

Control de motor de alta potencia BTS7960 con Arduino

Introducción

Qué es un BTS7960?

Dentro del mundo de los microcontroladores se pueden hacer muchas cosas por supuesto, y una de las cosas más habituales es controlar motores. Los microcontroladores como Arduino no pueden hacerlo directamente y para eso existen los controladores de potencia, también llamados "Drivers".

Existen muchos tipos de controladores de motores para Arduino. El BTS7960 es uno de ellos y es capaz de proporcionar altas corrientes para motores bastante potentes. El BTS7960 es capaz de proporcionar nada menos que hasta 43 Amperios que normalmente es más que suficiente para casi cualquier aplicación.



- 1: RPWM
- 2: LPWM
- 3: R_EN
- 4: L_EN
- 5: R_IS
- 6: L_IS
- 7: VCC
- 8: GND



El BTS7960 recibe las señales digitales desde un Arduino, que puede ser de 5V o 3.3V ya que soporta los dos voltaje. Por otro lado, recibe en su entrada el voltaje desde una fuente de alimentación externa que alimentará el motor que utilices con él.

El BTS7960 tiene unas características bastante decentes para su precio tan bajo:

- Alimentación: 5.5 a 27V
- Salida auxiliar: 5V
- Corriente en reposo: 7 uA a 25 ° C.
- Capacidad de PWM de hasta 25 kHz combinada con activación libre.
- Protección de sobrecorriente
- Bloqueo de sobretensión
- Intensidad máxima: 43 Amperios
- Circuito digital de control con entradas de nivel lógico
- Velocidad de giro ajustable por PWM (hasta 25 KHz)
- Dimensiones: 50x50x43 mm

El BTS7960 dispone de una salida auxiliar (VCC, GND) que proporciona 5V regulados para poder alimentar placas externas como Arduino.

Descripción de pines BTS7960

El controlador de motores BTS7960 solo tiene 8 pines y se define de ésta manera:

1. **R_PWM:** Pulso de control derecho
2. **L_PWM:** Pulso de control izquierdo
3. **R_EN:** Activa salida derecha
4. **L_EN:** Activa salida izquierda
5. **R_IS:** Salida medición de corriente derecha
6. **L_IS:** Salida medición de corriente izquierda
7. **VCC:** Salida de alimentación 5V
8. **GND:** Negativo

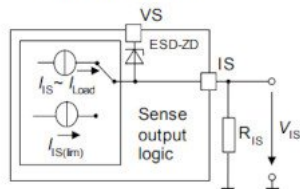
¿Cómo funciona un BTS7960?

El controlador de motores BTS7960 es muy sencillo ya que incorpora dos transistores MOSFET BTS7960 formando un puente en H junto a su circuitería básica. No dispone de microcontrolador interno ni ningún tipo de lógica más que un buffer 74HC244. Es un integrado de potencia puro y duro tal y como su puede ver en su esquema básico:

BTS 7960B
P-TO-263-7



Normal operation:
current sense mode



Fault condition:
error flag mode

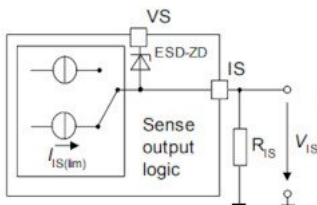
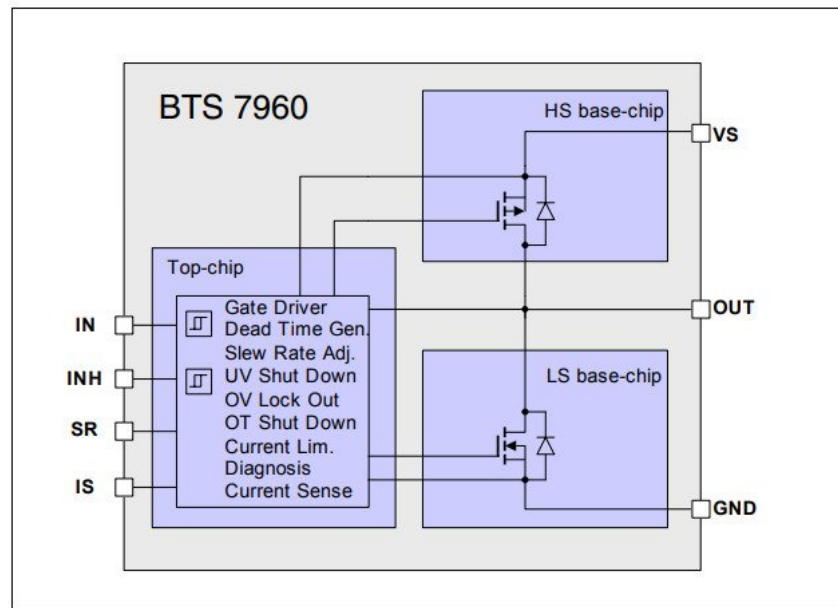
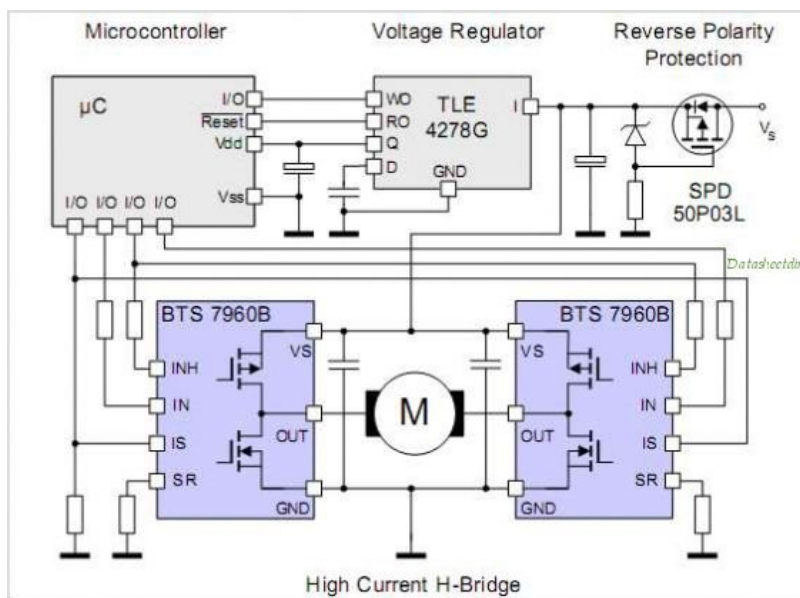


Figure 1 Block Diagram



También dispone de pines específicos de sensado de corriente para indicar fallo en el motor.

Un ejemplo de aplicación según la hoja de datos del fabricante es algo como esto, combinando dos integrados BTS7960 para formar un completo puente H de potencia:



Esquema de montaje

El esquema de conexiones para poder controlar un motor con un BTS7960 se compone de lo siguiente:

- Un módulo BTS7960
- Fuente de alimentación externa
- Un motor
- Microcontrolador de control (por ejemplo Arduino, pero puede ser cualquier otro o incluso una Raspberry Pi)

Primero conectaremos la fuente de alimentación externa y el motor:



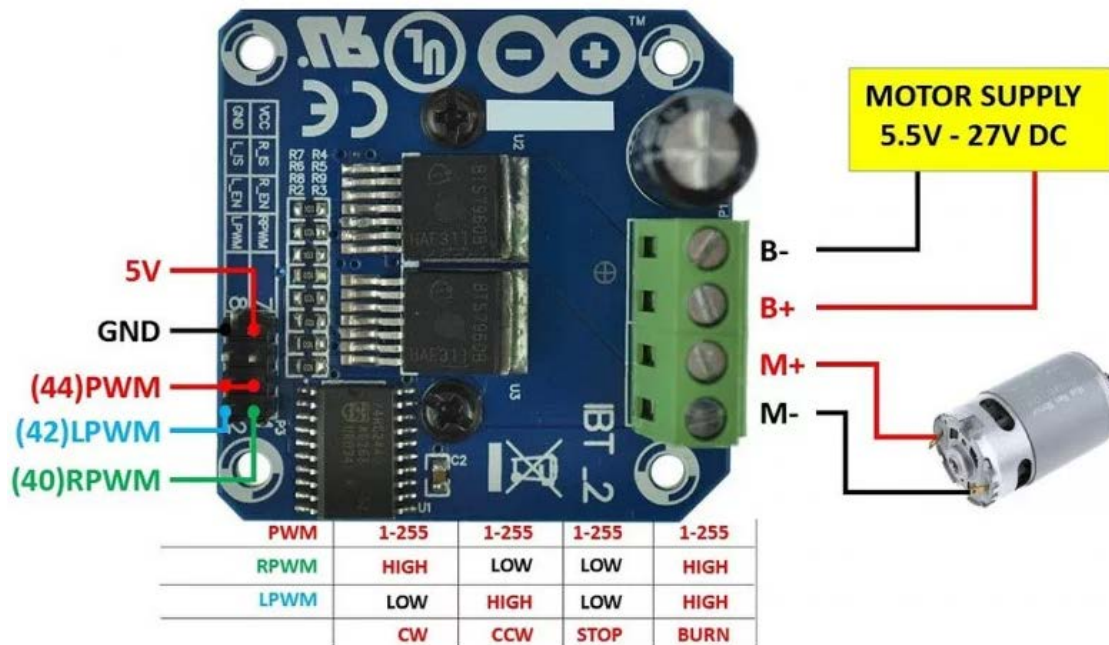
Las conexiones serán:

- **B-**: GND fuente
- **B** : VCC fuente
- **M** : VCC Motor
- **M-**: GND Motor

Si vas utilizar un motor de alta potencia, procura que los cables estén correctamente dimensionados para evitar que se calienten o generen pérdidas. No utilices cables de tipo DuPont, usa cable adecuado a la intensidad máxima que vas a utilizar, dejando un poco de margen. Puedes guiarte con ésta tabla, donde puedes ver la relación entre el diámetro del cable y la intensidad máxima que soportará. Un buen cable para empezar, será de entre 1 y 1.5mm de sección (AWG18/16) y soportará entorno a 10 Amperios.

Tabla AWG Según el Amperaje

Sección AWG	Sección mm ²	Corriente (Amperios)
20	0.5	3
18	1	7
16	1,5	10
14	2,5	15
12	4	20
10	6	30
8	10	40
6	16	55
4	25	70

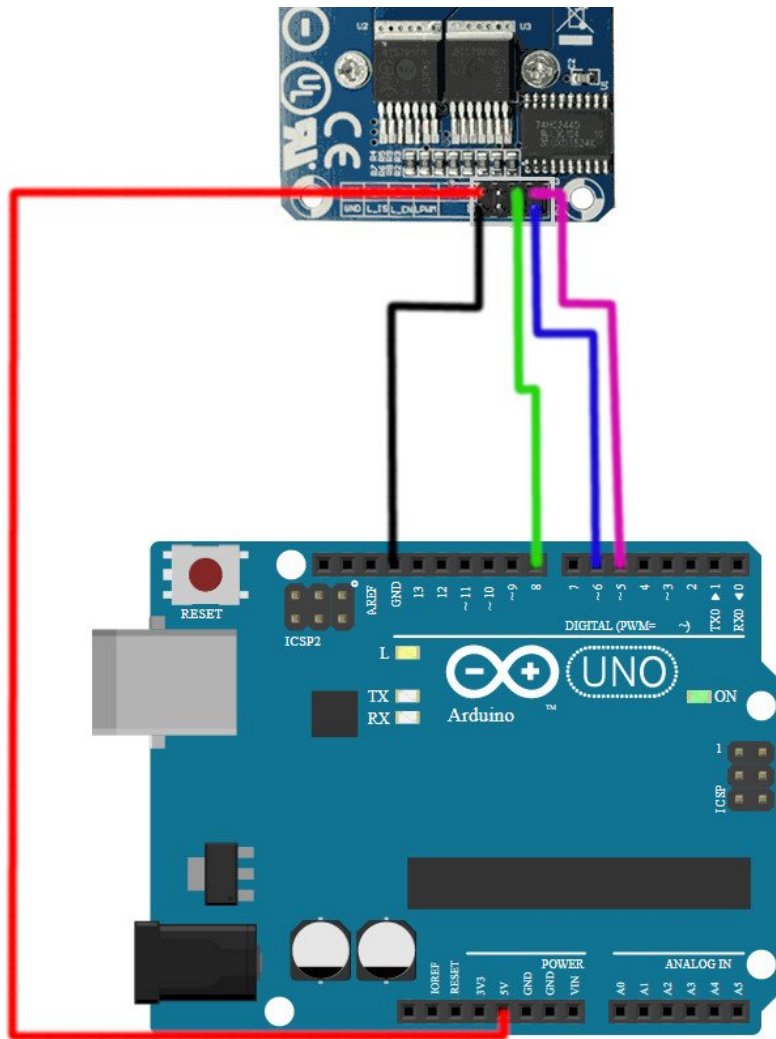


Conexiones y pines de BTS7960

Conexión de los pines BTS7960 a Arduino

En realidad y para hacer el ejemplo más sencillo, solo vamos a utilizar 3 pines digitales de Arduino para controlar el motor tanto en velocidad como en sentido de giro. Las demás irán conectadas al nivel lógico que le corresponde.

- **Pin R_IS** -> Sin conectar
- **Pin R_EN** -> Arduino pin 8
- **Pin R_PWM** -> Arduino 5
- **Pin L_IS** -> Sin conectar
- **Pin L_EN** -> Arduino pin 8
- **Pin L_PWM** -> Arduino 6
- **Pin Vcc** -> Arduino 5V
- **GND** -> Arduino GND



Recuerda que los pines que utilices para los pines **L_PWM** y **R_PWM** deben soportar pulsos **PWM**. En la mayoría de placas Arduino, los pines que soportan esa funcionalidad están marcados con un símbolo ~

Código de ejemplo

Cómo programar BTS7960 con Arduino

El código que vamos a ver ahora es muy sencillo ya que el controlador es muy fácil de manejar. Existen por supuesto librerías para controlar el BTS7960 con Arduino, pero como podrás ver, con una decenas de líneas podremos hacerlo perfectamente. Si aún así lo prefieres, tienes [ésta librería BTS7960 para Arduino](#) en Github.

El código de ejemplo inicializa los pines en la función Setup() y luego en el bucle principal Loop() se activa primero la rotación del motor en un sentido. Se espera 1 segundo y se activa el motor en sentido inverso.

Para el propósito de éste tutorial he fijado una variable con valor fijo para la velocidad. Le puedes cambiar al que prefieras con un valor de entre 0 y 254.

Acerca de la velocidad mínima

Aunque el valor de velocidad comience en 0, todos los motores tienen cierta inercia en el arranque. Vamos, que les cuesta más comenzar el giro que girar de forma sostenida. Por eso es normal que tu motor no comience a girar hasta que la velocidad mínima tenga un valor determinado, por ejemplo 30 o 40 (depende del motor).

```
// Control de motor con BTS7960
// BricoGeek.com

int L_EN = 8;
int R_EN = 8;
int L_PWM = 6;
int R_PWM = 5;

void setup() {
  pinMode(L_EN, OUTPUT);
  pinMode(R_EN, OUTPUT);
  pinMode(L_PWM, OUTPUT);
  pinMode(R_PWM, OUTPUT);

  // Activa la salida
  digitalWrite(R_EN, HIGH);
  digitalWrite(L_EN, HIGH);
}

void loop() {

  int velocidad = 128; // Valor entre 0 y 254

  // Giro derecha
  analogWrite(R_PWM, velocidad);
  analogWrite(L_PWM, 0);
```



```
delay(1000); // Esperamos 1 segundo

// Giro izquierda
analogWrite(R_PWM, 0);
analogWrite(L_PWM, velocidad);

delay(1000); // Esperamos 1 segundo
}
```

Conclusiones

Ahora que ya tienes una mejor conocimiento sobre cómo controlar un motor de corriente continua con Arduino y el BTS7960, puedes comenzar a experimentar. Esto no deja de ser un tutorial básico, pero el BTS7960 tiene funciones interesantes como el control de error a las que seguro le sacarás provecho!