Control DC and stepper motors with L298N Dual Motor Controller Modules and Arduino

by **tronixlabs** on November 25, 2014

Table of Contents

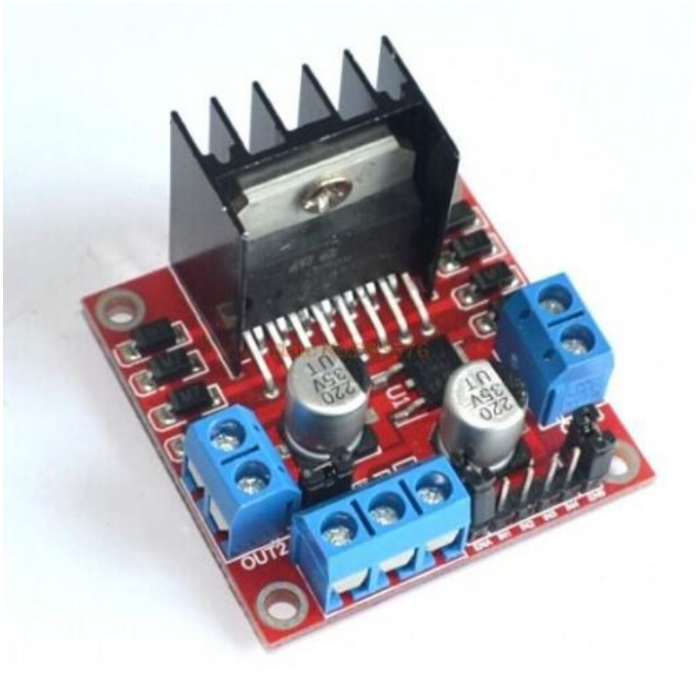
Author:tronixlabs Tronixlabs
Helping you make it with Australia's best value for supported hobbyist electronics from adafruit, DFRobot, Freetronics, Seeed Studio and more!

Intro: Control DC and stepper motors with L298N Dual Motor Controller Modules and Arduino

You don't have to spend a lot of money to control motors with an Arduino or compatible board. After some hunting around we found a neat motor control module based on the L298N H-bridge IC that can allows you to control the speed and direction of two DC motors, or control one bipolar stepper motor with ease.

The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. With the module used in this tutorial, there is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board.

So let's get started!

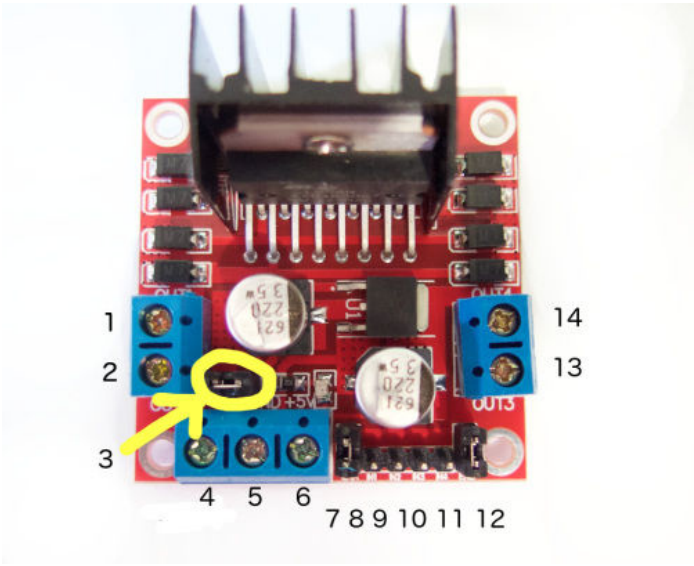


Step 1: Understanding the L298 module connections

First we'll run through the connections, then explain how to control DC motors then a stepper motor. At this point, review the connections on the L298N H-bridge module.

Consider the image – match the numbers against the list below the image:

1. DC motor 1 "+" or stepper motor A+
2. DC motor 1 "-" or stepper motor A-
3. 12V jumper – remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control
13. DC motor 2 "+" or stepper motor B+
14. DC motor 2 "-" or stepper motor B-



Step 2: Controlling DC Motors

To control one or two DC motors is quite easy with the L298N H-bridge module. First connect each motor to the A and B connections on the L298N module.

If you're using two motors for a robot (etc) ensure that the polarity of the motors is the same on both inputs. Otherwise you may need to swap them over when you set both motors to forward and one goes backwards!

Next, connect your power supply – the positive to pin 4 on the module and negative/GND to pin 5. If you supply is up to 12V you can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module.

This can be fed to your Arduino's 5V pin to power it from the motors' power supply. Don't forget to connect Arduino GND to pin 5 on the module as well to complete the circuit. Now you will need six digital output pins on your Arduino, two of which need to be PWM (pulse-width modulation) pins.

PWM pins are denoted by the tilde ("~") next to the pin number, for example in the image of the Arduino Uno's digital pins.

Finally, connect the Arduino digital output pins to the driver module. In our example we have two DC motors, so digital pins D9, D8, D7 and D6 will be connected to pins IN1, IN2, IN3 and IN4 respectively. Then connect D10 to module pin 7 (remove the jumper first) and D5 to module pin 12 (again, remove the jumper).

The motor direction is controlled by sending a HIGH or LOW signal to the drive for each motor (or channel). For example for motor one, a HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and a LOW and HIGH will cause it to turn in the other direction.

However the motors will not turn until a HIGH is set to the enable pin (7 for motor one, 12 for motor two). And they can be turned off with a LOW to the same pin(s). However if you need to control the speed of the motors, the PWM signal from the digital pin connected to the enable pin can take care of it.

This is what we've done with the DC motor demonstration sketch. Two DC motors and an Arduino Uno are connected as described above, along with an external power supply. Then enter and upload the following sketch:

```
// connect motor controller pins to Arduino digital pins
// motor one
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;
void setup()
{
 // set all the motor control pins to outputs
 pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
}
void demoOne()
{
 // this function will run the motors in both directions at a fixed speed
 // turn on motor A
 digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
 // set speed to 200 out of possible range 0~255
 analogWrite(enA, 200);
 // turn on motor B
 digitalWrite(in3, HIGH);
 digitalWrite(in4, LOW);
 // set speed to 200 out of possible range 0~255
 analogWrite(enB, 200);
 delay(2000);
 // now change motor directions
 digitalWrite(in1, LOW);
 digitalWrite(in2, HIGH);
 digitalWrite(in3, LOW);
 digitalWrite(in4, HIGH);
```

```
 delay(2000);
 // now turn off motors
 digitalWrite(in1, LOW);
 digitalWrite(in2, LOW);
 digitalWrite(in3, LOW);
 digitalWrite(in4, LOW);
}
void demoTwo()
{
 // this function will run the motors across the range of possible speeds
 // note that maximum speed is determined by the motor itself and the operating voltage
 // the PWM values sent by analogWrite() are fractions of the maximum speed possible
 // by your hardware
 // turn on motors
 digitalWrite(in1, LOW);
 digitalWrite(in2, HIGH);
 digitalWrite(in3, LOW);
 digitalWrite(in4, HIGH);
 // accelerate from zero to maximum speed
 for (int i = 0; i < 256; i++)
 {
   analogWrite(enA, i);
   analogWrite(enB, i);
   delay(20);
 }
 // decelerate from maximum speed to zero
 for (int i = 255; i >= 0; --i)
 {
   analogWrite(enA, i);
   analogWrite(enB, i);
   delay(20);
 }
 // now turn off motors
 digitalWrite(in1, LOW);
 digitalWrite(in2, LOW);
 digitalWrite(in3, LOW);
 digitalWrite(in4, LOW);
}
void loop()
{
 demoOne();
 delay(1000);
 demoTwo();
 delay(1000);
}
```

So what's happening in that sketch? In the function demoOne() we turn the motors on and run them at a PWM value of 200. This is not a speed value, instead power is applied for 200/255 of an amount of time at once.

Then after a moment the motors operate in the reverse direction (see how we changed the HIGHs and LOWs in thedigitalWrite() functions?). To get an idea of the range of speed possible of your hardware, we run through the entire PWM range in the function demoTwo() which turns the motors on and them runs through PWM values zero to 255 and back to zero with the two for loops.

Finally this is demonstrated in the video on this page – using our well-worn tank chassis with two DC motors.

Step 3: Controlling a Stepper Motor with Arduino and L298N

Stepper motors may appear to be complex, but nothing could be further than the truth. In this example we control a typical NEMA-17 stepper motor that has four wires, as shown in the image on this step.

It has 200 steps per revolution, and can operate at at 60 RPM. If you don't already have the step and speed value for your motor, find out now and you will need it for the sketch.

The key to successful stepper motor control is identifying the wires – that is which one is which. You will need to determine the A+, A-, B+ and B- wires. With our example motor these are red, green, yellow and blue. Now let's get the wiring done.

Connect the A+, A-, B+ and B- wires from the stepper motor to the module connections 1, 2, 13 and 14 respectively. Place the jumpers included with the L298N module over the pairs at module points 7 and 12. Then connect the power supply as required to points 4 (positive) and 5 (negative/GND).

Once again if your stepper motor's power supply is less than 12V, fit the jumper to the module at point 3 which gives you a neat 5V power supply for your Arduino. Next, connect L298N module pins IN1, IN2, IN3 and IN4 to Arduino digital pins D8, D9, D10 and D11 respectively.

Finally, connect Arduino GND to point 5 on the module, and Arduino 5V to point 6 if sourcing 5V from the module. Controlling the stepper motor from your sketches is very simple, thanks to the Stepper Arduino library included with the Arduino IDE as standard.

To demonstrate your motor, simply load the stepper_oneRevolution sketch that is included with the Stepper library. To find this, click the File > Examples > Stepper menu in the Arduino IDE.

Finally, check the value for

```
const int stepsPerRevolution = 200;
```

in the sketch and change the 200 to the number of steps per revolution for your stepper motor, and also the speed which is preset to 60 RPM in the following line:

```
myStepper.setSpeed(60);
```

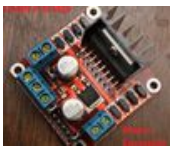
Now you can save and upload the sketch, which will send your stepper motor around one revolution, then back again. This is achieved with the function

```
<p>myStepper.step(stepsPerRevolution); // for clockwise </p><p> myStepper.step(-stepsPerRevolution); // for anti-clockwise
```

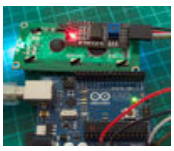
Finally, a quick demonstration of our test hardware is shown in the video on this step.

So there you have it, an easy an inexpensive way to control motors with your Arduino or compatible board. And if you enjoyed this article, or want to introduce someone else to the interesting world of Arduino – check out my book (now in a fourth printing!) " Arduino Workshop".

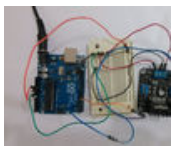
Related Instructables



Arduino Modules - L298N Dual H-Bridge Motor Controller by Reichenstein7



Using PCF8574 backpacks with LCDs and Arduino by tronixlabs



Arduino?? L298N?????? by jeffrey.sun

Getting Started With The MC33932 Dual Motor Shield by JayconSystems

Ultrasonic sensor robot car running with used motorcycle battery with LCD screen by amitray769

Dual H-Bridge - L298 Breakout Board - Homemade by BIGDOG1971

Comments