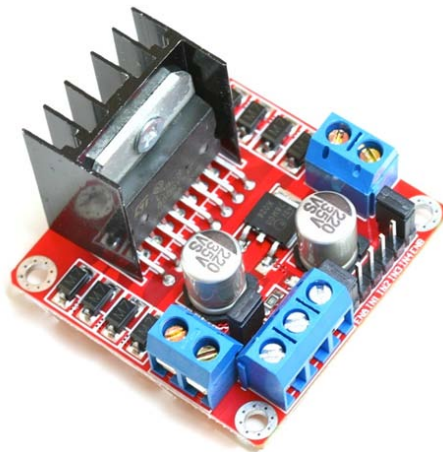


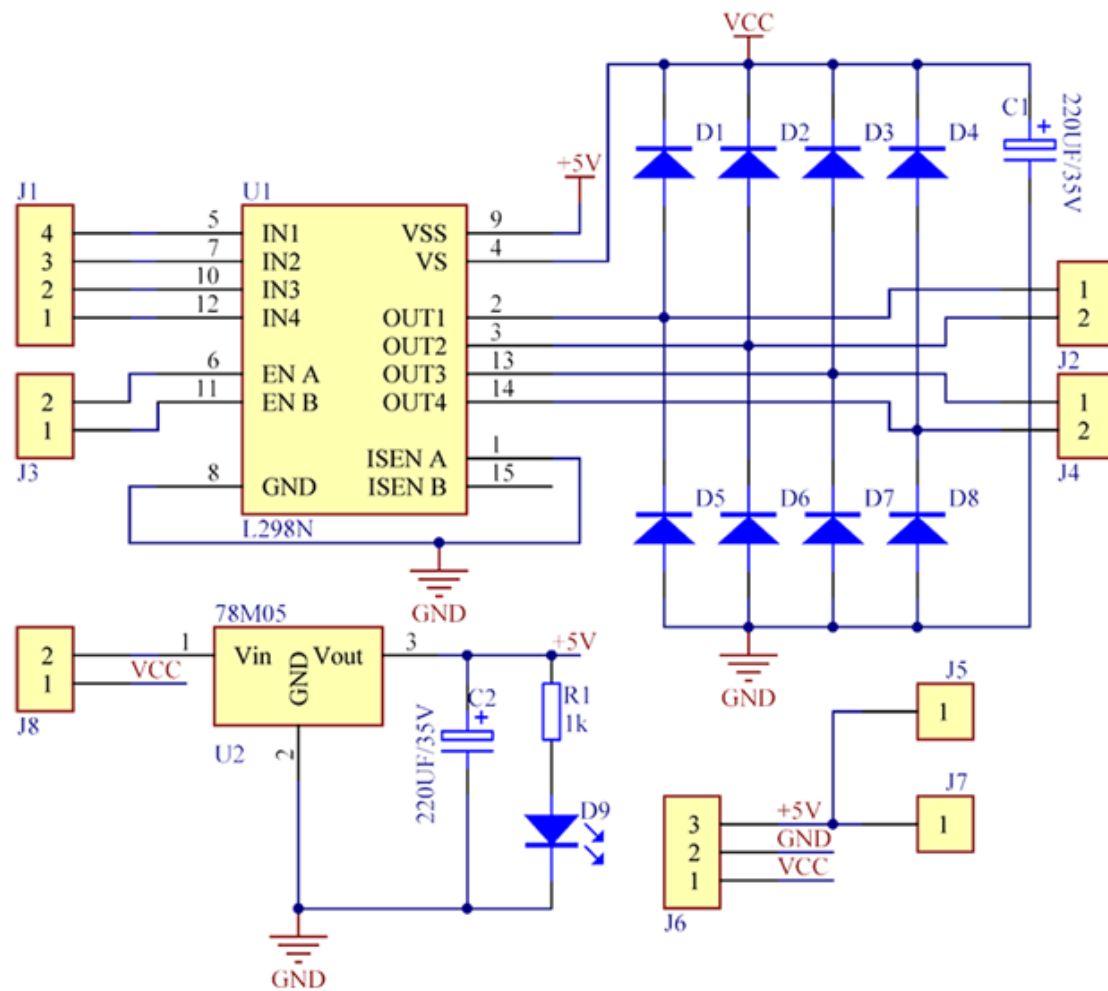
L298N Dual H-Bridge Motor Driver

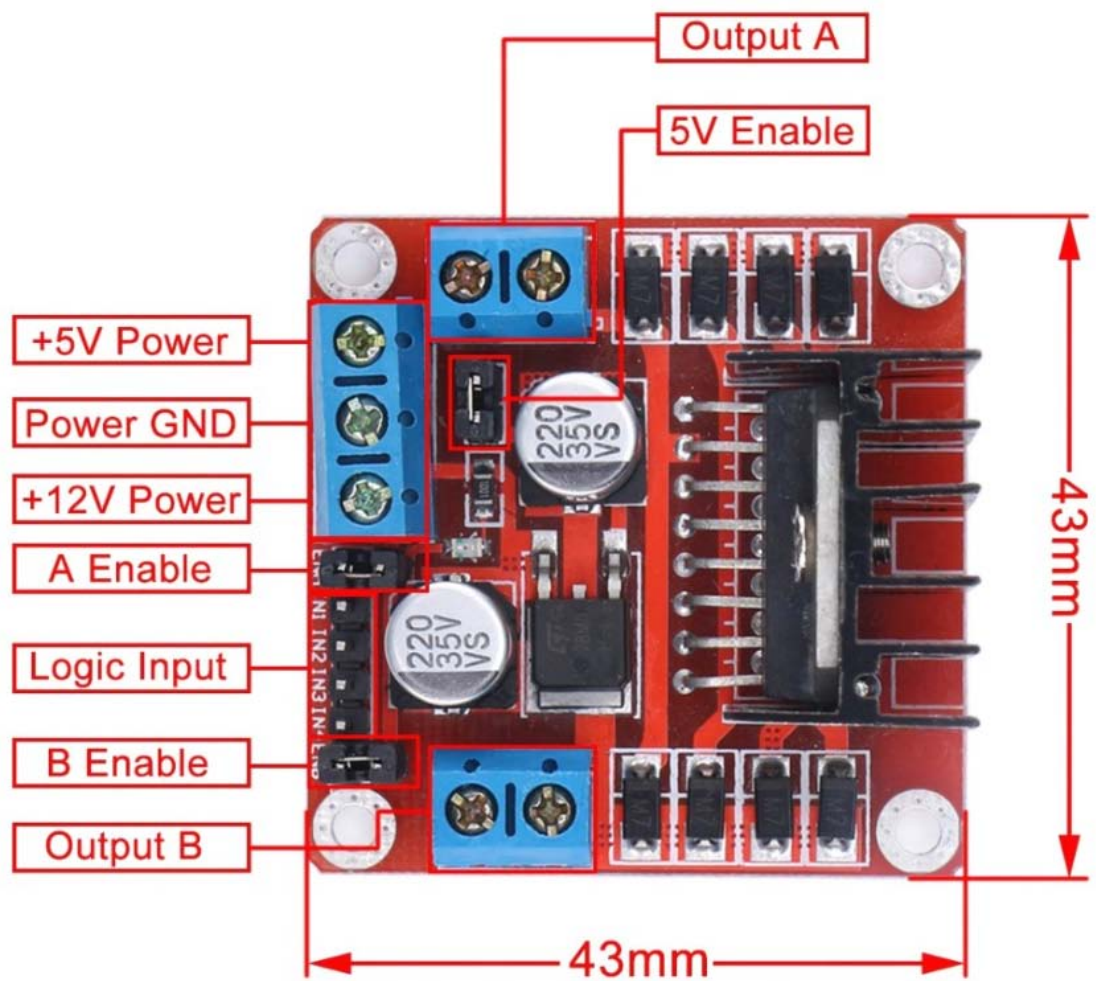
This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.



SKU: MD9031

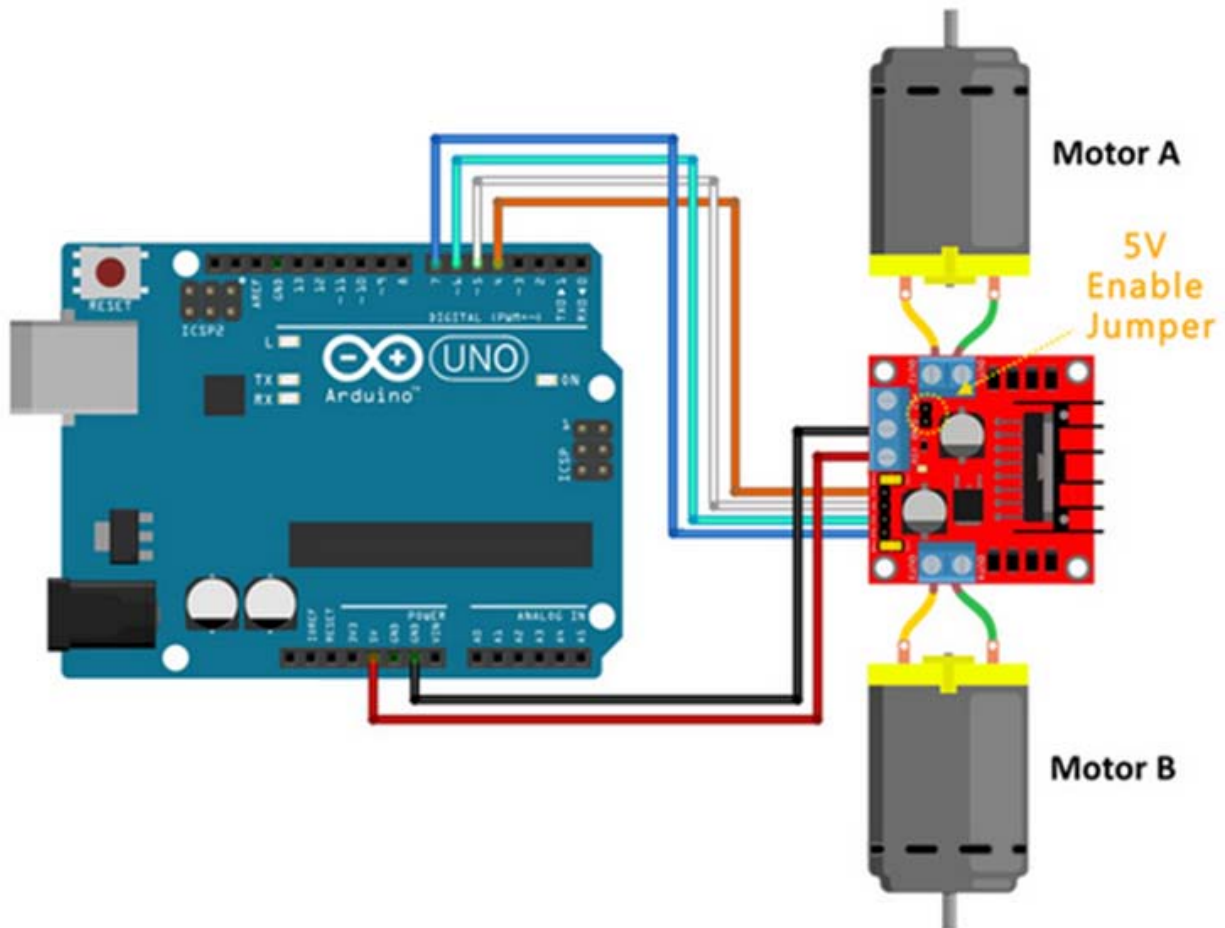
- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
 - Low: $-0.3V \leq V_{in} \leq 1.5V$.
 - High: $2.3V \leq V_{in} \leq V_{ss}$.
- Enable signal input voltage range :
 - o Low: $-0.3 \leq V_{in} \leq 1.5V$ (control signal is invalid).
 - o High: $2.3V \leq V_{in} \leq V_{ss}$ (control signal active).
- Maximum power consumption: 20W (when the temperature $T = 75\text{ }^{\circ}\text{C}$).
- Storage temperature: $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm





Controlling 2-DC Motor with +5V Arduino onboard Power Supply:

Below is the circuit connection use the on-board +5V power supply from Arduino board, and should be done without the 5V Enable Jumper on (Active 5V). This connection can drive two 5V DC motors simultaneously.



Copy and paste the sketch below to Arduino IDE and upload to Arduino Uno/Mega board.

```
/*=====

// Project      : Arduino Uno
// Description  : L298N Motor Driver
// Source-Code  : L298N_Motor.ino
// Program: Control 2 DC motors using L298N H Bridge Driver
//=====
*/

// Definitions Arduino pins connected to input H Bridge
int IN1 = 4;
int IN2 = 5;
int IN3 = 6;
int IN4 = 7;

void setup()
{
  // Set the output pins
  }
```

```
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
}

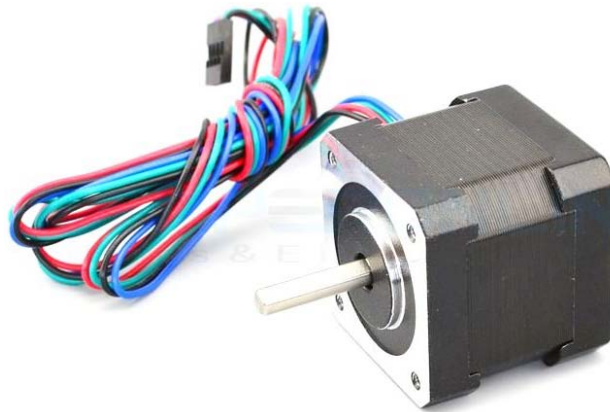
void loop()
{
    // Rotate the Motor A clockwise
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    delay(2000);
    // Motor A
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    delay(500);

    // Rotate the Motor B clockwise
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(2000);
    // Motor B
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
    delay(500);

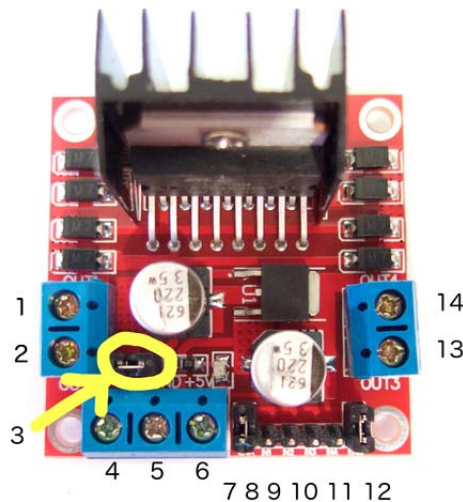
    // Rotates the Motor A counter-clockwise
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    delay(2000);
    // Motor A
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    delay(500);

    // Rotates the Motor B counter-clockwise
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(2000);
    // Motor B
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
    delay(500);
}
```

In this example we have a typical [NEMA-17](#) stepper motor with four wires:



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.



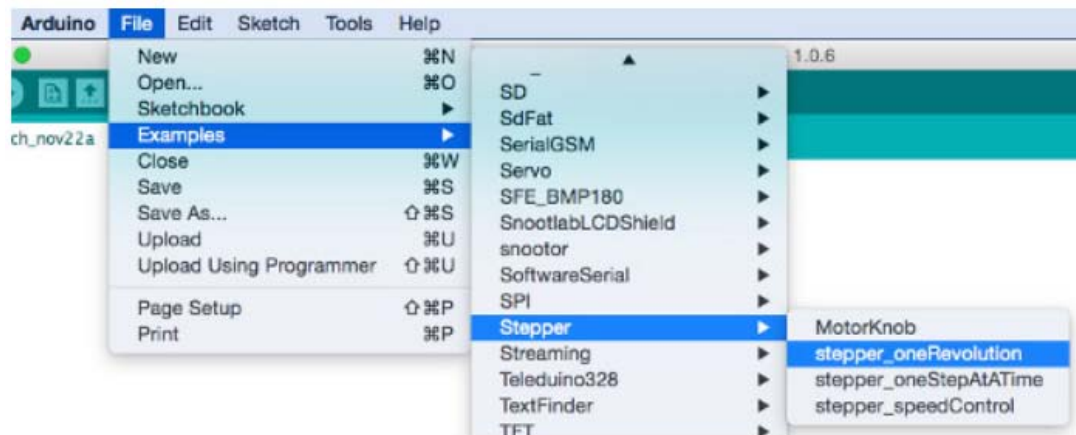
Connect the A+, A-, B+ and B- wires from the stepper motor to the module connections 1, 2, 13 and 14 respectively. Place the jumpers included with the L298N module over the pairs at module points 7 and 12. Then connect the power supply as required to points 4 (positive) and 5 (negative/GND).

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, for example:



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

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

