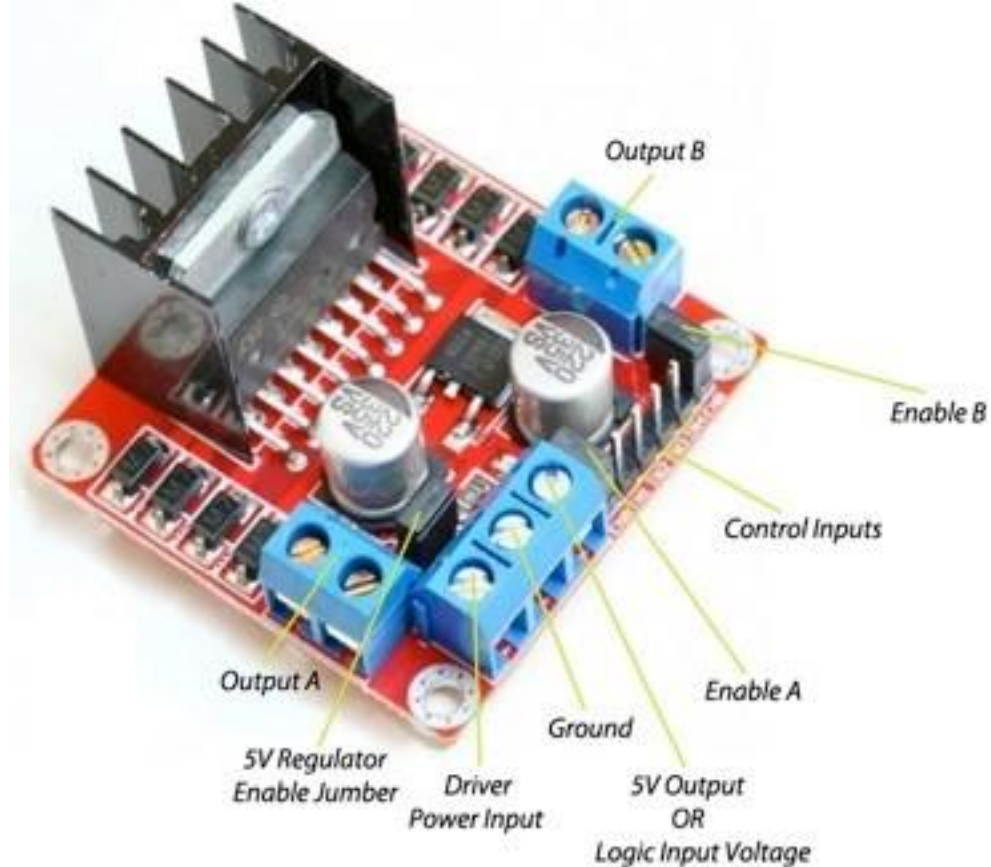


L298 Dual H-Bridge Motor Driver



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 microcontrollers. 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
// Robotic Car via H-Bridge (L298)
// June 5, 2012

//See Low Level for Command Definitions

//Define Pins

int enableA = 2;

009 int pinA1 = 1;
010
```

```

int pinA2 = 0;
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
033
void loop() {

```

	034
	035
<code>if(run)</code>	036
<code>{</code>	037
	038
<code>delay(2000);</code>	039
	040
<code>enableMotors();</code>	041
	042
<code>forward(1000);</code>	043
<code>coast(500);</code>	044
	045
<code>backward(1500);</code>	046
<code>coast(500);</code>	047
	048
<code>forward(500);</code>	049
<code>brake(500);</code>	050
	051
<code>turnLeft(500);</code>	052
<code>turnRight(500);</code>	053
	054
<code>disableMotors();</code>	055
	056
<code>run = false;</code>	057

```
    }
058
059
    }
060
061
    //Define Low Level H-Bridge Commands
062
063
    //enable motors
064
    void motorAOn()
065
    {
066
        digitalWrite(enableA, HIGH);
067
    }
068
069
    void motorBOn()
070
    {
071
        digitalWrite(enableB, HIGH);
072
    }
073
074
    //disable motors
075
    void motorAOff()
076
    {
077
        digitalWrite(enableB, LOW);
078
    }
079
080
```

```

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()
{
    101 digitalWrite(pinB1, HIGH);
    digitalWrite(pinB2, LOW);
}

```

```

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);
}

void motorBBrake()
{
    digitalWrite(pinB1, HIGH);
    digitalWrite(pinB2, HIGH);
}

//Define High Level Commands

void enableMotors()
{
    motorAOn();
    motorBOn();
}

void disableMotors()
{
    motorAOff();
    147 motorBOff();
}

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
171
void turnRight(int time)
172
{
173
    motorAForward();
174
```

```
motorBBackward();
delay(time);
}

void coast(int time)
{
    motorACoast();
    motorBCoast();
    delay(time);
}

void brake(int time)
{
    motorABrake();
    motorBBrake();
    delay(time);
}
```