

PRACTICA 3.

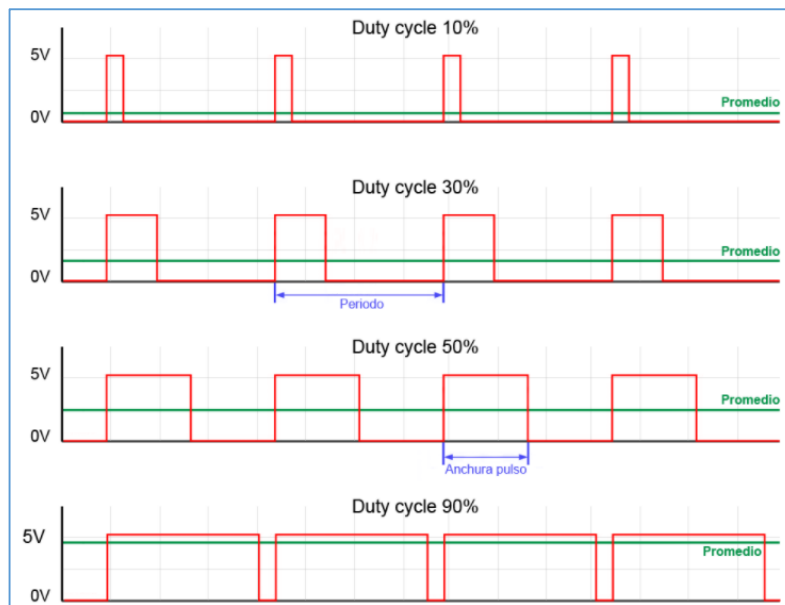
Materiales:

- 1 NodeMCU con cable USB
- 1 LED RGB con ánodo o cátodo común.
- 1 Conectores

SALIDAS ANALÓGICAS O PWM.

Lo primero que tenemos que entender es que la mayoría de automatismos (y NodeMCU no es una excepción) no son capaces de proporcionar una auténtica salida analógica. Ni siquiera pueden suministrar una salida analógica discretizada (es decir, a saltos) de tensión. Lo único que pueden proporcionar es una salida digital de $-V_{cc}$ o V_{cc} . (por ejemplo, 0V y 3.3V).

Para salvar esta limitación y simular una salida analógica la mayoría de los automatismos emplean un “truco”, que consiste en activar una salida digital durante un tiempo y mantenerla apagada durante el resto. El promedio de la tensión de salida, a lo largo del tiempo, será igual al valor analógico deseado.



La forma de hacer una señal analógica es a través de la modulación de ancho de pulso (PWM). En esta modulación se mantiene constante la frecuencia (es decir, el tiempo entre disparo de pulsos), mientras que se hace variar la anchura del pulso.

Podemos variar la intensidad luminosa en un LED mediante un PWM. El LED realmente se enciende y apaga varias veces por segundo, pero este parpadeo es tan rápido que el ojo no lo aprecia. El efecto global percibido es que el LED brilla con menor intensidad.

Otro ejemplo, al variar la velocidad de un motor DC con un PWM, en la mayoría de los casos la inercia del motor se encargará de que el efecto del PWM sea despreciable. No obstante, en función de la frecuencia podemos notar vibraciones o ruidos, en cuyo caso deberemos variar la frecuencia del PWM.

Todos los pines de E / S de uso general (GPIO) del NodeMCU permiten salidas de PWM (Pulse Width Modulated) por tal razón podemos usar todos sus pines GPIO como salidas de PWM, la resolución del PWM es de 10 bits a diferencia de un arduino que es de 8bits. Por ejemplo:

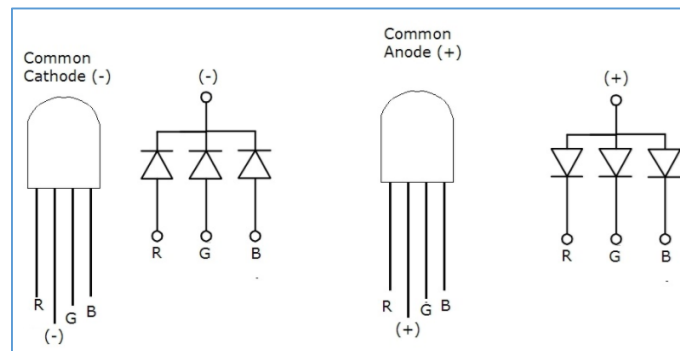
`analogWrite(led, 255);`

En este caso a través de un pin que anteriormente se llamó led se tienen una salida de voltaje proporcional a 255 teniendo en cuenta que el valor mínimo es 0 y el máximo 1023.

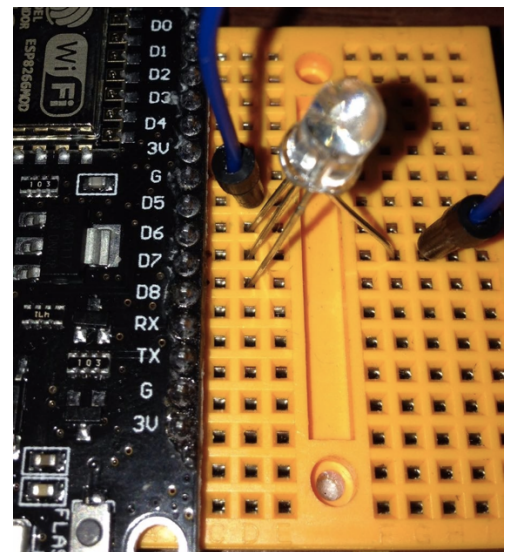
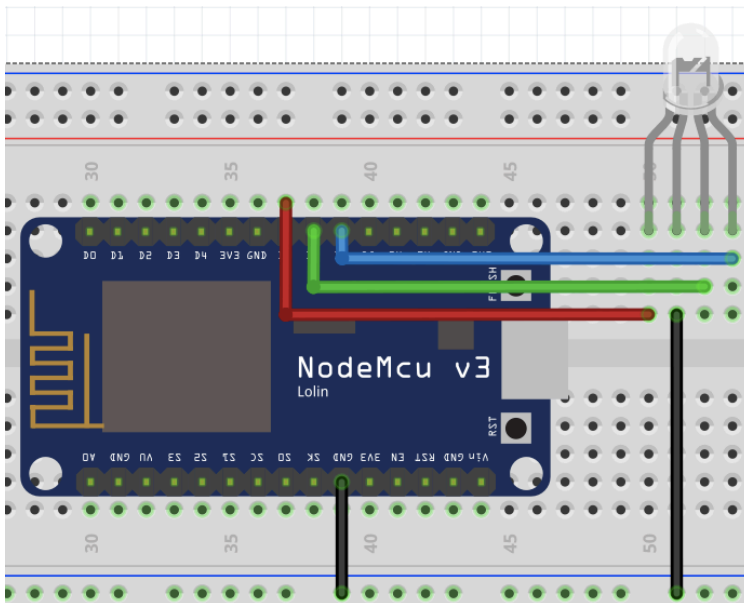
LED RGB:

El LED RGB es un componente electrónico que puede reproducir casi cualquier color de una manera perfecta, pudiéndose utilizar para reproducir imágenes y vídeos, o para iluminar una sala con un color determinado.

El acrónimo RGB proviene debido a que utiliza el sistema de colores RGB contruidos a base de Rojo, Verde (Green) y Azul (Blue) los tres colores primarios.



Podemos controlar la intensidad de cada uno de los colores (rojo, verde y azul) por separado, añadiendo las cantidades de color que necesitemos de cada uno de ellos para conseguir el color que estemos buscando.



Ejemplo 1: Determine la función de cada uno de los pines del diodo LED RGB y si es de ánodo o cátodo común. Para esto puede utilizar un multímetro o con los pines de 3V y GND del NodeMCU.

Implementemos un circuito con un LED RGB para que durante un segundo encienda con el color rojo, durante otro segundo en azul y el último segundo verde. También utiliza el monitor serial para indicar el color utilizado. Tenga en

cuenta que debe trabajar con 3 puertos como el GPIO 12, 13 y 14. En la siguiente gráfica se puede ver un ejemplo con estos puertos para 2 colores.

```
void setup() {
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(14, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(12, 1023);
  digitalWrite(13, 0);
  digitalWrite(14, 0);
  delay(1000);
  Serial.println("Este color es Rojo");
  digitalWrite(12, 0);
  digitalWrite(13, 1023);
  digitalWrite(14, 0);
  delay(1000);
  Serial.println("Este color es Azul");
  digitalWrite(12, 0);
  digitalWrite(13, 0);
  digitalWrite(14, 1023);
  delay(1000);
  Serial.println("Este color es Verde");
}
```

Revise que encienda el LED con los tres colores y que coincida con los valores del monitor serial. Corrija si es necesario.

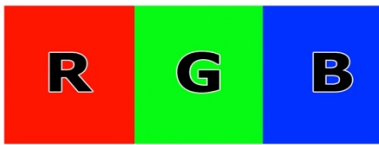
EL BUCLE FOR

El bucle for es una estructura que se utiliza cuando queremos que una serie de acciones se repita un número determinado de veces, para ello se necesita de una variable índice, una condición y un incrementador. Por ejemplo para escribir los números de 0 a 100 sería:

```
for(int x = 0; x < 100; x++) {
  println(x); // prints 0 to 99
}
```

En este caso, con el código anterior, se elige la variable **x** y se define que será inicializada en **0**, luego se propone un límite para el contador, en este caso interesa que la **x** llegue hasta **99** (por lo tanto 100 repeticiones) y finalmente se indica como variará el valor del contador por cada ciclo, en este caso el contador será **incremental**.

Ejemplo 2: Repitamos el ejercicio 1 pero esta vez utilizaremos el bucle for para establecer el modo de los pines del NodeMCU. También para este caso vamos a utilizar otros como colores amarillo, naranja y morado.

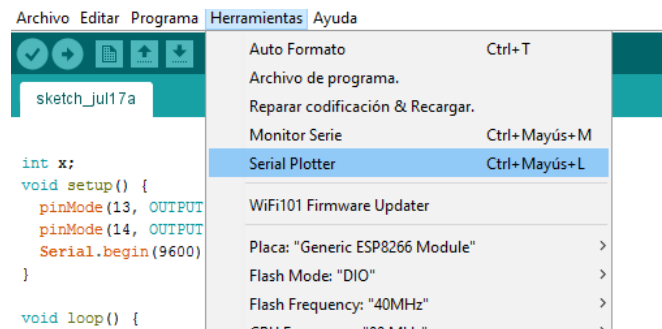


```
void setup() {  
    for (int i =12 ; i<15 ; i++)  
        pinMode(i, OUTPUT);  
}  
  
void loop() {  
}
```

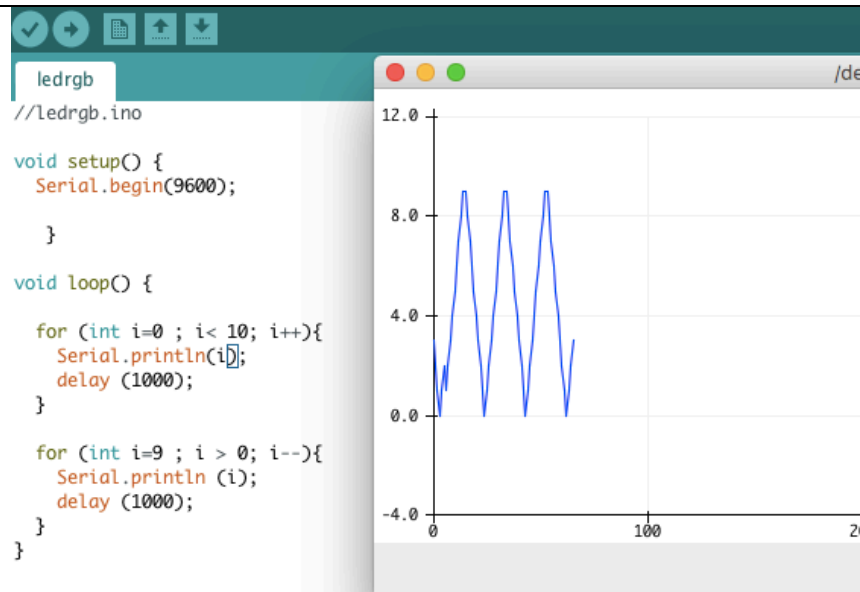
Teniendo en cuenta las figuras anteriores realizar un código que en el primer segundo encienda el LED amarillo, y cambie de color utilizando valores entre 0 y 1023 en los puertos de salida. Para establecer el modo de los pines de salida utiliza el código de la figura el cual utiliza el bucle for.

GRAFICAS POR PUERTO SERIAL.

Anteriormente si deseamos ver el valor de una variable activamos el Monitor Serie. Ahora, si desea mirarlo de forma gráfica cierras el Monitor serie y abres Serial Plotter como se muestra en la figura.

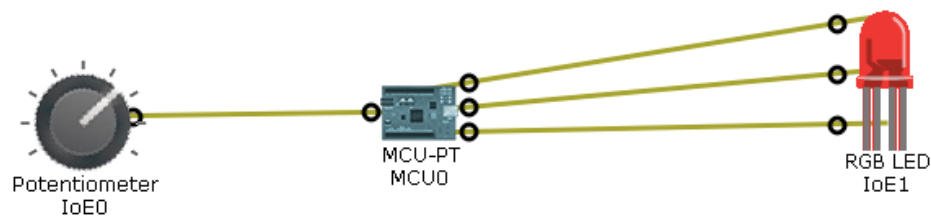


Ejemplo 3: Vamos a graficar una señal triangular con un valor entre 0 y 9 a través del plotter serial. Copie y analice el siguiente código.



TRABAJO EN CASA:

1. Realice utilizando el packet tracer una simulación con un led RGB que encienda de amarillo, azul y rojo conforme se aumente el valor de un potenciómetro.



2. Programar el NodeMCU a través de TUNIoT en modo estación WiFi, es decir que sea para que sea un host WiFi. El dispositivo estará conectado a una red WiFi con SSID y password y deberá mostrar por el monitor serial la IP utilizada. Verificar realizando ping desde cualquier otro dispositivo de la red al NodeMCU.