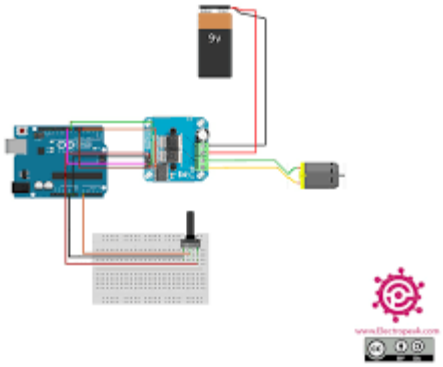
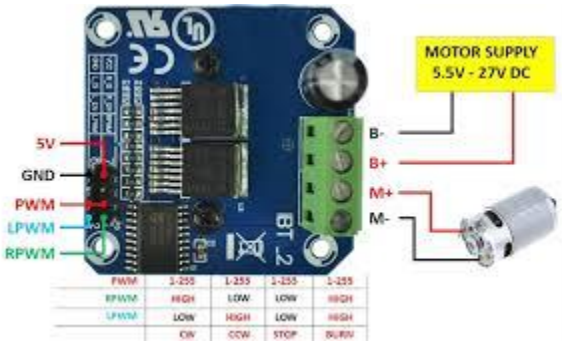
ARDUINO DC MOTOR DRIVER BTS7960B 43A





Arduino DC Motor Driver using BTS7960B (43A)

The BTS7960B is a high-current half-bridge motor driver capable of handling up to 43A. Here's a guide to using it with Arduino:

Hardware Overview

- **Driver IC**: BTS7960B (contains two half-bridges)
- **Max Current**: 43A (peak)
- **Voltage Range**: 5.5V to 27V
- **PWM Frequency**: Up to 25kHz
- **Features**: Overcurrent, overtemperature, undervoltage protection

Wiring Diagram

Copy

Download

Arduino BTS7960B Module

----- ----------

5V VCC (for logic)

GND GND

PWM Pin 1 RPWM (or LPWM)

PWM Pin 2 LPWM (or RPWM)

Digital Pin R_EN (enable right)

ARDUINO DC MOTOR DRIVER BTS7960B 43A

Digital Pin L_EN (enable left)

Motor connects between the BTS7960B output pins (usually labeled "OUT" pins).

Basic Arduino Code

Important Notes

1. **Power Supply**: Use a separate power source for the motor (not from Arduino)
2. **Current Handling**: Ensure your power supply can handle the motor's current requirements
3. **Heat Sink**: For high currents (>10A), a heat sink is recommended
4. **Protection Diodes**: The module should already include these, but verify
5. **Enable Pins**: Must be HIGH for the corresponding side to work

Advanced Usage

For better control, you can implement:

- Speed ramping for smoother starts/stops
- Current sensing (if your module has this feature)
- PID control for precise speed regulation
- Fault condition monitoring

- cpp
- Copy
- Download

```cpp
// Define pins
const int RPWM = 5;  // Right PWM (or forward)
const int LPWM = 6;  // Left PWM (or reverse)
const int R_EN = 7;  // Right enable
const int L_EN = 8;  // Left enable

void setup() {
  pinMode(RPWM, OUTPUT);
  pinMode(LPWM, OUTPUT);
  pinMode(R_EN, OUTPUT);
  pinMode(L_EN, OUTPUT);

  // Enable both sides
  digitalWrite(R_EN, HIGH);
  digitalWrite(L_EN, HIGH);
```

```
}

void loop() {
  // Rotate clockwise (forward)
  analogWrite(RPWM, 150);  // 150/255 speed
  analogWrite(LPWM, 0);
  delay(2000);

  // Rotate counter-clockwise (reverse)
  analogWrite(RPWM, 0);
  analogWrite(LPWM, 150);
  delay(2000);

  // Stop
  analogWrite(RPWM, 0);
  analogWrite(LPWM, 0);
  delay(1000);

  // Accelerate forward
  for(int i = 0; i <= 255; i++) {
    analogWrite(RPWM, i);
    analogWrite(LPWM, 0);
    delay(20);
  }
  delay(1000);
}
```