Arduino Motor Shield (L298)

Manual

This DFRobot L298 DC motor driver shield uses LG' high power H-bridge driver Chip L298P, which is able to drive DC motor, two-phase or four phase stepper motor with a maximum 2A current.

D_Robotics
4/7/2011

Arduino Motor Shield (L298P) (SKU:DRI0009)

I Introduction

This DFRobot L298 DC motor driver shield uses LG' high power H-bridge driver Chip L298P, which is able to drive DC motor, two-phase or four phase stepper motor with a maximum 2A current. The motor output uses eight high-speed Schottky diodes for protection. The circuit wiring of the shield is well organised and the stack design makes it directly mountable onto our Arduino compatible Duemilanove 328 or Mega.

The Shield can switch between PWM speed control mode and PLL phase-locked loop mode through setting the appropriate jumpers. The power supply can be achieved either via Arduino VIN input or PWRIN input on the sield through setting the appropriate jumpers. The speed control is achieved through conventional PWM which can be obtained from Arduino's PWM output Pins 5 and 6. The enable/disable function of the motor control is signaled by Arduino Digital Pins 4 and 7.

The Motor shield can be powered directly from Arduino or from external power source. It is strongly recommended to use external power supply to power the motor shield.

- Logic Control Voltage：5V (From Arduino)
- Motor Driven Voltage：4.8～35V (From Arduino or External Power Source)
- Logic supply current Iss：≤36mA
- Motor Driven current Io：≤2A
- Maximum power consumption：25W（T=75℃）
- PWM、PLL Speed control mode
- Control signal level:

  High：2.3V≤Vin≤5V
  Low：-0.3V≤Vin≤1.5V

II Diagram and Control Mode Setting

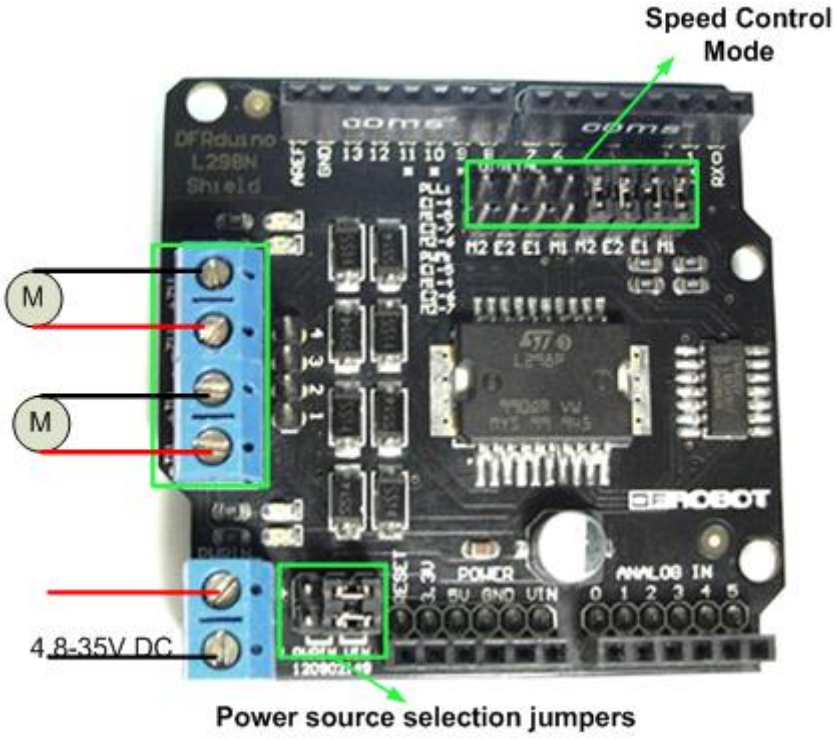

Figure 1 L298 Motor Shield

Control Mode Selection Jumpers (Figure 2 below): The shield supports PWM and PLL (Phased Locked Loop) control Modes. The PWM mode uses E1 and E2 to generate PWM signal. The PLL mode uses M1 and M2 to generate phase control signal.

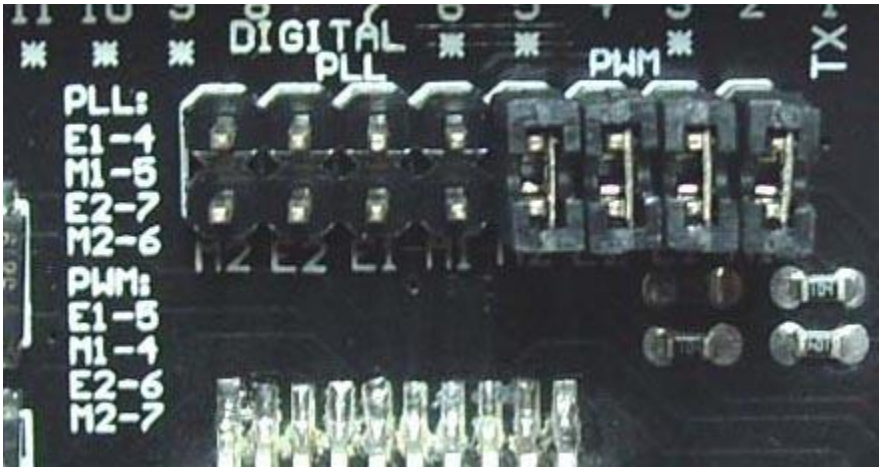

Figure 2 Control Mode Selection Jumpers

Motor Terminal (Figure 3 below): Two DC motors are connected to blue motor terminals. The male header behind the terminals are the same as the motor terminals.

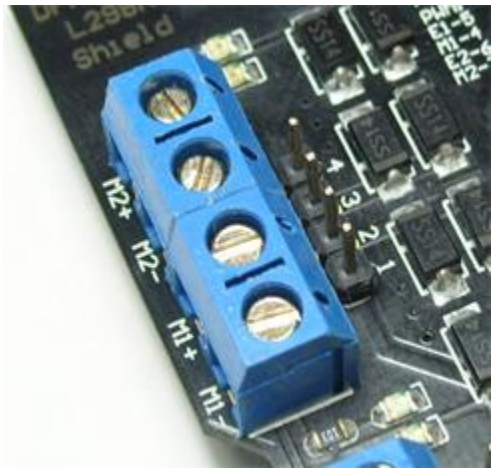

Figure 3 L298 Motor terminal

PWRIN (Figure 4 below)： The motors can be powered by external power supply when the motor current exceeds the limits provided from the Arduino. The switch between external and Arduino power is implemented by two jumpers.

- PWRIN: External Power
- VIN: Arduino Power

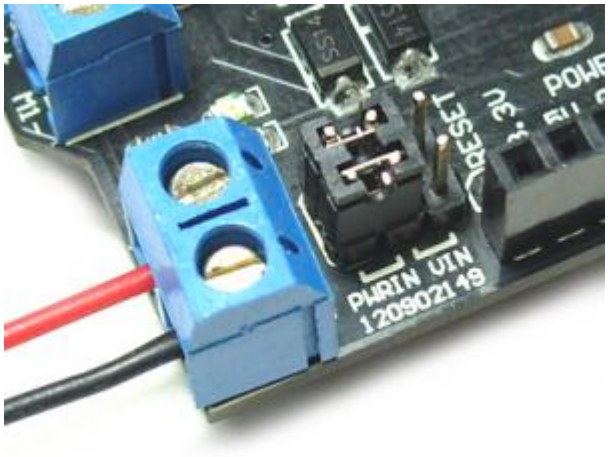

Figure 4 The motors are powered by external power supply

Figure 5 below demonstrates how the motor shield is set to be powered by Arduino power only.

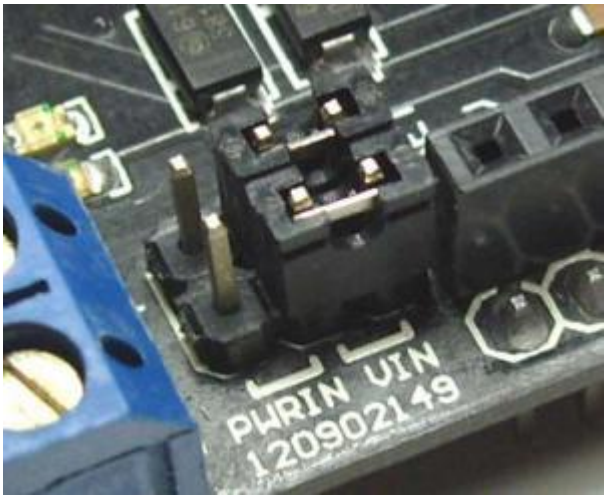

Figure 5 The motors are powered by Arduino power supply

NOTE: When the motor shield is powered by external power source, make sure the external power source and Arduino have the same GND.

Control Signal Truth Table：

E1	M1		E2	M2	
L	X	Motor 1 Disabled	L	X	Motor 2 Disabled
H	H	Motor 1 Backward	H	H	Motor 2 Backward
PWM	X	PWM Speed control	PWM	X	PWM Speed control

Note： H is High level; L is Low level; PWM is Pulse Width Modulation signal; X is any voltage level

III Pin Allocation

"PWM Mode"	
Pin	**Function**
Digital 4	Motor 2 Direction control
Digital 5	Motor 2 PWM control
Digital 6	Motor 1 PWM control
Digital 7	Motor 1 Direction control

"PLL Mode"	
Pin	**Function**
Digital 4	Motor 2 Enable control
Digital 5	Motor 2 Direction control
Digital 6	Motor 1 Direction control
Digital 7	Motor 1 Enable control

IV Sample Code

PWM Speed Control

```
1  //Arduino PWM Speed Control：
2  int E1 = 6;
3  int M1 = 7;
4  int E2 = 5;
5  int M2 = 4;
6
7  void setup()
8  {
9     pinMode(M1, OUTPUT);
10    pinMode(M2, OUTPUT);
11}
12
13void loop()
14{
15  int value;
16  for(value = 0 ; value <= 255; value+=5)
17  {
18    digitalWrite(M1,HIGH);
19    digitalWrite(M2, HIGH);
20    analogWrite(E1, value);   //PWM Speed Control
21    analogWrite(E2, value);   //PWM Speed Control
22    delay(30);
23  }
24}
```

PLL Speed Control

```
1  //Arduino PLL Speed Control：
2  int E1 = 7;
3  int M1 = 6;
4  int E2 = 4;
5  int M2 = 5;
6
7  void setup()
8  {
9     pinMode(M1, OUTPUT);
10    pinMode(M2, OUTPUT);
11}
12
13void loop()
14{
15  int value;
16  for(value = 0 ; value <= 255; value+=5)
17  {
```

```
18   digitalWrite(M1,HIGH);
19   digitalWrite(M2, HIGH);
20   analogWrite(E1, value);   //PLL Speed Control
21   analogWrite(E2, value);   //PLL Speed Control
22   delay(30);
23  }
24}
```