

El sensor de luz LDR

Hasta ahora hemos trabajado con resistencias de valor fijo, pero existen una serie de resistencias que varían según distintos parámetros físicos a las que se les somete como presión, luz y temperatura entre otros. Existe una gran variedad que se utilizan para construir lo que llamamos **sensores**.

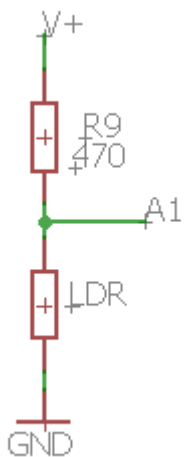
Montaje 8 Regular el led con la luz

En esta práctica vamos a diseñar un circuito que sea sensible a la luz. El objetivo **será regular la intensidad luminosa de un led con una LDR**, una resistencia sensible a la luz.

<https://www.youtube.com/embed/APpJGAgetVo?rel=0>

Montaje 8 CON EDUBASICA

En este montaje usaremos la resistencia LDR de la placa Edubásica. Como ya hemos comentado, la LDR modifica su resistencia en dependiendo de la cantidad de luz que incida sobre ella. El siguiente programa mostrará por consola ("Monitor Serial") las variaciones de luminosidad que detecte la LDR simplemente pasando la mano por encima de ella.



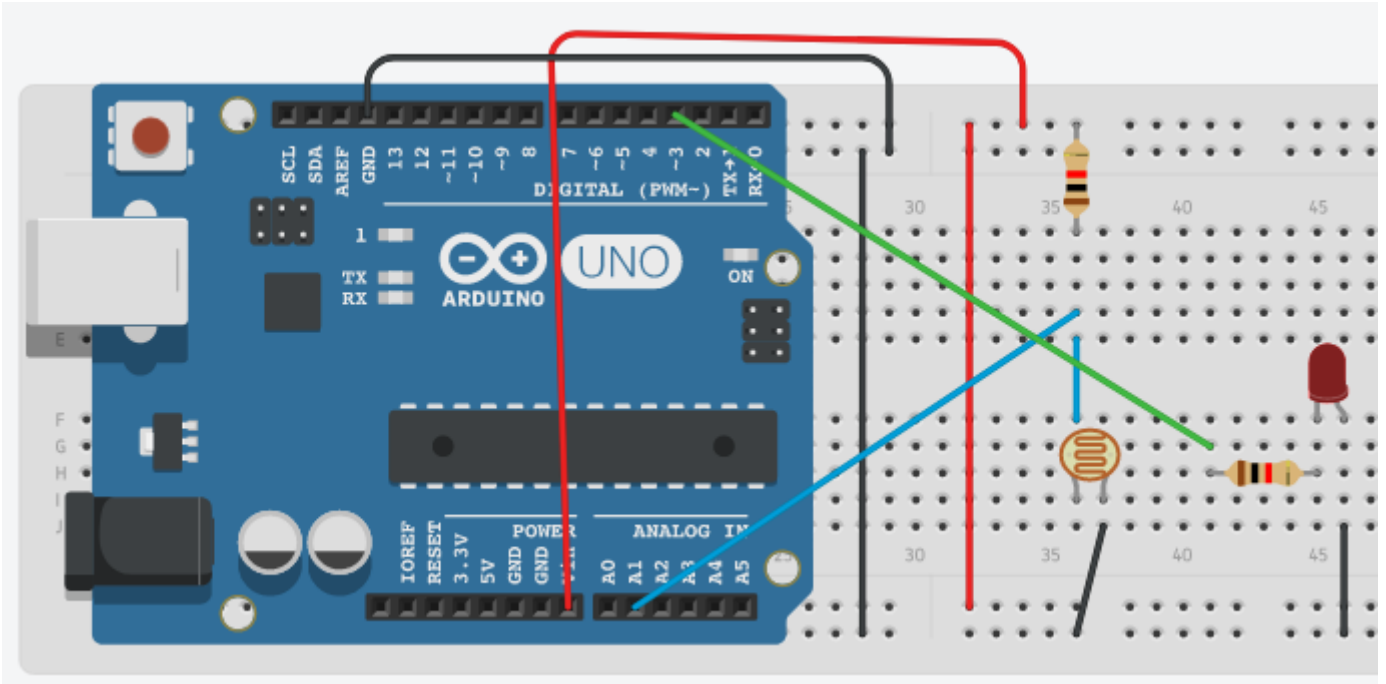
Valores entre 917 y 1024

La configuración es PULL-UP es decir, cuando el valor LDR aumenta (se acerca al OFF) la resistencia superior "tira = pull" hacia arriba = up. Los voltios suben. Si no hay luz, los voltios se acercan al máximo 5V que equivale a la lectura que el Arduino convierte en digital en 1024.

Cuando hay luz, el LDR baja su resistencia, por lo que los voltios bajan. Pero no llega a 0 Ohm, se queda pues en un valor de voltios bastante alto, 4.5V que corresponde más o menos al valor que lee Arduino en digital 917

Montaje 8 SIN EDUBASICA

Hay que conectar en formato pull-up el LDR a A1 y un led verde a D3, el valor de la resistencia si quieres utilizar los mismos valores que mostramos 917-1024 tendrá que ser 470 Ohm. Cuánto más alto sea, más bajo serán los valores.



Montaje 8 Programa

Como vemos en el esquema, el LDR está conectado a la entrada A1 del Arduino, por lo tanto la instrucción de lectura será **`analogRead(1)`** lo mapearemos correctamente a una variable llamada **`luz`** y utilizaremos el LED Verde conectado a D3 por lo que la instrucción de salida será **`analogWrite(3,luz)`**

Mapearemos con la instrucción **`map`** para convertir 917-1024 a valores de PWM 0-255

<https://create.arduino.cc/editor/javierquintana/326c66b3-c8b8-4879-980c-f81af7ab65d8/preview>

<https://create.arduino.cc/editor/javierquintana/326c66b3-c8b8-4879-980c-f81af7ab65d8/preview?embed>

Internamente la función map

¿Cómo funciona? ¿Cómo se podría quitar en estos anteriores programas?

Vimos en el ejemplo del potenciómetro que los valores a leer eran de 0-1024 y si lo queríamos convertir 0-255 era simplemente dividir por 4 y si es convertirlo de 0-5V se dividía por 204.6

Pero en este caso **EL LDR EMPIEZA DESDE 917 NO DESDE 0** como en el potenciómetro.

En una lectura previa de lo que nos devuelve **el LDR** y nos devuelve unos valores mínimo y máximo por ejemplo: **917, 1024** lo llamaremos **A1min y A1max** y queremos traducir estos valores al rango **0-255** pues es lo que podemos darle a un LED. Que lo llamaremos **luxmin y luxmax**

Podemos hallar la pendiente m y el corte con el eje y n de la recta $y = mx + n$ que convierte los valores analógicos 917-1024 a los valores 0-255. En el programa en Arduino si no queremos usar la función **map**, tenemos que escribir esa ecuación.

Aqui tienes una hoja de cálculo para calcular la pendiente y la intersección con el eje

