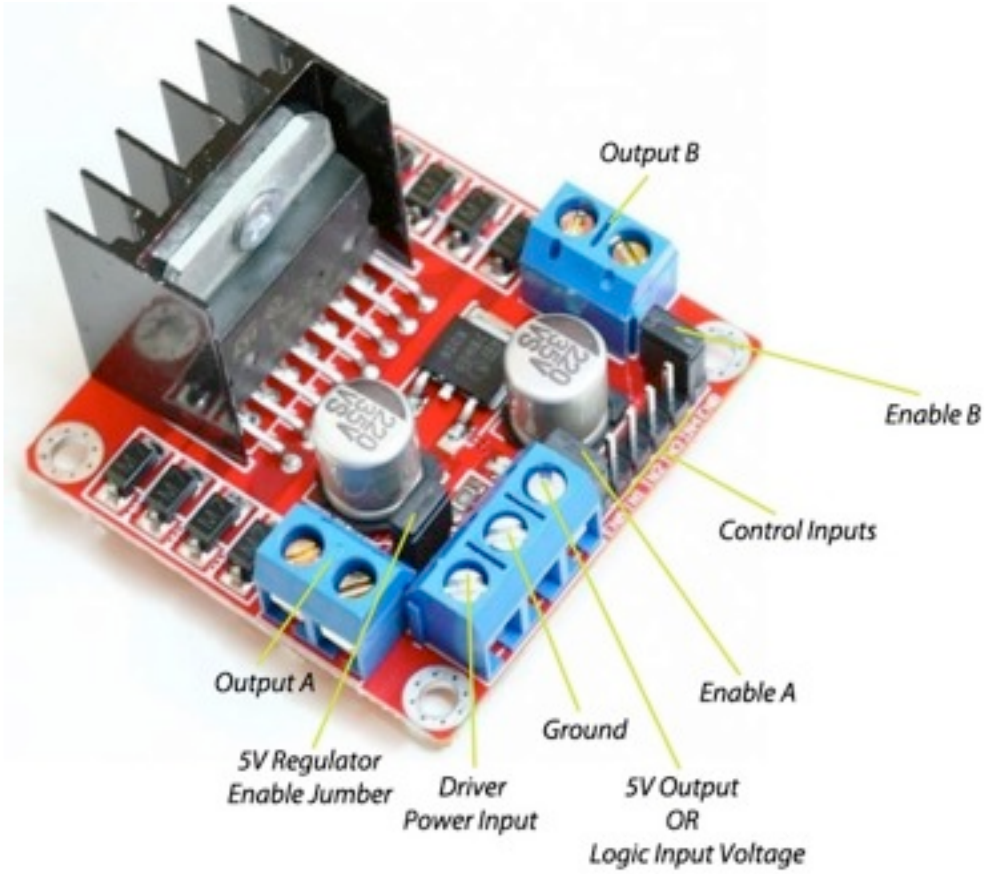
Future electronics

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 suppling 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.

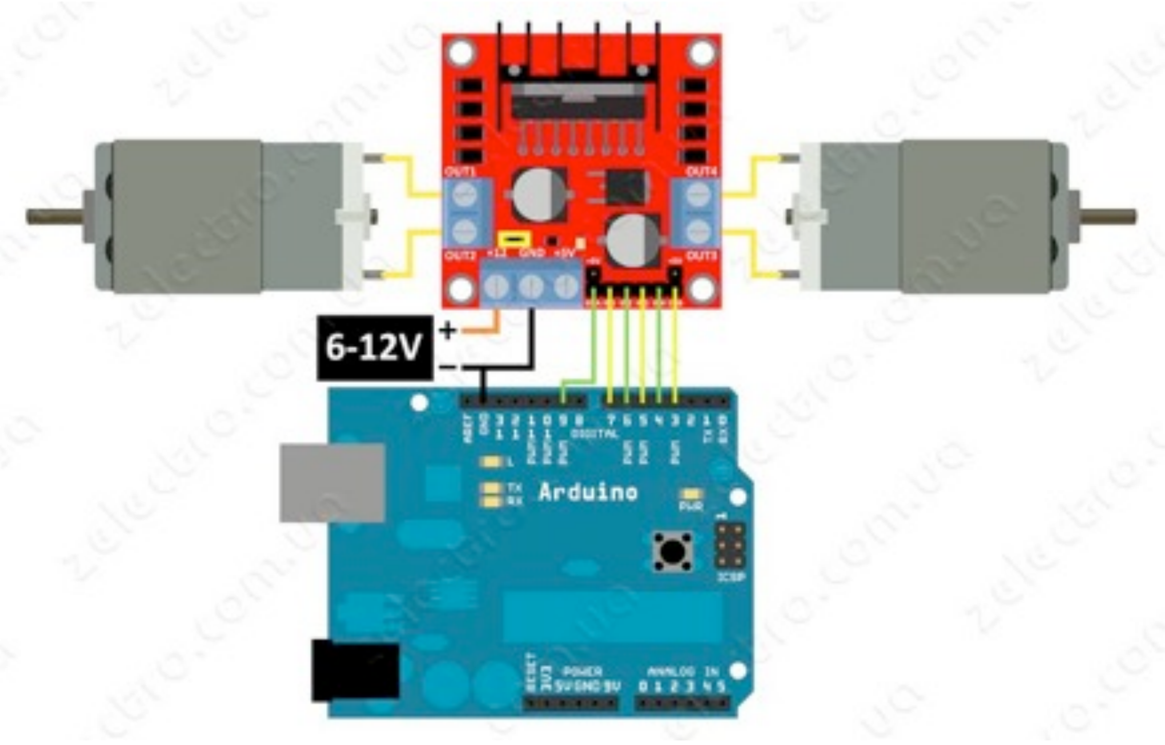
Features

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

//See Low Level for Command Definitions

//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;

}

//command sequence
```

```
void loop() {

 if(run)
 {

 delay(2000);

 enableMotors();

 forward(1000);
 coast(500);

 backward(1500);
 coast(500);

 forward(500);
 brake(500);

 turnLeft(500);
 turnRight(500);

 disableMotors();

 run = false;
 }

}

//Define Low Level H-Bridge Commands

//enable motors
void motorAOn()
{
 digitalWrite(enableA, HIGH);
}

void motorBOn()
{
 digitalWrite(enableB, HIGH);
}

 //disable motors
void motorAOff()
{
 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 contorls
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);
}

void motorBCoast()
```

```
{
 digitalWrite(pinB1, LOW);
 digitalWrite(pinB2, LOW);
}

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

//Define High Level Commands

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

void disableMotors()
{
 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);
}
```