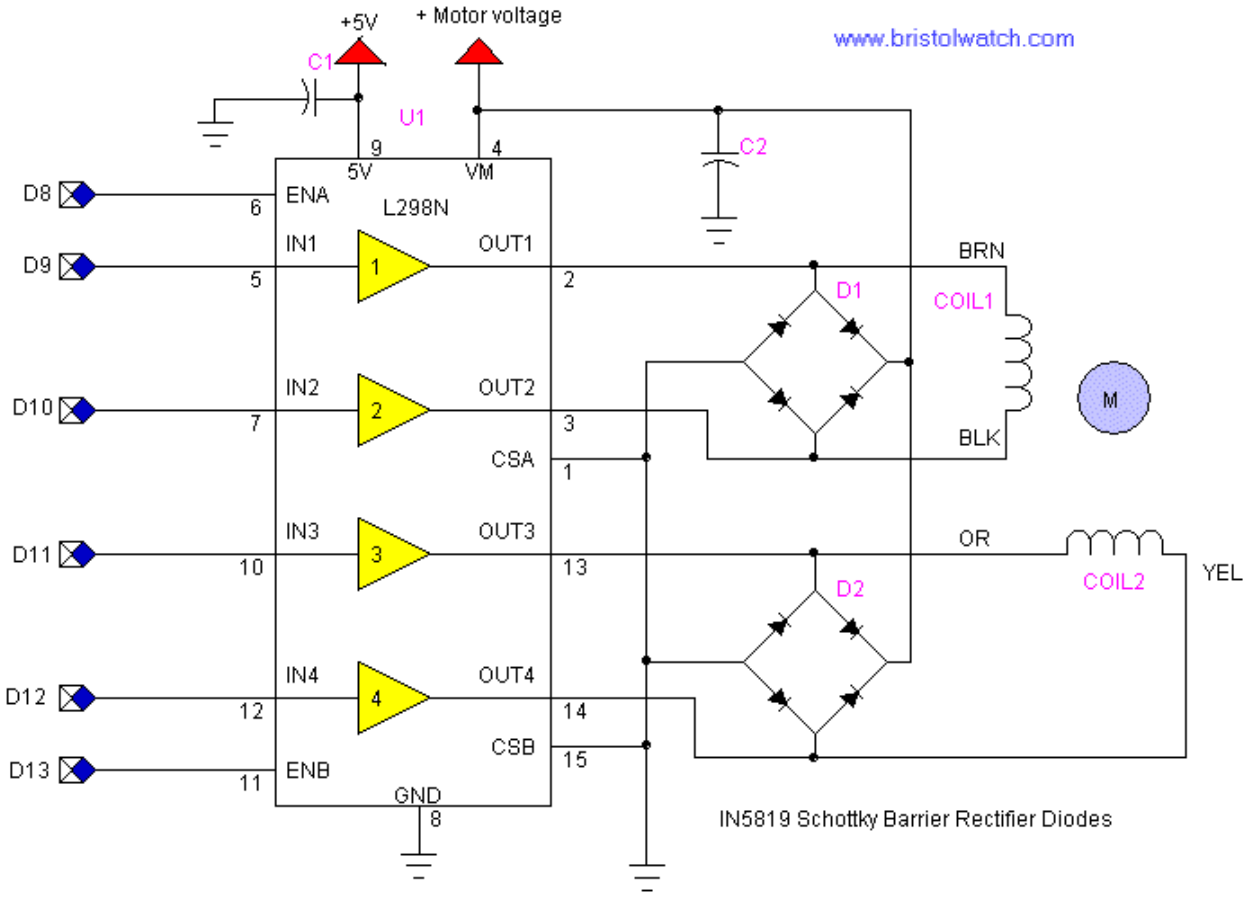

Figure 1 L298N Dual H-Bridge connected to a bi-polar stepper motor.

Connecting the Arduino to a L298N H-Bridge

by Lewis Loflin

Earlier we looked at L298N Motor Controller Theory and Projects to understand the basic operation of the L298N dual H-bridge motor controller. In this section I've connected the L298N to a bi-polar stepper motor and connected it to the ATMega168 aka Arduino micro-controller. Note the motor voltage is the voltage rating of the stepper motor up to 40 volts and a current limit of four amps.

The Arduino series of micro-controllers are an outstanding value for the hobbyist and student to learn the basics of programming and interfacing micro-controllers. In this example in both hardware and programming the code below will operate the stepper motor through the L298N. The steppers I used in the example are all 7.5 degrees per-step and require 48 steps to go 360 degrees. Stepper motors are very accurate and often don't need feedback to tell position.

A bi-polar stepper motor has only two coils and operates by reversing the polarity unlike a unipolar stepper motor that operates by switching four coils on/off.

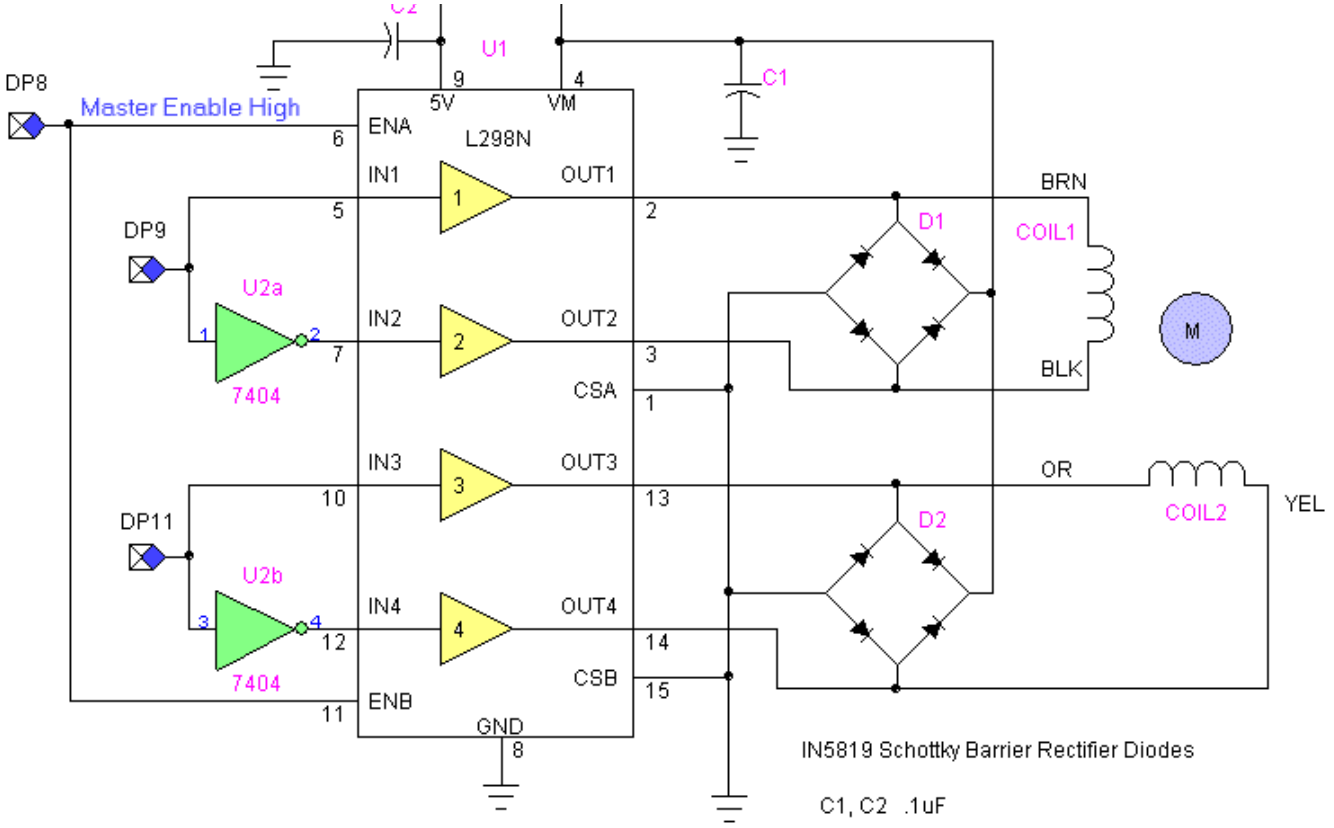

Figure 2

Here we have used only three connections to the Arduino micro-controller to operate the stepper motor. We tied ENA and ENB together and used two inverters from a 74LS04. We could still leave ENA and ENB separate if one wishes. In the program below simply remove any statements relating to ENB, yellow, or black which will free up three digital pins. It would also be a good idea change "#define orange 11" to #define orange 10" then wire digital pin 10 to IN3.


```
/*

 This demo uses a L298N dual H-bridge to operate a
 bi-polar stepper motor. The colors are the actual
 color of the wires on several units removed from
```

```
 color of the wires on several units removed from
 junk printers.

 Numbers such as PM425-048 and PM35s-048.

 This also worked on a unipolor stepper such
 as a Portescape s6mo48 from Ebay leaving the red/green
 wires that went to +12 disconnected and operating at 24 volts.

 */


#define CW 2
#define CCW 3

#define ENA 8
#define ENB 13

#define black 9  // IN1
#define brown 10  // IN2
#define orange 11  // IN3
#define yellow 12  // IN4



void setup()  {

  DDRB = 0x3f; // Digital pins 8-13 output
  PORTB = 0x00; // all outputs DP8-13 set to off

  pinMode(CW, INPUT);
  pinMode(CCW, INPUT);

  digitalWrite(CW, 1); // pullup on
  digitalWrite(CCW,1); // pullup on


}



void loop() {

  if (!digitalRead(CW)) forward(480, 0);
  if (!digitalRead(CCW)) reverse(480, 0);


} // end loop


void reverse(int i, int j) {

  // Pin 8 Enable A Pin 13 Enable B on
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);

  j = j + 10;
  while (1)   {

    digitalWrite(black, 0);
    digitalWrite(brown, 1);
    digitalWrite(orange, 1);
    digitalWrite(yellow, 0);
    delay(j);
    i--;
    if (i < 1) break;


    digitalWrite(black, 0);
    digitalWrite(brown, 1);
    digitalWrite(orange, 0);
    digitalWrite(yellow, 1);
    delay(j);
```

```
    delay(j);
    i--;
    if (i < 1) break;


    digitalWrite(black, 1);
    digitalWrite(brown, 0);
    digitalWrite(orange, 0);
    digitalWrite(yellow, 1);
    delay(j);
    i--;
    if (i < 1) break;

    digitalWrite(black, 1);
    digitalWrite(brown, 0);
    digitalWrite(orange, 1);
    digitalWrite(yellow, 0);
    delay(j);
    i--;
    if (i < 1) break;
  }

  // all outputs to stepper off
  digitalWrite(ENA, LOW);
  digitalWrite(ENB, LOW);

}  // end reverse()



void forward(int i, int j) {

  // Pin 8 Enable A Pin 13 Enable B on
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);

  j = j + 10;
  while (1)   {

    digitalWrite(black, 1);
    digitalWrite(brown, 0);
    digitalWrite(orange, 1);
    digitalWrite(yellow, 0);
    delay(j);
    i--;
    if (i < 1) break;



    digitalWrite(black, 1);
    digitalWrite(brown, 0);
    digitalWrite(orange, 0);
    digitalWrite(yellow, 1);
    delay(j);
    i--;
    if (i < 1) break;

    digitalWrite(black, 0);
    digitalWrite(brown, 1);
    digitalWrite(orange, 0);
    digitalWrite(yellow, 1);
    delay(j);
    i--;
    if (i < 1) break;

    digitalWrite(black, 0);
    digitalWrite(brown, 1);
    digitalWrite(orange, 1);
    digitalWrite(yellow, 0);
    delay(j);
    i--;
    if (i < 1) break;
```

```
  }

  // all outputs to stepper off
  digitalWrite(ENA, LOW);
  digitalWrite(ENB, LOW);

}  // end forward()
```

Figure 3

Shown above is a pre-assembled board I bought off Ebay for $8 with shipping. This included power connectors, diodes, LED indicators, and even a 5-volt regulator. This is in my opinion the smart way to go to save time, money, and effort.

- **ATMEGA168 Arduino Micro Controller Projects**
- **Using Hall Effect Switches and Sensors**
- **Basic Transistor Driver Circuits for Micro-Controllers**
- **Opto-Isolated Transistor Drivers for Micro-Controllers**
- **Build a H-Bridge Motor Control with Power MOSFETS**

- **Build a 12AV6 Vacuum Tube AM Radio**
- **Coils for Highly Selective Crystal Radio**
- **Adding a Push-Pull Output Stage to a LM386 Audio Amplifier**

Added January 2012: PICAXE Micro-controller Projects!

The PICAXE series of micro-controllers rank as the easiest and most cost effective way to use Microchip processors. I wanted

an easier and less expensive way to introduce my students to the "PIC" micro-controller. Here I hope to get those starting out past poorly written literature and lack of simple working code examples.

- Exploring the PICAXE Micro-Controller
- Understanding Micro-Controller Input/Output Ports
- Using the 74HC165 Shift Register with the PICAXE Micro-Controller
- Connecting the 74HC595 Shift Register to PICAXE Micro-controller
- Using 7-Segment Displays with the PICAXE Micro-Controller
- Potentiometers and Analog-to-Digital Conversion with the PICAXE
- PWM Motor Speed Control and the PICAXE Micro-Controller
- Connecting the PICAXE to the DS1307 Real Time Clock
- Connecting the PICAXE to an External EEPROM (24LC08)
- Connecting a Servo to a PICAXE
- Connecting the TLC548 ADC to the PICAXE
- Connecting the AD5220 Digital Potentiometer to the PICAXE