



## **EL TUTORIAL BÁSICO DEL KIT DE ARRANQUE PARA UNO**

**V1.0.19.7.24**

# Prefacio

## Nuestra empresa

Fundada en 2011, Inc. Elegoo es una próspera empresa de tecnología dedicada a hardware de código abierto investigación & desarrollo, producción y comercialización. Situado en Shenzhen, el Silicon Valley de China, hemos crecido a más de 150 + empleados con una fábrica de 10.763 + pies cuadrados.

Nuestras líneas de productos sonaban de DuPont cables, tableros UNO R3 para completar los kits del arrancador diseñados para clientes de cualquier nivel para aprender conocimientos de Arduino. Además, también vendemos productos de accesorios de frambuesa Pi como 2.8" TFT táctil y STM32. En el futuro, sería dedicar más energía y la inversión a los productos de impresora 3D y sucesivamente. Todos nuestros productos cumplen con estándares internacionales de calidad y son muy apreciadas en una variedad de diversos mercados en todo el mundo.

Sitio web oficial:<http://www.elegoo.com>

US Amazon storefront: <http://www.amazon.com/shops/A2WWHQ25ENKVJ1>

CA Amazon storefront: <http://www.amazon.ca/shops/A2WWHQ25ENKVJ1>

UK Amazon storefront: <http://www.amazon.co.uk/shops/AZF7WYXU5ZANW>

DE Amazon storefront: <http://www.amazon.de/shops/AZF7WYXU5ZANW>

FR Amazon storefront: <http://www.amazon.fr/shops/AZF7WYXU5ZANW>

ES Amazon storefront: <http://www.amazon.es/shops/AZF7WYXU5ZANW>

IT Amazon storefront: <http://www.amazon.it/shops/AZF7WYXU5ZANW>

## Nuestro Tutorial

Este tutorial está diseñado para principiantes. Aprenderás la información básica acerca de cómo utilizar la placa de control Arduino, sensores y componentes. Si quieres estudiar Arduino en mayor profundidad, recomendamos que lea el libro de cocina de Arduino escrito por Michael Margolis.

Algunos códigos de este tutorial son editados por Simon Monk. Simon Monk es autor de varios libros relacionados con Hardware de fuente abierta. Están disponibles en Amazon: programación Arduino, 30 proyectos de Arduino para el mal genio y la programación de la Raspberry Pi.

## Servicio al cliente

Como una tecnología de crecimiento continuo y rápida nos seguimos esforzando nuestro mejor para ofrecerle excelentes productos y servicio de calidad en cuanto a cumplir con sus expectativas y la compañía puede llegar a nosotros por simplemente colocar una línea en [Service@elegoo.com](mailto:Service@elegoo.com) o [EUserService@elegoo.com](mailto:EUserService@elegoo.com). Escuchamos su comentario crítico o sugerencia, sería muy importante paranosotros.

Y cualquiera de los problemas y las preguntas que tenga con nuestros productos será rápidamente contestada por nuestros ingenieros experimentados dentro de 12 horas (24 horas durante vacaciones

# Packing list

 [www.elegoo.com](http://www.elegoo.com)



RGB LED  
2PCS



Photoresistor  
(photocell)  
1PC



Resistor  
120PCS



UNO R3  
with USB  
1PC



Tilt Ball Switch  
1PC



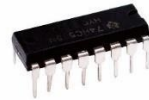
LED  
30PCS



Button(Small)  
5PCS



Active Buzzer  
1PC



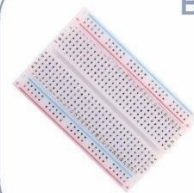
74HC595 IC  
1PC



F-M Dupont Wire  
5PCS



Breadboard  
Jumper Wire  
65PCS



Breadboard  
1PC

Contact us : [service@elegoo.com](mailto:service@elegoo.com)

# Contenido

Lección 0 Instación IDE .....	9
Lección 1 Añadir bibliotecas y Monitor serie abierta .....	20
Lección 2 Blink.....	29
Lección 3 LED.....	40
Lección 4 RGB LED.....	47
Lección 5 Entradas Digitales .....	56
Lección 6 Activar zumbador .....	61
Lección 7 nterruptor de bola de inclinación .....	69
Lección 8 cho LED con 74HC595 .....	109
Lección 9El monitor Serial .....	117
Lección 10Fotocélula .....	123

## Lección 0 Instalación IDE

### Introducción

Entorno de desarrollo integrado (IDE) de Arduino es el software de la plataforma Arduino. En esta lección, usted aprenderá cómo configurar tu ordenador para usar Arduino y cómo establecer sobre las lecciones que siguen.

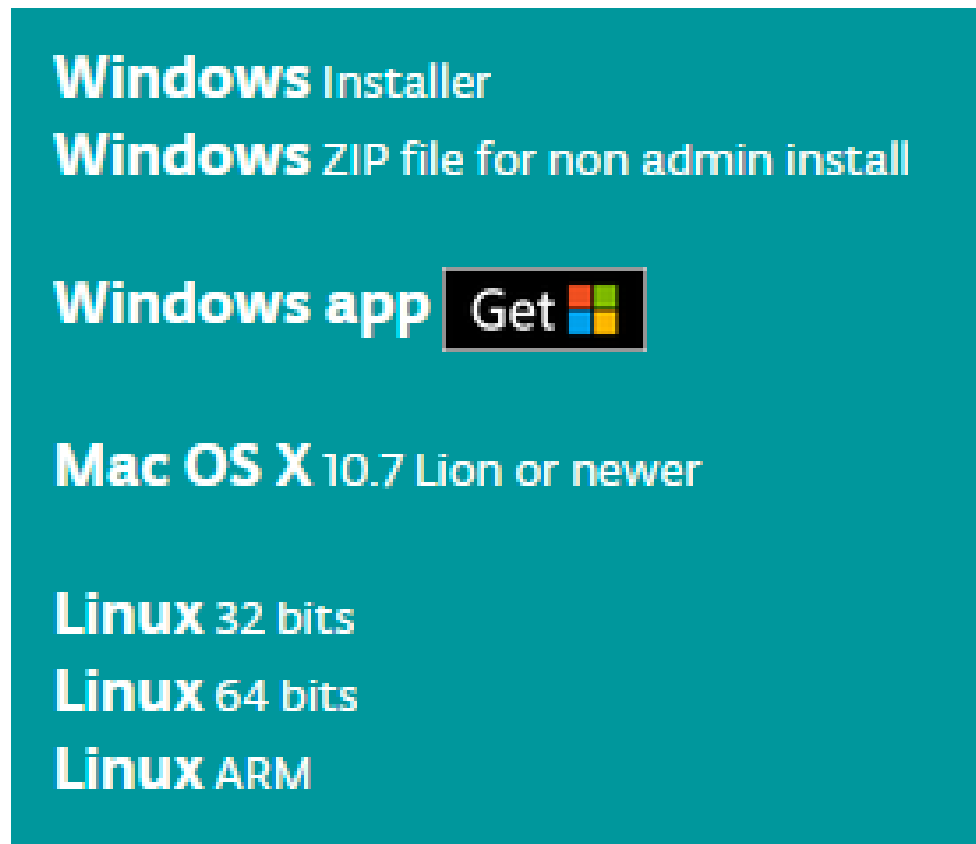
El software de Arduino que se utilizarán para programar tu Arduino está disponible para Windows, Mac y Linux. El proceso de instalación es diferente para las tres plataformas y lamentablemente hay una cierta cantidad de trabajo manual para instalar el software.

**Paso 1:** Ir a <https://www.arduino.cc/en/Main/Software> y a continuación de la página.



**La versión disponible en este sitio web es generalmente la última versión y la versión actual puede ser más reciente que la versión en el cuadro.**

**Paso 2 :** Descargar el desarrollo software que es compatible con el sistema operativo del ordenador. Windows tomar como un ejemplo aquí



Haga click en *Windows Installer*.






## Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



Haga click en DESCARGAR (JUSTDOWNLOAD).

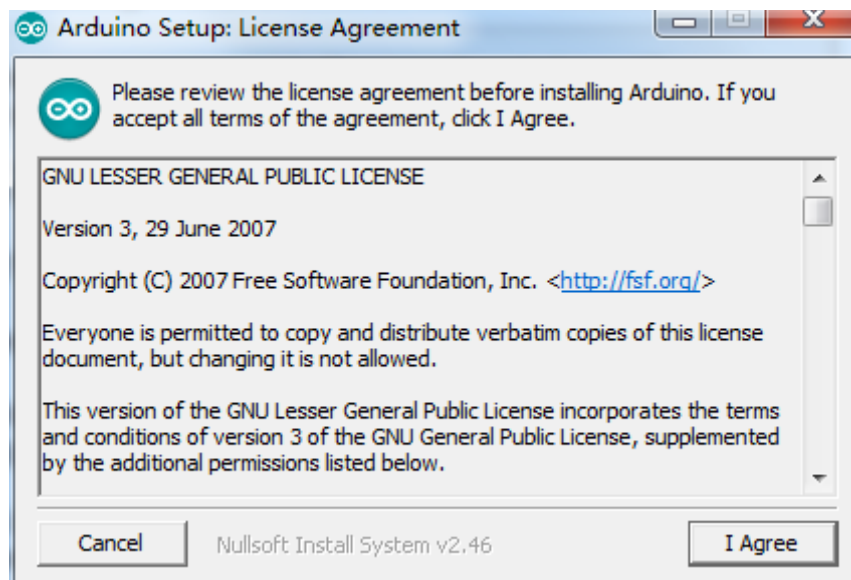
También está disponible en el material que nos proporciona la versión 1.8.0, y las versiones de nuestros materiales son las últimas versiones cuando se hizo este manual.

 arduino-1.8.0-linux32.tar.xz  
 arduino-1.8.0-linux64.tar.xz  
 arduino-1.8.0-macosx.zip  
 arduino-1.8.0-windows.exe  
 arduino-1.8.0-windows.zip

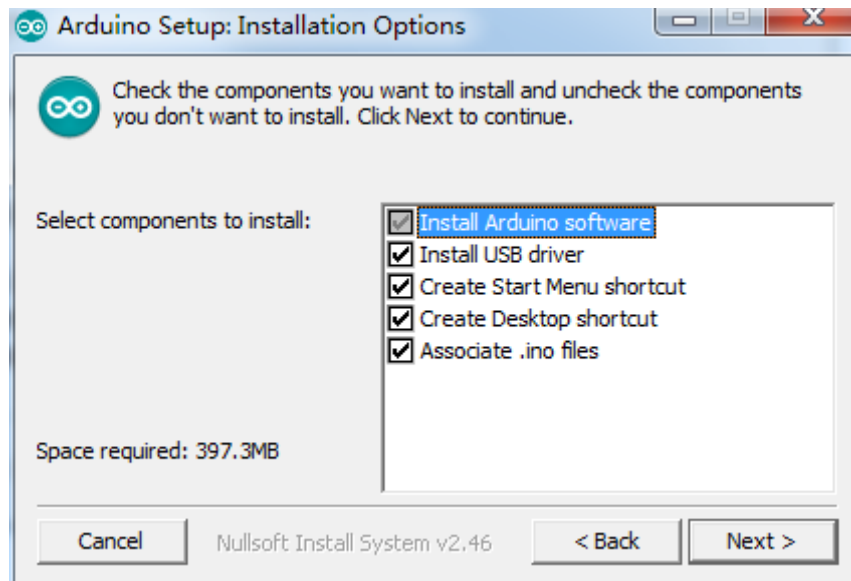
## Instalación de Arduino (Windows)

Instalar Arduino con el exe. Paquete de instalación.

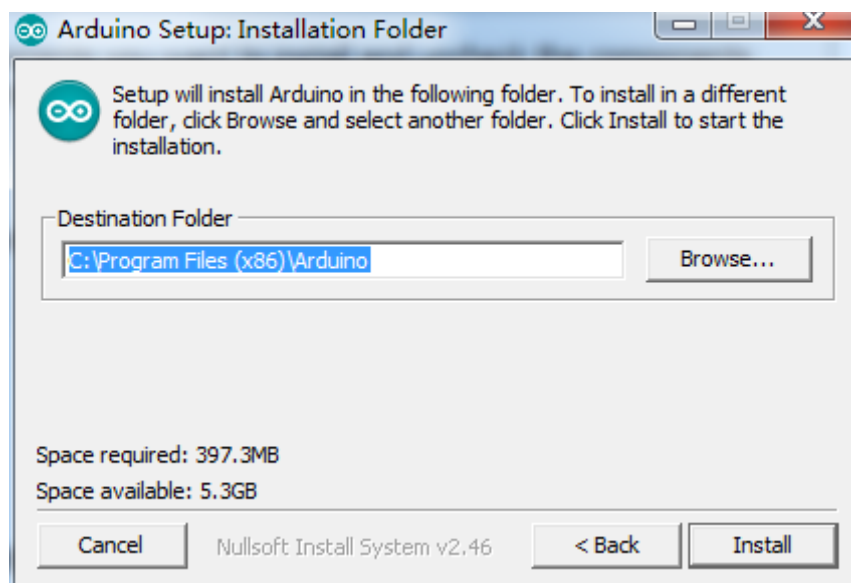
 arduino-1.8.0-windows.exe



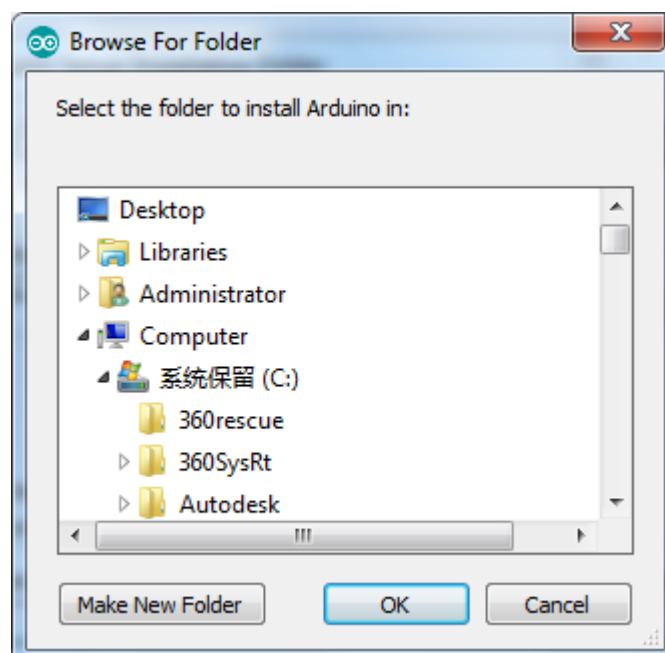
Haga click en *I Agree* to see de esta ventana



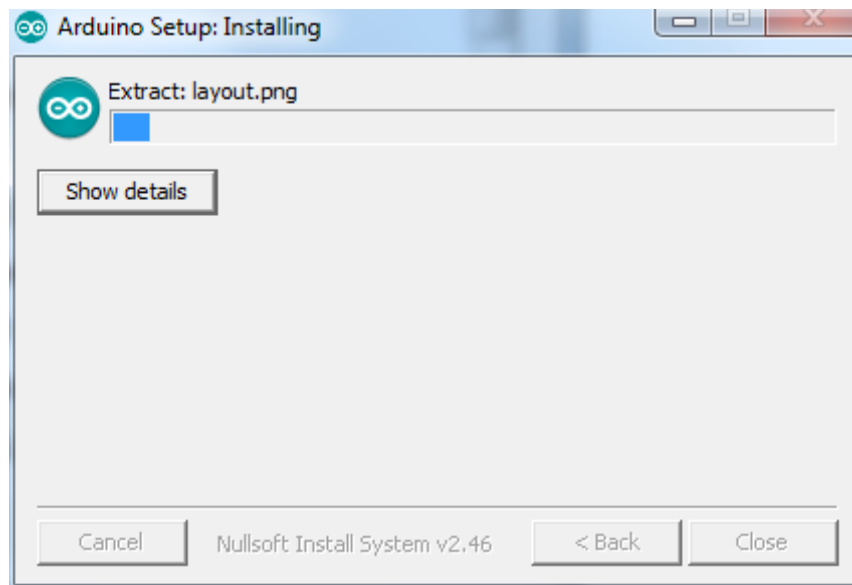
Click *Next*



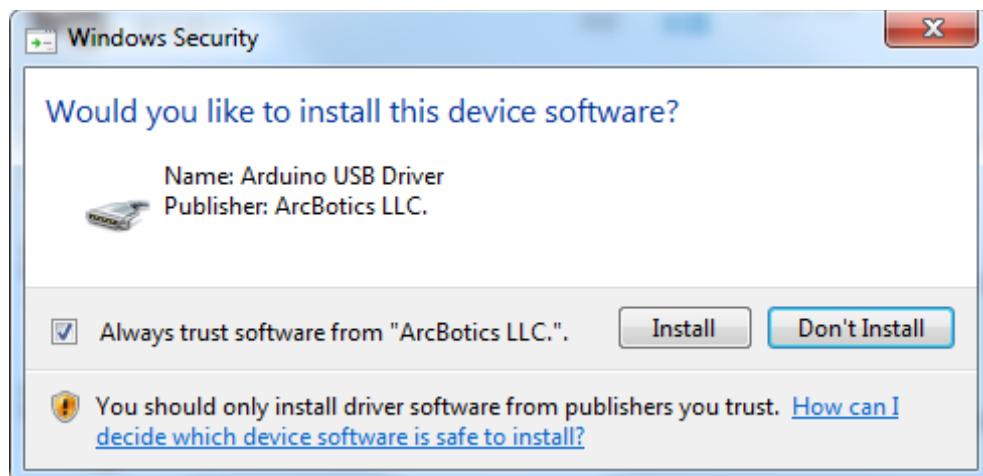
Puede pulsar **examinar...** elegir una ruta de instalación o directamente en el directorio que desee.



Haga Click en *Install* para comenzar la instalacion



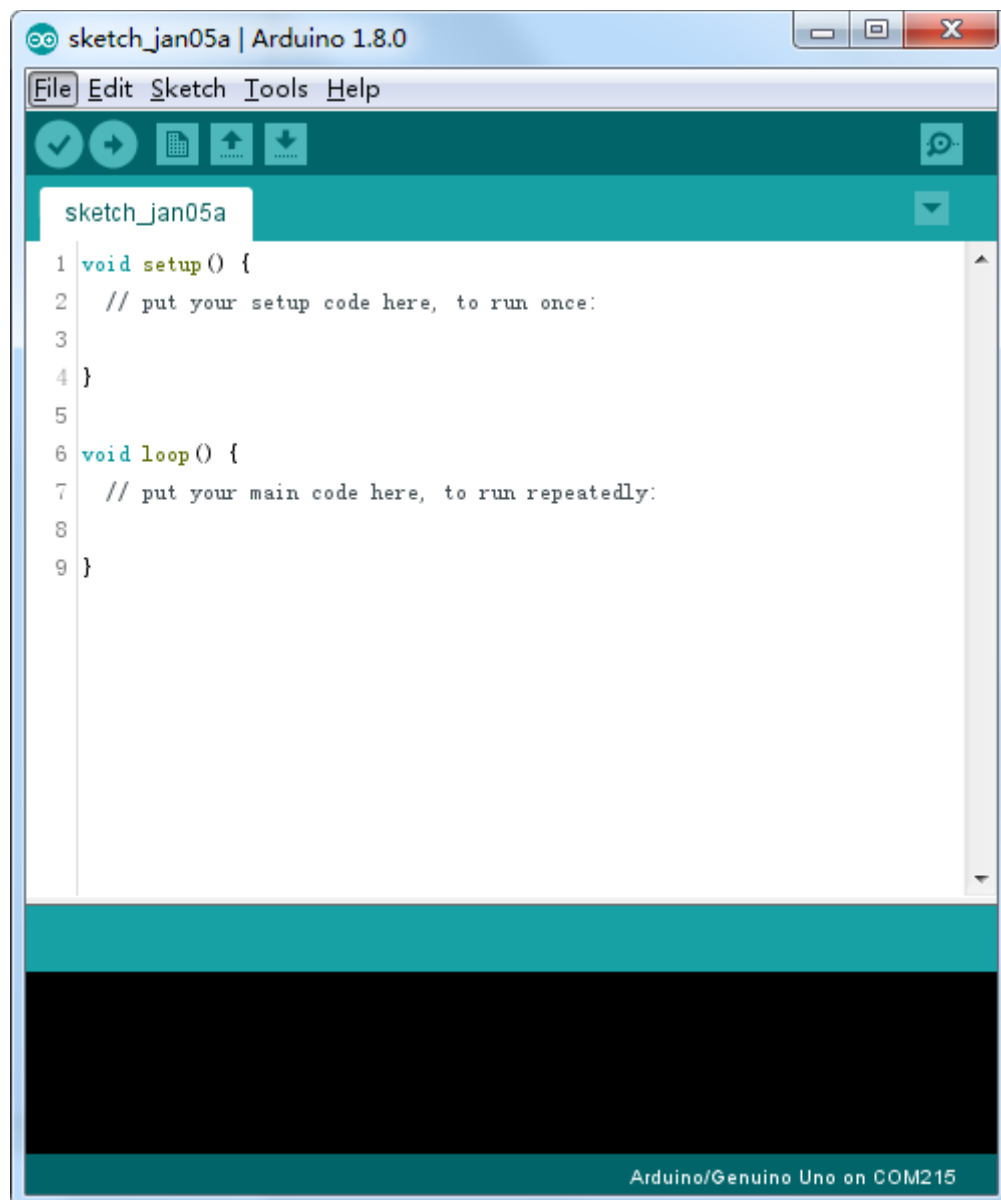
Por último, aparece la siguiente ventana, haga clic en *Installa* para finalizar la instalación.



A continuación, aparece el siguiente icono en el escritorio



Haga doble clic para entrar en el entorno de desarrollo deseado

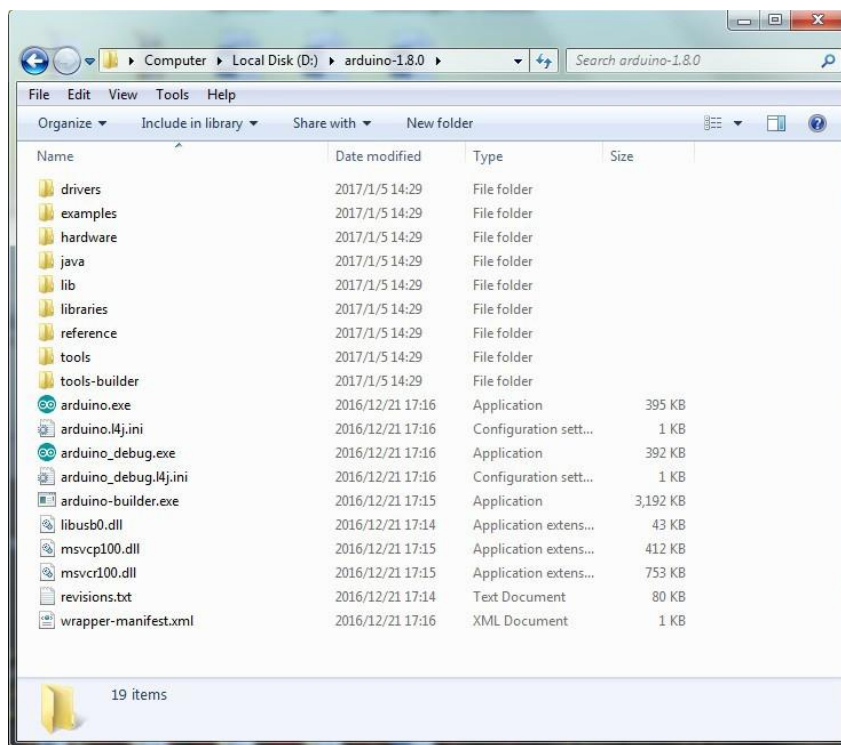


Directamente puede elegir el paquete de instalación para la instalación y omitir los contenidos abajo y saltar a la siguiente sección. Pero si quieres aprender algunos métodos que no sea el paquete de instalación, por favor lea la sección.

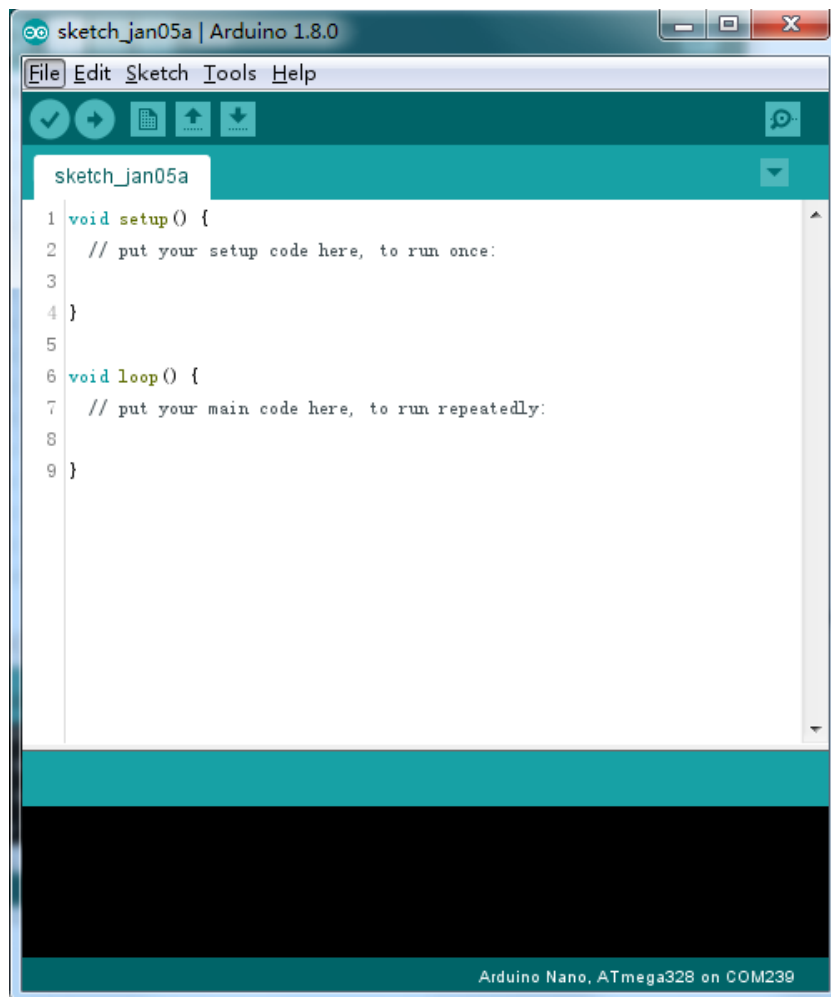
Descomprime el archivo zip descargado, haga doble clic para abrir el programa y entrar en el entorno de desarrollo deseado



arduino-1.8.0-windows.zip



 **arduino.exe**



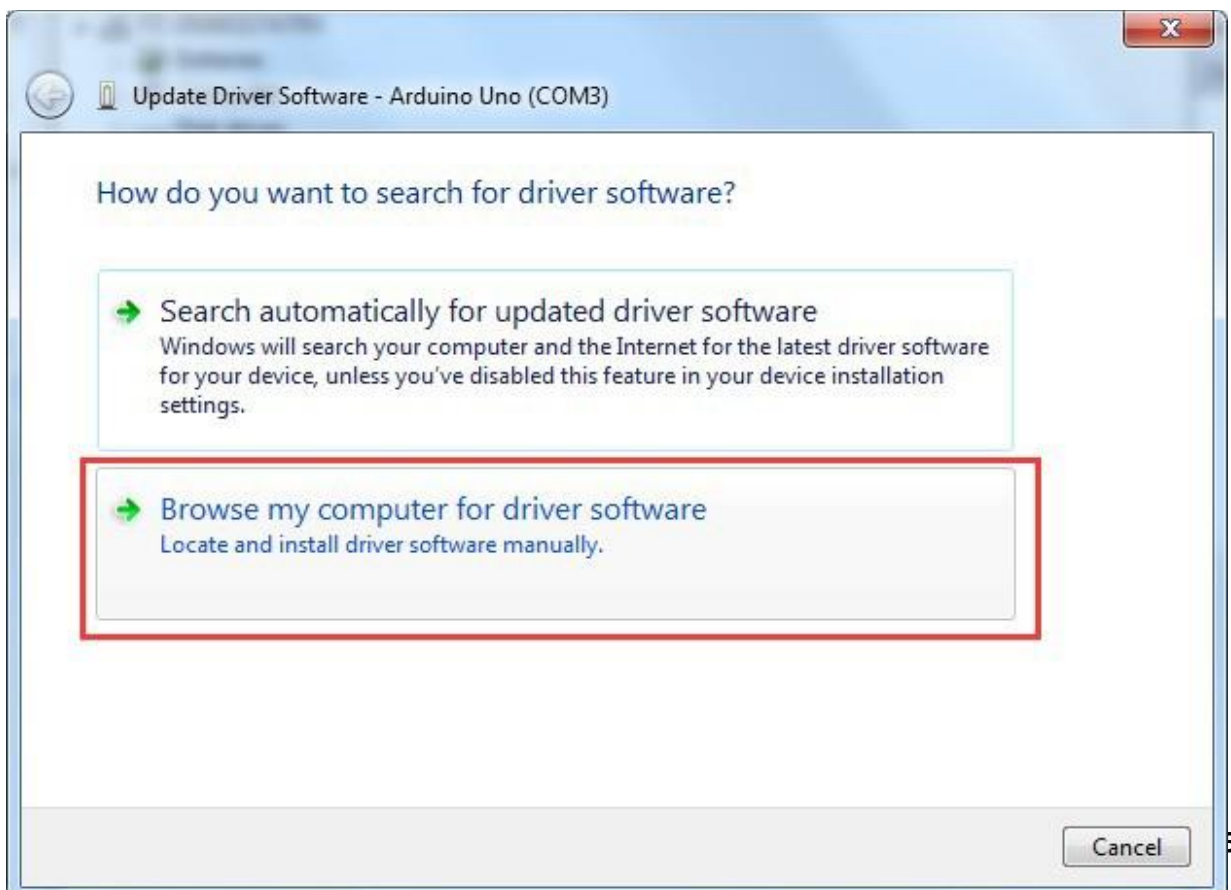
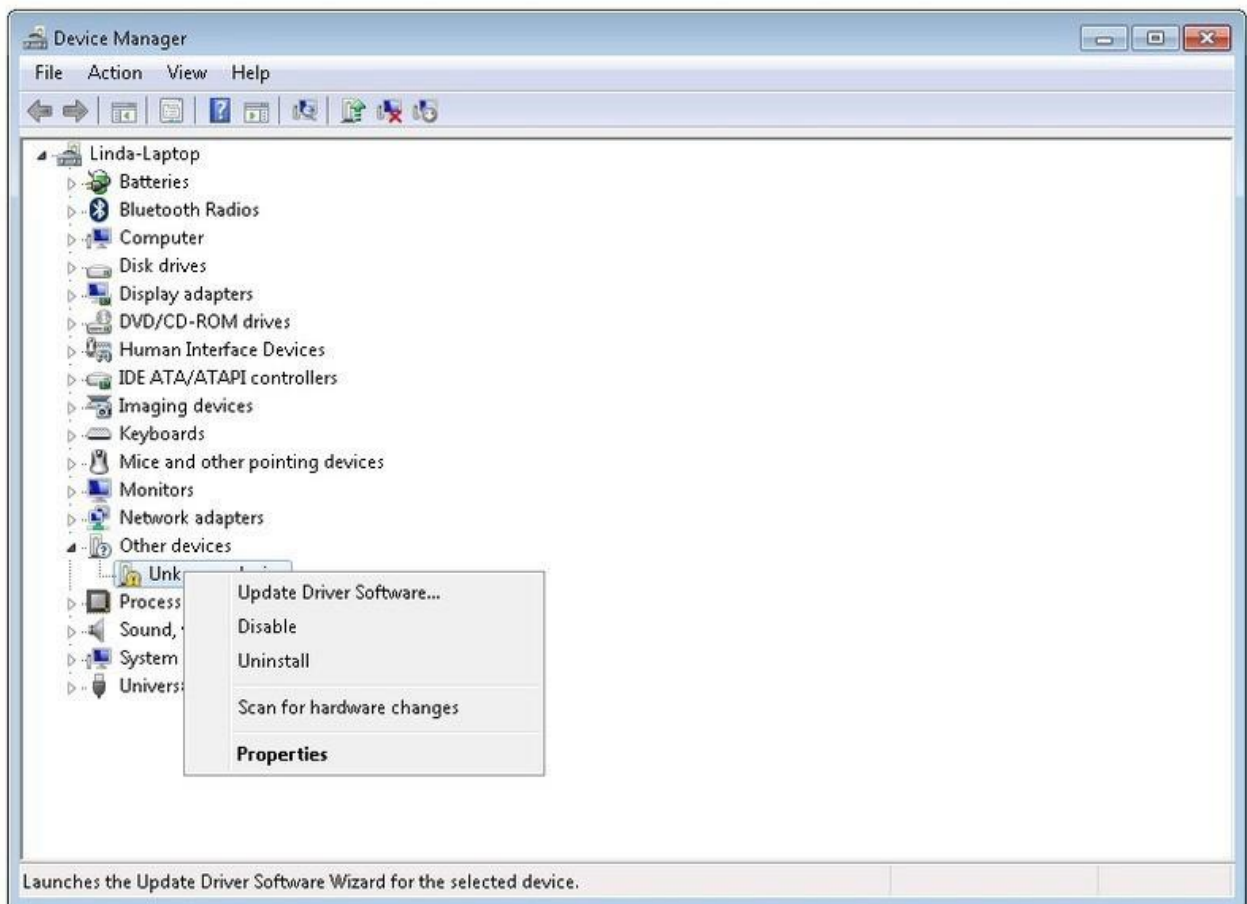
**Sin embargo, este método de instalación necesita instalación de driver..**

La carpeta de Arduino incluye el propio programa de Arduino y los controladores que permiten que el Arduino se conecte al ordenador mediante un cable USB. Antes de que inicie el software de Arduino, vas a instalar a los controladores USB.

Conecte su cable USB en el Arduino y el en el USB en tu ordenador. La luz en el LED se enciende y usted puede obtener un mensaje de 'Found New Hardware' de Windows. Ignore este mensaje y cancele cualquier intento que Windows hace para tratar de instalar los controladores automáticamente.

El mayor método de instalación de los controladores USB debe hacerse desde el administrador de dispositivos. Esto es accesible de diferentes maneras dependiendo de la versión de Windows. En Windows 7, primero tienes que abrir el Panel de Control, luego seleccione la opción de ver los iconos, y usted debe encontrar el administrador de dispositivos en la lista.

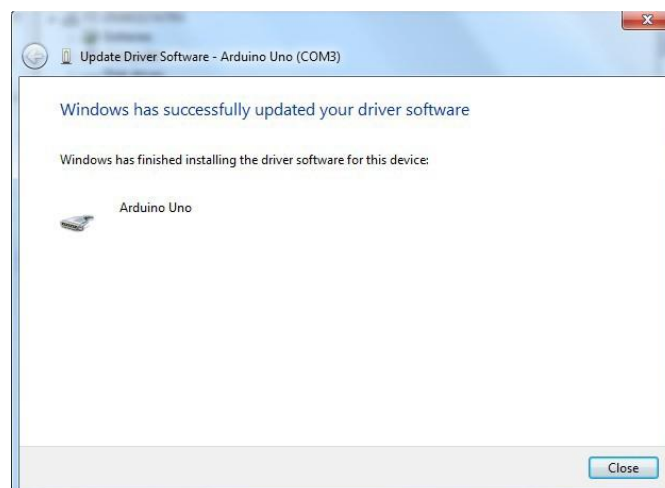
En 'Otros dispositivos', debería ver un icono de 'dispositivo desconocido' con un triángulo amarillo de advertencia junto a él. Se trata de tu Arduino.



Haga clic derecho sobre el dispositivo y seleccione la opción del menú superior (Update Driver Software...). Pedirá a 'Buscar automáticamente software de controlador actualizado' o "Examinar mi PC para el software de controlador". Seleccione la opción Buscar en este ordenador en la ruta del CD , por ejemplo en D:\arduino1.8.0\drivers.



Haga clic en 'Siguiente' y puede obtener una advertencia de seguridad, si es así, permitir que el software a instalar. Una vez instalado el software, usted recibirá un mensaje de confirmación.



Los usuarios de Windows pueden omitir las instrucciones de instalación, para sistemas Mac y Linux y saltar a la lección 1.

Usuarios de Mac y Linux pueden seguir leyendo esta sección

### Instalación de Arduino (Mac OSX)

Descargar y descomprimir el archivo zip, haga doble clic en Arduino.app para entrar en el IDE de Arduino; el sistema le pedirá que instale la biblioteca de tiempo de ejecución de Java si no lo tienes en tu ordenador. Una vez finalizada la instalación se puede ejecutar el IDE de Arduino.

 [arduino-1.8.0-macosx.zip](#)


### Instalación de Arduino (Linux)

Usted tendrá que utilizar el comando de instalación. Si está utilizando el sistema de Ubuntu, se recomienda instalar el IDE de Arduino desde el centro de software de Ubuntu.

 [arduino-1.8.0-linux32.tar.xz](#)

 [arduino-1.8.0-linux64.tar.xz](#)

**Consejos: Si tienes problemas en la instalación de los controladores, consulte el UNO R3, MEGA, NANO controlador preguntas más frecuentes.**

 [UNO R3, MEGA, NANO DRIVER FAQ](#)

# Lección 1 Añadir bibliotecas y Monitor serie abierta

## Instalación de bibliotecas adicionales de Arduino

Una vez que esté instalado con el software de Arduino y utilizando las funciones integradas, puede que desee ampliar la capacidad de tu Arduino con bibliotecas adicionales.

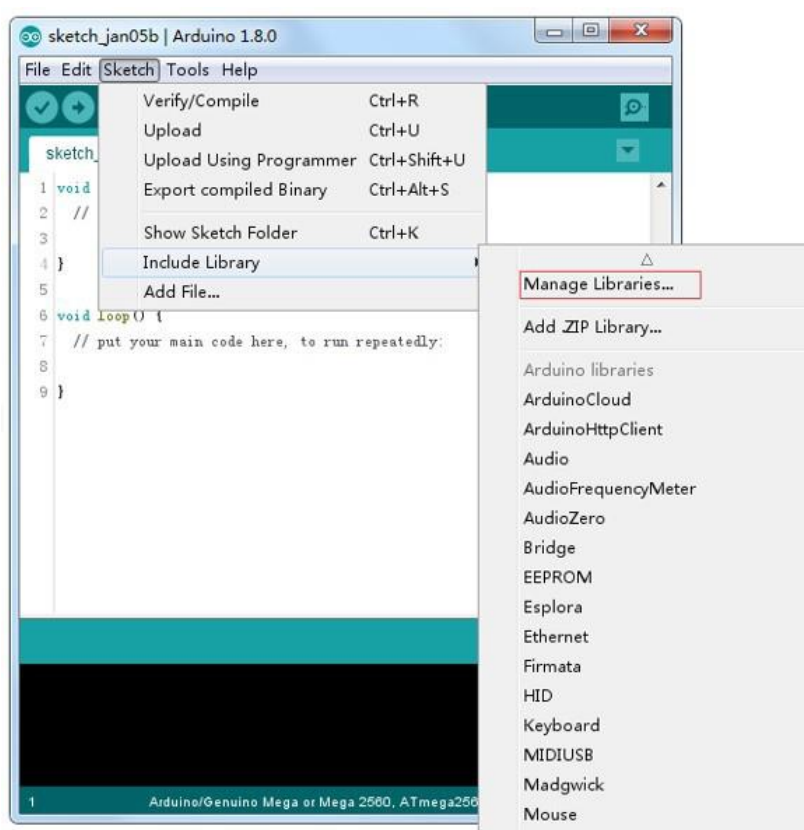
### ¿Cuáles son las bibliotecas?

Las bibliotecas son una colección de código que hace que sea fácil para usted conectar a un sensor, pantalla, módulo, etcetera. Por ejemplo, la librería LiquidCrystal incorporada facilita hablar con pantallas LCD de caracteres. Hay cientos de librerías adicionales disponibles en Internet para su descarga. Las bibliotecas integradas y algunas de estas bibliotecas adicionales aparecen en la referencia. Para utilizar las bibliotecas adicionales, necesitará instalarlas.

### Cómo instalar una biblioteca

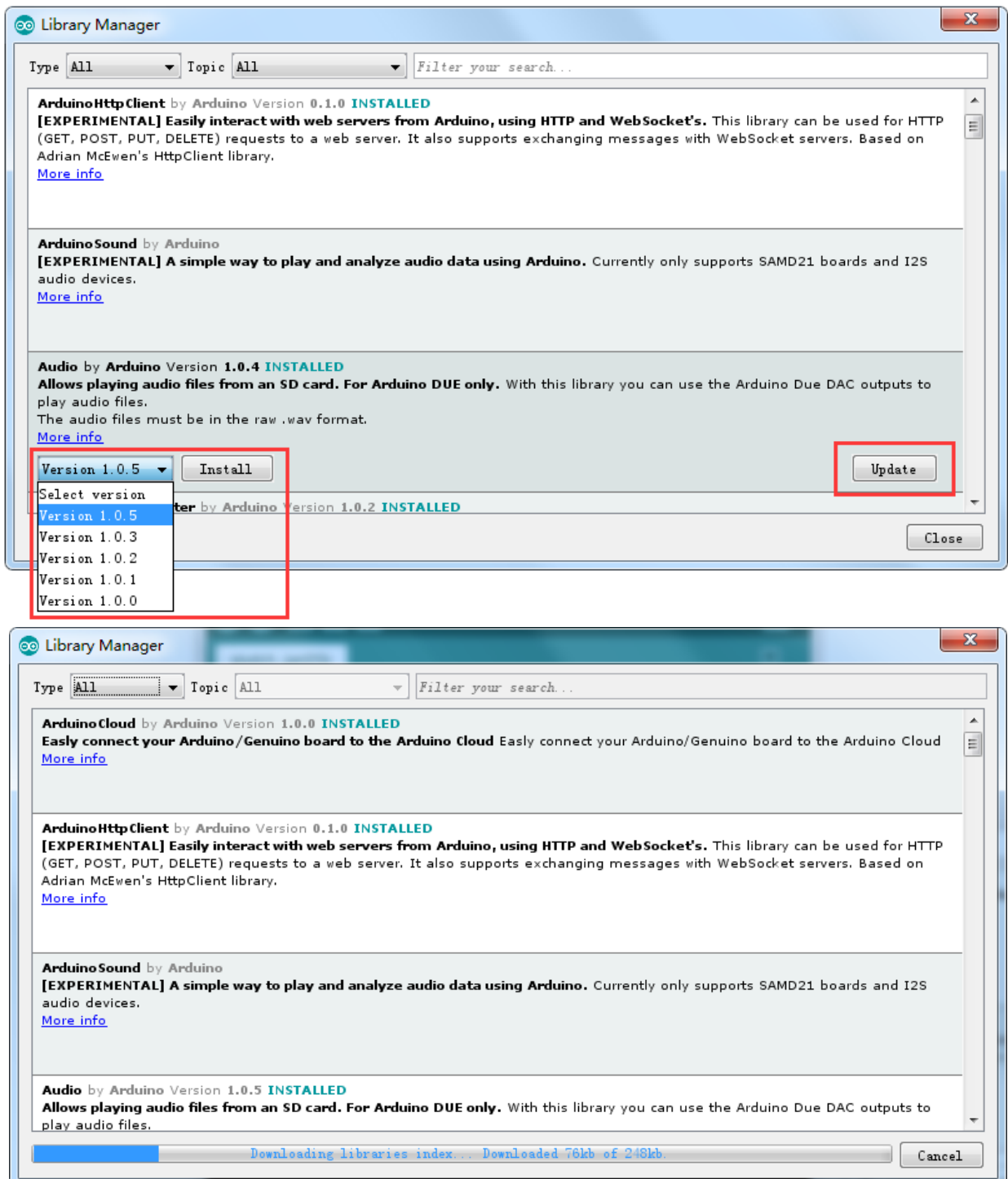
Mediante el administrador de la biblioteca

Para instalar una nueva biblioteca en el IDE de Arduino se puede utilizar el administrador de biblioteca (disponible desde IDE versión 1.8.0). Abra el IDE y haga clic en el menú "Dibujo" y luego la biblioteca incluyen > Gestión de bibliotecas.

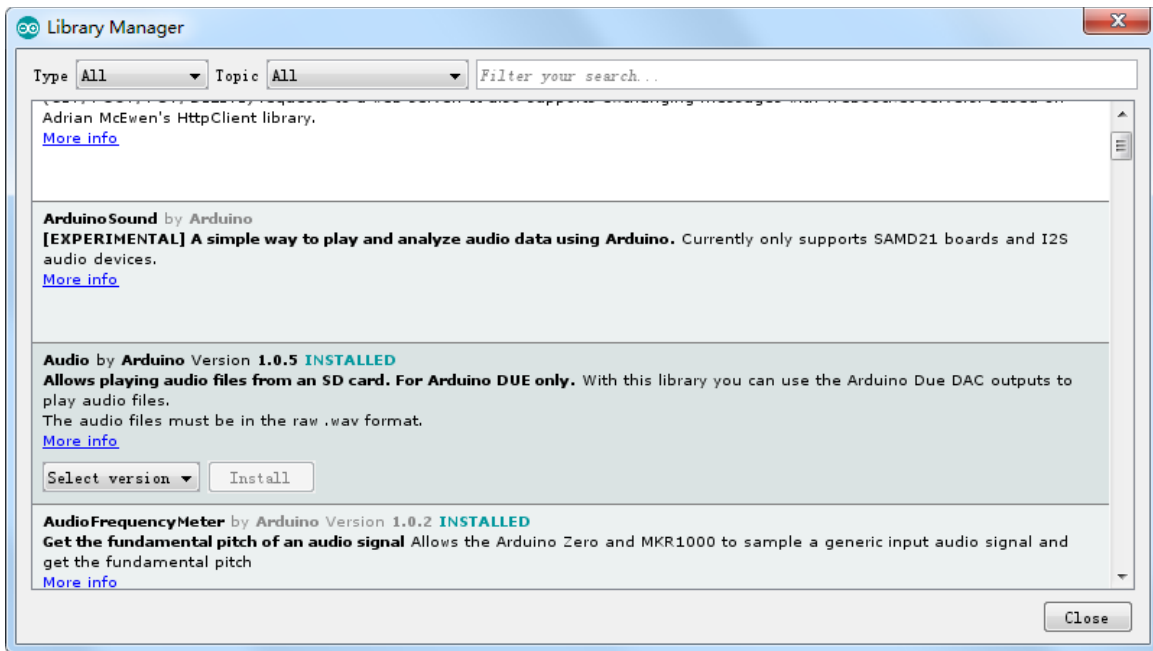


Entonces el director de la biblioteca se abrirá y usted encontrará una lista de bibliotecas que ya están instaladas o listas para su instalación. En este ejemplo vamos a instalar la biblioteca de puente. Desplazarse por la lista para encontrarla, a continuación, seleccione la versión de la biblioteca que desea instalar. A veces sólo está disponible una versión de la biblioteca. Si no aparece el menú de selección de versión, no te preocupes: es normal.

Hay veces que tienes que esperar, tal como se muestra en la figura. Por favor actualice y esperar



Finalmente haga click en instalar y esperar a que el IDE instale la nueva biblioteca. La descarga puede tardar un tiempo dependiendo tu velocidad de conexión. Una vez haya terminado, debe aparecer una etiqueta instalada junto a la biblioteca de puente. Una vez instalada puede cerrar el administrador de la biblioteca.

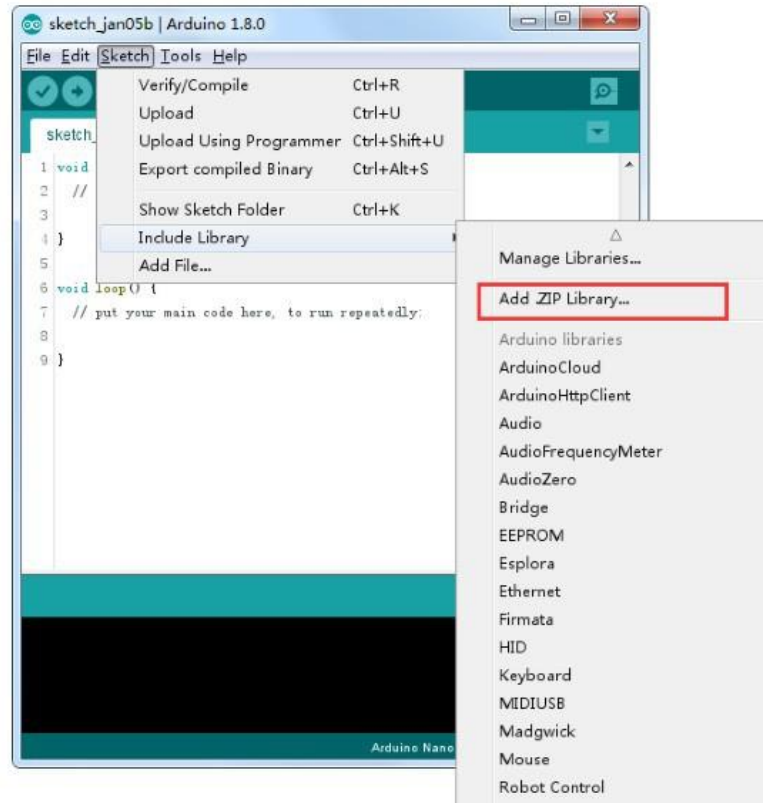


Ahora ya puede encontrar la nueva biblioteca disponible en el menú de biblioteca. Si quieres añadir tu propia biblioteca vaya a abrir un nuevo tema en [Github](#).

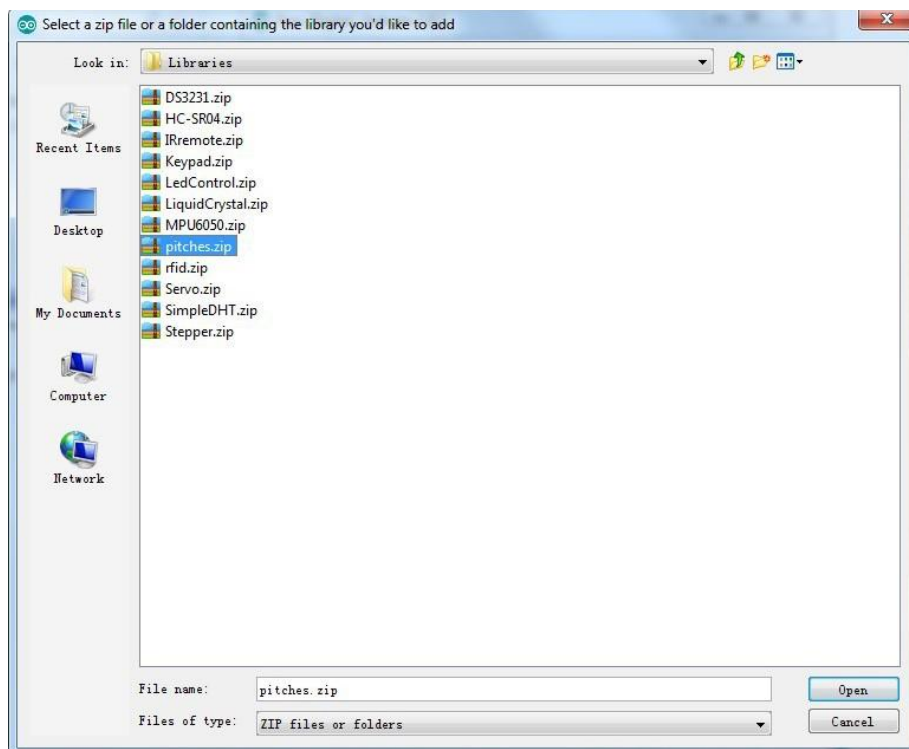
## Importar una biblioteca de .zip

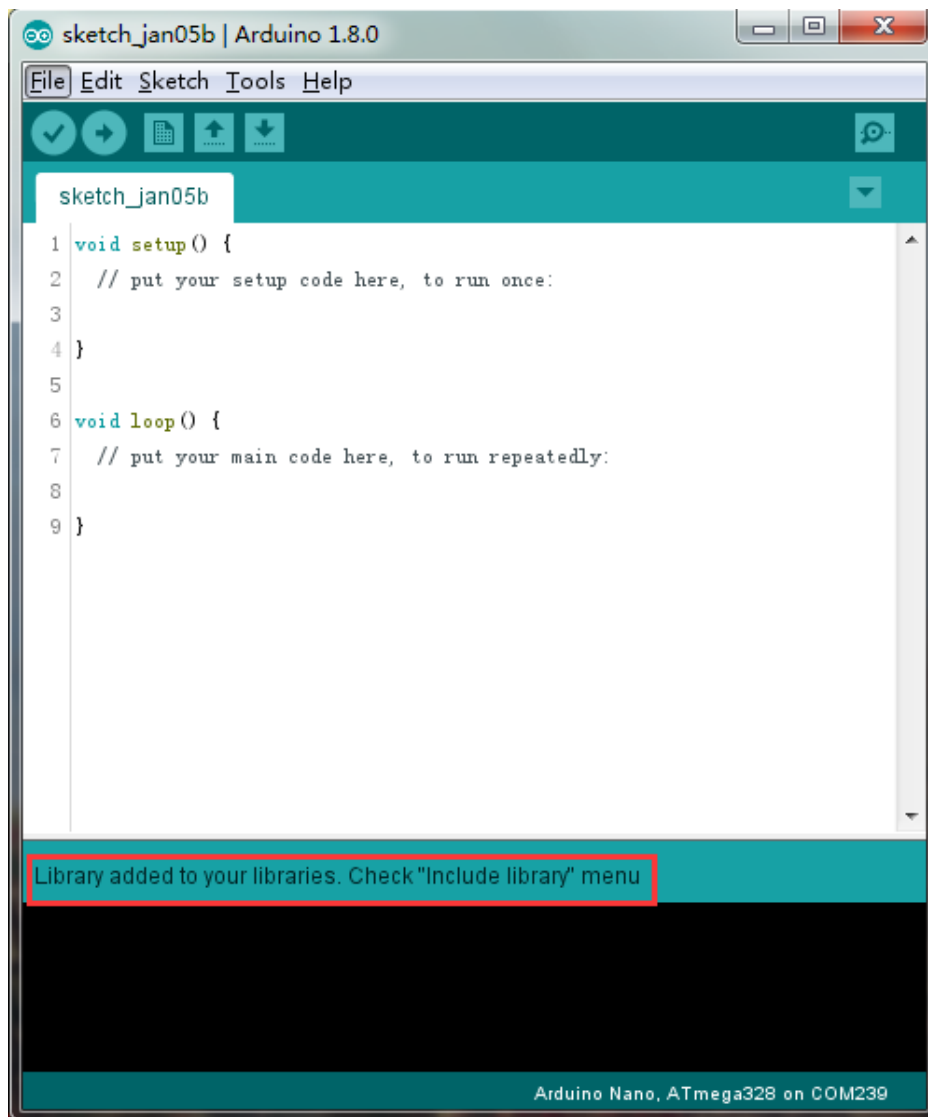
Las bibliotecas se distribuyen a menudo como un archivo ZIP o una carpeta. El nombre de la carpeta es el nombre de la biblioteca. Dentro de la carpeta será un archivo .cpp, un archivo .h y a menudo un fichero llamado keywords.txt, carpeta de ejemplos y otros archivos requeridos por la biblioteca. A partir de la versión 1.0.5, puede instalar bibliotecas de partido 3<sup>o</sup> en el IDE. Descomprime la librería descargada y dejarlo como está.

En el IDE de Arduino, desplácese a Sketch > Biblioteca incluyen. En la parte superior de la lista desplegable, seleccione la opción "agregar. BibliotecaZIP".



Se le pedirá para seleccionar la biblioteca que desea añadir. Desplácese hasta la ubicación del archivo .zip y luego pincha en abrir.





Volver al dibujo > menú de biblioteca de importación. Ahora debe ver la biblioteca en la parte inferior del menú desplegable. Está listo para ser utilizado en su lista. El archivo zip se ha incorporado en la carpeta de bibliotecas en el directorio de plantillas de Arduino.

**Nota:** la biblioteca estará disponible para utilizar en los dibujos, pero los ejemplos de la biblioteca no serán expuestos en el archivo > ejemplos hasta después del IDE se ha reiniciado. Los dos son los enfoques más comunes. Asimismo, pueden manejarse sistemas MAC y Linux. El manual de instalación que se introducirá por debajo como alternativa puede usarse rara vez y los usuarios que no lo necesiten pueden saltarlo

## Manual de instalación

Para instalar la biblioteca, primero salga de la aplicación de Arduino. Luego descomprima el archivo ZIP que contiene la biblioteca. Por ejemplo, para instalar una librería llamada "ArduinoParty", descomprime ArduinoParty.zip. Debería contener una carpeta called ArduinoParty, con archivos como ArduinoParty.cpp y ArduinoParty.h dentro. (Si los archivos .cpp y .h no en una carpeta, debe crear uno. En este caso, usted sería hacer una

carpeta llamada "ArduinoParty" y copiar todos los archivos que estaban en el archivo ZIP, como ArduinoParty.cpp y ArduinoParty.h.)

Arrastre la carpeta de ArduinoParty en esta carpeta (la carpeta de bibliotecas). Bajo Windows, lo probable es que se llamará "My Documents\Arduino\libraries". Para usuarios de Mac, lo probable es que se llamará "Bibliotecas de Arduino de documentos". En Linux, será la carpeta "libraries" en susketchbook.

La carpeta de la biblioteca Arduino debe ahora este aspecto (en Windows):

`My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp`

`My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h`

`My Documents\Arduino\libraries\ArduinoParty\examples`

o como esta (en Mac y Linux):

`Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp`

`Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h`

`Documentos/Arduino/bibliotecas/ArduinoParty/ejemplos`

....

Puede haber más archivos que solo los .cpp y .h, sólo asegúrese de que están todos allí. (La biblioteca no funcionará si pones los archivos .cpp y .h en la carpeta de bibliotecas o si está anidados en una carpeta extra. Visualizador:

`Documents\Arduino\libraries\ArduinoParty.cpp` y

`Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp` no funcionarán.)

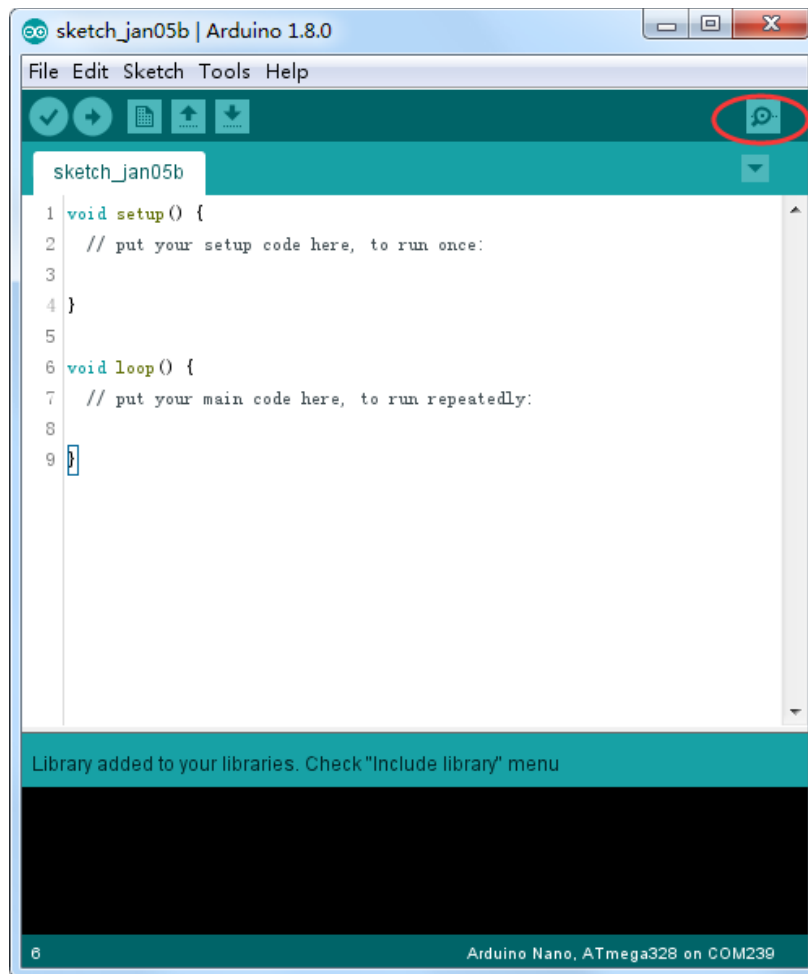
Reiniciar la aplicación Arduino. Asegúrese de que la nueva biblioteca aparece en el directorio -> elemento de menú de biblioteca de importación del software. ¡Eso es todo! ¡Ha instalado una biblioteca!

## **Arduino serie Monitor (Windows, Mac, Linux)**

Entorno de desarrollo integrado (IDE) de Arduino es el software de la plataforma Arduino. Y, porque utilizando un terminal es una gran parte del trabajo con Arduinos y otros microcontroladores, decidieron incluir un terminal de serie con el software. En el entorno de Arduino, esto se llama al Monitorserie.

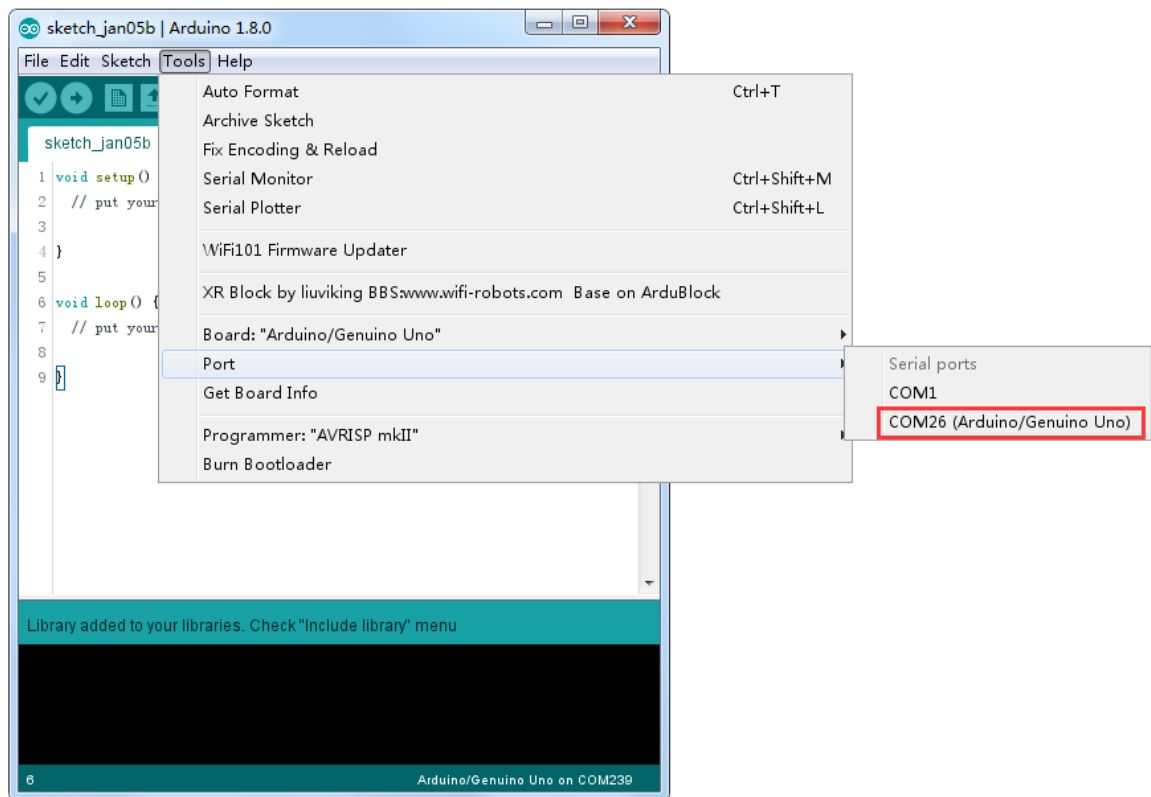
## Realizar la conexión

Monitor de serie viene con cualquier versión del IDE Arduino. Para abrirlo, simplemente haga clic en el icono Serial Monitor.

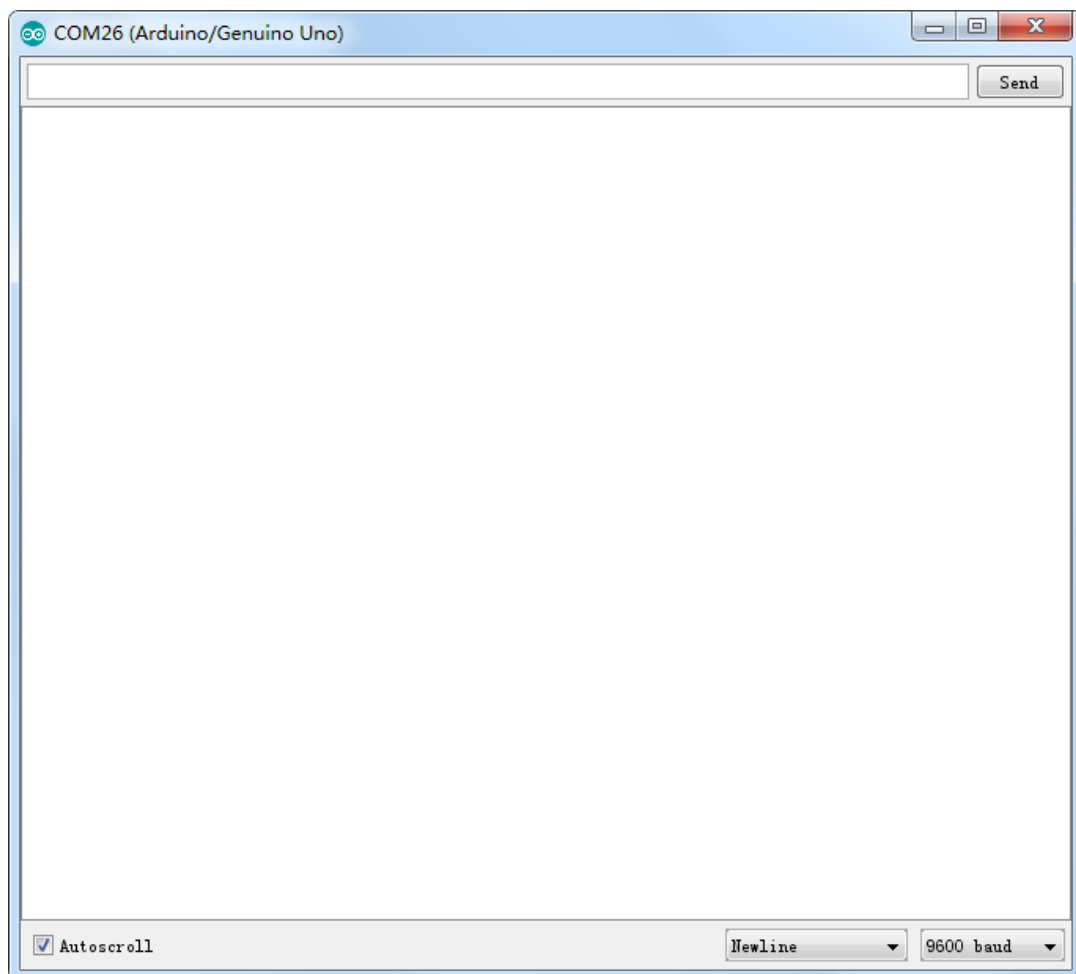


Seleccionar cuál de los puertos a abrir en el Monitor Serial es lo mismo que seleccionar un puerto para cargar código de Arduino. Vaya a herramientas -> Serial Port y seleccione el puerto correcto.

**Consejos:** Elegir el mismo puerto COM que tienes en el administrador de dispositivos.



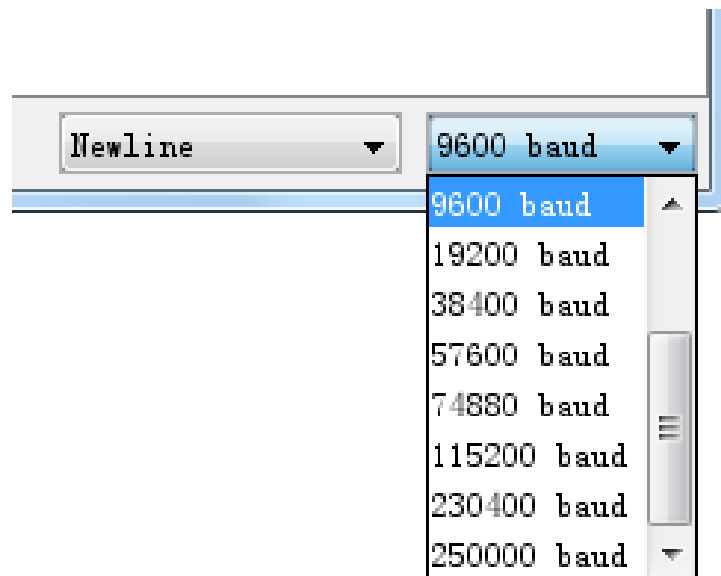
Una vez abierto, debería ver algo comoesto:



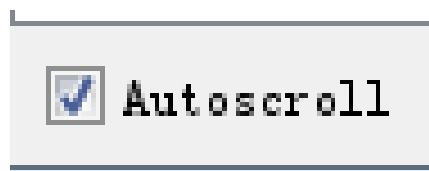
## Configuración

El Monitor Serial ha limitado opciones, pero lo suficiente para manejar la mayoría de sus necesidades de comunicación serial. El primer ajuste que se puede modificar es la velocidad en baudios. Haga clic en la velocidad en baudios tasa lista desplegable para seleccionar la velocidad correcta. (9600 baudios)

)



Por último, puede establecer el terminal desplazamiento automático o no marcando la casilla en la esquina inferior izquierda



### Pros

El Monitor Serial es una gran manera rápida y fácil para establecer una conexión en serie con el Arduino. Si ya trabaja en el IDE de Arduino, no hay realmente ninguna necesidad de abrir un terminal separado para Mostrardatos

### Contras

La falta de valores deja mucho que desear en el Monitor Serial, y, para comunicaciones serie avanzadas, no puede hacer el truco.

## Lección 2 Blink

### Resumen

En esta lección, usted aprenderá cómo programar el tablero de regulador UNO R3 a parpadear el LED integrado de Arduino y cómo descargar programas pasos básicos

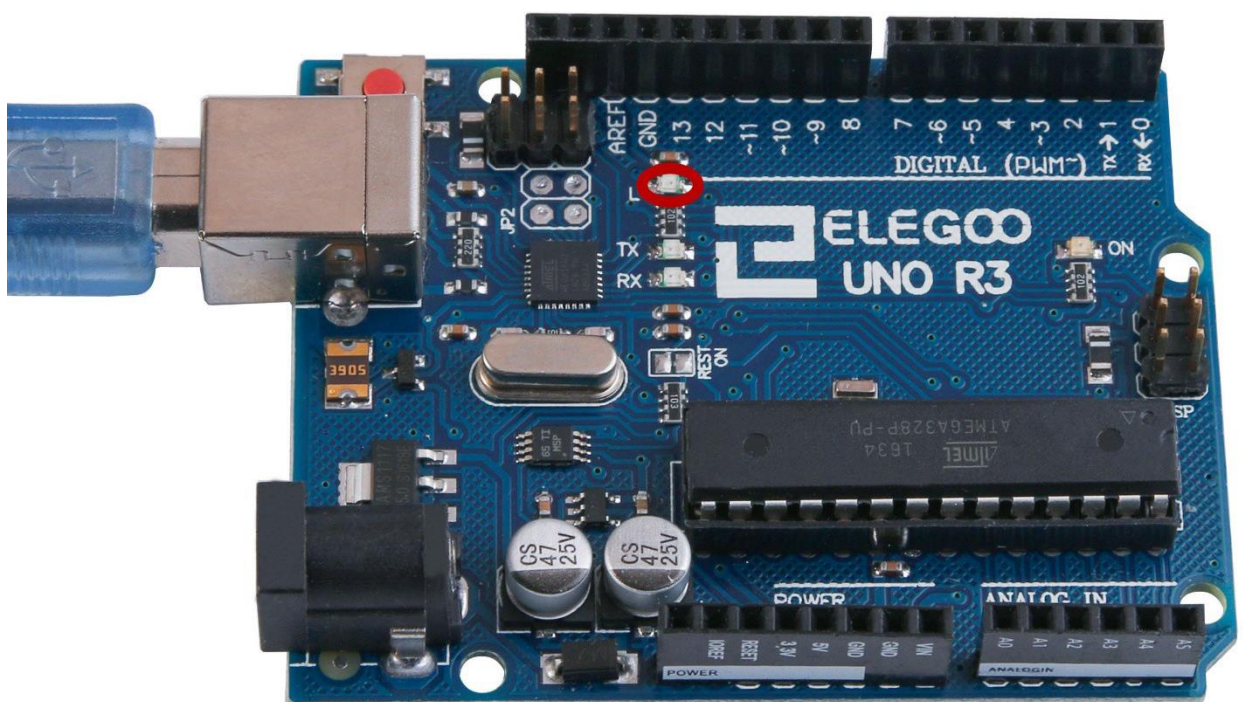
### Componente necesario:

(1) x Elegoo Uno R3

### Principal

La placa de UNO R3 tiene unas filas de conectores a ambos lados que se utilizan para conectar varios dispositivos electrónicos y plug-in 'escudos' que amplía su capacidad.

También tiene un indicador luminoso que usted puede controlar desde sus plantillas. Este LED está construido sobre el tablero de UNO R3 y se refiere a menudo como la 'L' LED ya que es como se etiqueta en el tablero.



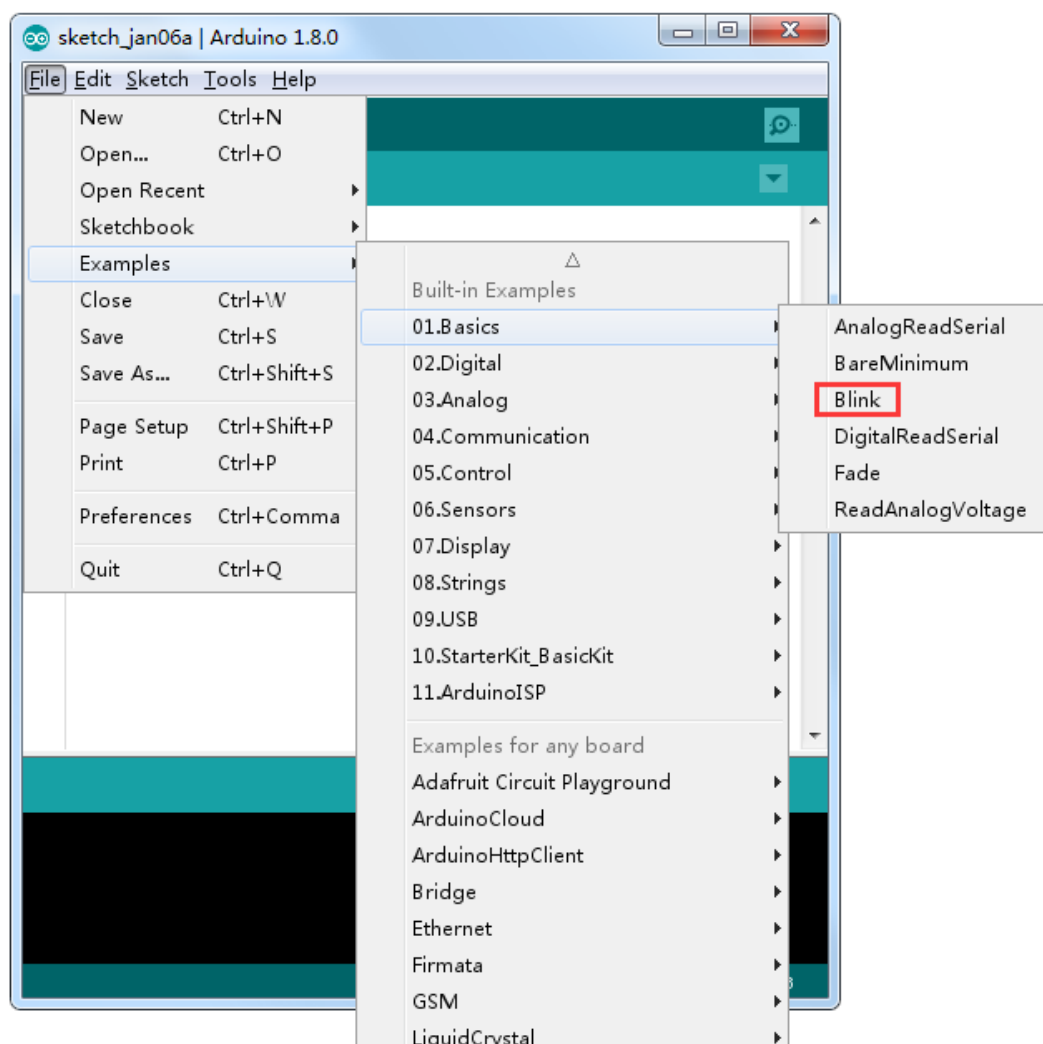
Usted puede encontrar que 'L' de la placa UNO R3 LED parpadea cuando se conecta a un enchufe del USB. Esto es porque las placas se envían generalmente con el sketch de 'Blink' pre-instalado.

En esta lección, vamos a reprogramar el tablero UNO R3 con nuestro propio directorio de Blink y luego cambiar la tasa a la que parpadea.

En la lección 0, configurar el IDE de Arduino y aseguró que podría encontrar el puerto serie correcto para conectarse a la placa UNO R3. Ahora ha llegado el momento para poner el programa de prueba y la placa de UNO R3.

El IDE de Arduino incluye una gran colección de dibujos de ejemplo, se puede cargar y usar. Esto incluye un directorio de ejemplo para hacer el parpadeo del LED de 'L'.

Cargar el sketch de 'Blink' que encontrarás en el sistema de menús del IDE bajo archivo > ejemplos > 01 conceptos básicos



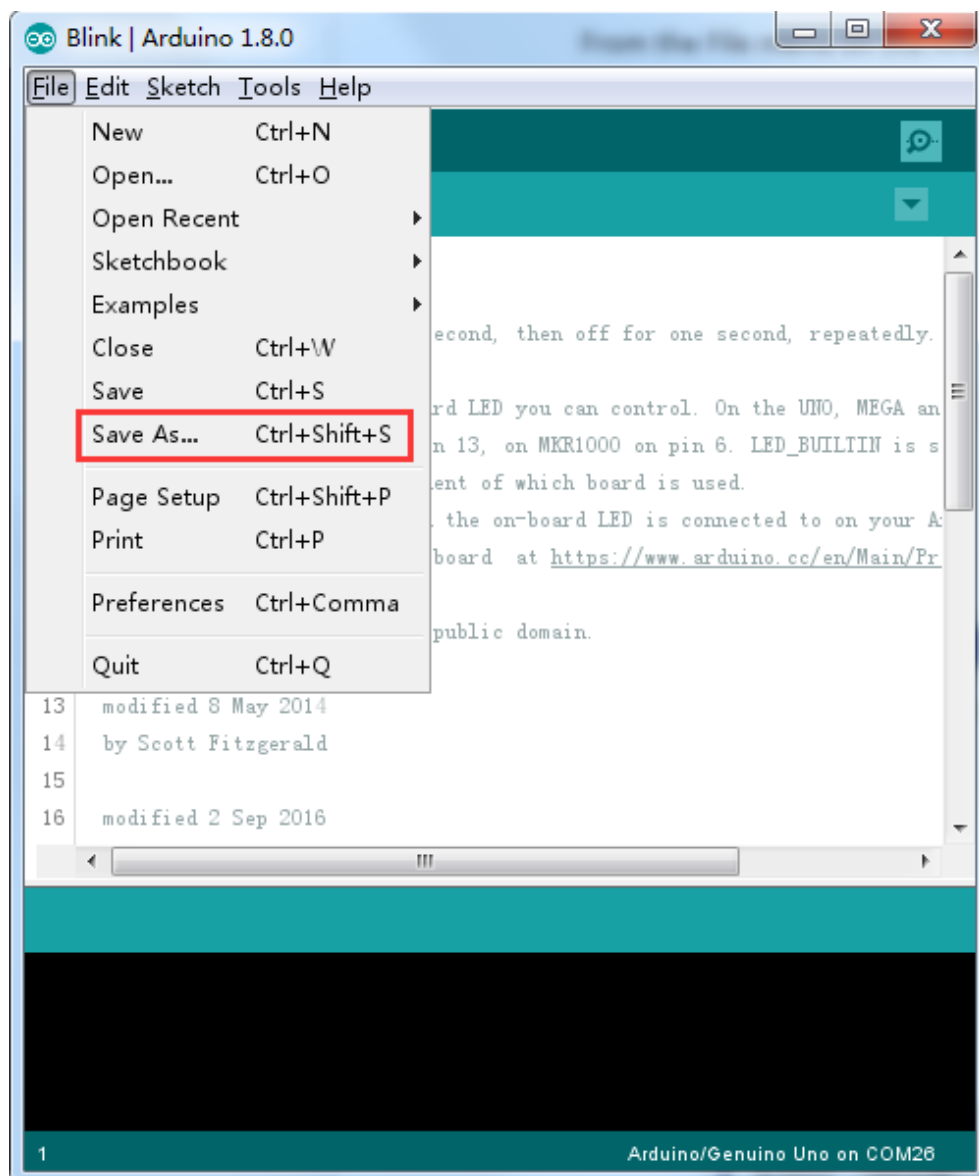
Cuando se abre la ventana de dibujo, agrandarla para que puedan ver el dibujo completo en la ventana.



Los dibujos de ejemplo incluidos con el IDE de Arduino son de 'sólo lectura'. Es decir, puedes subir a una Junta de UNO R3, pero si cambia, no se puede guardar como el archivo mismo.

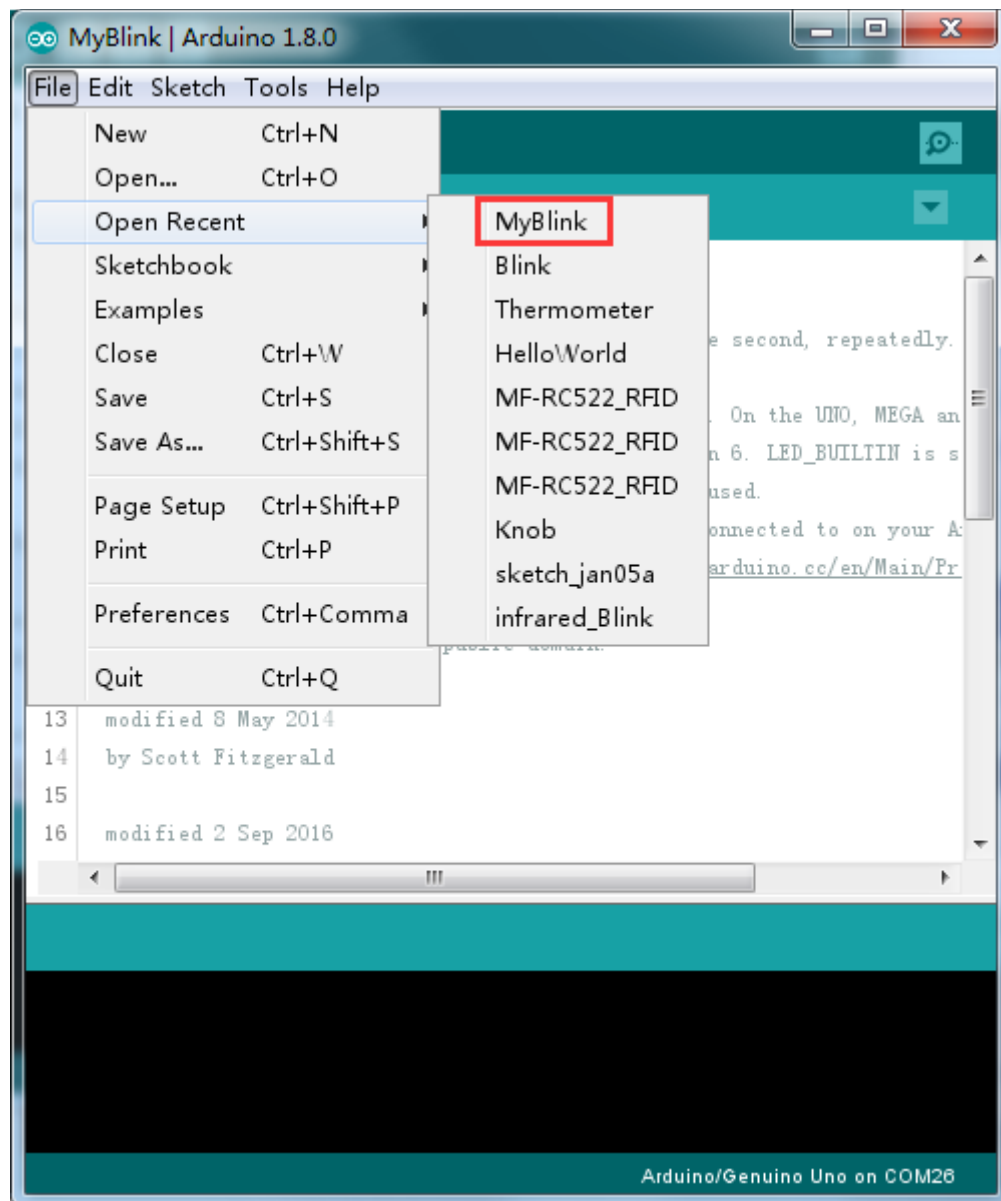
Puesto que vamos a cambiar este sketch, lo primero que tienes que hacer es guardar su propia copia.

En el menú archivo en el IDE de Arduino, seleccione 'Guardar como.' y guarde el dibujo con el nombre 'MyBlink'

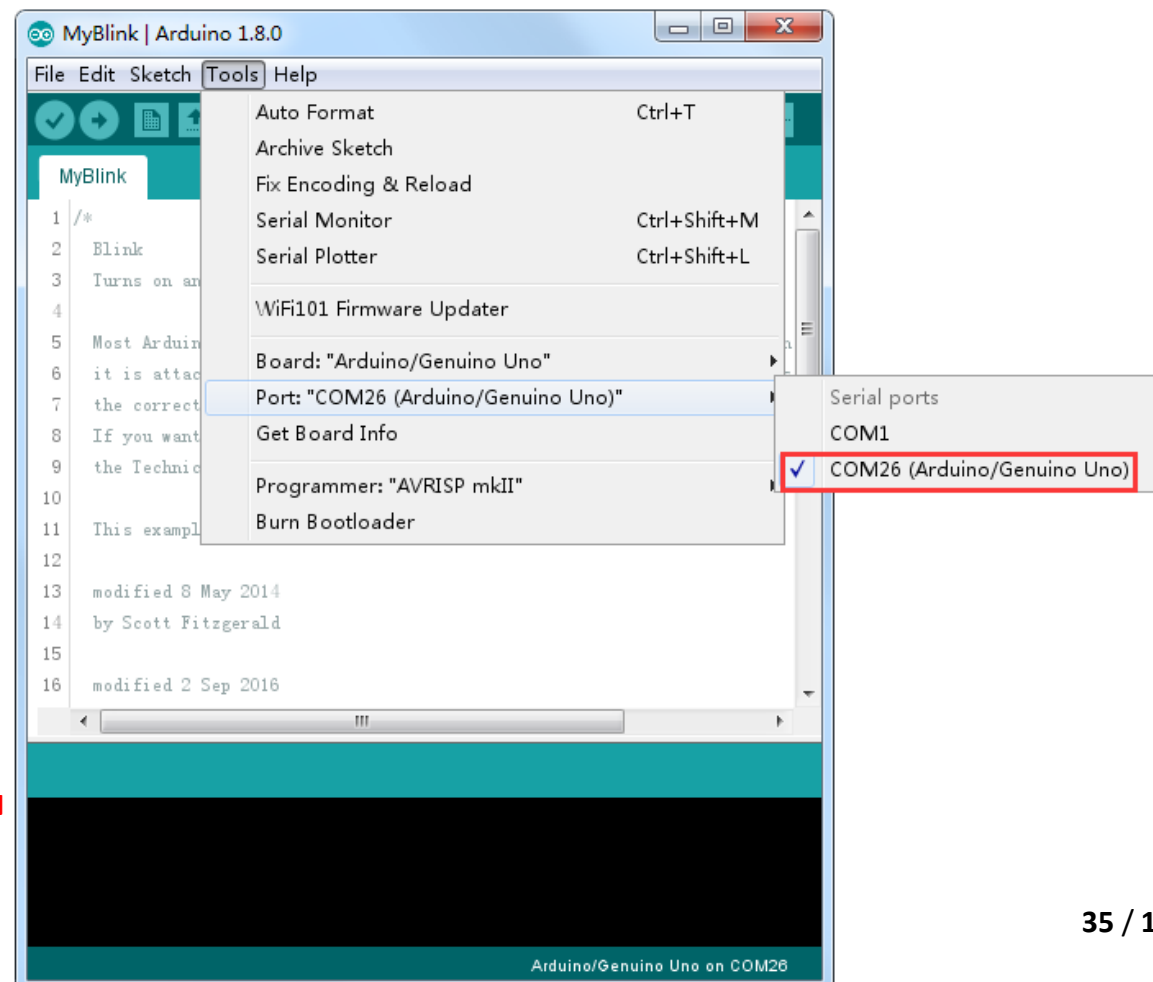
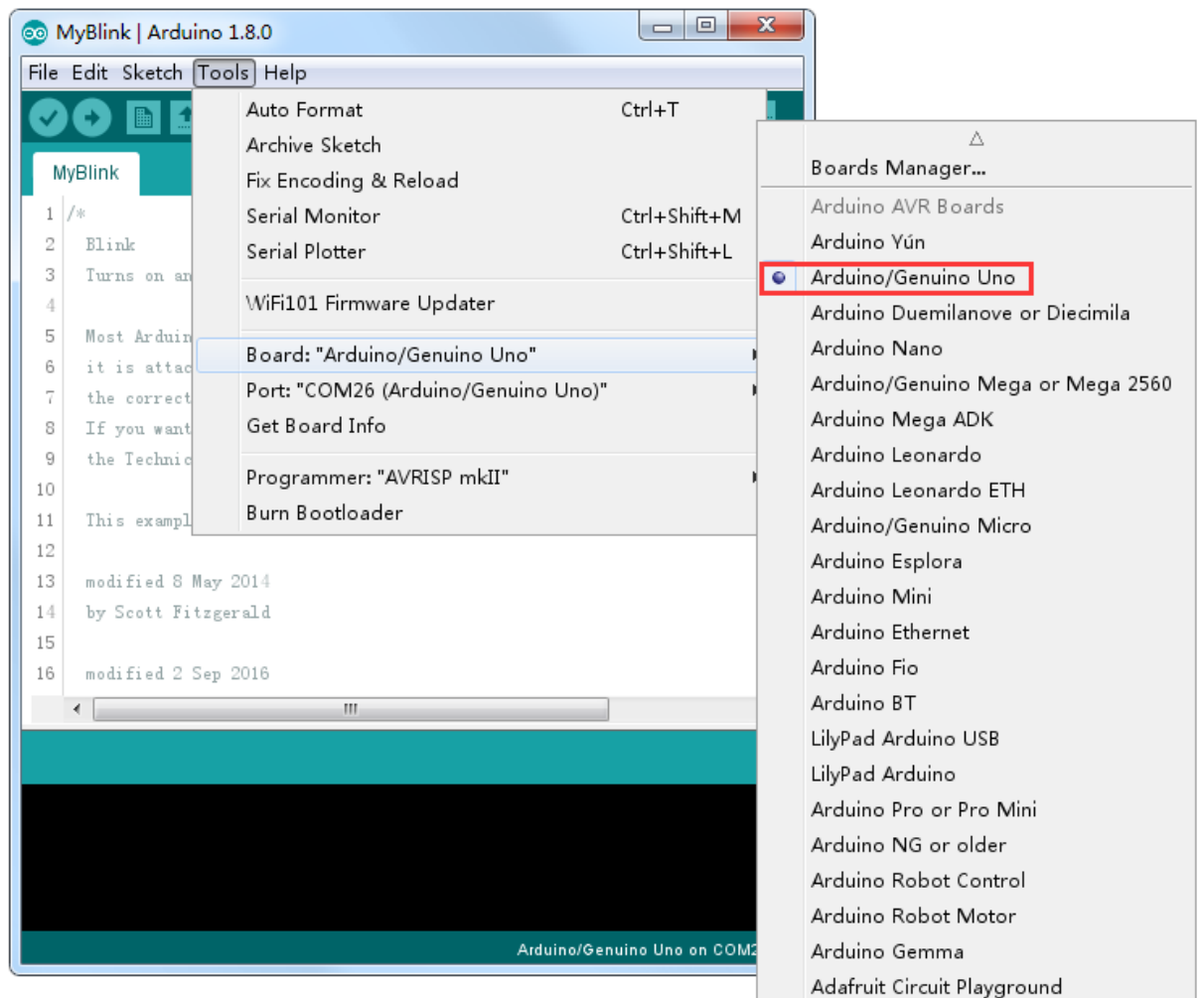




Ha guardado su copia de 'Blink' en su sketchbook. Esto significa que si alguna vez quiere encontrar otra vez, puede simplemente abrir usando el archivo > opción de menú de Sketchbook.



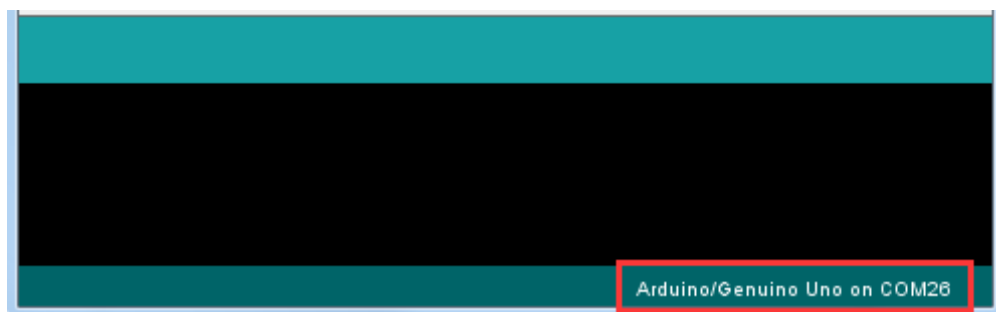
Conecte la placa de Arduino al ordenador con el cable USB y compruebe que la "Board Type" y "Puerto serie" están ajustados correctamente.



N

El tipo de tarjeta y puerto Serial aquí no son necesariamente la misma que se muestra en la imagen. Si usas 2560, entonces usted tendrá que elegir Mega 2560 como el tipo de Junta, otras opciones se pueden hacer de la misma manera. Y el puerto Serial para todo el mundo es diferente, a pesar de COM 26 elegido aquí, sería COM3 o COM4 en su ordenador. Un puerto COM correcto se supone que es COMX (arduino XXX), que es por los criterios de certificación.

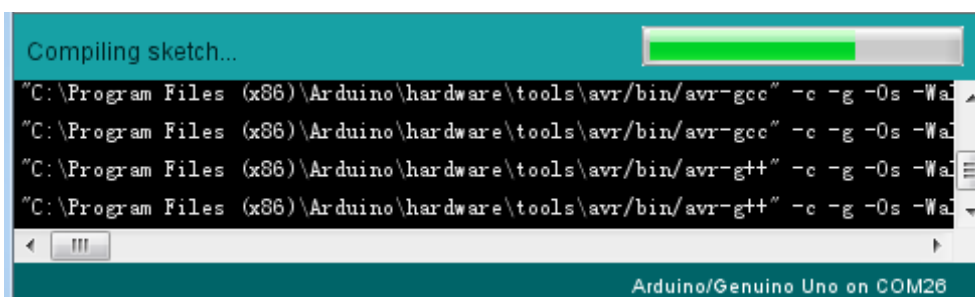
El IDE de Arduino mostrará la configuración actual de la Junta en la parte inferior de la ventana.



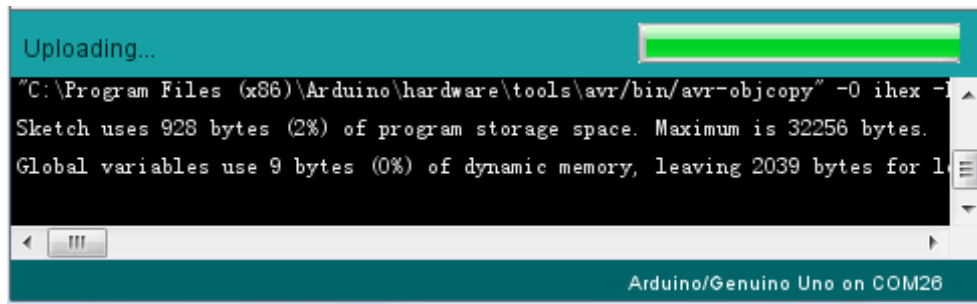
Haga clic en el botón 'Subir'. El segundo botón de la izquierda en la barra de herramientas.



Si usted mira el área de estado del IDE, verá una barra de progreso y una serie de mensajes. Al principio, que dice 'Bosquejo compilar...'. Esto convierte el dibujo en un formato adecuado para subir a la Junta.



A continuación, el estado cambiará a 'Subir'. En este punto, los LEDs de la Arduino deben comenzar a parpadear como se transfiere el dibujo.



Por último, el estado cambiará a 'Done'.



El otro mensaje nos dice que el bosquejo está utilizando 928 bytes de 32.256 bytes disponibles. Después de la etapa de compilación Sketch... podría obtener el siguiente mensaje de error:



Puede significar que su Junta no está conectado a todos, o no se ha instalado los drivers (si es necesario) o que se ha seleccionado el puerto serial incorrecto.

Si encuentras esto, volver a la lección 0 y verifique su instalación.

Una vez completada la carga, la Junta debe reiniciar y empiezan a parpadear.

El código abierto

Tenga en cuenta que una gran parte de este esquema se compone de comentarios. Estas no son instrucciones de programa real; por el contrario, sólo explican cómo funciona el programa. Están allí para subbeneficio.

Todo entre /\* y \* en la parte superior del bosquejo es un Comentario de bloque; explica lo

que el bosquejo es para.

Comentarios de una sola línea comienzan con `//` y todo hasta el final de esa línea se considera un comentario.

La primera línea de código es:

```
int led = 13;
```

Como explica el comentario sobre ella, esto es dar un nombre al pin que el LED está conectado a. Se trata de 13 en mayoría Arduinos, incluyendo la ONU y Leonardo.

A continuación, tenemos la función de 'configuración'. Otra vez, como dice el comentario, este se ejecuta cuando se presiona el botón de reset. También se ejecuta cada vez que la Junta se reinicia por alguna razón, como poder primero se aplica a él, o después de un bosquejo se ha subido

```
void setup() {  
  // Inicializa el pin digital como salida.  
  pinMode(led, OUTPUT);  
}
```

Cada sketch Arduino debe tener una función de 'configuración', y el lugar donde puede Agregar las instrucciones de su propio entre{y}.

En este caso, es un comando, que, como dice el comentario dice la placa Arduino que vamos a utilizar el pin LED como salida.

También es obligatorio para un boceto tener una función de 'lazo'. A diferencia de la función de 'setup' que se ejecuta sólo una vez, después de un reset, la función 'loop', después que haya terminado de ejecutar sus comandos, empezar inmediatamente otra vez.

```
void loop() {  
  digitalWrite(led, HIGH);    // Encienda el LED (alto es el nivel de voltaje)  
  delay(1000);                // Espere un segundo  
  digitalWrite(led, LOW);     // Apagar el LED por lo que la tensión baja  
  delay(1000);                // Espere un segundo  
}
```

Dentro de la función de bucle, los comandos en primer lugar activar el pin del LED (alto), girar a 'retraso' de 1000 milisegundos (1 segundo), entonces el pin LED apagado y pausa para otro segundo.

Ahora vas a que el LED parpadee más rápido. Como puede haber adivinado, la clave de esto

radica en cambiar el parámetro () para el comando 'retardo'.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the volt
33   delay(500) // wait for a second
34   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the vo
35   delay(500) // wait for a second
36 }
```

Este período de retardo en milisegundos, así que si desea que el LED parpadee dos veces tan rápidamente, cambiar el valor de 1000 a 500. Esto entonces pausa durante medio segundo cada retraso en lugar de un segundo entero.

Sube otra vez el bosquejo y verá el LED comienza a parpadear más rápidamente

## Lección 3 LED

### Resumen

En esta lección, usted aprenderá cómo cambiar el brillo de un LED usando diferentes valores de resistencia.

### Componente necesario:

(1) x Elegoo Uno R3

LED rojo de 5mm x (1)

(1) x resistencia de 220 ohmios

(1) x resistencia de 1 kohm

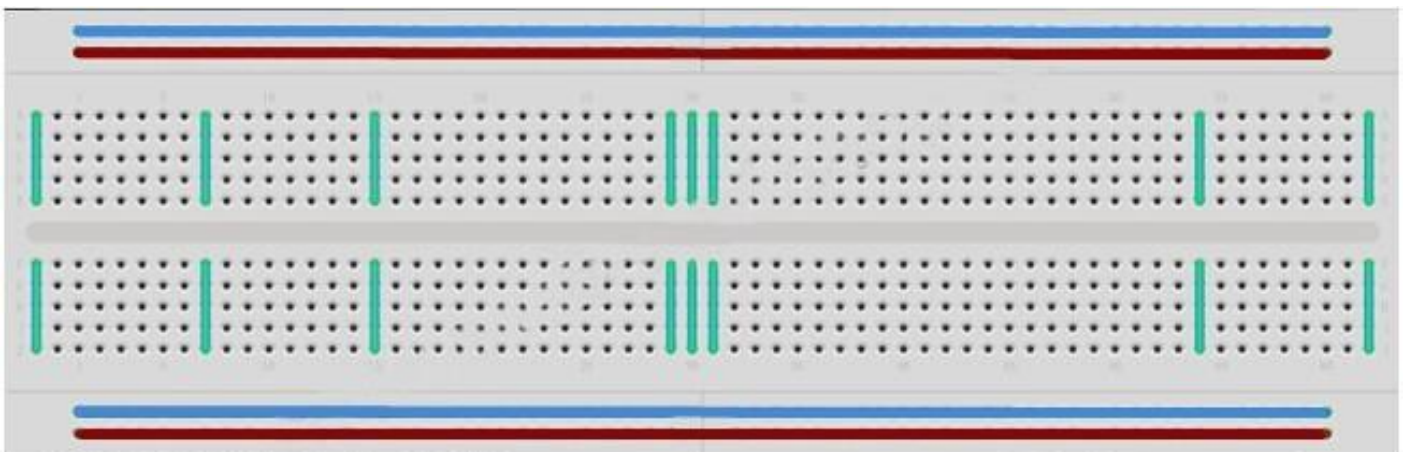
(1) x resistencia de ohmio 10 k

(2) x M M cables (cables de puente de macho a macho)

### Introducción del componente

#### PROTOBOARD MB-102 :

Un protoboard permite circuitos prototipo rápidamente, sin necesidad de soldar las conexiones. A continuación un ejemplo.



Breadboards vienen en varios tamaños y configuraciones. La clase más simple es sólo una rejilla de agujeros en un bloque de plástico. En el interior son tiras de metal que proporcionan una conexión eléctrica entre los agujeros en las filas más cortas. Empujando las patillas de dos componentes diferentes de la misma fila se une a ellos juntos eléctricamente. Un canal profundo, corriendo por el centro indica que hay una rotura en las conexiones, lo que significa, puede insertar un chip con las patillas a ambos lados del canal sin conectarlos juntos. Algunos breadboards tienen dos tiras de agujeros que corre a lo largo de los bordes laterales del tablero que son separados de la red principal. Éstos tienen tiras corriendo por la longitud de la tabla dentro y proporcionan una manera para conectar una tensión común. Son generalmente en pares para + 5 voltios y tierra. Estas tiras se denominan carriles y le permiten alimentar a muchos componentes o puntos en el tablero.

Si bien son breadboards para prototipos, tienen algunas limitaciones. Porque las conexiones son temporales y de acople, no son tan fiables como conexiones soldadas. Si tienes problemas intermitentes con un circuito, puede ser debido a una mala conexión en una protoboard.

### **LED:**

LED es un gran indicador. Utilizan muy poca electricidad durará para siempre.

En esta lección, usarás tal vez el más común de todos los LEDs: un LED de 5mm de color rojo. 5mm se refiere al diámetro del LED. Otros tamaños comunes son 3mm y 10mm.

Directamente no se puede conectar un LED a una batería o fuente de tensión porque 1) el LED tiene un positivo y un negativo llevar y no se encenderá si se coloca mal y debe utilizarse 2) un LED con una resistencia para limitar o 'ahogar' la cantidad de corriente que fluye a través de él; ¡de lo contrario, quemará



Si no utilizas una resistencia con un LED, entonces se puede quemar casi de inmediato, como demasiada corriente fluiría a través, calienta y destruye al 'cruce' donde se produce la luz.

Hay dos maneras de saber cuál es el positivo del LED y cuál la negativa.

En primer lugar, el positivo es más largo.

En segundo lugar, donde la pata del negativo entra en el cuerpo del LED, hay un borde plano para el caso del LED.

Si le sucede que tiene un LED que tiene un lado plano al lado del cable más largo, usted debe saber que la pata es el positivo.

### **RESISTENCIAS:**

Como su nombre lo indica, resistencias se resisten el flujo de electricidad. Cuanto mayor sea el valor de la resistencia, resiste más y la menos corriente fluiría a través de él. Vamos a usar esto para controlar cuánta electricidad fluye a través del LED y por lo tanto, como claramente brilla.

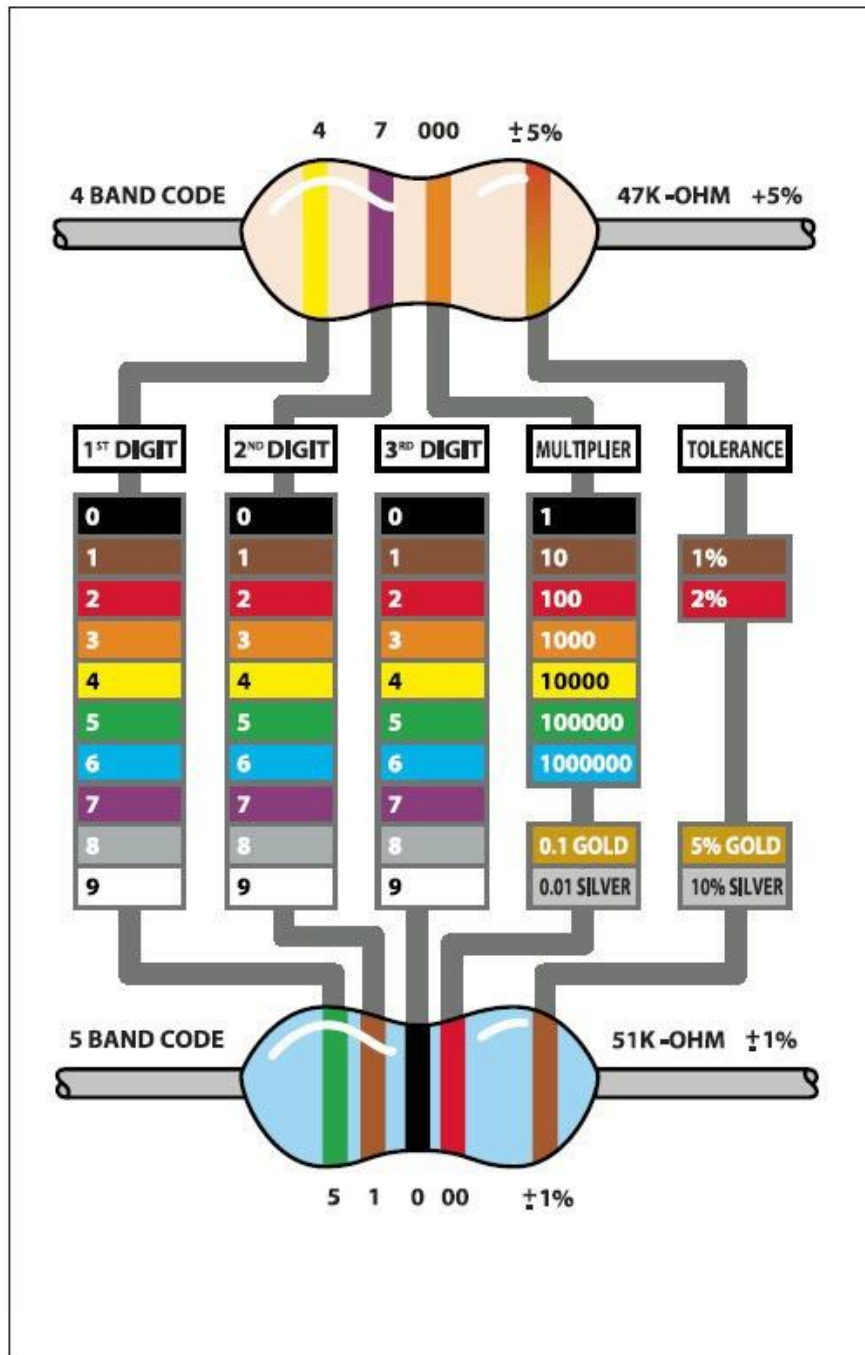


Pero primero, más sobre resistencias...

La unidad de resistencia se denomina Ohm, que se abrevia generalmente a  $\Omega$  la letra griega Omega. Porque un Ohm es un valor bajo de resistencia (no resiste mucho a todos), incluye los valores de resistencias en  $k\Omega$  (1.000  $\Omega$ ) y  $M\Omega$  (1.000.000  $\Omega$ ). Éstos se llaman kilo-ohms y mega-ohmios.

En esta lección, vamos a utilizar tres valores diferentes de resistencia: 220 $\Omega$ , 1k $\Omega$  y 10k $\Omega$ . Estas resistencias todas se ven iguales, excepto que tienen rayas de colores diferentes en ellos. Estas rayas decirte el valor de la resistencia.

El código de color resistor tiene tres franjas de colores y luego una banda de oro en un extremo.

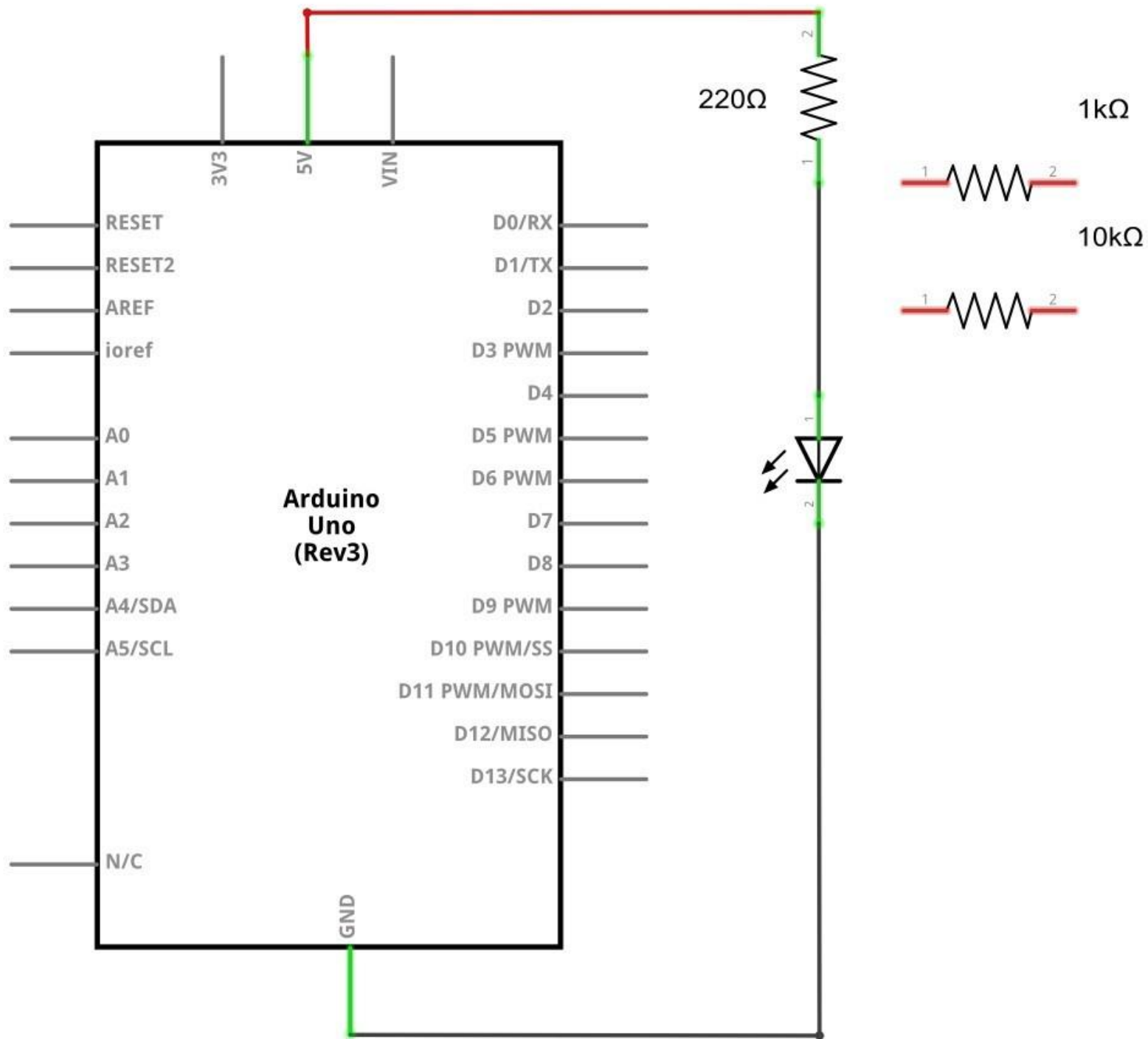


A diferencia de los LEDs, resistencias no tienen un cable positivo y negativo. Se puede conectar de cualquier manera alrededor.

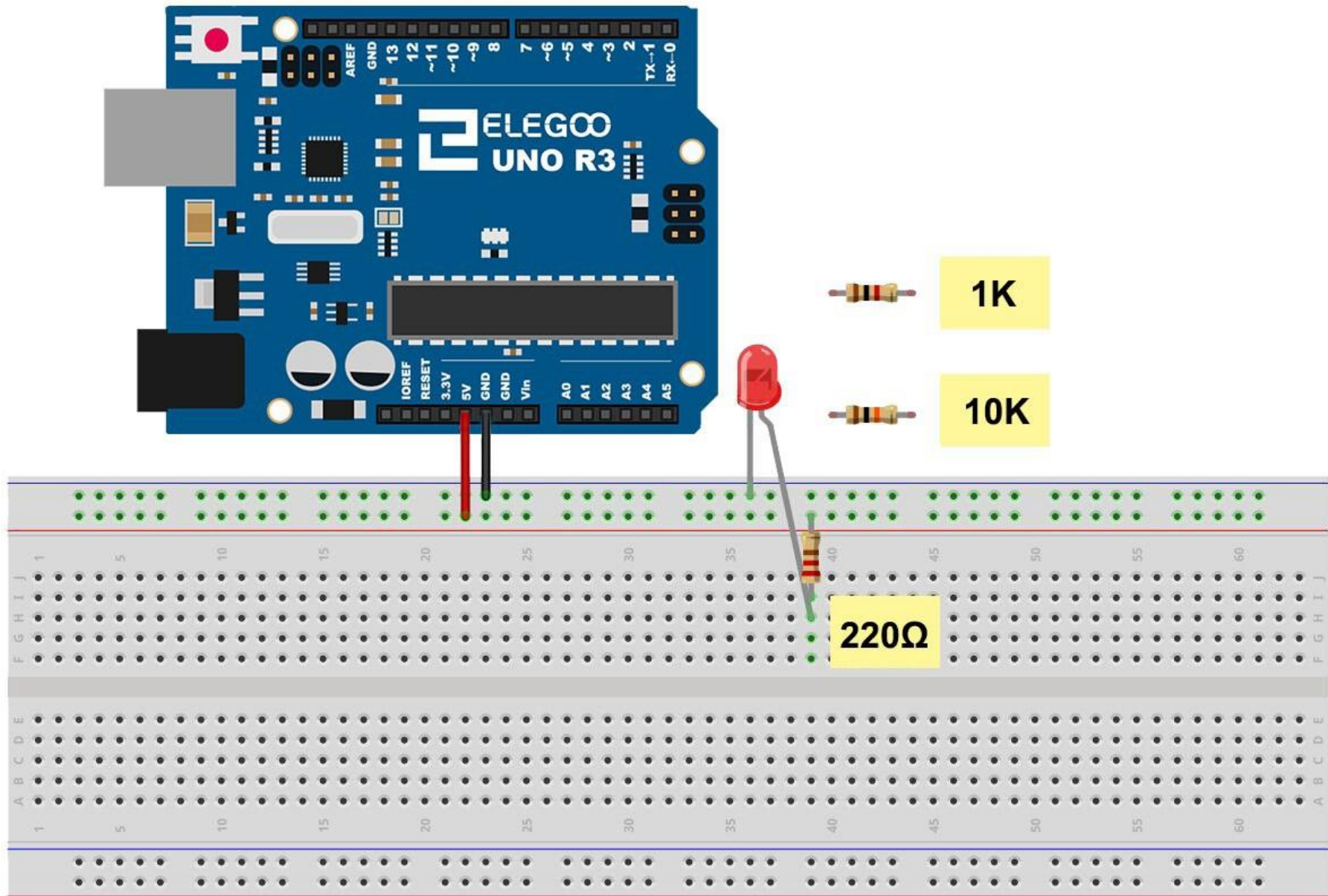
Si encuentra este método de enfoque demasiado complicada, puedes leer la bandera de anillo de color en nuestras resistencias directamente para determinar su valor de resistencia. O puede usar un multímetro digital en lugar de otro.

## Conexión

### Esquema



## Diagrama de Cableado



La UNO es una conveniente fuente de 5 voltios, que vamos a utilizar para alimentar el LED y la resistencia. No necesita hacer nada con su UNO, salvo que lo conecte un cable USB.

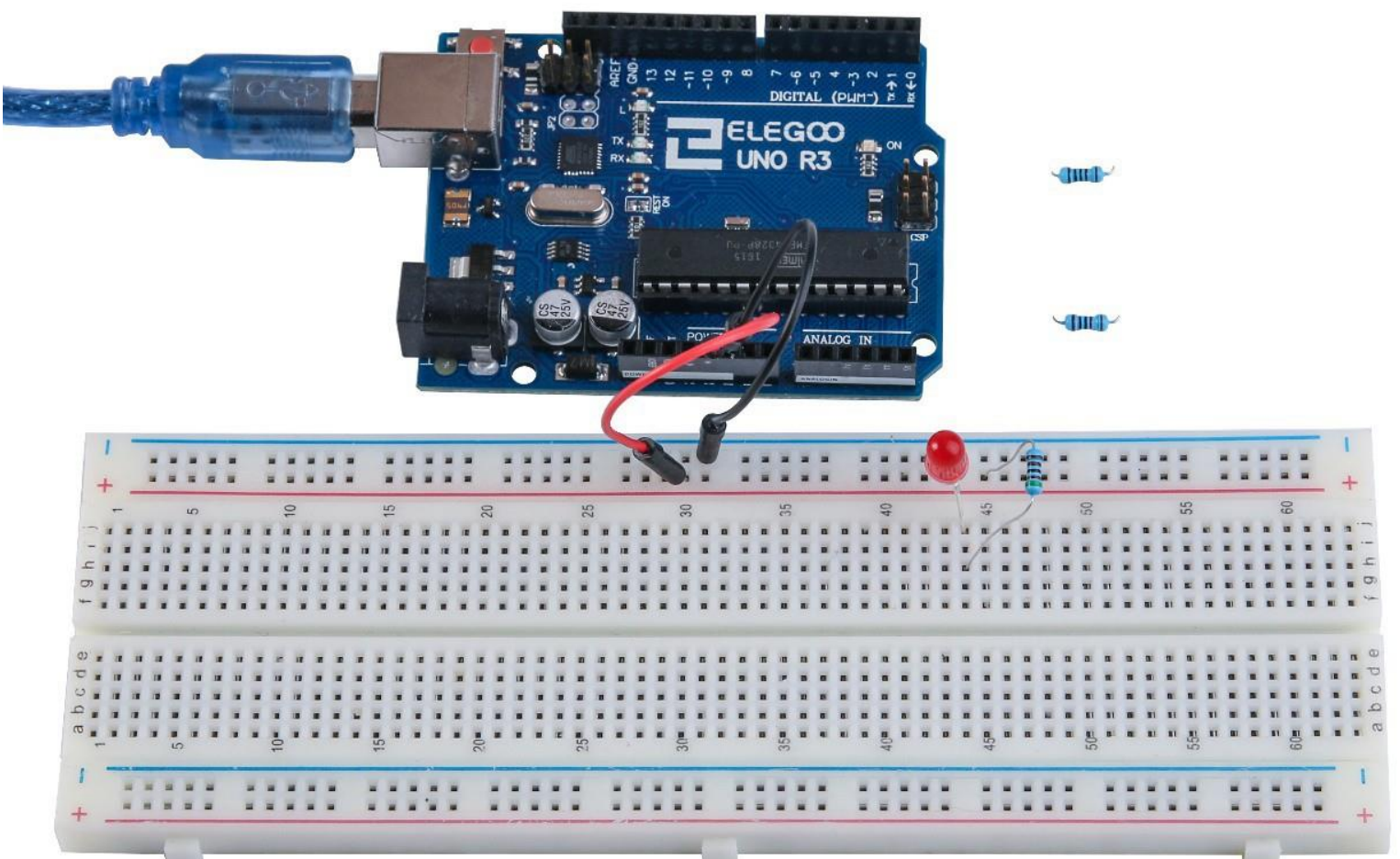
Con la resistencia de  $220\ \Omega$  en su lugar, el LED debe ser bastante brillante. Si cambia la resistencia  $220\ \Omega$  para la resistencia de  $1\text{k}\Omega$ , el LED aparecerá regulador un poco. Por último, con el resistor de  $10\text{k}\Omega$  en su lugar, el LED estará casi visible. Tire el cable de puente rojo de la placa y tocar en el agujero y eliminarlo, por lo que actúa como un interruptor. Sólo podrá notar la diferencia.

Por el momento, tienes 5V va a una pata de la resistencia, la otra pata de la resistencia va al lado positivo del LED y el otro lado del LED va a GND. Sin embargo, si nos mudamos la resistencia por lo que vino después el LED, como se muestra abajo, el LED seguirá la luz.

Probablemente desee volver a colocar el resistor  $220\Omega$ .

No importa qué lado del LED que ponemos la resistencia, siempre y cuando está allí en algún lugar

### Imagen de ejemplo



## Lección 4 RGB LED

### Resumen

RGB LED son una forma divertida y fácil para agregar color a sus proyectos. Puesto que es como regular 3 LED en uno, el uso y conexión no es muy diferente.

Vienen en 2 versiones: ánodo común o cátodo común.

Ánodo común utiliza 5V en el pin común, mientras que el cátodo común se conecta a tierra.

Como con cualquier LED, tenemos que conectar algunas resistencias en línea (3 total) así que podemos limitar la corriente absorbida.

En nuestro bosquejo, se comienzan con el LED en el estado de color rojo, entonces se descolora a verde, luego se descolora azul y finalmente hacia el color rojo. Haciendo esto que nos pasará por la mayor parte del color que se puede lograr.

### Componente necesario:

(1) x Elegoo Uno R3

(1) protoboard de 830 puntos de amarrex

M M de x (4) cables (cables de puente de macho a macho)

(1) x RGB LED

(3) resistencias de 220 ohmios x

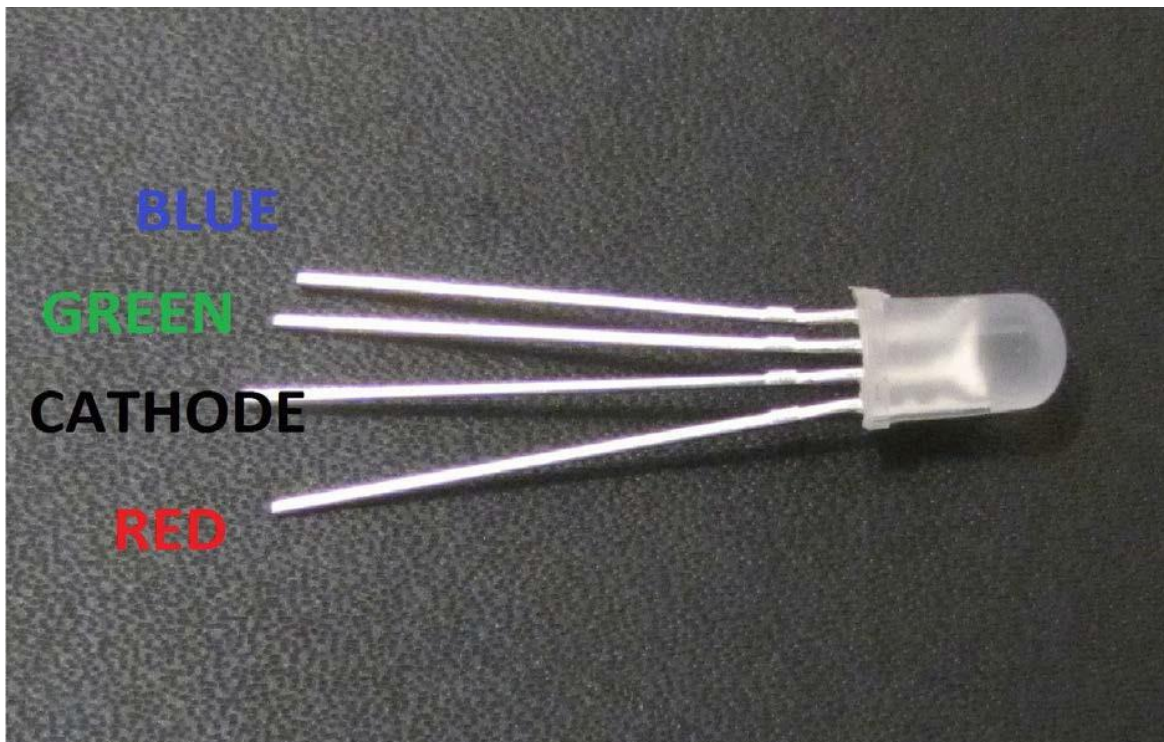
## Introducción del componente

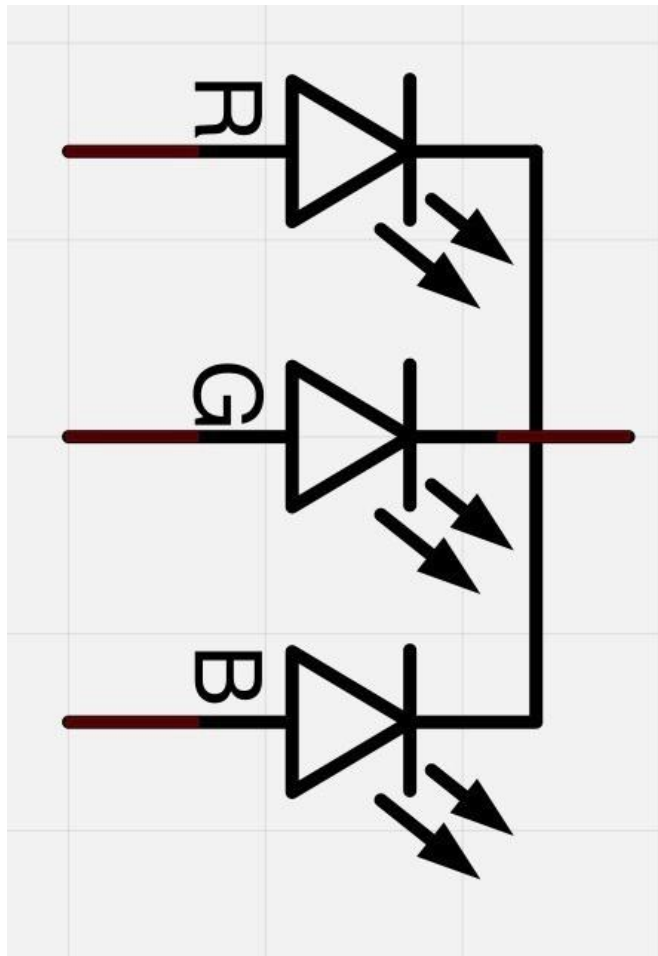
### RGB:

A primera vista, LEDs RGB (rojo, verde y azul) sólo parecen regular LED. Sin embargo, dentro del paquete del LED generalmente, hay realmente tres LEDs, uno rojo, uno verde y sí, uno azul. Controlando el brillo de cada uno de los LEDs individuales, usted puede mezclar prácticamente cualquier color quedesea.

Mezclamos colores del mismo modo que sería mezclar pintura en una paleta - ajustando el brillo de cada uno de los tres LEDs. La manera dura para hacer esto sería usar valor diferentes resistencias (o resistencias variables) como hicimos con el en la lección 2, pero eso es un mucho trabajo! Afortunadamente para nosotros, Kit UNO R3 tiene una función analogWrite que puede utilizar con pines marcados con un ~ a la salida de una cantidad variable de energía los LEDs apropiados.

El LED RGB tiene cuatro conductores. Hay un cable a la conexión positiva de cada uno de los LEDs individuales dentro del paquete y un patilla única que está conectado a los tres lados negativos de los LEDs.





En las fotografías le mostramos 4 electrodo LED. Cada perno separado de color verde o azul o de rojo se llama ánodo. Siempre conectará «+» a él. Cátodo va a "-" (tierra). Si se conecta otra forma diferente el LED no se encenderá.

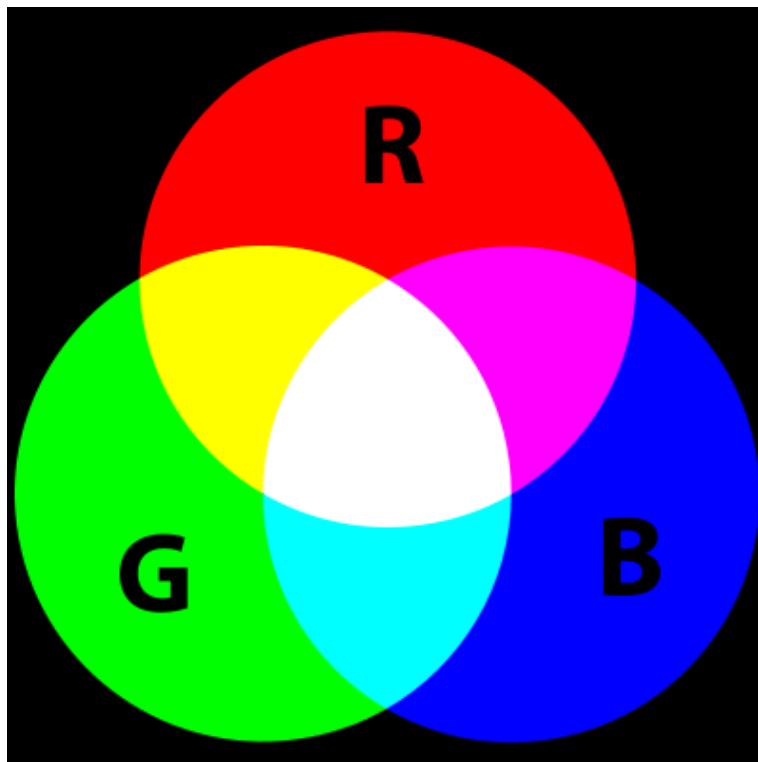
La común conexión negativa del paquete LED es el segundo pasador de la parte plana. También es el más largo de las cuatro patas y se conectarán a la tierra.

Cada LED dentro del paquete requiere su propio resistor de  $220\Omega$  para prevenir demasiada corriente que fluye a través de él. Los tres conductores del positivo de los LEDs (uno rojo, uno verde y uno azul) están conectados a los pines de salida UNO con estas resistencias.

### **Color:**

La razón por la que usted puede mezclar cualquier color que usted tiene gusto variando las cantidades de rojo, verde y azul de la luz es que el ojo tiene tres tipos de receptor de luz (rojo, verde y azul). Su ojo y el cerebro procesan las cantidades de rojo, verde y azul y convierten en un color del espectro.

En cierto modo, mediante el uso de los tres LEDs, estamos jugando un truco en el ojo. Esta misma idea se utiliza en televisores, donde la pantalla LCD tiene puntos de color rojo, verde y azul junto a unos a otros que componen cada píxel.



Si establece el brillo de todos los tres LEDs al ser el mismo, el color general de la luz será blanco. Si apagamos el LED azul, para que sólo los LEDs rojo y verdes son el mismo brillo, la luz aparecerá amarillo.

Que podemos controlar el brillo de cada una de las partes de rojas, verdes y azules del LED por separado, lo que es posible mezclar cualquier color que nos gusta.

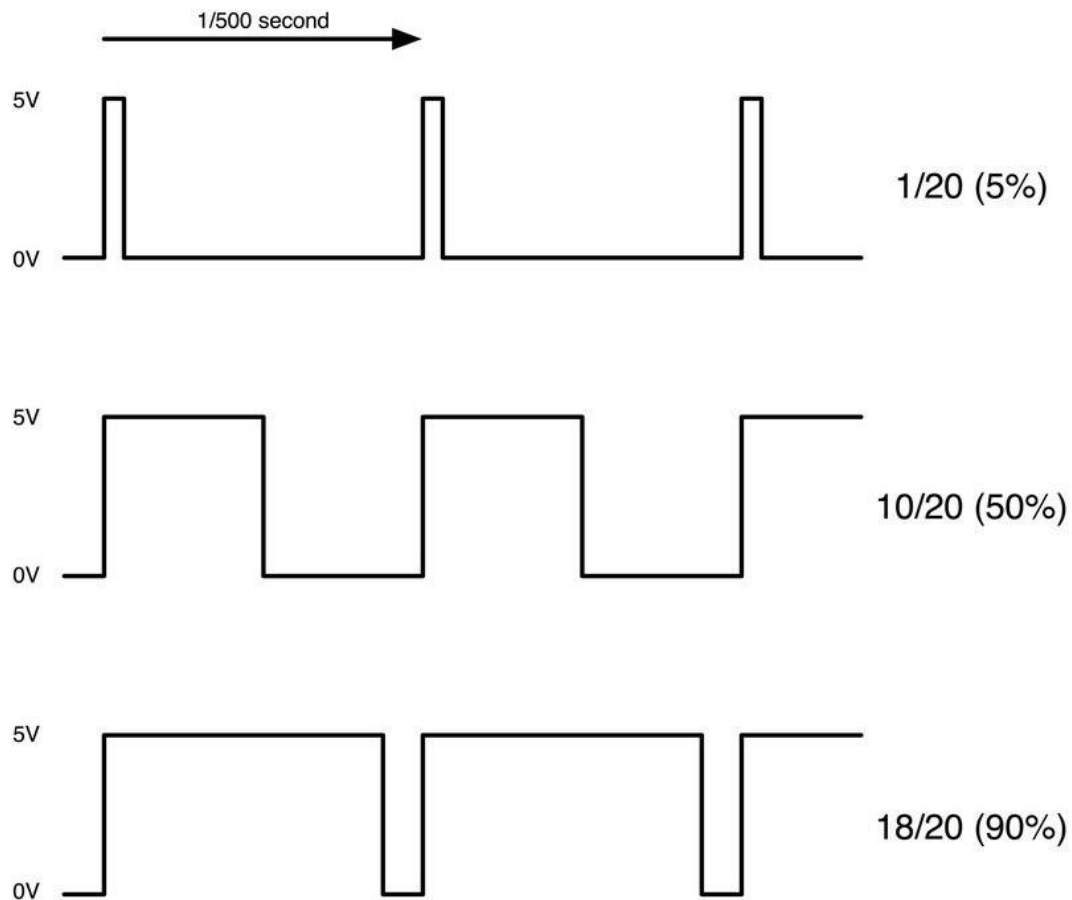
Negro no es tanto un color como una ausencia de luz. Por lo tanto, lo más cercano que podemos llegar a negro con el LED es apagar los tres colores.

## Teoría (PWM)

Modulación de ancho de pulso (PWM) es una técnica para el control de potencia.

También utilizamos aquí para controlar el brillo de cada uno de los LEDs.

El siguiente diagrama muestra que la señal de uno de los PWM pins en la UNO.



Aproximadamente cada 1/500 de segundo, la salida PWM producirá un pulso. La duración de este pulso es controlada por la función 'analogWrite'. Así 'analogWrite(0)' no producirá ningún pulso en todo 'analogWrite(255)' producirá un pulso que dura todo el camino hasta el pulso siguiente vencimiento, para que la salida es en realidad todo el tiempo.

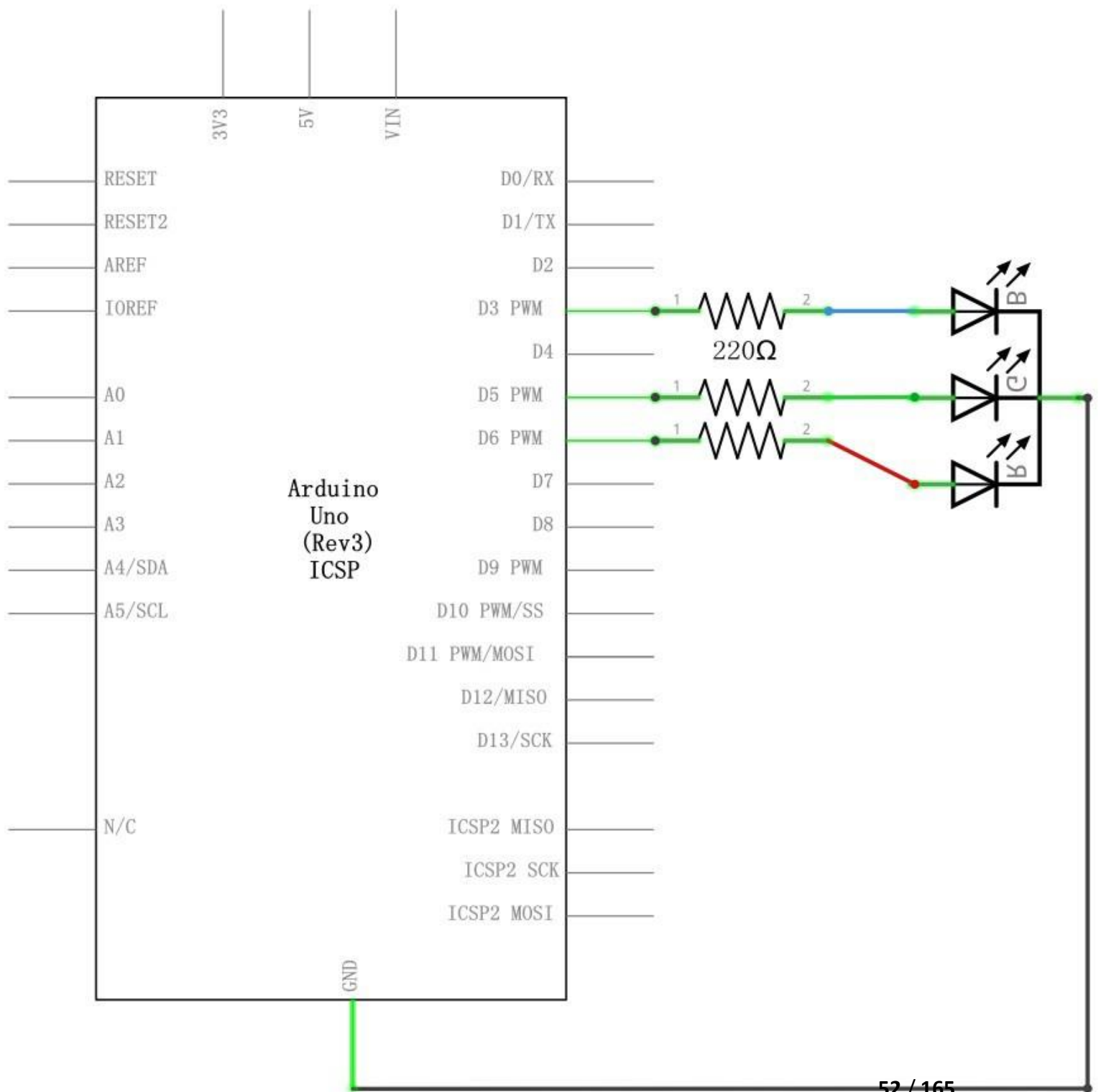
Si especificamos un valor en el analogWrite que está en algún lugar entre 0 y 255, se producirá un pulso. Si el pulso de salida es alto para el 5% del tiempo, entonces lo que nosotros estamos manejando sólo recibirá el 5% de potencia.

Si, sin embargo, la salida es 5V para el 90% del tiempo, la carga recibirá el 90% de la

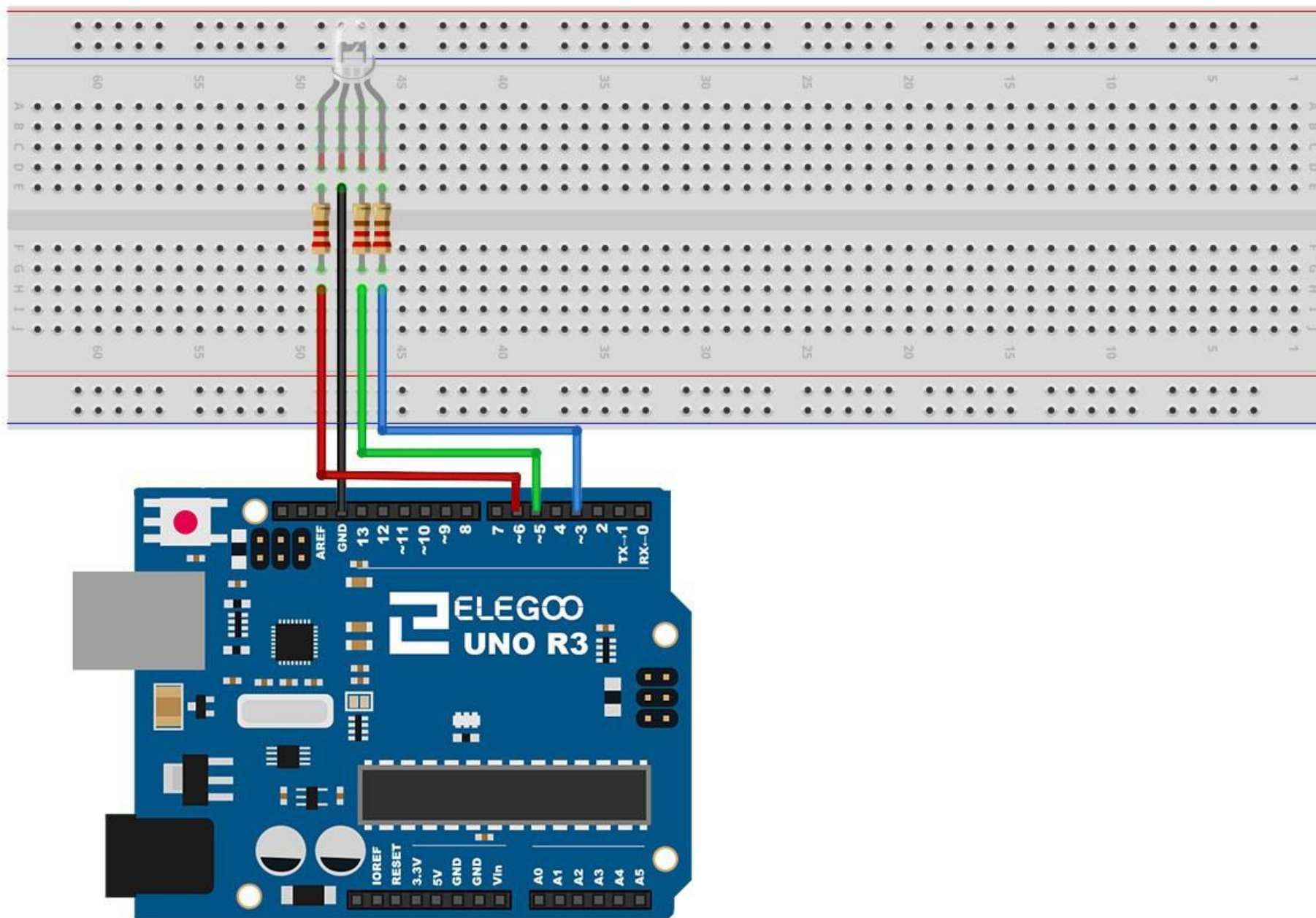
potencia entregada a él. No podemos ver los LEDs de encendido y apagado a esa velocidad, así que, sólo parece que está cambiando el brillo.

## Conexión

### Esquema



## Diagrama de cableado



## Código

Después de efectuar el cableado, abrir el programa en la carpeta-Lección 4 código RGB LED y haga clic en cargar para cargar el programa. Ver Lección 2 para obtener más información sobre programa cargar si hay algún error.

Nuestro código utilizará para bucles para recorrer los colores.

El primer bucle FOR pasará de rojo a verde.

El segundo bucle irá de verde a azul.

El último bucle irá desde el azul al rojo.

Probar el boceto y luego a diseccionarla endetalle...

El dibujo comienza especificando que los pines van a utilizar para cada uno de los colores:

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6
```

El siguiente paso es escribir la función 'setup'. Como hemos aprendido en lecciones anteriores, la función de configuración se ejecuta una sola vez después de que el Arduino se ha restablecido. En este caso, todo lo que tiene que hacer es definir los tres pines que estamos utilizando como salidas.

```
void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
}
```

Antes de echar un vistazo a la función 'loop', veamos la última función en el proyecto.

Las variables de definición

```
redValue = 255; // choose a value between 1 and 255 to change the color.
```

```
greenValue = 0;
```

```
blueValue = 0;
```

Esta función tiene tres argumentos, uno para el brillo de los LEDs rojos, verdes y azules. En cada caso de que el número será en el rango 0 a 255, donde 0 significa apagado y 255 significa brillo máximo. La función entonces llama 'analogWrite' para ajustar el brillo de cada LED.

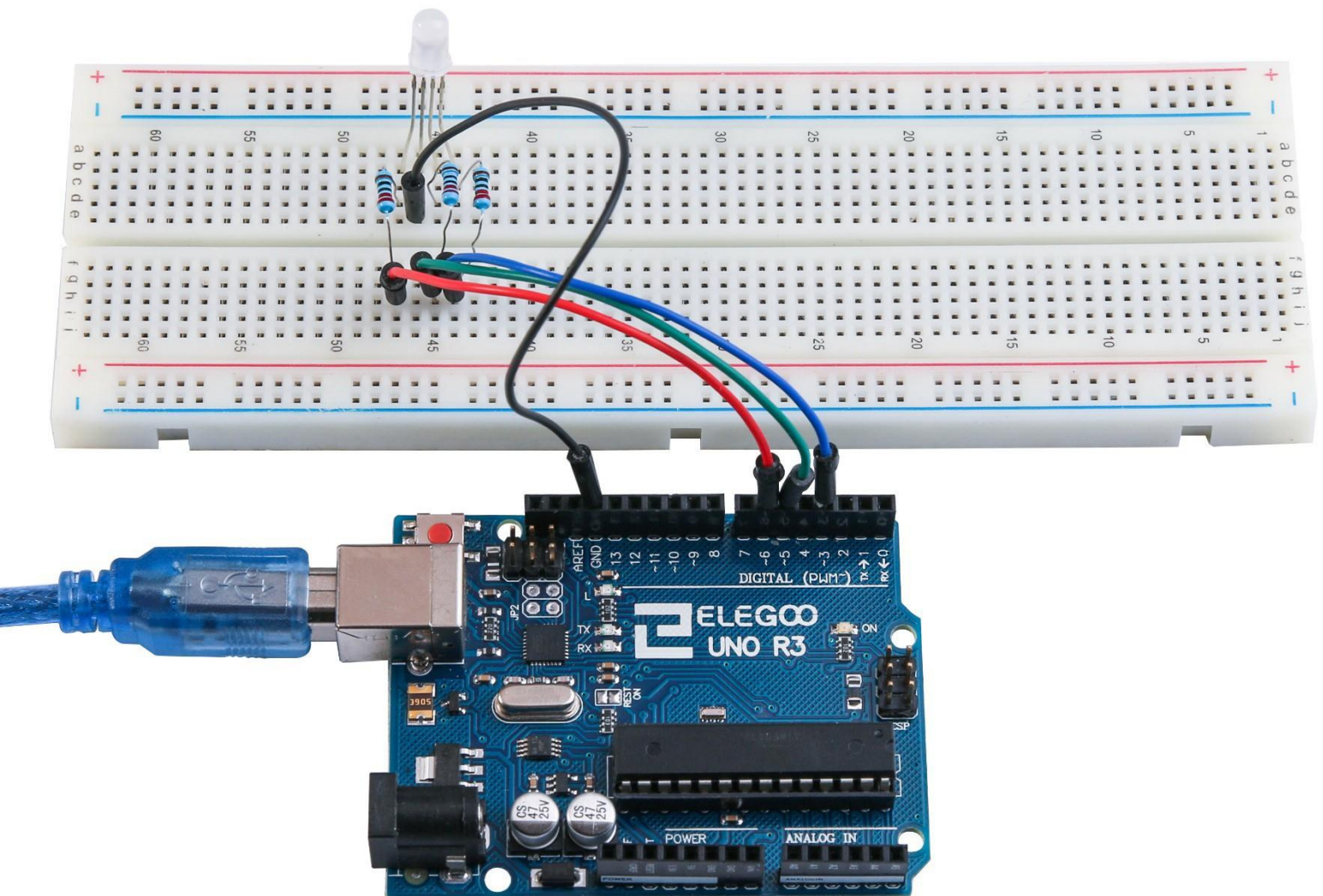
Si nos fijamos en la función 'loop' se puede ver que ajuste la cantidad de luz roja, verde y azul que queremos mostrar y luego una pausa por un segundo antes de pasar al siguiente color.

```
#define delayTime 10 // fading time between colors
```

```
Delay(delayTime);
```

Trate de añadir algunos colores de los tuyos el dibujo y ver el efecto en tu LED.

## Imagen de ejemplo



## Lección 5 Entradas Digitales

### Resumen

En esta lección, usted aprenderá a utilizar los botones con entradas digitales para encender y apagar un LED.

Presionar el botón se encenderá el LED; pulsar el otro botón se apagará el LED.

### Componente necesario:

(1) x Elegoo Uno R3

(1) protoboard de 830 puntos de amarrex

LED rojo de 5mm x (1)

(1) x resistencia de 220 ohmios

interruptores de presión x(2)

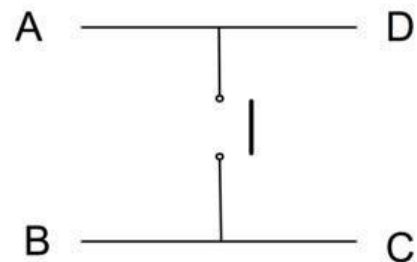
(7) x M M cables (cables de puente de macho a macho)

### Introducción del componente

#### INTERRUPTORES DE EMPUJE:

Los interruptores son componentes muy simples. Cuando pulse un botón o girar una palanca, conectan dos contactos para que la electricidad fluya a través de ellos.

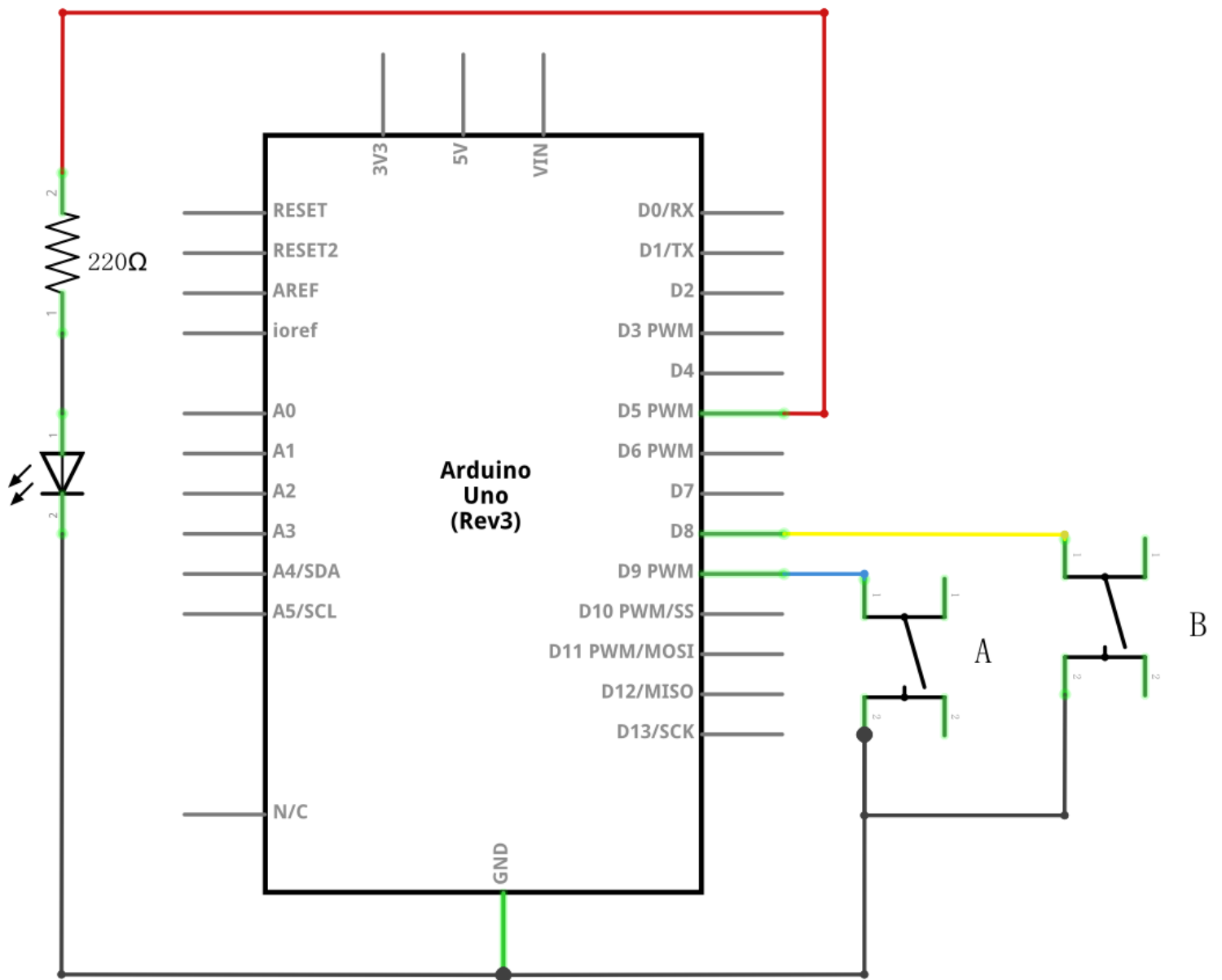
Los interruptores táctiles poco utilizados en esta lección tienen cuatro conexiones, que pueden ser un poco confusas



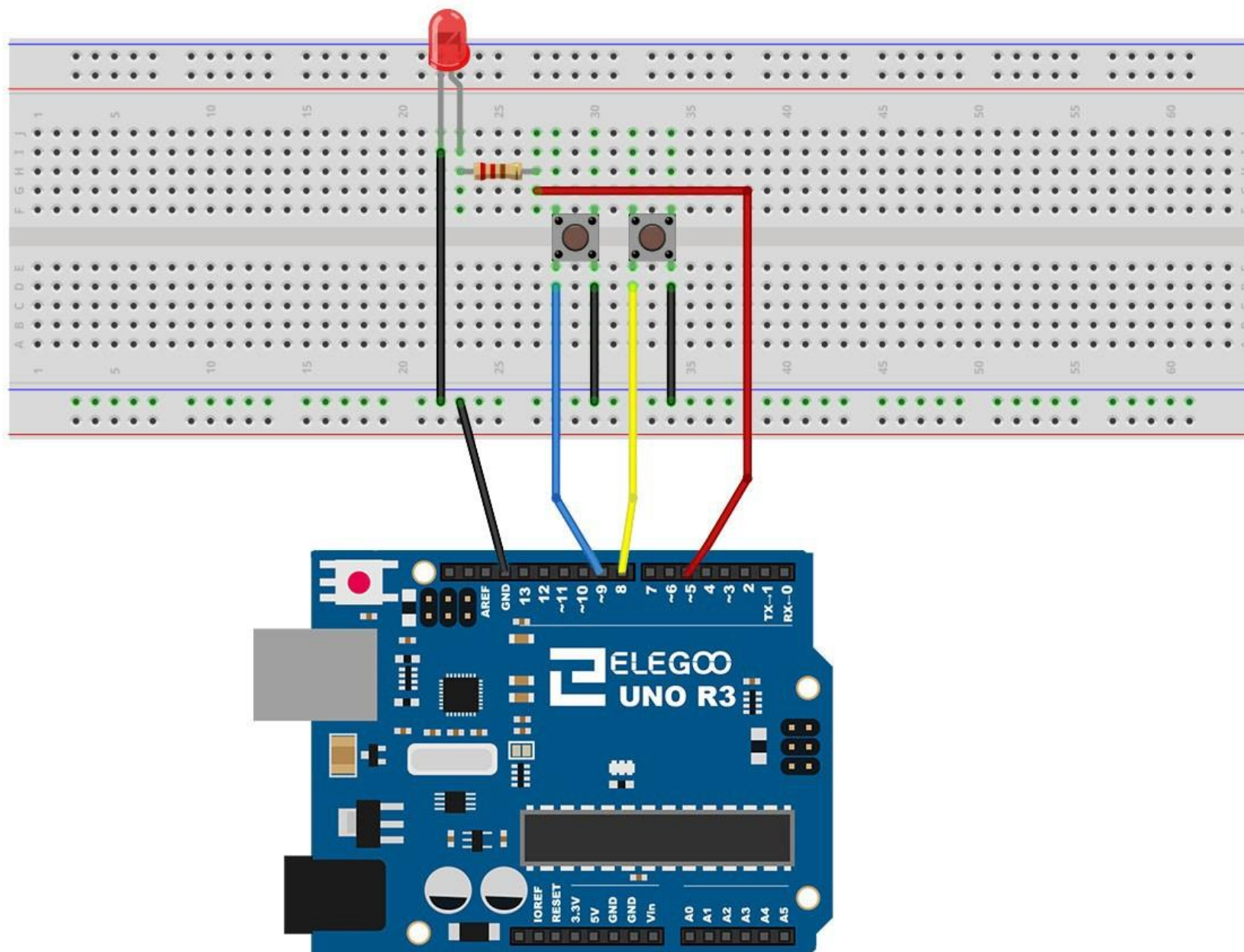
En realidad, hay realmente dos conexiones eléctricas. Dentro del paquete de interruptor, pins B y C se conectan entre sí, como son A y D.

## Conexión

### Esquema



## Diagrama de cableado



Aunque los cuerpos de los interruptores son cuadrados, los pasadores sobresalen de los lados opuestos del interruptor. Esto significa que los pines sólo estarán lo suficientemente separados cuando se colocan correctamente en la placa de pruebas. Recuerde que el LED tiene que tener el cable negativo más corto a la izquierda.

## Código

Después de cablearlo, por favor, abra el programa en el código de carpeta - Lección 5 entradas digitales y presione UPLOAD para cargar el programa. Si se aparecen errores, ver Lección 2 para obtener más información sobre el tutorial de subir programa.

El bosquejo en su placa UNO de carga. Presionando el botón izquierdo se encenderá el LED mientras que pulsando el botón derecho apagará.

La primera parte del proyecto define tres variables para las tres patas que se van a utilizar. El 'ledPin' es el pin de salida y 'buttonApin' se refiere al interruptor más cerca de la parte superior de la placa y 'buttonBpin' para el otro interruptor.

La función de 'configuración' define el ledPin como una salida normal, pero ahora tenemos las dos entradas para ocuparse. En este caso, utilizamos el conjunto el pinMode ser 'INPUT\_PULLUP' como este:

```
pinMode(buttonApin, INPUT_PULLUP);  
pinMode(buttonBpin, INPUT_PULLUP);
```

El modo pin de INPUT\_PULLUP significa que el pin debe ser utilizado como una entrada, pero que si nada mas se conecta a la entrada, se debe 'sacarse' a alta. En otras palabras, el valor predeterminado de la entrada es alta, a menos que se tiró bajo por la acción de pulsar el botón.

Por esta razón los interruptores están conectados a tierra. Cuando un interruptor se presiona, se conecta la clavija de entrada a la tierra, para que ya no es alta.

Puesto que la entrada es normalmente alta y va sólo baja cuando se pulsa el botón, la lógica es un poco boca abajo. Nosotros nos encargaremos de esto en la función 'loop'

```
void loop()  
{  
  if (digitalRead(buttonApin) == LOW)  
  {
```

```

    digitalWrite(ledPin, HIGH);
}
if (digitalRead(buttonBpin) == LOW)
{
    digitalWrite(ledPin, LOW);
}
}

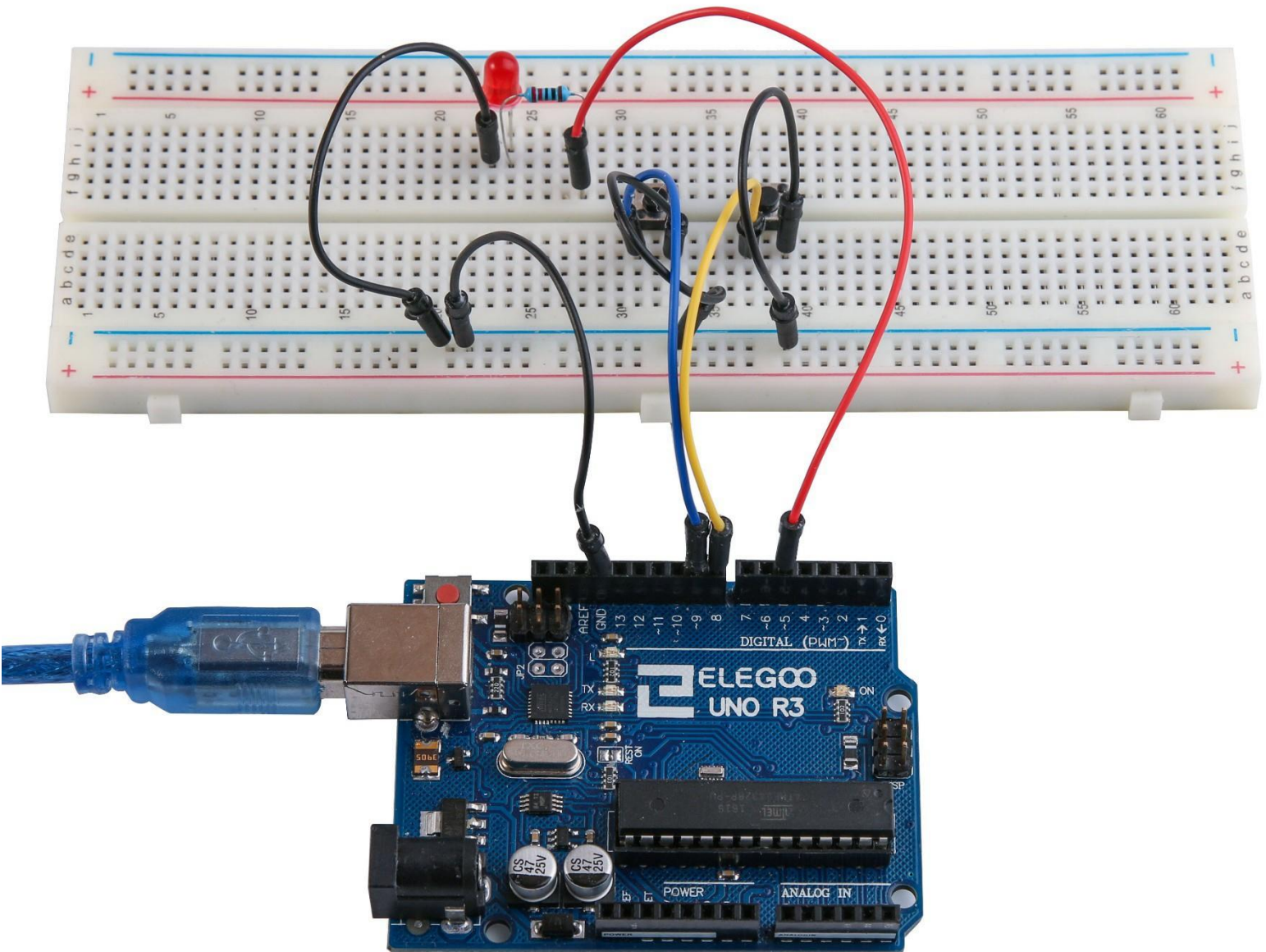
```

En la función 'loop' hay dos declaraciones de 'si'. Uno para cada botón. Cada uno hace un 'digitalRead' en la entrada adecuada.

Recuerde que si se presiona el botón, la entrada correspondiente será baja, si el botón A es bajo, entonces un 'digitalWrite' en el ledPin enciende.

Del mismo modo, si se presiona el botón B, un bajo se escribe en el ledPin.

## Imagen de ejemplo



## Lección 6 Activar zumbador

### Resumen

En esta lección, aprenderá a generar un sonido con un timbre activo.

### Componente necesario:

- (1) x Elegoo Uno R3
- (1) zumbador de Active x
- (2) x F M cables (cables de hembra a macho DuPont)

### Introducción del componente

#### ZUMBADOR:

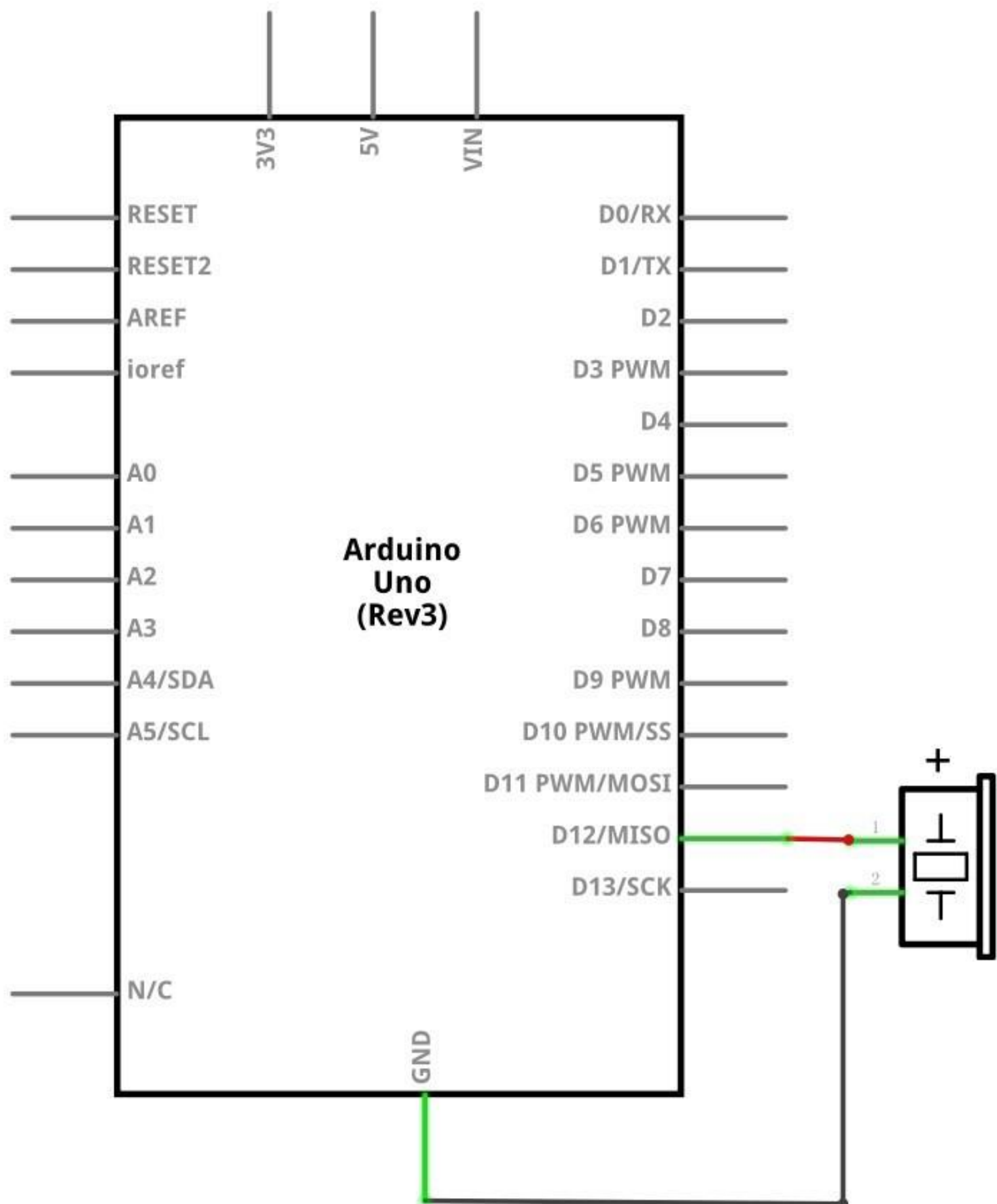
Zumbadores electrónicos son alimentados por CC, equipado con un circuito integrado. Son ampliamente utilizados en computadoras, impresoras, fotocopadoras, alarmas, juguetes electrónicos, dispositivos electrónicos automotores, teléfonos, alarmas y otros productos electrónicos para dispositivos de voz. ZUMBADORES se pueden categorizar como activa y pasiva. Gire que los pernos de dos zumbadores boca arriba. Con un tablero de circuito verde es un zumbador pasivo, mientras que el otro cerrado con una cinta negra es un activo.

La diferencia entre los dos es que un buzzer activo tiene una fuente oscilante integrada, por lo que va a generar un sonido cuando se electrifica. Una alarma pasiva no tiene esa fuente para que no pitan si se utilizan señales de DC; en cambio necesitará usar ondas cuadradas cuya frecuencia es entre 2K y 5K manejarlo. El zumbador activo es a menudo más car os que e l pasivo de bido am últiple sc ic uitos oscilantes incorporados.

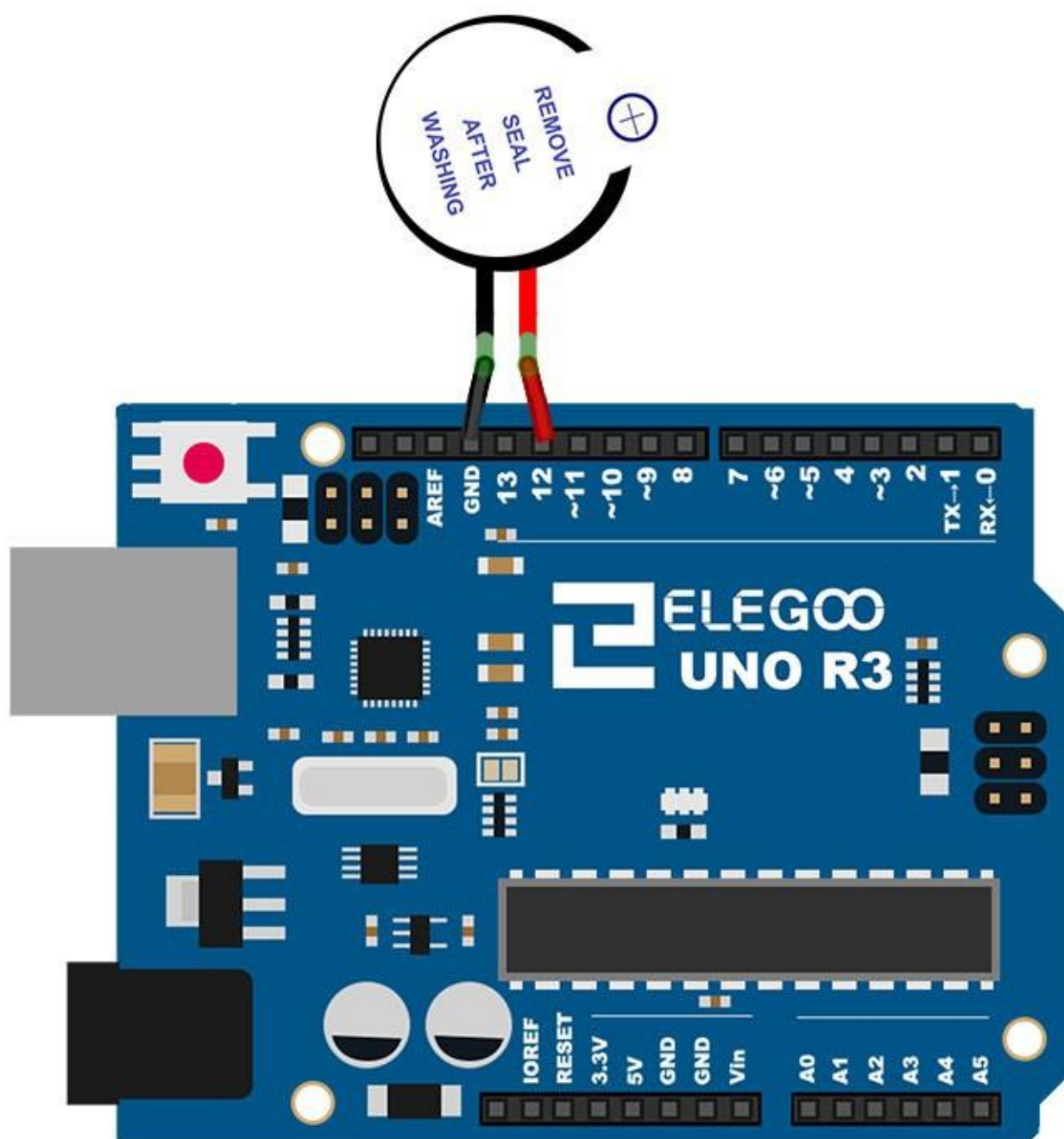


## Conexión

### Esquema



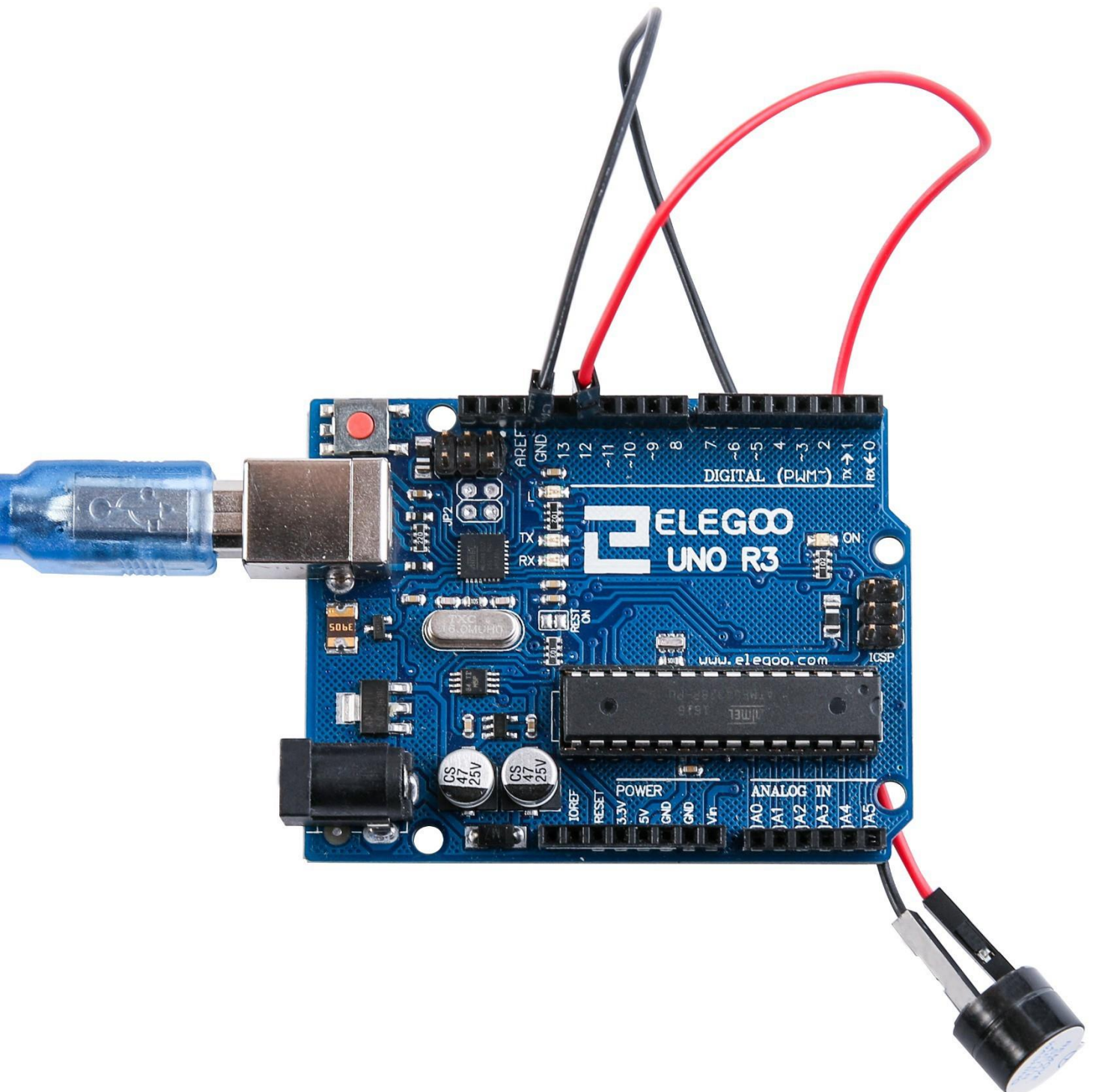
## Diagrama de conexiones



## Código

Después de efectuar el cableado, abrir el programa en el código carpeta - Lección 6 haciendo sonidos y haga clic en cargar para cargar el programa. Ver Lección 2 para obtener más información sobre programa cargar si hay algún error.

## Imagen de ejemplo



## Lección 7 Interruptor de bola de inclinación

### Resumen

En esta lección, usted aprenderá cómo utilizar un interruptor de bola de inclinación para detectar el pequeño ángulo de inclinación.

### Componentes Requeridos:

- (1) x Elegoo Uno R3
- (1) x interruptor de inclinación bola
- (2) x F-M wires (cables de hembra a macho DuPont)



### Introducción del componente

#### Sensor de inclinación

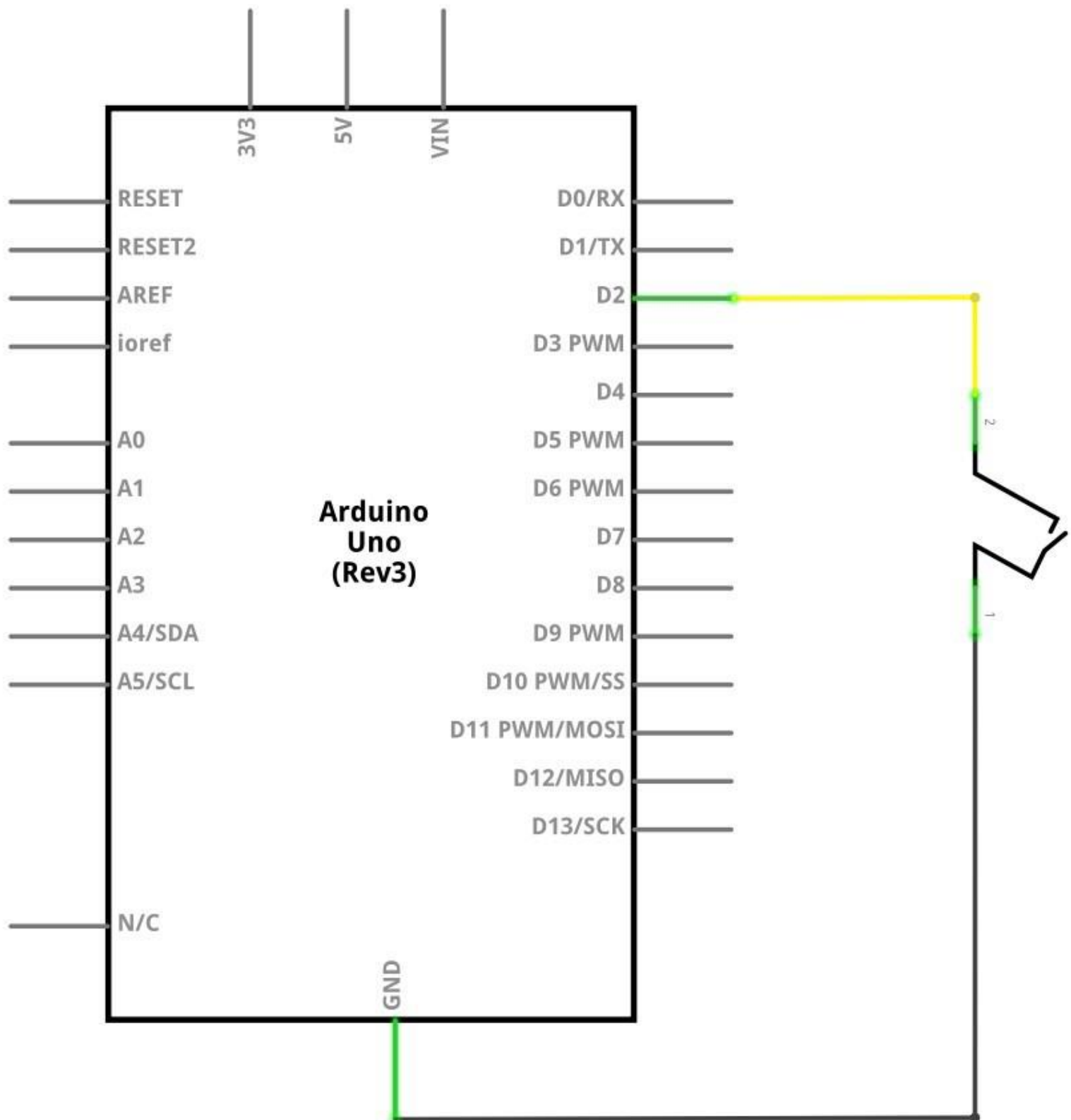
Los sensores de inclinación (interruptor de bola de inclinación) le permiten detectar orientación o inclinación. Son pequeños, económicos, de bajo consumo y fáciles de usar. Si se usan correctamente, no se desgastarán. Su simplicidad los hace populares para los juguetes, los adminículos y los aparatos. A veces, se conocen como "interruptores de mercurio", "interruptores de inclinación" o "sensores de bola rodante" por razones obvias.

Se componen generalmente de una cavidad de una cierta clase (cilíndrica es popular, aunque no siempre) con una masa libre conductora adentro, tal como una gota del mercurio o bola rodante. Un extremo de la cavidad tiene dos elementos conductores (polos). Cuando el sensor está orientado de tal manera que dicho extremo está hacia abajo, la masa rueda sobre los polos y los cortocircuitos, actuando como un interruptor de tiro.

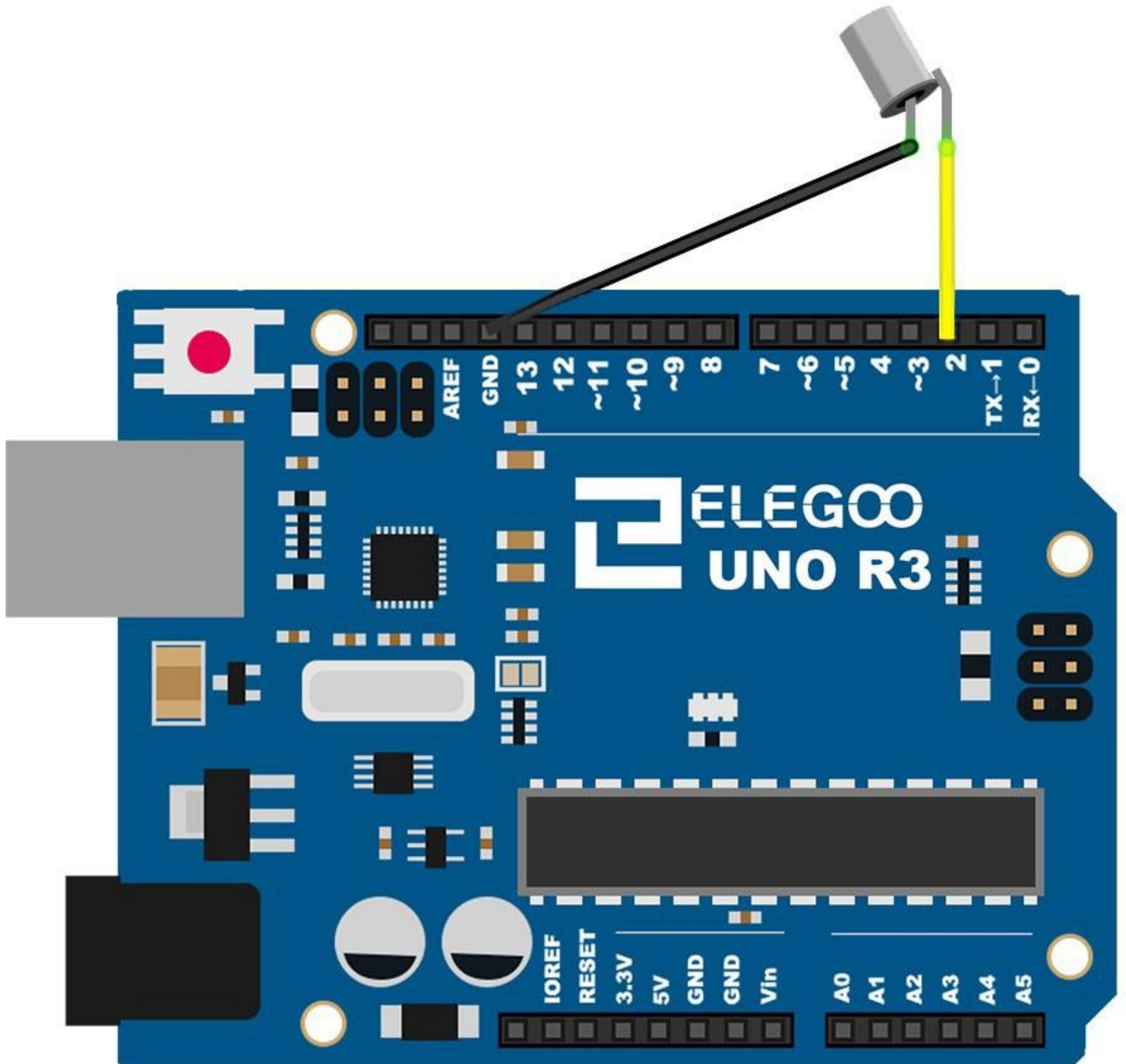
Aunque no es tan preciso ni flexible como un acelerómetro completo, los interruptores de inclinación pueden detectar movimiento u orientación. Otro beneficio es que los grandes pueden cambiar de energía por su cuenta. Los acelerómetros, por otro lado, producen voltaje digital o analógico que luego deben analizarse utilizando circuitos extra.

## Conexión

### Esquema



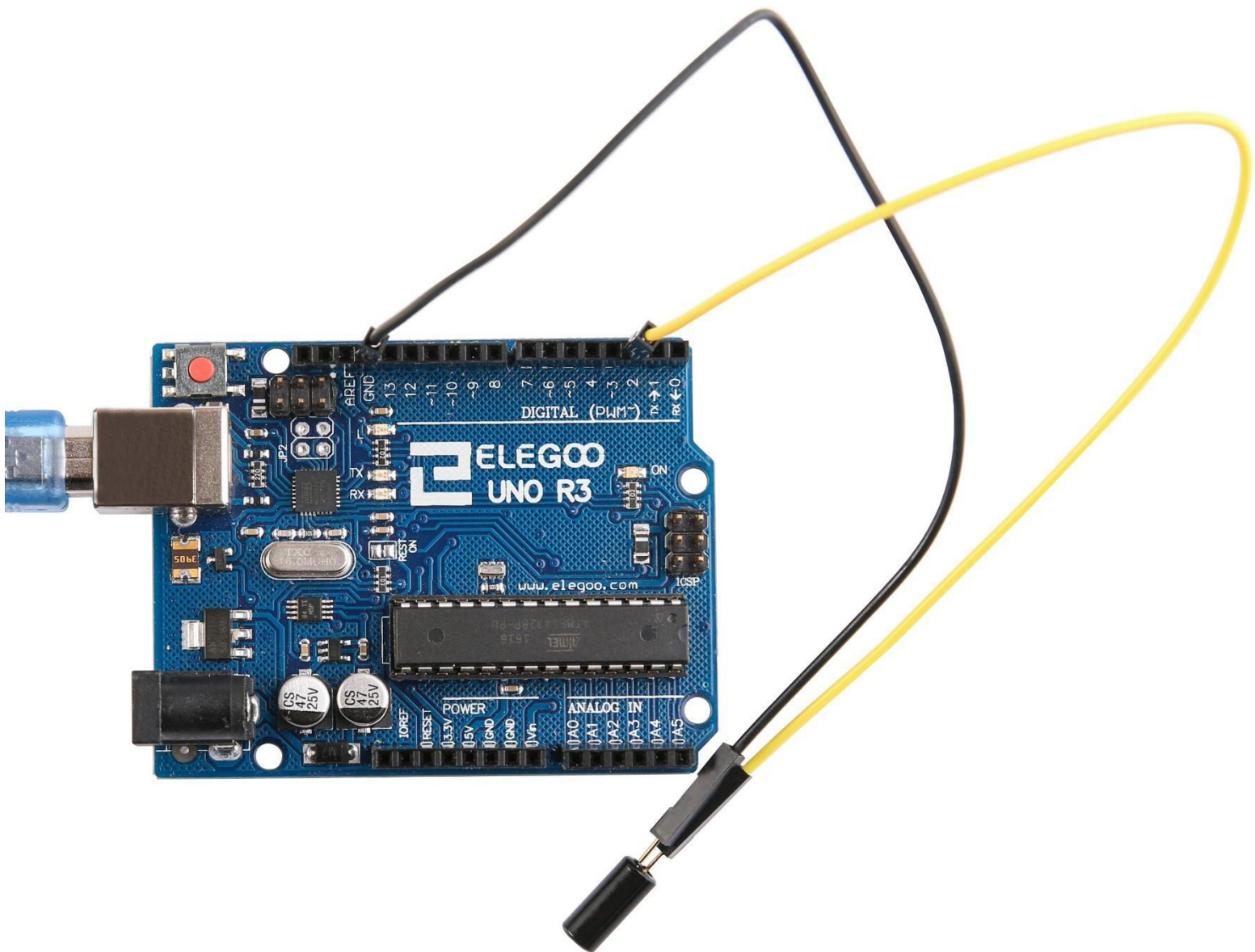
## Diagrama de conexión



## Código

Después de efectuar el cableado, por favor, abra el programa en el código de carpeta - lección 8 interruptor de la bola y haga clic en UPLOAD para cargar el programa. Ver Lección 2 para obtener más información sobre programa cargar si hay algún error.

## Imagen de ejemplo



## Lección 8 Ocho LED con 74HC595

### Resumen

En esta lección, usted aprenderá cómo utilizar ocho LEDs rojo grandes con un UNO sin necesidad de renunciar a 8 patas de salida

Aunque usted podría conectar ocho LEDs con una resistencia a un pin UNO rápidamente empezaría a quedarse sin patas en su UNO. Si no tienes un montón de cosas conectadas a la ONU. Está bien hacerlo - pero a menudo tiempos queremos botones, sensores, servos, etc. y antes de saberlo que no tienes pernos de izquierda. Así, en lugar de hacer eso, vas a usar un chip llamado el 74HC595 Serial a paralelo convertidor. Este chip tiene ocho salidas (perfectos) y tres entradas que utilizas para alimentar datos en él un poco a la vez.

Este chip hace un poco más lento para los LEDs (sólo se puede cambiar el LED unos 500.000 veces por segundo en lugar de 8.000.000 por segundo) pero todavía es muy rápido, forma más rápido que los seres humanos puede detectar, así que vale!

### Componente necesario:

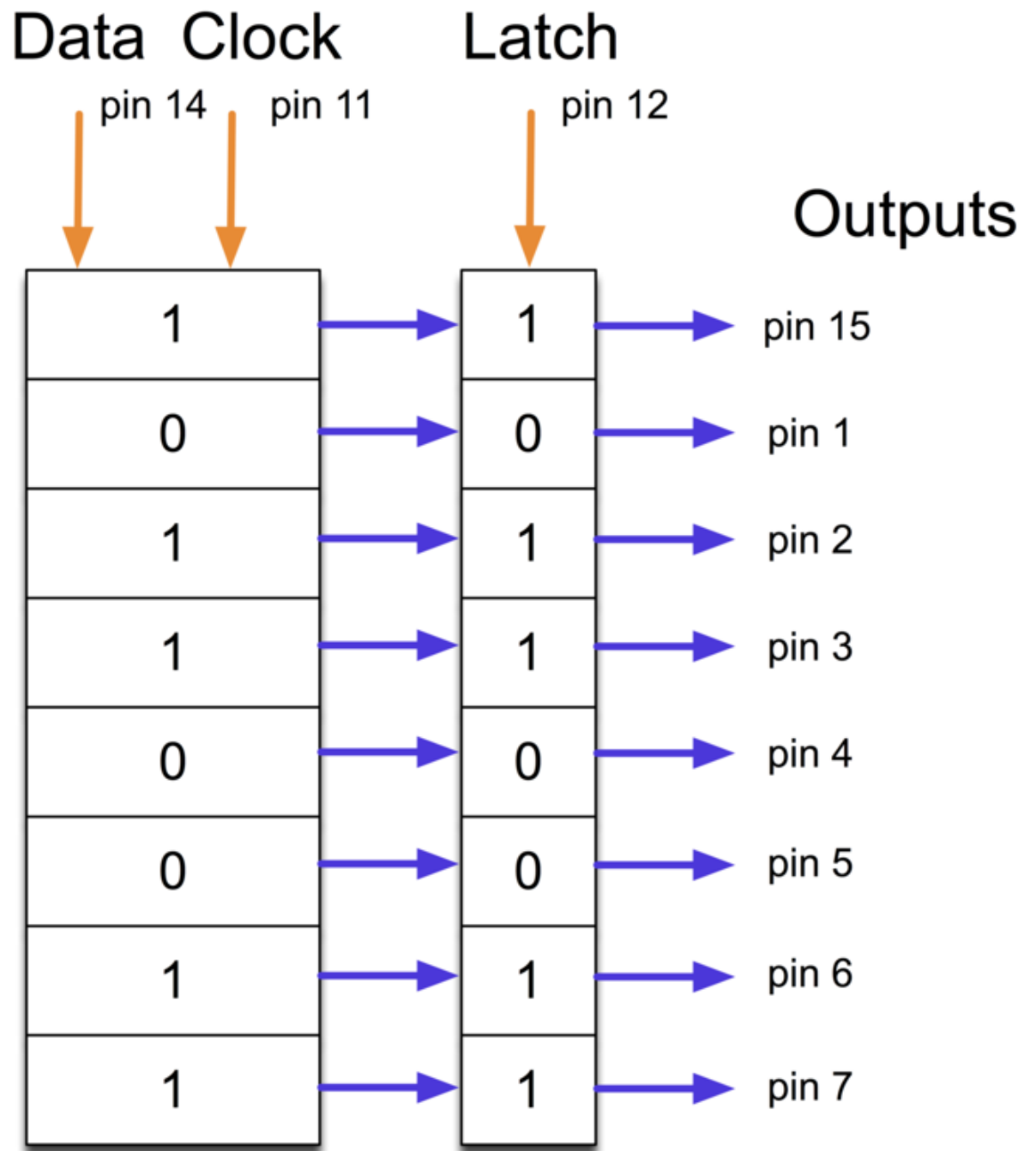
- (1) x Elegoo Uno R3
- (1) x 830 puntos tie breadboard
- (8) x leds
- (8) x resistencias de 220 ohmios
- (1) x IC x 74hc595
- (14) x M M cables (cables de puente de macho amacho)



### Introducción del componente

#### 74HC595 Registro dedesplazamiento:

El registro de desplazamiento es un tipo de chip que tiene lo que puede considerarse como posiciones de memoria ocho, cada uno de ellos puede ser un 1 o un 0. Para definir cada uno de estos valores encendido o apagado, alimentamos en los datos mediante los pines del chip 'Datos' y 'El reloj'.



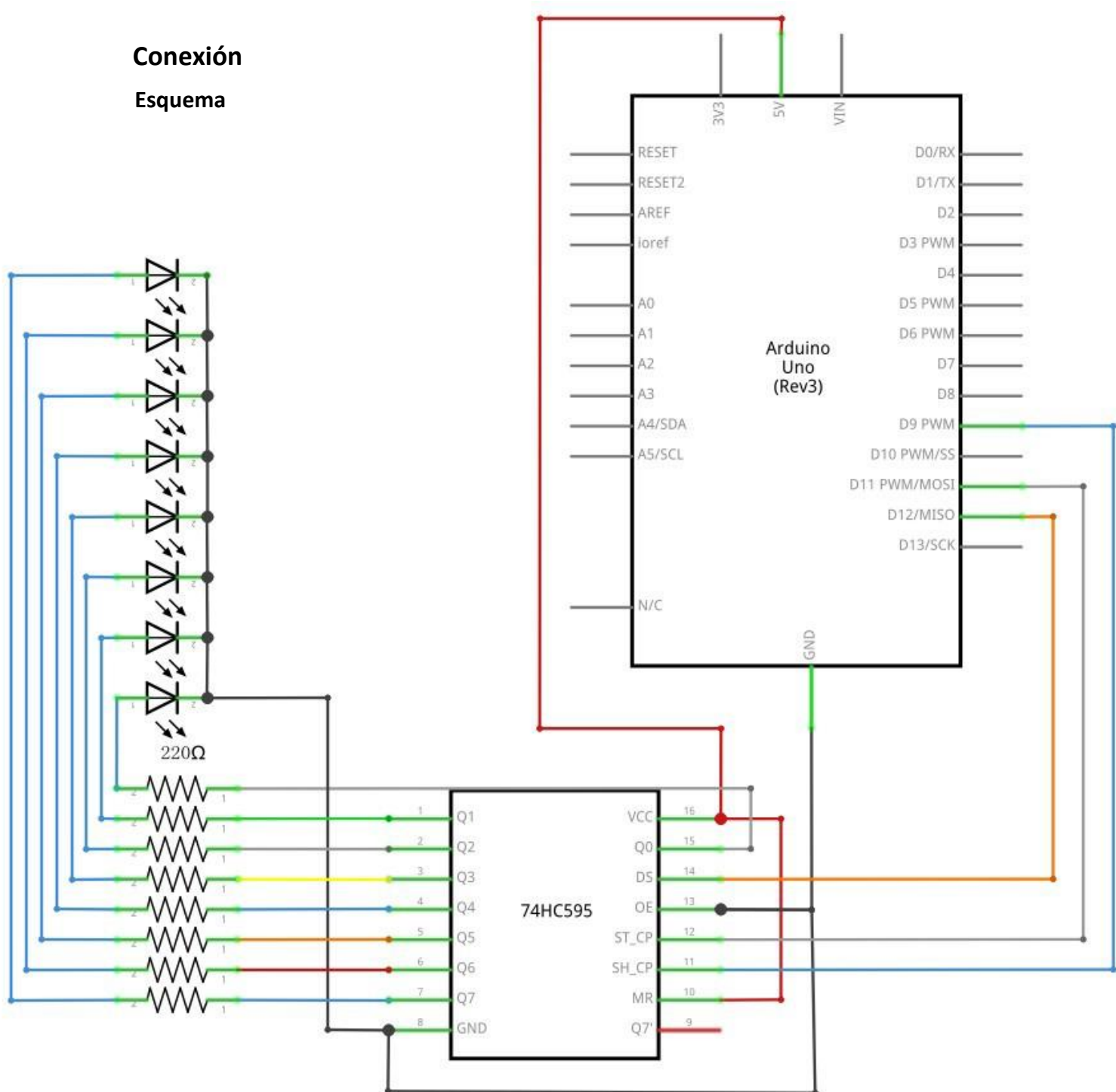
entonces un 1 obtiene empujado en el registro de desplazamiento; de lo contrario,

un 0. Cuando se han recibido los ocho impulsos, permitiendo el pin 'Pestillo' copia esos ocho valores en el registro de cierre. Esto es necesario; de lo contrario, parpadear mal los LEDs como se carga los datos en el registro de desplazamiento.

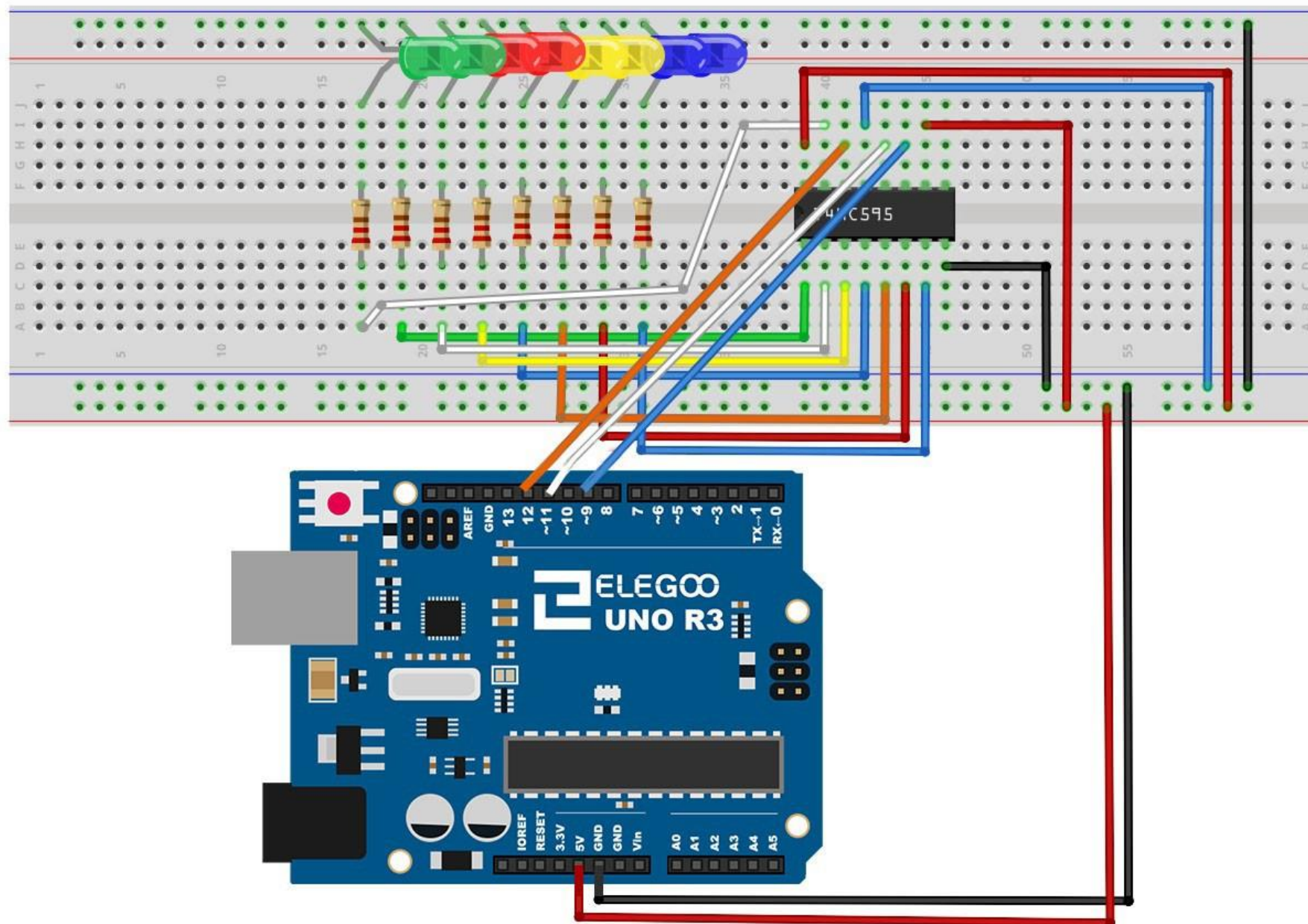
El chip también tiene un pin de salida activado (OE), que se utiliza para activar o desactivar las salidas a la vez. Podría conectar esto a un pin PWM capaz UNO y usar 'analogWrite' para controlar el brillo de los LEDs. Este pin es baja activa, por lo que nos ate a la tierra GND.

## Conexión

### Esquema



## Diagrama de cableado



Ya que tenemos ocho LEDs y ocho resistencias para conectar, hay realmente muy pocas conexiones a realizar.

Es probablemente más fácil poner el chip 74HC595 en primer lugar, como casi todo lo demás se conecta a él. Ponerlo de modo que la muesca en forma de U poco hacia la parte superior de la placa. Pin 1 del chip es a la izquierda de esta muesca.

Digital 12 del UNO va al pin #14 del registro de desplazamiento

Digital 11 del UNO va al pin #12 del registro de desplazamiento

9 digital a partir de la UNO va al pin #11 del registro de desplazamiento

Todos sino una de las salidas de la IC está en el lado izquierdo del chip. Por lo tanto, para facilitar la conexión, es donde están los LEDs, también.

Después de la viruta, poner las resistencias en su lugar. Usted necesita tener cuidado de que ninguno de los cables de las resistencias tocan. Usted debe comprobar esto otra vez antes de conectar la energía a la ONU. Si le resulta difícil organizar las resistencias sin sus conductores tocando, entonces ayuda a acortar los cables que están mintiendo más cercanos a la superficie de la placa.

A continuación, coloque los LEDs en la protoboard. Cuanto más positivo lleva LED debe ser hacia el chip, de cualquier lado de la placa están en.

Conecte los conductores del puente como se muestra arriba. No olvide que va desde el pin 8 del IC a la columna GND de la placa.

Carga el bosquejo aparece un poco más adelante y probar. Cada LED debe encenderse alternadamente hasta que todos los LEDs están encendidos y luego se apagara y el ciclo se repite.

## Código

Después de cableado, por favor, abra el programa en el código de carpeta lección 24 8 LED con 74HC595 y haga clic en UPLOAD para cargar el programa. Ver Lección 2 para más detalles sobre el programa cargar si hay algún error.

Lo primero que hacemos es definir los tres pines que vamos a utilizar. Estos son los UNO salidas digitales que se conectarán a los pines de datos, reloj y cierre de los 74HC595.

```
int latchPin = 11;
```

```
clockPin int = 9;
```

```
int dataPin = 12;
```

A continuación, se define una variable llamada 'leds'. Esto se utiliza para sostener el patrón de que LED actualmente es activado o desactivados. Datos de tipo 'byte' representan números de ocho bits. Cada bit puede estar encendido o apagado, esto es perfecto para realizar un seguimiento de cuáles de nuestros ocho LEDs son on u off.

```
leds de byte = 0;
```

La función de 'configuración' sólo establece los tres pernos que estamos utilizando para ser de salidas digitales.

```
void setup()
{
  pinMode (latchPin, salida);
  pinMode (dataPin, salida);
  pinMode (clockPin, salida);
}
```

La función 'loop' inicialmente apaga todos los LEDs, al darle a los variable 'leds' el valor 0. A continuación, llama 'updateShiftRegister' que enviará el patrón de 'leds' para el registro de desplazamiento para que el LED se apague. Se tratará con 'updateShiftRegister' funcionamiento másadelante.

La función loop hace una pausa de medio segundo y entonces empieza a contar de 0 a 7 usando el bucle 'for' y la variable 'i'. Cada vez utiliza la función de Arduino 'verdadera' para establecer el bit que controla ese LED en la variable 'leds'. A continuación también llama 'updateShiftRegister' para que los leds actualizar para reflejar lo que está en la variable 'leds'.

Hay entonces medio segundo de retraso antes de 'i' se incrementa y se ilumina el LED próximo.

```
void loop()
{
  LED = 0;
  updateShiftRegister(); Delay(500);
  para (int i = 0; i < 8; i++)
  {
    bitSet(leds, i);
  }
}
```

```
updateShiftRegister(); Delay(500);  
}  
}
```

La función 'updateShiftRegister', en primer lugar se establece la latchPin baja, entonces llama al UNO función 'shiftOut' antes de poner el 'latchPin' alta otra vez. Esto toma cuatro parámetros, los dos primeros son los pines para datos y el reloj respectivamente.

El tercer parámetro especifica que final de los datos que desea iniciar en el. Vamos a empezar con la derecha más poco, que se conoce como el 'Bit menos significativo' (LSB).

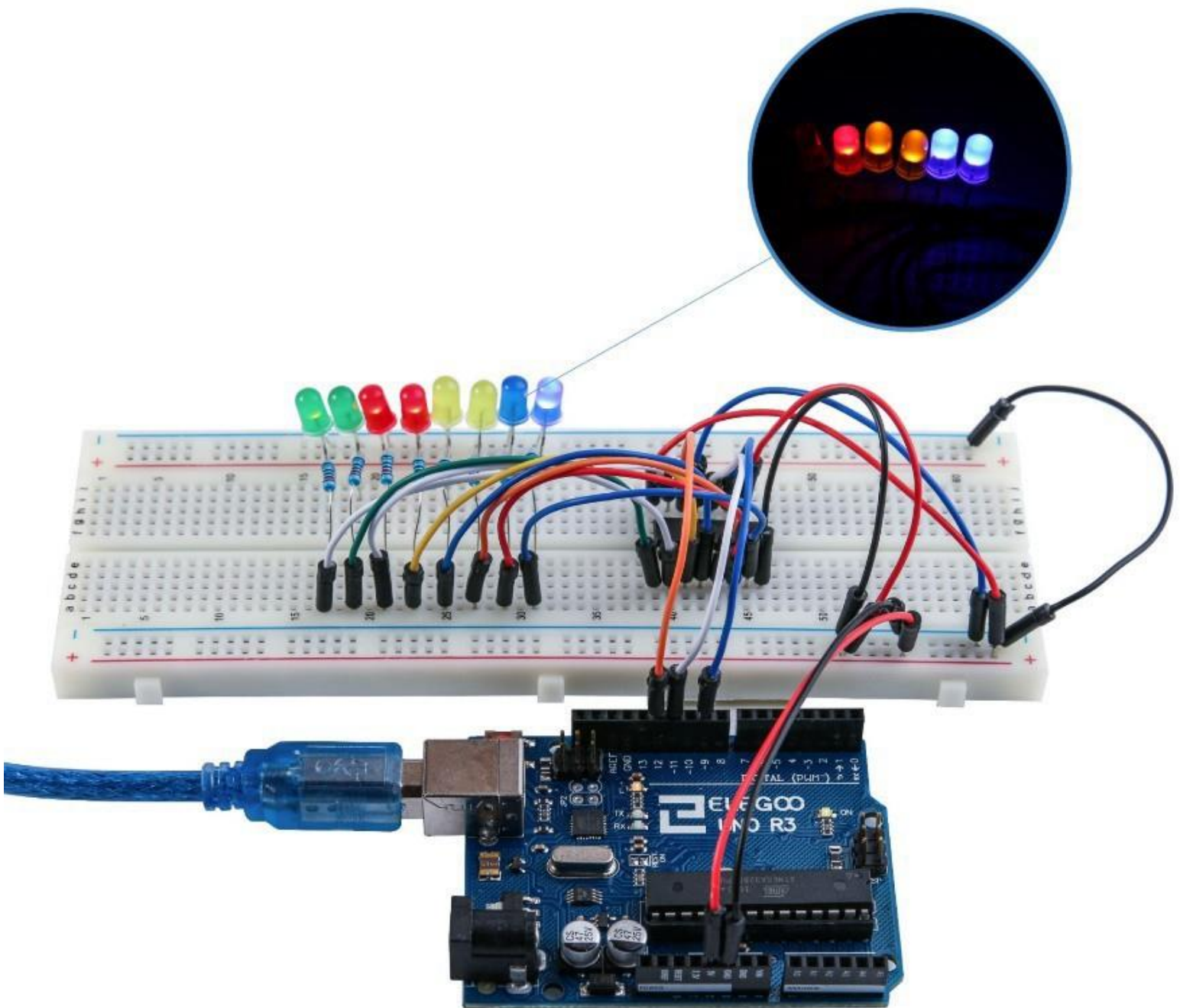
El último parámetro es los datos reales para ser cambiado de puesto en el registro de desplazamiento, que en este caso es 'leds'.

```
void updateShiftRegister()  
{  
  digitalWrite (latchPin,bajo);  
  shiftOut (dataPin, clockPin, LSBFIRST, leds);  
  digitalWrite (latchPin,HIGH);  
}
```

Si usted deseó dar vuelta a uno de los LED apagado en lugar, llamaría una función similar de Arduino (bitClear) con la variable de 'leds'. Esto ajustará ese poco de 'leds' para ser 0 y entonces sólo necesitará seguir con una llamada a 'updateShiftRegister' para actualizar la actual LED.

.

Imágen de ejemplo



## Lección 9 El monitor Serial

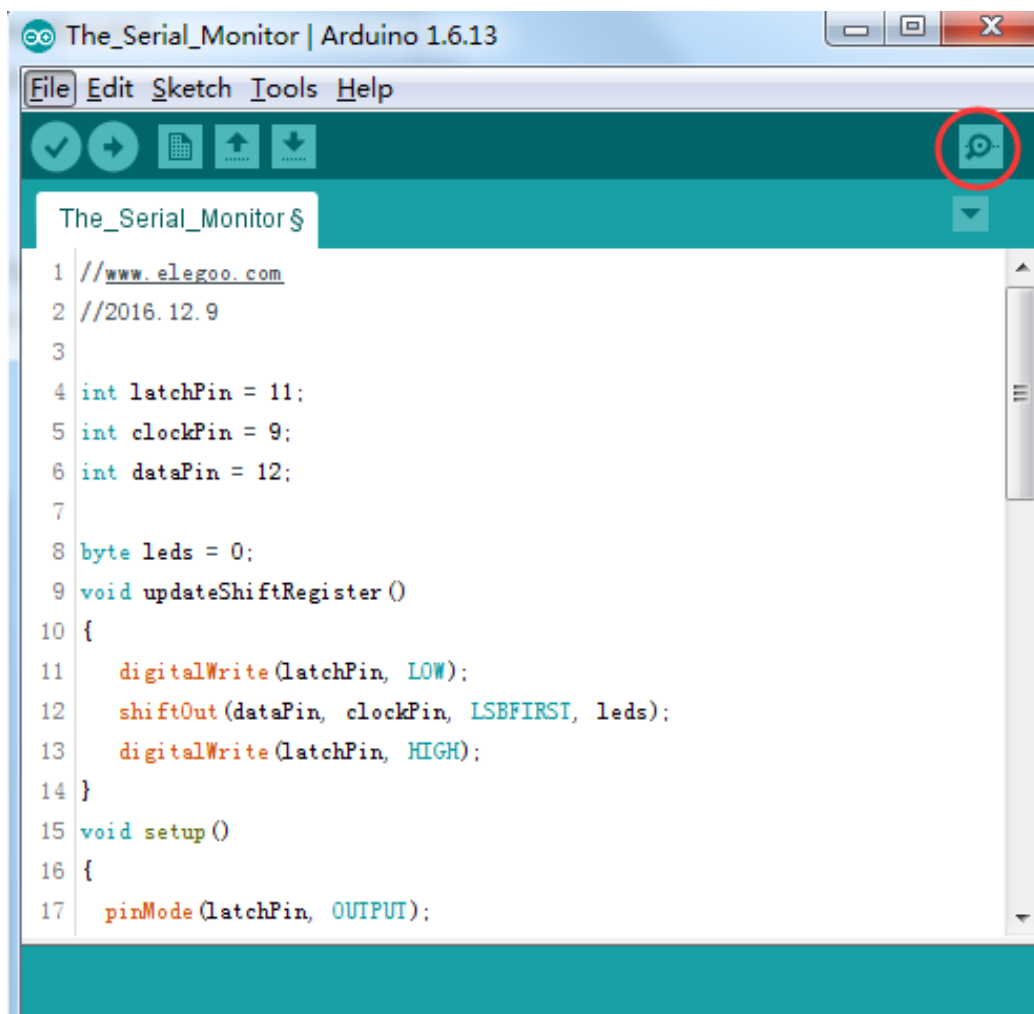
### Resumen

En esta lección, se basará en la lección 8, añadiendo la facilidad de controlar los LEDs desde el ordenador con el Monitor serie del Arduino. El monitor serial es el 'cable' entre el ordenador y tu UNO. Le permite enviar y recibir mensajes de texto, útiles para la depuración y también control de la ONU de un teclado! Por ejemplo, usted será capaz de enviar comandos desde el ordenador para encender LEDs.

En esta lección, utilizará exactamente las mismas piezas y una disposición similar del protoboard como lección 8. Por lo tanto, si aún no lo ha hecho, siga lección 8 ahora.

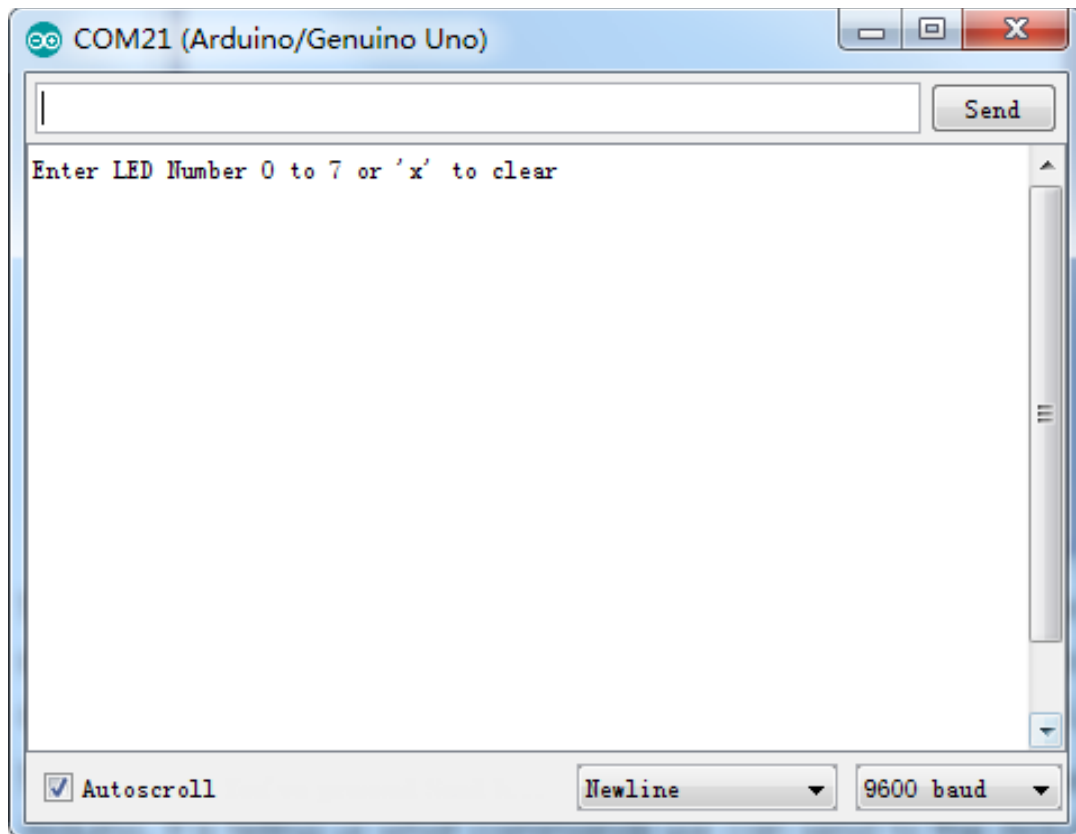
### Medidas adoptadas

Después de que han subido este cableado sobre el UNO, haga clic en el botón derecho en la barra de herramientas en el IDE de Arduino. Es en un círculo el botón a continuación.



Se abrirá la siguiente ventana.

Haga clic en el botón Serial Monitor para encender el monitor serie. Se introducen los conceptos básicos sobre el monitor serial en detalles en la lección 1.

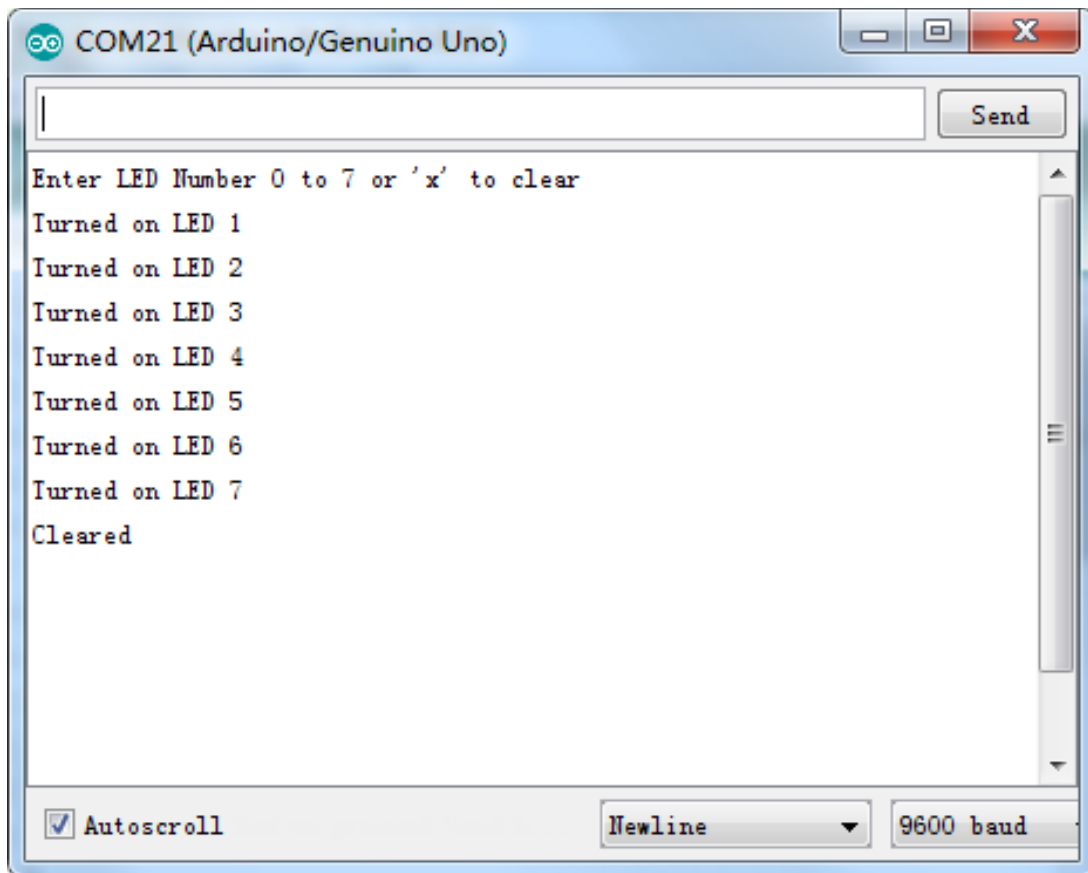


Esta ventana se llama al Monitor Serial y es parte del software del IDE de Arduino. Su trabajo es la que permite a ambos enviar mensajes desde tu ordenador a una placa UNO (por USB) y también recibir mensajes de la placa UNO.

El mensaje "Enter LED número 0 a 7 or 'x' para borrar" ha sido enviado por el Arduino. Nos está diciendo qué comandos podemos enviar a Arduino: o enviar la 'x' (para apagar todos los LEDs) o el número de LED que desea activar (donde 0 es el LED de la parte inferior, 1 es la siguiente, hasta 7 para el LED superior).

Trate de escribir los siguientes comandos en la parte superior del Monitor Serial que es el nivel con el botón 'Enviar'. Presione 'Enviar', después de escribir cada uno de estos caracteres: x 0 3 5

Escribir x voluntad no tienen ningún efecto si los LEDs ya están todos fuera, pero al entrar en cada número, el correspondiente LED deberá encenderse y usted recibirá un mensaje de confirmación de la placa UNO. El Monitor Serial aparecerá como se muestra a continuación.



Escriba x otra vez y pulse 'Enviar' para apagar todos los LEDs

## Código

Después de efectuar el cableado, por favor abra el Serial Monitor del programa en la carpeta de código-lección 25 y haga clic en cargar para cargar el programa. Ver [Lección 2](#) para obtener más información sobre programa cargar si hay algún error.

Como era de esperar, el cableado se basa en el cableado utilizado en la lección 24. Por lo tanto, sólo cubrimos los nuevos bits aquí. Le resultará útil para referirse al dibujo completo en el IDE de Arduino.

En la función de 'configuración', hay tres nuevas líneas al final:

```
void setup()
{
  pinMode (latchPin, salida);
  pinMode (dataPin, salida);
  pinMode (clockPin, salida);
  updateShiftRegister();
}
```

```

Serial.Begin(9600);

mientras (!. Serie); Espere hasta que es listo - Leonardo

Serial.println ("Enter LED número de 0 a 7 o 'x' para borrar");
}

```

En primer lugar, tenemos el comando 'Serial.begin(9600)'. Esto inicia la comunicación serial, para que la UNO puede enviar comandos a través de la conexión USB. El valor 9600 es la configuración velocidad de la conexión. Esto es la rapidez con la que los datos debe ser enviado. Esto puede cambiar a un valor más alto, pero también tendrás que cambiar al monitor de Arduino Serial el mismo valor. Hablaremos de esto más adelante; por ahora, dejar en 9600.

El comienzo de la línea con 'mientras' asegura que hay algo en el otro extremo de la conexión USB para Arduino hablar antes de que comience el envío de mensajes. De lo contrario, el mensaje puede ser enviado, pero no aparece. Esta línea es realmente sólo es necesaria si está utilizando a un Arduino Leonardo porque el Arduino UNO se restablece automáticamente la placa Arduino al abrir el Monitor de la serie, mientras que esto no sucede con el Leonardo.

La última de las nuevas líneas en 'configuración' envía el mensaje que vemos en la parte superior del Monitor serie.

La función de 'bucle' es donde sucede toda la acción

```

void loop()
{
    if (Serial.available())
    {
        char ch = Serial.read();
        if (ch >= '0' && ch <= '7')
        {
            int led = ch - '0';
            bitSet(leds, led);
            updateShiftRegister();
            Serial.print("Turned on LED ");
            Serial.println(led);
        }
        if (ch == 'x')
        {
            leds = 0;
        }
    }
}

```

```

        updateShiftRegister();
        Serial.println("Cleared");
    }
}
}

```

Todo lo que ocurre dentro del bucle está contenido dentro de una instrucción 'if'. Así que a menos que la llamada a la función incorporada de Arduino 'Serial.available()' es 'true' entonces nada sucederá.

Serial.Available() devuelve 'true' si los datos han sido enviados a la ONU y allí están listos para ser procesados. Los mensajes entrantes se llevan a cabo en lo que se llama un búfer y Serial.available() devuelve true si ese buffer es no vacío.

Si un mensaje ha sido recibido, es a la siguiente línea de código:

```
char ch = Serial.read();
```

Esto lee el siguiente carácter del búfer y elimina del buffer. También asigna a la variable 'ch'. La variable 'ch' es de tipo 'char' que significa 'carácter' y como su nombre indica, tiene un carácter único.

Si usted ha seguido las instrucciones en el prompt en la parte superior del Monitor Serial, luego este personaje será o bien un número dígito entre 0 y 7 o la letra 'x'.

La instrucción 'if' en la línea siguiente comprueba para ver si es un solo dígito por ver si 'ch' es mayor o igual que el carácter '0' y menor o igual que el personaje '7'. Parece un poco extraño comparar caracteres de esta manera, pero es perfectamente aceptable.

Cada carácter está representado por un número único, conocido su valor ASCII. Esto significa que cuando se comparan caracteres usando  $< =$  y  $> =$  es realmente los valores ASCII que se estaban comparando.

Si pasa la prueba, llegamos a la siguiente línea:

```
int led = ch-'0';
```

¡Ahora estamos actuando aritmética en los personajes! Estamos restando el dígito '0' de cualquier dígitos fue introducido. Por lo tanto, si escribió '0' y luego '0' a '0' será igual a 0. Si escribió '7' y '7'-'0' será igual al número 7 ya que es realmente los valores ASCII que se utilizan en la sustracción.

Desde sabemos que el número del LED que queremos encender, nos basta establecer este bit en la variable 'leds' y actualizar el registro de desplazamiento.

```
bitSet (leds, led);
```

```
updateShiftRegister();
```

Las dos líneas escriben de nuevo un mensaje de confirmación en el Monitorserie.

```
Serial.Print ("encendido LED");
```

```
Serial.println(LED);
```

La primera línea utiliza Serial.print en lugar de Serial.println. La diferencia entre los dos es que Serial.print no se inicia una nueva línea después de imprimir lo que está en su parámetro. Usamos esto en la primera línea, porque estamos impresión el mensaje en dos partes. En primer lugar el general bits: 'Enciende LED' y luego el número del LED.

El número del LED se realiza en un 'int' variable en lugar de ser una cadena de texto. Serial.Print puede tomar ya sea una cadena de texto dentro de comillas dobles, o un 'int' o para el caso casi cualquier tipo de variable.

Después de la instrucción 'if' que maneja el caso, cuando un dígito se ha manejado, hay una segunda instrucción 'if' que comprueba si la 'ch' es la letra 'x'.

```
Si (ch == 'x')
```

```
{ LE
```

```
D = 0;
```

```
updateShiftRegister();
```

```
Serial.println("cleared");
```

```
}
```

Si es así, entonces se borran todos los LEDs y envía un mensaje de confirmación.

## Lección 10 Fotocélula

### Resumen

En esta lección, usted aprenderá cómo medir la intensidad de la luz utilizando una entrada analógica. Se construirá en la lección 8 y utilice el nivel de luz para controlar el número de LEDs que se encenderán.

La fotocélula es en la parte inferior de la placa, donde estaba el bote por encima.

### Componentes necesarios:

- (1) x Elegoo Uno R3
- (1) x 830 tie puntos breadboard
- (8) x leds
- (8) x resistencias de 220 ohmios
- (1) x resistencia de 1 kohm
- (1) x IC 74hc595
- (1) x fotoresistor(fotocélula)
- (16) x M M cables (cables de puente de macho amacho)



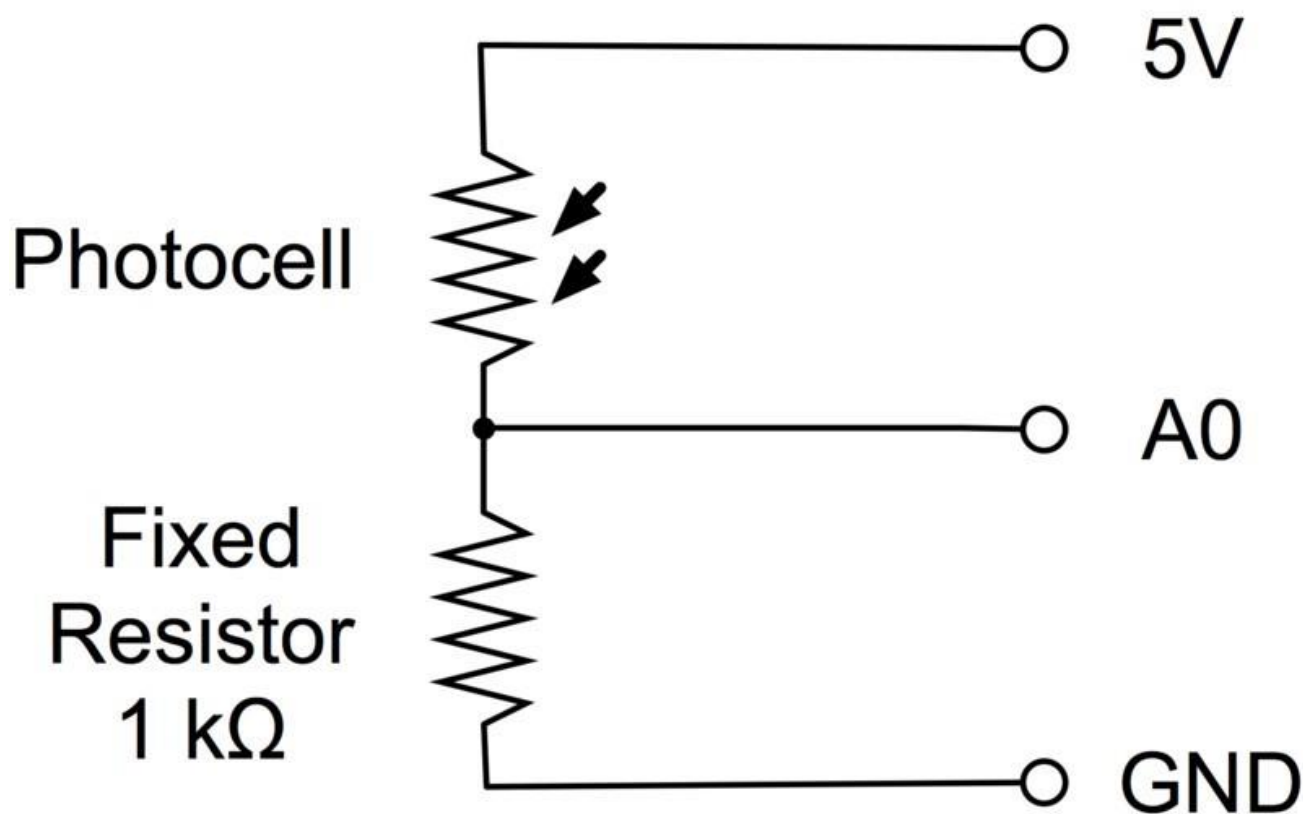
### Introducción del componente

#### FOTOCÉLULA:

La fotocélula utilizada es de un tipo llamado un resistor dependiente de luz, a veces llamado un LDR. Como su nombre indica, estos componentes actúan como una resistencia, excepto que la resistencia cambia en respuesta a cuanto luz está cayendo sobre ellos.

Esta tiene una resistencia de cerca de 50 k $\Omega$  en cerca de oscuridad y  $\Omega$  500 en luz brillante. Para convertir este valor variable de la resistencia en algo que podemos medir en la entrada analógica de la Junta de un R3 de UNO, debe ser convertida en un voltaje.

La forma más sencilla de hacerlo es combinar con una resistencia fija.

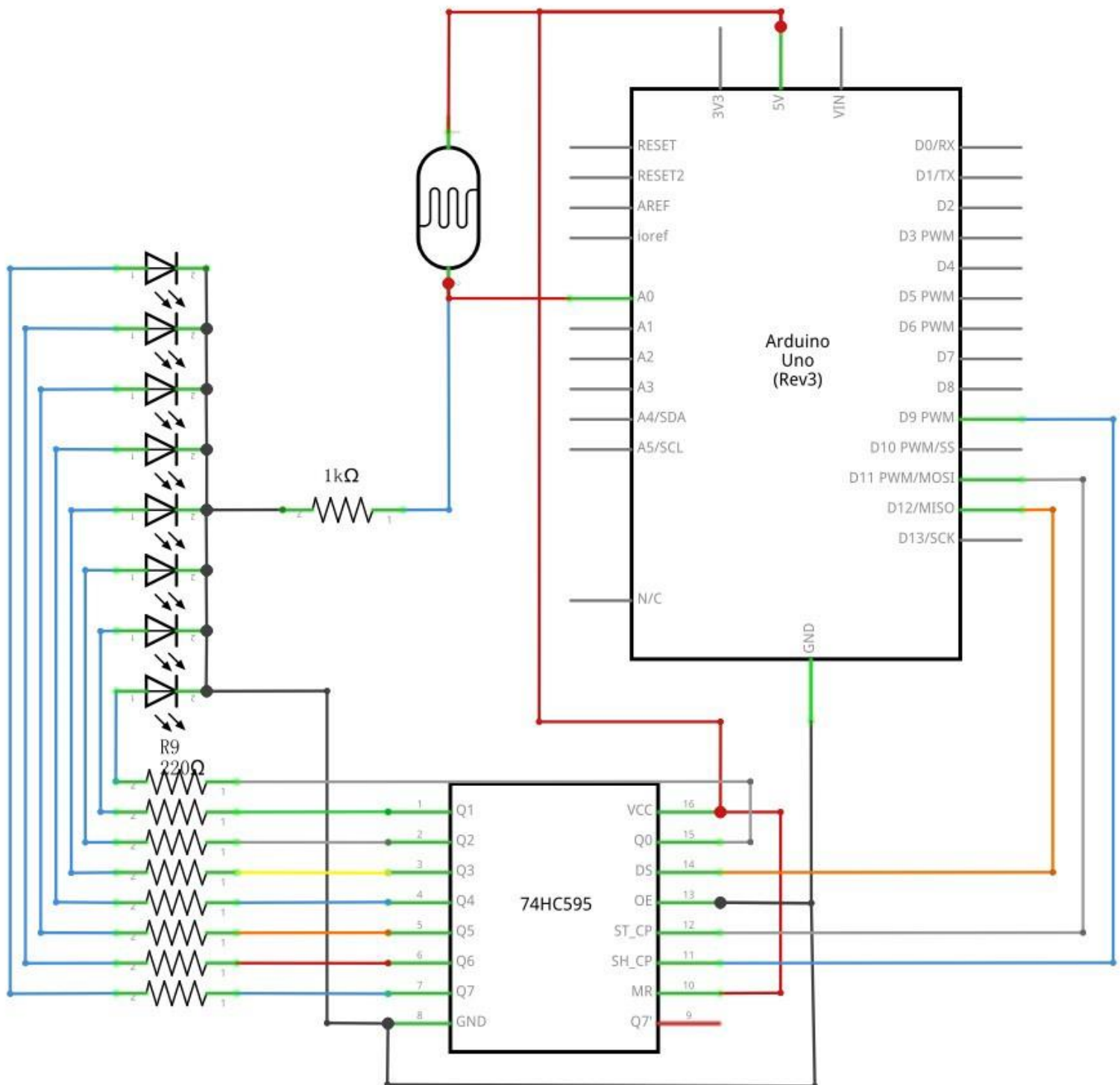


La resistencia y fotocélula junto se comportan como una sola. Cuando la luz es muy brillante, entonces la resistencia de la fotocélula es muy baja en comparación con la resistencia de valor fijo, y así es como si el bote se dio vuelta a máximo.

Cuando la fotocélula está en una luz apagada, la resistencia es mayor que la resistencia fija de 1 kΩ y es como si el recipiente estuviera girando hacia GND.

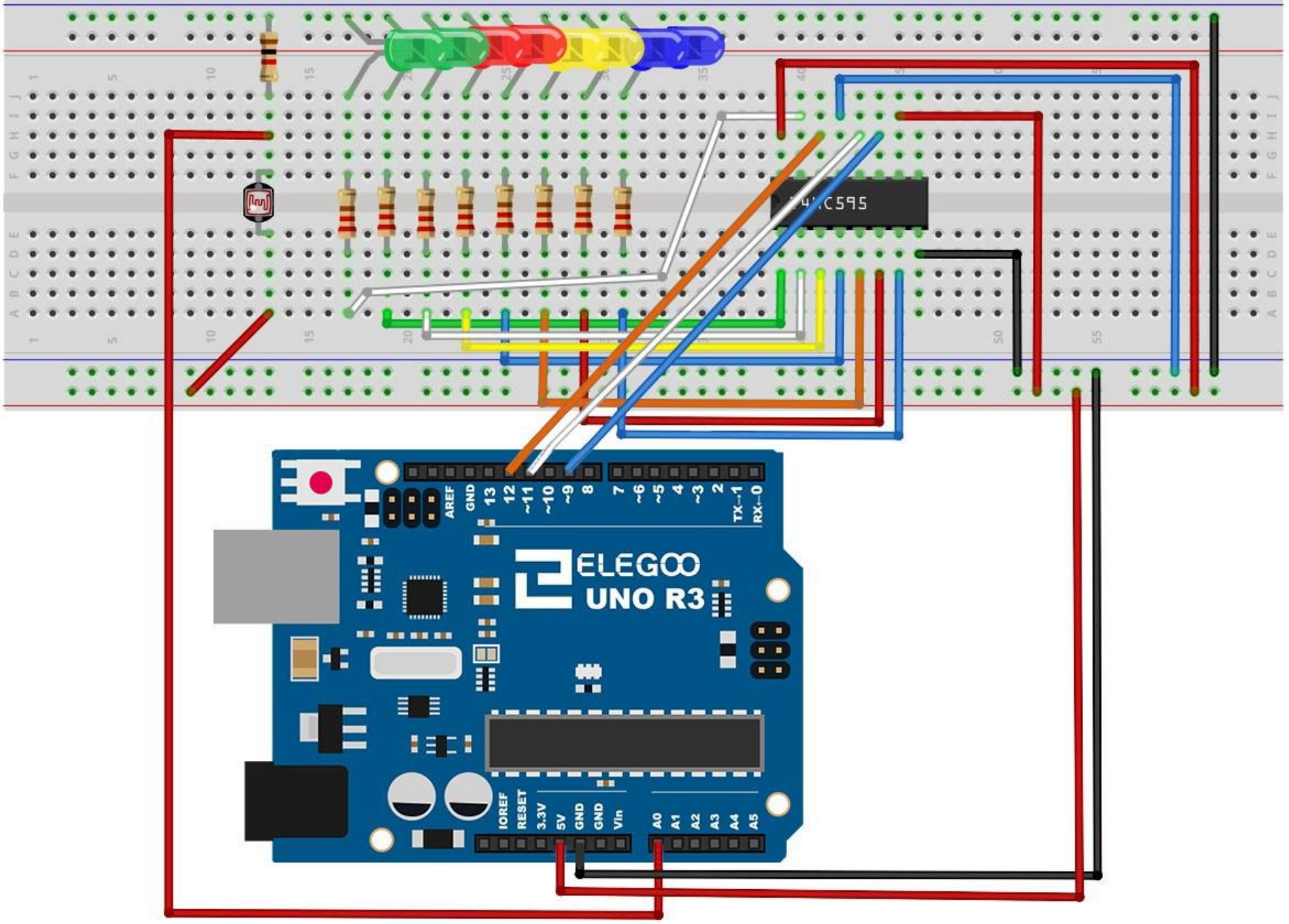
Cargue el croquis dado en la siguiente sección y trate de cubrir la fotocélula con el dedo y, a continuación, sosténgalo cerca de una fuente de luz.

## Conexión Esquema



The image shows an Arduino Uno R3 microcontroller board connected to a breadboard-based circuit. The breadboard contains a 74HC595 shift register IC, several resistors, and a 7-segment display. The circuit is wired as follows:

- Power and Ground:** The 5V pin of the Arduino is connected to the VCC pin of the 74HC595 and one end of a red wire. The GND pin of the Arduino is connected to the GND pin of the 74HC595 and the other end of the red wire. A black wire connects the GND pin of the 74HC595 to the GND pin of the Arduino.
- Data and Control:** The TX-0 pin of the Arduino is connected to the DATA pin of the 74HC595. The TX-1 pin of the Arduino is connected to the CLOCK pin of the 74HC595. The RX-0 pin of the Arduino is connected to the MASTER/SLAVE SELECT pin of the 74HC595.
- Resistors:** Several resistors are connected to the output pins of the 74HC595, likely to limit current to the 7-segment display.
- 7-segment Display:** The 7-segment display is connected to the output pins of the 74HC595, with each segment having its own current-limiting resistor.



## Código

Después de efectuar el cableado, por favor, abra el programa en el código de carpeta - lección 26 fotocélula y haga clic en **UPLOAD** para cargar el programa. Ver [Lección 2](#) para obtener más información sobre programa cargar si hay algún error.

Lo primero que note es que hemos cambiado el nombre del pin analógico a ser 'apagado' en lugar de 'potPin' ya que no tenemos una olla conectada.

El sólo cambio substancial al dibujo es la línea que calcula cuántos de los LEDs de luz:

```
int numLEDsLit = reading / 57; // all LEDs lit at 1k
```

Este tiempo, dividimos la lectura cruda 57 en lugar de 114. En otras palabras, nos dividirlo por la mitad, tanto como lo hicimos con el bote para dividirlo en nueve zonas, de ningún LED encendido a ocho todas iluminada. Este factor adicional es para tener en cuenta la resistencia fija de kΩ 1. Esto significa que cuando la fotocélula tiene una resistencia de 1 kΩ (igual a la resistencia fija), la lectura cruda será  $1023 / 2 = 511$ . Esto se proporcionan a todos los LED se enciende y luego un poco (numLEDsLit) será 8.

## Imagen de ejemplo

