

# PRÁCTICAS DE ARDUINO

## Práctica nº 7: Control de intensidad de iluminación

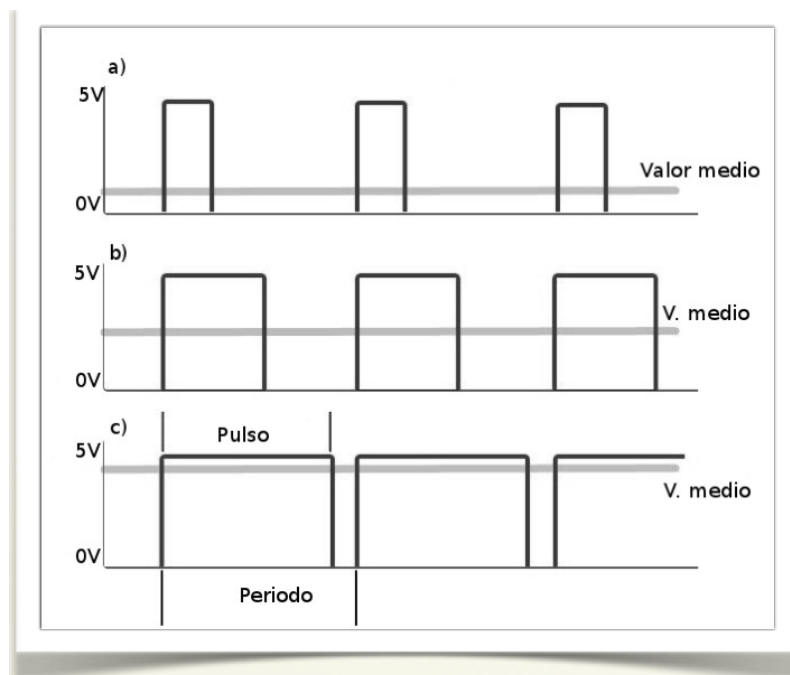
En esta práctica vamos a utilizar una señal PWM para realizar el control de la intensidad de iluminación de un diodo LED utilizando un potenciómetro.

### ¿QUÉ ES UNA SEÑAL PWM?

La señal PWM (Pulse Width Modulation, Modulación de Ancho de Pulso) es una señal que utiliza el microcontrolador para generar una señal continua sobre el proceso a controlar. Por ejemplo, la variación de la intensidad luminosa de un led, el control de velocidad de un motor de corriente continua,...

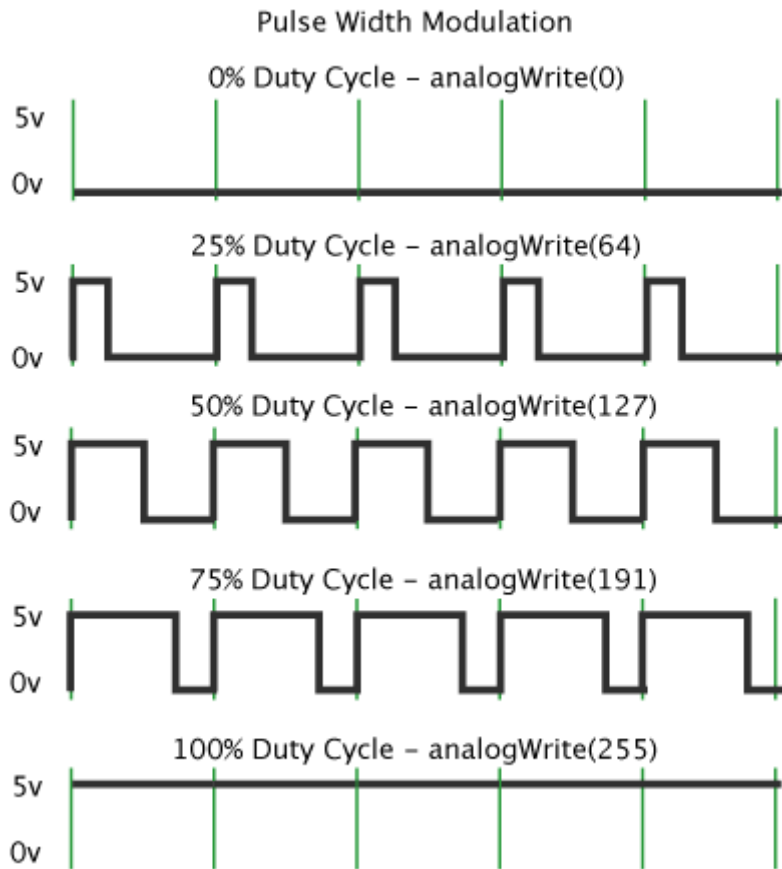
Para que un dispositivo digital, microcontrolador de la placa Arduino, genere una señal continua lo que hace es emitir una señal cuadrada con pulsos de frecuencia constante y tensión de 5V. A continuación, variando la duración activa del pulso (ciclo de trabajo) se obtiene a la salida una señal continua variable desde 0V a 5V.

Veamos gráficamente la señal PWM:



Los pines digitales de la placa Arduino que se utilizan como salida de señal PWM generan una señal cuadrada de frecuencia constante (490Hz), sobre esta señal periódica por programación podemos variar la duración del pulso como vemos en estos 3 casos:

- La duración del pulso es pequeña y la salida va a tener un valor medio de tensión bajo, próximo a 0V.
- La duración del pulso es casi la mitad del período de la señal, por tanto, la salida va a tener un valor medio de tensión próximo a 2,5V.
- La duración del pulso se aproxima al tiempo del período y el valor medio de tensión de salida se aproxima a 5V.

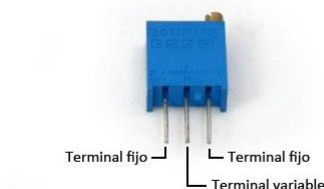


## ¿QUÉ ES UN POTENCIÓMETRO?

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  $R_{max}$ . Valores habituales de  $R_{max}$  son 5k, 10k o 20k ohmios.

Internamente un potenciómetro está constituido por un contacto móvil que se desplaza a lo largo de una pista resistiva. De esta forma, al mover el potenciómetro movemos el contacto a lo largo de la pista, y variando la longitud del tramo de pista con el que estamos en contacto, y por tanto variando su resistencia.

Normalmente un potenciómetro tiene tres terminales. Los dos extremos están unidos a ambos lados de la pista, por lo que siempre registrarán la resistencia máxima  $R_{max}$ . El terminal restante corresponde con el contacto móvil. Este terminal varía su resistencia respecto a los otros dos terminales a medida que accionamos el potenciómetro, siendo la suma de la resistencia a los otros terminales igual a  $R_{max}$ .



Respecto a la geometría, **podemos encontrar potenciómetros de tipo lineal o rotativo.**



Lineal



Rotativo

Por último, en cuanto a la relación entre posición y resistencia, **encontramos potenciómetros de tipo lineal, parabólico o exponencial**. Los lineales presentan una proporcionalidad entre resistencia y desplazamiento, lo cual significa un comportamiento más intuitivo. Mientras, los exponenciales permiten mayor precisión en valores de resistencia bajos, por lo que resultan adecuados cuando hace falta un ajuste fino en un amplio rango.

## MATERIALES

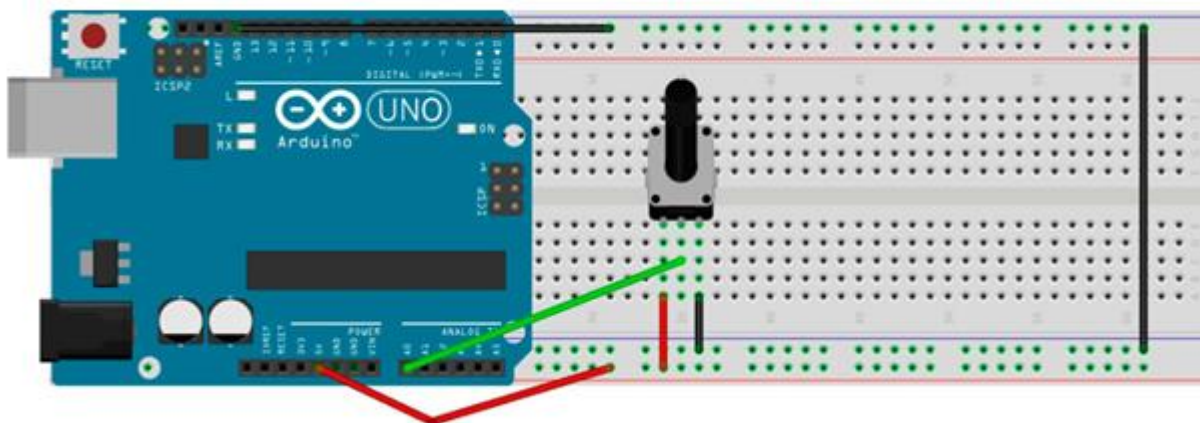
- Arduino UNO.
- Cable USB tipo A-B.
- 1 placa Protoboard.
- Cables de conexión.
- Un potenciómetro de 10K
- Un LED
- 1 Resistencia de  $220\Omega$ .

## UTILIZANDO EL POTENCIÓMETRO CON ARDUINO

En la placa Arduino UNO tenemos 6 pines analógicos, desde A0 hasta A5 y su uso común es la lectura de datos de dispositivos analógicos como es el caso del potenciómetro. Tienen una resolución de 10 bits lo que implica que tenemos **1024 valores diferentes**, es decir, podemos leer un rango de tensiones desde 0V hasta 5V detectando cambios de voltaje de 0.004V ( $5/1024$ ). Por lo que los valores que obtendremos irán desde 0 hasta 1023.

Y como la mejor manera de entender algo son los ejemplos, empezamos con uno que mediante el monitor serie podremos ir viendo que valores vamos obteniendo en un pin analógico según vayamos modificando la posición del potenciómetro.

En el siguiente esquema te muestro el conexionado.



## PROGRAMACIÓN.

Y el código que debes cargar en la placa es el siguiente.

```
Prueba_potenciometro

//Variable donde almacenaremos el valor del potenciometro
long valor;

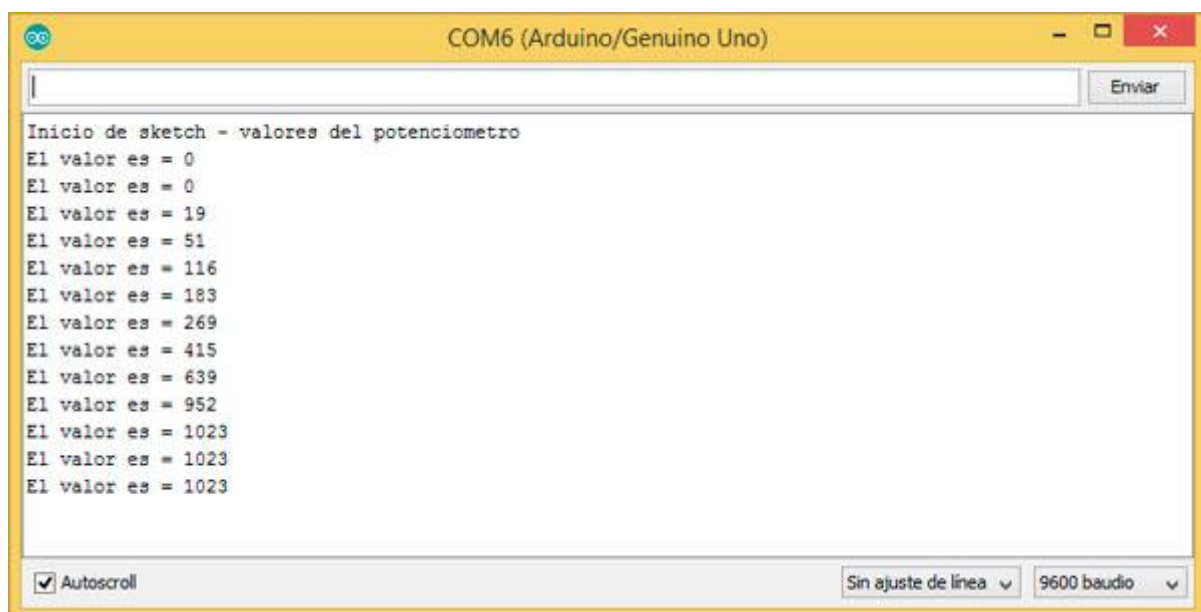
void setup() {
  //Inicializamos la comunicación serial
  Serial.begin(9600);

  //Escribimos por el monitor serie mensaje de inicio
  Serial.println("Inicio de sketch - valores del potenciometro");
}

void loop() {
  // leemos del pin A0 valor
  valor = analogRead(A0);

  //Imprimimos por el monitor serie
  Serial.print("El valor es = ");
  Serial.println(valor);
  delay(1000);
}
```

Una vez hayas subido el código del potenciómetro a la placa, abre el monitor serie. La salida dependerá de la posición del potenciómetro, ves girando la rueda y verás como el valor se va modificando.



En un extremo, la resistencia será mínima y dejará pasar los 5 V (valor analógico de 1023), y en otro extremo la resistencia será máxima y dejará pasar 0 V (valor analógico de 0).

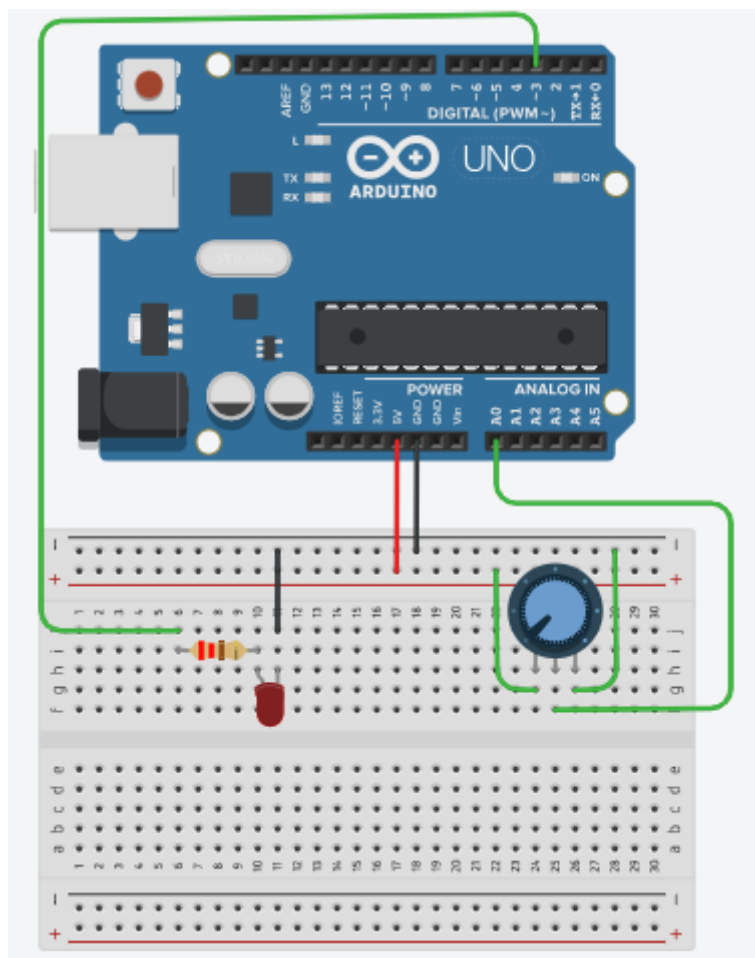
## MONTAJE.

Vamos a montar:

- Un potenciómetro de 10K conectado a la entrada analógica A0.

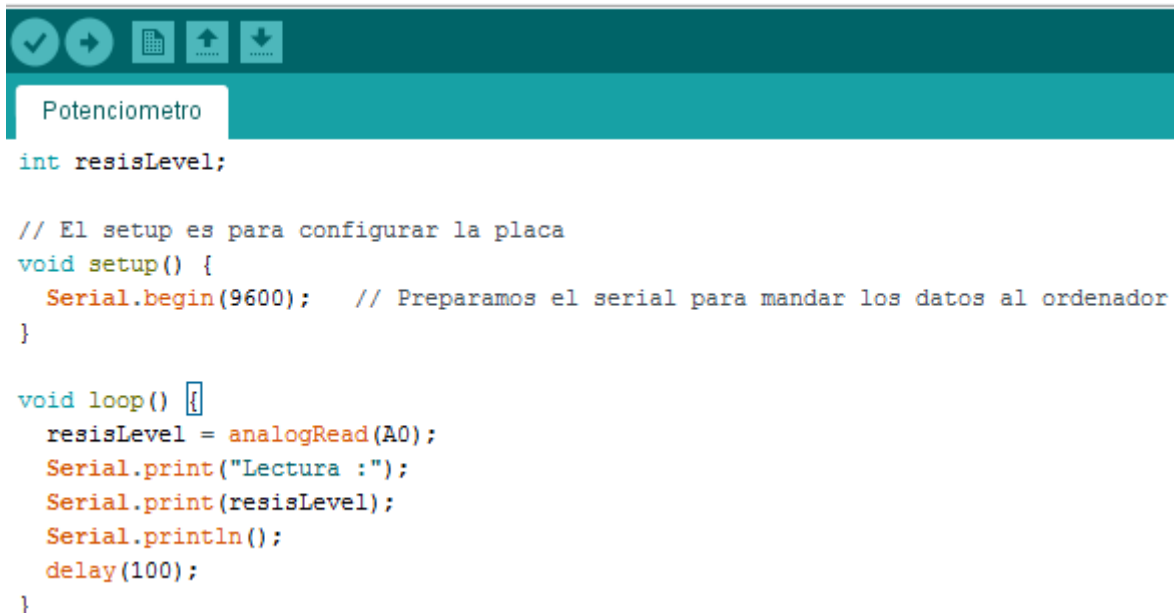


- Un LED conectado al pin digital D3, con su correspondiente resistencia de protección de 220 ohmios. La salida D3 está marcada con ~ por tanto permite usarla como salida PWM. Esto nos permite variar la luminosidad de salida del LED desde 0 hasta 255 y al mismo tiempo podemos verlo en el Monitor Serie.



## PROGRAMACIÓN.

Este programa nos permitirá variando el potenciómetro (a veces se necesita un pequeño destornillador para girar un tornillo) controlar la intensidad de luminosidad del LED y ver la variación de valores PWM en el Monitor Serie.

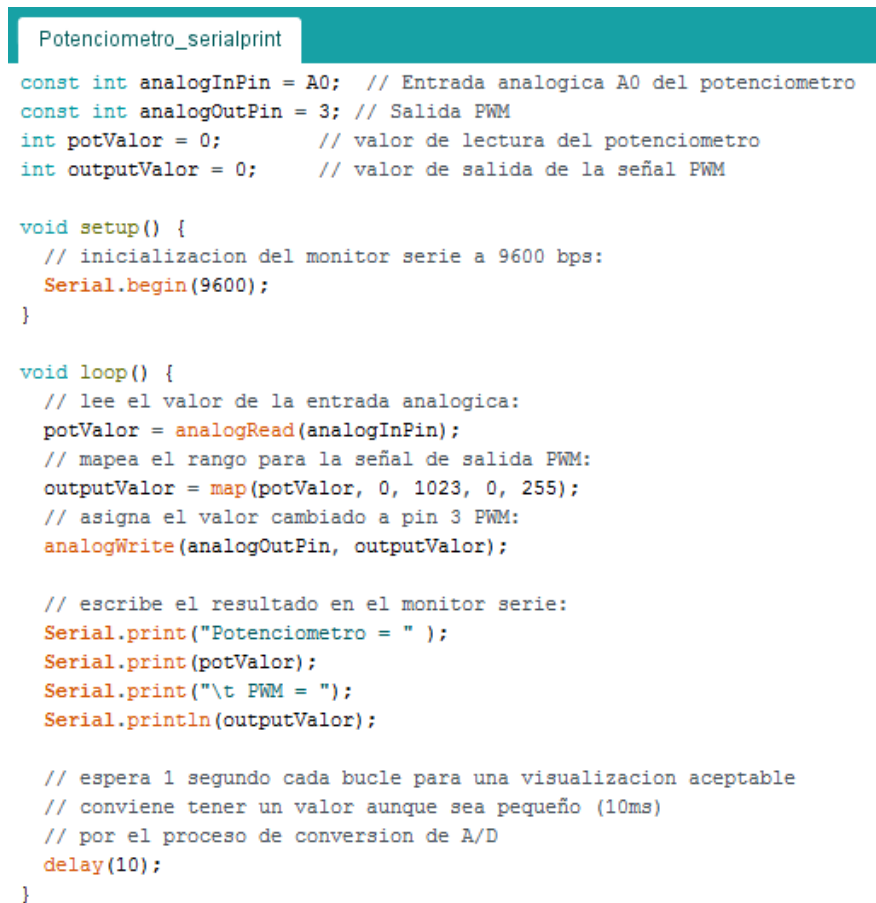


```
int resisLevel;

// El setup es para configurar la placa
void setup() {
  Serial.begin(9600);  // Preparamos el serial para mandar los datos al ordenador
}

void loop() {
  resisLevel = analogRead(A0);
  Serial.print("Lectura :");
  Serial.print(resisLevel);
  Serial.println();
  delay(100);
}
```

A continuación, se muestra una variación de este programa donde además de mostrar en el monitor serie los valores leídos y aparece el valor aplicado a la señal PWM para variar la luminosidad. Recordar que la entrada A0 donde está conectada el Potenciómetro podemos leer hasta 1024 valores diferentes (10 bits) y la salida PWM hasta 256 valores diferentes (8bits).



```
const int analogInPin = A0; // Entrada analogica A0 del potenciómetro
const int analogOutPin = 3; // Salida PWM
int potValor = 0;           // valor de lectura del potenciómetro
int outputValor = 0;        // valor de salida de la señal PWM

void setup() {
  // inicializacion del monitor serie a 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // lee el valor de la entrada analogica:
  potValor = analogRead(analogInPin);
  // mapea el rango para la señal de salida PWM:
  outputValor = map(potValor, 0, 1023, 0, 255);
  // asigna el valor cambiado a pin 3 PWM:
  analogWrite(analogOutPin, outputValor);

  // escribe el resultado en el monitor serie:
  Serial.print("Potenciometro = ");
  Serial.print(potValor);
  Serial.print("\t PWM = ");
  Serial.println(outputValor);

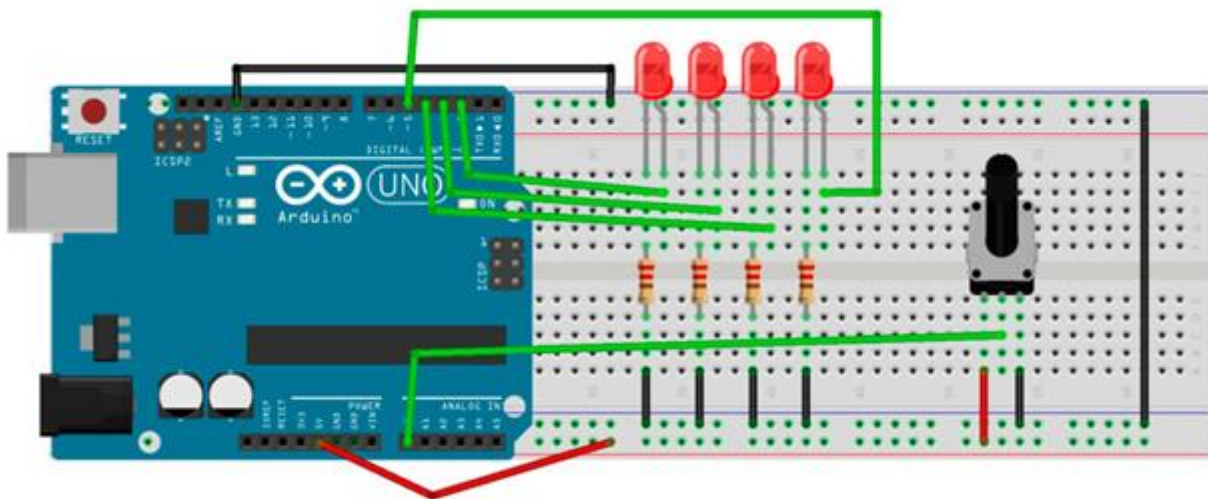
  // espera 1 segundo cada bucle para una visualizacion aceptable
  // conviene tener un valor aunque sea pequeño (10ms)
  // por el proceso de conversion de A/D
  delay(10);
}
```

Este programa lee la entrada analógica A0, mapea el resultado al rango de 0 a 255 y utiliza el resultado para poner la anchura del pulso PWM. También se escribe en el monitor serie el valor binario de A0 y el valor mapeado para la señal PWM.

De esta manera controlamos la luminosidad del led.

## MONTAJE. Encendiendo LEDs con un potenciómetro

Vamos con el segundo ejemplo, ahora vamos añadir 4 LEDs a nuestro circuito. Así según vayamos modificando la posición del potenciómetro iremos encendiendo un LED u otro. Monta el circuito del siguiente esquema



Hemos visto que podemos leer 1024 valores diferentes, desde 0 hasta 1023. Si dividimos entre 4 este rango y asignamos un rango a cada LED, podremos controlar que LED encender dependiendo del valor a la entrada del pin analógico.

Los rangos de valores que tenemos son los siguientes:

- De 0 a 255, encender LED 1
- De 256 a 511, encender LED 2
- De 512 a 767, encender LED 3
- De 768 a 1023, encender LED 4

Ahora, según el potenciómetro esté variando la entrada analógica, podremos encender un LED u otro. Esto lo vamos a programar a través de varios *if* anidados.

## El código

Vamos a utilizar la propiedad del potenciómetro de resistencia variable para encender uno u otro LED. Haremos una serie de *if* para determinar que LED encender y apagar los otros, según el valor que obtengamos en la entrada del pin analógico. El código sería el siguiente.



```
//Variable donde almacenaremos el valor del potenciometro
long valor;

//Declaramos los pins de los LEDs
int LED_1 = 2;
int LED_2 = 3;
int LED_3 = 4;
int LED_4 = 5;

void setup() {
  //Inicializamos la comunicación serial
  Serial.begin(9600);

  //Escribimos por el monitor serie mensaje de inicio
  Serial.println("Inicio de sketch - valores del potenciometro");
}

void loop() {
  // leemos del pin A0 valor
  valor = analogRead(A0);

  //Imprimimos por el monitor serie
  Serial.print("El valor es = ");
  Serial.println(valor);

  if(valor >= 0 && valor <=255)
  {
    digitalWrite(LED_1, HIGH);
    digitalWrite(LED_2, LOW);
    digitalWrite(LED_3, LOW);
    digitalWrite(LED_4, LOW);
  }

  if (valor >= 256 && valor <=511)
  {
    digitalWrite(LED_1, LOW);
    digitalWrite(LED_2, HIGH);
    digitalWrite(LED_3, LOW);
    digitalWrite(LED_4, LOW);
  }

  if (valor >= 512 && valor <=767)
  {
    digitalWrite(LED_1, LOW);
    digitalWrite(LED_2, LOW);
    digitalWrite(LED_3, HIGH);
    digitalWrite(LED_4, LOW);
  }
}
```



```
if (valor >= 768 && valor <=1023)
{
    digitalWrite(LED_1, LOW);
    digitalWrite(LED_2, LOW);
    digitalWrite(LED_3, LOW);
    digitalWrite(LED_4, HIGH);
}
}
```

## EJERCICIOS PROPUESTOS

- Realiza cada uno de los montajes y comprueba que cada programa funcionan correctamente.
- Realiza un montaje que muestre el encendido de 5 LEDs utilizando un potenciómetro y prográmalo.