

Pines digitales

Los pines del Arduino se pueden configurar como entradas o salidas. Este documento explica el funcionamiento de los pines en esos modos. Si bien el título de este documento hace referencia a los pines digitales, es importante tener en cuenta que la gran mayoría de los pines analógicos Arduino (Atmega) pueden configurarse y utilizarse exactamente de la misma manera que los pines digitales.

Propiedades de los pines configurados como ENTRADA

Los pines de Arduino (Atmega) se configuran de manera predeterminada en las entradas, por lo que no es necesario que se declaren explícitamente como entradas con `pinMode()` cuando se usan como entradas. Se dice que los pines configurados de esta manera están en un estado de alta impedancia. Los pines de entrada hacen demandas extremadamente pequeñas en el circuito que están muestreando, equivalente a una resistencia en serie de 100 megaohmios delante del pin. Esto significa que se necesita muy poca corriente para mover el pin de entrada de un estado a otro, y puede hacer que los pines sean útiles para tareas tales como implementar [un sensor táctil capacitivo](#), leer un LED como [fotodiodo](#) o leer un sensor analógico con un esquema como [RCTime](#).

Sin embargo, esto también significa que los pines configurados como `pinMode(pin, INPUT)` sin nada conectado a ellos, o con cables conectados a ellos que no están conectados a otros circuitos, informarán cambios aparentemente aleatorios en el estado del pin, captando ruido eléctrico del entorno, o acoplando capacitivamente el estado de un pin cercano.

Resistencias Pullup con pines configurados como ENTRADA

A menudo es útil dirigir un pin de entrada a un estado conocido si no hay ninguna entrada presente. Esto se puede hacer agregando una resistencia pullup (a +5V), o una resistencia pulldown (resistencia a tierra) en la entrada. Una resistencia de 10K es un buen valor para una resistencia pullup o pulldown.

Propiedades de los pines configurados como INPUT_PULLUP

Hay 20K resistencias pullup integradas en el chip Atmega a las que se puede acceder desde el software. Se accede a estas resistencias pullup incorporadas configurando el `pinMode()` como `INPUT_PULLUP`. Esto invierte efectivamente el comportamiento del modo ENTRADA, donde ALTO significa que el sensor está apagado y BAJO significa que el sensor está encendido.

El valor de este pullup depende del microcontrolador utilizado. En la mayoría de las placas basadas en AVR, el valor está garantizado entre 20kΩ y 50kΩ. En el Arduino Due, está entre 50kΩ y 150kΩ. Para conocer el valor exacto, consulte la hoja de datos del microcontrolador en su placa.

Al conectar un sensor a un pin configurado con `INPUT_PULLUP`, el otro extremo debe estar conectado a tierra. En el caso de un interruptor simple, esto hace que el pin lea ALTO cuando el interruptor está abierto, y BAJO cuando se presiona el interruptor.

Las resistencias pullup proporcionan suficiente corriente para iluminar tenuemente un LED conectado a un pin que se ha configurado como entrada. Si los LED en un proyecto parecen estar funcionando, pero muy tenuemente, esto es probablemente lo que está sucediendo.

Las resistencias pullup están controladas por los mismos registros (ubicaciones de memoria de chip interno) que controlan si un pin es ALTO o BAJO. En consecuencia, un pin que está configurado para tener resistencias pullup activadas cuando el pin es una ENTRADA, tendrá el pin configurado como ALTO si el pin se cambia a una SALIDA con `pinMode ()`. Esto también funciona en la otra dirección, y un pin de salida que se deja en un estado ALTO tendrá las resistencias pullup establecidas si se cambia a una entrada con `pinMode ()`.

Antes de Arduino 1.0.1, era posible configurar los pull-ups internos de la siguiente manera:

```
pinMode (pin, INPUT); // establecer pin para entrada  
  
digitalWrite (pin, ALTO); // enciende las resistencias pullup
```

NOTA: El pin digital 13 es más difícil de usar como entrada digital que los otros pines digitales porque tiene un LED y una resistencia unidos a la placa en la mayoría de las placas. Si habilita su resistencia pull-up interna de 20k, colgará alrededor de 1.7V en lugar de los 5V esperados porque el LED integrado y la resistencia en serie bajan el nivel de voltaje, lo que significa que siempre regresa BAJO. Si debe usar el pin 13 como entrada digital, establezca su `pinMode ()` en INPUT y use una resistencia externa desplegable.

Propiedades de los pines configurados como SALIDA

Se dice que los pines configurados como SALIDA con `pinMode ()` están en un estado de baja impedancia. Esto significa que pueden proporcionar una cantidad sustancial de corriente a otros circuitos. Los pines Atmega pueden generar (proporcionar corriente positiva) o hundirse (proporcionar corriente negativa) hasta 40 mA (miliamperios) de corriente a otros dispositivos / circuitos. Esto es suficiente corriente para iluminar un LED (no olvide la resistencia en serie), o hacer funcionar muchos sensores, por ejemplo, pero no suficiente corriente para hacer funcionar la mayoría de los relés, solenoides o motores.

Los cortocircuitos en los pines Arduino, o intentar ejecutar dispositivos de alta corriente desde ellos, pueden dañar o destruir los transistores de salida en el pin, o dañar todo el chip Atmega. A menudo, esto dará como resultado un pin "muerto" en el microcontrolador, pero el chip restante seguirá funcionando adecuadamente. Por esta razón, es una buena idea conectar los pines de SALIDA a otros dispositivos con resistencias de 470Ω o 1k, a menos que se requiera un consumo máximo de corriente de los pines para una aplicación en particular.