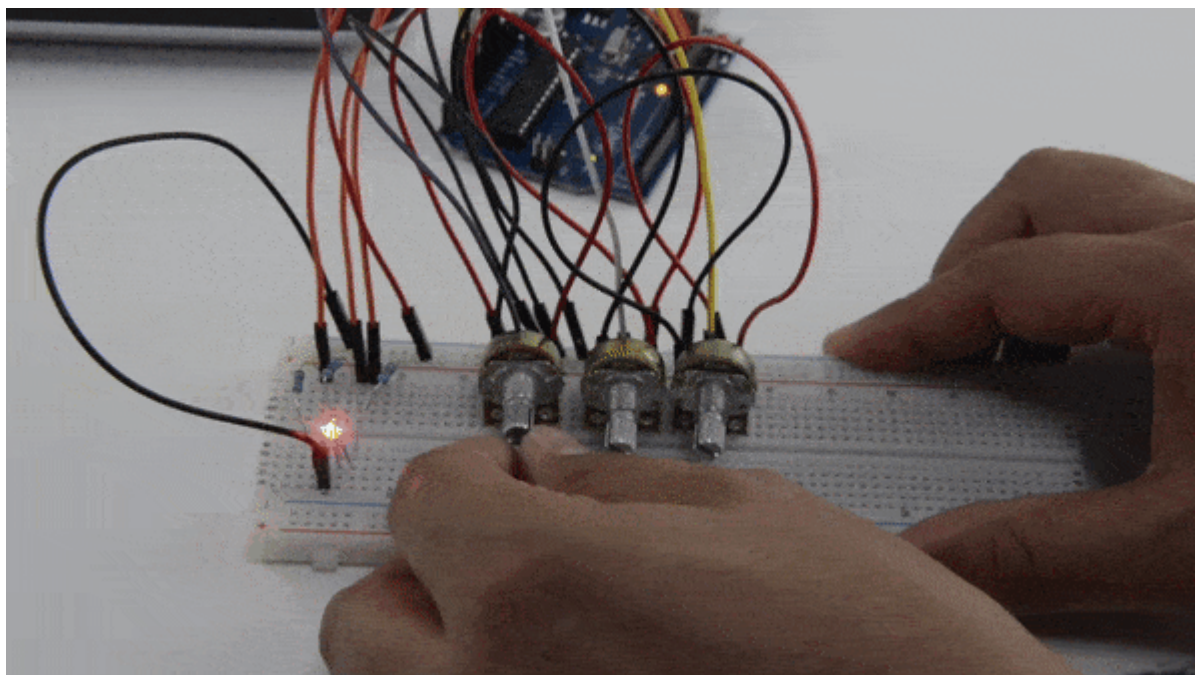


PROYECTO N° 2: Controlar un LED RGB



Aprende a programar un circuito para controlar los colores y brillo que produce un LED RGB. Utilizarás 3 potenciómetros, uno para cada color del LED. Según la combinación de brillo de cada uno, puedes formar cualquier color.



NIVEL DE DIFICULTAD: PRINCIPIANTE.

DURACIÓN DE LA ACTIVIDAD: 30 min.

MATERIALES:

- 3 Resistencias de 220
- 3 Potenciómetros 10k
- 1 Placa Build&Code UNO
- 1 ProtoBoard
- 1 LED RGB

¿Qué es un LED RGB?

Un LED RGB es un LED que puede emitir luz de cualquier color a partir de la combinación de sus tres colores básicos: rojo (Red), verde (Green), azul (Blue). En otras palabras, es la unión de tres LEDs de los colores básicos (rojo, verde y azul) en un encapsulado común. En función de la tensión que pongamos en cada pin podemos conseguir la mezcla de color que deseemos con relativa sencillez.

Un LED RGB tiene 4 patas, 1 para controlar cada color y la que queda es por donde entra o sale la electricidad, dependiendo de si es de ánodo (la pata más

larga es por donde entra la electricidad) o cátodo común (la pata más larga es por donde sale la electricidad).

Delante de cada una de las patas de color del LED RGB siempre se tendrá que poner una resistencia con un valor suficiente para no romper el LED. En este caso, pondremos una resistencia de 220 ohmios.

¿Qué es un Potenciómetro?

Un potenciómetro es un dispositivo que permite variar su resistencia de forma manual, entre un valor mínimo R_{min} (normalmente 0 ohmios) y un valor máximo R_{max} . Los valores habituales de R_{max} son 5k, 10k o 20k ohmios.

Un potenciómetro es una resistencia variable que va desde un valor mínimo a otro valor máximo. Al hacer girar el potenciómetro entre un mínimo y un máximo, hacemos variar el valor de la resistencia interna de manera proporcional al giro.

Normalmente un potenciómetro tiene 3 pines. Uno de los 2 pines de los extremos está unido al valor de la resistencia mínima R_{min} y otro al valor de la resistencia máxima R_{max} . El pin restante corresponde al valor variable de la resistencia, y este valor es el resultado de la posición del potenciómetro, que estará más cerca o más lejos de los valores R_{min} o R_{max} .

Los pines de los extremos se conectan a los puertos 5V y GND de la placa Build&Code UNO, y el tercer pin (cuyo valor depende del movimiento del potenciómetro) se conecta a uno de los pins analógicos de la placa.

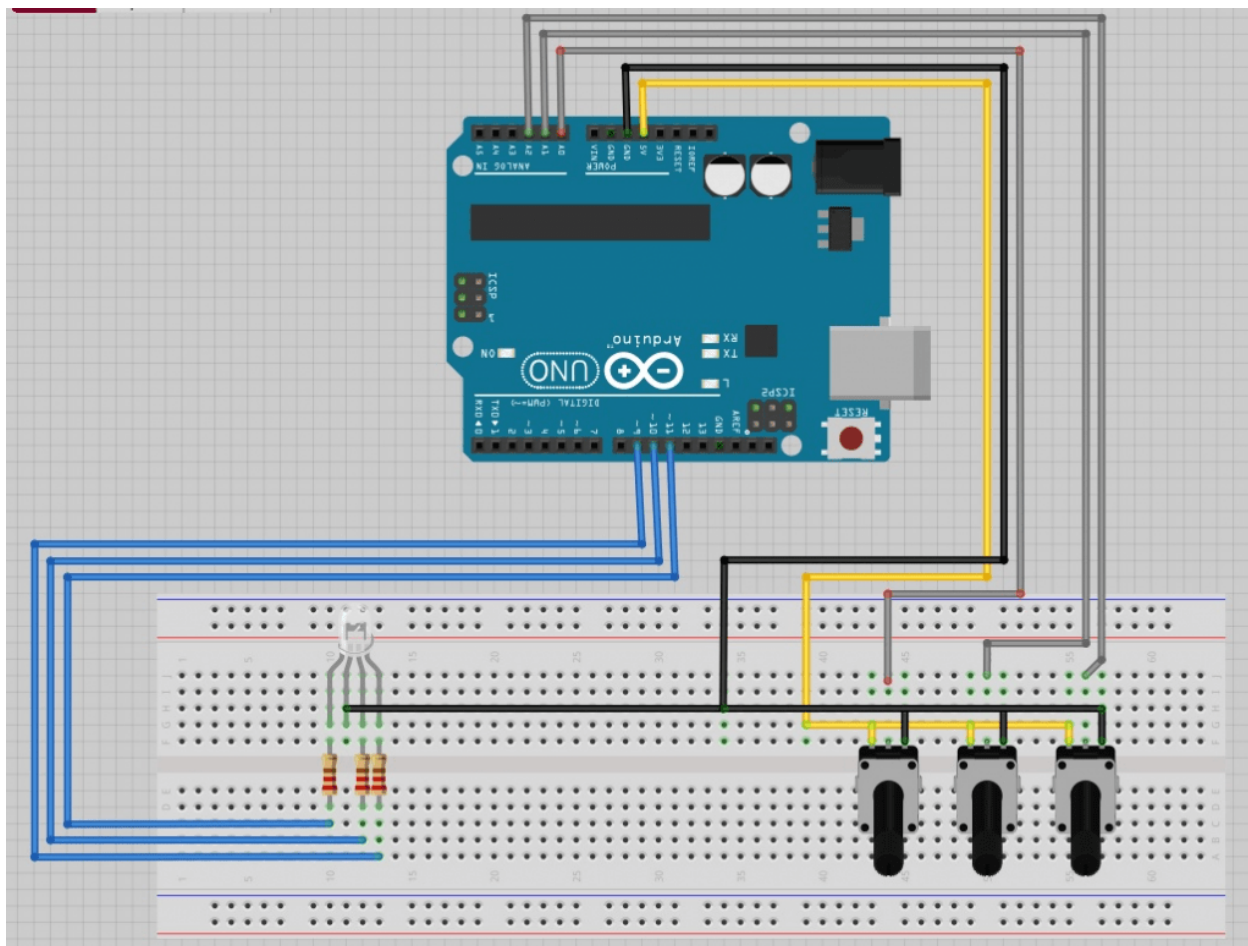
Como todo sensor analógico, el potenciómetro enviará información a la entrada analógica. La información irá desde 0 a 1023. Cuando el potenciómetro esté en su valor mínimo enviará 0 y cuando esté en su valor máximo, enviará 1023. Si se encuentra en su valor medio, enviará 516.

CONEXIONES:

1. La placa protoboard recibe la electricidad del pin de 5V de la placa Build&Code UNO y vuelve de la protoboard a la Build&Code UNO a través del pin GND (tierra). Conecta todos los GND del circuito entre sí para que todos tengan el mismo valor de GND. En la imagen verás que el cable color **negro** es el GND, donde todos los *Grounds* de cada

componente están conectados entre sí y al GND de la placa. El color **amarillo** representa los 5V, que alimentan a cada uno de los potenciómetros.

2. Conecta el pin central de tus sensores, los potenciómetros, a un pin analógico diferente cada uno, ya que el potenciómetro es un sensor analógico. En la imagen, los 3 cables color **gris** van del pin central de cada uno de los potenciómetros a 3 entradas analógicas.
3. Conecta cada uno de los pines del LED RGB a un pin digital PWM, que será el encargado de hacer brillar cada color más o menos en función de la lectura recibida del sensor. La imagen muestra cómo al LED RGB llegan 3 cables color **azul** de 3 salidas digitales, que pueden trabajar como PWM: 9, 10 y 11.



CÓDIGO DE PROGRAMACIÓN

Puedes realizar este ejercicio utilizando el programa Arduino o un *software* de programación por bloques compatible. A continuación encontrarás el código de programación necesario.

Código Arduino

Realizarás un programa que estará leyendo la información de los potenciómetros de manera continua. En función de la lectura que mida cada sensor, cada color básico del LED RGB brillará más o menos.

Cada potenciómetro hará lecturas de 0 a 1023, en función de cuánto lo hayas girado.

El pin digital PWM estará trabajando como un pin de salida, pero en vez de dar un valor alto o bajo de manera constante, éste puede hacer cambios muy rápidos de valores para jugar con valores medios, lo cual permite emitir distintos niveles de brillo.

Es decir, que si en 1 segundo el LED se enciende y apaga 100 veces, y está el mismo tiempo encendido que apagado (50 veces encendido y 50 veces apagado), nuestro ojo verá el LED con un brillo equivalente a una potencia del 50% del total. Si el LED estuviese mucho más tiempo encendido que apagado, nuestro ojo detectaría un brillo muy leve.

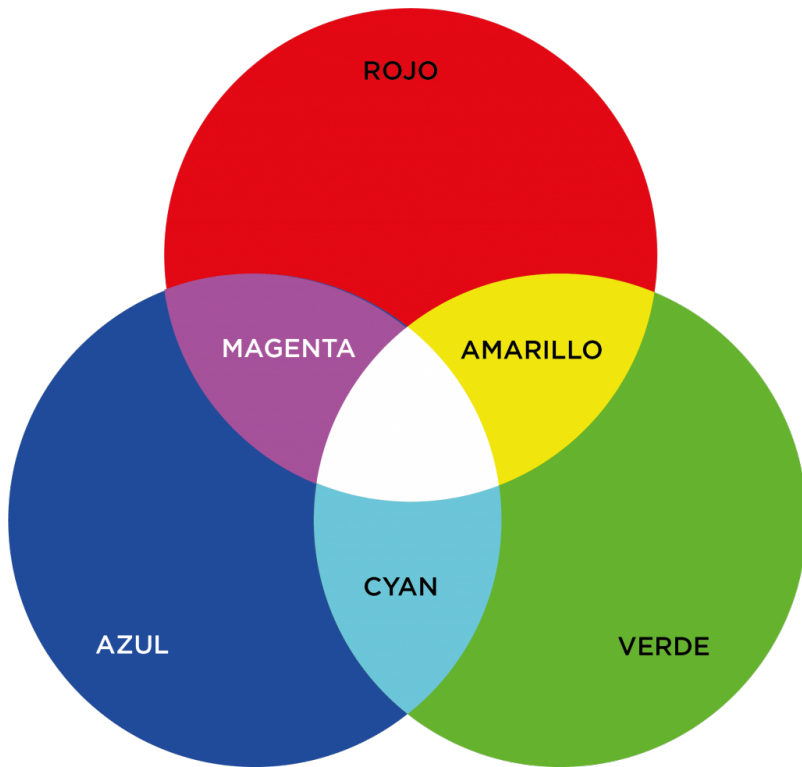
El PWM tiene una resolución de 256 estados, es decir que va desde 0 a 255. Si escribes el valor 15, apenas se encenderá, ya que estará más tiempo apagado que encendido. Si escribes el valor 127, lo verás brillar con el 50% de la potencia máxima, ya que estará el 50% del tiempo encendido y el otro 50% apagado. Si le asignas un valor de 250, prácticamente lo verás como si brillara al 100%.

Para relacionar la entrada analógica del potenciómetro y la del pin digital PWM, hay que dividir los 2 valores máximos:

$$\frac{1024}{256} = 4$$

Esto quiere decir que si divides los valores del sensor entre 4 estarás estableciendo una relación lineal entre el giro del potenciómetro y el brillo de cada color del LED RGB.

Este gráfico te será de gran ayuda para construir los colores a partir de los colores primarios RGB:



1. Descarga el software Arduino y realiza el proceso de instalación.
2. Abre el programa Arduino y, una vez en él, copia el siguiente programa:

```
int entrada0 = A0; //Definimos las 3 entradas analógicas
A0, A1, A2
int entrada1 = A1;
int entrada2 = A2;

int LEDR = 9; //Definimos las 3 salidas digitales PWM
int LEDG = 10;
int LEDB = 11;

int valor0=0; //creamos las 3 variables valor de tipo
entero para almacenar los valores guardadas de las lecturas
analógicas
int valor1=0;
int valor2=0;

void setup ()
{
  Serial.begin(9600); //iniciamos comunicación para mostrar
por pantalla la información recibida de la placa

  pinMode(LEDR,OUTPUT); // los pins digitales serán de
salida
```

```

pinMode(LEDG,OUTPUT);
pinMode(LEDB,OUTPUT);
}

void loop()
{
    int valor0 = analogRead(entrada0); // lee el valor del
potenciómetro 0
    int valor1 = analogRead(entrada1); // lee el valor del
potenciómetro 1
    int valor2 = analogRead(entrada2); // lee el valor del
potenciómetro 2

    analogWrite(LEDG, valor0/4); // Como las entradas
analógicas tienen una resolución máxima de 1024 estados y
el PWM tiene una resolución de 256, tendremos que dividir
el valor de la entrada
    //analógica entre 4 para hacer proporcional la lectura de
la entrada analógica con la intensidad de salida de la
salida digital PWM
    analogWrite(LEDG, valor1/4);
    analogWrite(LEDB, valor2/4);
}

```

Código para *software* de programación por bloques compatible

1. Descarga el *software* y realiza el proceso de instalación.
2. Abre el programa y, una vez en él, copia el siguiente programa:



Algunos de los valores que aparecen en la programación surgen de establecer relaciones lineales entre los valores que pueden tomar los componentes implicados.

En este caso, el valor se divide por 4 para relacionar las entradas analógicas del potenciómetro y las entradas digitales PWM del LED RGB: $1024/256 = 4$

RESULTADO DEL EJERCICIO

El LED RGB conectado a la placa Build&Code UNO mediante el circuito de la protoboard, cambiará de color de acuerdo al giro de los 3 potenciómetros.