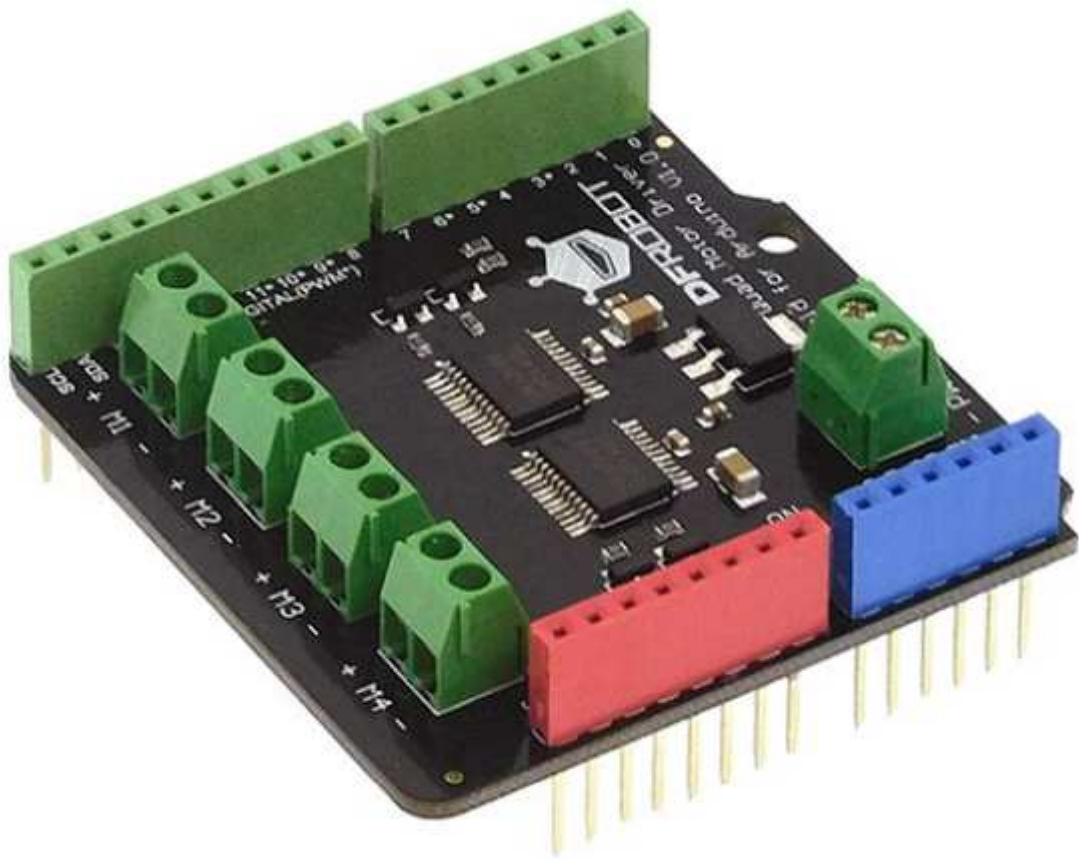


# Shield moteurs

## CC 4 x 1,2 A DRI0039



## ***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.

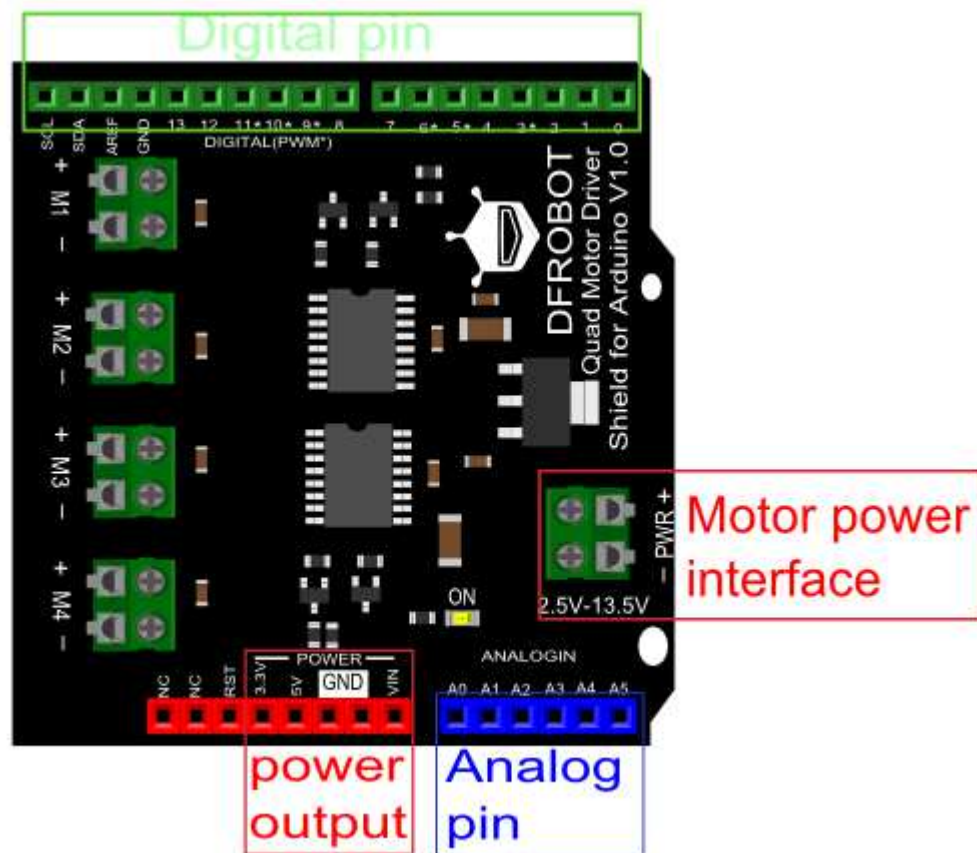
This module is suitable for all kinds of DIY production.

## ***Specification***

---

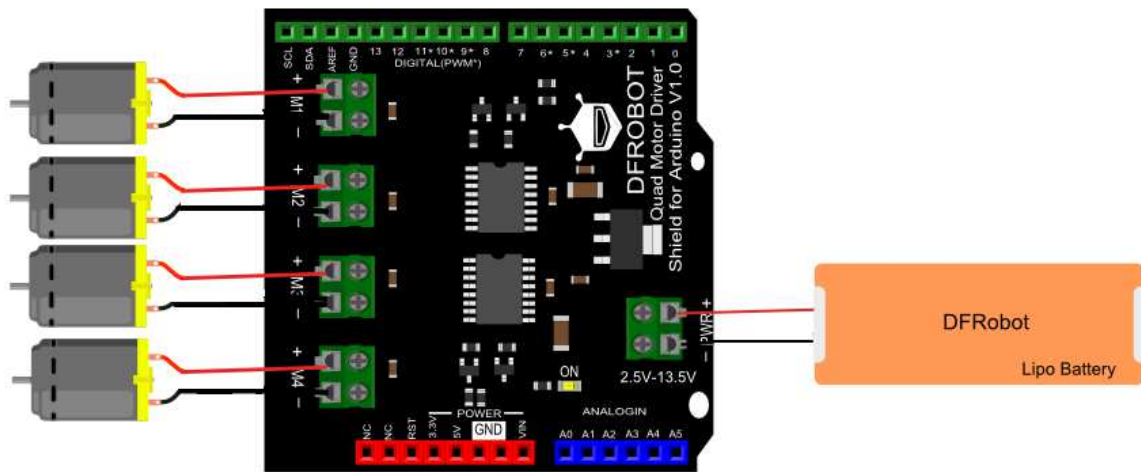
- VM Motor Power Supply : 2,5 V ~ 13,5 V
- VCC Logic Power Supply : 2,7 ~ 5,5 V
- Output Current : 1,2A continuous current (per channel)
- Start/Peak Current : 2A (continuous pulse) / 3,2A (single pulse)
- Arduino Control Port : Digital pins 3, 4, 5, 6, 7, 8, 11, 12
- Working Temperature : -20 ~ 85 °C
- Dimensions : 53mm x 52mm / 2,09 x 2,05
- Weight : 24g

## Board Overview



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

## Connection diagram



## Sample code – Arduino

```
/*!  
 * @file QuadMotorDriverShield.ino  
 * @brief QuadMotorDriverShield.ino Motor control program  
 *  
 * Every 2 seconds to control motor positive inversion  
 *  
 * @author linfeng(490289303@qq.com)  
 * @version V1.0  
 * @date 2016-4-5  
 */  
const int E1 = 3; ///const int E2 = 11; ///const int E3 = 5; ///const int E4 = 6; ///  
const int M1 = 4; ///const int M2 = 12; ///const int M3 = 8; ///const int M4 = 7; ///  
void M1_advance(char Speed) ///{
```

```

    digitalWrite(M1,LOW);
    analogWrite(E1,Speed);
}
void M2_advance(char Speed) ///

```

```
void loop() {  
  M1_advance(100);  
  M2_advance(100);  
  M3_advance(100);  
  M4_advance(100);  
  delay(2000); ///  
  M1_back(100);  
  M2_back(100);  
  M3_back(100);  
  M4_back(100);  
  delay(2000); ///  
}
```