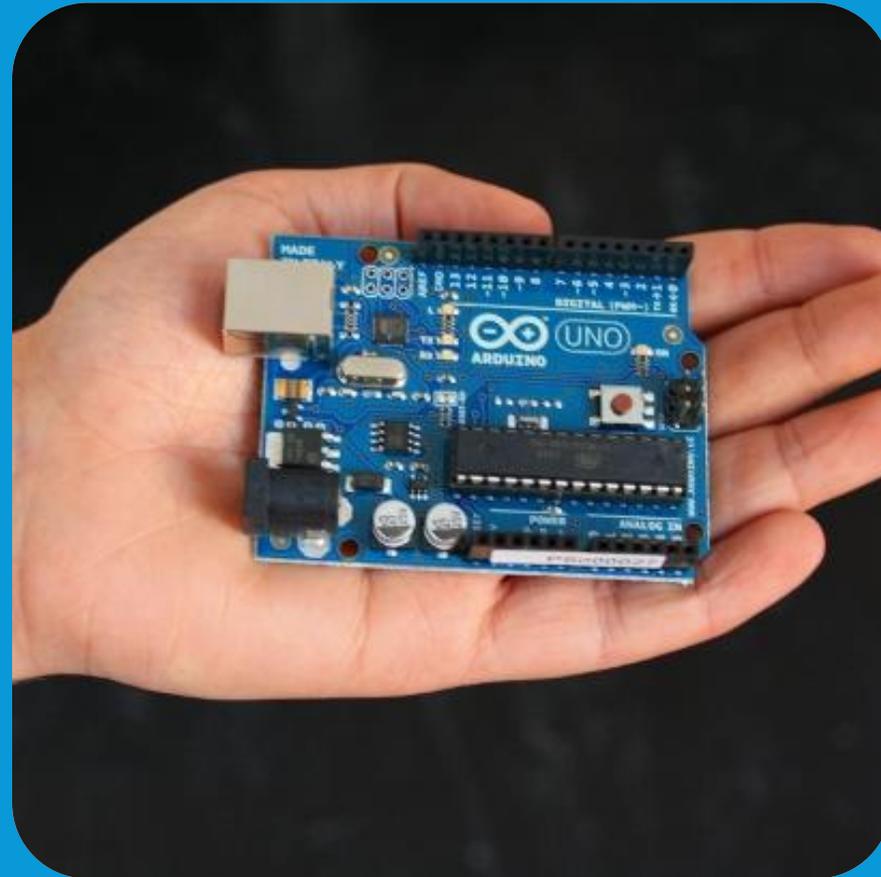


TALLER ARDUINO II



ARDUINO UNO

Resumen de características

| | |
|--|---------------------------------------|
| Microcontrolador | ATMega328 |
| Voltaje al que opera | 5 V |
| Voltaje entrada (recomendado) | 7-12 V |
| Voltaje entrada (límites) | 6-20 V |
| Pines de Entrada/salida digitales | 14 (de los cuales ,6 cuentan con PWM) |
| Pines de Entrada analógica | 6 |
| Corriente DC por pin de E/S | 40 mA |
| Corriente DC para pin de 3.3 V | 50 mA |
| Memoria Flash | 32 KB (ATMega328) |
| SRAM | 2 KB (ATMega328) |
| EEPROM | 1 KB (ATMega328) |
| Velocidad del reloj | 16 MHz |

RECUERDO

ADVERTENCIA

Sólo usar **pin 13** para esta experiencia.
Único pin con RESISTENCIA INTEGRADA.

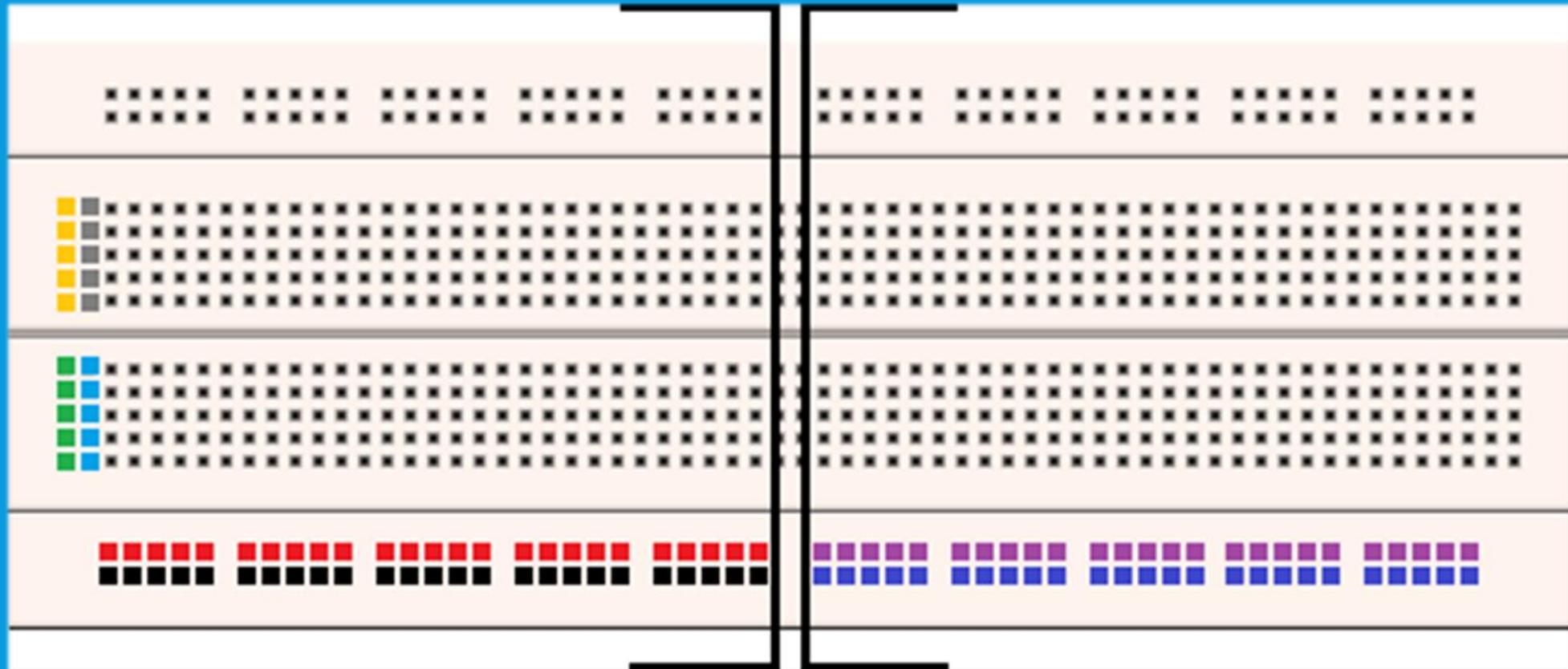
A no ser que quieran freír sus pines y componentes...

Explicación: Ley de OHM

$$I \times R = V$$

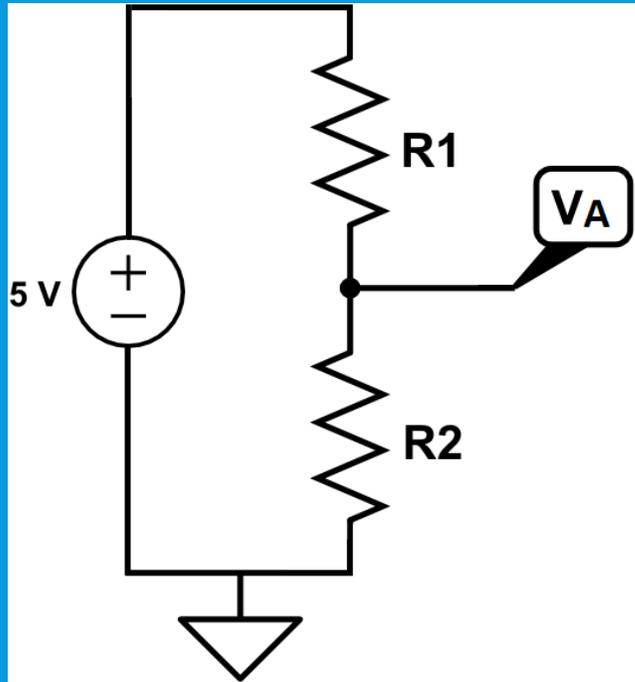


PROTOBOARD



DIVISOR DE VOLTAJE

Divisor de voltaje



$$i = \frac{5[V]}{R_1 + R_2} = \frac{V_A}{R_2} \Rightarrow V_A = \frac{R_2}{R_1 + R_2} 5[V]$$

Este cálculo se obtiene de aplicar ley de OHM en ambas resistencias, y sabiendo que la corriente que circula por ambas es igual.

DIVISOR DE VOLTAJE

Divisor de voltaje

- Valores para R_1 y R_2 definen V_a ...

Fijando una resistencia y variando la otra, voltaje V_a varía.

- Arduino puede leer voltajes de 0 a 5 volts desde sus *pin*s analógicos.
- Los interpreta como valores de 0 a 1023 respectivamente.

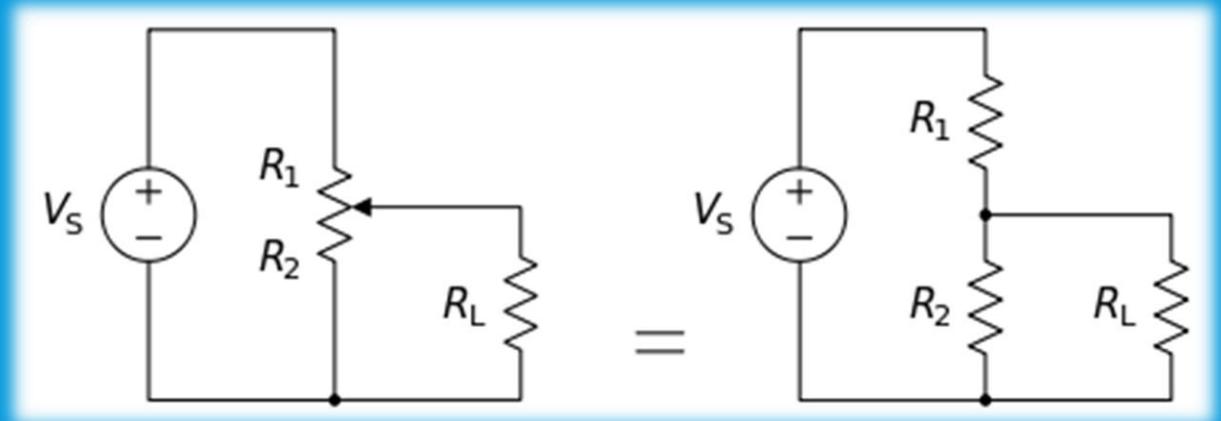
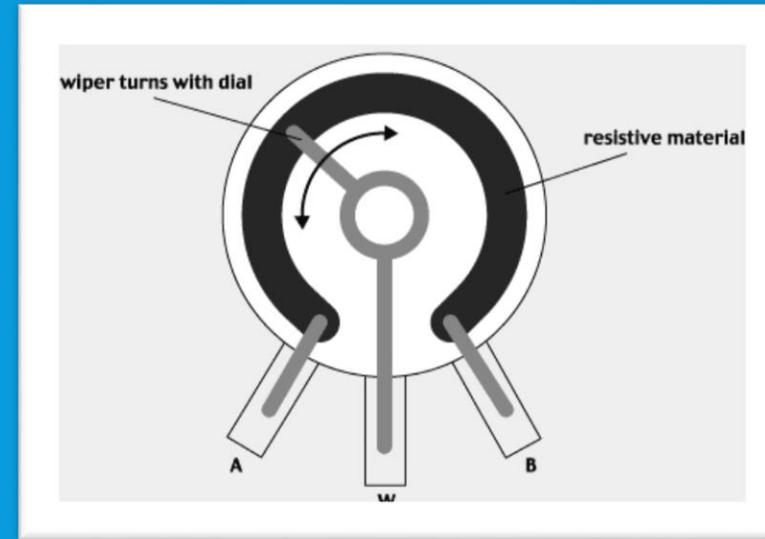
¿Cómo variar una resistencia?

¡Hay muchos tipos de resistencias variables!

POTENCIÓMETRO

Divisor de voltaje

Ejemplo: **Potenciómetro**



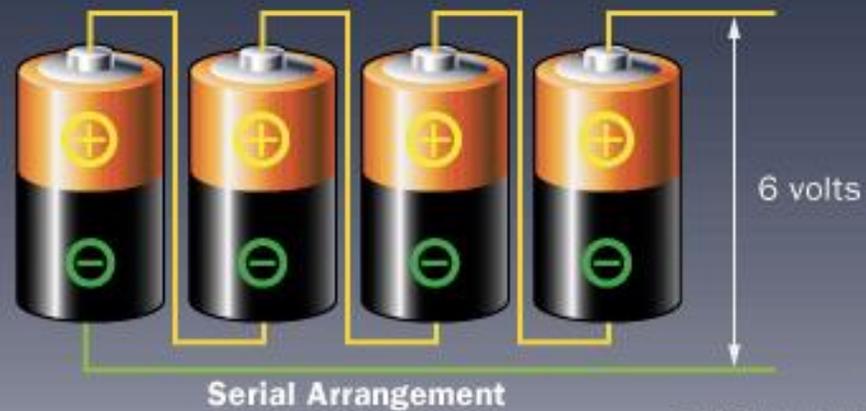
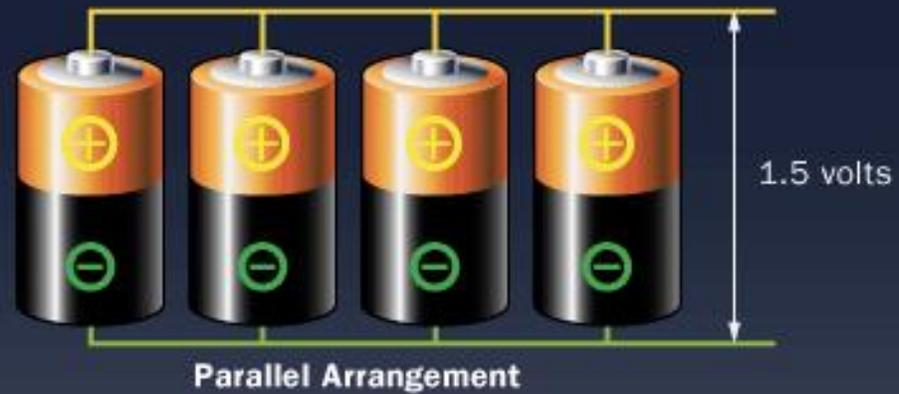
REGULAR POTENCIA

ENTONCES

- Entonces, si agregamos una carga **RL** al circuito, podemos aumentar la corriente entregada mediante:
 - Cambios en la resistencia del circuito.
 - Cambios en la fuente del circuito.
- ... Nada automático directamente.

ARREGLOS DE FUENTES

How Batteries Work Battery Packs

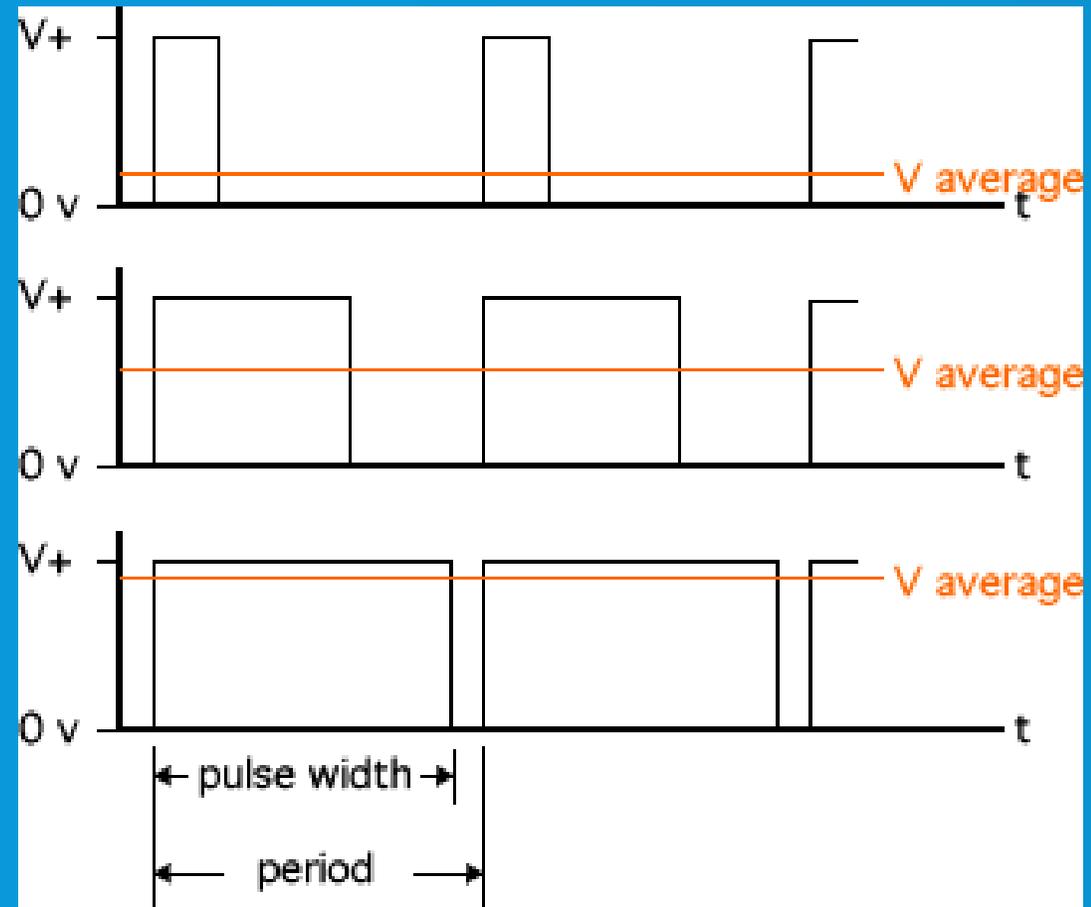


PWM

Explicación: **PWM** (Pulse Width Modulation)

Técnica utilizada por microcontroladores para emular voltajes intermedios a GND y VCC.

¡Arduino sólo sabe enviar 0 y 5 volts!

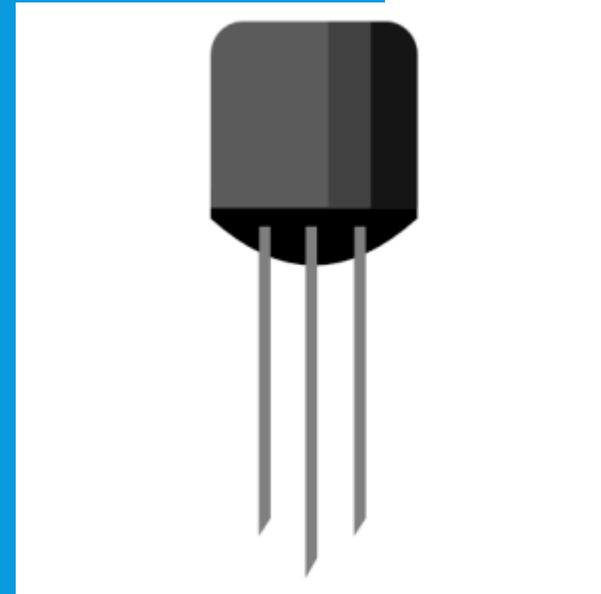


TRANSISTORES

¿Para qué nos sirven?

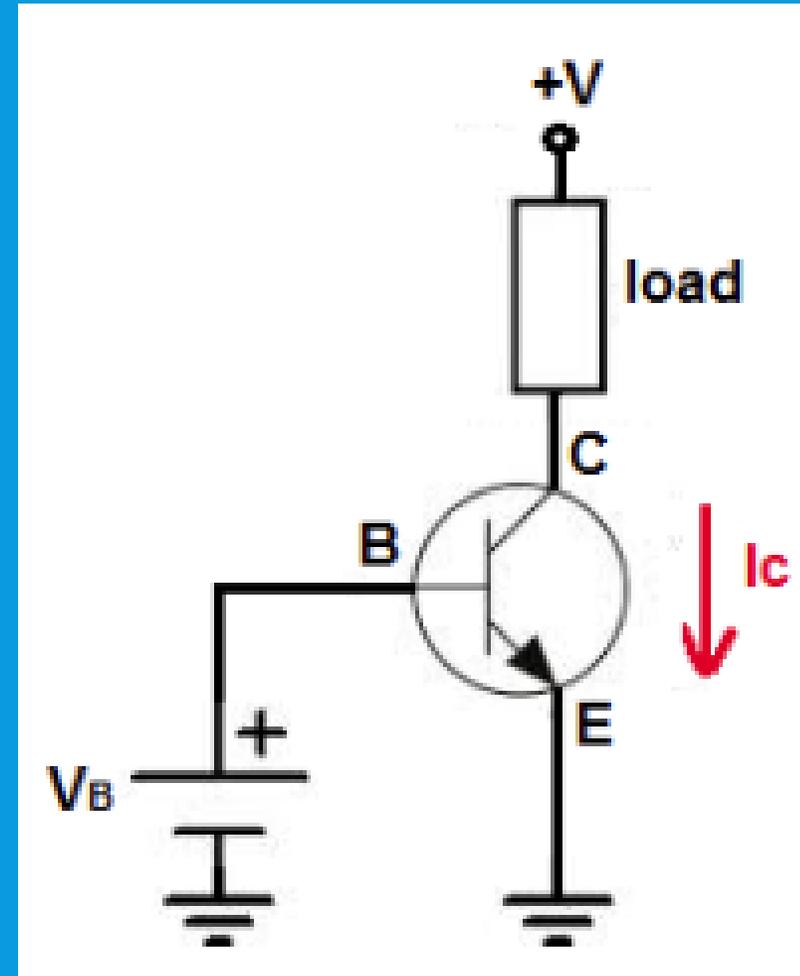
- Pueden actuar como un switch(ON/OFF) o regular potencia en un circuito 'aislado' del Arduino.
- Se pueden controlar mediante PWM.

¡Así no quemamos el Arduino!...
Sólo todo lo demás...



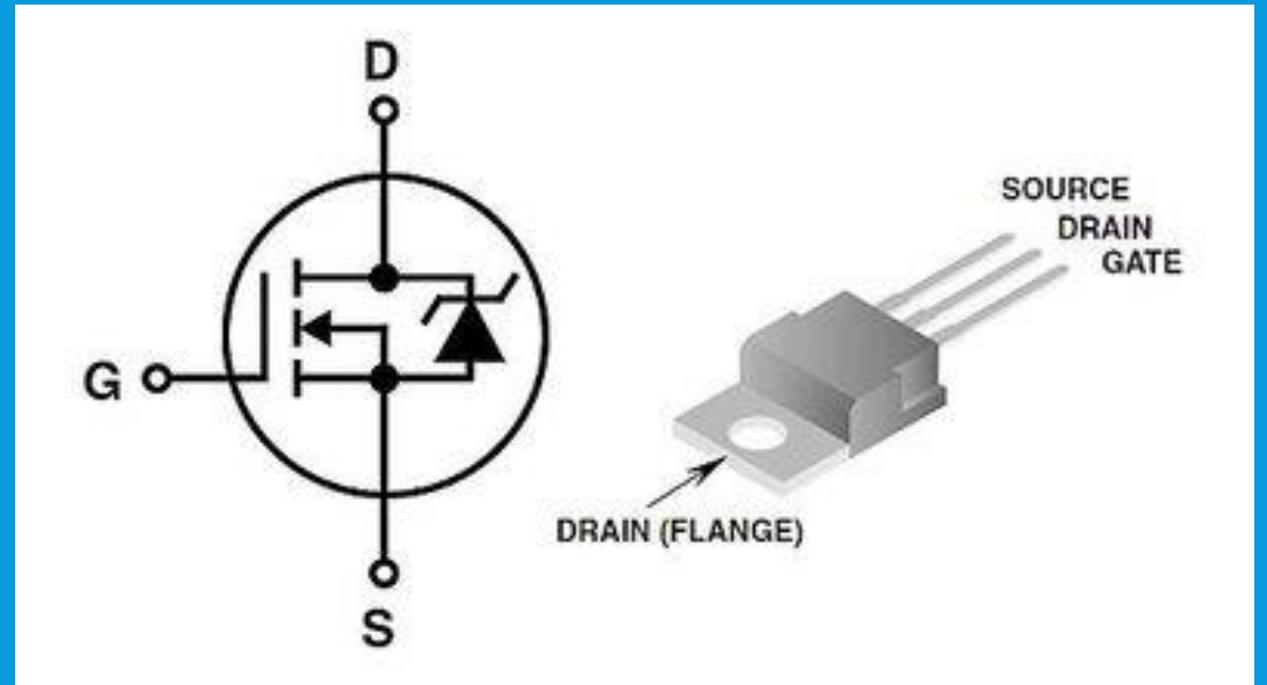
TIPOS DE TRANSISTORES

- Tipo BJT (Bipolar Junction)
- NPN: El circuito no funciona hasta que se le envía una señal pequeña.
- Posee 3 terminales, Base, colector y emisor.
- La señal pequeña en la base permite el flujo de corriente entre emisor y colector.
- También existen los PNP y funcionan al revés => No nos importan.



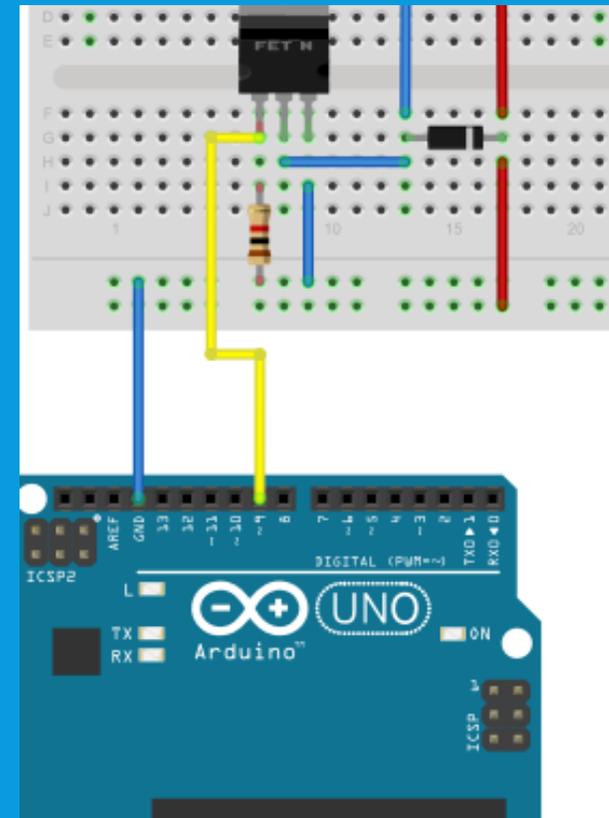
TIPOS DE TRANSISTORES

- **MOSFET:** (Metal-oxide-semiconductor Field-effect transistor)
- En términos prácticos, lo usaremos igual que el NPN.
- Posee 3 terminales, gate, drain y source.
- El gate sería como la base.

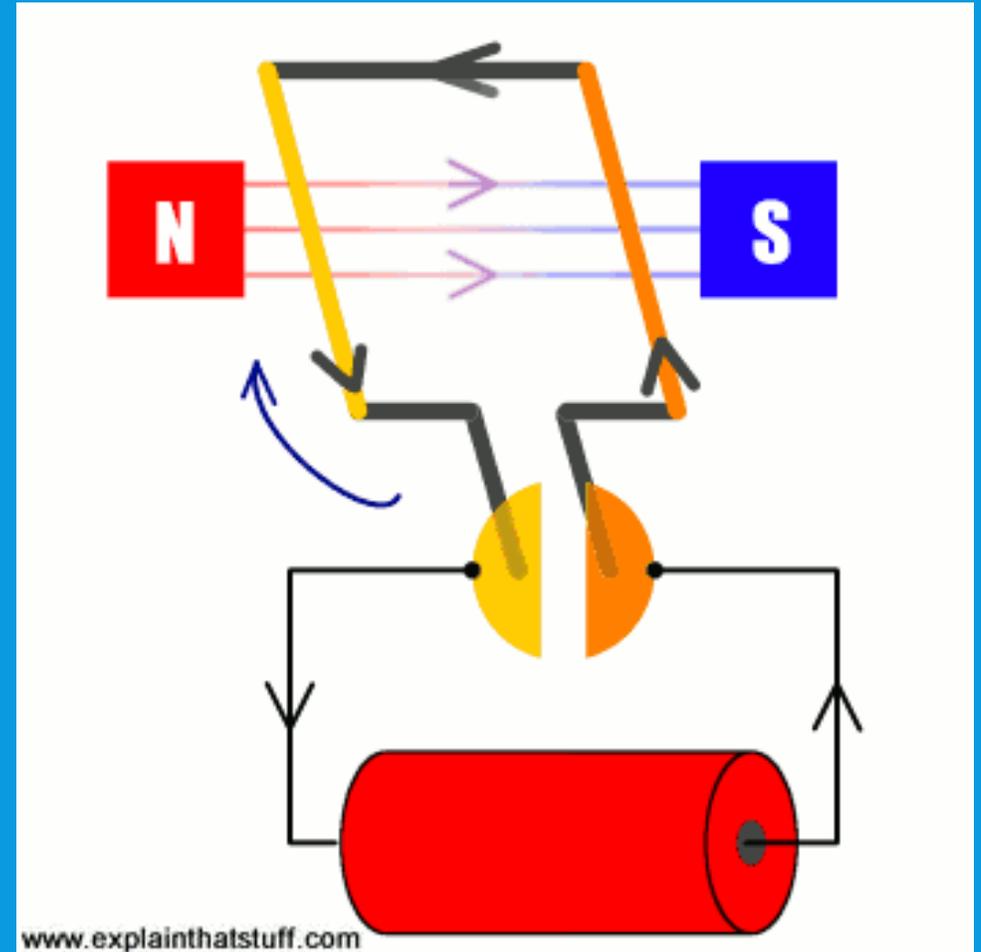


EXPERIENCIA 1: CONTROLAR MOTOR DC

- **Meta:** Cambiar potencia en el motor ingresando valores al monitor serial.
- **Materiales:**
 - 1 Arduino UNO
 - 1 Cable USB
 - 1 protoboard
 - 1 motor dc pequeño
 - 1 resistencia de 1 k Ω
 - 1 diodo
 - 1 transistor
 - Jumpers



MOTOR DC



PRIMERO: PROGRAMACIÓN

- Tenemos que asegurarnos que nos podemos comunicar con ARDUINO:
- **Serial.begin(9600):** Inicia la comunicación serial.
- **Serial.println(x):** Imprime la variable x en el monitor serial y se salta una línea. (con Serial.print() no se salta línea)
- **Serial.read():** Lee lo ingresado en el monitor serial.
- **Serial.available():** Boolean, se ingresó información al serial?
- **Serial.parseInt():** Transforma información del monitor serial a enteros.
- **Serial.readString():** Transforma información del monitor serial a String.

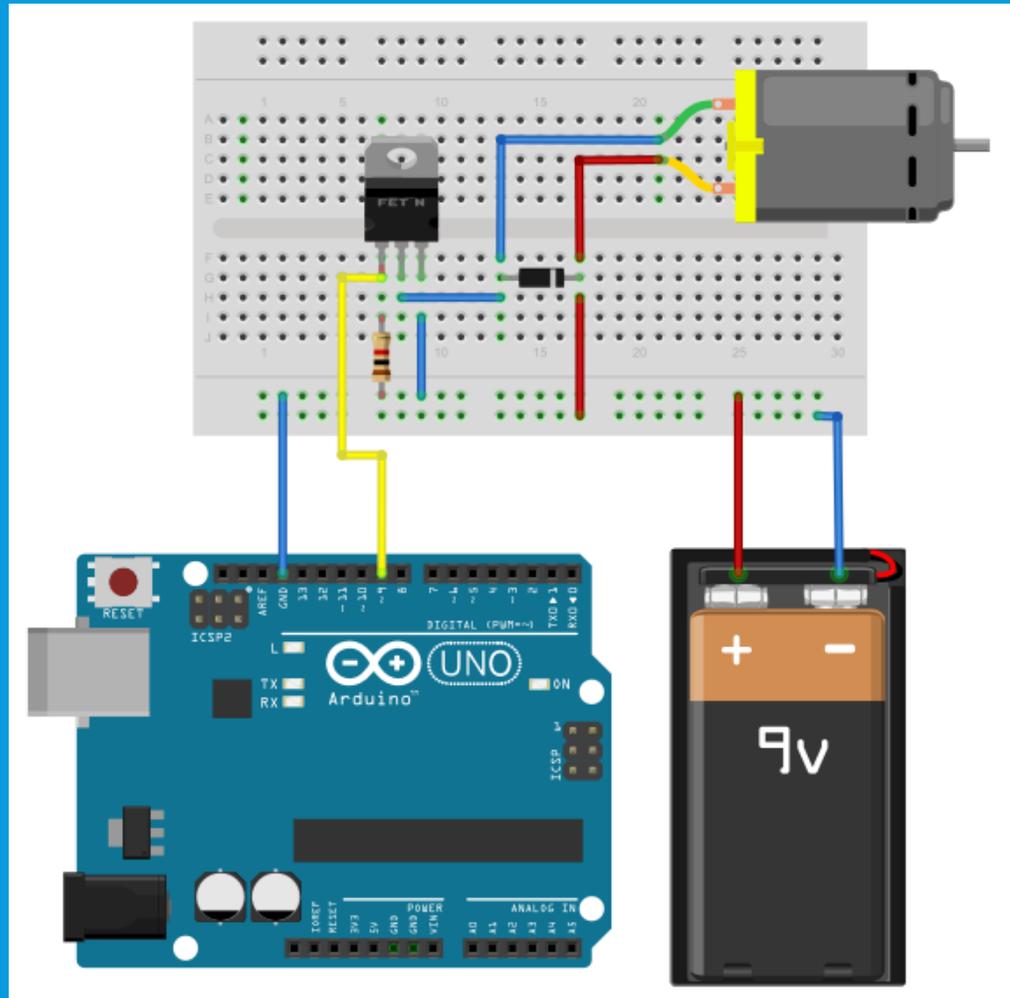
CÓDIGO

- Variables? Un entero para almacenar el valor ingresado.
- En el Setup? Declarar pin con pwm como OUTPUT, iniciar comunicación serial.
- En el loop? Reconocer cuando se envía información (available), guardar información en el entero (parseInt), escribir valor en el monitor serial (println) y enviar señal pwm (analogWrite).

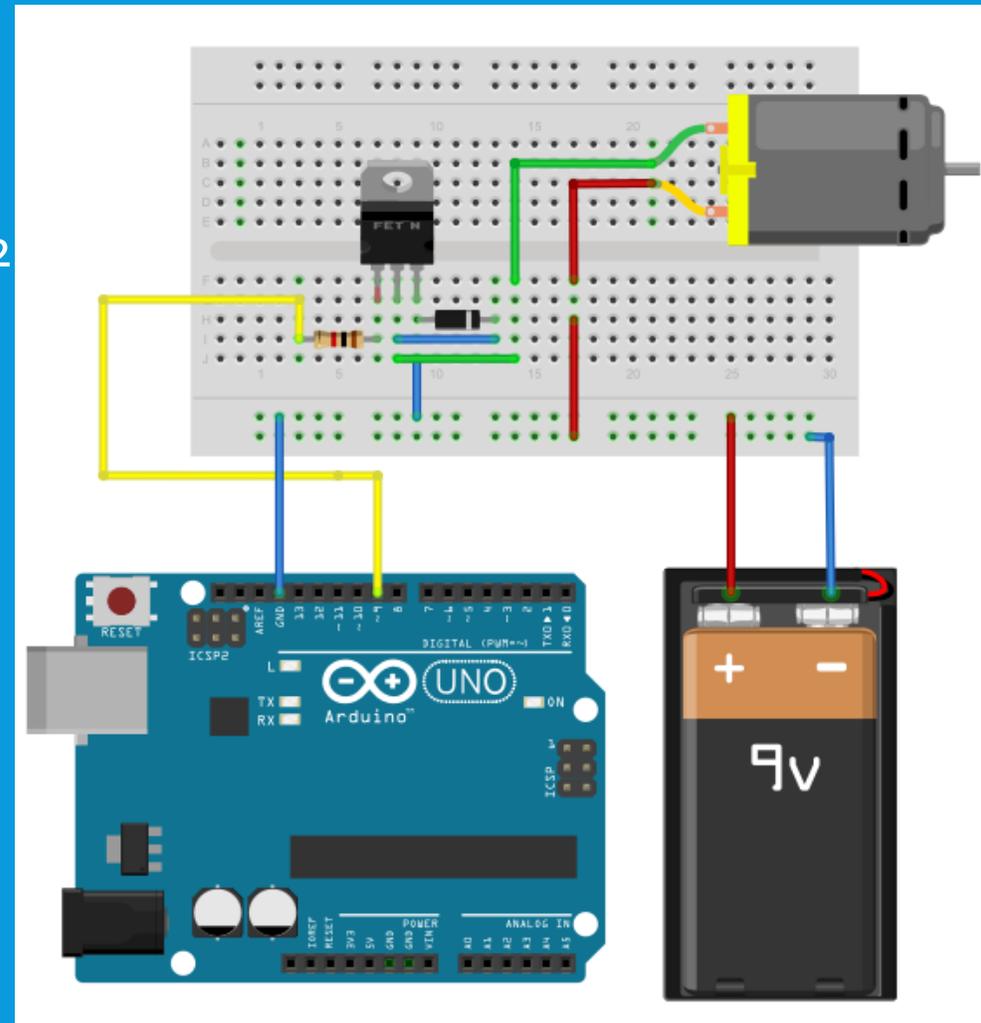
```
1  int num = 0;
2  int pin = 9;
3
4  void setup() {
5    pinMode(pin, OUTPUT);
6    Serial.begin(9600);
7  }
8
9  void loop() {
10
11     if(Serial.available()>0){
12       num = Serial.parseInt();
13       Serial.println(num);
14       analogWrite(pin,num);
15     }
16
17 }
```

SEGUNDO: CIRCUITO

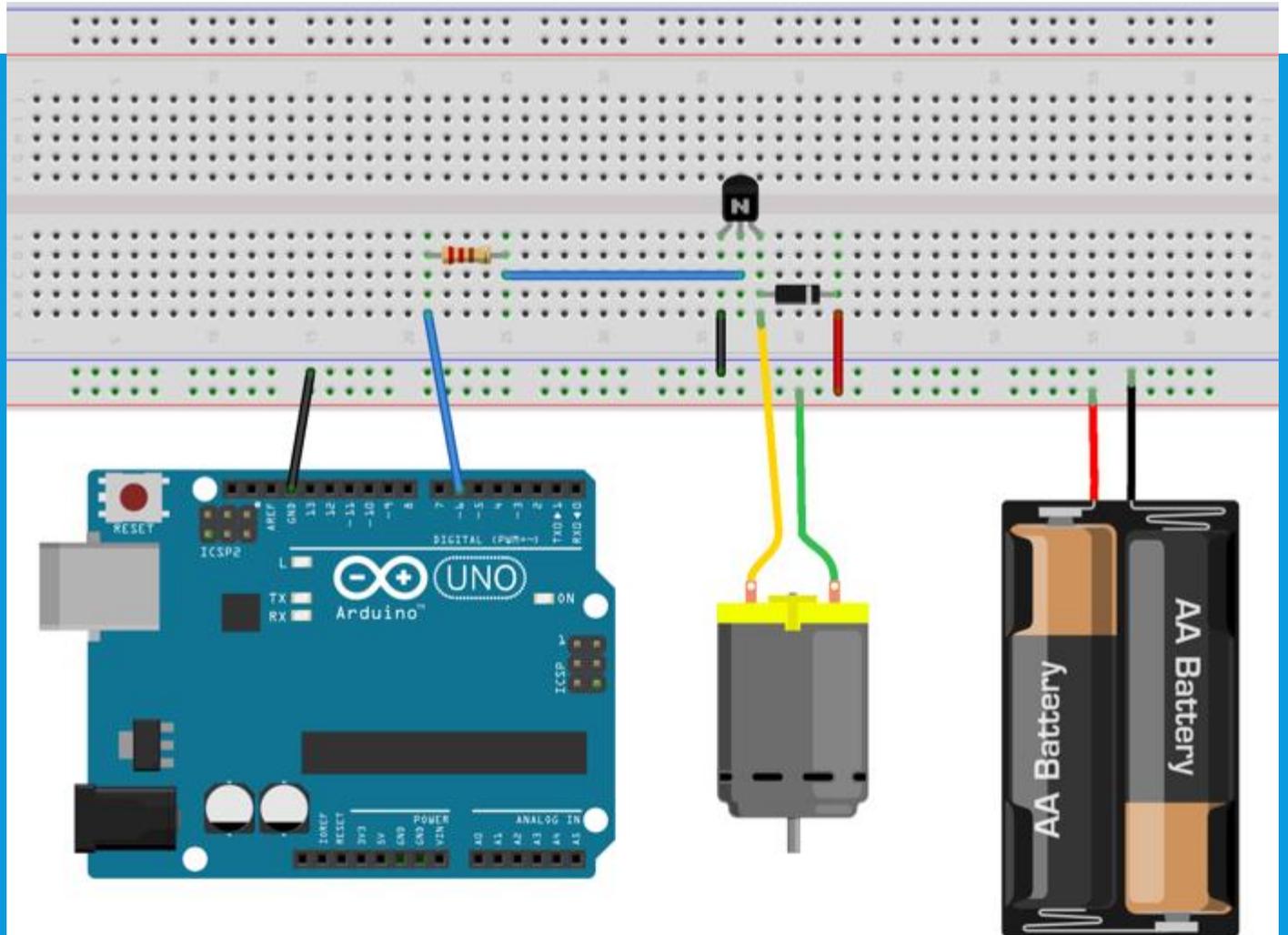
• IRF 540



• TIP 122

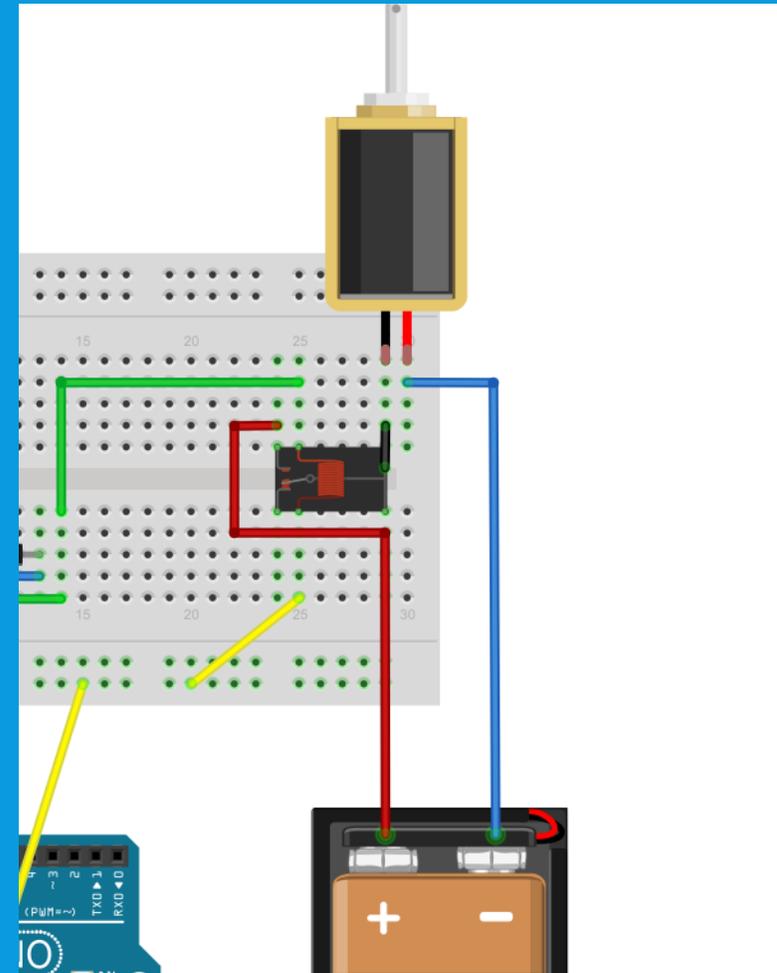


- PN2222A

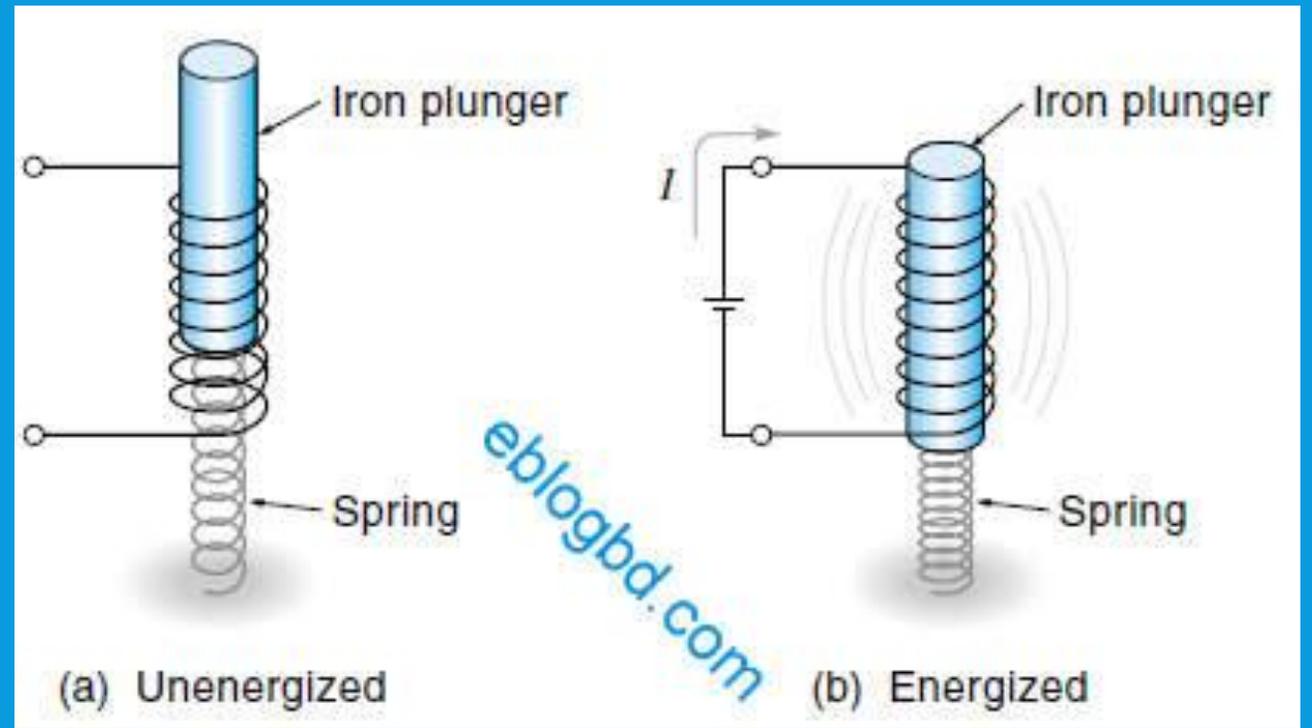


EXPERIENCIA 2: RELÉ

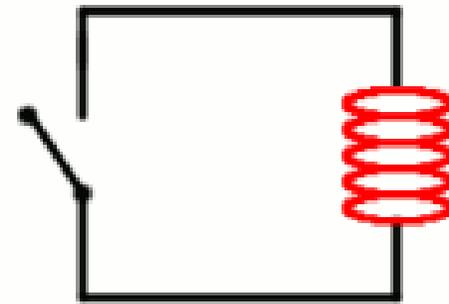
- **Meta:** Abrir y cerrar válvula utilizando un relé y comunicación serial.
- **Materiales:**
 - Circuito anterior +
 - 1 válvula de solenoide
 - 1 Relé
 - Jumpers



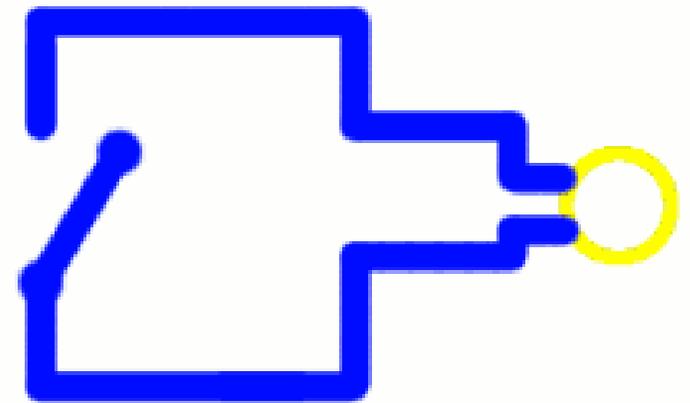
VÁLVULA DE SOLENOIDE



RELÉ



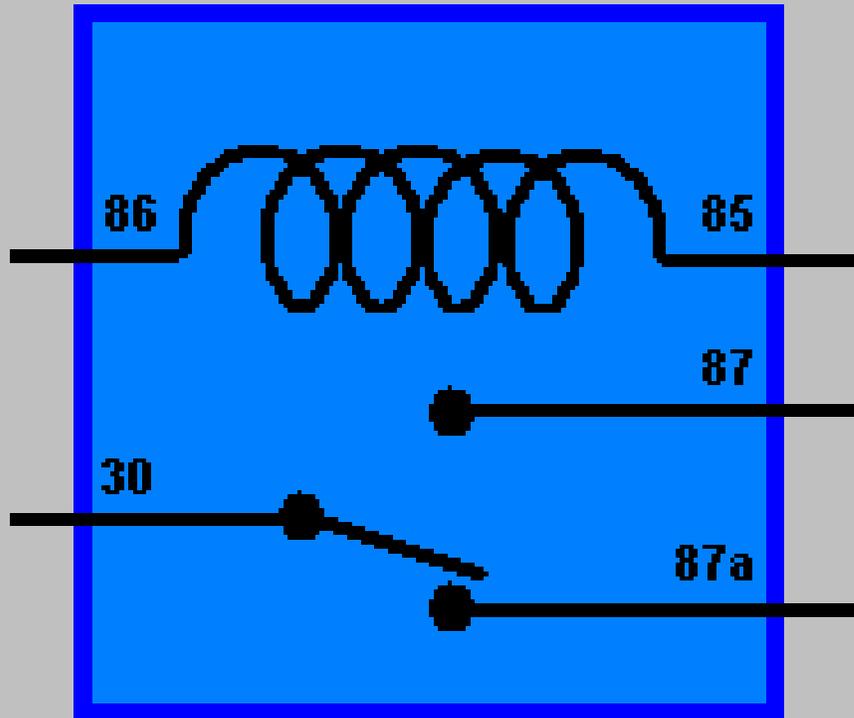
Input circuit
(low-current)



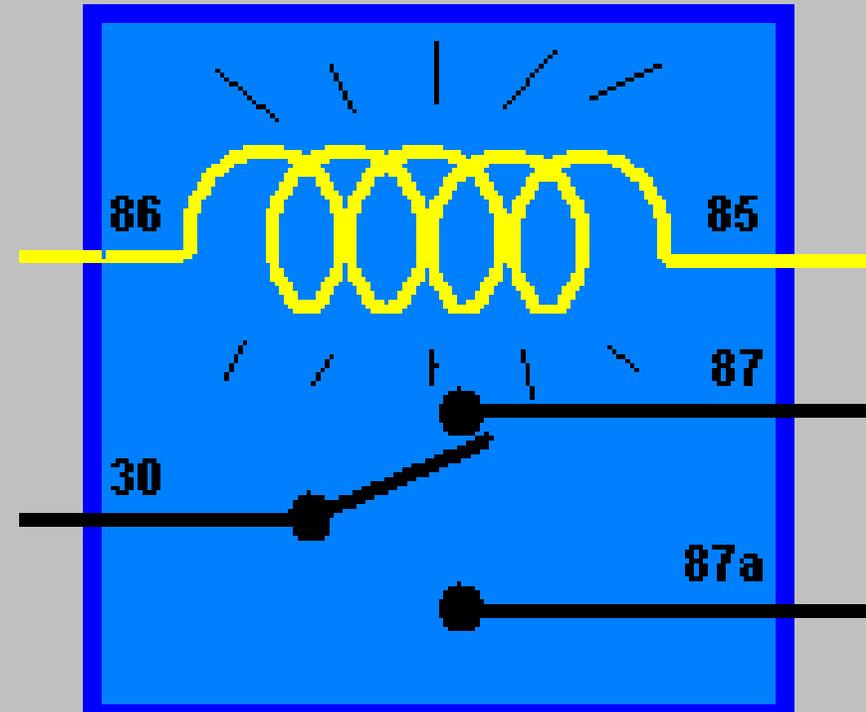
Output circuit
(high-current)

www.explainthatstuff.com

5 LEAD RELAY



RELAY 'OFF' (coil unpowered)



RELAY 'ON' (coil powered)

RELÉ VS TRANSISTOR

- El relé no permite regular la carga.
- Es más simple utilizar el relé, mucho más cuando se compra integrado en una tarjeta, como un shield.
- El relé provee una separación física de los circuitos => mejor protección.
- El relé no se calienta.
- El relé soporta mayor tensión y corriente.

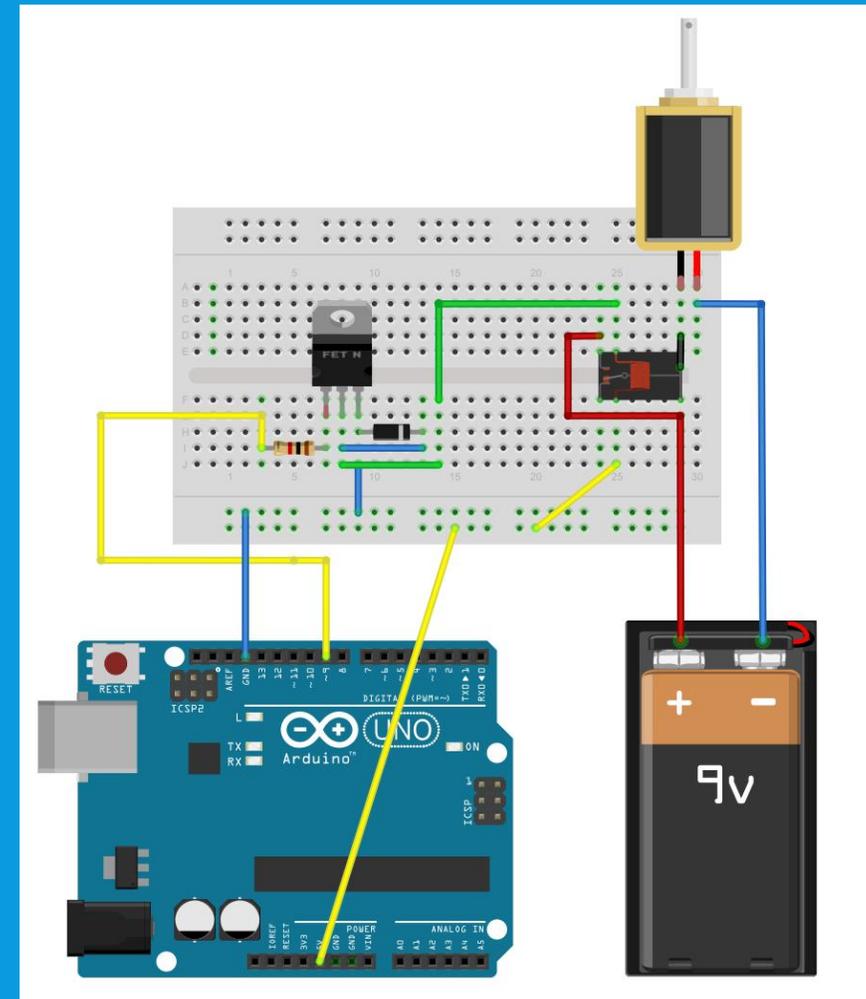
PROGRAMACIÓN

- Variables? Un char para almacenar el caracter ingresado.
- En el Setup? Declarar pin digital como OUTPUT, iniciar comunicación serial.
- En el loop? Reconocer cuando se envía información (available), guardar información en el char (read), comparar char y enviar señal (digitalWrite).

```
1 char estado = 'n';
2 int pin = 9;
3
4 void setup() {
5   pinMode(pin, OUTPUT);
6   Serial.begin(9600);
7 }
8
9 void loop() {
10
11   if(Serial.available()>0){
12     estado = Serial.read();
13     if (estado=='y'){
14       digitalWrite(pin,HIGH);
15       Serial.println("Abierto");
16     }
17     if (estado=='n'){
18       digitalWrite(pin,LOW);
19       Serial.println("Cerrado");
20     }
21   }
22
23 }
```

CIRCUITO

- Se utiliza el mismo circuito anterior, reemplazando el motor por la bobina del relé. La bobina se puede alimentar con el Arduino.
- Un terminal de la fuente se conecta al NO del relé, el otro a un terminal de la válvula.
- El terminal libre de la válvula se conecta al COM.



PULSADOR

- Alternativamente, podemos usar un pulsador para abrir y cerrar válvula.
- Requiere pulsador y resistencia.
- El botón se conecta a 5V.
- La señal se lee en un pin digital (`digitalRead`) y controla la señal al relé (`digitalWrite`).



CIRCUITO

