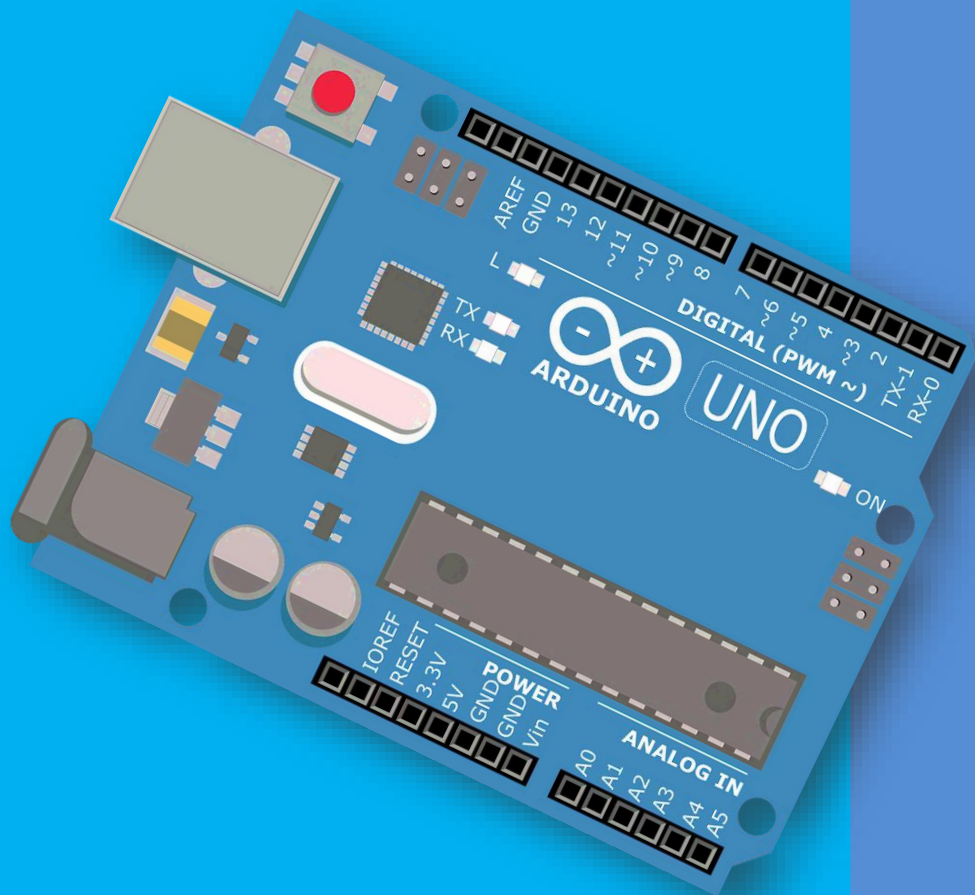


CURSO DE ARDUINO

Clase 28: ¿Cómo se conecta un pulsador?

Instructor: Konrad Peschka



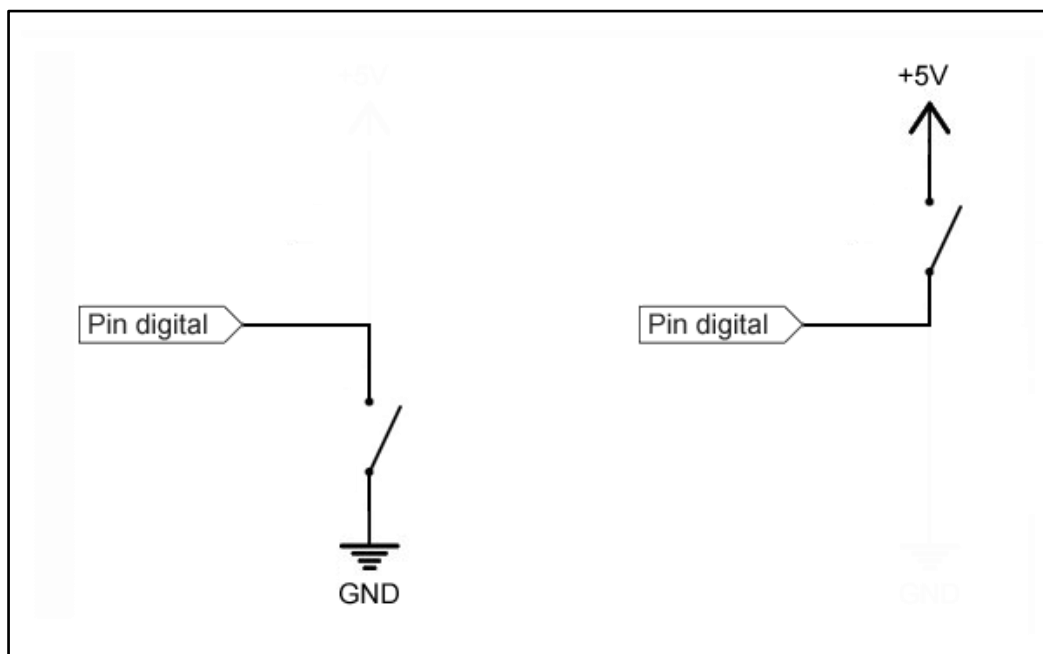
¿Cómo se conecta un pulsador?

Vimos como leer un sensor que proporcione una señal digital con dos niveles de tensión LOW y HIGH. Nos quedó pendiente aprovechar las entradas digitales para leer el estado de un interruptor o pulsador con Arduino,

Para entender el funcionamiento de montaje correcto de pulsadores vamos a presentar el razonamiento lógico por el cual se llega al mismo, para lo cual utilizaremos dos montajes incorrectos que nos ayudarán a entender el papel de cada elemento en el montaje definitivo.

Conexión directa

Nuestra primera idea para leer un pulsador podría ser conectar directamente un PIN digital de Arduino a un valor de tensión de referencia, ya sea este 0V o 5V. Al cerrar el pulsador la tensión en el PIN sería el valor de referencia (0V o 5V depende del montaje) y podríamos realizar la lectura como en cualquier entrada digital.



Esquema de conexión directa

¿Cuál es el problema?

Bueno, esto funcionará cuando el interruptor está cerrado. Pero, ¿Qué pasa cuando el interruptor está abierto? En este caso estamos dejando el PIN totalmente desconectado de cualquier tensión, algo que denominaremos estado de alta impedancia. la entrada está en un estado indeterminado, es decir, que puede asumir cualquier valor y el Arduino funcionará de forma inesperada, afectando la continuidad del programa. Por tanto, es necesario evitar esta situación en nuestros diseños.

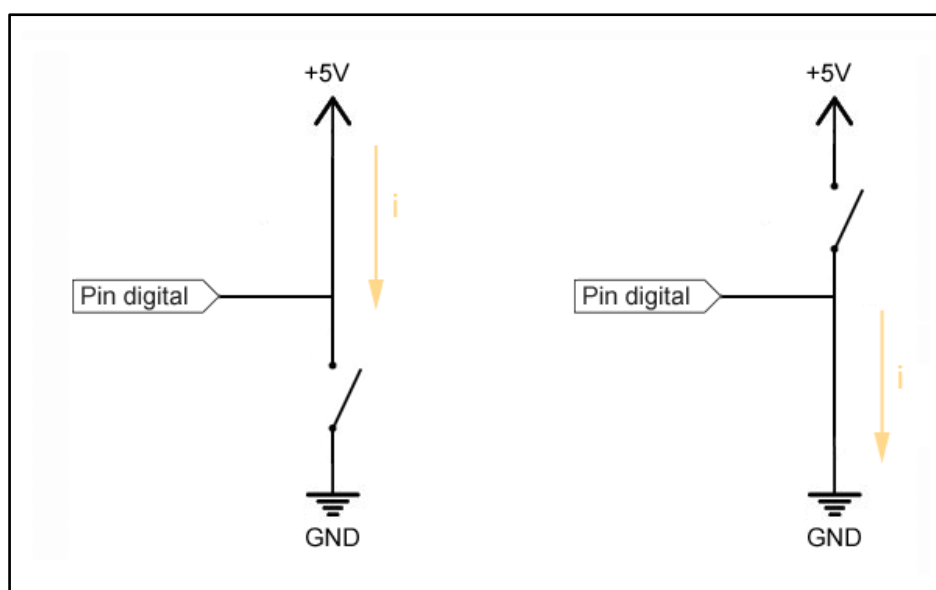
¿Cómo podemos resolver este estado de indeterminación?

Bueno, esto nos lleva directamente a nuestro segundo intento.

Doble conexión

Lo siguiente que podríamos pensar es conectar el PIN a dos referencias de tensión, alternadas en función del estado del interruptor:

- Para medir un valor LOW al accionar el interruptor, podemos conectar el PIN de forma fija a 5V, y a 0V a través del interruptor. Con el interruptor abierto leeríamos HIGH, y al cerrar el interruptor se forzaría 0V en el PIN, por lo que leeríamos LOW.
- Para medir un valor HIGH al accionar el interruptor, podemos conectar el PIN de forma fija a 0V, y a 5V a través del interruptor. Con el interruptor abierto leeríamos LOW, y al cerrar el interruptor se forzaría 5V en el PIN, por lo que leeríamos HIGH.



Esquema doble conexión

¿Cuál es el problema de este montaje?

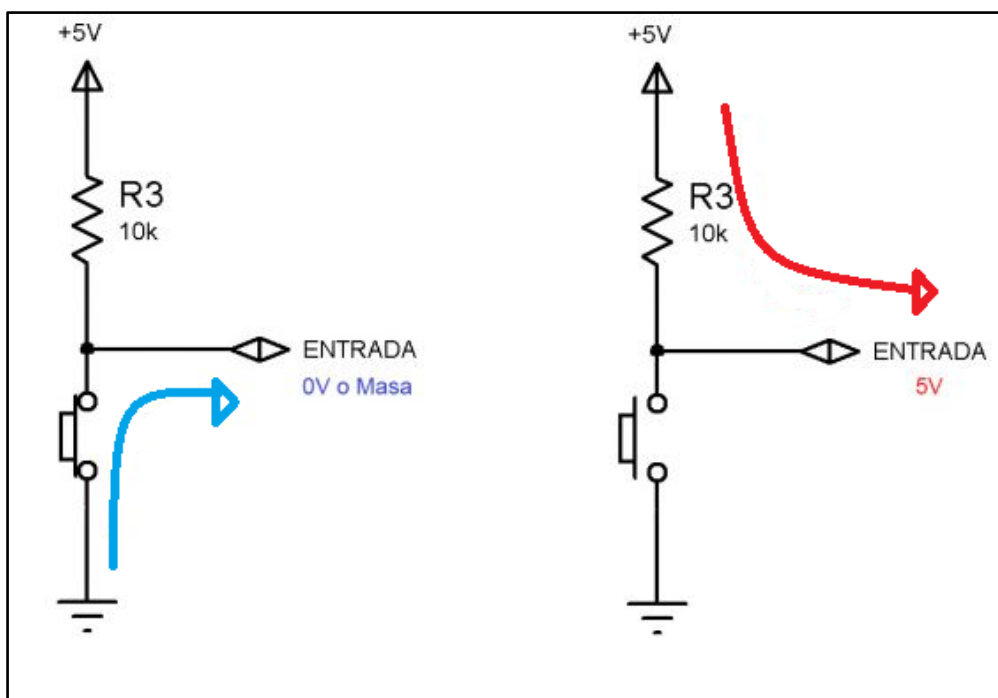
Pues que al pulsar el interruptor estamos conectando directamente los valores de 0V y 5V, lo que significa que estamos **causando un cortocircuito**. Esto provocaría el paso de un elevado de corriente y un rápido calentamiento de componentes y conductores.

Montaje correcto con resistencias Pull-Up, Pull-Down

Para resolver el montaje correctamente necesitaremos la presencia de dos nuevas amigas, las resistencias de Pull Down y Pull Up

Resistencia de Pull-Up

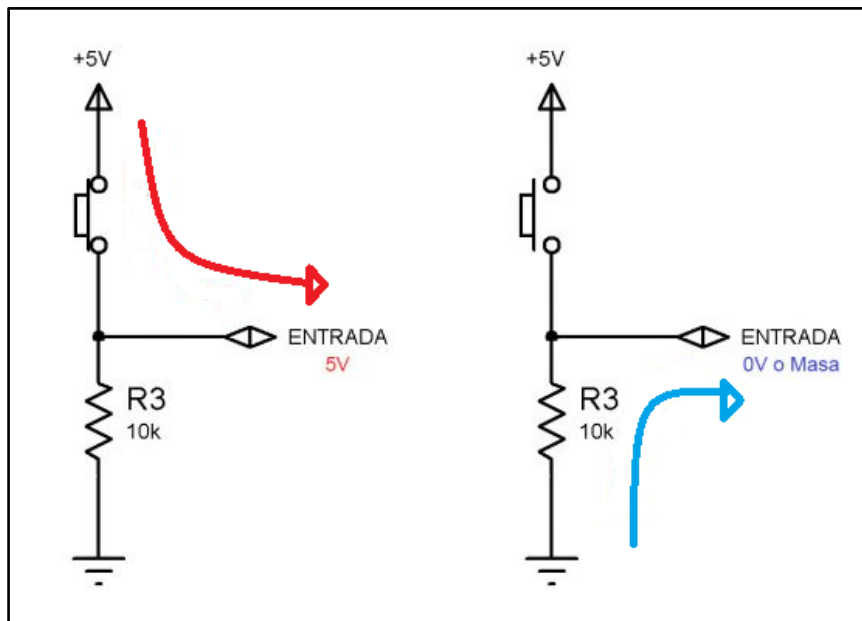
Cuando configuramos una entrada digital con una resistencia de Pull-Up, estamos asegurando, que en todo momento vamos a tener una señal ALTA hasta el momento en el que se produzca una pulsación del interruptor, en este momento la entrada digital quedará conectada directamente a masa, o lo que es lo mismo, a un nivel de tensión 0.



Esquema de conexión Pull-Up

Resistencia de Pull-Down

Su funcionamiento es idéntico al de las resistencias de Pull-Up, pero invirtiendo la lógica, esta configuración asegurará un nivel lógico de 0 voltios hasta que sea pulsado el pulsador.



Esquema de conexión Pull-Down

En Arduino podemos aplicar una resistencia Pull-Up en las entradas digitales a través de código. Simplemente poniendo un pin en modo `INPUT_PULLUP` con la instrucción `pinMode`. Resulta muy sencillo y nos evitará tener que añadir más elementos a nuestros circuitos.

- `pinMode(ledPin, INPUT_PULLUP);` //Con la función `pinMode` se puede usar la resistencia interna de Pull-Up del Arduino

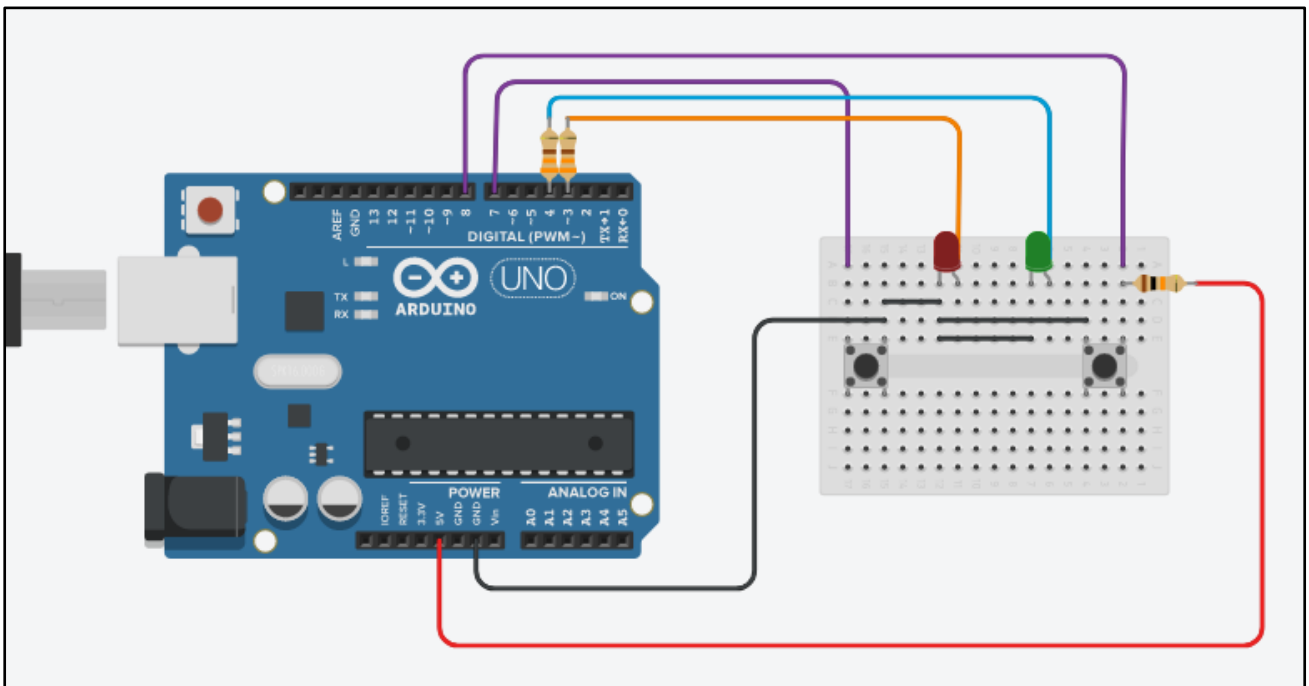


IMPORTANTE: el pin 13 del Arduino **NO** se puede utilizar como entrada o INPUT, ya que tiene un led asociado como OUTPUT a la misma. Si bien no nos va a marcar como un error cuando lo queramos usar como INPUT, el problema lo vamos a tener cuando intentemos utilizar ese pulsador que SIMEPRE esta a LOW por la resistencia y el LED.

Circuito Propuesto

El circuito propuesto consta de un pulsador conectado a un Arduino usando la resistencia interna de Pull-Up del Arduino y otro pulsador con resistencia de Pull-Down externa, cada pulsador deberá apagar un led por separado, el objetivo del circuito es verificar el correcto funcionamiento de la resistencia de Pull-Up interna del Arduino para ello usaremos:

- 1 Arduino uno
- 1 Resistencia de 10 K Ω
- 2 Diodos Leds color rojo, y verde
- 2 Resistencias 330 Ω
- 1 Protoboard
- 2 Pulsadores



Resistencias Pull-Up en Arduino

Solución de la Programación

Programa: Resistencias Pull-Up en Arduino

```
const int led_rojo = 3;
const int led_verde = 4; //Declarar Variables y Constantes
const int ent_pulsador = 7;
const int ent_pullDown = 8; //en este caso variables tipo
entero
int lectura;
int lectura_pull;

void setup()
{
  pinMode(led_rojo, OUTPUT);
  pinMode(led_verde, OUTPUT); // Asignación de puertos como entrada/salida
  pinMode(ent_pulsador, INPUT_PULLUP); //Se utiliza la resistencia
  pinMode(ent_pullDown, INPUT); // interna de Pull_Up
}

void loop()
{
  lectura = digitalRead(ent_pulsador); //Se lee el estado del pulsador
  lectura_pull = digitalRead(ent_pullDown); //Se lee el estado del pulsador/resistencia
  digitalWrite(led_rojo, lectura); //Se escribe en el puerto 3 el estado de la
  //la lectura del pulsador
  digitalWrite(led_verde, lectura_pull); ///Se escribe en el puerto 4 el estado de
  //la lectura del pulsador/resistencia de
  //Pull-Up
}
```