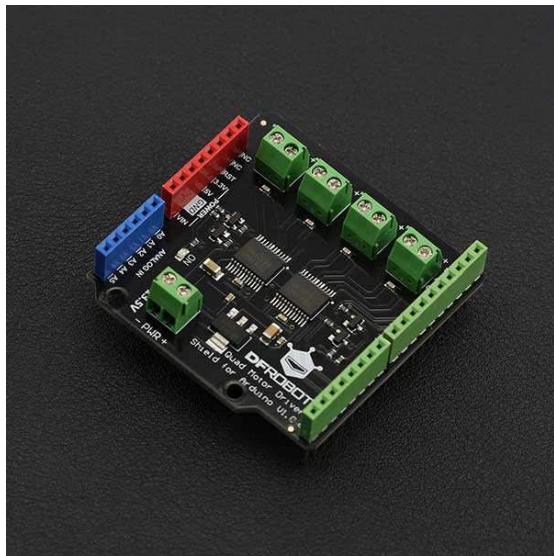




Quad Motor Driver Shield for Arduino SKU:DRI0039

From Robot Wiki



Contents

- 1 Introduction
- 2 Specification
- 3 Board Overview
- 4 Tutorial
 - 4.1 You Will Need
 - 4.2 Connection Diagram
 - 4.3 Sample Code
 - 4.4 Result
- 5 FAQ

Introduction

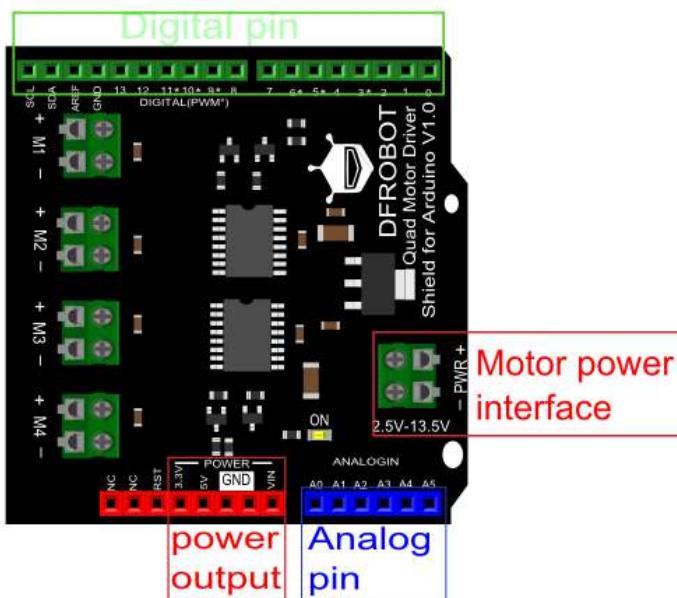
The latest quad motor driver shield for Arduino, compatible with 5V/3.3V Arduino controller, can control up to four DC motors with 8 pins at the same time. The shield supports PWM speed control and polarity control. The driver shield includes two TB6612FNG motor driver chips, compared with the traditional L298N chip, efficiency is improved and the component size also greatly reduced. The chip doesn't heat in to the rated range, and a single path maximum output 1.2A continuous current. The module includes a built-in low voltage detection circuit and thermal shutdown protection circuit, which is safe and reliable. This module is suitable for all kinds of DIY production.

Specification

- VM Motor Power Supply: 2.5 V ~ 13.5 V
- VCC Logic Power Supply: 2.7 ~ 5.5 V
- Output Current: 1.2A continuous current (per channel)
- Start/Peak Current: 2A (continuous pulse) / 3.2A (single pulse)
- Arduino Control Port: Digital pins 3, 4, 5, 6, 7, 8, 11, 12
- Working Temperature: -20 ~ 85 °C
- Dimensions: 53mm x 52mm/2.09 x 2.05

Board Overview

Motor I/O					
Motor	Direction	Forward	Backward	Speed	Speed range
M1	4	LOW	HIGH	3	0-255
M2	12	HIGH	LOW	11	0-255
M3	8	LOW	HIGH	5	0-255
M4	7	HIGH	LOW	6	0-255

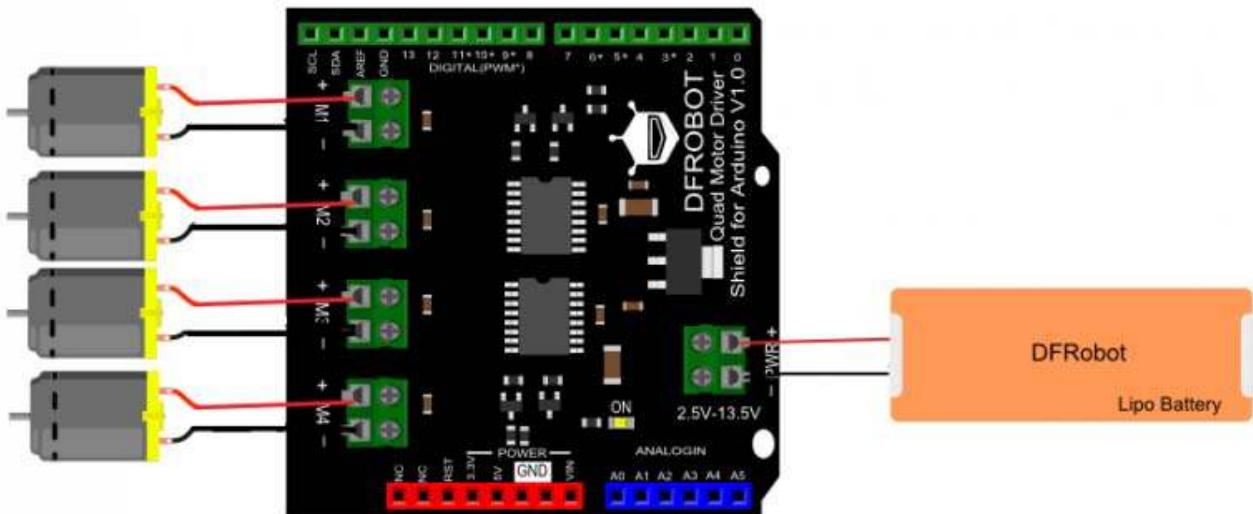


Tutorial

You Will Need

- **Hardware**
 - DFRduino UNO R3 x 1
 - Quad Motor Shield for Arduino x 1
 - DC motor x 4
 - Jumper Wires
- **Software**
 - Arduino IDE [Click to download Arduino IDE](https://www.arduino.cc/en/Main/Software) <https://www.arduino.cc/en/Main/Software>

Connection Diagram



Sample Code

Sample Code

```
1 /*!
2 * @file QuadMotorDriverShield.ino
3 * @brief QuadMotorDriverShield.ino Motor control program
4 *
5 * Every 2 seconds to control motor positive inversion
6 *
7 * @author linfeng(490289303@qq.com)
```

```
8 * @version V1.0
9 * @date 2016-4-5
10 */
11 const int E1 = 3; //<Motor1 Speed
12 const int E2 = 11; //<Motor2 Speed
13 const int E3 = 5; //<Motor3 Speed
14 const int E4 = 6; //<Motor4 Speed
15
16 const int M1 = 4; //<Motor1 Direction
17 const int M2 = 12; //<Motor2 Direction
18 const int M3 = 8; //<Motor3 Direction
19 const int M4 = 7; //<Motor4 Direction
20
21
22 void M1_advance(char Speed) //<Motor1 Advance
23 {
24   digitalWrite(M1,LOW);
25   analogWrite(E1,Speed);
26 }
27 void M2_advance(char Speed) //<Motor2 Advance
28 {
29   digitalWrite(M2,HIGH);
30   analogWrite(E2,Speed);
31 }
32 void M3_advance(char Speed) //<Motor3 Advance
33 {
34   digitalWrite(M3,LOW);
35   analogWrite(E3,Speed);
36 }
37 void M4_advance(char Speed) //<Motor4 Advance
38 {
39   digitalWrite(M4,HIGH);
40   analogWrite(E4,Speed);
41 }
```

```
42
43 void M1_back(char Speed) ///<Motor1 Back off
44 {
45   digitalWrite(M1,HIGH);
46   analogWrite(E1,Speed);
47 }
48 void M2_back(char Speed) ///<Motor2 Back off
49 {
50   digitalWrite(M2,LOW);
51   analogWrite(E2,Speed);
52 }
53 void M3_back(char Speed) ///<Motor3 Back off
54 {
55   digitalWrite(M3,HIGH);
56   analogWrite(E3,Speed);
57 }
58 void M4_back(char Speed) ///<Motor4 Back off
59 {
60   digitalWrite(M4,LOW);
61   analogWrite(E4,Speed);
62 }
63
64
65
66 void setup() {
67   for(int i=3;i<9;i++)
68     pinMode(i,OUTPUT);
69   for(int i=11;i<13;i++)
70     pinMode(i,OUTPUT);
71 }
72
73 void loop() {
74   M1_advance(100);
75   M2_advance(100);
```

```
76 M3_advance(100);  
77 M4_advance(100);  
78 delay(2000); //;<Delay 2S  
79 M1_back(100);  
80 M2_back(100);  
81 M3_back(100);  
82 M4_back(100);  
83 delay(2000); //;<Delay 2S  
84 }
```

Result

You should see the motor go forwards and backwards every two seconds

FAQ

For more questions or interesting projects, you can [Visit the forum!](#)