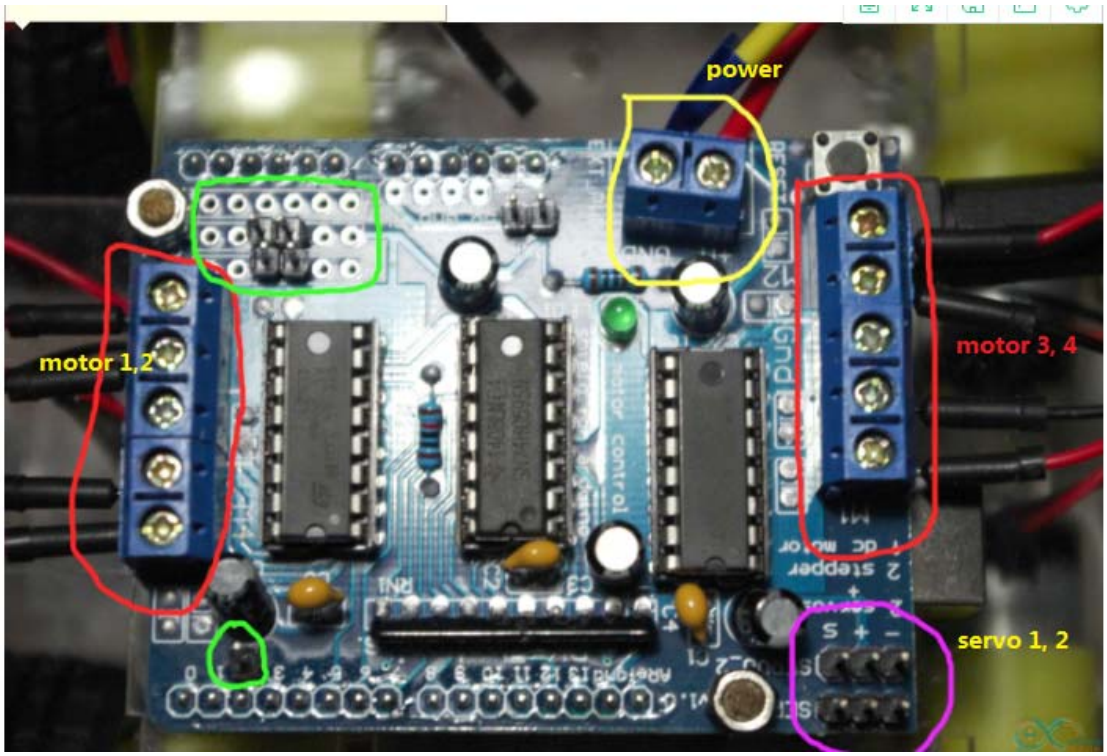
Driver shield for Arduino

Introduction

Arduino is a good introduction to electronic production, by using the motor expansion board, it can be a good robot development platform. Here is a full-featured motor expansion board that drives a variety of simple to slightly more complex projects. This is a commonly used DC motor drive module, using the L293D chip small current DC motor drive chip. The pins are made to be compatible with Arduino and also facilitate enthusiasts to conduct the rapid Arduino-based development.

Features

- 2pcs 5V server motor ports connected to Arduino's high-resolution high-precision timer - no jitter!
- Up to 4 bidirectional DC motors each with an 8-bit speed selection (approximately 0.5% resolution)
- Up to 2 stepper motors (unipolar or bipolar) single coil, dual coil, staggered or microstep.
- 4 pcs H-Bridges: L293D chip provides .0.6A (peak 1.2A) current per bridge and is protected by thermal power off, 4.5V to 36V.
- The pull-down resistor ensures that the motor remains stopped when powered on.
- Large terminal terminals make wiring easier (10 - 22AWG) and power supply.
- With Arduino reset button.
- 2 large terminal external power terminals guarantee logic and motor drive power separation.
- Compatible with Mega, Diecimila, and Duemilanove.
- All accessories are available from the Explorer's online shop.
- Download the easy-to-use Arduino software library for quick development.
- Support to drive 4 channel DC motor; or drive 2 channel stepper motor.



User Manual

Reminder

- Before upload the code to Arduino, please copy the AFMotor.h to the INSTALLED Arduino libraries, e.g., D:\Program Files\arduino-0021\libraries
- Then, re-open the Arduino IDE, and one can use the code.
- When the working current is up to 1A, must stick heat-sinks for the chip.

Working voltage and current for motor

Motors require a lot of energy, especially cheap ones because they are inefficient. The most important thing is to know the voltage that the motor will use. If you are lucky, your motor may have some parameter specifications. Some small amateur motors only operate at 1.5V, but they are often used as 6V or 12V motors. The motor drive is designed to run from 4.5V to 36V (in theory, anyway, it should be reduced to about 2.5V).

Current requirements: The second thing is to find out how much current your motor will have. The driver chip matched with the driver board is designed to provide up to 600 mA current and 1.2A peak current for each motor. When your current approaches 1A, you may want to add a heat sink to the driver chip. Otherwise, it will be thermal failure and may burn down the driver chip.

About the use of SN754410: Some people use SN754410 motor driver chips because of their pin compatibility, have output diodes and can provide the current of each motor 1A, the peak current of 2A. After carefully reviewing the data manual and discussing with their technical support engineers, it seems that this output diode is only used as ESD protection if you use it as kickback-protection is a Hack and performance is not guaranteed. For this reason, instead of SN754410, the kit uses L293D with built-in recoil protection diodes. If you're willing to take the risk and need more current, don't hesitate to switch to SN754410.

Need more power? Buy another L293D driver and weld it back to back with a driver chip. Look! Double the current capacity, you can weld two or more till it doesn't give you more benefits.

You can't drive a motor with a 9V battery, so don't waste your time and batteries! Use a large lead-acid battery or nickel-hydrogen battery pack. It is also highly recommended that you use two separate power sources, one for Arduino and the other for motor drive. 99% of the strange problems are caused by noise from a common power supply line.

And/or the power supply does not have enough power!

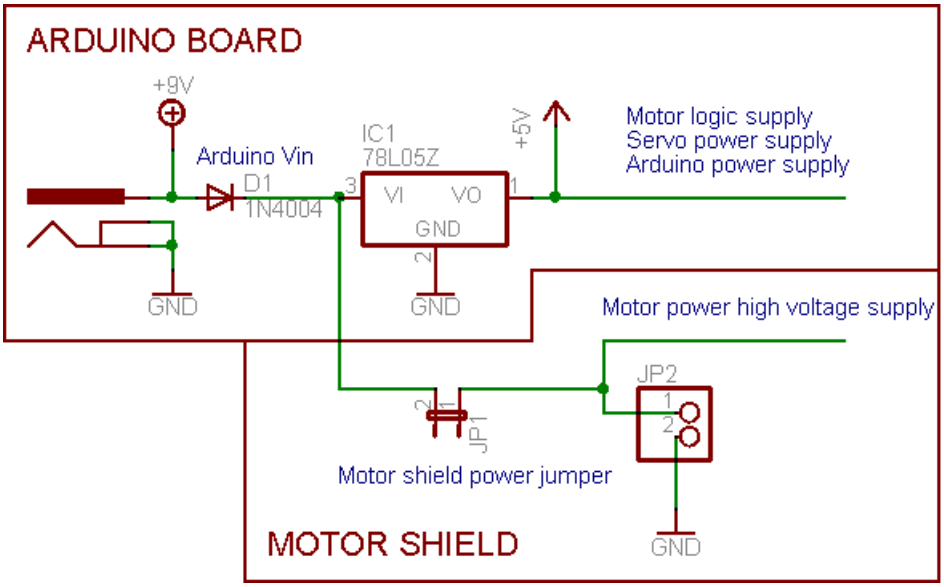
How to set up Arduino + Shield drive board to supply power for motor?

The steering gear is powered by Arduino's conventional 5V. It doesn't matter for a small amateur steering gear. If you use more power, cut the line connected to + and connect it to your own steering power supply!

The DC motor is powered by a high voltage power supply, which is not a conventional 5V power supply. Do not connect motor power to 5V wire. It's a very, very bad idea unless you know what you're doing!

There are two ways to power your motor at high voltage. One is the DC socket on the Arduino board and the other is the 2-wire terminal on the motor drive board marked EXT_PWR. The DC socket on the Arduino board has a protective diode so you won't mess things up if you get the wrong power supply. However, the EXT_PWR terminal on the drive board does not protect the diode (for a good reason). So be very careful not to take the opposite or you may burn your drive board or Arduino!

Working principle



If you only want to use a separate DC power supply to power Arduino and the motor, simply plug the power into Arduino's DC socket or the 2-wire terminal EXT_PWR on the drive board. Then the power supply on the drive board jumps in and plugs in.

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

Please note that you may encounter an Arduino reset if the power supply is insufficient, so the power supply method for this motor project is not recommended to you.

If you want Arduino to be powered by USB, the motor is powered by another DC source. Insert a USB adapter. Then connect the power to the PWR_EXT terminal on the drive board. Be careful not to jump on the drive board. It is recommended that this method be used to power your motor project.

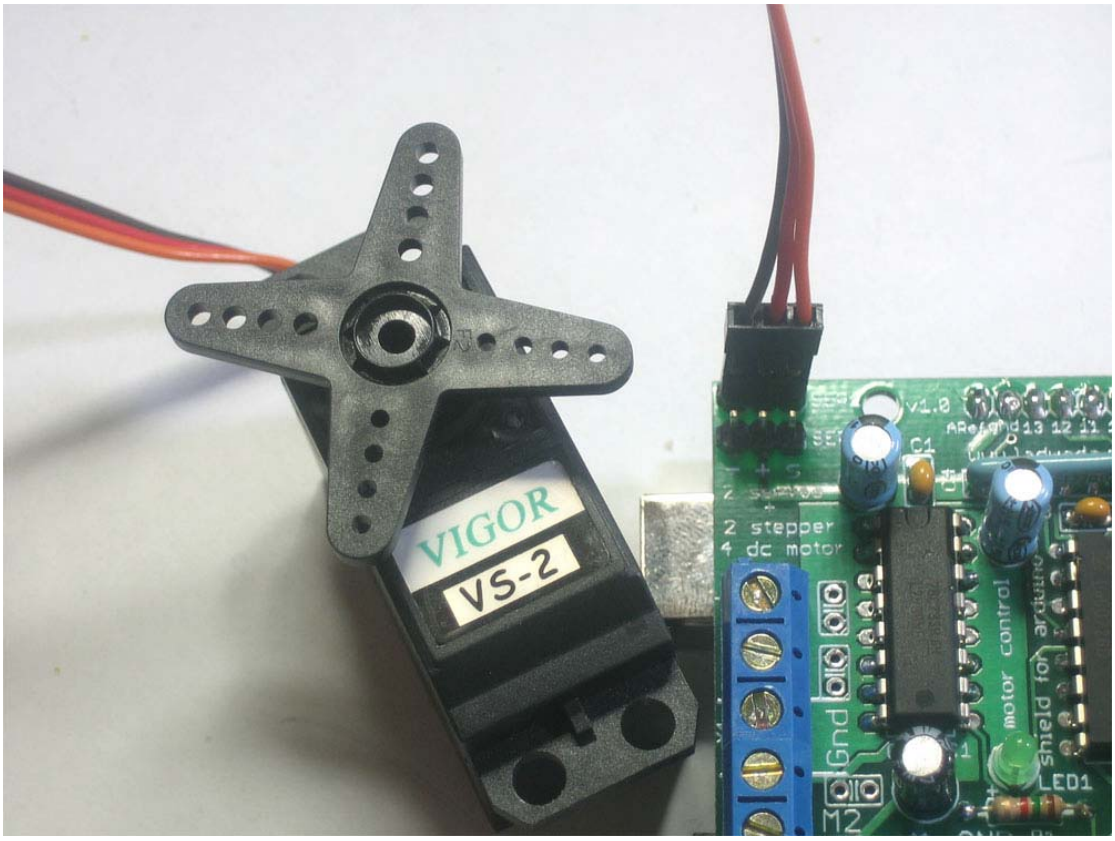
(If you use Diecimila Arduino, don't forget to set the Arduino power supply to jump to USB. If you use Diecimila Arduino, you can plug in DC power to Arduino and then jump the power supply on the drive board.

If you want to use two separate DC power supplies to power Arduino and the motor, insert the power into Arduino's DC socket and connect the power supply to the PWR_EXT terminal of the drive board. Make sure to remove the jump on the motor drive board.

If you use Diecimila Arduino, set Arduino to jump to EXT. It is recommended that this method be used to power your motor project.

That's all right. If you want to use a DC motor/stepper motor system, the LED on the motor drive board should be brightened to indicate a good motor power supply.

Servos

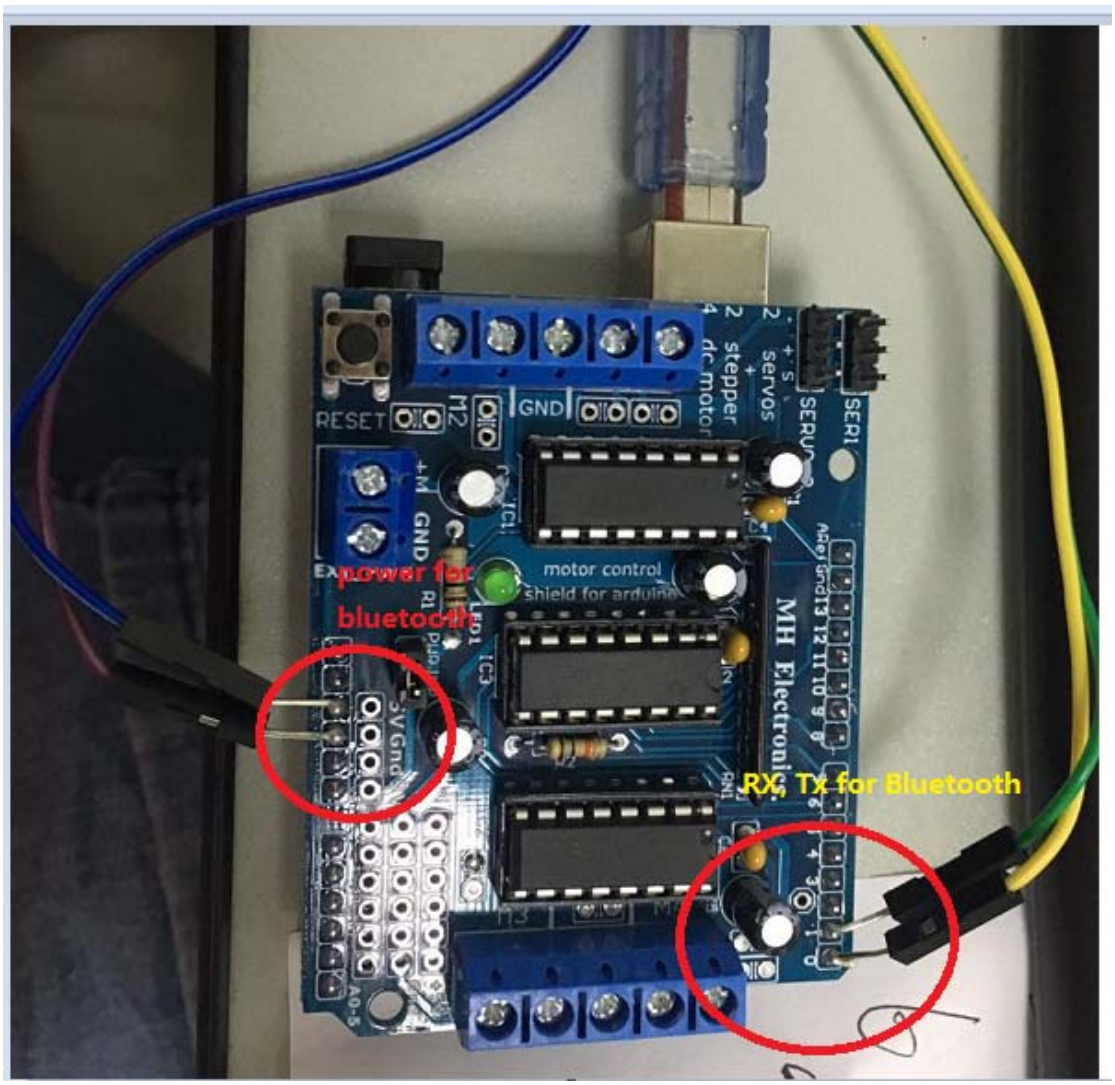


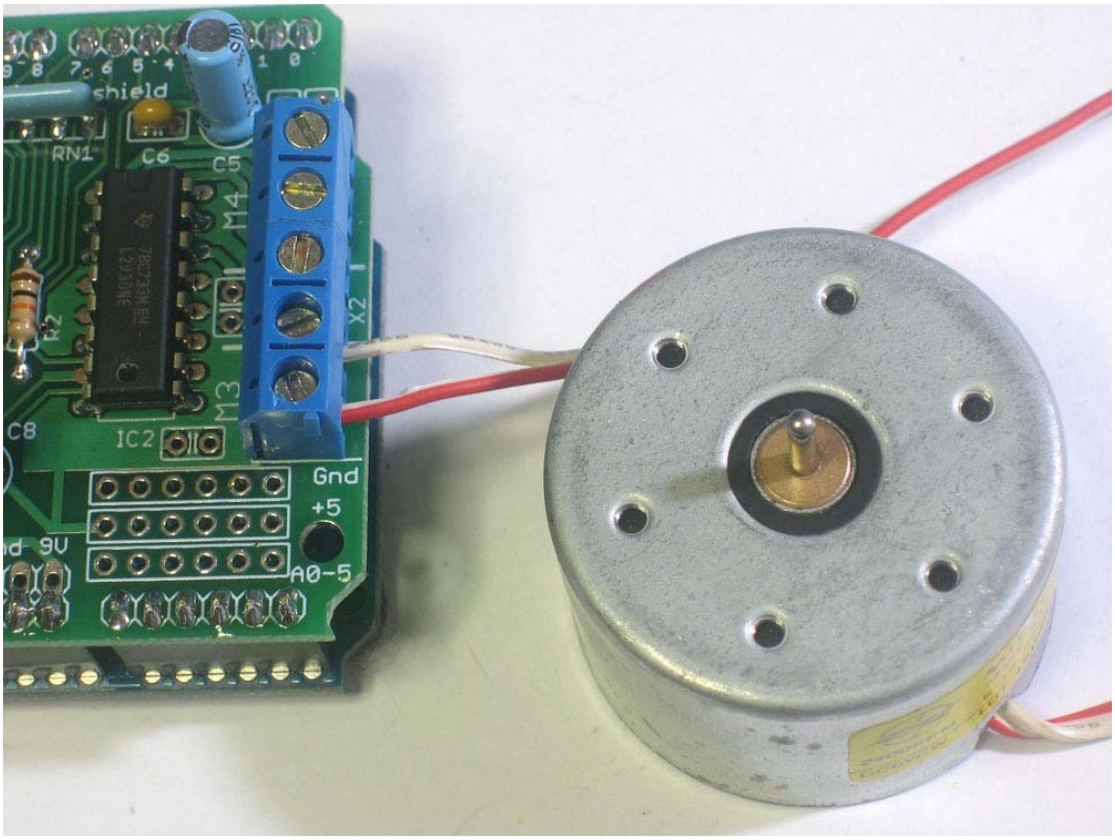
Amateur steering gear

Amateur steering gear is the easiest to control. They have a three-wire interface of 5V, GND, and signal input. The motor drive simply takes out 16-bit PWM output into 2 rows and 3-terminal plugs, so it can be easily plugged in and out. They can be very power-intensive, so a 9V battery won't last for more than a few minutes!

The advantage of using on-board PWM is that it is very accurate and can process logic in the background. You can use Arduino's Servo library to control the steering gear.

It's very simple to use the steering gear. See the official Aduino documentation for examples of heat and use of them and related steering gear programs.


power for
bluetooth
RX, Tx for Bluetooth
motor control
shield for arduino
MH Electronic
RESET
GND
stepper
dc motor
Servos
SERI1



DC motor

DC motors are used in all types of robotic projects. The motor drive board can drive up to four motors in two directions. That means they can turn it upside down and upside down. Using high quality built-in PWM speed can also change by 0.5%, which means that the speed is very stable and will not change!

Note that H-bridge chip is not really able to load more than 0.6A or peak value more than 1.2A, so it is more suitable for low-power motors. Refer to the data manual about motor parts to confirm the suitability.

Connect the motor simply by connecting two leads to the terminal M1, M2, M3, or M4. Then follow these steps in your program:

Ensure that include <MSMotorShield.h> header file

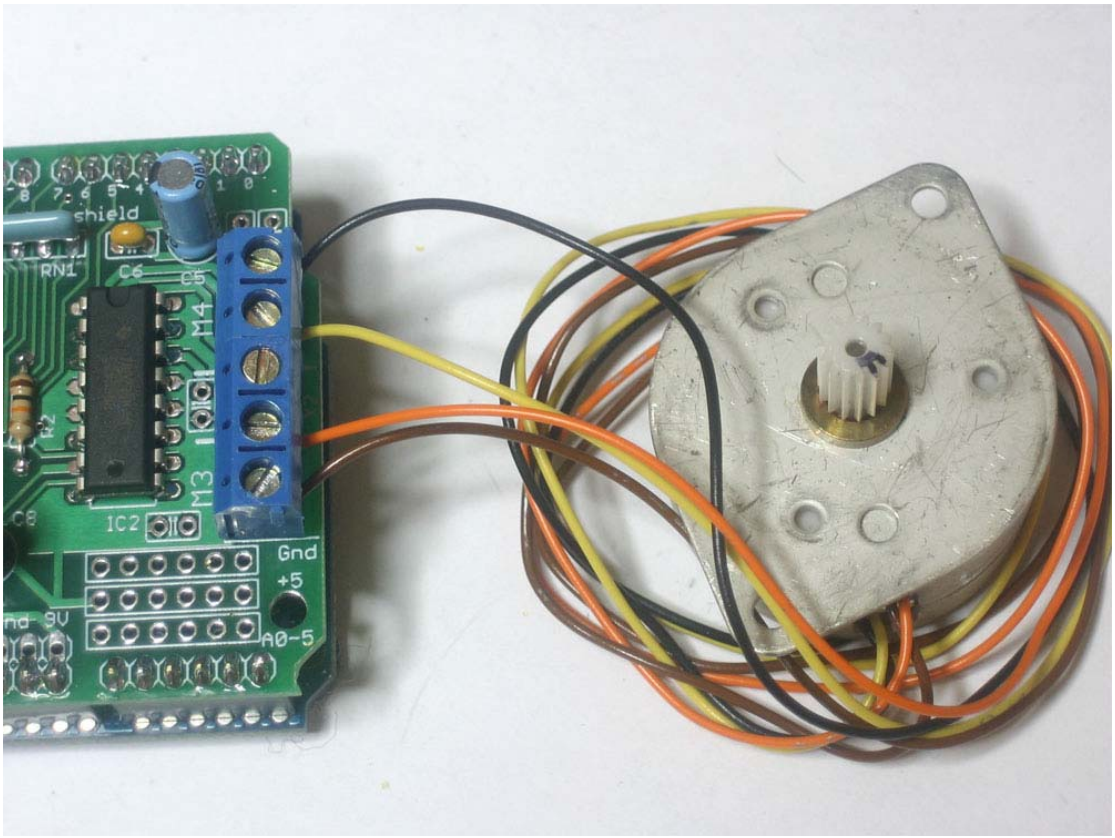
Using DCMotor (motor#, frequency) to create DCMotor object, the constructor takes even two parameters. The first is the port 1, 2, 3, 4 of the motor to be connected. Frequency is the value of the speed control signal. For motors 1 and 2, you can choose MOTOR 12_64KHZ, MOTOR 12_8KHZ, MOTOR 12_2KHZ, or MOTOR 12_1KHZ. A high frequency like 64KHz will not produce sound, but a low frequency like 1KHz will save more power. Motor 3 and 4 can only operate at 1KHz frequency, and will ignore any given frequency setting.

Then you can set the speed of the motor through the setSpeed (speed) function, which ranges from 0 (stop) to 255 (full speed). You can set any speed you want.

Turn on the motor by calling run (direction) direction is FORWARD, BACKWARD or RELEASE. Of course, Arduino doesn't really know whether the motor is going forward or backward. So if you want to change the direction you think you're turning, you just need to switch the motor lead.

```
#include <MSMotorShield.h>

DCMotor motor(2, MOTOR12_64KHZ); // create motor #2, 64KHz pwm

void setup() {
Serial.begin(9600); // set up Serial library at 9600 bps
Serial.println("Motor test!");

motor.setSpeed(200); // set the speed to 200/255
}

void loop() {
Serial.print("tick");

motor.run(FORWARD); // turn it on going forward
delay(1000);

Serial.print("tock");
motor.run(BACKWARD); // the other way
delay(1000);

Serial.print("tack");
motor.run(RELEASE); // stopped
delay(1000);
}
```

Stepper Motor



Bipolar four-wire stepping motor

Stepping motor control of half precision is very good. It is also ideal for many robots and CNC projects. This motor drive supports up to two stepper motors. The library can support two-phase sum very well.

To connect a unipolar stepping motor, first of all, it is necessary to know which pins are connected to which windings (coils) and which pins are connected to the intermediate windings (coils). If it is a 5-wire stepper motor, there will be a middle tap of two sets of coils. There are many tutorials on how to reverse engineer coil pins. The middle tap should be connected to the output GND terminal of the motor drive board, and then coil 1 should be connected to a motor port (M1 or M3) and coil 2 should be connected to other motor ports (M2 or M4).

For bipolar stepper motors, it is the same as unipolar stepper motors except that there is no fifth lead connected to a low one. The code is the same.

Driving stepper motors is slightly more complicated than DC motors, but it's still simple here.

Ensure that include <MSMotorShield.h> header file

Using MS_ Stepper (steps, stepper_ #) to create MS_ Stepper object, the constructor takes even two parameters. The first is how many steps a stepper motor needs per turn, a 7.5 degree/step motor has 360/7.5=48 steps. Which port is stepper # connected to? If you use M1 and M 2, it's port 1, if you use M3 and M4, it's port 2.

SetSpeed (rpm) function is used to set the number of turns per minute that you want the stepper motor to rotate.

Then every time you want the stepper motor to turn, you just call the step (steps, direction, steptype) function. # Steps is how many steps you want the stepper motor to take. Direction is FORWARD or BACKWARD, steptype is SINGLE, DOUBLE, INTERLEAVE or MICROSTEP. "Single" refers to single coil enablement, "double" refers to double coil enablement (for the need

for strong moments), "interleave" refers to alternating between single and double to obtain double resolution (of course speed halved). "Microstepping" is a way to create smooth motion between steps using PWM. There are a lot of articles about the pros and cons of different stepping. You can use the step method you want. You can change it in operation as you want to minimize power consumption, maximize torque or higher accuracy.
The default motor will remain in its stop position. If you want to release all the coils, you need to call the realease () function. The step command will return immediately after the action is completed. It would be cool if there were some edition libraries that could be used as background steps.)

```
#include <MSMotorShield.h>

MS_Stepper motor(48, 2);

void setup() {
Serial.begin(9600); // set up Serial library at 9600 bps
Serial.println("Stepper test!");

motor.setSpeed(10); // 10 rpm

motor.step(100, FORWARD, SINGLE);
motor.release();
delay(1000);
}

void loop() {
motor.step(100, FORWARD, SINGLE);
motor.step(100, BACKWARD, SINGLE);

motor.step(100, FORWARD, DOUBLE);
motor.step(100, BACKWARD, DOUBLE);

motor.step(100, FORWARD, INTERLEAVE);
motor.step(100, BACKWARD, INTERLEAVE);

motor.step(100, FORWARD, MICROSTEP);
motor.step(100, BACKWARD, MICROSTEP);
}
```

If you want the stepper motor to step at once, you will need to write the following code:

If you want two stepper motors to step at once you'll need to write something like this:

```
void doublestep (int steps, int direction, int style) {
while (steps--) {
```

```
motor1.step(1, direction, style);
motor2.step(1, direction, style);
}
}
```

Common problem

How many motors can I connect?

Up to four DC motors or two stepping motors and two steering motors (servo motors) can be connected at the same time.

There are the following options available:

1. Connect 4 DC motors and 2 steering gears
2. Connect 2 DC motors, 1 stepper motor and 2 steering gear
3. Connect 2 stepper motors and 2 steering gear

Which ports are connected to DC/stepper motors?

DC or stepping motors are not directly connected to Arduino. They are connected to the 74HC595 latch and then controlled by Arduino. You can't control the motor directly. You have to use the motor drive board library to drive the motor.

Which ports of Arduino are occupied by this driver?

All six analog ports are not occupied. They can also be used as digital ports (pin14-19). These ports are generally used to connect sensors, etc.

Digital ports 2 and 13 are not occupied.

The first port is occupied

Digital Port 11: DC Motor #1/Step #1 (PWM)

Digital Port 3: DC Motor #2/Step #1 (PWM)

Digital Port 5: DC Motor #3/Step #2 (PWM)

Digital Port 6: DC Motor #4/Step #2 (PWM)

Digital ports 4, 7, 8 and 12 are occupied when using any DC or stepper motor. The DC or stepper motor is controlled by 74HC595 (series-parallel) latch.

If the following ports of the steering gear are used, they will be occupied:

Digital Port 9: Actuator Control #1

Digital Port 10: Actuator Control #2

How to connect unused ports?

Analog ports (0-5 can also be used as digital ports 14-19) are drawn in the lower right corner and you can use them directly.

Common abnormalities:

The following error occurred when I ran the sample code:

"Error: MSMotorShield. h: No such file or directory..."

Make sure that you install the MSMotorShield library file, and extract the attached library file into the libraries directory for Arduino installation purposes.

For example: D: Program Files arduino-0021 Libraries

Then reopen the Arduino IDE to use the sample code for reference.

How to install libraries into Arduino IDE?

Install Arduino Compatible Full-Function Driver Board Library into Arduino file as attached

(registration is required to see the attachment)

Just extract the library files in the attachment into the libraries directory for Arduino installation purposes

For example: D: Program Files arduino-0021 Libraries

Then reopen the Arduino IDE to use the sample code for reference.

If there are any suggestions on how to improve this introductory guide? The concept is not explained clearly? Need more sample code?