

LUIS LLAMAS

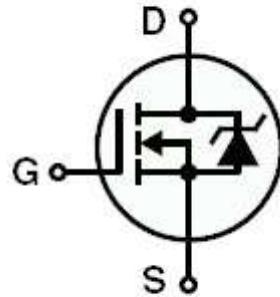
Ingeniería, informática y diseño

TUTORIALES ARDUINO TUTORIALES ARDUINO BASICOS

CONTROLAR GRANDES CARGAS CON ARDUINO Y TRANSISTOR MOSFET

18 JULIO, 2016

[f](#) [t](#) [p](#) [g+](#) [in](#) [e](#)



En una entrada anterior vimos como emplear un transistor BJT como un interruptor controlado eléctricamente, para alimentar cargas a niveles de tensión e intensidad muy superiores a los que podríamos alimentar con las salidas digitales o las salidas PWM de Arduino.

En esta entrada veremos cómo conseguir el mismo comportamiento empleando un transistor MOSFET. Los MOSFET disponen de ventajas en ciertos aspectos frente a los BJT pero, para lo que nos ocupa en esta entrada, la mayor ventaja es que nos permiten manipular

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#) Aceptar Reject

Durante esta entrada haremos constante referencia a los transistores BJT y su funcionamiento, que tomaremos como base para explicar los transistores MOSFET. Por lo que si aún no estáis familiarizados con su funcionamiento, es buen momento para repasar la entrada de transistores BJT.

Anuncio:

También existen placas comerciales con **MOSFET IRF520N** para conectar directamente a Arduino.

¿QUÉ ES UN TRANSISTOR MOSFET?

Un transistor MOSFET es un dispositivo electrónico ampliamente utilizado en la electrónica moderna. Por ejemplo, es el principal integrante de la mayoría de procesadores, donde llega a haber millones de transistores en cada integrado.

Los MOSFET son una subfamilia de la familia de transistores FET (field effect transistor). Existen muchas otras subfamilias de FET, como los JFET (junction field effect transistor), los CMOS, y los TFT.

Como los transistores BJT, los transistores FET son útiles en múltiples aplicaciones. Algunas de las principales son **actuar como amplificador y actuar como interruptor controlado eléctricamente**. En esta entrada estamos interesados en esta última función del transistor.

Otro similitud con los transistores BJT es que existen multitud de modelos de transistores FET, cada uno con sus propias características. Igualmente se presentan en una gran variedad de integrados, por lo que no es posible distinguir a simple vista las características de un transistor, debiendo consultar su Datasheet para saber sus características.



ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

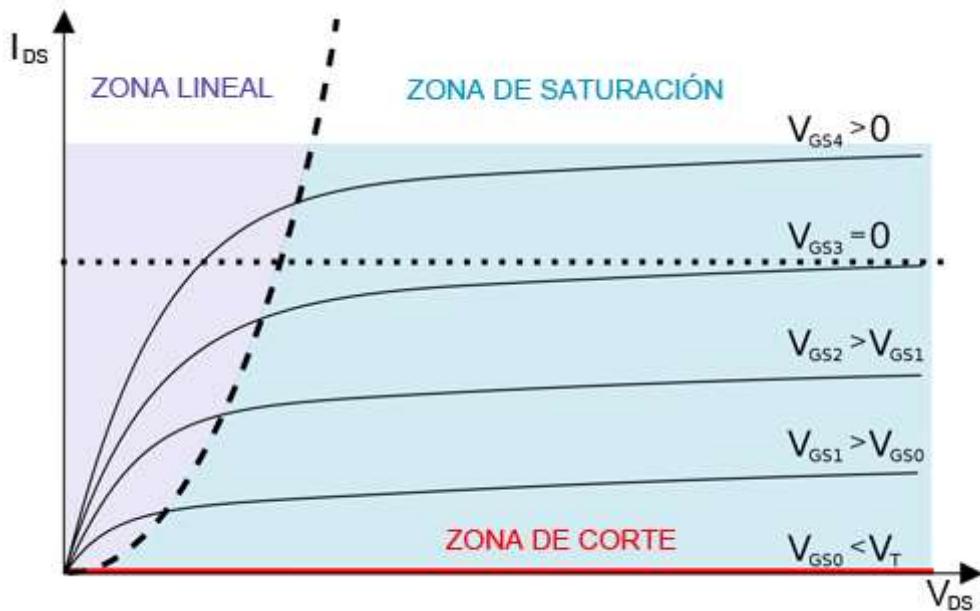
[Leer más](#)

Aceptar

Reject

Al igual que los transistores BJT **un transistor FET dispone de tres terminales**, aunque sus nombres son diferentes de los que encontramos en los transistores BJT.

- **Gate (puerta)**, similar a la base del BJT
- **Source (fuente)**, similar al emisor del BJT
- **Drain (drenaje)**, similar al colector del BJT



También de forma similar a los transistores BJT **los transistores FET presentan 3 modos de funcionamiento**, aunque la zona activa del BJT es sustituida por la zona lineal u ohmánica. (Y no solo es un simple cambio de nombre, realmente ambas zonas presentan funcionamientos muy diferentes)

- **Corte**, el transistor de comporta como un circuito abierto entre Source y Drain
- **Saturación**, se comporta como un cortocircuito entre Source y Drain
- **Zona lineal**, se comporta como una resistencia de valor variable

Y de forma similar al caso de los transistores BJT, para encender una carga estamos interesados en hacer funcionar el FET como un interruptor controlado eléctricamente, para lo cual **usaremos los modos en los modos corte y saturación**, evitando la zona lineal.

Sin embargo, los transistores FET tienen importantes diferencias con los transistores BJT. En primer lugar, su funcionamiento no se basa en la unión de materiales semiconductores, sino

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#) Aceptar Reject

la base. Por tanto, **los FET son dispositivos controlados por tensión**, mientras que los BJT son dispositivos controlados por corriente.

Otra gran ventaja es que **en modo de saturación los transistores MOSFET se comportan como una resistencia de muy pequeño valor**, mientras que los transistores BJT siempre imponían una caída de tensión. Esto permite a los transistores MOSFET manejar enormes cargas con poca disipación de energía.

Existen más diferencias entre ambos dispositivos que, aunque no influyan directamente en esta entrada, conviene repasar.

Los MOSFET son dispositivos mucho más simétricos que los BJT (el comportamiento de Drain a Source y viceversa es similar). Además, disponen de una alta impedancia desde Gate (del orden de $100M\Omega$), lo cual es una gran ventaja a la hora de combinarlos para formar circuitos digitales.

En general los tiempos de conmutación son más rápidos que los BJT. Además generan menos ruido y son menos sensibles a la temperatura.

Finalmente, los transistores MOSFET son más sencillos de fabricar, además de poder simular el comportamiento de una resistencia con ellos. Esto les convierte en grandes candidatos para formar integrados y procesadores.

DETERMINAR EL ESTADO DEL MOSFET

Si recordamos la entrada de transistores BJT, un transistor BJT se comporta como un amplificador lineal de la corriente en la base (I_b) y la intensidad del colector (I_c), con un cierto factor hFE , lo que da lugar a una serie de modelos y cálculos.

En el caso de los MOSFET, la intensidad que atraviesa el drenaje (I_d) tiene una relación cuadrática con la tensión entre Gate y Source (V_{gs}). Dada esta relación cuadrática, en lugar de plantear un modelo matemático os recomendamos **consultar las gráficas del Datasheet** para determinar el punto de operación del transistor.

No obstante, para poder usar y, sobre todo, para poder escoger correctamente un modelo de transistor para nuestro montaje, debemos entender dos aspectos del funcionamiento de los

FET

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#)

punto de saturación, la resistencia Rds disminuye drásticamente (a este valor de Rds en saturación se denomina frecuentemente Rdson)

Por otro lado, **el Gate del transistor se comporta como un condensador**. Es decir, el transistor necesita absorber una cierta cantidad de carga eléctrica (equivalentemente, de intensidad durante un cierto tiempo) para cambiar de modo de trabajo.

$$G_{capacitancia} = I_d \cdot Tiempo$$

Resumiendo, para llegar a saturar el MOSFET necesitaremos dos cosas

- ! Superar un umbral en la tensión en la Gate (Vgs)
- ! Proporcionar la carga suficiente para que el MOSFET sature

Estos dos hechos son fundamentales para entender el funcionamiento de un MOSFET y poder escoger el modelo apropiado para nuestro montaje, y cuando resulta preferible usar un transistor BJT.

POTENCIA DISIPADA EN EL MOSFET

Otro aspecto fundamental para el diseño de circuitos con MOSFET es la potencia soportada por el mismo, dado que el transistor deberá ser capaz de disipar esta energía sin dañarse.

La potencia soportada por el MOSFET es la resistencia del mismo, por el cuadrado de la corriente que lo atraviesa.

$$P_{MOSFET} = R_{ds} \cdot I_d^2$$

De forma cuantitativa, para cada una de las zonas de trabajo posible

- ➔ **En la zona corte** Rds es la determinada por Vgs, pero Ics es nula, por lo que la potencia disipada es cero
- ➔ **En la zona saturación** Ids es grande, pero Rdson es muy pequeña, por lo que la potencia

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#) Aceptar Reject

Por tanto, **un transistor MOSFET solo tiene que soportar una potencia realmente importante en la zona lineal**, especialmente a medida que nos acercamos al punto de saturación. Una vez saturado, el MOSFET es capaz de soportar una gran intensidad, con pequeña disipación de energía.

Por tanto, si queremos usar el MOSFET como interruptor **evitaremos la zona lineal** porque en esta zona la energía disipada es elevada, la cual se convierte en calor y en un incremento de temperatura que puede dañar el transistor.



Si la potencia disipada por el MOSFET es demasiado elevada, necesitaremos añadir un disipador.

USAR UN MOSFET CON ARDUINO

Llegamos a la parte de elegir un MOSFET que funcione correctamente con Arduino, y aquí es donde nuestra relación idílica con los transistores FET se complica.

Cuando vamos a elegir un MOSFET podemos dejarnos deslumbrar por los grandes valores de intensidad nominal (20-60A), que los hacen parecer mucho mejores que un **transistor BJT o un par Darlington** (0.5-4A).

Además, aunque algo más caros que los BJT, los MOSFET siguen siendo dispositivos baratos. Encontraremos todo tipo de precios, ya que hay multitud de modelos con diferentes características. Pero, en general, podemos decir que un rango habitual de precios es entre 0,10 a 0,60€.

Sin embargo para elegir un MOSFET para Arduino debemos tener en cuenta que **las salidas de Arduino operan a la tensión nominal de 5V** (o 3.3V, en función del modelo) y proporcionan **una corriente máxima de 40 mA**, siendo el máximo recomendable 20 mA.

La mayoría de MOSFET tiene una tensión nominal V_{gs} de 10V, por lo que a los 5V que proporciona la salida de Arduino la intensidad máxima que puede proporcionar MOSFET **es mucho menor a su intensidad nominal**. Incluso en algunos modelos de transistores puede no ser suficiente para saturar el MOSFET. Esta situación lógicamente, es incluso peor en el caso de modelos de Arduino de 3.3V.

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#) Aceptar Reject

MODELOS DE MOSFET PARA ARDUINO

Existen muchos modelos de MOSFET disponibles pero **no todos son recomendados para usar directamente con un procesador como Arduino** por las limitación de tensión e intensidad en sus salidas.

Modelos habituales de MOSFET son el IRF520, IRF530 e IRF540, con una intensidad nominal Id de, respectivamente, 9.2A, 14A y 28A. Sin embargo, al emplear estos transistores con un Arduino y una Vgs de 5V, los valores de Id caen a 1A, 2A, y 11A.

Por otro lado la serie N, IRF520n, IRF530n e IRF540n, con una intensidad nominal Id de 9.7A, 17A y 33A, al conectarlos directamente con Arduino y una tensión Vgs de 5V, los valores de Id caen a 3A, 11A y 12A, mejor que los anteriores pero muy lejos de ser espectacular.

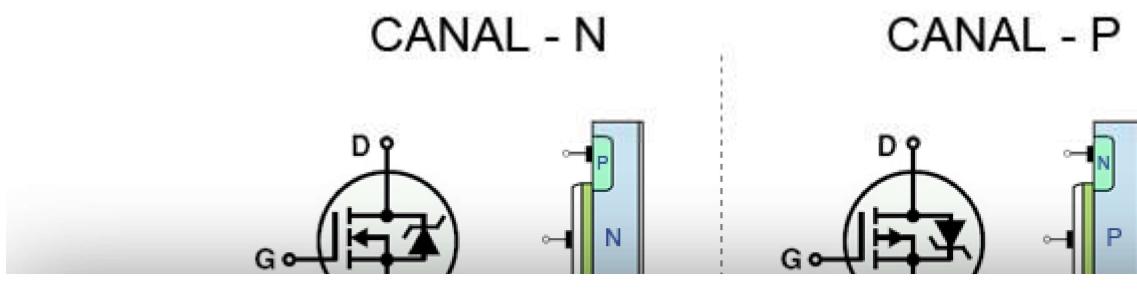
Para resolver esto, existe un tipo especial de condensadores MOSFET llamados **transistores de nivel lógico** (logic level), diseñados específicamente para conmutar a las bajas tensiones típicas en TTL. Por contra, precio es algo superior a los MOSFET habituales.

Así, la serie de transistores lógicos IRL520, IRL530 y IRL540 saturan sin problemas a 5V proporcionando una Id, respectivamente, de 9.2A, 15A, y 28A.

Pero no son los únicos transistores MOSFET lógicos disponibles. Entre los muchos modelos encontramos el IRLZ44 que proporciona una Id de 50A, o el IRLB3034PbF que proporciona una brutal Id de 190A

ESQUEMA DE MONTAJE

De forma similar a los transistores BJT, que se presentaban en dos subfamilias llamadas NPN y PNP, existen dos variantes de transistores FET **llamados de Canal N y Canal P**.



ESTA WEB USA COOKIES

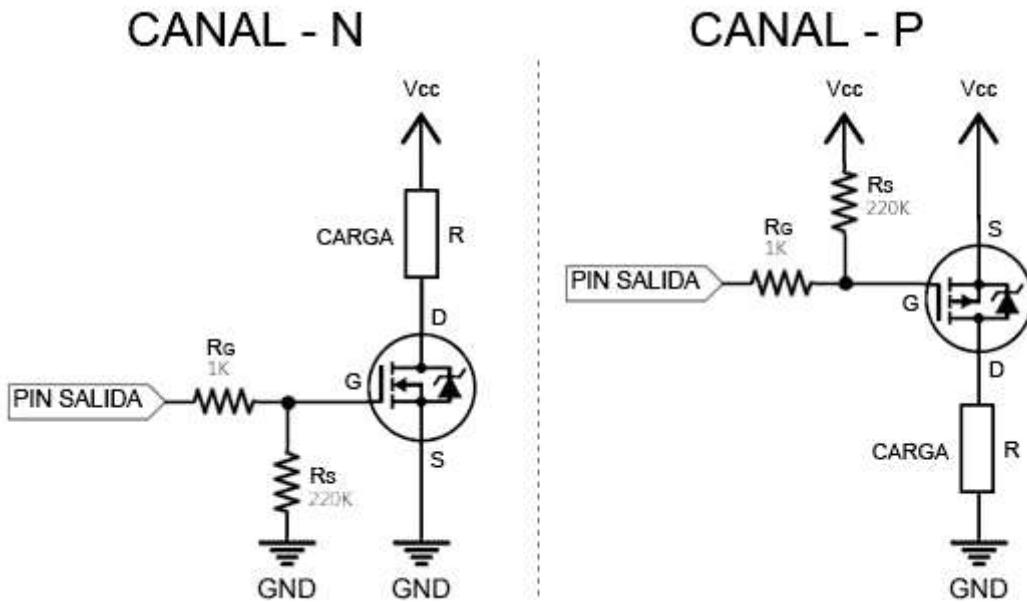
Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#)

Aceptar

Reject

El funcionamiento de ambos variantes es análogo, pero determina la posición en el montaje.



Regla nemotécnica: El transistor Canal P se pone en la parte "Positiva" del circuito. El transistor Canal N se pone en la parte "Negativa" del circuito

Las dos resistencias empleadas en el montaje son necesarias para el correcto funcionamiento del sistema, y cumplen funciones diferentes.

Por un lado R_g , la resistencia en Gate, sirve para limitar la corriente que "demanda" Gate. Valores más altos suponen menores intensidades, y por tanto menor consumo en Arduino. Por contra, disminuir el valor de la resistencia favorece las transiciones más rápidas, con lo que el transistor pasa menos tiempo en la zona lineal, y se calienta menos en las comutaciones. Valores habituales son 470 a 4k7.

Por otro lado, R_s simplemente pone el transistor a un estado conocido (GND) cuando el Pin está en un estado indeterminado (alta impedancia), por ejemplo, durante el arranque del programa, que podrían provocar encendidos y apagados del MOSFET. Un valor alto de resistencia, de 100K a 1M, es suficiente para poner el Gate a tierra.

MOSFET CON CARGAS INDUCTIVAS

ESTA WEB USA COOKIES

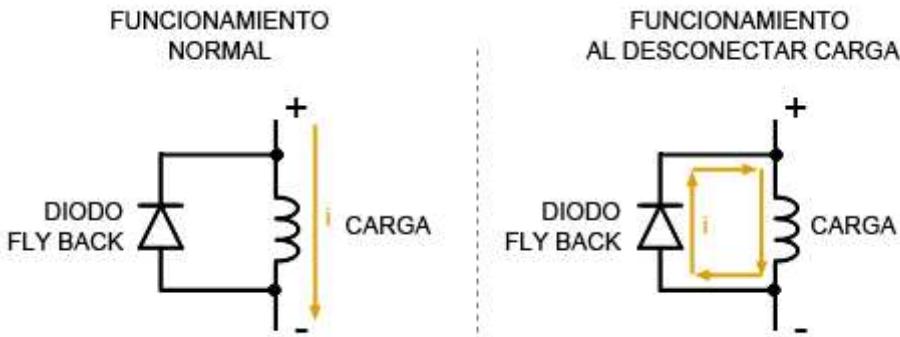
Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#)

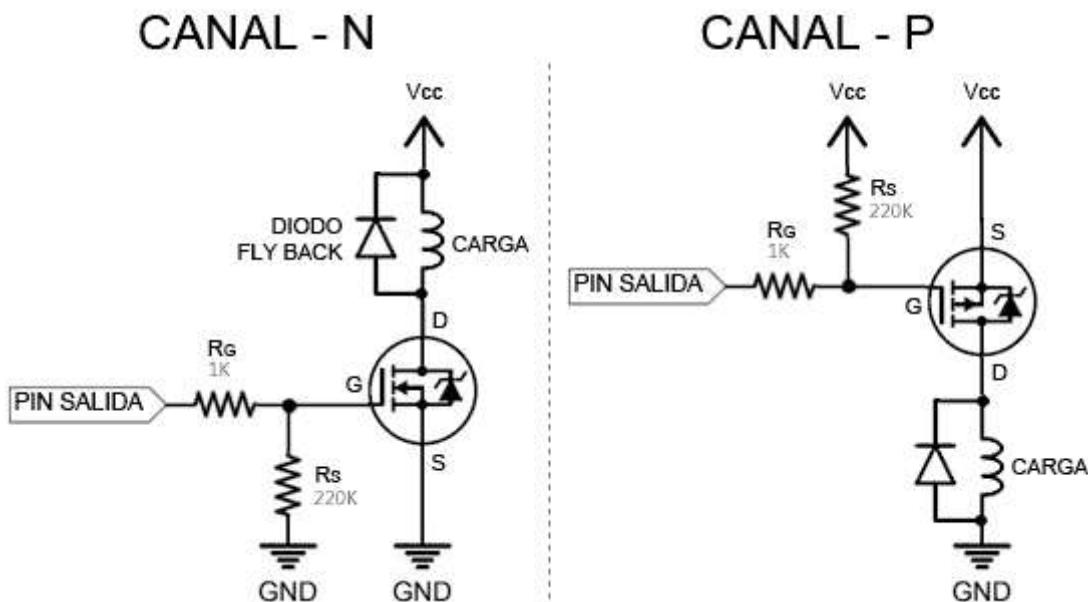
Aceptar

Reject

Este diodo proporciona un recorrido de mínima resistencia, que permite disipar las corrientes inducidas producidas por el campo magnético de la carga inductiva cuando esta es desconectada de la corriente, y que podrían dañar al transistor o a Arduino..



Por tanto, para cargas inductivas e incorporando el diodo de Flyback, el montaje sería el siguiente.



MOSFET CON SALIDAS PWM

En la entrada de los BJT vimos que resultaban apropiados para generar una señal PWM amplificada, sin más necesidad que conectarlos a una salida analógica de Arduino.

Podríamos pensar que los transistores MOSFET son incluso mejores para generar salidas

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#)

Aceptar

Reject

siquiera en los modelos de transistores de nivel lógico.

Esto es debido a que tanto la tensión y, sobre todo, **la intensidad proporcionada por una salida de Arduino no es suficiente para saturar rápidamente al MOSFET**. Esto hace que el transistor pase excesivo tiempo en la zona lineal, aumentando las pérdidas.

Para lidiar con estas limitaciones podemos:

- ➔ Limitarnos a cargas pequeñas, que demanden intensidades menores a la nominal
- ➔ Plantearnos sustituirlo por un par Darlington
- ➔ Hacer una etapa previa de amplificación de potencia con un BJT (lo vemos a continuación)



Antes de usar un MOSFET en un PWM conviene reflexionar un poco antes de hacer el montaje, y tener siempre en cuenta la potencia y temperatura soportada por el transistor.

USAR UNA ETAPA BJT COMO DRIVER

Hemos visto a lo largo de la entrada que las limitaciones en tensión y corriente que imponen las salidas de Arduino (y en general de cualquier procesador o autómata) nos obligan a reducir las cargas que podemos alimentar con un MOSFET, o a emplear transistores especiales de nivel lógico.

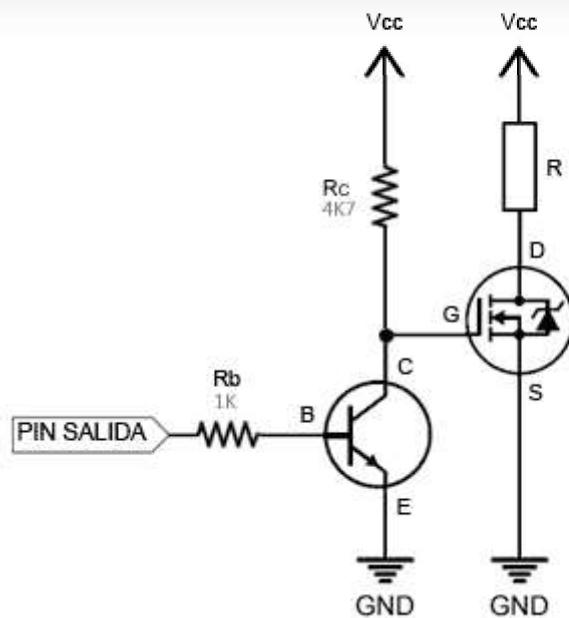
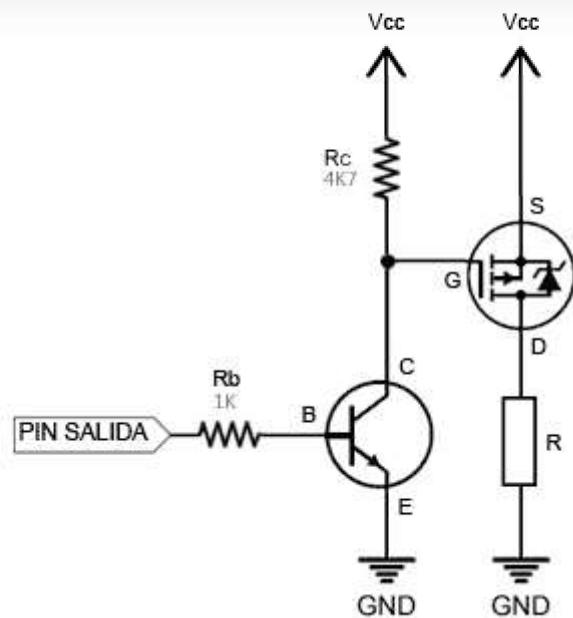
Esta situación es incluso peor en el caso de las salidas PWM, dado que la limitación de corriente obliga al MOSFET a estar más tiempo en la zona lineal, aumentando la potencia disipada y su temperatura.

Una forma de eliminar estas restricciones es emplear una etapa de preamplificación, entre la salida de Arduino y el MOSFET. Esta etapa, o driver, puede ser un circuito formado por un simple BJT de pequeño tamaño (N2222, BC337 o similar).

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#) Aceptar Reject

CANAL - N**CANAL - P**

Esta etapa recibe la salida de Arduino, y proporciona al MOSFET la tensión e intensidad que necesita para saturar completamente, y con una velocidad de conmutación rápida.

Con este tipo de circuito aprovechar por completo el MOSFET, permitiendo proporcionar grandes intensidades incluso en salidas PWM. Pero, por contra, añade componentes y complejidad a los montajes.

Por tanto, no existe una solución única para escoger cuando es más apropiado un BJT, un MOSFET directamente, o un MOSFET con preamplificación. Es una decisión de diseño que debéis tomar de forma individual para cada uno de vuestros montajes, según todo lo que hemos visto en esta entrada.



Si te ha gustado esta entrada y quieres leer más sobre Arduino puedes consultar la sección **Tutoriales de Arduino**

Anuncio:

Arduino,Arduino basico,Digitales,I/O,Salidas

ESTA WEB USA COOKIES

Utilizamos cookies propias y de terceros para mejorar nuestros servicios. Más información en

[Leer más](#) [Aceptar](#) [Reject](#)