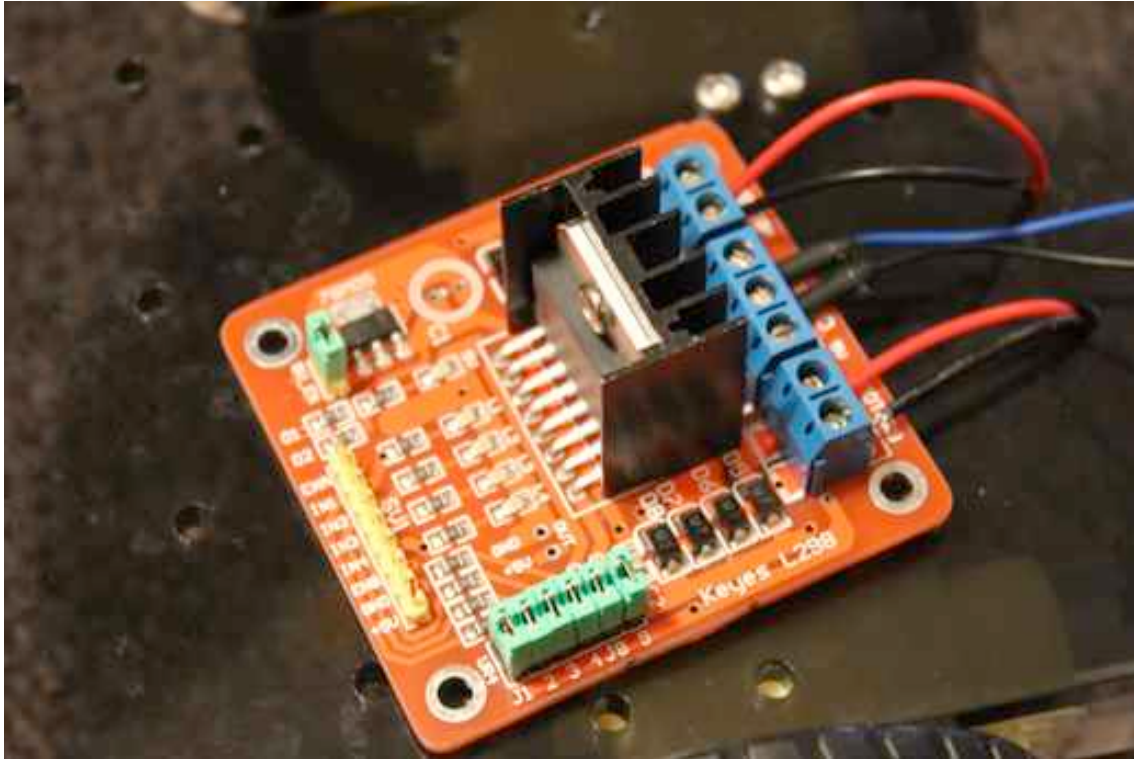


## L298 Dual H-Bridge Motor Driver



### **Motor Connection and external power source**

we got 2 Motor connectors (the 2 blue exterior ones on the right side). On the board they are labeled MotorA and MotorB.

In the middle, there is a set of 3 terminals: Vms, GND, 5V. Now I think the 5V is just there to provide a 5V power source if you need it for something like an ultrasonic sensor. Vms, that is your motor's external power source. Due to the limited current provided by the Arduino or microcontroller power source , the motors are super weak. So for real testing, just put outer power source.

## 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
//See Low Level for Command Definitions 004
005
006
007
```

```
//Define Pins
int enableA = 2;
int pinA1 = 1;
int pinA2 = 0;

int enableB = 7;
int pinB1 = 6;
int pinB2 = 5;

//Define Run variable
boolean run;

void setup() {

    pinMode(enableA, OUTPUT);
    pinMode(pinA1, OUTPUT);
    pinMode(pinA2, OUTPUT);

    pinMode(enableB, OUTPUT);
    pinMode(pinB1, OUTPUT);
    pinMode(pinB2, OUTPUT);

    run = true;

```

```
} 031

//command sequence 032

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

    052

    turnRight(500); 053
```

```
disableMotors(); 054
055
run = false; 056
} 057
} 058
} 059
060
//Define Low Level H-Bridge Commands 061
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);
}

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);
}

void motorBBackward()
{
    digitalWrite(pinB1, LOW);
    digitalWrite(pinB2, HIGH);
}

//coasting and braking

void motorACoast()
{
    digitalWrite(pinA1, LOW);
    digitalWrite(pinA2, LOW);
}

void motorABrake()
{
    digitalWrite(pinA1, HIGH);
    digitalWrite(pinA2, HIGH);
}

```

```
    }
    123

    void motorBCoast()
    124
    {
    125
    digitalWrite(pinB1, LOW);
    126
    digitalWrite(pinB2, LOW);
    127
    }
    128
    129

    void motorBBrake()
    130
    {
    131
    digitalWrite(pinB1, HIGH);
    132
    digitalWrite(pinB2, HIGH);
    133
    }
    134
    135

    //Define High Level Commands
    136
    137

    void enableMotors()
    138
    {
    139
    motorAOn();
    140
    motorBOn();
    141
    }
    142
    143

    void disableMotors()
    144
    145
```



```
{
motorAOff();
motorBOff();
}

void forward(int time)
{
motorAForward();
motorBForward();
delay(time);
}

void backward(int time)
{
motorABackward();
motorBBackward();
delay(time);
}

void turnLeft(int time)
{
motorABackward();
motorBForward();
}
```

```
    delay(time);
}

void turnRight(int time)
{
    motorAForward();
    motorBBackward();
    delay(time);
}

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

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

