

DC motors direction and speed control

by(Roman)

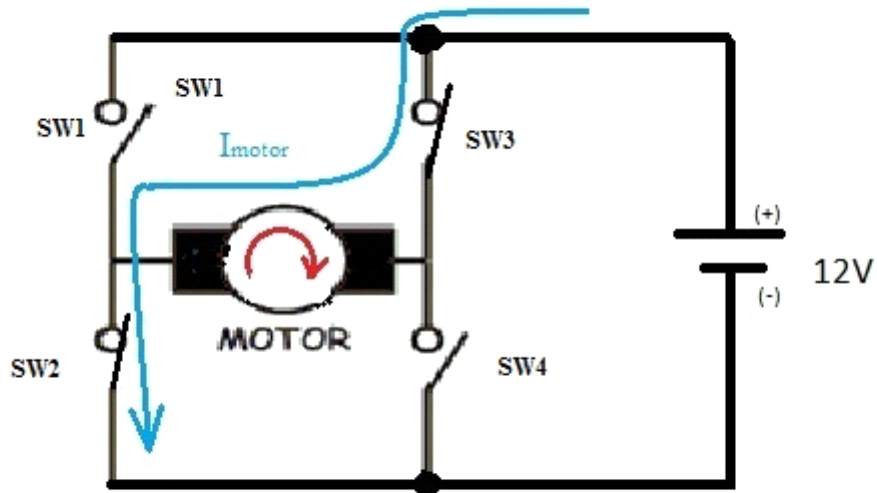
Purpose: This article is intended for beginners whose project contains DC motors that have power supply Voltage and Current rates higher than MCU (microcontroller) can provide. It will also cover controlling rotation speed using PWM signal generated by microcontroller. For better understanding basic electricity knowledge needed (such as what is capacitor and DC voltage polarity). At the end of this tutorial reader should be able to use L293D motor driver in his project for controlling at least 2 DC motors.

What for motor driver is needed?

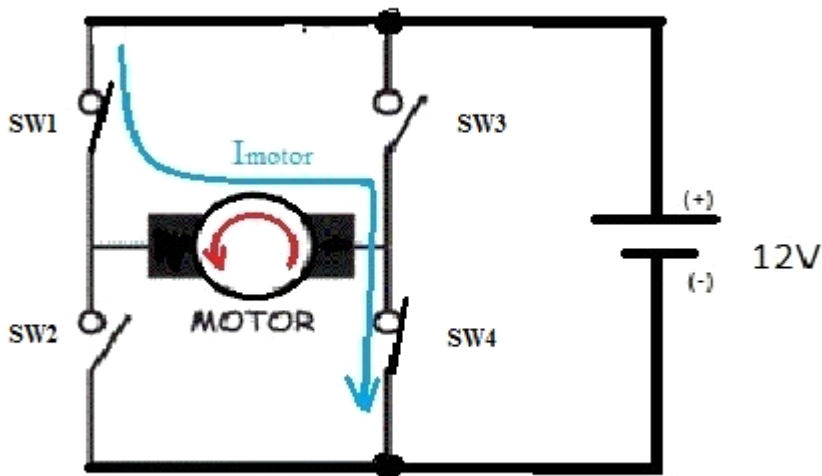
Usually microcontrollers are power supplied by low voltage power supply(see my previous article: AVR Power Supply) and in the meantime it's required to control DC motors at the higher Power consumption rate than controller can provide, so not to burn out you microcontroller you will need an intermediate device to convert voltage levels and steer high current.

DC Motors - polarity/direction and PWM/speed

As you already know DC motors rotation direction depends on the polarity of its power supply(ex. +/- forward , -/+ backward). So if we can control polarity of power supplied applied to motor we'll be able to control rotation direction. Simple way to make it is to use switchers like in the picture bellow.



And

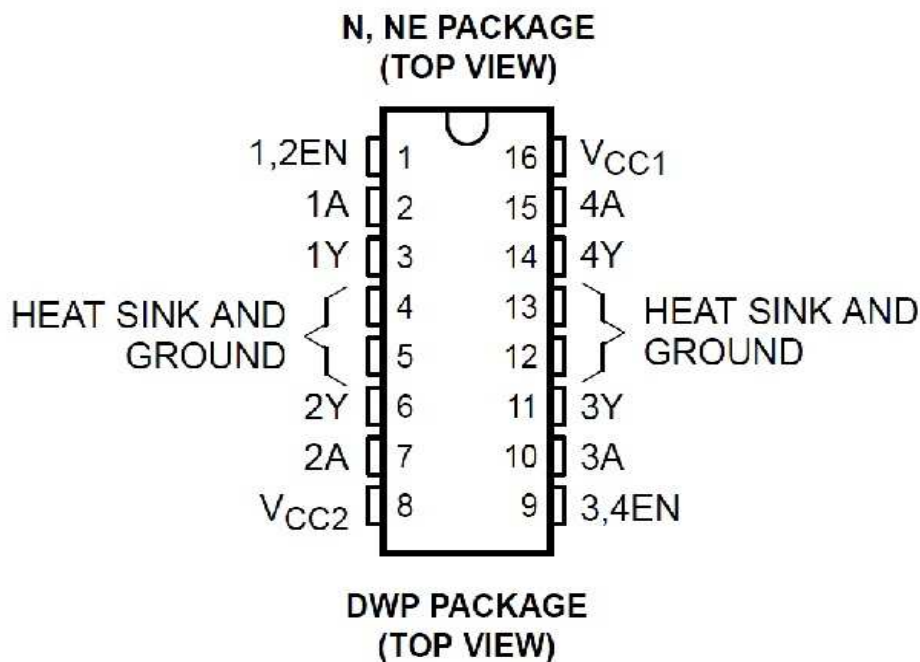


So as you can easily notice motor is spinning right when **switch 3** and **switch 2** is closed. And opposite in case of closing **switch 1** and **switch 4**. Need to be careful with simultaneous closing of **switch 1** and **switch 2**, the same **switch 3** and **switch 4** because short circuit will happen and definitely your battery will not like it at all.

Almost the same way is working and L293 H-Bridge motor driver. Its functionality is of course more complicated, but believe me, it's the same complicated like example before.

L293D by dissection

L293D from Texas Instruments is designed to provide bidirectional current up to 600mA at voltages from 4,5V to 36V. It has two independent channels (it means can control 2 DC motors) that are designed to work in positive-supply applications. Package view:



Every data that you will ever need you will find in its datasheet. I will present you only summary data that is needed for a simply current flow direction control. L293D allows control of any load with currents not bigger than 600mA; this current can rise up to 1A in case of using heat-sink.

Channel 1

PIN 1 - enables channel 1 for steering first load (in our case 12V dc motor)

PIN 2, 7 - logical level input (can be from MCU any I/O port pin)

PIN 3, 6 - Power output 1 (here we are connecting 12V DC motor)

Channel 2

PIN 9 - enables channel 2 for steering first load (in our case 12V dc motor)

PIN 10, 15 - logical level input (can be from MCU any I/O port pin)

PIN 11, 14 - Power output 2 (here we are connecting 12V DC motor)

Power supply

PIN 8 - Power supply for logic part of our bridge. From 4,5V to 7V.

PIN 16 - Power supply for controlled load. From 4,5 to 36V and up to 600mA

PIN 4, 5, 12, 13 - Ground(-) and heat sink connection.

As you can see L293D has two separate power supplies. How about logical inputs? These inputs have following requirements:

		MIN	MAX	UNIT
V_{IH}	High-level input voltage	2.3	7	V
V_{IL}	Low-level input voltage	-0.3	1.5	V
T_A	Operating temperature	0	70	°C

So it is perfect compatible with my ATmega169 which I supply with +3.3V and it obviously gives +3V for logic high. So you can directly connect I/O pins from your AVR controller to PINS 2, 7, 10, 15 of LD293D driver. Driver's PIN 1, 9(Enable/Disable) pins are also controlled with the same requirements like above. Right now remains to clarify what logical combinations to give to L293D input to control our motors.

INPUTS		OUTPUT
PIN 2(1A)	PIN1(Enable)	
1	1	1(+12V)
0	1	0(0V)
X	0	OFF

x- irrelevant state because driver is off.

So if to give on PIN 2, 7 of driver following combination we will have direction of motor rotation:

PIN 2(Input 1A)	PIN 7(input 2A)	PIN 1(Enable)	Motor
1	1	1	halt
1	0	1	forward
0	1	1	backward
0	0	1	halt
X	X	0	halt

X- irrelevant because driver is disabled. And motor is connected to PINS 3, 6 (1Y, 2Y).

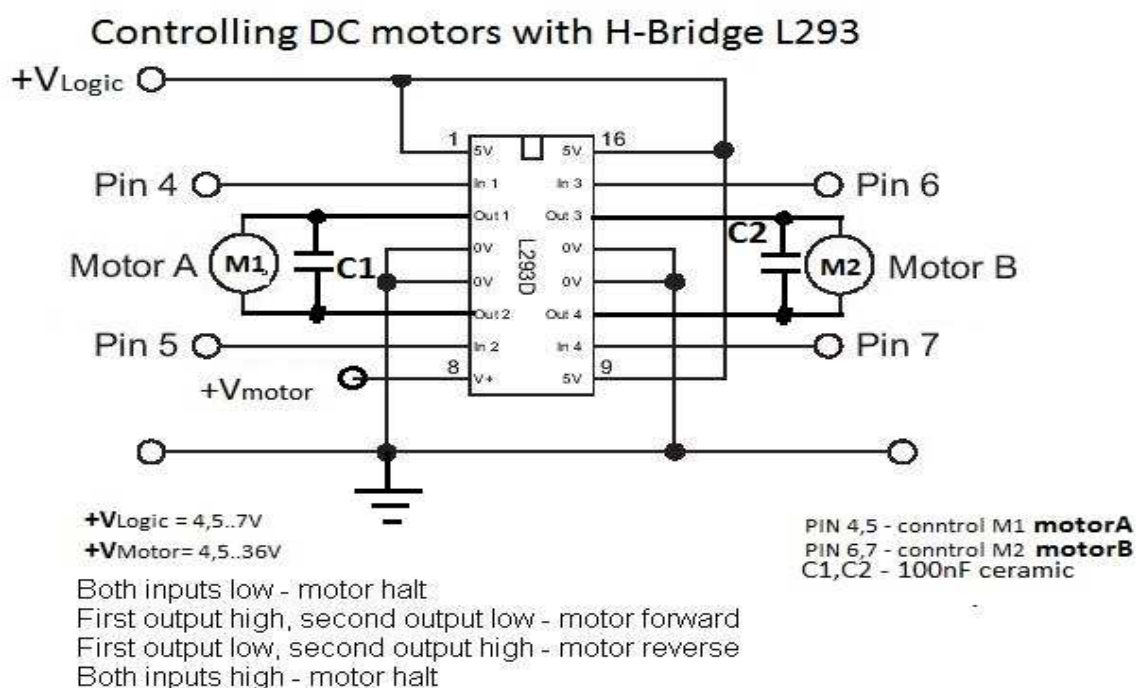
More detailed about enable feature of driver we'll discuss latter when speed control will be needed.

Time for practical application

You will need following components:

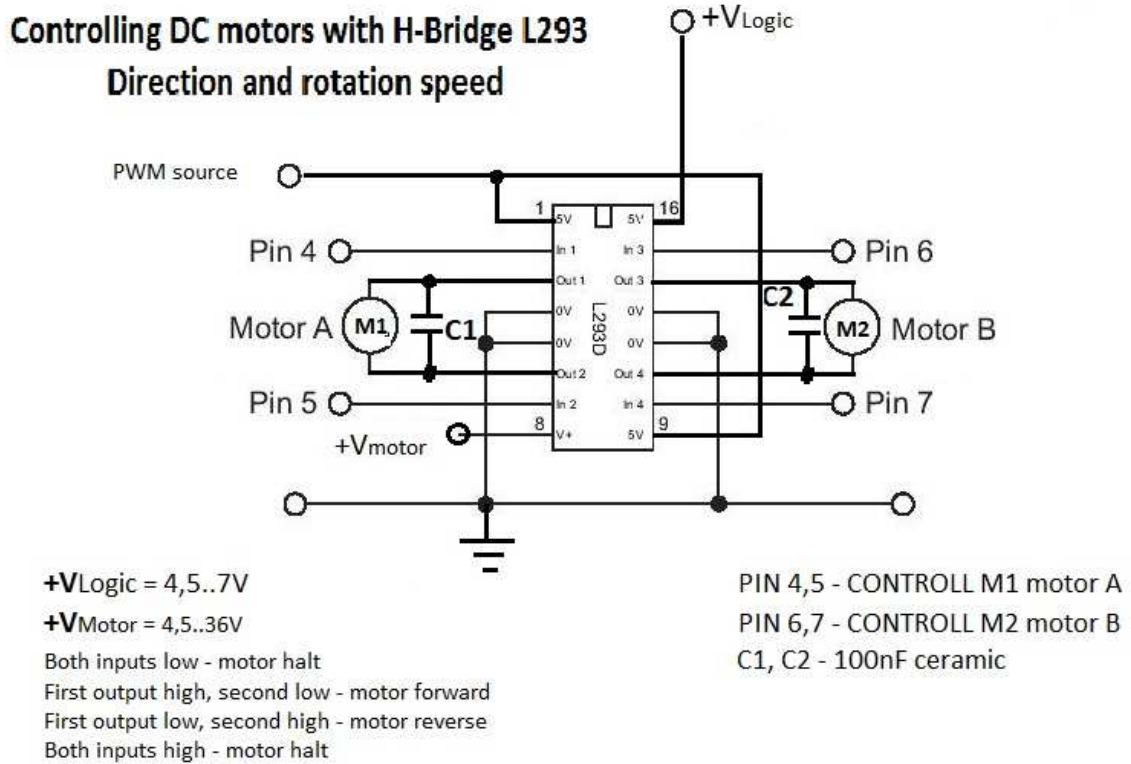
- L293D.....1 piece(≈1.80\$)
- Electrolytic capacitor $\geq 10\mu\text{F}$1 piece(≈0.30\$)

Now we need to assemble following scheme:



As you see enable pins are connected directly to the + power supply (forever High (1)) so we will not care about separate enable or disable driver. So in this case we'll control only rotation direction of the motors. No rotation speed control is available in this situation. Capacitors C1, C2 are needed to compensate noises produced by motor and also to prolong motor's lifetime. Immediate change of polarity applied to motors can cause big voltages backflow (in case of using motors of big power), because of motor inductive properties. they are called Output Clamp Diodes for Inductive Transient Suppression. So to protect your circuit from over voltages, diodes are needed. Our L293D already have this diodes build-in so you no need to worry about protection. Also driver have Thermal shutdown feature which better not to test. I don't mean that this feature is not working but anyway I don't like to overheat too much, my components. Now let's consider need to control motors speed.

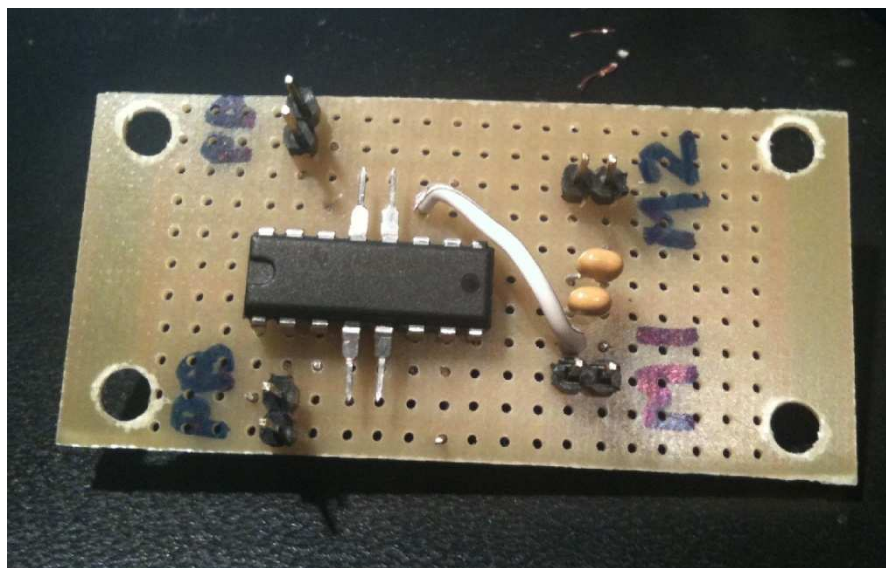
We will need somehow to apply PWM signal at 12V amplitude. The easiest way to make it, is to switch on/off full driver. Like in the scheme bellow:



so now applying PWM source directly from the I/O pin from MCU we can control DC motor rotation speed.

Soldering. Assembling all elements in together can be done on the universal PCB soldering board. Just to follow upper mentioned scheme. Do not forget capacitors.

Below is a picture of my amateur assembly of L293D driver



As you see I marked with PB and PD inputs for driver control and with M1 and M2 pins for motor's power.

What PWM frequency to choose

Every DC motor has its own electrical characteristics and there is no universal frequency for any DC motor, for hobby DC motors it is usually between 1kHz and 15kHz. I control my DC motors with 10kHz PWM signal generated by ATmega169 using Timers and Interrupts, with 14 steps of pulse width. It is more than enough in my robotic application. Remember when testing speed control on dc motors; it must be loaded (not at free run without any torque load) because you may not see the effect of slowing down.

Also be careful with motors operating current and stall current. Operating current is average current that your motor requires for functioning. And stall current is current drawn by you motor, normally powered up and stopped by force. This is the biggest current that your motor can draw. Why to pay attention at these currents? Because current is the main factor that makes your components to warm up and sometimes to burn.