

**Realización de un controlador para un motor BLDC basado en una tarjeta ARDUINO****Objetivos:**

- Desarrollar un algoritmo en ARDUINO que controle el funcionamiento de un motor BLDC en modo de tracción y en modo de frenado regenerativo en las dos direcciones de giro
- Ejecutar el algoritmo utilizando un puente de transistores trifásico y comprender el funcionamiento del mismo

**Material:**

- Motor de tracción BLDC.
- Placa controladora con puente trifásico con transistores MOSFET y drivers IRS2003 controlado por una tarjeta ARDUINO PRO-MICRO (compatible con ARDUINO LEONARDO).

**Instrumentación necesaria:**

- Fuente de alimentación.
- Osciloscopio.
- Multímetro o sensor electrónico de corriente.
- Batería de 36 V

**Introducción.**

En esta práctica se debe realizar un controlador para un motor BLDC utilizando una tarjeta con un puente de transistores trifásico y un controlador ARDUINO PRO-MICRO. Los transistores MOSFET del puente están gobernados por tres drivers IRS2003 que generan las señales de control de sus terminales de puerta (consultar el esquemático del circuito y el datasheet del IRS2003). Los MOSFET de la parte alta del puente son también de canal N, por lo que el driver debe generar en su puerta las tensiones necesarias para que se polaricen en zona ohmica. De esa función se encargan los condensadores de "bootstrap" C1, C4 y C6.

Cada driver tiene dos señales de entrada, una para el transistor "alto" y la otra para el transistor "bajo", que deben ser convenientemente programadas para generar las señales de control de los transistores. Las señales de los transistores altos están conectadas a salidas PWM del ARDUINO. En total hay seis señales de control y tres salidas de potencia, una para cada fase del motor.

El principio de funcionamiento debe ser el siguiente:

Se debe implementar una función combinacional para cada transistor de potencia del puente cuyas entradas son las 6 posibles combinaciones de los sensores de efecto Hall (las funciones combinacionales son diferentes si el giro es en un sentido o en el opuesto y si está en modo motor o en frenado regenerativo).

Cada vez que hay un cambio en alguno de estos sensores se debe activar una interrupción y actualizar el estado de los transistores del puente trifásico. Para ello, las entradas de los sensores de efecto Hall están conectadas a los pines de interrupción de ARDUINO: D2, D3 y D7.

La velocidad se regula mediante el ciclo de trabajo de las señales PWM aplicadas a los transistores altos del puente trifásico. El valor máximo permitido del ciclo de trabajo debe ser del 80%, para que los transistores de la parte alta del puente se polaricen correctamente, ya que tiene que existir un intervalo de tiempo suficiente para que se carguen los condensadores de "bootstrap", que establecen la tensión de conducción de los mismos.

La frecuencia de las señales PWM debe fijarse en torno a 32 KHz para que no sea audible (\*).

**Realización de un controlador para un motor BLDC basado en una tarjeta ARDUINO****Desarrollo de la práctica****1º.- Algoritmo de control para sentido de giro directo (marcha adelante).**

Especificaciones del Algoritmo:

- **Entradas digitales:** 3 entradas digitales para los sensores de efecto Hall con posibilidad de generar interrupciones (tener en cuenta que las salidas de estos sensores son de colector abierto).
- **Salidas digitales:** 6 salidas digitales para controlar los drivers del puente. Tres de ellas, las de los transistores altos, tienen capacidad para generar señales PWM.
- **Entradas analógicas:** Una entrada analógica (la A0 o la A1), que está conectada al potenciómetro utilizado como acelerador. Cuando la entrada tenga un nivel de tensión inferior al 5% de su valor máximo, el motor debe permanecer parado (ciclo de trabajo de las señales PWM al 0%). Cuando dicha entrada alcance su valor máximo, el ciclo de trabajo de las señales de control será del 80%. Entre estos dos valores la relación entre el valor de la entrada y el ciclo de trabajo debe ser lineal.

Evaluación del algoritmo:

1. Desarrollar el algoritmo y cargarlo en la tarjeta ARDUINO.
2. Conectar los sensores de efecto Hall y de potencia del motor a la tarjeta controladora.
3. Regular el potenciómetro de la tarjeta controladora al mínimo.
4. Conectar la fuente de alimentación a la tarjeta con el cable correspondiente y regular su tensión a un valor de 30V. El motor debe permanecer parado y la intensidad que proporcione la fuente debe ser muy pequeña. Si esto no es así, comunicarlo al profesor para que revise la tarjeta controladora.
5. Regular el potenciómetro de la tarjeta controladora a su valor medio y comprobar que el motor gira suavemente en el sentido correcto. Es posible que para que arranque haya que hacer girar a su eje con la mano en el primer instante. Comprobar también que se acelera cuando se gira el acelerador en sentido horario y que se para antes de que llegue a su tope mínimo.

**2º.- Algoritmo de control para sentido de giro directo e inverso.**

Especificaciones del algoritmo:

- Hay que añadir una nueva entrada digital (la D20 o la D21, accesibles en el conector J15) para seleccionar entre marcha adelante y marcha atrás.
- Si el motor está girando en un sentido y se cambia el valor de la señal anterior, el motor debe pararse totalmente antes de empezar a girar en sentido contrario.
- Cuando el sentido de giro seleccionado sea la marcha atrás, el ciclo de trabajo de las señales PWM debe ser del 20% cuando el potenciómetro de aceleración esté al máximo y debe tener un comportamiento lineal entre el 5% de su valor máximo (motor parado por debajo de él) y dicho valor máximo.

Evaluación del algoritmo.

- Seguir el mismo procedimiento que en el apartado 1º y comprobar que el motor se para antes de comenzar a girar en sentido contrario.

**3º.- Algoritmo de control para frenado regenerativo.**

Especificaciones del algoritmo:

- Hay que añadir una nueva entrada digital (la que no se haya utilizado anteriormente) para habilitar el frenado regenerativo, cuyo funcionamiento debe ser el siguiente:
- Si la señal está activada, se habilitará el modo de frenado regenerativo cuando el acelerador baje del 5% de su valor máximo o antes de detenerse cuando se cambie el sentido de giro. El ciclo de trabajo en este modo será siempre del 30%
- Si la señal no está activada, el funcionamiento será como en el apartado 2º.

Evaluación del algoritmo:

**Realización de un controlador para un motor BLDC basado en una tarjeta ARDUINO**

1. Desconectar la alimentación del puente trifásico de la fuente de alimentación y conectarla a la batería de 36 V a través de un multímetro configurado para medir intensidad en la entrada de 10 A o a través de un sensor electrónico de intensidad de 5A.
2. Poner a girar el motor en marcha adelante y comprobar que el multímetro muestra un valor positivo de intensidad. Si no es así, intercambiar sus terminales.
3. Activar la entrada de frenado regenerativo y posteriormente llevar el potenciómetro del acelerador al mínimo. Observar que, durante unos instantes, la medida de la intensidad pasa de positiva a negativa hasta que el motor se detenga.
4. Repetir el paso anterior, pero, en lugar de llevar el acelerador al mínimo, mantener su posición y cambiar el sentido de giro. El motor debe pararse regenerando y volver a arrancar en sentido contrario.

(\*) Poner en el setup las siguientes instrucciones:

```
TCCR1B = TCCR1B & 0b11111000 | 0x01; // frecuencia en los pines 9 y 10: aproximadamente 31250 Hz  
TCCR3B = TCCR3B & 0b11111000 | 0x01; // frecuencia en el pin 5: aproximadamente 31250 Hz  
TCCR4B = TCCR4B & 0b11111000 | 0x01; // frecuencia en los pines 6 y 13: aproximadamente 31250 Hz
```