



Arduino DC Motor Driver with TB6612FNG

The TB6612FNG is a popular dual H-bridge motor driver that can control two DC motors or one stepper motor. Here's a guide to using it with Arduino.

Basic Connections

TB6612FNG Pinout:

[Copy](#)

[Download](#)

VM - Motor voltage (4.5-13.5V)

VCC - Logic voltage (2.7-5.5V)

GND - Ground

A01 - Motor A output 1

A02 - Motor A output 2

B01 - Motor B output 1

B02 - Motor B output 2

PWMA - Motor A PWM input

AIN1 - Motor A input 1

AIN2 - Motor A input 2

PWMB - Motor B PWM input

BIN1 - Motor B input 1

BIN2 - Motor B input 2

STBY - Standby (high to enable)

Arduino Connections Example:

[cpp](#)

[Copy](#)

[Download](#)

// Motor A

const int PWMA = 5; // PWM pin

const int AIN1 = 7; // Direction pin 1

const int AIN2 = 8; // Direction pin 2

// Motor B

const int PWMB = 6; // PWM pin

const int BIN1 = 9; // Direction pin 1

const int BIN2 = 10; // Direction pin 2

const int STBY = 4; // Standby pin

Basic Motor Control Functions

[cpp](#)

[Copy](#)

[Download](#)

void setup() {

// Set all pins as outputs

pinMode(PWMA, OUTPUT);

pinMode(AIN1, OUTPUT);



```
pinMode(AIN2, OUTPUT);
pinMode(PWMB, OUTPUT);
pinMode(BIN1, OUTPUT);
pinMode(BIN2, OUTPUT);
pinMode(STBY, OUTPUT);

digitalWrite(STBY, HIGH); // Disable standby
}

// Motor A control
void motorA(int speed, bool reverse) {
  digitalWrite(AIN1, !reverse);
  digitalWrite(AIN2, reverse);
  analogWrite(PWMA, speed);
}

// Motor B control
void motorB(int speed, bool reverse) {
  digitalWrite(BIN1, !reverse);
  digitalWrite(BIN2, reverse);
  analogWrite(PWMB, speed);
}

// Stop both motors
void stopMotors() {
  digitalWrite(AIN1, LOW);
  digitalWrite(AIN2, LOW);
  digitalWrite(BIN1, LOW);
  digitalWrite(BIN2, LOW);
  analogWrite(PWMA, 0);
  analogWrite(PWMB, 0);
}

void loop() {
  // Example usage
  motorA(200, false); // Motor A forward at speed 200/255
  motorB(150, true); // Motor B backward at speed 150/255
  delay(2000);

  stopMotors();
  delay(1000);
}
```

Important Notes

1. **Power Supply:**

- VM should be connected to your motor power source (4.5-13.5V)
- VCC should be connected to Arduino's 5V for logic
- Make sure all grounds are connected together

2. Standby Mode:

- Pull STBY high to enable the driver
- Pull STBY low to put the driver in standby (motors stop)

3. PWM Frequency:

- The TB6612FNG works well with Arduino's default PWM frequency
- For smoother operation, you can change PWM frequency (advanced)

4. Current Handling:

- Each channel can handle up to 1.2A continuous (3.2A peak)
- Add appropriate capacitors near the power inputs for stability

Advanced Usage

For more precise control, you can create a motor control class:

cpp

Copy

Download

```
class DCMotor {
private:
    int pwmPin, in1Pin, in2Pin;

public:
    DCMotor(int pwm, int in1, int in2) {
        pwmPin = pwm;
        in1Pin = in1;
        in2Pin = in2;

        pinMode(pwmPin, OUTPUT);
        pinMode(in1Pin, OUTPUT);
        pinMode(in2Pin, OUTPUT);
    }

    void drive(int speed) {
        bool reverse = (speed < 0);
        speed = abs(speed);
        if(speed > 255) speed = 255;

        digitalWrite(in1Pin, !reverse);
        digitalWrite(in2Pin, reverse);
        analogWrite(pwmPin, speed);
    }

    void brake() {
        digitalWrite(in1Pin, HIGH);
        digitalWrite(in2Pin, HIGH);
        analogWrite(pwmPin, 255);
    }
};

// Create motor objects
```

```
DCMotor motorA(PWMA, AIN1, AIN2);  
DCMotor motorB(PWMB, BIN1, BIN2);
```

```
void setup() {  
  pinMode(STBY, OUTPUT);  
  digitalWrite(STBY, HIGH);  
}
```

```
void loop() {  
  motorA.drive(200); // Forward  
  motorB.drive(-150); // Reverse  
  delay(2000);  
  
  motorA.brake(); // Quick stop  
  motorB.drive(0); // Coast stop  
  delay(1000);  
}
```

