

ECE 480, Design Team 3



Pololu High-Power Motor Driver 18v15

Using H-bridge motor driver (Pololu High-Power Motor Driver 18v15) to control the direction of the DC motor.

Application Note

Name: Ali Alsatarwah

Date: 04/04/2011

Facilitator: Prof. Dr. Lixin Dong

Executive Summary:

The Pololu High-Power Motor Driver 18v15 is a discrete MOSFET H-bridge driver enables to controls DC motors in two directions. Pololu High-Power Motor Driver 18v15 is also can be used to control the power (speed) of the DC motor by varying the Pulse –width modulation (PWM) input. This application note is intended to help the user understand how to use this device in application that need to control the direction and the speed of DC motor. An example of an application uses Arduino to send input to the Pololu High-Power Motor Driver 18v15 in order to control high power DC motor is included in this application note.

Keywords: Pulse –width modulation, Arduino, DC motor

Table of contents:

Introduction & Background.....	1
Using the the Pololu High-Power Motor Driver 18v15:.....	2-4
Notes before using the the Pololu High-Power Motor Driver 18v15.....	2
Dimensions.....	3
Basic Pins connection.....	3-4
Pin Connection (Direction and speed control).....	4
Truth Table for the Pololu High-Power Motor Driver 18v15 direction control.....	4
Issue with the Pololu High-Power Motor Driver 18v15.....	4
Sample code.....	5-7
-Arduino Fio with Xbee Radio Examples.....	5
- Overview.....	5
- programing Code.....	5-7
Reference.....	8

Introduction and background:

H-bridge is used in many important applications that often related to controlling robotics units and allows DC motors to run forward and backward. They come in different sizes and shapes depending on the application the used in. Most H-bridges for low voltage and low currents come in integrated circuits that have several pins. Others H-bridges that used in high current and high voltage applications built from discrete components or quite large integrated circuits. H-bridge which also called “Full Bridge” is named because it has four switching elements at the “corners” of the H a motor forms the cross bar.

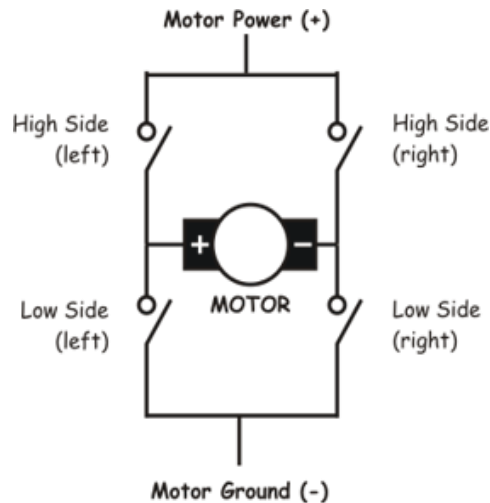


Figure1: Basic H-Bridge

The basic operation of mode of an H-bridge is fairly simple: if (low side left) and (high side right) are turned on the left lead of the motor will be connected to ground, while the right lead is connected to the power supply. Current starts flowing through the motor which energizes the motor in (let's say) the forward direction and the motor shaft starts spinning. If (high side left) and (low side right) are turned on, the converse will happen, the motor gets energized in the reverse direction, and the shaft will start spinning in that way.

The Pololu High-Power Motor Driver 18v15 is build based on discrete MOSFET H-bridge. The H-bridge is made up of one N-channel MOSFET per leg, and most of the board's performance is determined by these MOSFETs (the rest of the board contains the circuitry to take user inputs and control the MOSFETs).

Using The Pololu High-Power Motor Driver 18v15:

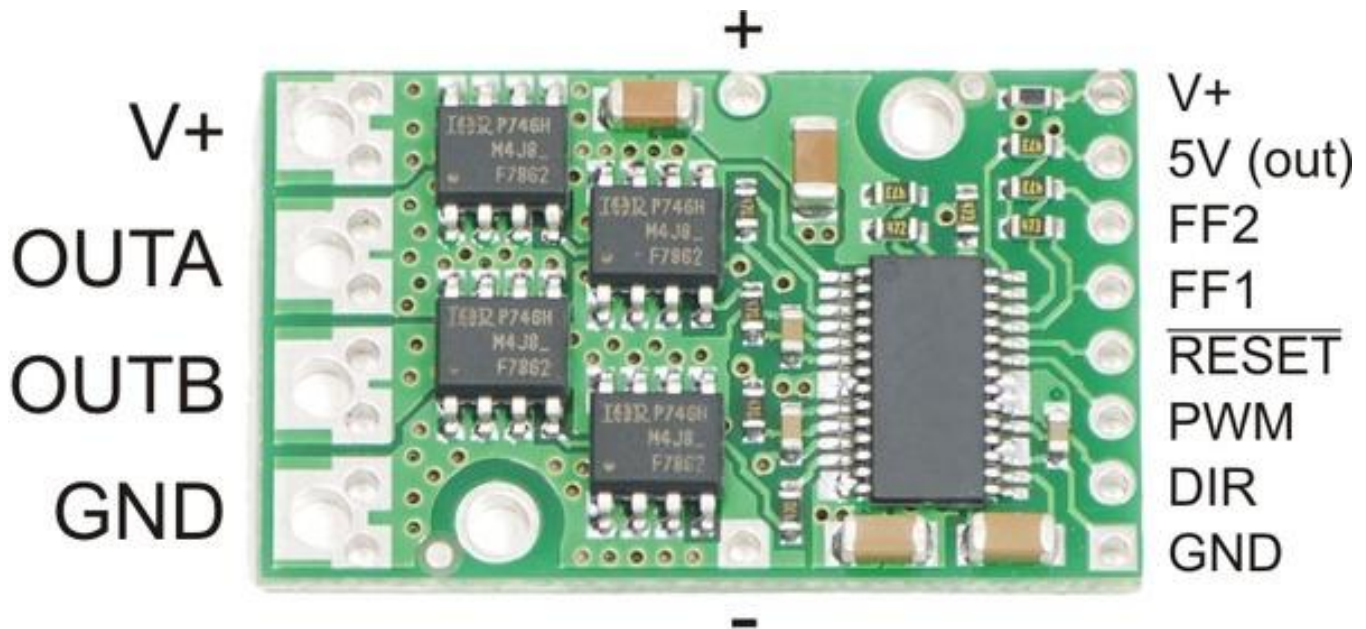


Figure 2: The top view of The Pololu High-Power Motor Driver 18v15

A)Notes:

-Capacitor:

The Pololu High-Power Motor Driver 18v15 comes in kit as in the figure above. Though, a capacitor connected across the power supply is recommended either between the “+” and “-” signs in the figure above or directly between the power supply. An option of the capacitor value is 150 uF. The reason it is better to use a capacitor with application that has DC motors, is that adding it would let the supply charge up the capacitor to the output voltage of the supply. Then, when the motor starts, the capacitor can quickly drive the current into the motor within a short period, which will help the motor to start up quicker than if it had to depend on the power supply alone.

-Heat sink:

The Pololu High-Power Motor Driver 18v15 can drive up to 15 A of continuous current without need for heat sink. With addition of heat sink to the MOSFET the motor driver can be run at up to 21 A continuous current.

-PWM Frequency:

The maximum Pulse –width modulation frequency can the The Pololu High-Power Motor Driver 18v15 support without any switching losses in the The Pololu High-Power Motor Driver 18v15 is 40 kHz.

-Dead time:

The Pololu High-Power Motor Driver 18v15 has a dead time of about 3 us per cycle resulting in unavailability of cycles at higher frequencies.

-Locked-antiphase operation:

Another mode of operation to control the DC motor through the Pololu High-Power Motor Driver 18v15 using the PWM is locked- antiphase mode. This mode uses the Pulse –width modulation applied to the DIR with PWM held high. In locked antiphase the PWM is applied to both H- bridge legs simultaneously. If the PWM is the same duty cycle (percent ON) then of course the average is ZERO and your h-bridge does not

produce any DC. If your h-bridge (Pololu High-Power Motor Driver 18v15) were connected to a motor, the motor would not turn. However, if the PWM duty cycle were to change, such that one h-bridge leg got 5% more ON time than the other, then the motor would rotate slowly in one direction. If that same leg received less PWM duty cycle, say -45% then the motor would spin VERY fast in the opposite direction.

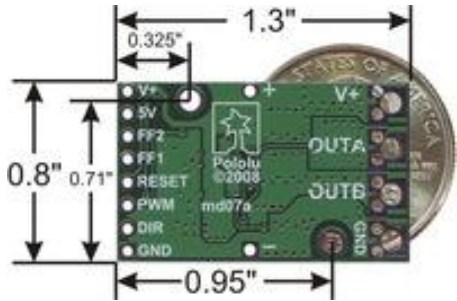
-Fault Conditions:

The Pololu High-Power Motor Driver 18v15 can detect states that reported from pins FF1 and FF2. The three fault states are short circuit output, over temperature, and under voltage state. The truth table of each state for the Pololu High-Power Motor Driver 18v15 is:

FF1	FF2	Fault state	Disable outputs	Latched until reset
L	L	NO FAULT	NO	NO
L	H	SHORT CIRCUIT	YES	YES
H	L	OVER TEMPERTUER	NO	NO
H	H	UNDER VOLTAGE	YES	NO

B) Dimensions:

Size	1.3"x0.8"
Weight	2.8g



C) Pins connection:

PIN	Default state	Details
V+		This is the main 5.5 – 30 V motor power supply connection, which should typically be made to the larger V+ pad.
5V (out)		This regulated 5V output provides a few milliamps. This output should not be connected to other external power supply lines.
GND		Ground connection for logic and motor power supplies.
RESET	HIGH	The reset pin, when pulled low, puts the board into a low-power sleep mode and clears any latched fault flags.

D) Pin Connection (Direction and speed control):

PIN	Default state	Details
OUTA		A Motor output pin (A,B for the same motor)
OUTB		A Motor output pin (A,B for the same motor)
PWM	Low	PWM signal on this pin corresponds to a PWM output on the motor outputs. The PWM duty cycle controls the speed of the motor.
DIR	FLOAT	When DIR set low a current flow from OUT A-OUTB, otherwise the current flow from OUTB-OUTA.

E)Truth Table for the Pololu High-Power Motor Driver 18v15 direction control.

PWM	DIR	OUTA	OUTB	Operation
H	L	L	H	FORWARD
H	H	H	L	BACKWARD
L	X	L	L	BRAKE

*The Maximum logic Voltage: 5.5V

*High=above 2.3

*Low= lower than 2.3

*Break=2.3

According to the device testing in the lab.

Issues with the Pololu High-Power Motor Driver 18v15:

Reverse power protection or any over-current or over-temperature protection are not available in the Pololu High-Power Motor Driver 18v15. This may cause damage to the Pololu High-Power Motor Driver 18v15 when an overcurrent has been drawn through the motor drive for whatever reason. A solution to this issue is by adding external current sensors to the Pololu High-Power Motor Driver 18v15.

Sample Code:

This section contains an example code for controlling a DC motor by sending different Pulse –width modulation inputs to the Pololu High-Power Motor Driver 18v15 from the The Arduino Fio with Xbee Radio. The example uses a joystick that delivers different voltage depending on the position of the joystick. This voltage is sent to the Arduino which would convert into duty cycle that can be represented as Pulse –width modulation that supply into the Pololu High-Power Motor Driver 18v15 in order to vary the speed of the DC motor.

1)Arduino Fio with Xbee Radio Examples:

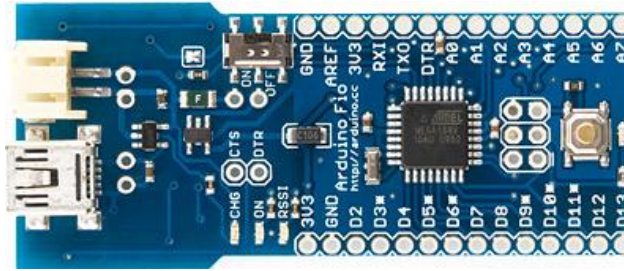


Figure 3: The top view of the Arduino Fio microcontroller

2)Overview

The Arduino Fio is a microcontroller board based on the ATmega328P runs at 3.3V and 8 MHz. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 8 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. It has connections for a Lithium Polymer battery and includes a charge circuit over USB. An XBee socket is available on the bottom of the board so Arduino Fio can be programmed wirelessly. The main purpose of the Arduino Fio is wireless communication or wireless control.

3)Simple Example:

This example assumed the following connections exist between the Arduino Fio and the Pololu High-Power Motor Driver 18v15:

- Arduino Fio digital pins to the PWM input top the Pololu High-Power Motor Driver 18v15.
PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

Any pin could be choosing from the PWM pins above with taking in account that any change in the pin number requires change pin definition in the code of the sample program.

```
void setup()
{
  Serial.begin(57600); // Begins serial transmission at a rate
  pinMode(13,OUTPUT); // of 9600 bps
  pinMode(3,OUTPUT); //Set up digital pins 3,5,6,9,10,13 for output
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}
int duty = 127; //Declare and initialize the gripper duty cycle
                // 127 is the starting position, range is 63-153

void loop()//Begin main loop
{
  //Check for at least two things in the serial buffer
  // because the protocol sends a character followed by a value
  if(Serial.available() >=2)
  {
    byte letter = 0; //declare/initialize the letter variable
    if(Serial.peek() == 'a')//check the value without consuming it
    {
      Serial.read();//consume it
```

```

if(Serial.peek() != 'a' || //make sure the next thing in the buffer
  Serial.peek() != 'b' || // isn't a letter, this is in case of packet
  Serial.peek() != 'c' || // loss
  Serial.peek() != 'd' ||
  Serial.peek() != 'e')
{
  //The gripper requires a duty cycle between 25% and 60%
  // to go from closed to all the way open. This
  // corresponds to an analogWrite between 63 and 153
  duty = Serial.read() * 0.352 + 63;
  if(duty < 63){duty=63;}
  else if(duty>153){duty=153;}
  analogWrite(3,duty);
}
}

else if(Serial.peek() == 'b')
{
  Serial.read();
  if(Serial.peek() != 'a' ||
    Serial.peek() != 'b' ||
    Serial.peek() != 'c' ||
    Serial.peek() != 'd' ||
    Serial.peek() != 'e')
  {
    // analogWrite(5,Serial.read());
  }
}

else if(Serial.peek() == 'c')
{
  Serial.read();
  if(Serial.peek() != 'a' ||
    Serial.peek() != 'b' ||
    Serial.peek() != 'c' ||
    Serial.peek() != 'd' ||
    Serial.peek() != 'e')
  {
    //analogWrite(6,Serial.read());
  }
}

else if(Serial.peek() == 'd')
{
  Serial.read();
  if(Serial.peek() != 'a' ||
    Serial.peek() != 'b' ||
    Serial.peek() != 'c' ||
    Serial.peek() != 'd' ||
    Serial.peek() != 'e')
  {
    //analogWrite(9,Serial.read());
  }
}

else if(Serial.peek() == 'e')
{
  Serial.read();
  if(Serial.peek() != 'a' ||
    Serial.peek() != 'b' ||
    Serial.peek() != 'c' ||
    Serial.peek() != 'd' ||

```

```

        Serial.peek() != 'e')
    {
        //analogWrite(10,Serial.read());
    }
}
else
{
    Serial.read();
}
}
}
else//If no letter appears in the serial buffer
{ //If a non letter is in the serial buffer, throw it away
    if(Serial.available())
    {
        Serial.read();
    }
    //If no letters are writte, shut the motors down
    analogWrite(3,127);//127 is idle position for gripper
    analogWrite(5,0);
    analogWrite(6,0);
    analogWrite(9,0);
    analogWrite(10,0);
}
}
}

```

Reference

http://www.modularcircuits.com/h-bridge_secrets2.htm

www.maelabs.ucsd.edu/mae156/A/.../Motors-and-Capacitors.doc

<http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/>

<http://www.pololu.com/docs/0J44>

<http://www.arduino.cc/en/Main/ArduinoBoardFio>

<http://www.pololu.com/catalog/product/755>

<http://www.edaboard.com/thread3117.html>