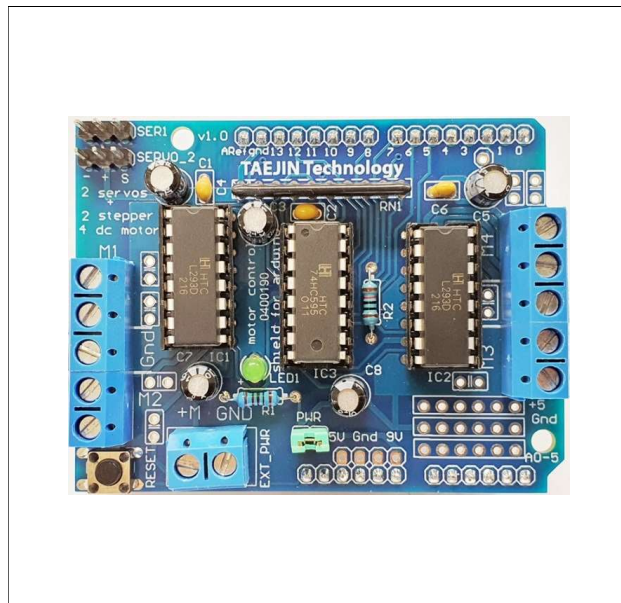


L293D Motor Shield

FEATURES

- 2 connections for 5V 'hobby' servos connected to the Arduino's high-resolution dedicated timer - no jitter
- 4 H-Bridges: L293D chipset provides 0.6A per bridge (1.2A peak) with thermal shutdown protection, internal kickback protection diodes. Can run motors on 4.5VDC to 25VDC.
- Up to 4 bi-directional DC motors with individual 8-bit speed selection (so, about 0.5% resolution)
- Up to 2 stepper motors (unipolar or bipolar) with single coil, double coil or interleaved stepping.
- Pull down resistors keep motors disabled during power-up
- Big terminal block connectors to easily hook up wires (18-26AWG) and power
- Arduino reset button brought up top
- 2-pin terminal block and jumper to connect external power, for separate logic/motor supplies
- Tested compatible with Arduino Mega 1280 & 2560, Diecimila, Duemilanove, and UNO
- Download the easy-to-use Arduino software library, check out the ready-to-go examples



ORDERING INFORMATION

Part Number	MMS0293-01
Device	L293D Motor Shield Module
Package	Assembled Module

APPLICATIONS

- Robotics
- Mechatronics
- Simple to medium-complexity projects

DESCRIPTION

The L293D Motor Shield is a great motor controller for Arduino. It control servos, steppers and DC motors with an Arduino.

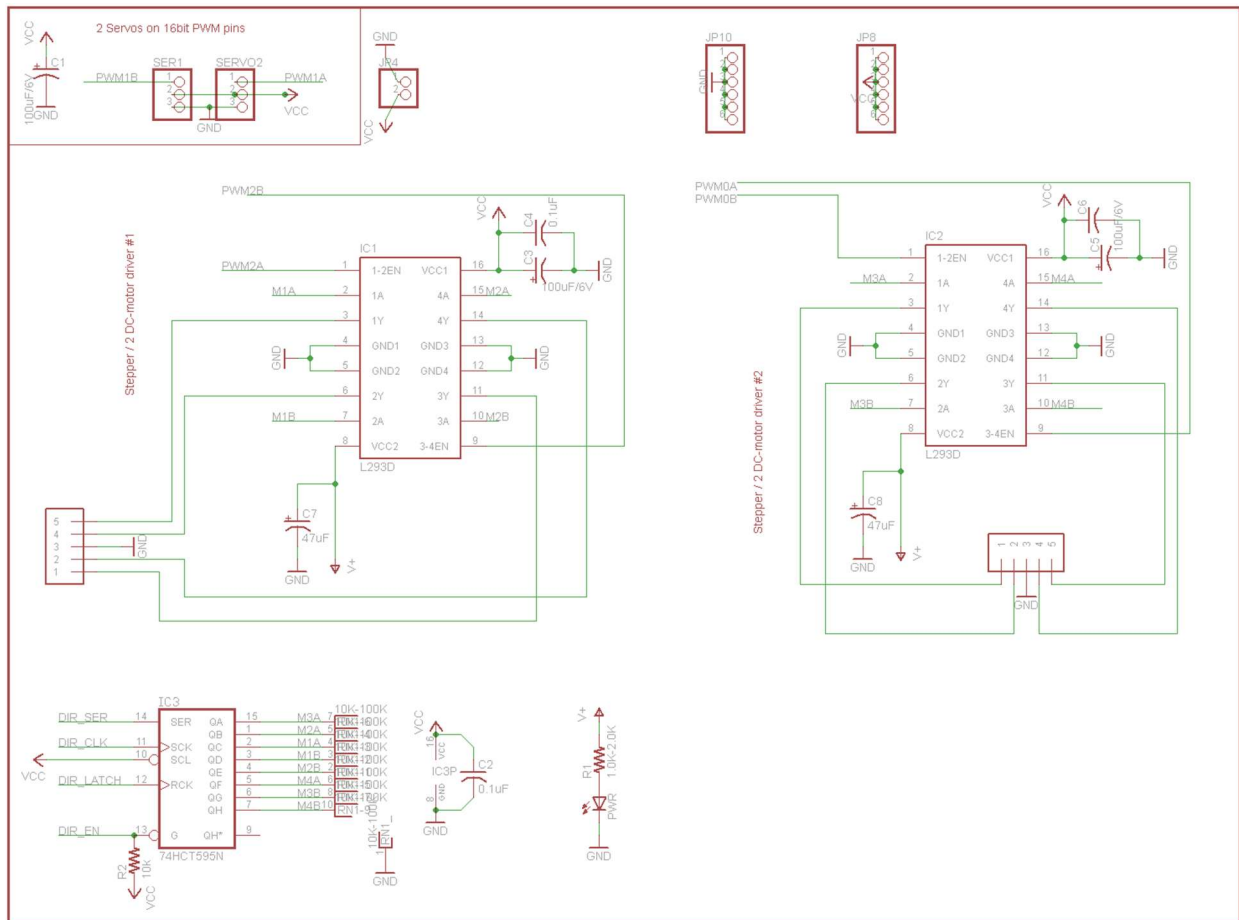
L293D Motor Shield

PARTS LIST

Name	Description	P/N	Qty.	Distributor
PCB	Printed Circuit Board	Motor Shield PCB	1	TAEJIN
IC1, IC2	L293D, Quadruple Dual H-Bridge	L293DN	2	TAEJIN
IC3	74HC595, Serial to Parallel Output Latch	74HC595N	1	TAEJIN
IC1', IC2', IC3'	16 Pin Sockets	DIP 16 pin Sockets	3	Generic
LED1	LED for Power Indicator	3mm LED, Green	1	Generic
R1	1.0K Resistor for LED1	1.0K 1/4W 5% Resistor	1	Generic
R2	10K Pulldown Resistor	10K 1/4W 5% Resistor	1	Generic
RN1	10 Pin Bussed 10K Resistor Network	10K 5% Resistor Network	1	Generic
C2, C4, C6	0.1uF Capacitor	0.1μF/50V Ceramic Capacitor	3	Generic
C1, C3, C5, C7, C8	100μF Capacitor	100μF/16V Electrolytic Capacitor	5	Generic
X1	5-Position Terminal Block (Or 1 3-Position and 2-Position)	5-Position 0.2Inch Terminals	2	Generic
X2	2-Position Terminal Block	2-Position 0.2Inch Terminals	1	Generic
RESET	Tactile Switch	6mm Tactile Switch	1	Generic
PWR	Jumper/Shunt	0.1 Inch Jumper	1	Generic
	36 Pin Male Header (1x34)	1x36 Male Header	1	Generic

L293D Motor Shield

SCHEMATICS



< Connection with Arduino Board >

L293D Motor Shield

LAYOUT



I/O PORT CONFIGURATION AND DESCRIPTION

Designator	Port Name	Port Description
+M	EXT_PWR	External Power In. (6V to 12V)
GND	GND	Ground for External Power In.
RESET	RESET	Reset Switch for Arduino Board
PWR	Power Jumper	Disconnect the jumper if external power used.
M1	M1	DC Motor #1
M2	M2	DC Motor #2
M3	M3	DC Motor #3
M4	M4	DC Motor #4
Gnd	GND	Ground for Motors
SER1	SERVO1	Servo Motor #1
SERVO2	SERVO2	Servo Motor #2
JP1	DO-D7	Digital I/O. Connect to Arduino board.
JP2	D8-D13, gnd, ARef	Digital I/O, GND, and AREF. Connect to Arduino board.
JP3	Vin, Gnd, 5V, 3V, RST	VIN, GND, 5V, 3V, RST. Connect to Arduino board.
JP4	A0-A5	Analog In. Connect to Arduino board.

MOTOR CONNECTIONS

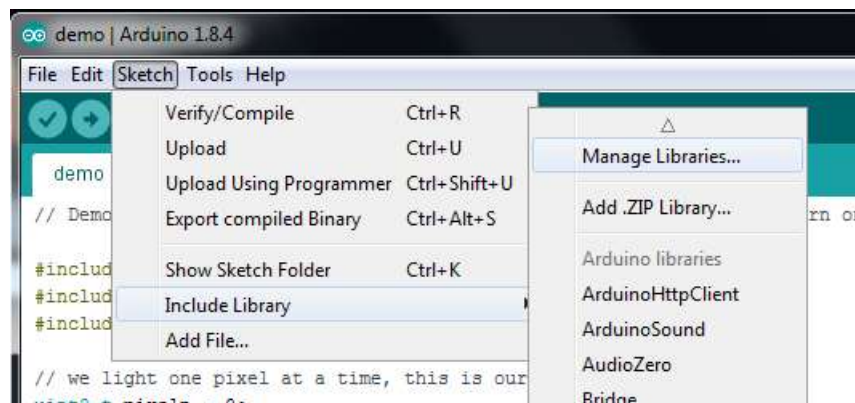
Types of Motors	Maximum Connection	Used Ports
Servo Motors	2	SER1, SERVO2
Stepper Motors	2	M1, M2, M3, M4
DC Motors	4	M1, M2, M3, M4

L293D Motor Shield

INSTALL THE ARDUINO LIBRARY

Before you can use the Motor shield, you must install the AF_Motor Arduino library - this will instruct the Arduino how to talk to the Adafruit Motor shield, and it isn't optional.

Open up the Arduino library manager:



adafruit_products_library_manager_menu.png

Search for Adafruit Motor library and install it. Make sure it is the library for the V1 motor shield.



L293D Motor Shield

POWER USAGE

POWERING YOUR DC MOTORS, VOLTAGE AND CURRENT REQUIREMENTS

Motors need a lot of energy, especially cheap motors since they're less efficient. The first important thing to figure out what voltage the motor is going to use. If you're lucky your motor came with some sort of specifications. Some small hobby motors are only intended to run at 1.5V, but it's just as common to have 6-12V motors. The motor controllers on this shield are designed to run from 4.5V to 25V. Most 1.5-3V motors will not work.

Current requirements

The second thing to figure out is how much current your motor will need. The motor driver chips that come with the kit are designed to provide up to 600 mA per motor, with 1.2A peak current. Note that once you head towards 1A you'll probably want to put a heatsink on the motor driver, otherwise you will get thermal failure, possibly burning out the chip.

Need more power?

Buy another set of L293D drivers and solder them right on top of the ones on the board (piggyback). Voila, double the current capability. You can solder 2 more chips on top before it probably isn't going to get you much benefit

You can't run motors off of a 9V battery so don't even waste your time/batteries.

Use a big Lead Acid or NiMH battery pack. It's also very much suggested that you set up two power supplies (split supply) one for the Arduino and one for the motors. 99% of 'weird motor problems' are due to noise on the power line from sharing power supplies and/or not having a powerful enough supply.

HOW TO SET UP THE ARDUINO + SHIELD FOR POWERING MOTORS

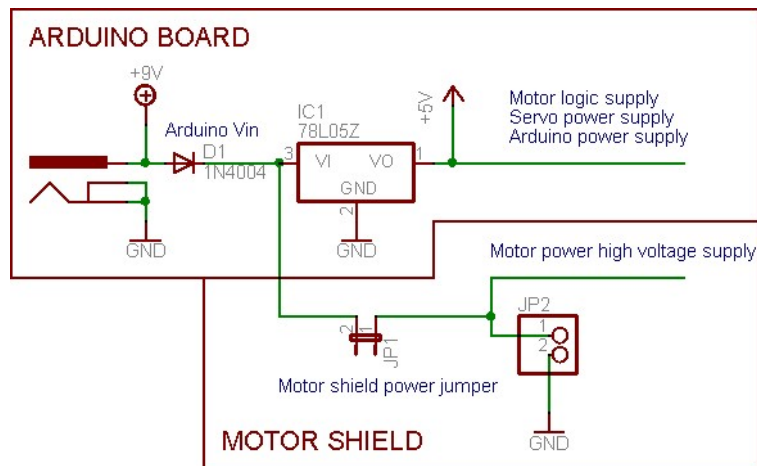
Servos are powered off of the same regulated 5V that the Arduino uses. This is OK for the small hobby servos suggested. If you want something beefier, cut the trace going to + on the servo connectors and wire up your own 5-6V supply.

The DC motors are powered off of a 'high voltage supply' and NOT the regulated 5V. Don't connect the motor power supply to the 5V line. This is a very bad idea unless you are sure you know what you're doing.

There are two places you can get your motor 'high voltage supply' from. One is the DC jack on the Arduino board and the other is the 2-terminal block on the shield that is labeled EXT_PWR. The DC Jack on the Arduino has a protection diode so you won't be able to mess things up too bad if you plug in the wrong kind of power. However the EXT_PWR terminals on the shield do not have a protection diode (for a fairly good reason). Be utterly careful not to plug it in backwards or you will destroy the motor shield and/or your Arduino.

Here's how it works:

L293D Motor Shield



If you would like to have a single DC power supply for the Arduino and motors, simply plug it into the DC jack on the Arduino or the 2-pin PWR_EXT block on the shield. Place the power jumper on the motor shield.

If you have a Diecimila Arduino, set the Arduino power source jumper to EXT.

Note that you may have problems with Arduino resets if the battery supply is not able to provide constant power, and it is not a suggested way of powering your motor project.

If you would like to have the Arduino powered off of USB and the motors powered off of a DC power supply, plug in the USB cable. Then connect the motor supply to the PWR_EXT block on the shield. Do not place the jumper on the shield. This is a suggested method of powering your motor project

(If you have a Diecimila Arduino, don't forget to set the Arduino power jumper to USB. If you have a Diecimila, you can alternately do the following: plug the DC power supply into the Arduino, and place the jumper on the motor shield.)

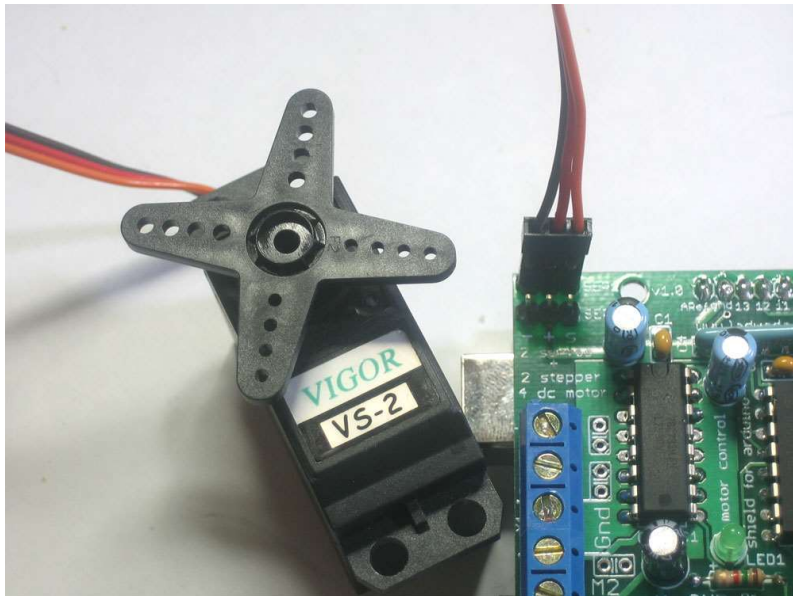
If you would like to have 2 separate DC power supplies for the Arduino and motors. Plug in the supply for the Arduino into the DC jack, and connect the motor supply to the PWR_EXT block. Make sure the jumper is removed from the motor shield.

If you have a Diecimila Arduino, set the Arduino jumper to EXT. This is a suggested method of powering your motor project.

Either way, if you want to use the DC motor/Stepper system the motor shield LED should be lit indicating good motor power.

L293D Motor Shield

USING RC SERVOS



Hobby servos are the easiest way to get going with motor control. They have a 3-pin 0.1" female header connection with +5V, ground and signal inputs. The motor shield simply brings out the 16bit PWM output lines to two 3-pin headers so that its easy to plug in and go. They can take a lot of power so a 9V battery wont last more than a few minutes.

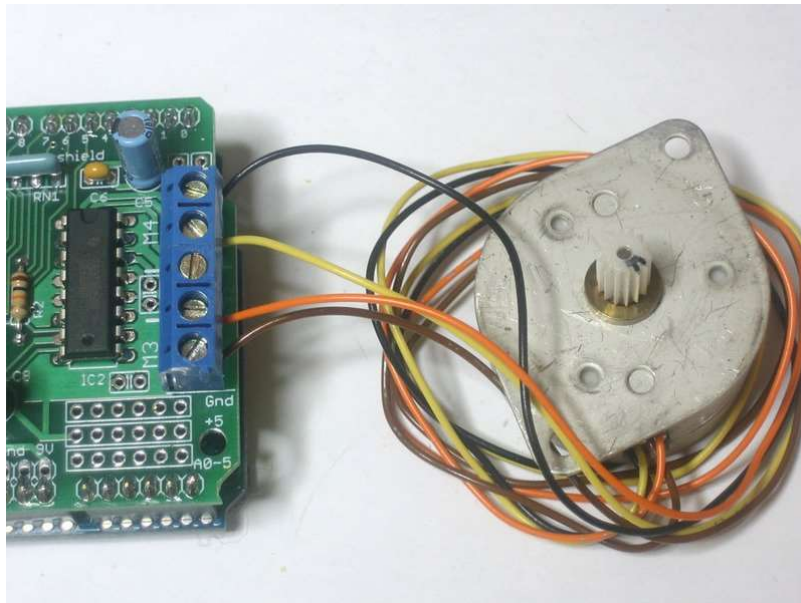
The nice thing about using the onboard PWM is that its very precise and goes about its business in the background. You can use the built in Servo library

Using the servos is easy, please read the official Arduino documentation for how to use them and see the example Servo sketches in the IDE.

Power for the Servos comes from the Arduino's on-board 5V regulator, powered directly from the USB or DC power jack on the Arduino. If you need an external supply, cut the trace right below the servo pins (on v1.2 boards) and connect a 5V or 6V DC supply directly. Using an external supply is for advanced users as you can accidentally destroy the servos by connecting a power supply incorrectly.

L293D Motor Shield

USING STEPPER MOTORS



Stepper motors are great for (semi-)precise control, perfect for many robot and CNC projects. This motor shield supports up to 2 stepper motors. The library works identically for bi-polar and uni-polar motors

For unipolar motors: to connect up the stepper, first figure out which pins connected to which coil, and which pins are the center taps. If it's a 5-wire motor then there will be 1 that is the center tap for both coils. There's plenty of tutorials online on how to reverse engineer the coils pinout. The center taps should both be connected together to the GND terminal on the motor shield output block. then coil 1 should connect to one motor port (say M1 or M3) and coil 2 should connect to the other motor port (M2 or M4).

For bipolar motors: it's just like unipolar motors except theres no 5th wire to connect to ground. The code is exactly the same.

Running a stepper is a little more intricate than running a DC motor but its still very easy

1. Make sure you `#include <AFMotor.h>`
2. Create the stepper motor object with `AF_Stepper(steps, stepper#)` to setup the motor H-bridge and latches. Steps indicates how many steps per revolution the motor has. a 7.5degree/step motor has $360/7.5 = 48$ steps. Stepper# is which port it is connected to. If you're using M1 and M2, its port 1. If you're using M3 and M4 it's port 2
3. Set the speed of the motor using `setSpeed(rpm)` where rpm is how many revolutions per minute you want the stepper to turn.
4. Then every time you want the motor to move, call the `step(#steps, direction, steptype)` procedure. #steps is how many steps you'd like it to take. direction is either FORWARD or BACKWARD and the step type is SINGLE, DOUBLE, INTERLEAVE or MICROSTEP.
"Single" means single-coil activation, "double" means 2 coils are activated at once (for higher torque) and "interleave" means that it alternates between single and double to get twice the resolution (but of course its half the speed). "Microstepping" is a method where the coils are PWM'd to create smooth motion

L293D Motor Shield

between steps. There's tons of information about the pros and cons of these different stepping methods in the resources page.

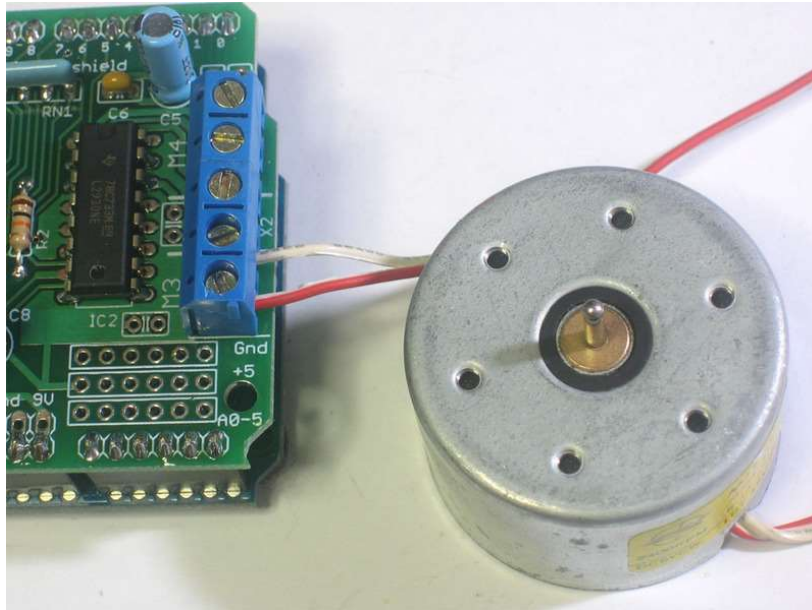
You can use whichever stepping method you want, changing it "on the fly" to as you may want minimum power, more torque, or more precision.

5. By default, the motor will 'hold' the position after its done stepping. If you want to release all the coils, so that it can spin freely, call `release()`.
6. The stepping commands are 'blocking' and will return once the steps have finished.

Because the stepping commands 'block' - you have to instruct the Stepper motors each time you want them to move. If you want to have more of a 'background task' stepper control, check out `AccelStepper` library (install similarly to how you did with `AFMotor`) which has some examples for controlling two steppers simultaneously with varying acceleration.

L293D Motor Shield

USING DC MOTORS



DC motors are used for all sort of robotic projects.

The motor shield can drive up to 4 DC motors bi-directionally. That means they can be driven forwards and backwards. The speed can also be varied at 0.5% increments using the high-quality built in PWM. This means the speed is very smooth and won't vary.

Note that the H-bridge chip is not meant for driving loads over 0.6A or that peak over 1.2A so this is for small motors. Check the datasheet for information about the motor to verify it's OK.

To connect a motor, simply solder two wires to the terminals and then connect them to either the M1, M2, M3, or M4. Then follow these steps in your sketch

1. Make sure you `#include <AFMotor.h>`
2. Create the `AF_DCMotor` object with `AF_DCMotor(motor#, frequency)`, to setup the motor H-bridge and latches. The constructor takes two arguments. The first is which port the motor is connected to, 1, 2, 3 or 4. frequency is how fast the speed controlling signal is. For motors 1 and 2 you can choose `MOTOR12_64KHZ`, `MOTOR12_8KHZ`, `MOTOR12_2KHZ`, or `MOTOR12_1KHZ`. A high speed like 64KHz won't be audible but a low speed like 1KHz will use less power. Motors 3 & 4 are only possible to run at 1KHz and will ignore any setting given.
3. Then you can set the speed of the motor using `setSpeed(speed)` where the speed ranges from 0 (stopped) to 255 (full speed). You can set the speed whenever you want.
4. To run the motor, call `run(direction)` where direction is `FORWARD`, `BACKWARD` or `RELEASE`. Of course, the Arduino doesn't actually know if the motor is 'forward' or 'backward', so if you want to change which way it thinks is forward, simply swap the two wires from the motor to the shield.

L293D Motor Shield

REVISION NOTICE

The description in this datasheet is subject to change without any notice to describe its electrical characteristics properly.