



15A Single DC Motor Driver

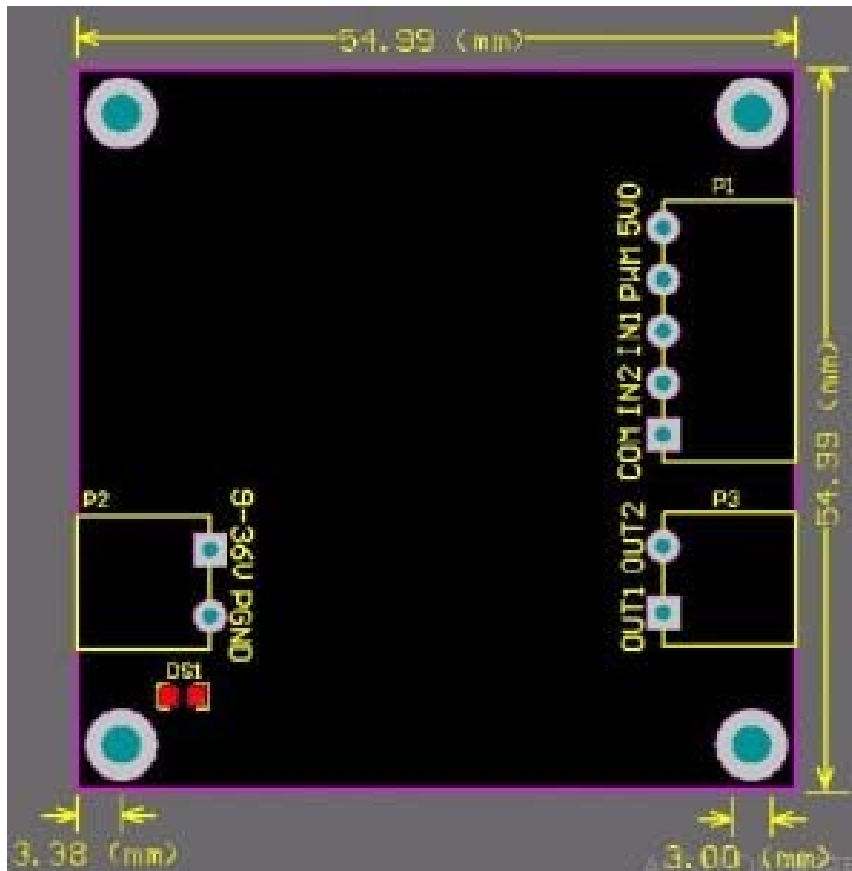
SKU: DRI0042



Introduction

This is an ultra small **DC motor driver** for small scale projects using the popular IC - LM2575. This provides a powerful way to drive 36V @ 15A motors. UVLO (Under Voltage Latch-Out) features a safe guard for your system. It is very considerate to own an 5V output for Arduino or any other devices, and its interface includes ESD (electro-static discharge) protection.

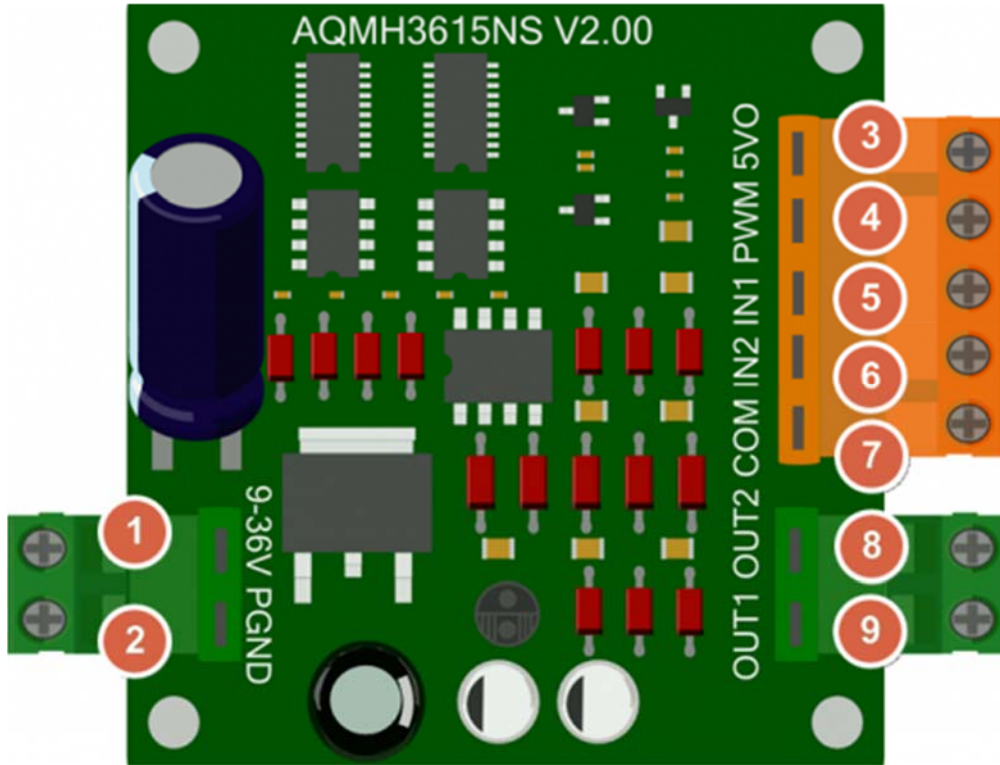
Specification



Dimension

- Voltage supply: DC 12 ~ 36 V
- Voltage supply limit: 6.5 ~ 40 V
- Control signal Level (Compatible 3.3V/5V)
High: DC 2.0 ~ 5.5 V
Low: DC 0 ~ 0.8 V
- **Maximum continuous operating current:**
12A (no heat sink)
15A (with simple heat sink)
20A (with Thick Heat Sink/ Cooling System)
- Peak current: 110 A
- Speed control: PWM
Minimum valid Pulse Width: 200 ns
- Maximum Output Current @5V: 500 mA
- Working Temperature: -25 ~ 85 °C
- Mounting Hole: M3
- Dimension (Length * Width * Height): 55 x 55 x 20(mm)/2.165 x 2.165 x 0.787 (in)
- Weight: 42g

Board Overview



Num	Label	Description
1	9 - 36V	Power Supply, +
2	PGND	Power Supply, GND/ -
3	5VO	5V Output, you can use this power to feed Arduino through Arduino's 5V port, NOT Vin which requires 7-12V input
4	PWM	Speed control signal input
5	IN1	Motor steering control signal input 1, Note1
6	IN2	Motor steering control signal input 2
7	COM	GND
8	OUT2	Motor_+
9	OUT1	Motor_-

Control Method

IN1	IN2	PWM	OUT1, OUT2 Motor Behavior
0	0	x	Stop
1	1	x	Vacant
1	0	1	Forward 100%
0	1	1	Reverse 100%
1	0	PWM	Forward at PWM speed
0	1	PWM	Reverse at PWM speed

In this table

- "0": TTL_Low
- "1": TTL_High
- "PWM": PWM speed setting
- "x": Any TTL, and it is default TTL_Low while no PWM signal.

Note1: IN1 & IN2

To protect your motor, before switching the motor steering direction, make sure firstly to BRAKE motor by setting **IN1 - 0 & IN2 - 0**, especially when the PWM was set as 100%, full speed. And the suggestion time to brake is >0.1S, depending on your motor.

Tutorial

Here you will learn how to use PWM signal to control the motor through this motor driver, do some wiring according to the Connection Diagram below, and then upload the sample code below to the Arduino card, here we use a UNO as the controller, you could see your motor run forward for 3 second and then run reversely for another 3 seconds and repeat this behavior then.

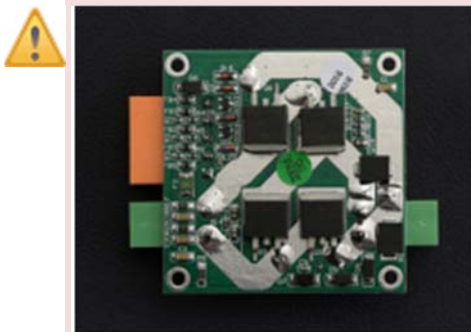
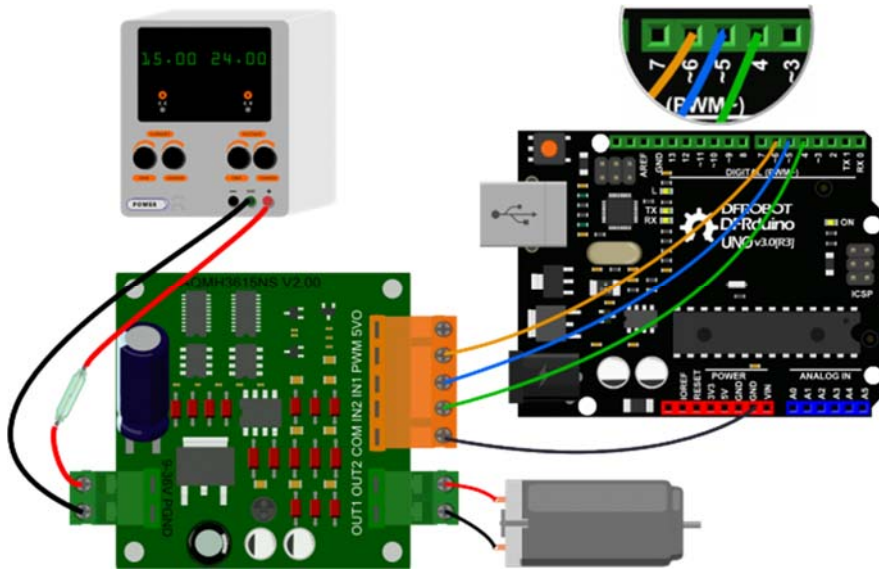
This tutorial will cover how to use PWM to control a motor using the shield.

Requirements

- **Hardware**
 - 1 x **DFRduino UNO** (or similar)
 - 1 x **DC Motor Driver**
 - 1 x DC Motor
 - 1 x Fuse@20A
 - Jumper wires

- **Software**
 Arduino IDE, Click to Download Arduino IDE from Arduino® <https://www.arduino.cc/en/Main/Software%7C>

Connection Diagram



Back view, conductive layer

Facility Safety and the Personal Safety:

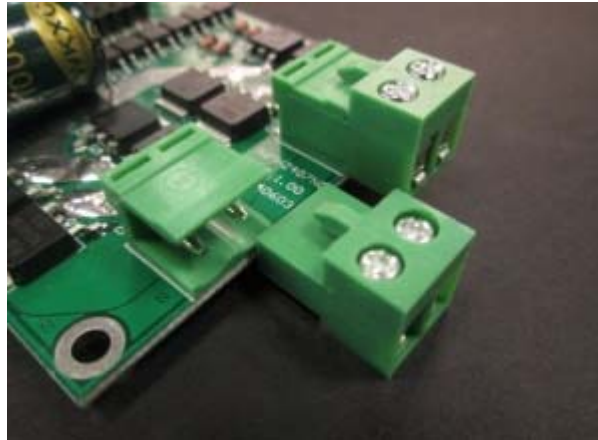
Please add a fuse@20A between the Power source and the module (9-36V).
 The 5VO port is for 5V output for other module, can NOT be used as Power_In.

Installation Tips

The module needs a cooling system once your motor runs over 15 amps, like [Electronics Blower Fan](#), [Heatsink](#), etc.
 The [backside](#) of the driver, as right-hand picture shows, that its conducting layer can be very wide, so you should take some measurements to ensure it won't **short-circuit**, like applying 1mm layer conductive epoxy or any other similar materials.

Important: Pluggable Connector

You may find that the connector is designed as pluggable type to adapt with Male or Female wires.



Sample Code

```
/*
 * @file Motor driver DRI0042_Test.ino
 * @brief DRI0042_Test.ino Motor control program
 *
 * control motor positive inversion
 *
 * @author bernie.chen@dfrobot.com
 * @version V1.0
 * @date 2016-8-10
 */
const int IN1=5;
const int IN2=4;
const int PWM=6;

void setup() {
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(PWM, OUTPUT);
}
```

```

void loop() {
  Motor_Brake();
  delay(100);
  Motor_Forward(200); //Forward, PWM setting 0-255
  delay(3000);
  Motor_Brake();
  delay(100);
  Motor_Backward(200); //Reverse, PWM setting 0-255
  delay(3000);
}

void Motor_Forward(int Speed) {
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  analogWrite(PWM,Speed);
}

void Motor_Backward(int Speed) {
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  analogWrite(PWM,Speed);
}

void Motor_Brake(){
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);
}

```

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum**