

ACTIVITÉ PRATIQUE

Commande en vitesse deux sens de rotation d'un moteur à courant continu - Driver MR143 -

1 Objectif

Commander en vitesse un ou deux moteurs à courant continu avec le driver moteur MR143 et selon le temps, **réaliser** le pilotage du driver par une carte à micro-contrôleur Arduino.

2 Caractéristiques du driver MR143

Le MR143 est un driver ¹ permettant de contrôler deux moteurs CC jusqu'à 2 A à partir d'une sortie PWM d'un micro contrôleur (Arduino ou compatible).

Ce module est basé sur deux A4953 annexe B permettant le contrôle de la vitesse et du sens de rotation.

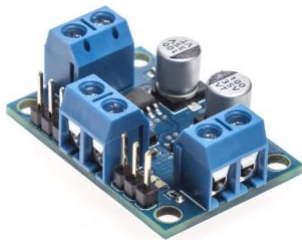


Figure 1: Driver Moteur MR143

- Alimentation: 7 à 30 Vcc
- Sorties: 2 x 2 A
- Entrées/sorties: TTL
- Signal de commande: voir fiche technique
- Leds d'indication
- T° de service: -40 à 85 °C
- Dimensions: 36 x 24 x 13 mm

¹ autrement dit un variateur de vitesse ou plus exactement un variateur de tension dans le cas présent !

3 Travail demandé

3.1 Décodage de la documentation constructeur du MR143

À partir de la description précédente et de la documentation technique du driver moteur MR143 annexe A, répondre aux questions suivantes :

- **Indiquer** la tension maximale d'alimentation ainsi que l'intensité moteur maximale autorisée.

Tension et intensité maximales

- **Reporter** succinctement le tableau *Tab2-Conditions* dans le cadre ci-après en le transcrivant en français

Transcription *Tab2-Conditions*

- **Expliquer** oralement le comportement attendu du moteur à partir du tableau *Tab2-Conditions*.
- **Représenter** succinctement l'allure d'un signal PWM pour un rapport cyclique $\alpha = 0.25$, $\alpha = 0.5$ et $\alpha = 0.75$ en faisant apparaître la période T et le T_{ON} , temps au niveau haut.

Signal PWM pour $\alpha : 0.25$ / $\alpha : 0.5$ / $\alpha : 0.75$

- **Indiquer** sur quelle broche du MR143 doit être imposé le signal PWM pour faire une variation de vitesse en marche avant.

Broche PWM MR143 pour marche avant

3.2 Élaboration du schéma électrique et essais du MR143 + GBF

À partir de la documentation technique du driver moteur MR143 annexe [A](#), **réaliser** un schéma de câblage permettant de valider le fonctionnement du driver associé à un GBF ² pour un seul moteur ³.

Schéma câblage validation fonctionnement MR143 + GBF

3.3 Câblage et essais Driver + GBF

Réaliser le câblage conformément au schéma réalisé précédemment et **procéder** aux essais ⁴ pour des fréquences PWM différentes et des rapports cycliques différents..

Visualiser sur l'oscilloscope (sur la deuxième voie disponible ⁵) l'allure de la tension de sortie du driver moteur MR143 et **indiquer** comment se faire la variation de vitesse.

²Générateur de signal (GBF) qui permettra de commander le driver moteur MR143 en PWM

³Le driver moteur MR143 peut commander deux moteurs CC mais dans le cadre de l'activité, nous ne testerons que sur un seul moteur

⁴**Bien veiller** à régler le GBF à 5 V max. Utiliser un oscilloscope pour visualiser le signal PWM émit par le GBF

⁵On rappelle que la première voie visualise la commande PWM générée par le GBF

3.4 Association driver moteur MR143 et une carte Arduino Méga

3.4.1 Décodage du *code exemple* fourni page C

Ouvrir l'IDE Arduino et y coller le code fourni à [ici](#) ⁶.

Sur l'annexe C, **commenter** chacune des lignes de code en expliquant succinctement chacune des instructions.

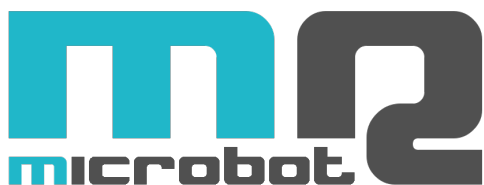
Réaliser le schéma de câblage permettant à la carte Arduino de piloter le driver moteur MR143.

Schéma câblage fonctionnement MR143 + Arduino Méga

Réaliser le câblage et **procéder** aux essais.

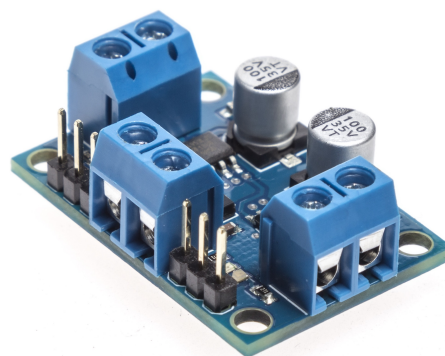
⁶Il s'agit du même code que celui fourni annexe C

A Documentation technique MR143



Name: **Dual DC Motor Driver 4A V3**

Code: **MR001-004.3**



The *Dual DC Motor Driver 4A V3* allows to independently drive two DC motors, controlling both velocity and direction.

The minimum supply voltage allowed is 7V, so you can use also two-cell LiPo batteries (7.4V) that grant small dimensions and low weight characteristics. The maximum supply voltage supported by this board is 30V.

This board also provides direction LED indicators for both channels; this is very useful during setup stage to verify the firmware behaviour (also without applying a real motor to the output).

INSTRUCTIONS

Two via terminal blocks are the two outputs for motors (M1 and M2). The 3 pins strip connectors next to the power terminal block are used to control the 2 channels of this *Dual DC Motor Driver 4A V3*. Each of them has signals as reported on table 1.

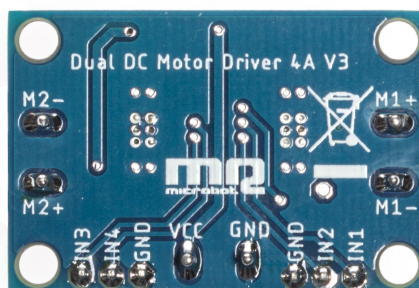
Name	Description
IN1	Input A of channel 1 (TTL input)
IN2	Input B of channel 1 (TTL input)
IN3	Input A of channel 2 (TTL input)
IN4	Input B of channel 2 (TTL input)
GND	ground

Tab.1 – Connections

To understand the meaning of these signals and their use you can read the following table (Tab.2), where all conditions are reported. Note that there are reported only conditions for one channel because conditions for both channels are just the same.

Inputs		M+ e M- output
A	B	
1	1	Motor stopped
1	0	Forward
0	1	Backward
0	0	Motor is in free running

Tab.2 – Conditions



SPECIFICATIONS

Name	Description
Supply voltage	7 - 30V
Supply current (logic)	10mA typ.
Output current	4A (2A per canale)
Data I/O voltage	TTL standard
Dimensions	36 x 24 mm
Weight	9,1g
Operating temperature	-40°C to +85°C

Tab.3 – Specifications

B Extrait datasheet A4953



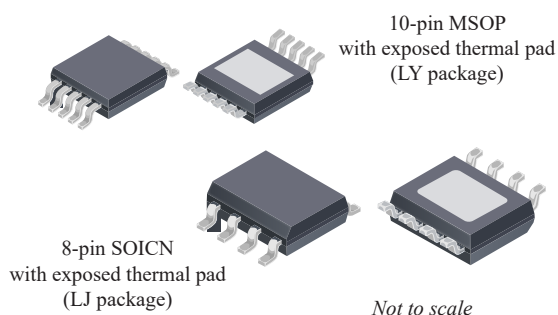
A4952 and A4953

Full-Bridge DMOS PWM Motor Drivers

FEATURES AND BENEFITS

- Low $R_{DS(on)}$ outputs
- Overcurrent protection (OCP)
 - Motor short protection
 - Motor lead short to ground protection
 - Motor lead short to battery protection
- Low Power Standby mode
- Adjustable PWM current limit
- Synchronous rectification
- Internal undervoltage lockout (UVLO)
- Crossover-current protection
- Fault output (A4952 only)
- Selectable retry (A4952 only)

PACKAGES:



DESCRIPTION

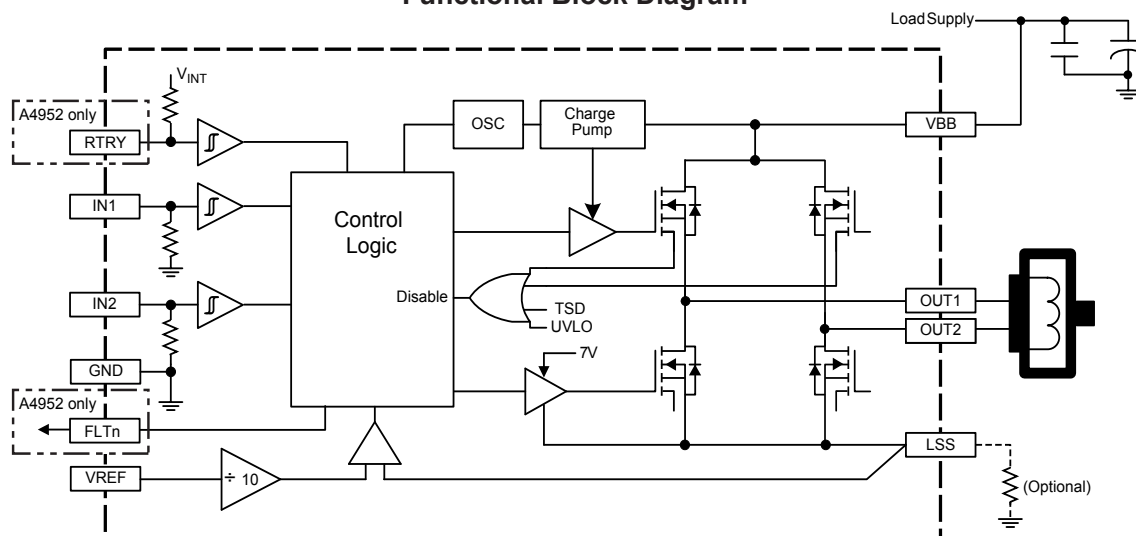
Designed for pulse width modulated (PWM) control of DC motors, the A4952 and A4953 are capable of peak output currents to ± 2 A and operating voltages to 40 V.

Input terminals are provided for use in controlling the speed and direction of a DC motor with externally applied PWM control signals. Internal synchronous rectification control circuitry is provided to lower power dissipation during PWM operation.

Internal circuit protection includes overcurrent protection, motor lead short to ground or supply, thermal shutdown with hysteresis, undervoltage monitoring of V_{BB} , and crossover-current protection.

The A4952 is provided in a low-profile 10-pin MSOP package (suffix LY) and the A4953 is provided in a low-profile 8-pin SOICN package (suffix LJ). Both packages have an exposed thermal pad, and are lead (Pb) free, with 100% matte tin leadframe plating.

Functional Block Diagram



C Exemple de code Arduino

[Télécharger le code](#)

```
1  /*#####
2  Author:
3  * Mirko Prosseda (01-2018)
4  * email: mirko.prosseda@gmail.com
5  *
6  Description:
7  * Dual DC Motor Driver 30V 4A V3 test sketch v1.0
8  * Speed and direction for each motor channel are regulated by
9  * potentiometers attached to Arduino pins A0 and A1
10 *
11 Connections:
12 * BOARD -> ARDUINO
13 * IN1     -> 3
14 * IN2     -> 5
15 * IN3     -> 6
16 * IN4     -> 9
17 * GND     -> GND
18 #####*/
19
20
21 // Define constants and variables
22 const int Dir1a = 3;
23 const int Dir1b = 5;
24 const int Dir2a = 6;
25 const int Dir2b = 9;
26 const int analogPin1 = A0;
27 const int analogPin2 = A1;
28
29 int analogValue1;
30 int analogValue2;
31 byte pwmValue1;
32 byte pwmValue2;
33
34 // Initialization
35 void setup()
36 {
37   pinMode(Dir1a, OUTPUT);
38   pinMode(Dir1b, OUTPUT);
39   pinMode(Dir2a, OUTPUT);
40   pinMode(Dir2b, OUTPUT);
41   digitalWrite(Dir1a, 0);
42   digitalWrite(Dir1b, 0);
43   digitalWrite(Dir2a, 0);
44   digitalWrite(Dir2b, 0);
45 }
```



```
46 |
47 | // main loop
48 | void loop()
49 | {
50 |     // read both potentiometers values
51 |     analogValue1 = analogRead(analogPin1);
52 |     analogValue2 = analogRead(analogPin2);
53 |
54 |     // apply direction adjustment for channel 1
55 |     if(analogValue1 > 512)
56 |     {
57 |         pwmValue1 = (analogValue1 - 512)/2; // evaluate new pwm value
58 |         analogWrite(Dir1a,pwmValue1);
59 |         digitalWrite(Dir1b, LOW);
60 |     }
61 |     else
62 |     {
63 |         pwmValue1 = (511 - analogValue1)/2; // evaluate new pwm value
64 |         digitalWrite(Dir1a, LOW);
65 |         analogWrite(Dir1b,pwmValue1);
66 |     }
67 |
68 |     // apply direction adjustment for channel 2
69 |     if(analogValue2 > 512)
70 |     {
71 |         pwmValue2 = (analogValue2 - 512)/2; // evaluate new pwm value
72 |         analogWrite(Dir2a,pwmValue2);
73 |         digitalWrite(Dir2b, LOW);
74 |     }
75 |     else
76 |     {
77 |         pwmValue2 = (511 - analogValue2)/2; // evaluate new pwm value
78 |         digitalWrite(Dir2a, LOW);
79 |         analogWrite(Dir2b,pwmValue2);
80 |     }
81 | }
```