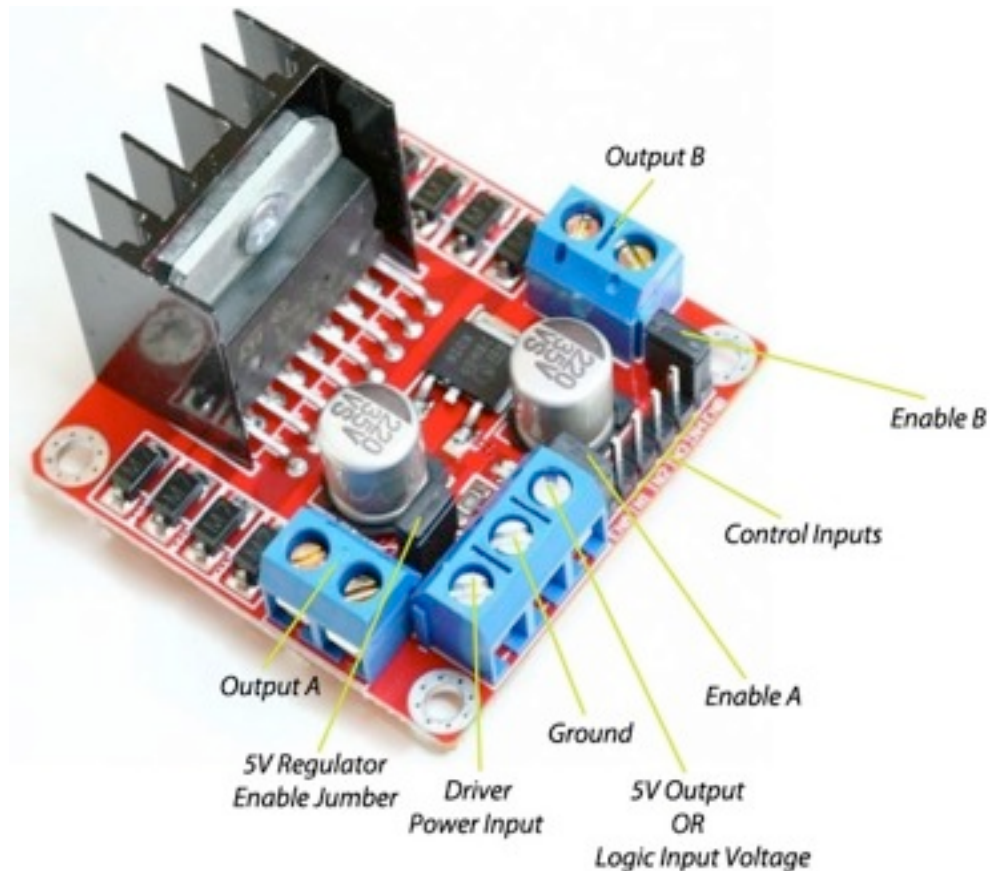


L298 Dual H-Bridge Motor Driver



This driver module is based on L298N H-bridge, a high current, high voltage dual full bridge driver manufactured by ST company. It can drive up to 2 DC motors 2A each. It can also drive one stepper motor or 2 solenoids.

The driver can control both motor RPM and direction of rotation. The RPM is controlled using PWM input to ENA or ENB pins, while of rotation direction is controlled by supplying high and low signal to EN1-EN2 for the first motor or EN3-EN4 for second motor. This Dual H-Bridge driver is capable of driving voltages up to 46V.

Features

- Dual H bridge drive (can drive 2 DC motors)
- Chip L298N
- Logical voltage 5V
- Drive voltage 5V-35V
- Logic current 0mA-36mA
- Drive current 2A(For each DC motor))
- Weight 30g
- Size: 43*43*27mm

Connecting with arduino or microcontroller

So that's that, next is how we hook it up to the Arduino or other micro-controllers. There are 8 pins:

1- GND

2- + 5 V (power for driver (not motor))

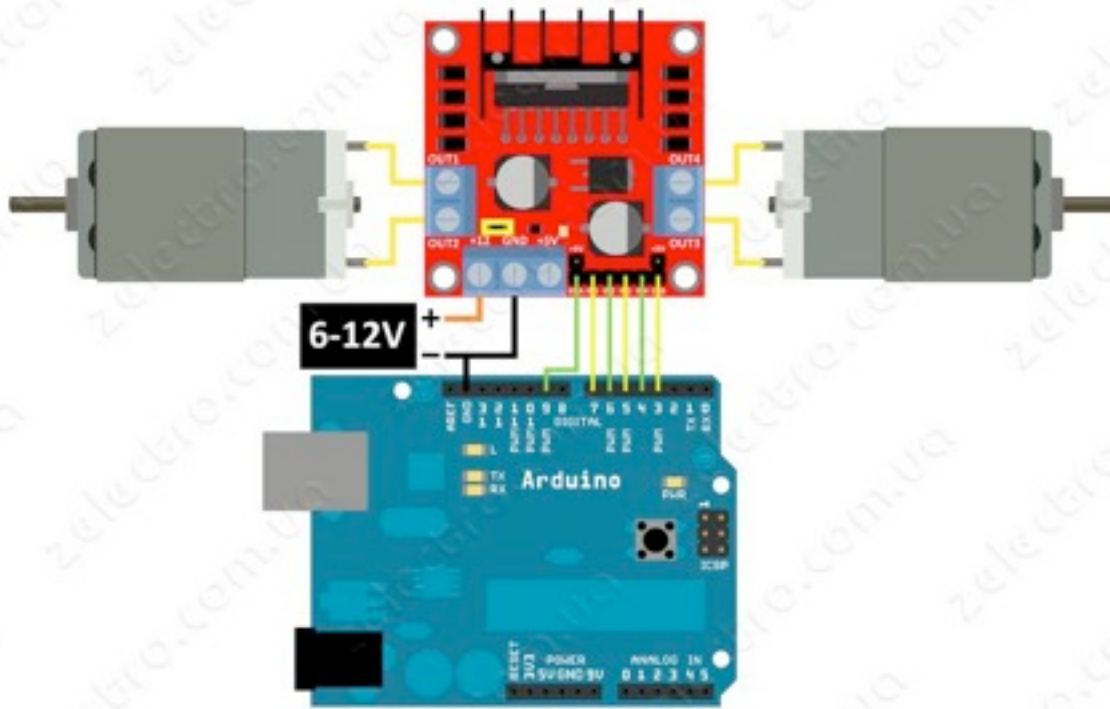
3- ENA: Motor enable for Motor A (high/low)

4, 5- IN1, IN2: These pins define Motor A direction of rotation (one is high and the other is low)

6-ENB: Motor enable for Motor B (high/low)

7,8- IN3, IN4: These pins define Motor B direction of rotation (one is high and the other is low)

For Motor Brake, both IN1 and IN2 or IN3 and IN4 are set high.



Arduino Code:

```
// Yu Hin Hau 001
// Robotic Car via H-Bridge (L298) 002
// June 5, 2012 003
004
//See Low Level for Command Definitions 005
006
//Define Pins 007
int enableA = 2; 008
int pinA1 = 1; 009
```

```
int pinA2 = 0; 010
011
012
int enableB = 7; 013
int pinB1 = 6; 014
int pinB2 = 5; 015
016
//Define Run variable 017
boolean run; 018
void setup() { 019
020
    pinMode(enableA, OUTPUT); 021
    pinMode(pinA1, OUTPUT); 022
    pinMode(pinA2, OUTPUT); 023
024
    pinMode(enableB, OUTPUT); 025
    pinMode(pinB1, OUTPUT); 026
    pinMode(pinB2, OUTPUT); 027
028
    run = true; 029
030
} 031
032
//command sequence
```

```
void loop() { 033
  034
  if(run) 035
  { 036
    037
    delay(2000); 038
    039
    enableMotors(); 040
    041
    forward(1000); 042
    coast(500); 043
    044
    backward(1500); 045
    coast(500); 046
    047
    forward(500); 048
    brake(500); 049
    050
    turnLeft(500); 051
    turnRight(500); 052
    053
    disableMotors(); 054
    055
```

```
run = false; 056
} 057
} 058
} 059
} 060
//Define Low Level H-Bridge Commands 061
//enable motors 062
//enable motors 063
void motorAOn() 064
{ 065
digitalWrite(enableA, HIGH); 066
} 067
} 068
void motorBOn() 069
{ 070
digitalWrite(enableB, HIGH); 071
} 072
} 073
//disable motors 074
void motorAOff() 075
{ 076
digitalWrite(enableB, LOW); 077
} 078
```

```
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
101

void motorBOff()
{
    digitalWrite(enableA, LOW);
}

//motor A controls
void motorAForward()
{
    digitalWrite(pinA1, HIGH);
    digitalWrite(pinA2, LOW);
}

void motorABackward()
{
    digitalWrite(pinA1, LOW);
    digitalWrite(pinA2, HIGH);
}

//motor B controls
void motorBForward()
{
    digitalWrite(pinB1, HIGH);
```

```
digitalWrite(pinB2, LOW); 102
} 103
104
105
void motorBBackward() 106
{ 107
digitalWrite(pinB1, LOW); 108
digitalWrite(pinB2, HIGH); 109
} 110
111
//coasting and braking 112
void motorACoast() 113
{ 114
digitalWrite(pinA1, LOW); 115
digitalWrite(pinA2, LOW); 116
} 117
118
void motorABrake() 119
{ 120
digitalWrite(pinA1, HIGH); 121
digitalWrite(pinA2, HIGH); 122
} 123
124
void motorBCoast()
```



```
125
{
126
    digitalWrite(pinB1, LOW);
127
    digitalWrite(pinB2, LOW);
128
}
129
130
void motorBBrake()
131
{
132
    digitalWrite(pinB1, HIGH);
133
    digitalWrite(pinB2, HIGH);
134
}
135
136
//Define High Level Commands
137
138
void enableMotors()
139
{
140
    motorAOn();
141
    motorBOn();
142
}
143
144
void disableMotors()
145
{
146
    motorAOff();
147
    motorBOff();
```

```

}
148
149
150
void forward(int time)
151
{
152
motorAForward();
153
motorBForward();
154
delay(time);
155
}
156
157
void backward(int time)
158
{
159
motorABackward();
160
motorBBackward();
161
delay(time);
162
}
163
164
void turnLeft(int time)
165
{
166
motorABackward();
167
motorBForward();
168
delay(time);
169
}
170
```

```
void turnRight(int time) 171
{ 172
    motorAForward(); 173
    motorBBackward(); 174
    delay(time); 175
} 176
177
void coast(int time) 178
{ 179
    motorACoast(); 180
    motorBCoast(); 181
    delay(time); 182
} 183
184
void brake(int time) 185
{ 186
    motorABrake(); 187
    motorBBrake(); 188
    delay(time); 189
} 190
```