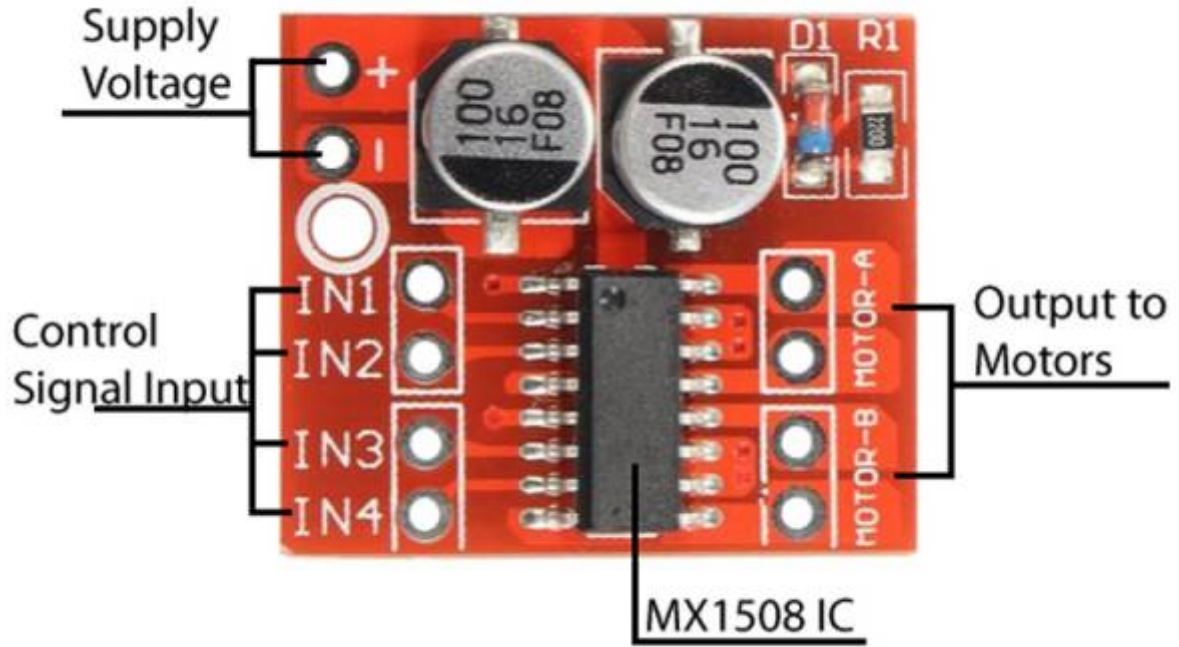
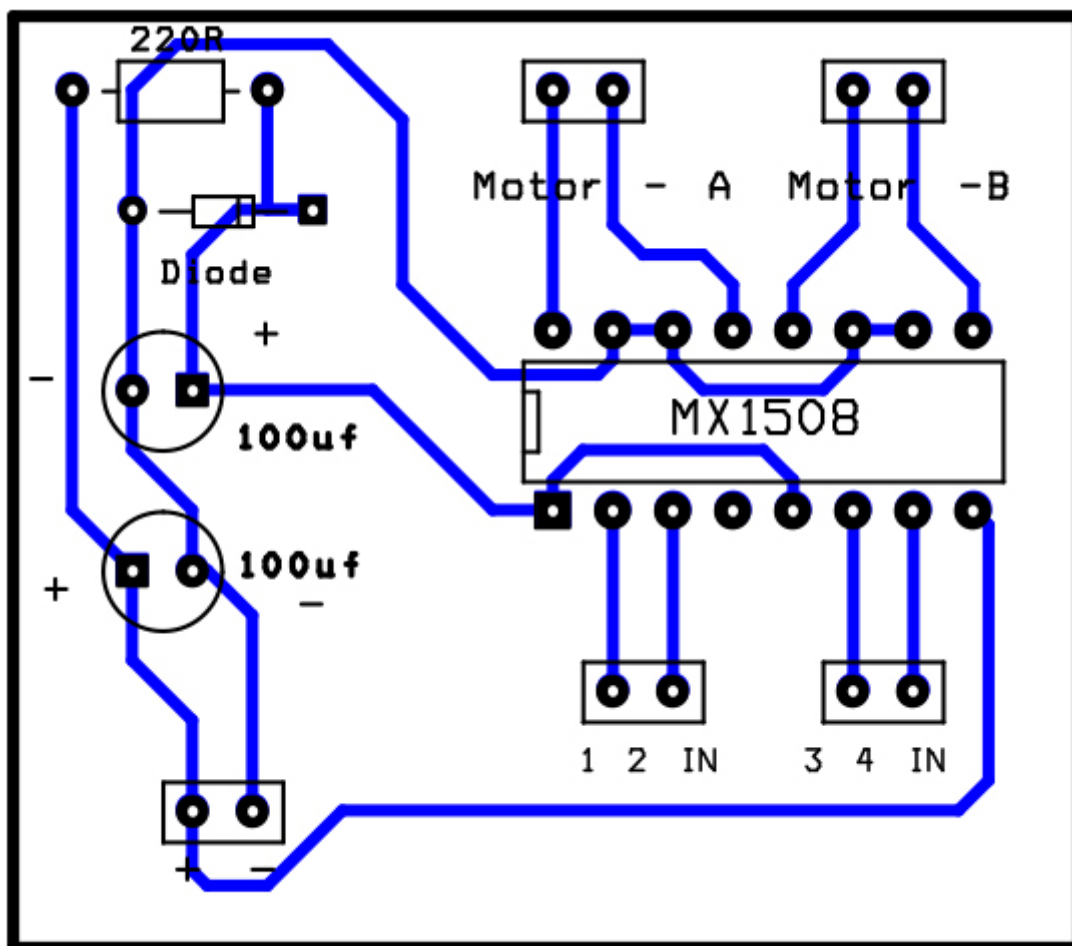


MX1508





Caractéristiques du module pilote de moteur MX1508

Le pilote MX1508 est l'un des pilotes les moins chers pour faire fonctionner des moteurs à courant continu avec un courant maximum de 2 ampères. Les petites dimensions et la configuration simple sont les caractéristiques positives de ce module.

Le MX1508 est un pilote de moteur à double pont complet qui permet de connecter simultanément deux moteurs à courant continu.

Ce pilote dispose d'un circuit de protection contre la température qui désactive le pilote en cas de surchauffe. Les fonctionnalités de ce module sont les suivantes :

- Plage de tension de fonctionnement : 2 à 9,6 V CC.
- Courant de sortie maximum : 1,5 A (crête 2 A)
- Température maximale de fonctionnement : 15 °C



Le contrôleur de moteur MX1508 suit la configuration du pont en H, ce qui est pratique pour piloter des moteurs pas à pas.

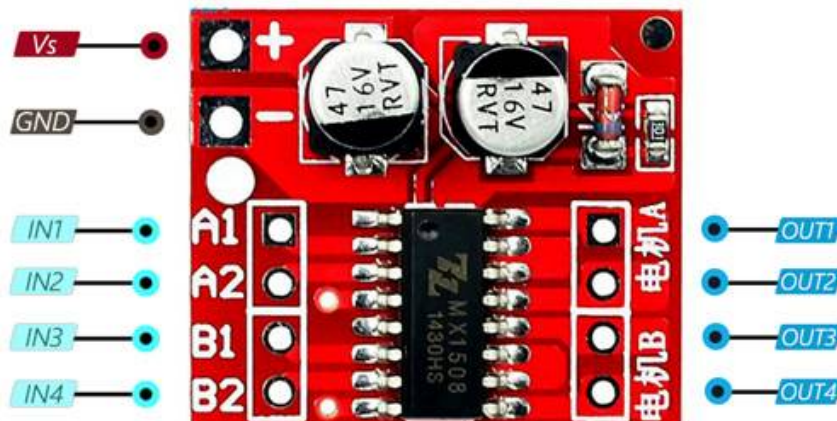
Fiche technique

[MX1508 : fiche technique EN](#)

Brochage du module pilote de moteur MX1508

Ce capteur possède 10 broches :

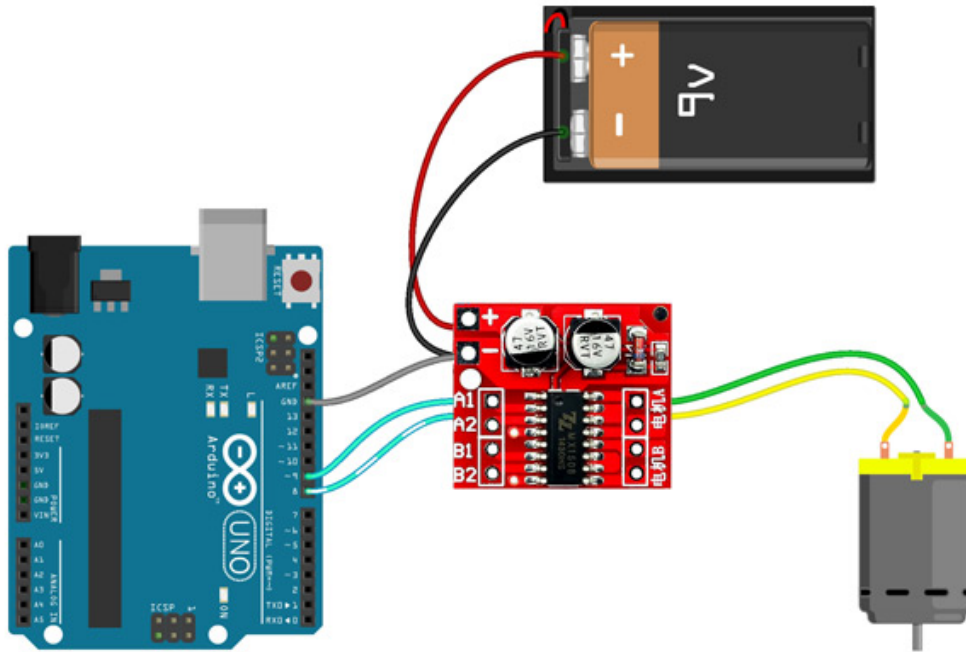
- Vs : Alimentation du module (entrée)
- GND : terre
- IN1 : Entrée 1
- IN2 : Entrée 2
- IN3 : Entrée 3
- IN4 : Entrée 4
- SORTIE1 : Sortie 1
- SORTIE2 : Sortie 2
- OUT3 : Sortie 3
- OUT4 : Sortie 4



www.Electropeak.com



Exemple d 'utilisation



Code arduino Exemple

[exemple_mx1508.ino](#)

```
/*
  MX1508 DC MOTOR DRIVER MODULE
  modified on 25 Sep 2020
  by Saeed Olfat @ Electropeak
  Home
*/

void setup() {
  pinMode(8, OUTPUT); //IN2
  pinMode(9, OUTPUT); //IN1
}

void loop() {

  // Full speed forward
  digitalWrite(8, HIGH);
```

```
digitalWrite(9, LOW);

delay(3000);

// Full speed backward
digitalWrite(8, LOW);
digitalWrite(9, HIGH);

delay(3000);

// 0 to 100% speed in forward mode
for (int i=0;i<256;i++)
{ digitalWrite(8, HIGH);
  analogWrite(9, i);
  delay(20);
}

delay(50);

// 0 to 100% speed in backward mode
for (int i=255;i<0;i--)
{ digitalWrite(8, LOW);
  analogWrite(9, i);
  delay(20);
}

delay(50);
}
```

Comme vous pouvez le voir dans le code, le moteur avance d'abord pendant 3 secondes et recule pendant 3 secondes. Ensuite le moteur s'arrête et sa vitesse augmente de 0 à 100% en marche avant avec une accélération constante, puis le même mouvement s'effectue en marche arrière.

Utilisation avec une bibliothèque

[Librairie MX1508](#)

[Video utilisation Librairie MX1508](#)

==== Code utilisation librairie MX1508]]

[LibMX1508.ino](#)

```
#include <MX1508.h>

#define PINA 9
#define PINB 10
#define NUMPWM 2
```

```
// MX1508 schematics(in Chinese) can be found here at:  
http://sales.dzsc.com/486222.html  
/*  
 * MX1508(uint8_t pinIN1, uint8_t pinIN2, DecayMode decayMode,  
 NumOfPwmPins numPWM);  
 * DecayMode must be FAST_DECAY or SLOW_DECAY,  
 * NumOfPwmPins, either use 1 or 2 pwm.  
 * I recommend using 2 pwm pins per motor so spinning motor forward and  
 backward gives similar response.  
 * if using 1 pwm pin, make sure its pinIN1, then set pinIN2 to any  
 digital pin. I dont recommend this setting because  
 * we need to use FAST_DECAY in one direction and SLOW_DECAY for the  
 other direction.  
 */  
MX1508 motorA(PINA,PINB, FAST_DECAY, NUMPWM);  
  
void setup() {  
  Serial.begin(115200);  
}  
/*  
 * Ramp up to pwm = 100, by increasing pwm by 1 every 50 millisecond.  
 * then ramp down to pwm = -100, by decreasing pwm every 50  
 millisecond.  
 * positive value pwm results in forward direction.  
 * negative value pwm results in opposite direction.  
 */  
void loop() {  
  // put your main code here, to run repeatedly:  
  static unsigned long lastMilli = 0;  
  static bool cwDirection = true; // assume initial direction(positive  
 pwm) is clockwise  
  static int pwm = 1;  
  
  if(millis()-lastMilli > 50){ // every 50 millisecond  
    if (cwDirection && pwm++ > 100 ) {  
      cwDirection = false;  
    } else if (!cwDirection && pwm-- < -100) {  
      cwDirection = true;  
    }  
    motorA.motorGo(pwm);  
    lastMilli = millis();  
    Serial.println(motorA.getPWM()); // we can just print pwm but just  
 showing that member function getPWM() works.  
  }  
}
```

From:

<http://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<http://chanterie37.fr/fablab37110/doku.php?id=start:arduino:mx1508&rev=1700583578>

Last update: **2023/11/21 17:19**

