

330ohms[®]



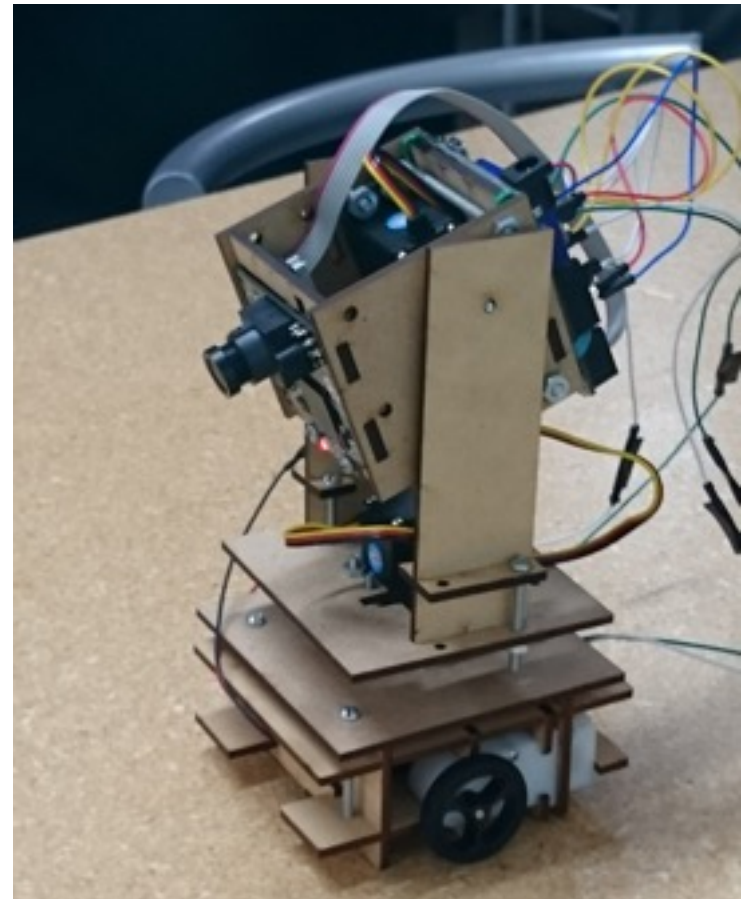
MAKERSPACE

Curso Básico de Arduino



¿Quiénes somos?

- Somos una empresa mexicana que facilita el acceso a herramientas y productos innovadores y tecnológicos a la población de nuestro país, para proporcionar elementos que permitan el desarrollo de sistemas de alta calidad que sean capaces de competir a nivel internacional.





¿Qué hacemos?

En 330ohms nos dedicamos a:

- Venta de componentes electrónicos
- Desarrollo de proyectos
- Impartición de cursos y asesorías
- Servicio de corte, grabado láser, e impresión 3D
- Makerspace.



Contenido del curso

1 Introducción a Arduino

- 1.1 Descripción de la tarjeta Arduino UNO
- 1.2 Otras tarjetas Arduino y Shields
- 1.3 Instalación del entorno de desarrollo
- 1.4 El lenguaje de programación de Arduino

2 Los puertos digitales de la tarjeta Arduino UNO

- 2.1 ¿Qué es un circuito eléctrico?
- 2.2 Salidas digitales
- 2.3 Entradas digitales

3 Los puertos analógicos de la tarjeta Arduino UNO

- 3.1 Modulación por ancho de pulso
- 3.2 Entradas analógicas
- 3.3 Salidas analógicas

Contenido del curso

4 Despliegue de información en una pantalla LCD

- 4.1 Conexión de una pantalla LCD
- 4.2 Lectura de sensores y despliegue en LCD
- 4.3 Algunas funciones de la pantalla LCD
- 4.4 Programación de un menú interactivo básico

5 Manejo básico de actuadores

- 5.1 Servomotores
- 5.2 Motores de rotación continua
- 5.3 Conexión del puente H L293D
- 5.4 Motores de pulsos (paso a paso)



1 Introducción a Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. Está dirigida a cualquier persona que hace proyectos interactivos.





1.1 Descripción de la tarjeta Arduino UNO



Microcontrolador ATmega328



1.1 Descripción de la tarjeta Arduino UNO



14 pines que pueden configurarse como entradas o salidas digitales



1.1 Descripción de la tarjeta Arduino UNO



6 entradas analógicas



1.1 Descripción de la tarjeta Arduino UNO

Oscilador cerámico de 16 MHz





1.1 Descripción de la tarjeta Arduino UNO

Conector USB





1.1 Descripción de la tarjeta Arduino UNO

Jack para alimentación con fuente externa





1.1 Descripción de la tarjeta Arduino UNO



Botón de reinicio

1.3 Instalación del entorno de desarrollo



- Arduino UNO



- Cable estándar USB
(Tipo A - B)

1.3 Instalación del entorno de desarrollo

- Descarga el software de Arduino
- Descomprime y ejecuta.
- Conecta tu tarjeta Arduino a la computadora por medio del cable USB
- Sigue los pasos del asistente para instalar



1.4 El lenguaje de programación de Arduino

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

1 Arduino Uno on /dev/tty.usbmodemfd131

1.4 El lenguaje de programación de Arduino

Estructura	Variables	Funciones
<ul style="list-style-type: none"> • setup() • loop() <p>Estructuras de control</p> <ul style="list-style-type: none"> • if • if else • for • switch case • while • do... while • break • continue • return • Goto <p>Sintaxis</p> <ul style="list-style-type: none"> • ; (punto y coma) • { } (llaves) • // (comentarios en una línea) 	<p>Constantes</p> <ul style="list-style-type: none"> • HIGH LOW • INPUT OUTPUT INPUT_PULLUP • True False • integer constants • floating point constants <p>Tipos de datos</p> <ul style="list-style-type: none"> • void • boolean • char • unsigned char • byte • int • unsigned int • word • long • unsigned long 	<p>Digital I/O</p> <ul style="list-style-type: none"> • pinMode() • digitalWrite() • digitalRead() <p>Analog I/O</p> <ul style="list-style-type: none"> • analogReference() • analogRead() • analogWrite() –PWM <p>Solo Due</p> <ul style="list-style-type: none"> • analogReadResolution() • analogWriteResolution() <p>I/O Avanzado</p> <ul style="list-style-type: none"> • tone() • noTone() • shiftOut() • shiftIn

1.4 El lenguaje de programación de Arduino

Estructura	Variables	Funciones
<ul style="list-style-type: none"> • <i>/**/ (multiline comment)</i> • <i>#define</i> • <i>#include</i> <p>Operadores Aritméticos</p> <ul style="list-style-type: none"> • <i>= (operador de asignación)</i> • <i>+</i> (suma) • <i>-</i> (resta) • <i>*</i> (multiplicación) • <i>/</i> (división) • <i>%</i> (modulo) <p>Operadores Lógicos</p> <ul style="list-style-type: none"> • <i>==</i> (igual a) • <i>!=</i> (diferente a) • <i><</i> (menor que) • <i>></i> (mayor que) • <i><=</i> (menor o igual a) • <i>>=</i> (mayor o igual a) 	<ul style="list-style-type: none"> • <i>short</i> • <i>float</i> • <i>double</i> • <i>string</i> – Char array • <i>String</i> – objeto • <i>array</i> <p>Conversiones</p> <ul style="list-style-type: none"> • <i>char()</i> • <i>byte()</i> • <i>int()</i> • <i>word()</i> • <i>long()</i> • <i>float()</i> <p>Alcance variable y calificadores</p> <ul style="list-style-type: none"> • <i>variable scope</i> • <i>static</i> • <i>volatile</i> • <i>const</i> 	<ul style="list-style-type: none"> • <i>pulseIn()</i> <p>Time</p> <ul style="list-style-type: none"> • <i>millis()</i> • <i>micros()</i> • <i>delay()</i> • <i>delayMicroseconds()</i> <p>Matemáticos</p> <ul style="list-style-type: none"> • <i>min()</i> • <i>max()</i> • <i>abs()</i> • <i>constrain()</i> • <i>map()</i> • <i>pow()</i> • <i>sqrt()</i> <p>Trigonometría</p> <ul style="list-style-type: none"> • <i>sin()</i> • <i>cos()</i> • <i>tan()</i>

1.4 El lenguaje de programación de Arduino

Estructura	Variables	Funciones
Operadores Booleanos <ul style="list-style-type: none">• && (and)• (or)• !(not) Operadores de punto de acceso <ul style="list-style-type: none">• * operador de diferencia• & operador de referencia Operadores bit a bit <ul style="list-style-type: none">• & (bit a bit and)• (bit a bit or)• ^ (bit a bit xor)• ~ (bit a bit not)• << (desplazamiento de bit izquierda)	Utilidades <ul style="list-style-type: none">• sizeof()	Números aleatorios <ul style="list-style-type: none">• randomSeed()• random() Bits y Bytes <ul style="list-style-type: none">• lowByte()• highByte()• bitRead()• bitWrite()• bitSet()• bitClear()• bit()

2 Los puertos digitales de la tarjeta Arduino UNO

14 pines que pueden configurarse como entradas o salidas digitales



Funciones Digital I/O

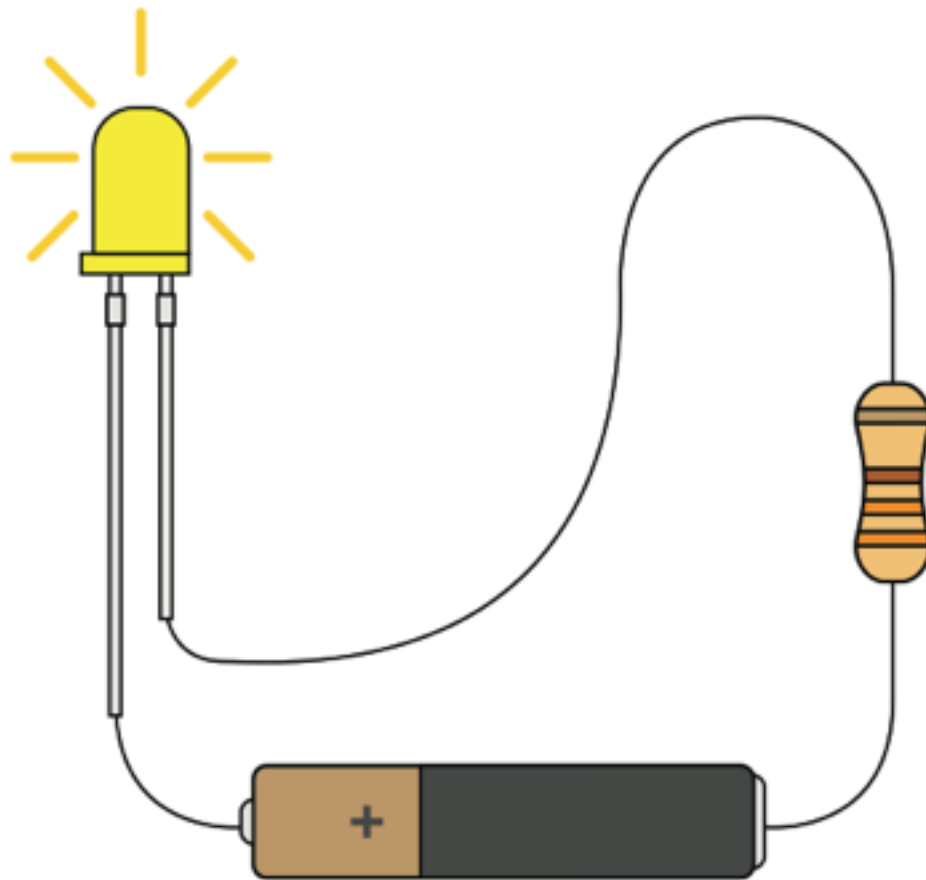
`pinMode(pin, modo)`

`digitalWrite(pin, valor)`

`digitalRead(pin)`

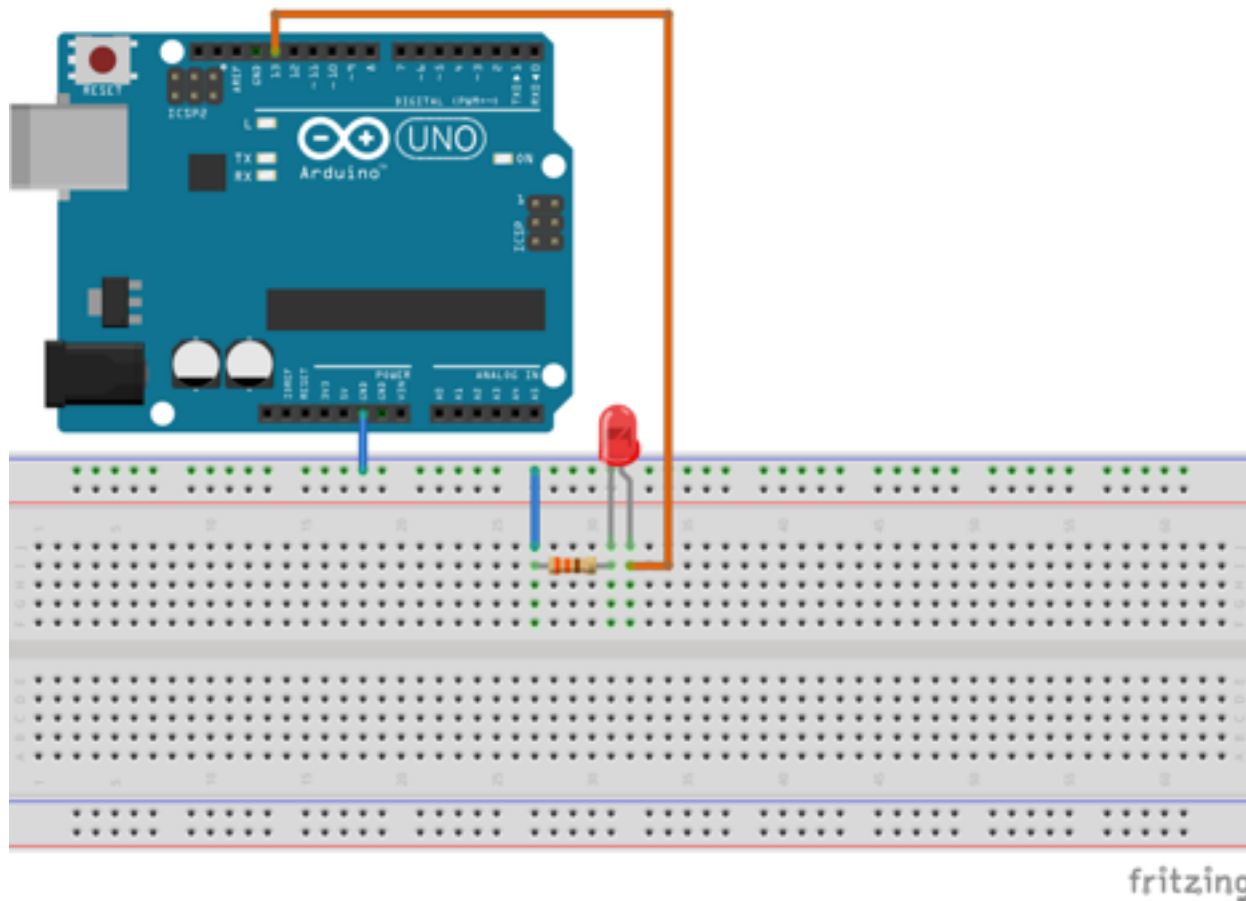


2.1 ¿Qué es un circuito eléctrico?



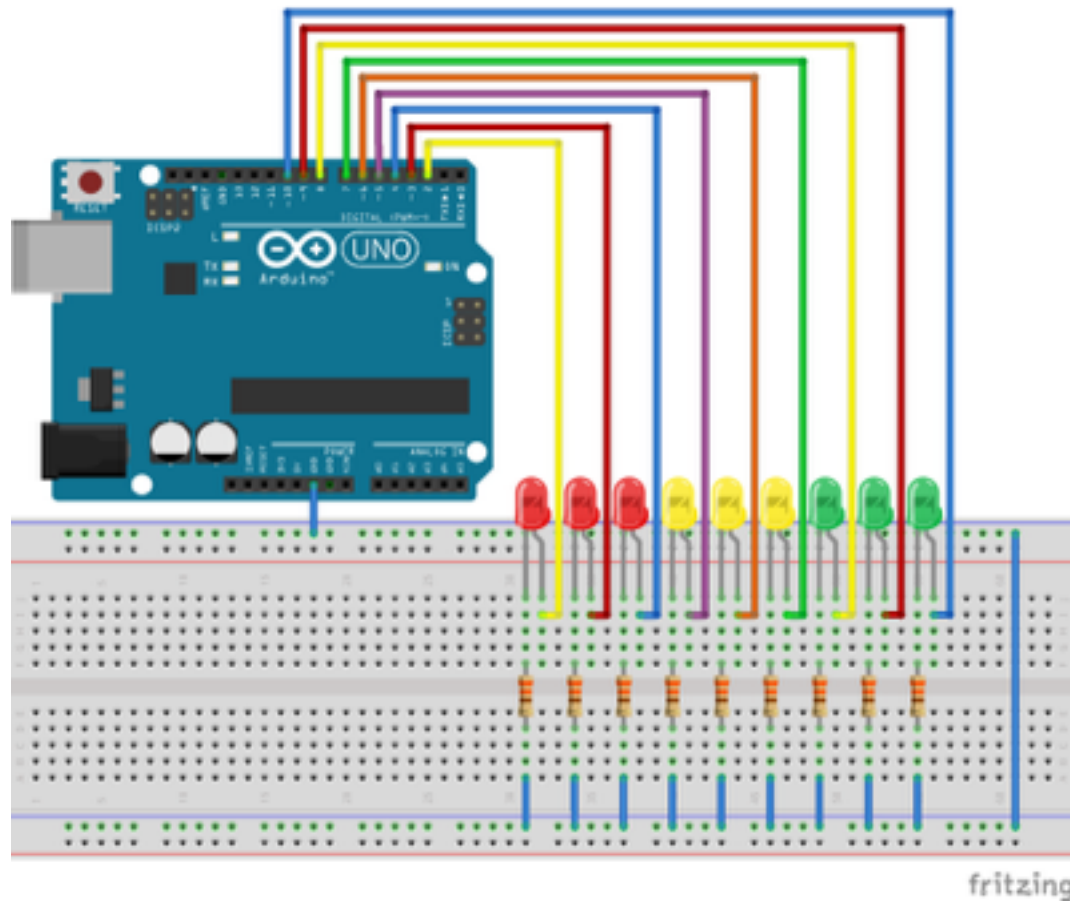
2.2 Salidas digitales

Circuito 1: Hacer parpadear un led externo conectado a la tarjeta Arduino



2.2 Salidas digitales

Circuito 2: Encender un conjunto de leds para realizar las diferentes secuencias



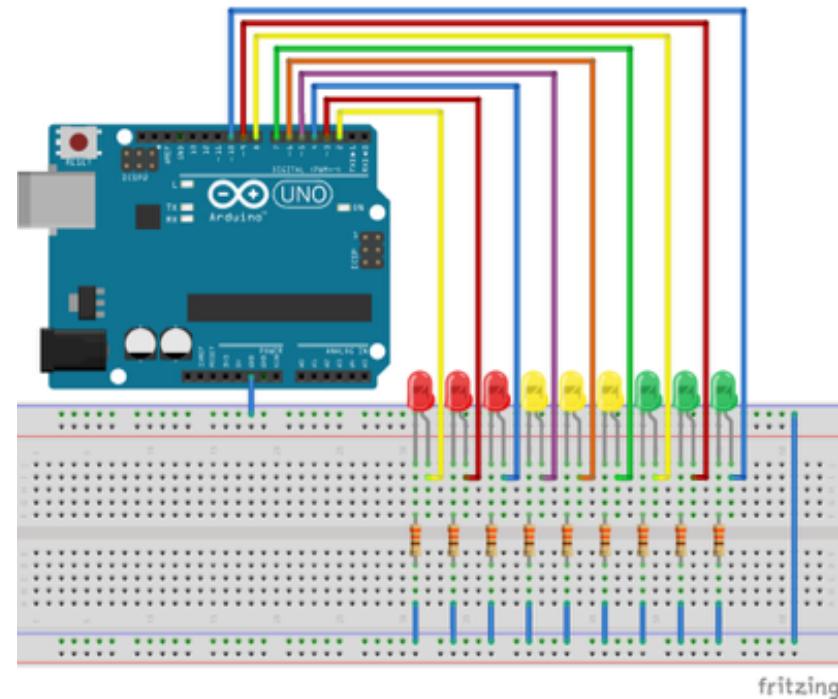
2.2 Salidas digitales

Circuito 2: Encender un conjunto de leds para realizar las siguientes secuencias

Encender uno a uno cada led hasta que todos estén encendidos. Comenzar con el led del extremo izquierdo y continuar hasta el led del extremo derecho

Encender cada led, de modo que sólo uno esté encendido a la vez, comenzando por el led del extremo izquierdo y continuar hasta el led del extremo derecho

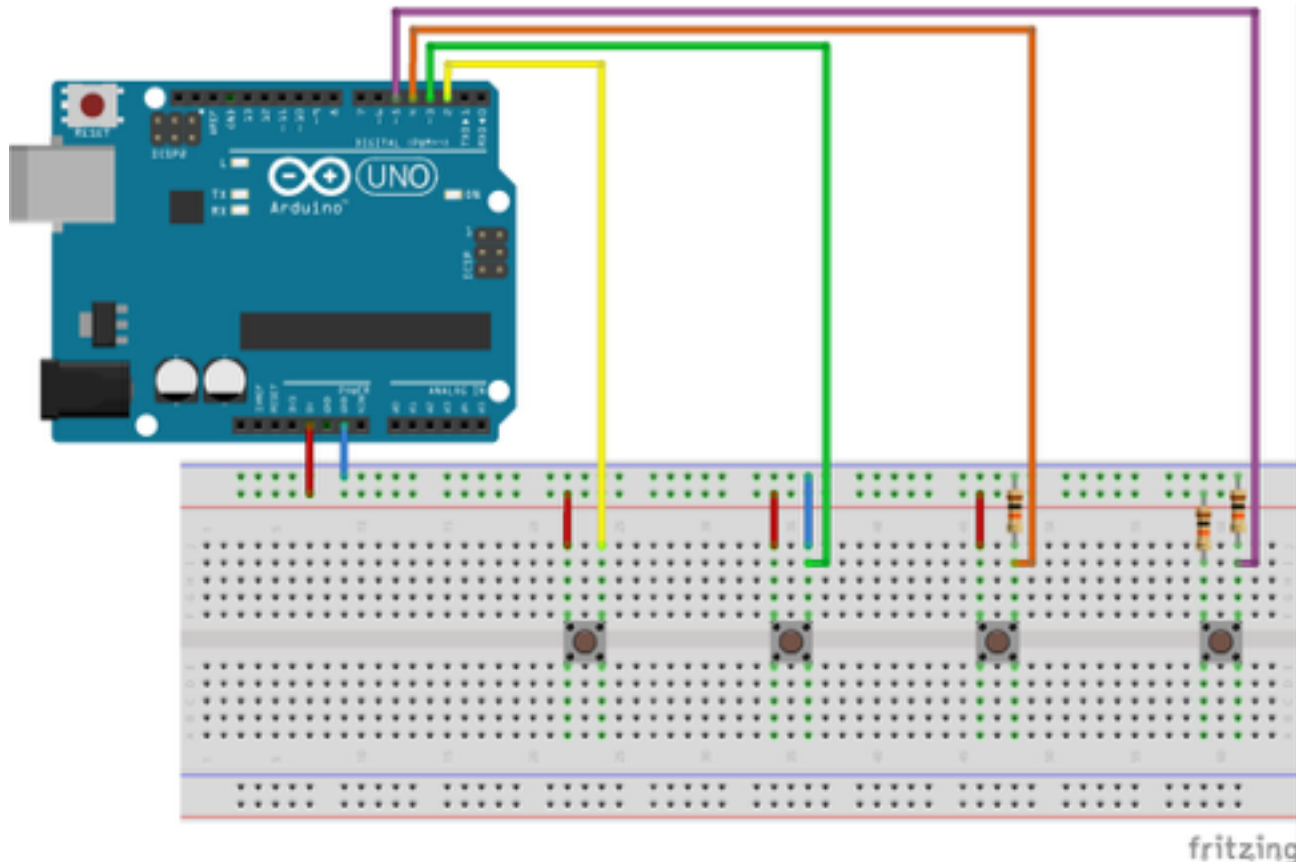
Encender cada led, de modo que sólo uno esté encendido a la vez, comenzando por el led del extremo izquierdo y continuar hasta el led del extremo derecho, posteriormente hacer lo mismo de derecha a izquierda y así sucesivamente.





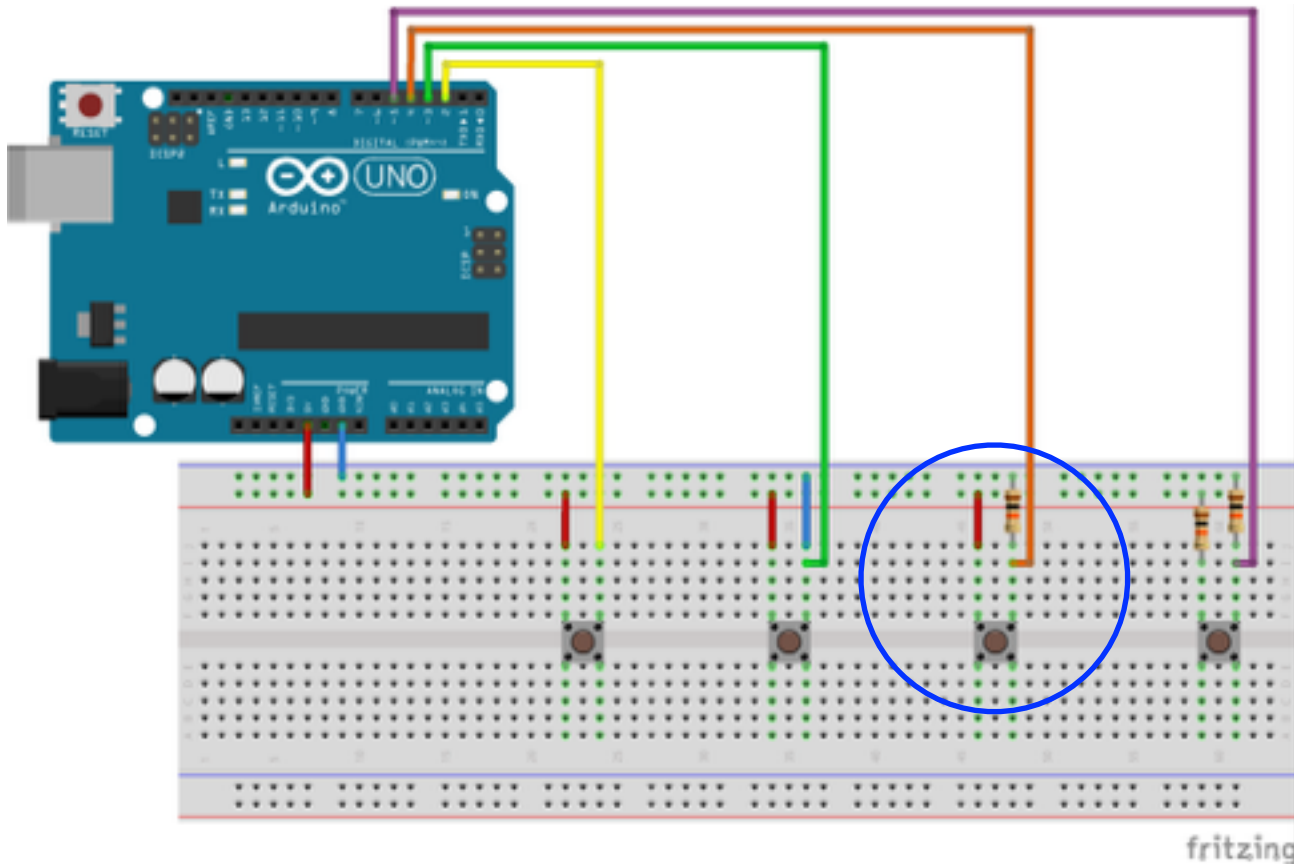
2.3 Entradas digitales

¿Cómo se conecta un Push-button?



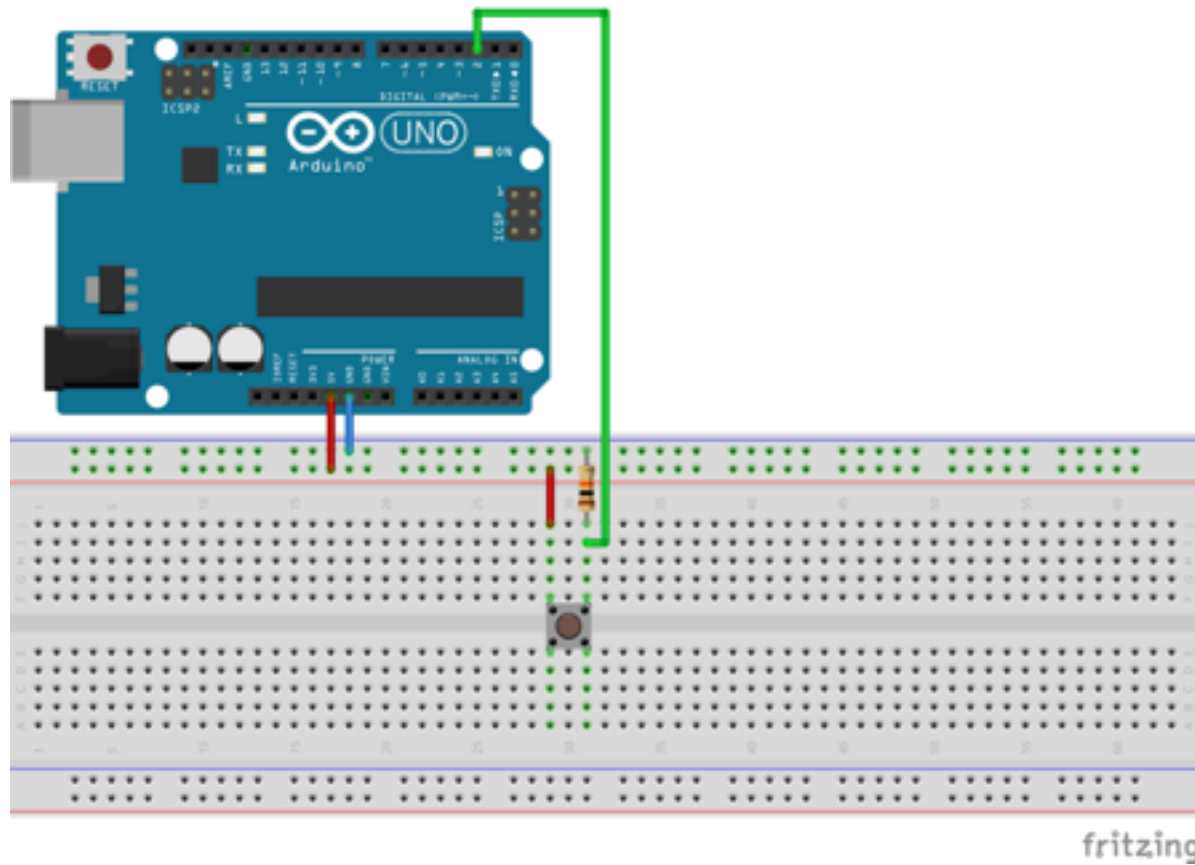
2.3 Entradas digitales

¿Cómo se conecta un Push-button?



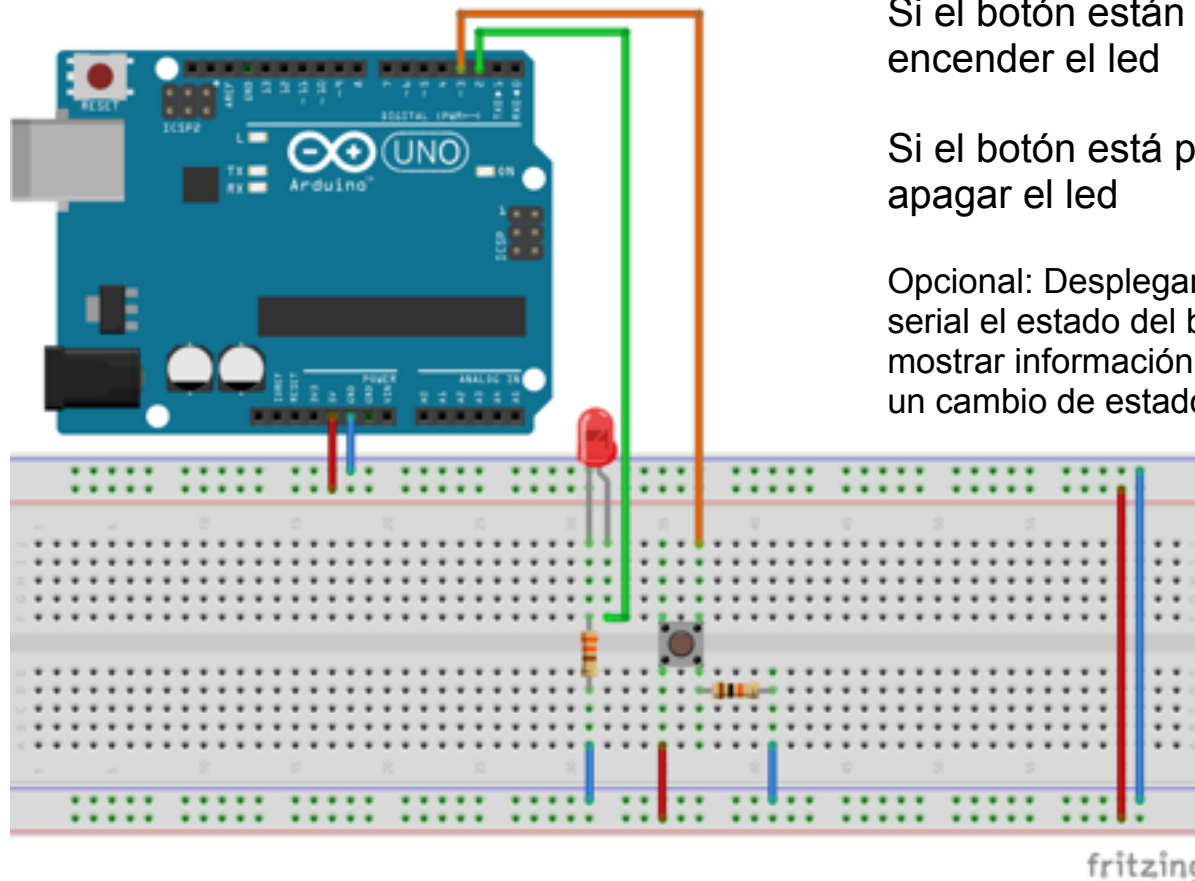
2.3 Entradas digitales

Circuito 3: Visualizar en el monitor serial el estado de un botón



2.3 Entradas digitales

Circuito 4: Controlar el encendido de un led con un botón



Si el botón están sin presionar,
encender el led

Si el botón está presionado,
apagar el led

Opcional: Desplegar en el monitor
serial el estado del botón, pero sólo
mostrar información cada que ocurra
un cambio de estado



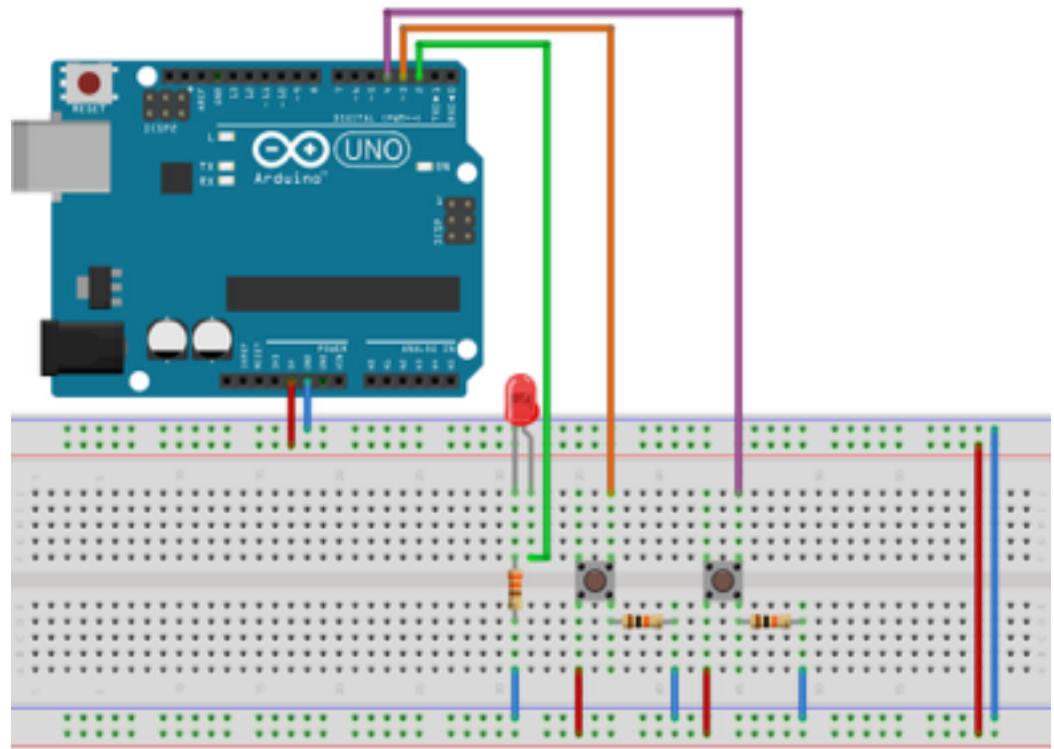
2.3 Entradas digitales

Circuito 5: Controlar el encendido de un led con dos botones

Si los dos botones están sin presionar, apagar el led

Si hay un botón presionado, encender el led

Si los dos botones están presionados, apagar el led



3 Los puertos analógicos de la tarjeta Arduino UNO

6 entradas analógicas



Funciones Analog I/O

`analogReference(tipo)`

`analogRead(pin)`

`analogWrite(pin, valor)`

PWM: Modulación por ancho de pulso

~ Salidas de PWM

3 Los puertos analógicos de la tarjeta Arduino UNO

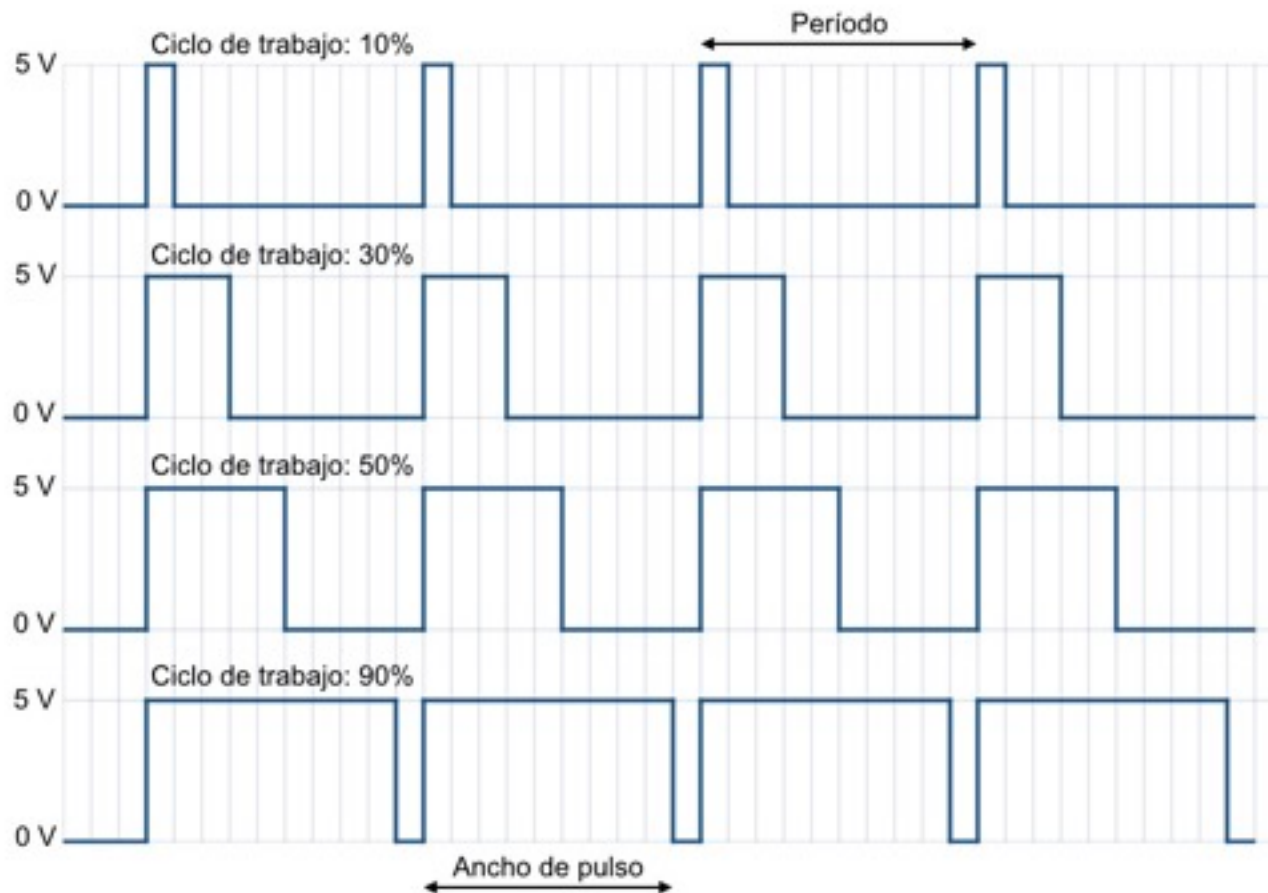
Funciones

Analog I/O

`map(value, fromLow, fromHigh, toLow, toHigh)`



3.1 Modulación por ancho de pulso





3.2 Entradas analógicas

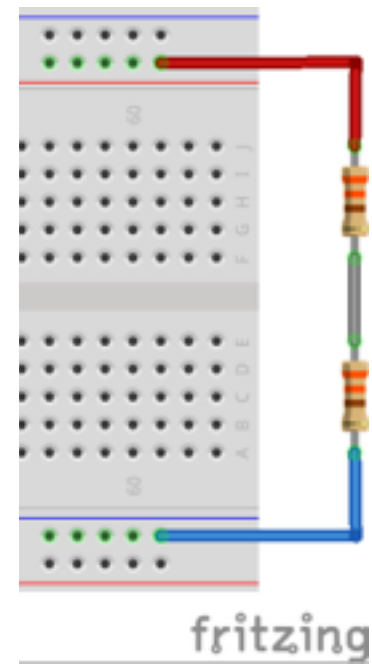
El divisor de voltaje

El voltaje total es la suma de los voltajes

$$V_T = V_1 + V_2$$

Ley de Ohm:

$$V = R \cdot i$$



3.2 Entradas analógicas

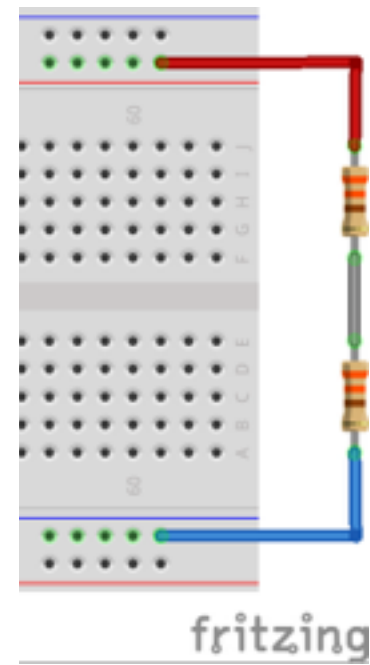
El divisor de voltaje

El voltaje total puede reescribirse como:

$$R_T \cdot i_T = R_1 \cdot i_1 + R_2 \cdot i_2$$

Ya que los componentes están conectados en serie:

$$i_T = i_1 = i_2$$



3.2 Entradas analógicas

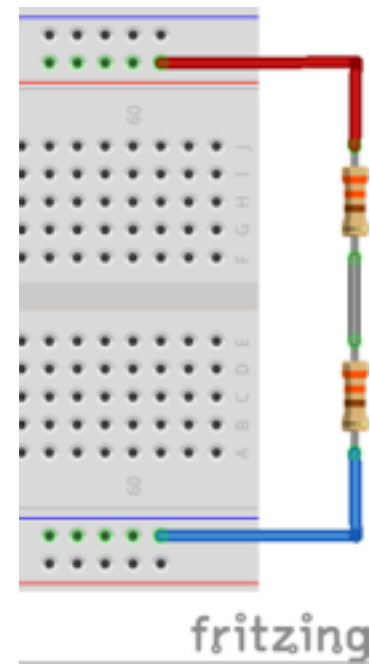
El divisor de voltaje

El voltaje total puede reescribirse como:

$$R_T \cdot i = R_1 \cdot i + R_2 \cdot i$$

$$R_T = R_1 + R_2$$

En conexiones en serie la resistencia equivalente es la suma de las resistencias.





3.2 Entradas analógicas

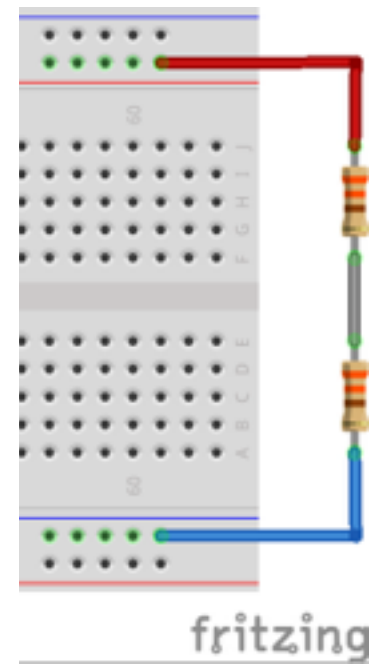
El divisor de voltaje

Al aplicar la ley de Ohm a cada parte del circuito, se tiene:

$$V_T = R_T + i$$

$$V_1 = R_1 + i$$

$$V_2 = R_2 + i$$



3.2 Entradas analógicas

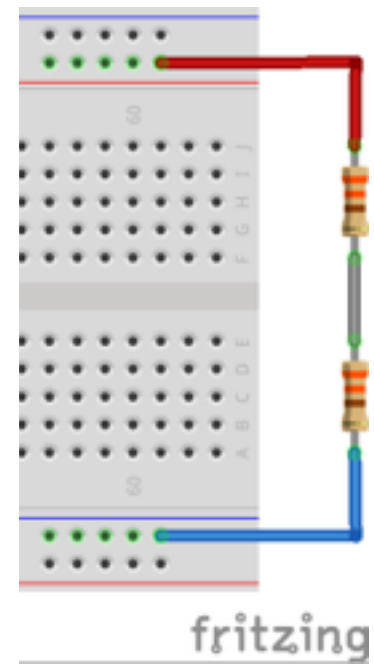
El divisor de voltaje

Al despejar la corriente, se tiene:

$$i = \frac{V_T}{R_T}$$

$$i = \frac{V_1}{R_1}$$

$$i = \frac{V_2}{R_2}$$

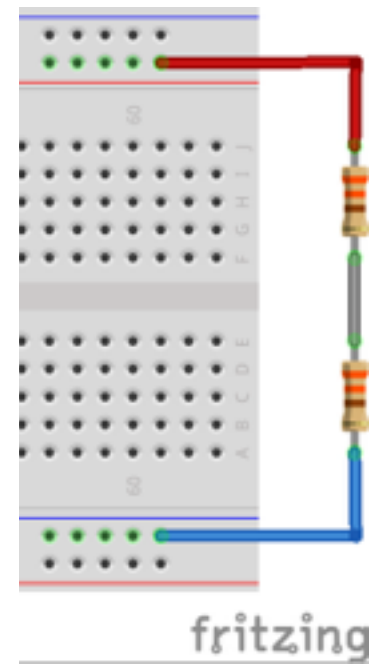


3.2 Entradas analógicas

El divisor de voltaje

$$\frac{V_T}{R_T} = \frac{V_1}{R_1} = \frac{V_2}{R_2}$$

$$\frac{V_T}{R_1 + R_2} = \frac{V_1}{R_1} = \frac{V_2}{R_2}$$

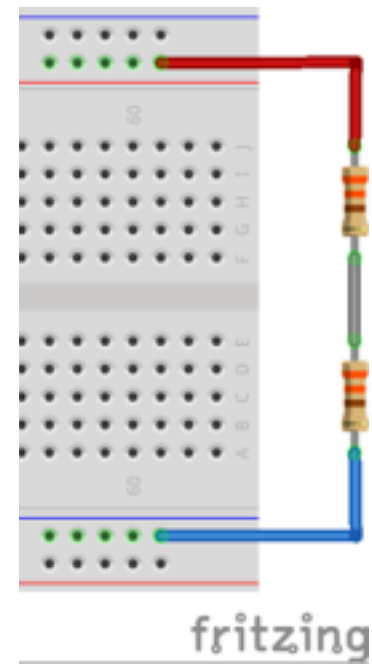


3.2 Entradas analógicas

El divisor de voltaje

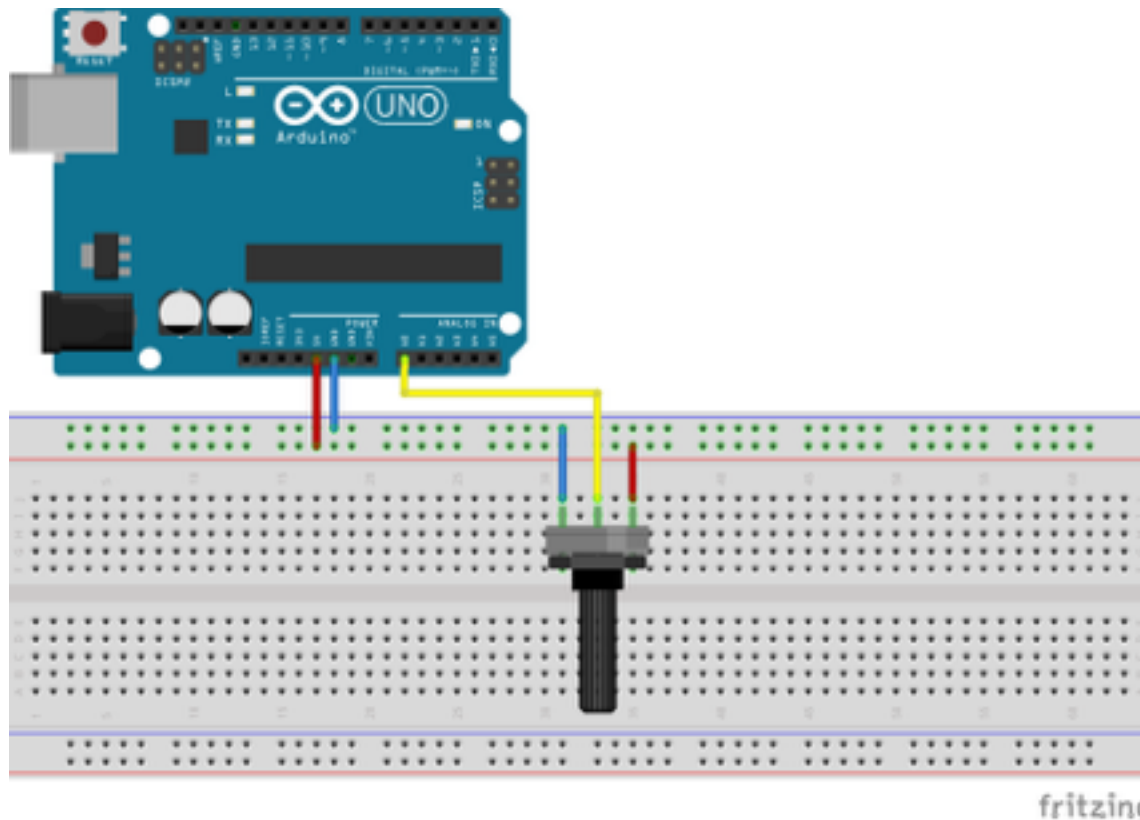
$$\frac{V_1}{R_1} = \frac{V_T}{R_1 + R_2} \longrightarrow V_1 = \frac{R_1}{R_1 + R_2} \cdot V_T$$

$$\frac{V_2}{R_2} = \frac{V_T}{R_1 + R_2} \longrightarrow V_2 = \frac{R_2}{R_1 + R_2} \cdot V_T$$



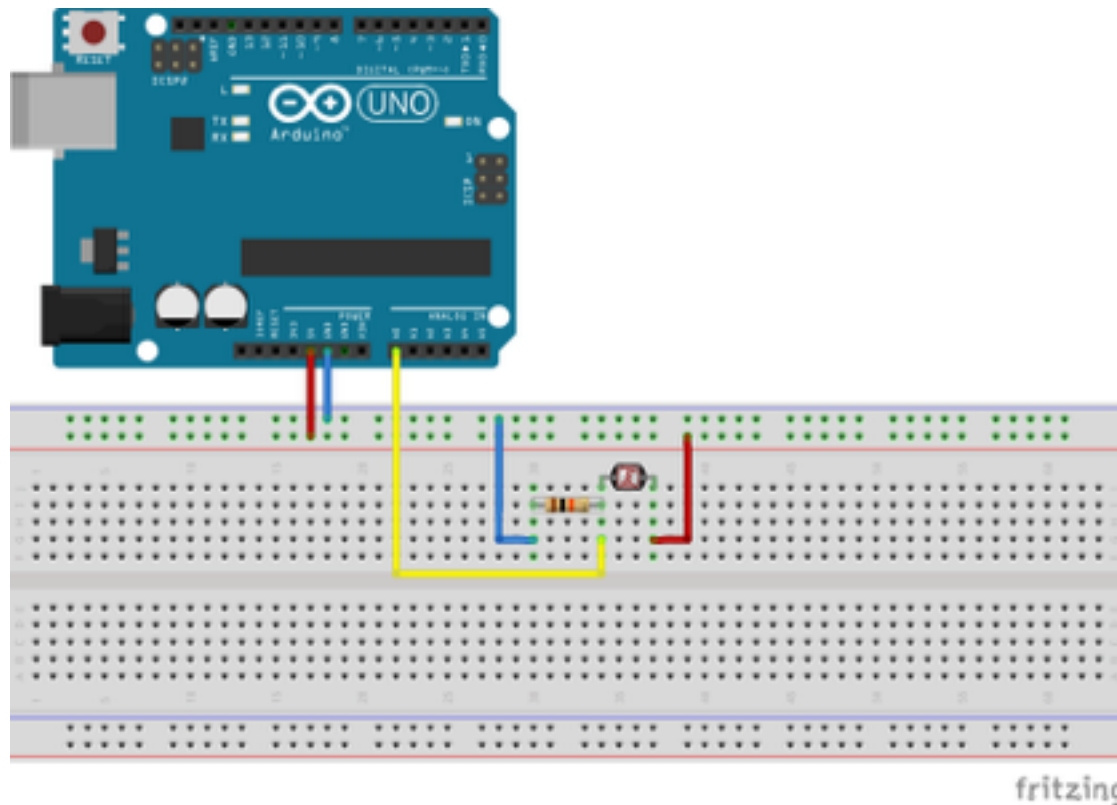
3.2 Entradas analógicas

Circuito 6: Mostrar en el monitor serial la lectura de un pin de entrada analógica conectado a un potenciómetro



3.2 Entradas analógicas

Circuito 7: Mostrar en el monitor serial la lectura de un pin de entrada analógica conectado a una fotoresistencia

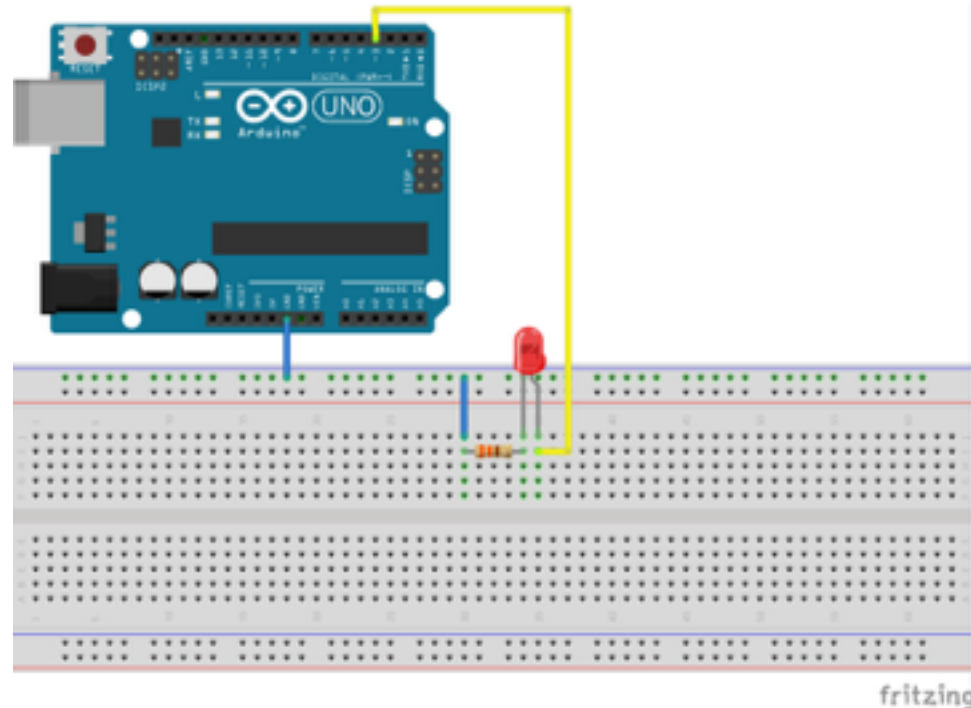


3.3 Salidas analógicas

Circuito 8: Controlar el encendido de un led de modo gradual

Hacer que un led encienda gradualmente y se quede encendido

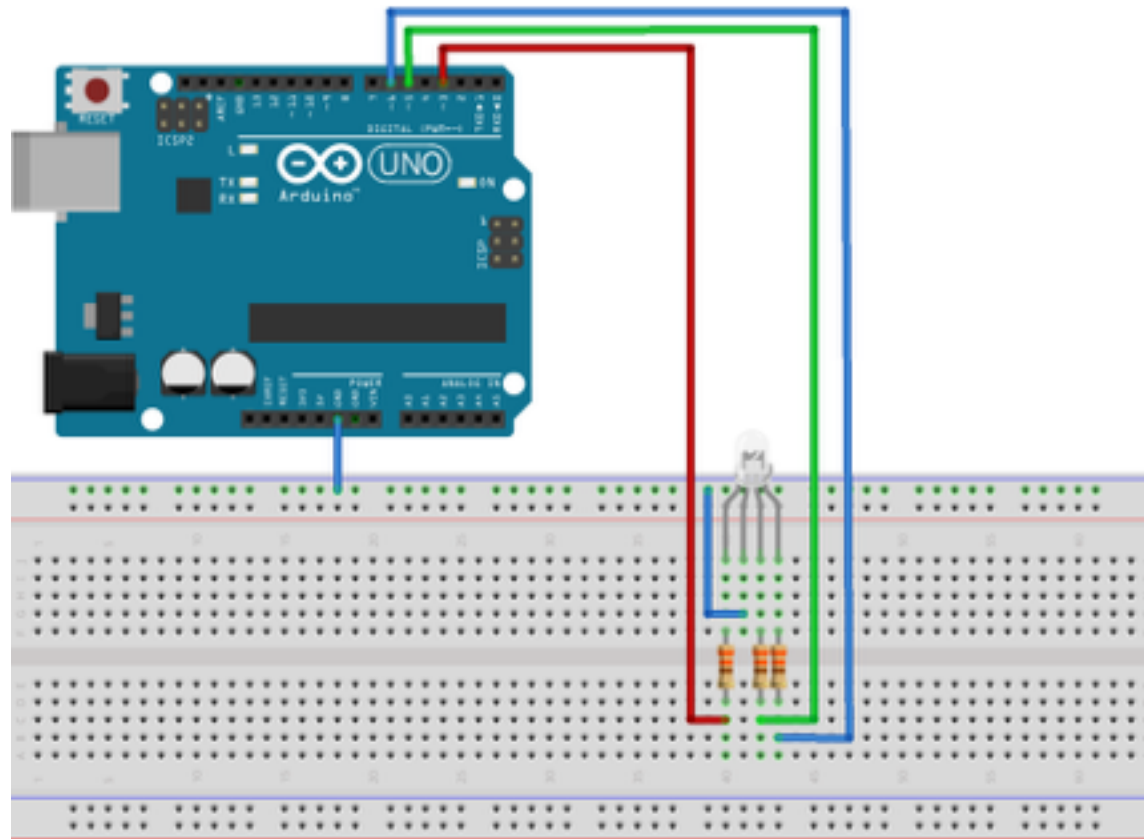
Hacer que un led encienda gradualmente y luego se apague gradualmente





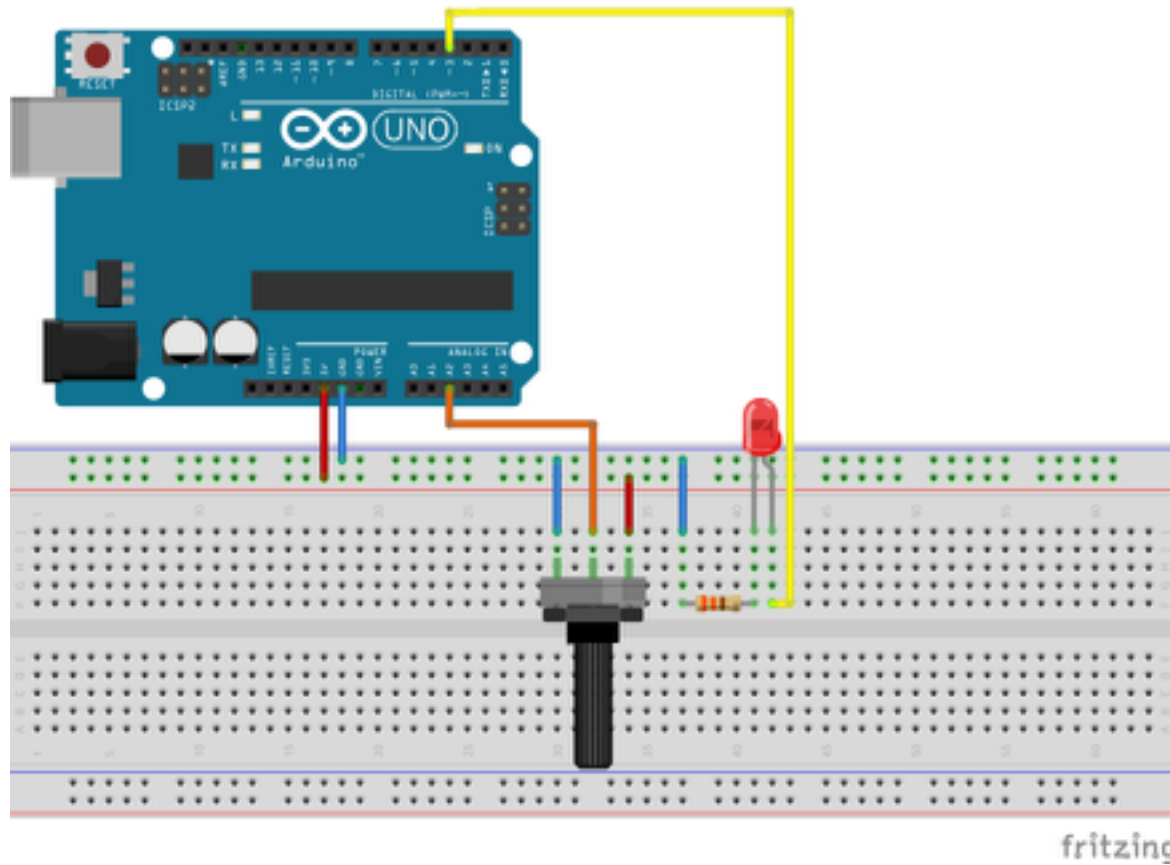
3.3 Salidas analógicas

Circuito 9: Controlar el color con que enciende un led RGB



3.3 Salidas analógicas

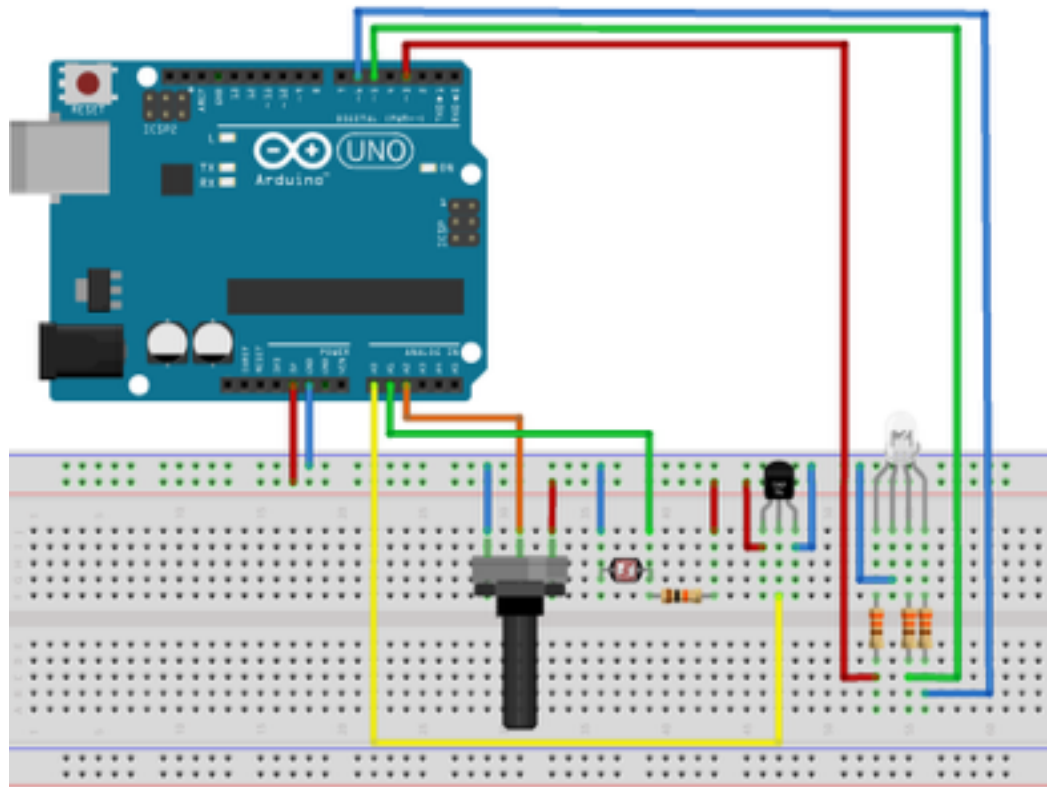
Circuito 10: Controlar el encendido gradual de un led con un potenciómetro





3.3 Salidas analógicas

Circuito 11: Controlar un led RGB con un potenciómetros, una fotoresistencia y un sensor de temperatura



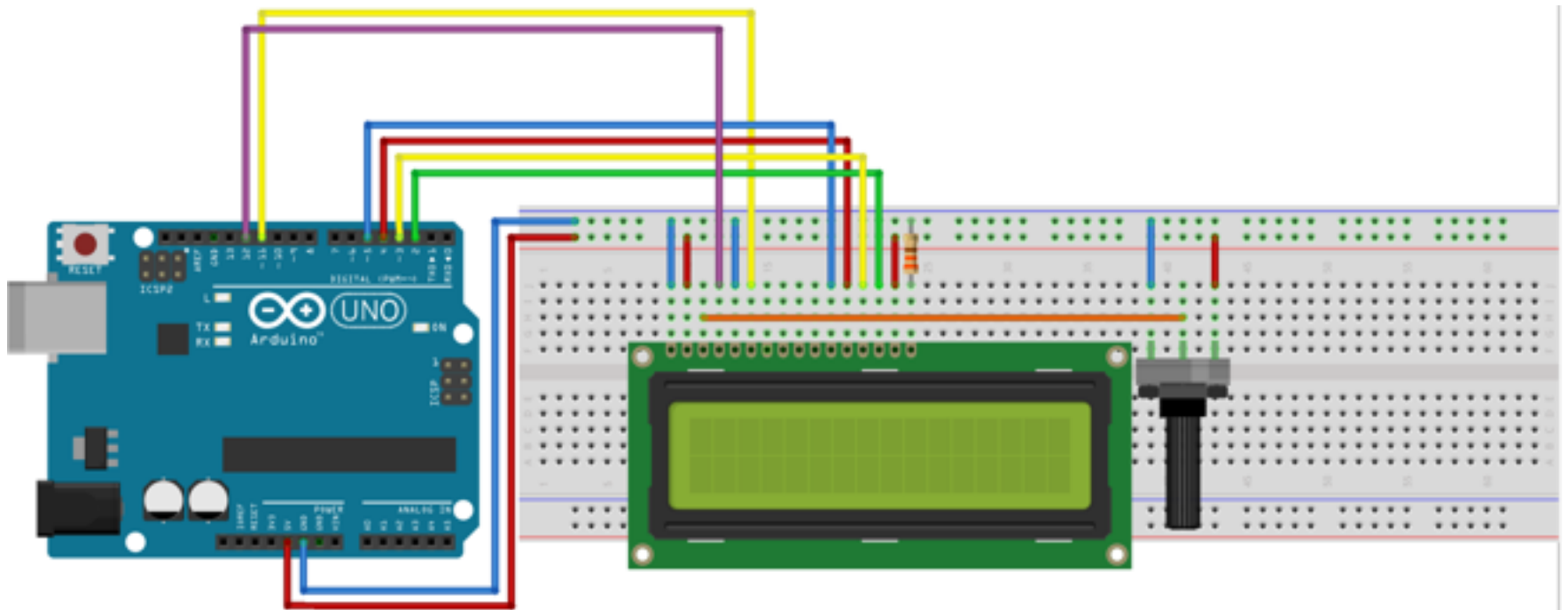


4 Despliegue de información en una pantalla LCD





4.1 Conexión de una pantalla LCD



4.3 Algunas funciones de la pantalla LCD

Funciones

Biblioteca LiquidCrystal

`LiquidCrystal` lcd(rs, habilitación, d4, d5, d6, d7)

`lcd.begin`(columnas, renglones)

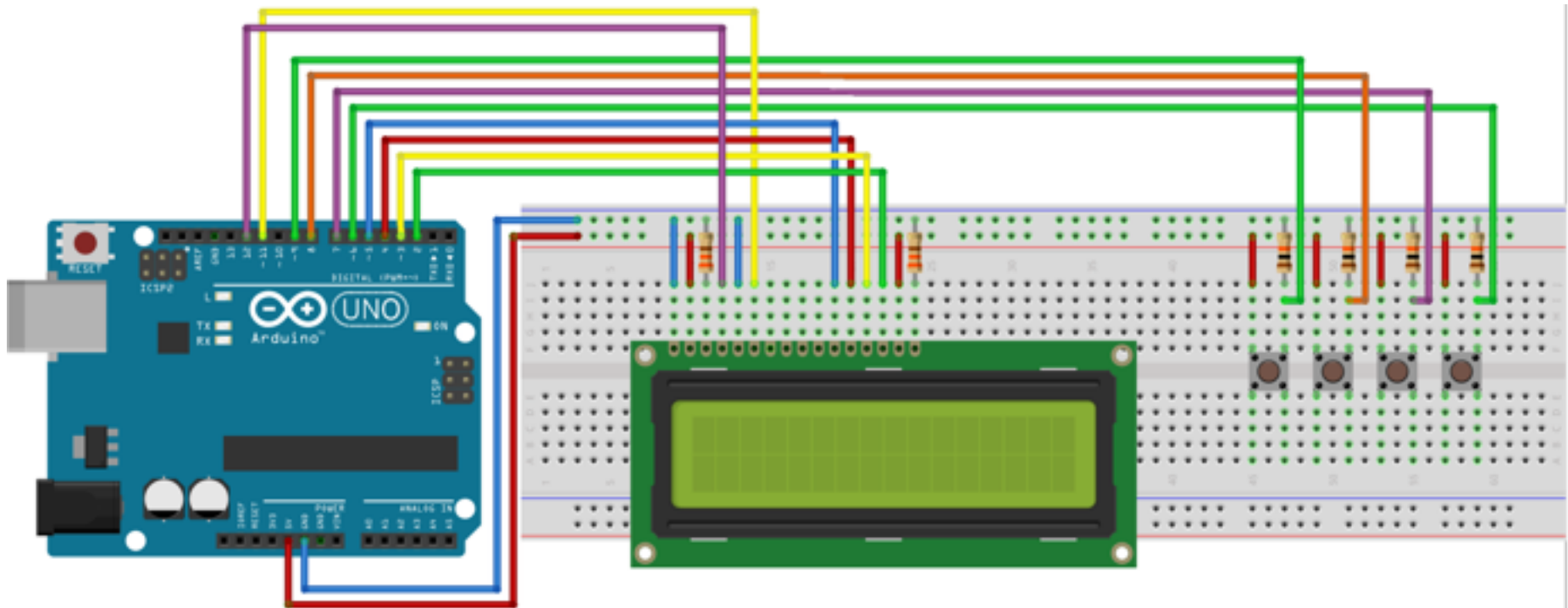
`lcd.print`(dato)

`lcd.setCursor`(columna, renglón)

`lcd.clear`()

4.4 Programación de un menú interactivo básico

Circuito 12: Menú interactivo desplegado en pantalla LCD y controlado con cuatro botones



4.4 Programación de un menú interactivo básico

Estructuras

Estructuras de control

if. Si se cumple la condición, realiza la acción indicada

if...else. Si se cumple la condición realiza la acción A, si no, realiza la acción B

for. Repite un conjunto de instrucciones hasta que un contador llegue al valor deseado

while. Realiza la acción mientras se cumpla la condición

do... while. Realiza la acción, si se cumple la condición realiza nuevamente

4.4 Programación de un menú interactivo básico

Circuito 12:

Pantalla 1:

Selecciona:

1. Coca 2. Pepsi

Si se presiona el botón 1:

Pantalla 2:

Coca: \$15.00

Pagado: \$0.00

(Se incrementa al presionar el botón 3)

Si se presiona el botón 2:

Pantalla 2:

Pepsi: \$14.00

Pagado: \$0.00

(Se incrementa al presionar el botón 3)

Si se presiona el botón 4:

Pantalla 3:

Cancelado, tome su dinero

Si se termina de pagar:

Pantalla 3:

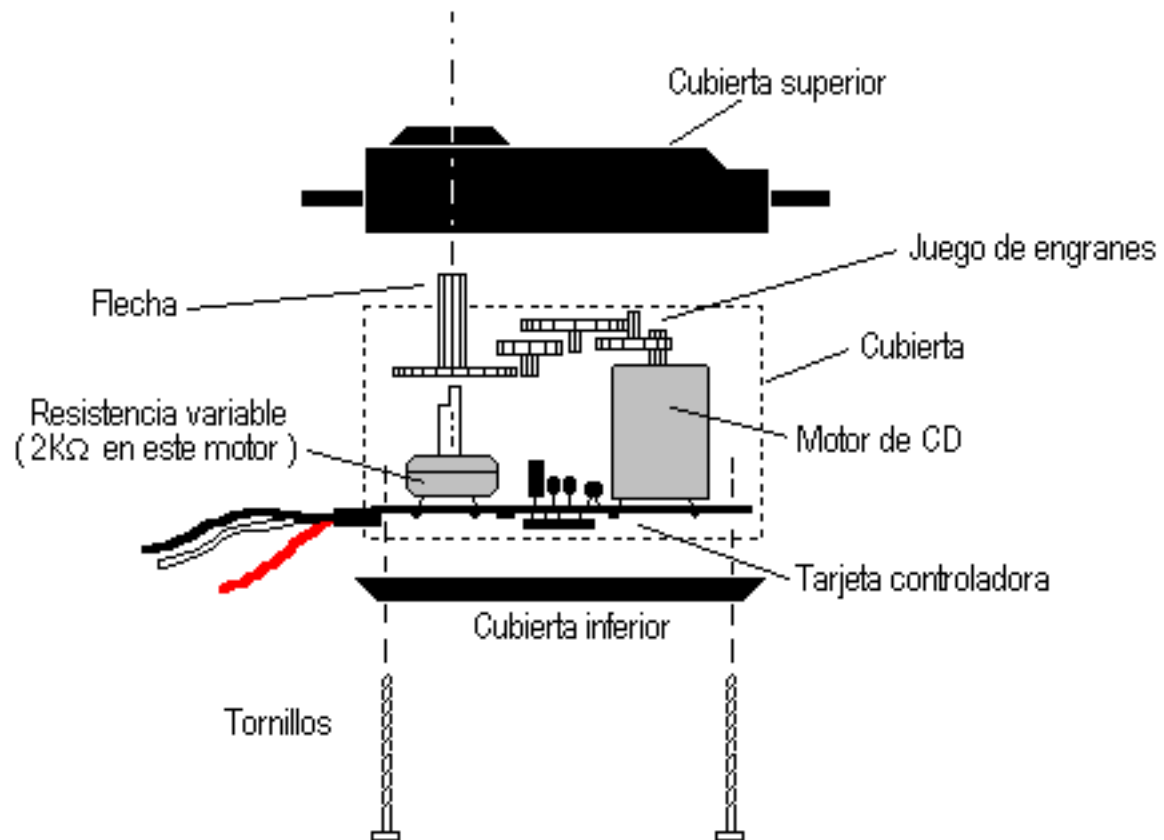
Gracias! Tome su bebida

5 Manejo básico de actuadores

- Motores de rotación continua
- Motores de Pulsos (Motores paso a paso)
- Servomotores

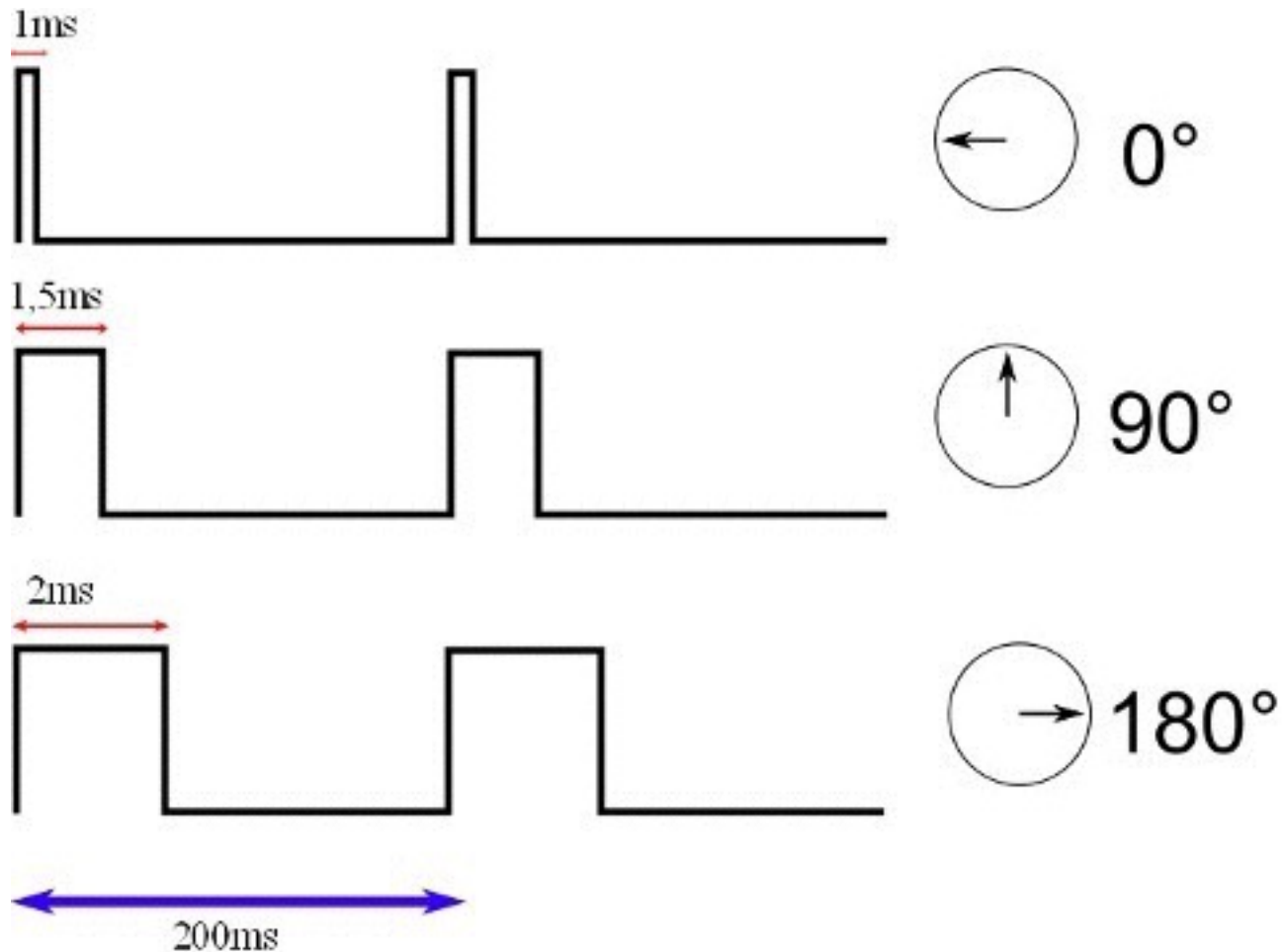
5.1 Servomotores

- Motor de CD
- Control de lazo cerrado
- Tres cables



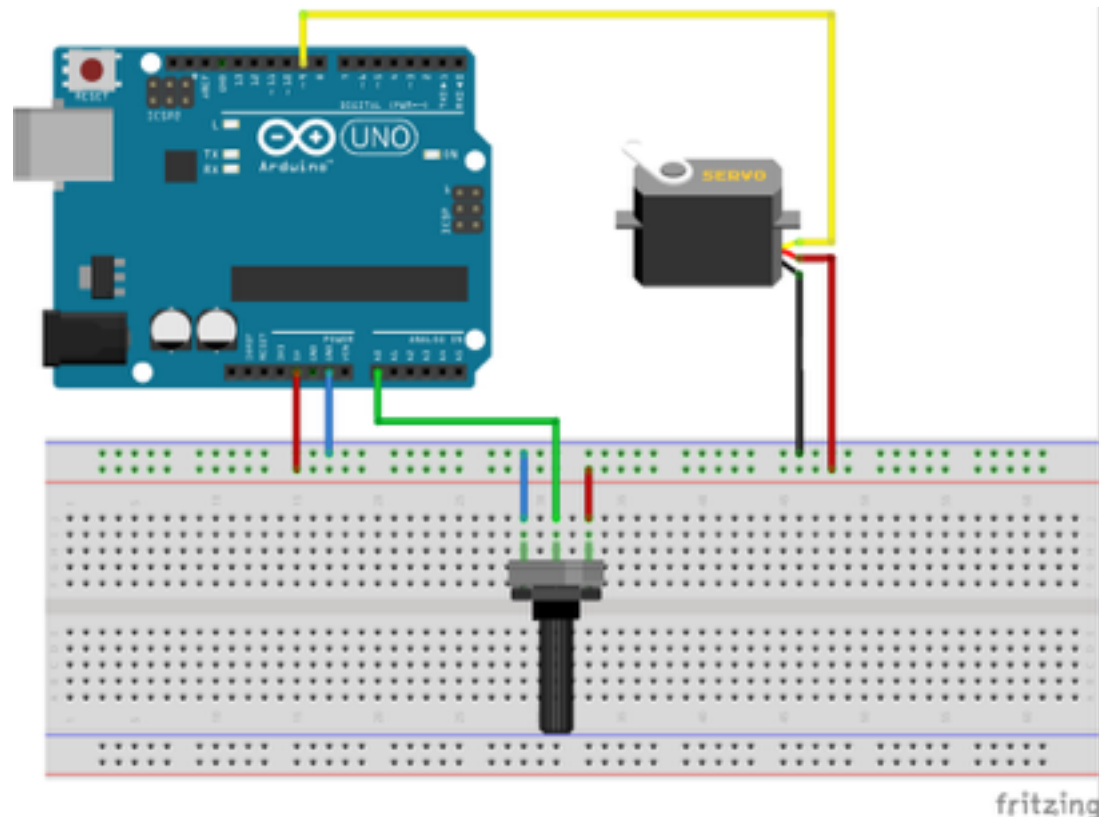


5.1 Servomotores



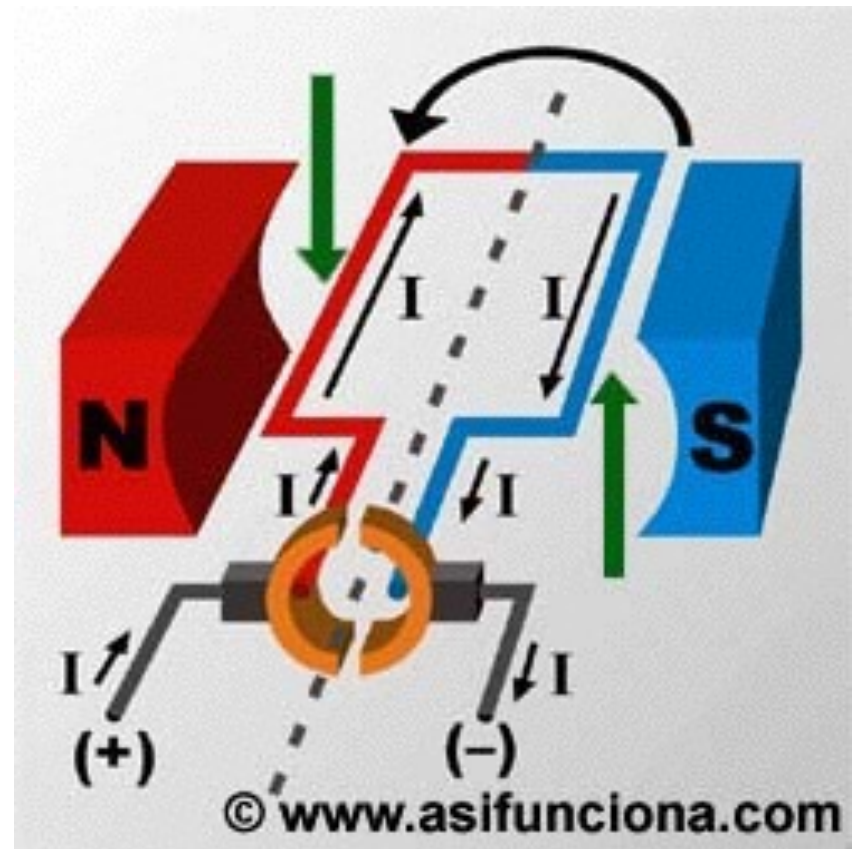
5.1 Servomotores

Circuito 13: Controlar la posición del eje de un servomotor con un potenciómetro



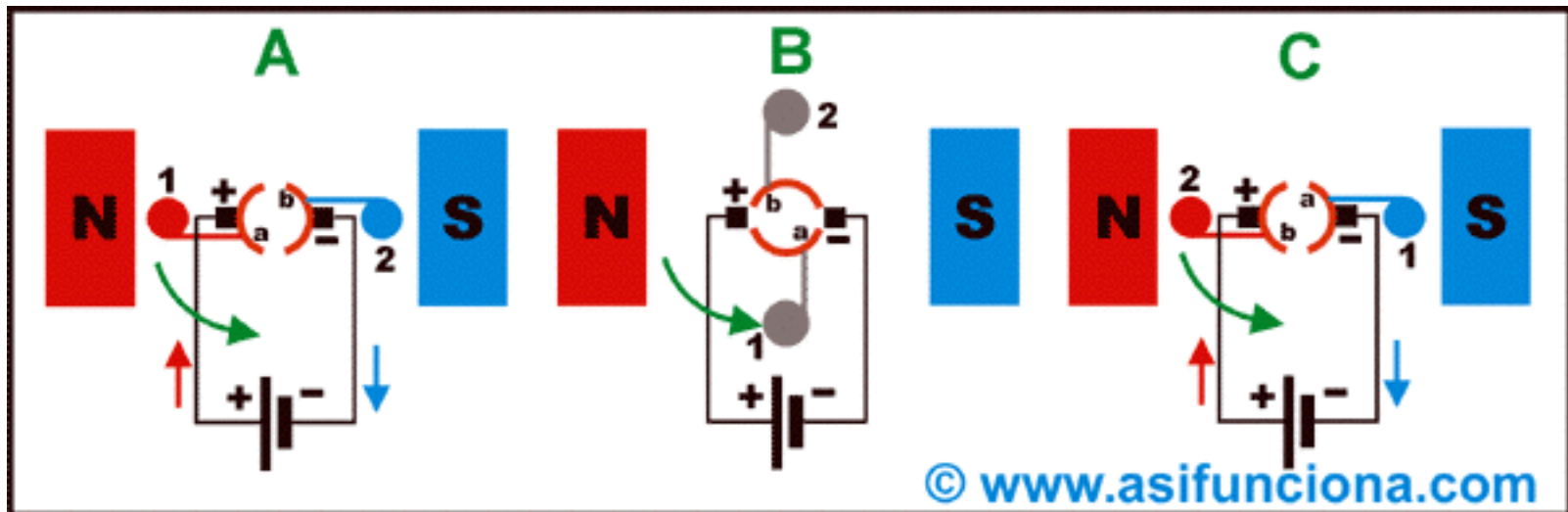
5.2 Motores de rotación continua

- Corriente eléctrica
- Campo magnético
- Fuerza resultante



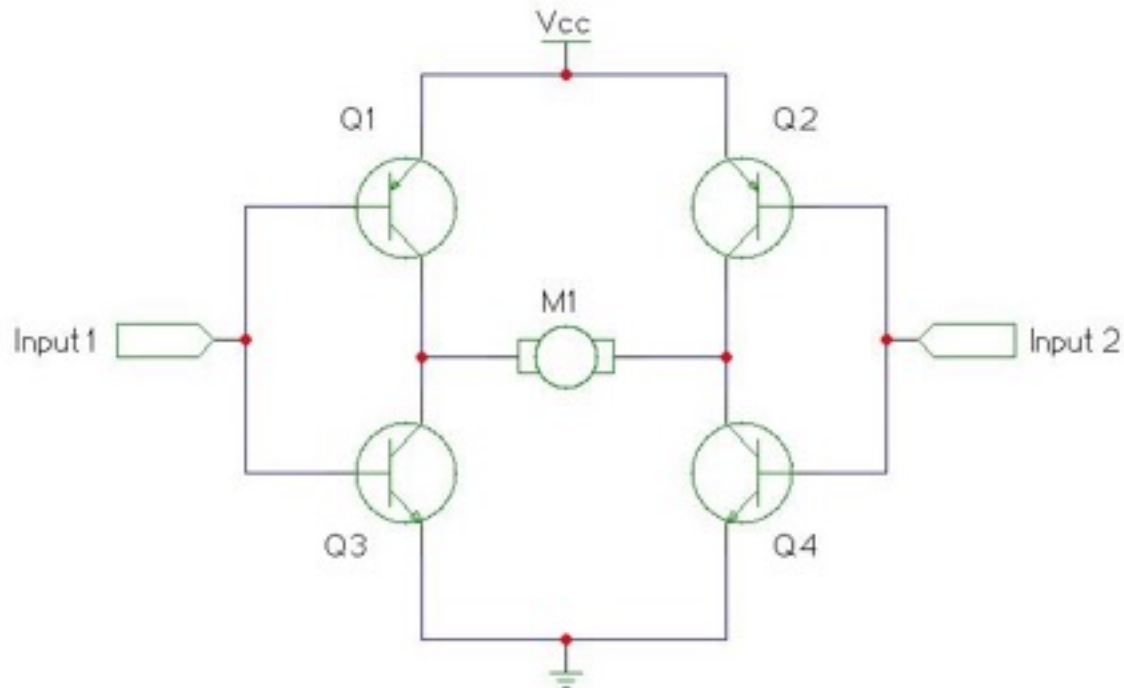
5.2 Motores de rotación continua

- El sentido de giro depende del sentido de la corriente





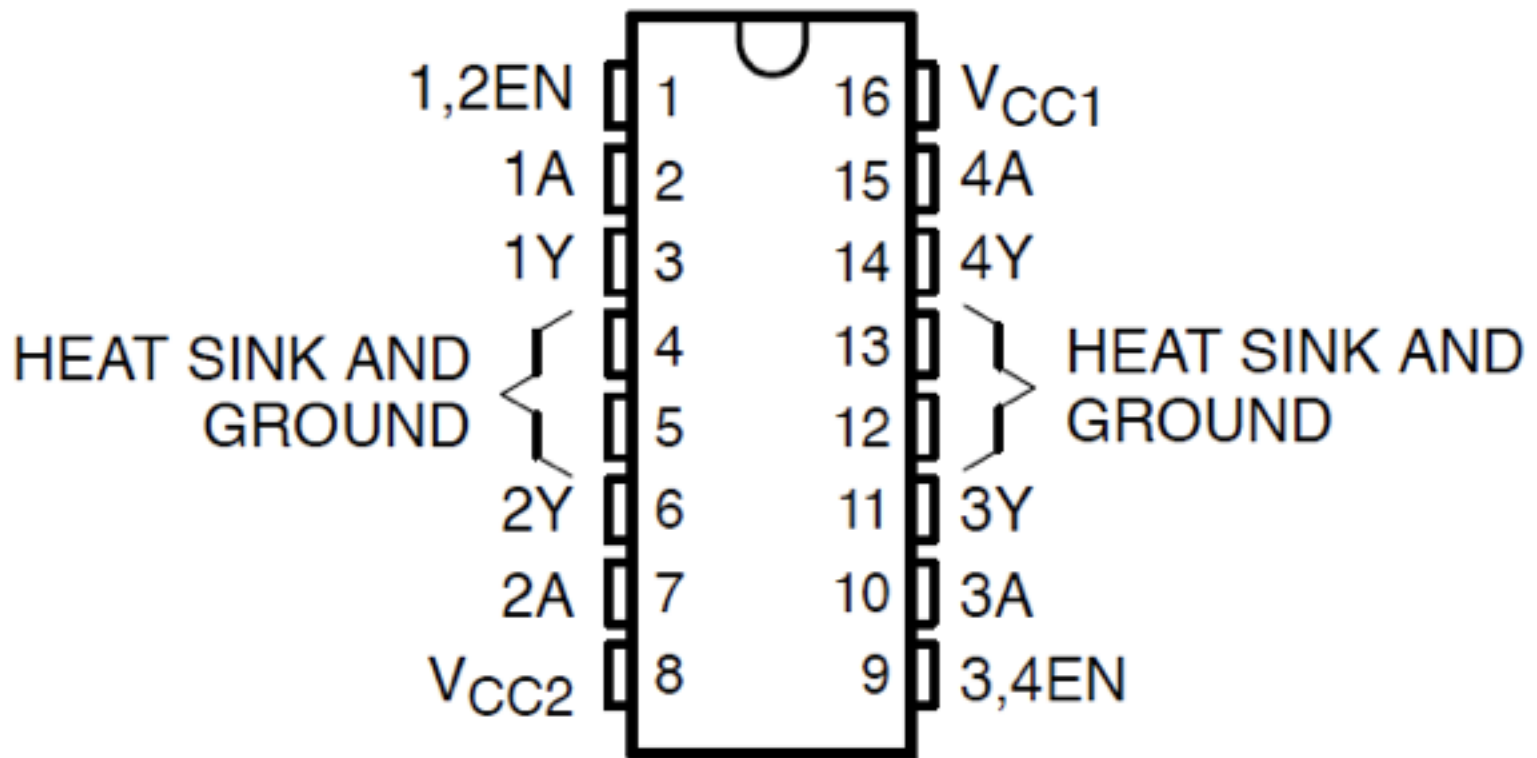
5.3 Conexión de un Puente H





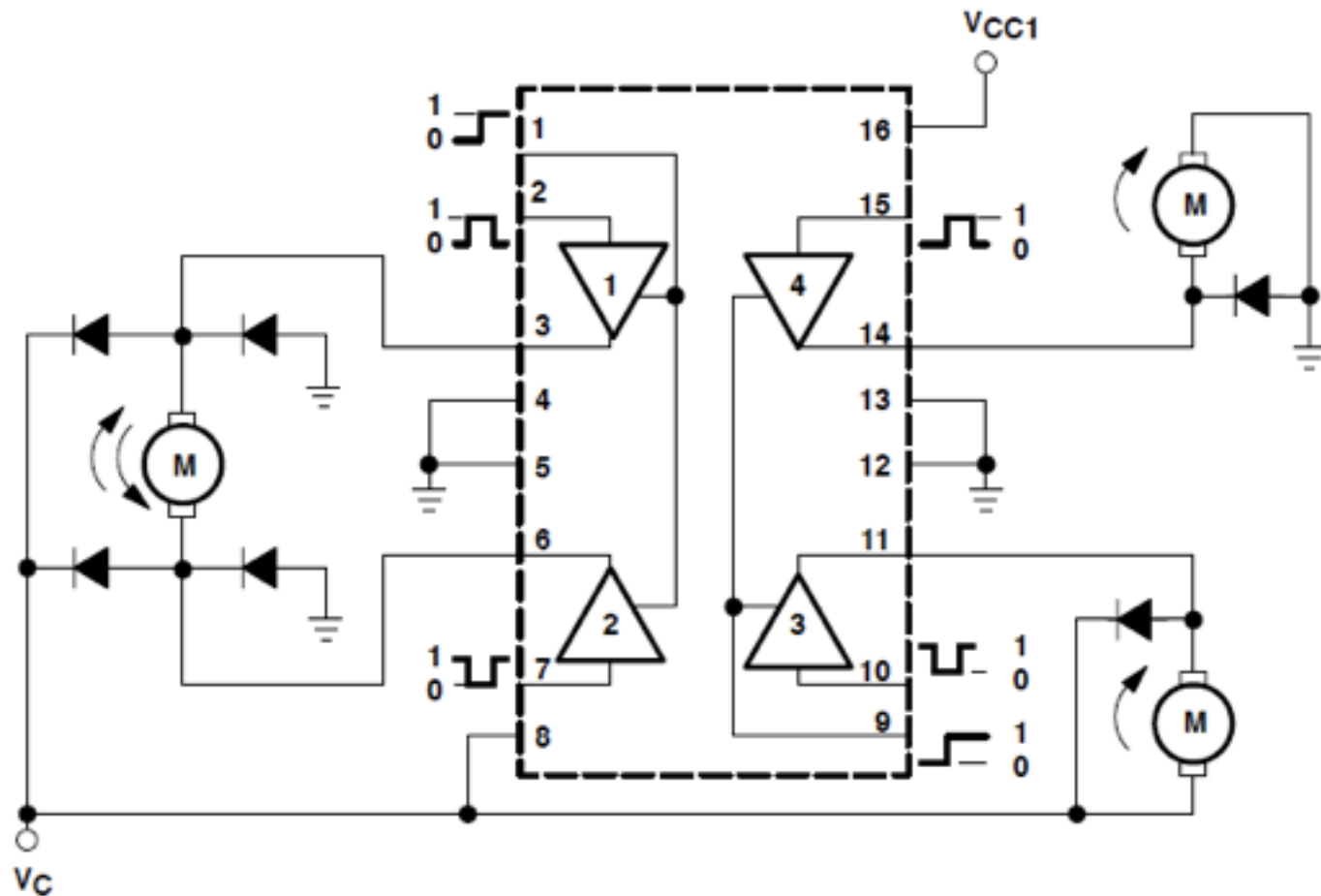
5.3 Conexión de un Puente H

**N, NE PACKAGE
(TOP VIEW)**





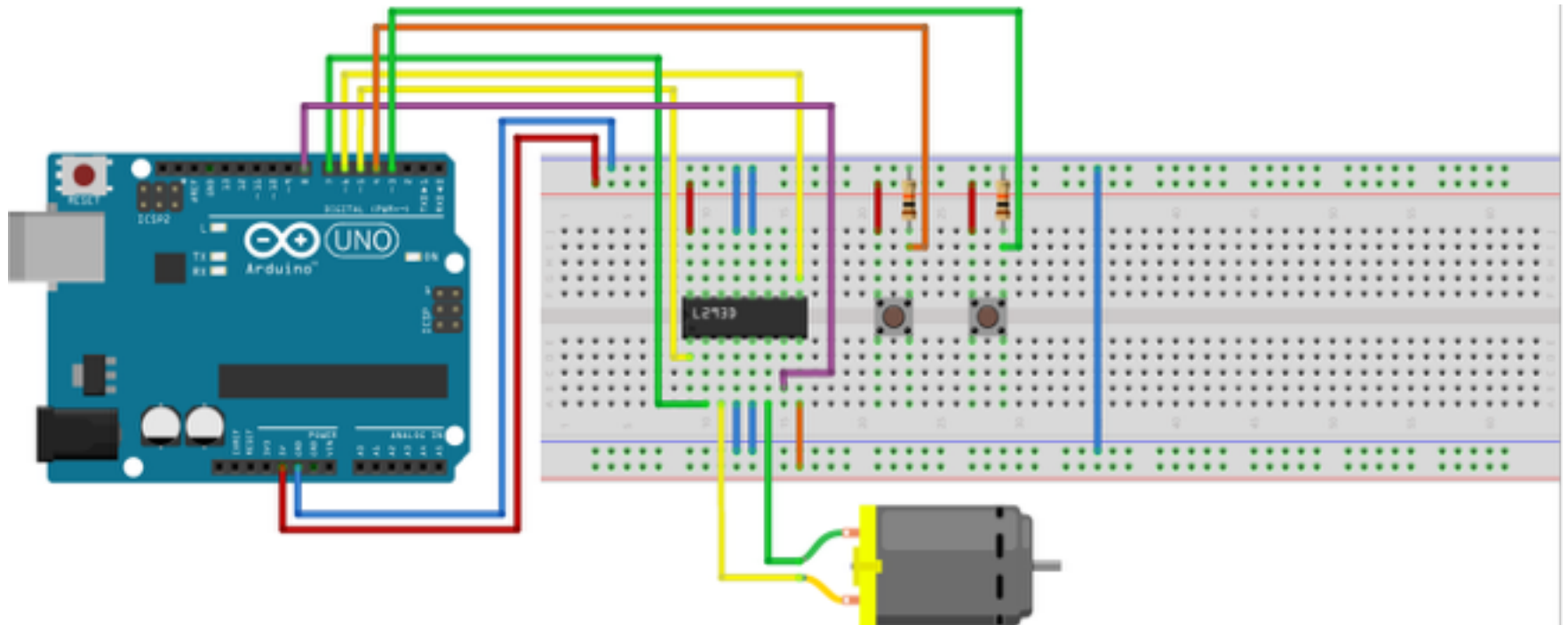
5.3 Conexión de un Puente H





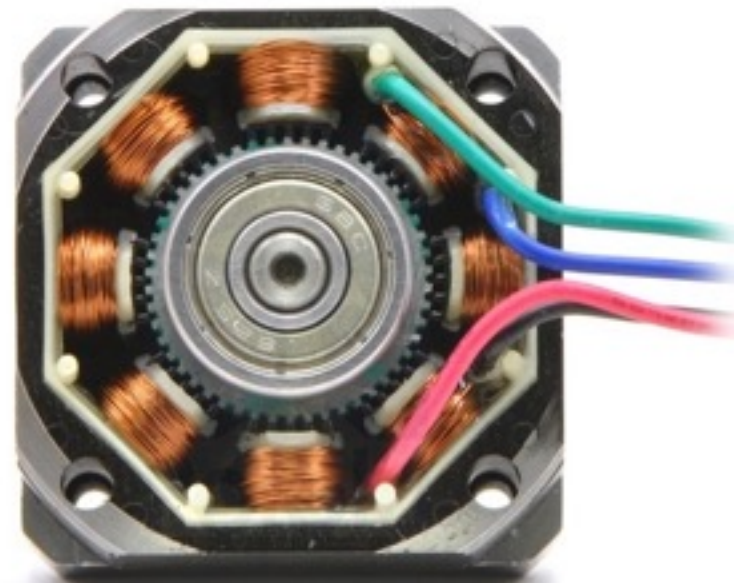
5.3 Conexión de un Puente H

Circuito 14: Controlar el encendido y el sentido de giro de un motor DC



5.4 Motores de Pulsos (paso a paso)

- Movimiento discreto
- Número de polos
- Unipolar o bipolar





5.4 Motores de Pulsos

Tipos de control:

- Paso simple

Paso	A	B	C	D	
1	1	0	0	0	
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	



5.4 Motores de Pulsos

Tipos de control:

- Paso doble

Paso	A	B	C	D	
1	1	1	0	0	
2	0	1	1	0	
3	0	0	1	1	
4	1	0	0	1	



5.4 Motores de Pulsos

Tipos de control:

- Medio paso

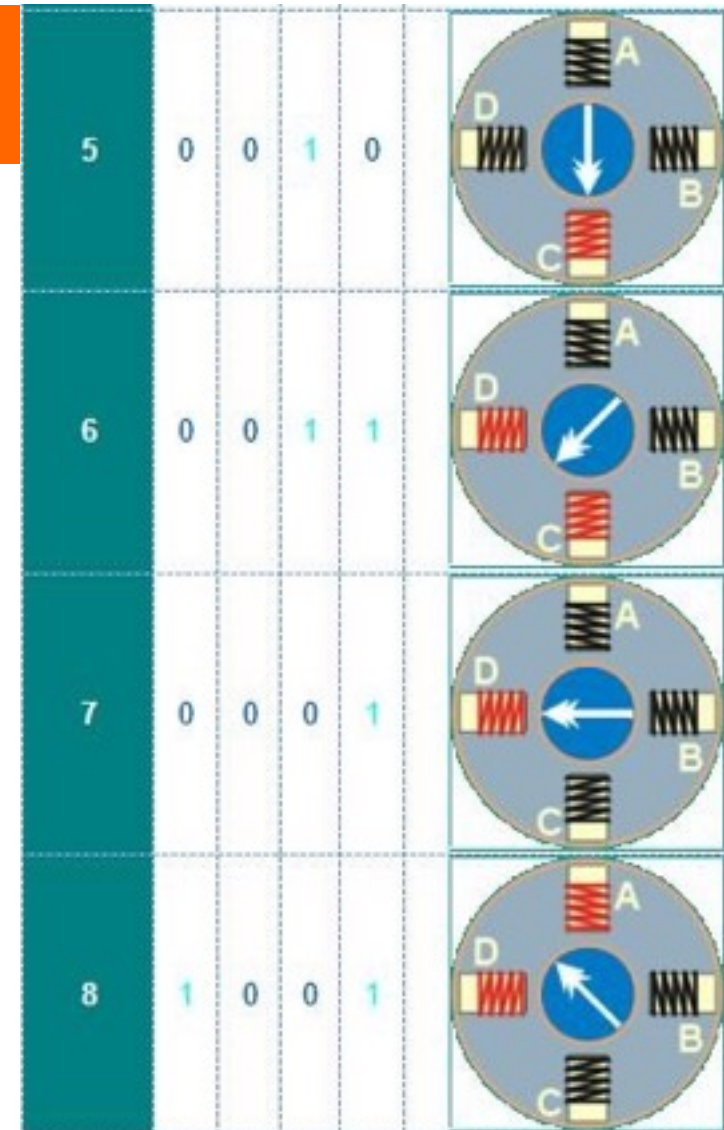
Paso	A	B	C	D	
1	1	0	0	0	
2	1	1	0	0	
3	0	1	0	0	
4	0	1	1	0	



5.4 Motores de Pulsos

Tipos de control:

- Medio paso



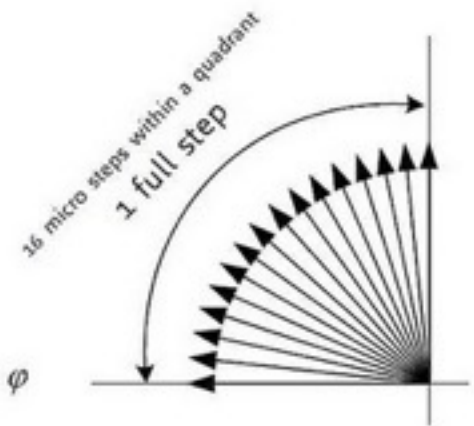
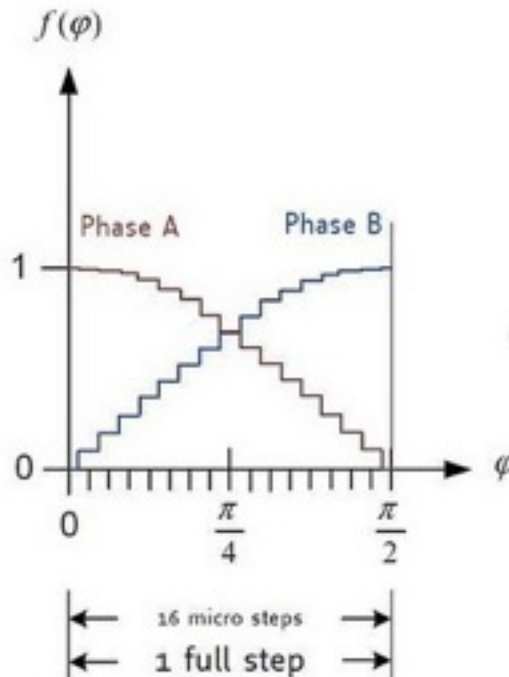


5.4 Motores de Pulsos (paso a paso)

Tipos de control:

- Micro-paso

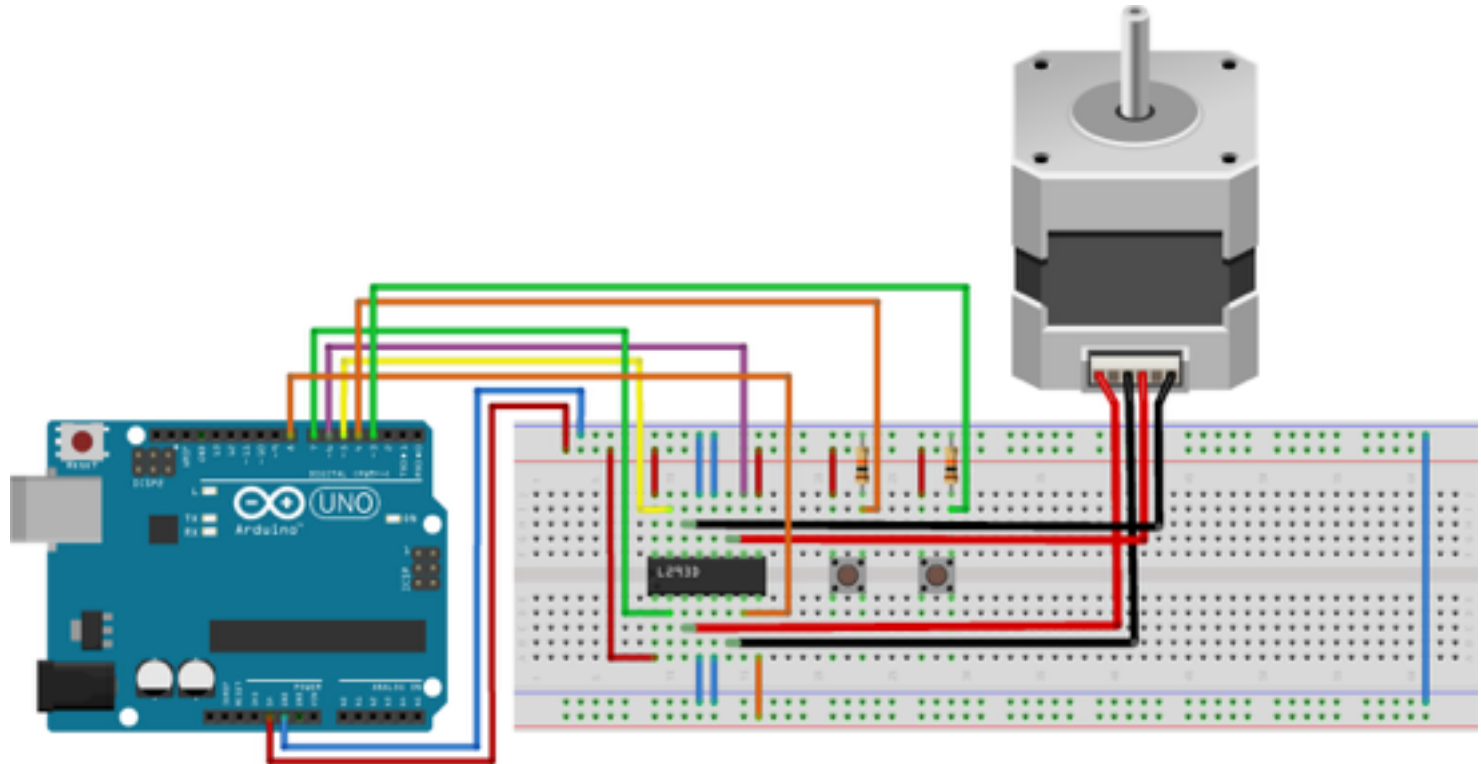
MICROSTEP OPERATION



sinewave and cosinewave
with 16 analog values

5.4 Motores de Pulsos (paso a paso)

Circuito 15: Controlar el encendido y el sentido de giro de un motor de pulsos



330ohms®

MAKERSPACE

