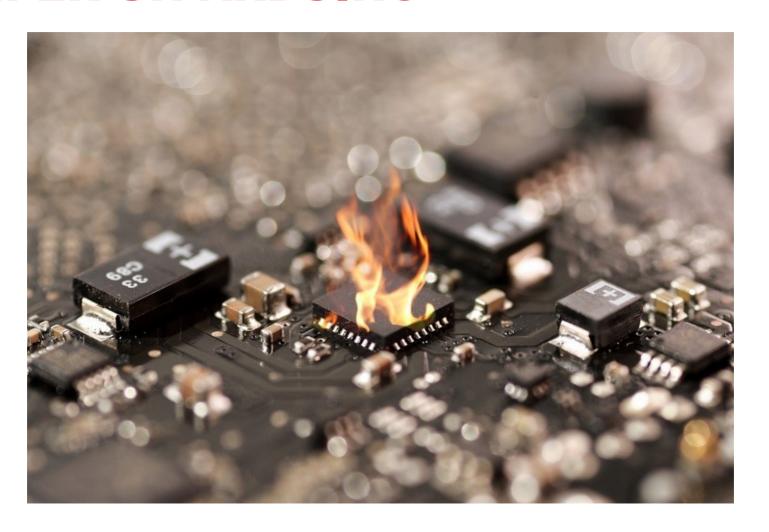


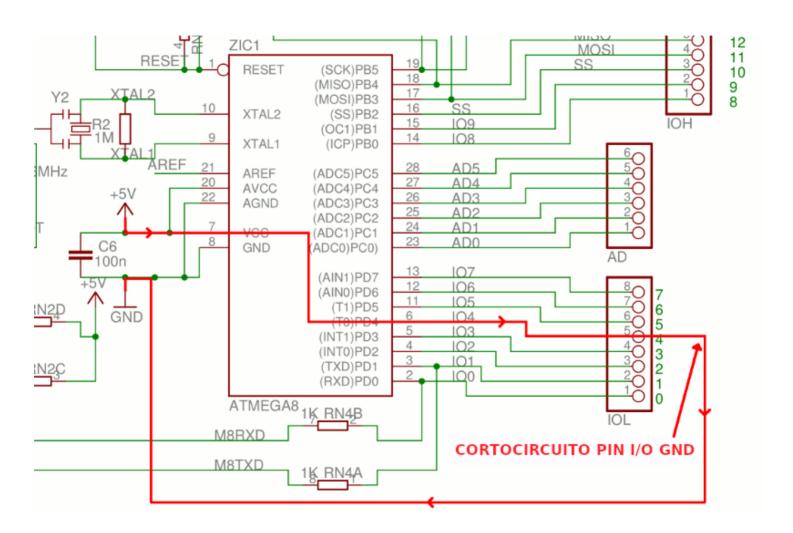
TALLER DE ROBÓTICA CON ARDUINO



#1: CORTOCIRCUITO ENTRE GND Y PINES E/S

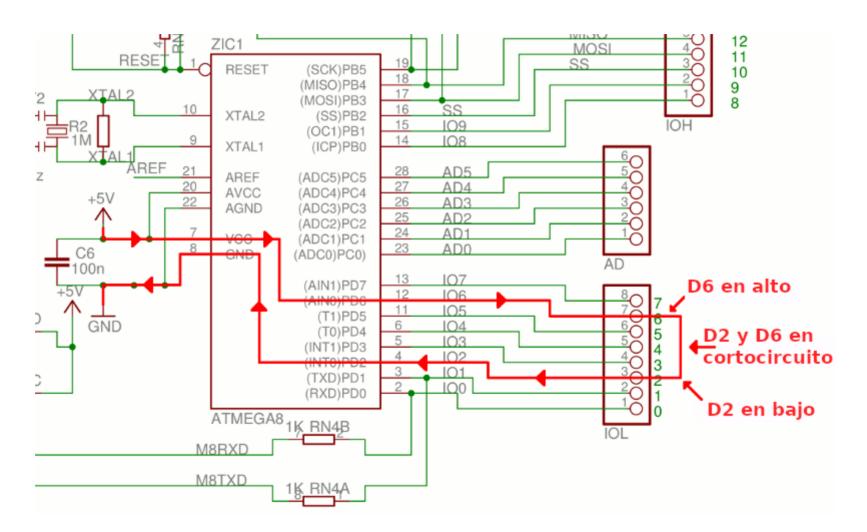
Configurar un pin de E/S para que sea una salida y luego escribir un estado alto en él. Conectar el pin a GND, se creará una condición de sobrecorriente en el pin y se destruirá.

La hoja de datos del microcontrolador especifica una **corriente máxima por pin de 40mA**. Con una resistencia interna típica de solo 25Ω por pin, un cortocircuito a tierra puede permitir que fluyan hasta 200mA, más que suficiente para destruir el pin del microcontrolador.



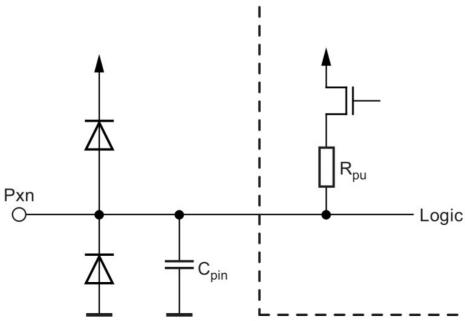
#2: CORTOCIRCUITO ENTRE PINES DE E/S

Configurar dos pines de E/S para que sean salidas, luego escribir un estado alto en uno y un estado bajo en el otro. Ahora conectar los pines juntos. Se creará una condición de sobrecorriente en ambos pines de E/S y serán destruidos.



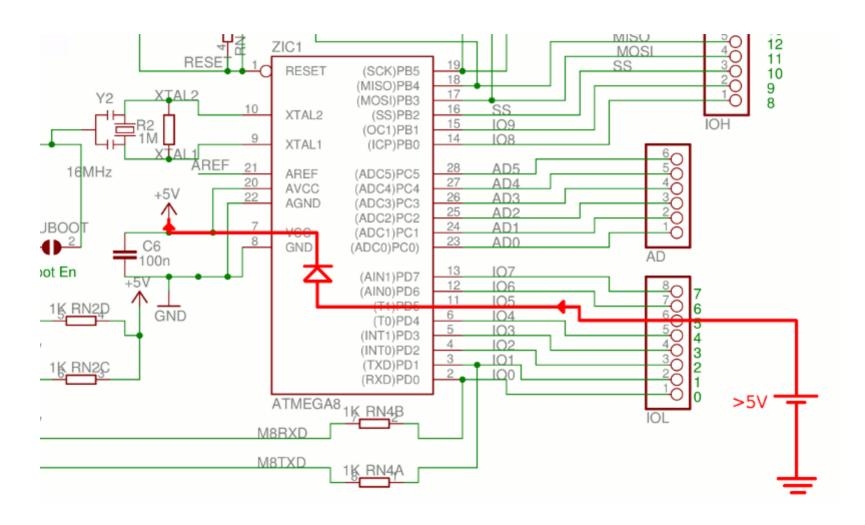
#3: APLICAR SOBRETENSIÓN A LOS PINES DE E/S

Aplicar una tensión superior a **5,5V** a cualquier pin de E/S. El pin será destruido. Este método de destrucción polariza hacia adelante el diodo de protección ESD integrado en el microcontrolador.



Si la tensión en el pin de E/S es **0,5V mayor que la tensión de fuente** (5V), el diodo de protección comienza a conducir corriente. Esto está bien para desviar un evento de sobretensión de corta duración, como un descarga electrostática (ESD), pero ese diodo no debe estar encendido todo el tiempo. Simplemente se quemará y dejará de proteger al pin.

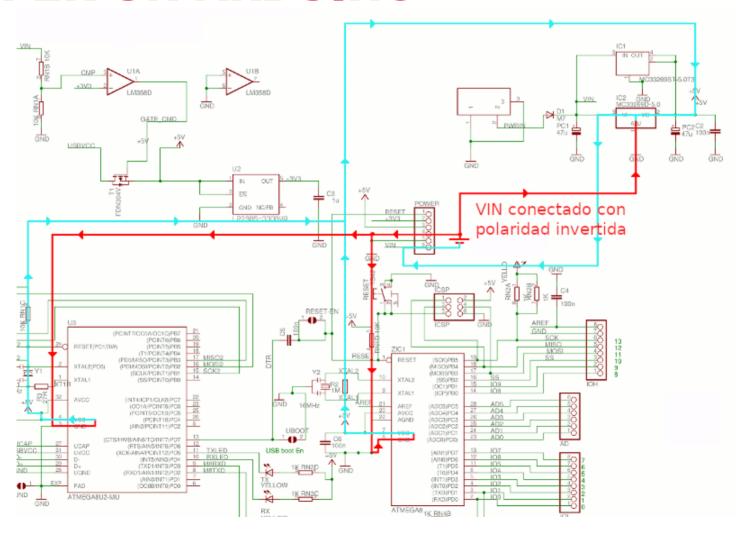
- Si el diodo de protección interno no se abre, la sobretensión destruirá el pin de E/S.
- Si el diodo de protección falla por un cortocircuito, es aún peor porque ahora la sobretensión se aplica a todo el suministro de +5V en el Arduino. Esto significa que llegará a otros componentes, como el chip USB, y también los destruirá.



#4: APLICAR TENSIÓN CON POLARIDAD INVERTIDA EN VIN

Conectar el Arduino a través del pin VIN, pero con la polaridad invertida. Destruirá varios dispositivos en el Arduino.

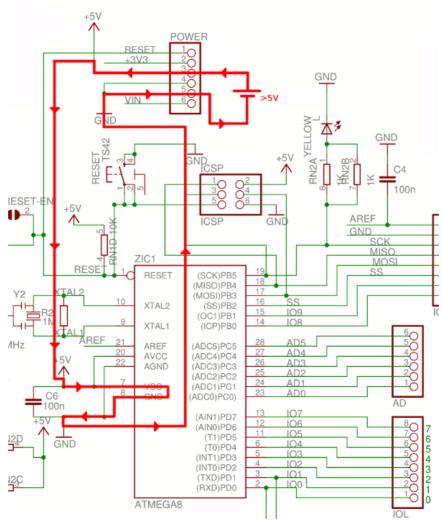
No hay protección contra inversión de tensión en el conector VIN. La corriente fluirá desde el pin GND del ATmega328P hacia arriba a través del pin de 5V, y de regreso a través del regulador de 5V hacia VIN. Se destruirán los microcontroladores y el regulador de 5V.



#5: APLICAR MÁS DE 5V AL CONECTOR DE 5V

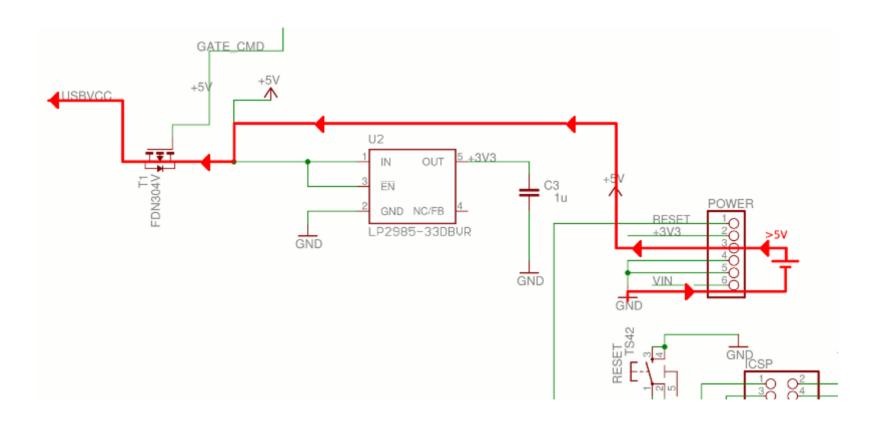
Aplicar una tensión de 6V o más al conector de 5V. Muchos componentes del Arduino serán destruidos y esta tensión también puede aparecer en el puerto USB de su computadora, posiblemente dañándolo.

No hay protección en el conector de 5V, está conectado directamente al microcontrolador ATmega328P, al microcontrolador de interfaz USB ATmega16U2 y al regulador de 5V, todos los cuales pueden resultar dañados por tensiones superiores a 6V.



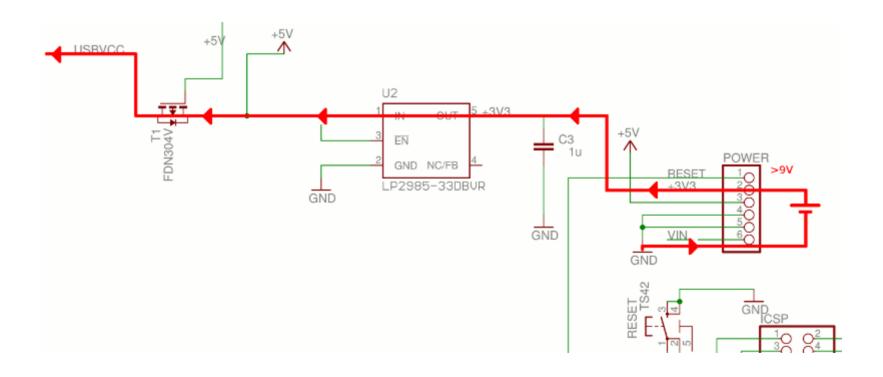
Es un error pensar que el regulador de 5V asegurará que la tensión de 5V permanezca en 5V pase lo que pase. ¡NO LO HARÁ! Lo único que puede hacer el regulador de 5V es controlar la corriente que proviene del conector de alimentación externo. Si la corriente proviene de una fuente de alimentación externa conectada directamente al conector de 5V, el regulador no puede hacer nada al respecto.

Otra consecuencia de aplicar más de 5V al conector de 5V es un posible daño al puerto USB de la PC. Si el Arduino se alimenta desde USB, este voltaje excesivo puede hacer que la corriente fluya hacia atrás a través del MOSFET T1 y de regreso al puerto USB de la PC.



#6: APLICAR MÁS DE 3,3V AL CONECTOR DE 3,3V

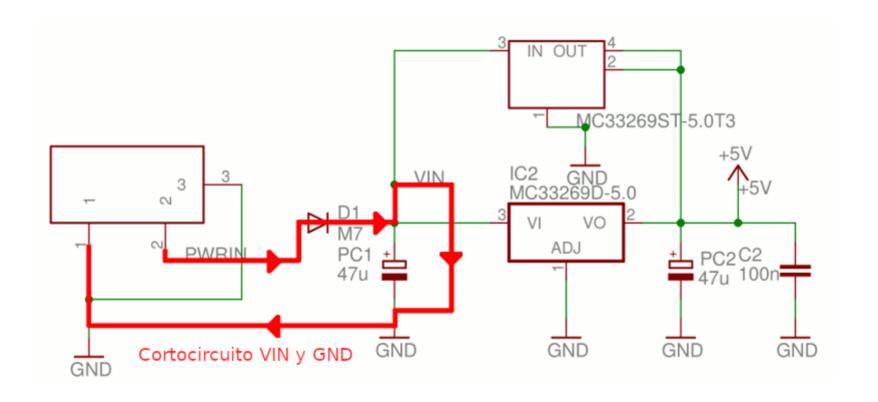
Aplicar una tensión de 3,6V o más al conector de 3,3V. Cualquier módulo de 3,3V enchufado u otros dispositivos alimentados desde este pin serán destruidos. Si se aplica al menos 9V, esta tensión puede destruir el regulador de 3,3V y también devolver la corriente al puerto USB de la PC. El conector de 3,3V no tiene circuitos de protección.



#7: CORTOCIRCUITO ENTRE VIN Y GND

Alimentar el Arduino desde el conector de alimentación externo y realizar un cortocircuito entre VIN y GND. El diodo de protección D1 se destruirá y las pistas del PCB pueden derretirse y destruirse.

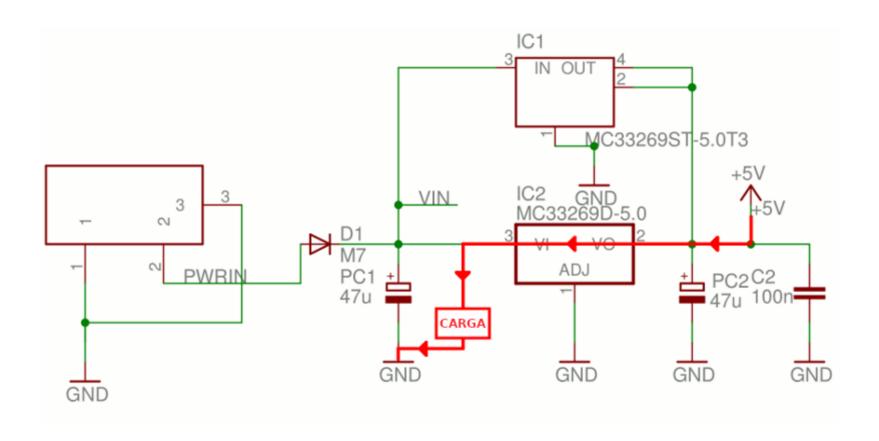
No hay protección de corriente en el conector VIN, y la cantidad de corriente que fluye está limitada solo por la resistencia de las pistas de la PCB y la capacidad de la fuente de alimentación. Si es lo suficientemente alta, el diodo D1 se destruirá y las pistas pueden derretirse debido al calor causado por esta gran corriente.



#8: APLICAR ALIMENTACIÓN EXTERNA DE 5V CON CARGA EN VIN

Si está alimentando la placa desde 5V aplicados al conector de 5V y tiene circuitos conectados al pin VIN (o hay un cortocircuito entre VIN y GND), la corriente fluirá hacia atrás a través del regulador de 5V y lo destruirá.

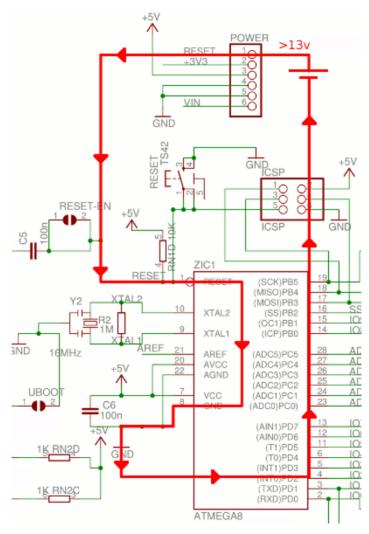
No hay protección de tensión inversa en el regulador de 5V, por lo que la corriente puede fluir desde el conector de 5V, hacia atrás a través del regulador y hacia lo que esté conectado a VIN.



#9: APLICAR MÁS DE 13V AL PIN DE REINICIO

Aplicar una tensión superior a 13V al conector de RESET. El microcontrolador ATmega328P resultará dañado.

El conector de RESET está directamente conectada al pin de RESET del ATmega328P. Si bien este pin tolera 13V, tensiones más altas dañarán el dispositivo.

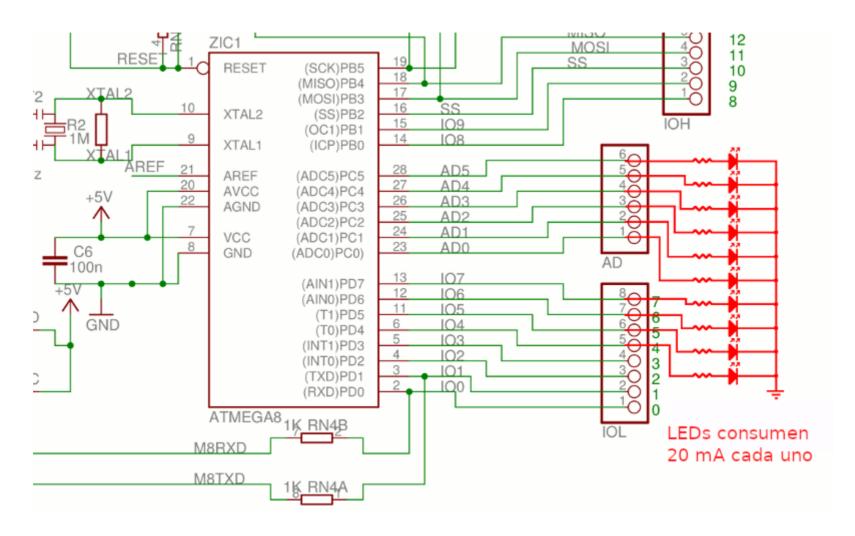


#10: EXCEDER LA CORRIENTE TOTAL DEL MICROCONTROLADOR

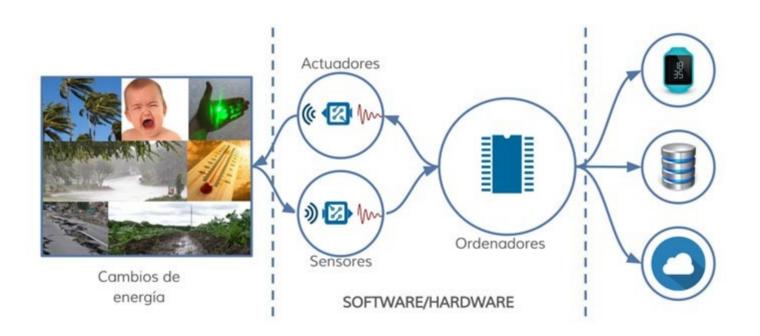
Configurar al menos 10 pines como salidas y establecer un estado alto en ellos, luego extraer 20mA de cada uno (por ejemplo, encendiendo 10 LEDs). Ahora ha superado la corriente total que puede suministrar el microcontrolador y se dañará.

La corriente total procedente de todos los pines de E/S no debe exceder los 200mA de acuerdo con la hoja de datos del ATmega328P.

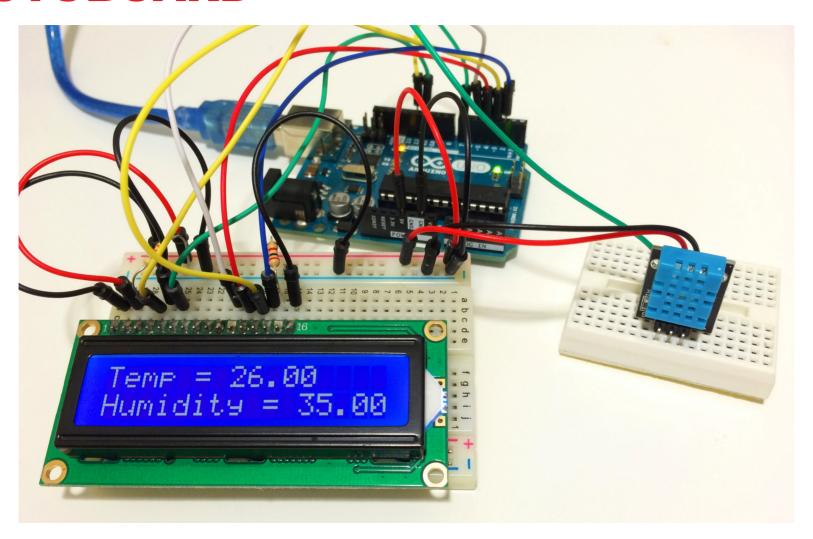
»»» 10 Ways to Destroy An Arduino



COMPUTACIÓN FÍSICA



PROTOBOARD

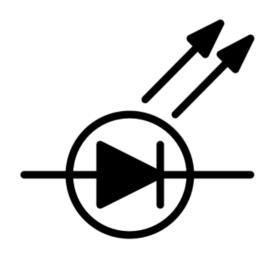


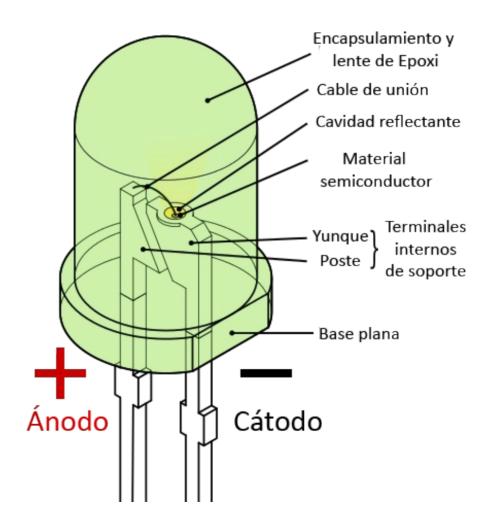
PROTOBOARD

»»» Uso del protoboard

LED

Un diodo emisor de luz (LED, del inglés light-emitting diode) es una fuente de luz constituida por un material semiconductor dotado de dos terminales: ánodo y cátodo.





Pines configurados como salidas se dice que están en un estado de *baja impedancia*. Esto significa que pueden proveer una substancial cantidad de corriente a otros circuitos.

Los pines del ATmega pueden originar (proveer corriente) o consumir (absorber corriente) hasta 40mA hacia otros dispositivos o circuitos. Esto los hace útiles para alimentar LEDs. Cargas mayores a 40mA (por ejemplo motores) necesitarán de un transistor u otro circuito de interfaz.

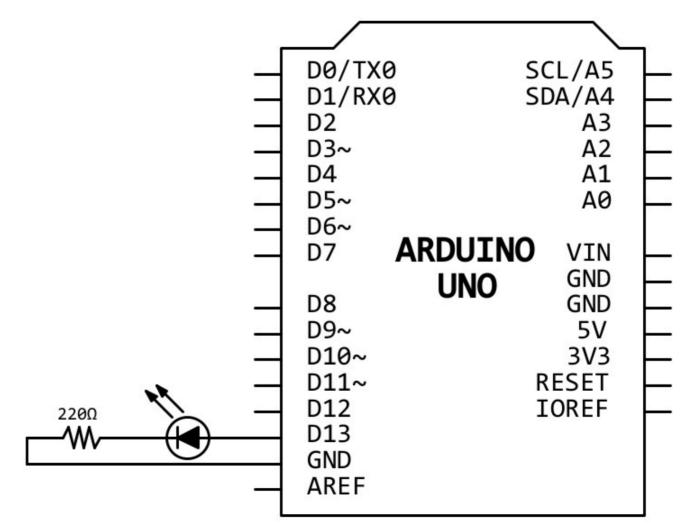
Se **define** con la instrucción pinMode(pin, OUTPUT). Por ejemplo:

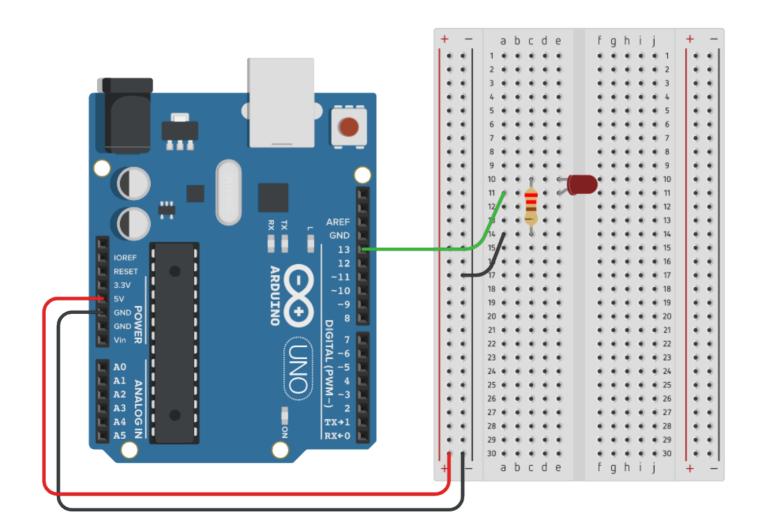
```
pinMode(13, OUTPUT);
```

Se **escribe** con la instrucción digitalWrite(pin, estado). Por ejemplo:

```
digitalWrite(13, HIGH);
digitalWrite(13, LOW);

>>>> pinMode()
>>>> digitalWrite()
```

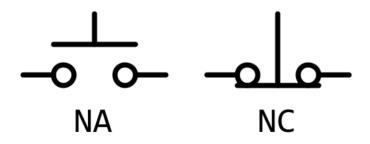




PULSADOR

Se utilizan las palabras botón, tecla, o pulsador para describir a un componente, el cual al ejercer presión sobre él (generalmente con un dedo), produce un efecto determinado.

Podemos encontrar dos tipos: Normal Abierto (NA) y Normal Cerrado (NC).





Los pines configurados como entradas se dice que están en un estado de *alta impedancia*. Hacen muy poca demanda en el circuito en el que se usan, equivalente a una resistencia en serie de $100 \text{M}\Omega$ en frente del pin. Esto los hace útiles para leer un sensor.

Se **define** con la instrucción pinMode(pin, INPUT). Por ejemplo:

```
pinMode(7, INPUT);
```

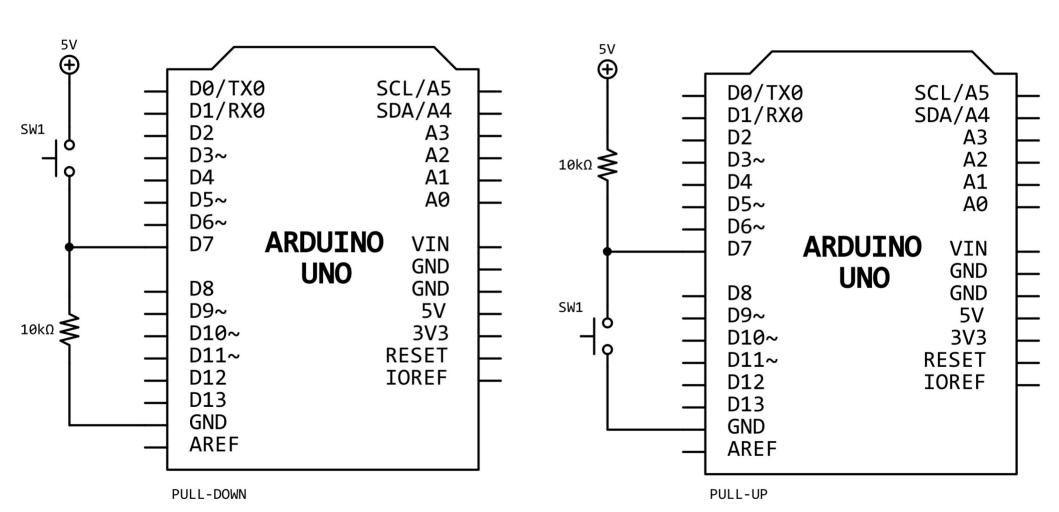
Se **lee** con la instrucción digitalRead(pin) y devuelve los valores (0, LOW, false ó 1, HIGH, true). Por ejemplo:

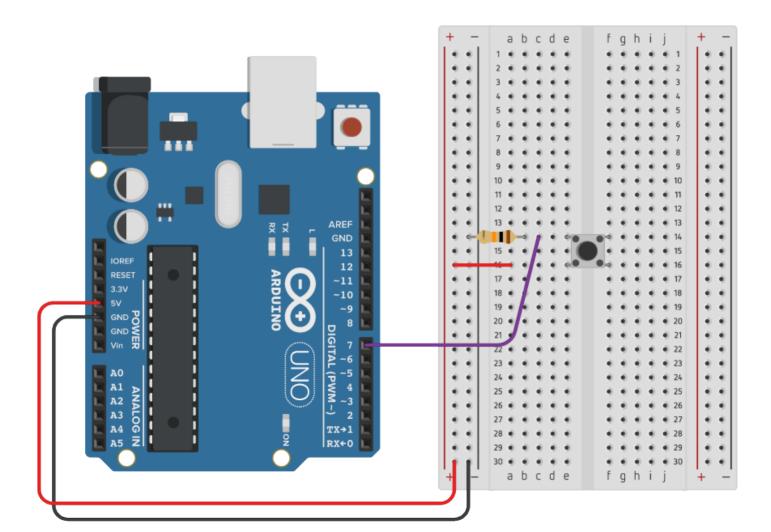
```
digitalRead(7);

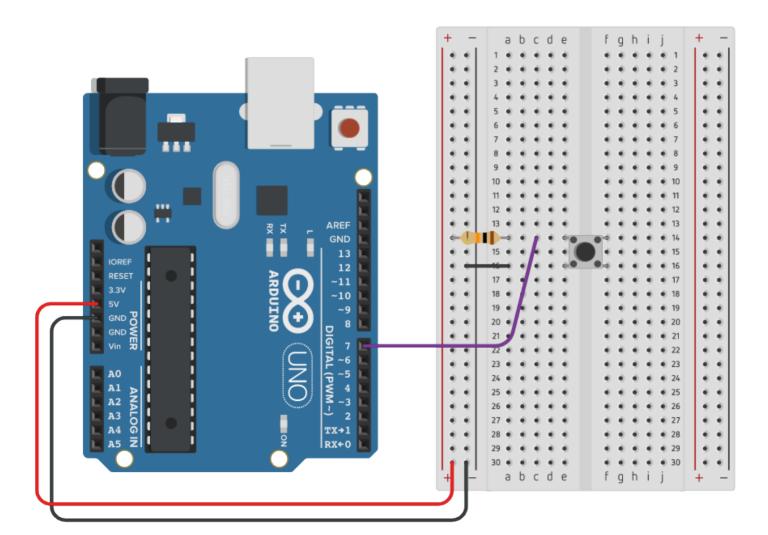
>>>> pinMode()
>>>> digitalRead()
```

Si estas leyendo un pulsador, cuando este en el estado abierto el pin de entrada estará "flotando", con resultados impredecibles. Para asegurarse que la lectura sea correcta se debe usar una resistencia de acoplamiento a positivo o a tierra. El propósito de esta resistencia es la empujar al pin a un estado conocido cuando el pulsador esta abierto. Se utiliza generalmente una resistencia de $10k\Omega$, ya que es lo suficientemente baja para prevenir una entrada flotante, y al mismo tiempo lo suficientemente alta para no dejar pasar demasiada corriente cuando el pulsador este cerrado.

- Si se usa una resistencia con acoplamiento a tierra (**pull-down**), el pin de entrada será LOW cuando el pulsador esta abierto y HIGH cuando este cerrado.
- Si se usa una resistencia con acoplamiento a positivo (**pull-up**), el pin de entrada será HIGH cuando el pulsador este abierto y LOW cuando este cerrado.



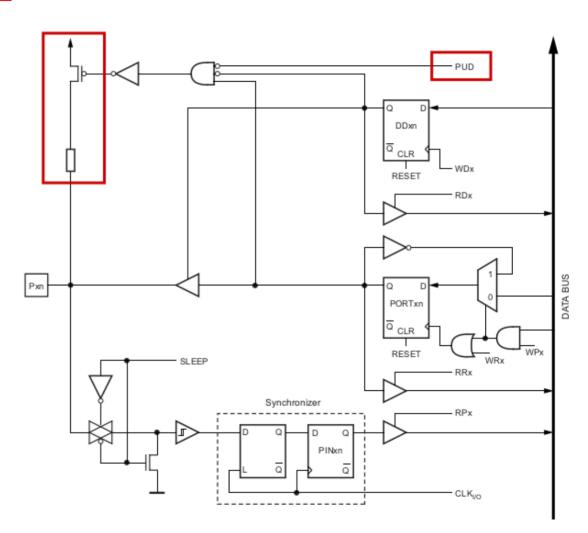


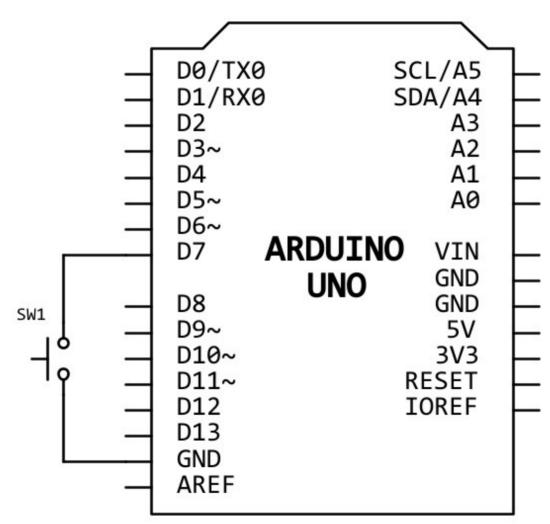


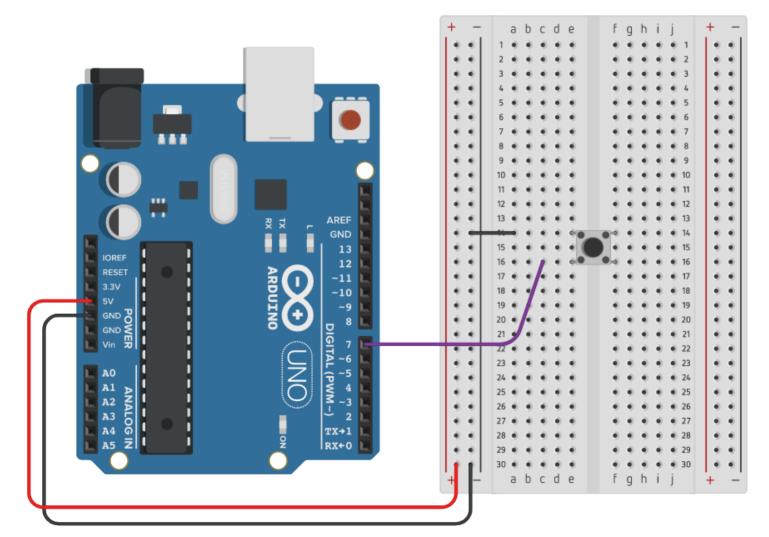
El microcontrolador ATmega tiene resistencias, de entre $20k\Omega$ y $50k\Omega$, acopladas a positivo internamente (pull-up) que puedes usar. Si deseas usar estos en vez de resistencias externas, puedes usar el argumento INPUT_PULLUP en pinMode().

```
pinMode (7, INPUT_PULLUP);

>>>> pinMode()
```







DESAFÍO

¡A experimentar!

CRÉDITOS

Lucas Martín Treser

Imtreser@gmail.com – www.automatismos-mdq.com.ar



Atribución-NoComercial 4.0 Internacional (CC BY-NC 4.0)