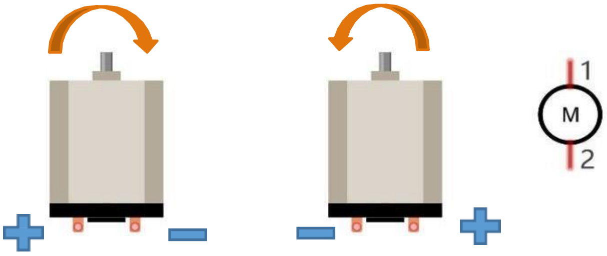
Lesson 2-Testing TT Motor

Knowledge of the TT Motor

When motor is connected to the power supply, it will rotate in one direction. Reverse the polarity of power supply, the motor will rotate in the opposite direction.

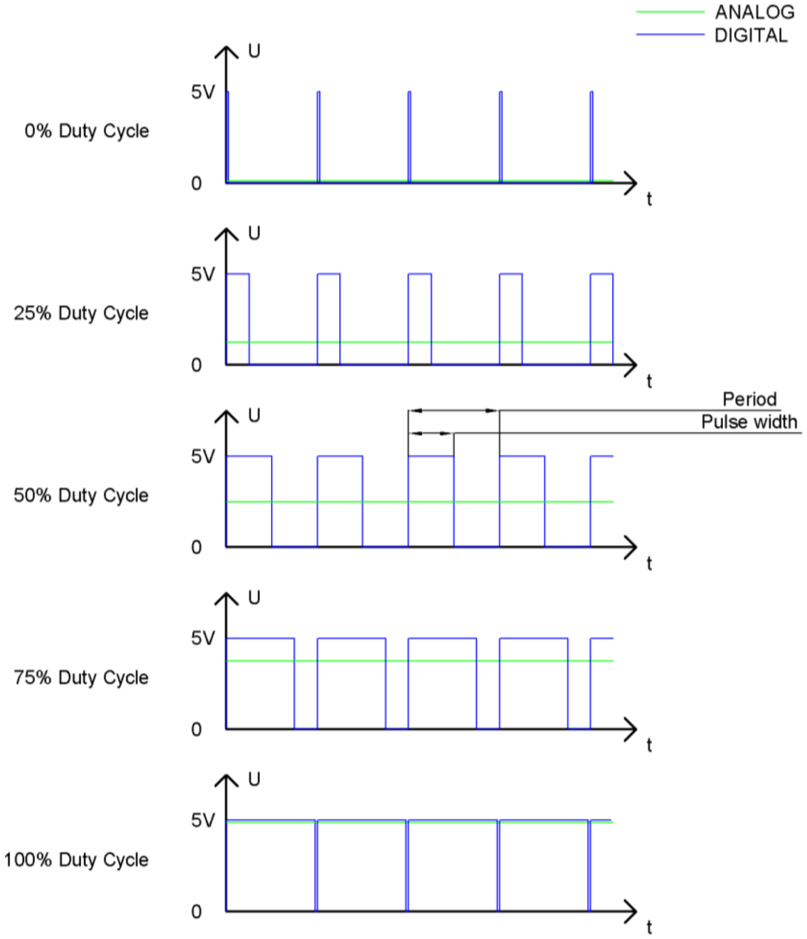
And the speed of motor depends on the voltage between two ends. The larger the voltage, the lager the speed.



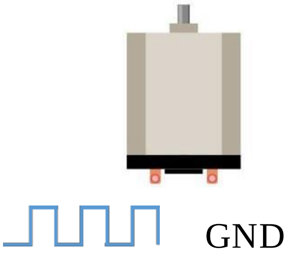
PWM

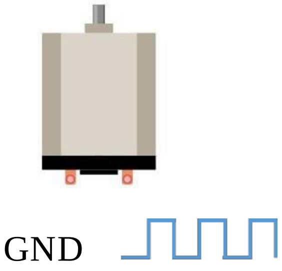
PWM, Pulse Width Modulation, uses digital pins to send certain frequencies of square waves, that is, the output of high levels and low levels, which alternately last for a while. The total time for each set of high levels and low levels is generally fixed, which is called the period (the reciprocal of the period is frequency). The time of high level outputs are generally called “pulse width”, and the duty cycle is the percentage of the ratio of pulse duration, or pulse width (PW) to the total period (T) of the waveform.

The longer the output of high levels last, the larger the duty cycle and the higher the corresponding voltage in analog signal will be. The following figures show how the analogs signal voltage vary between 0V-5V (high level is 5V) corresponding to the pulse width 0%-100%:



The longer the PWM duty cycle is, the higher the output power will be. Now that we understand this relationship, we can use PWM to control the brightness of an LED or the speed of DC motor and so on.

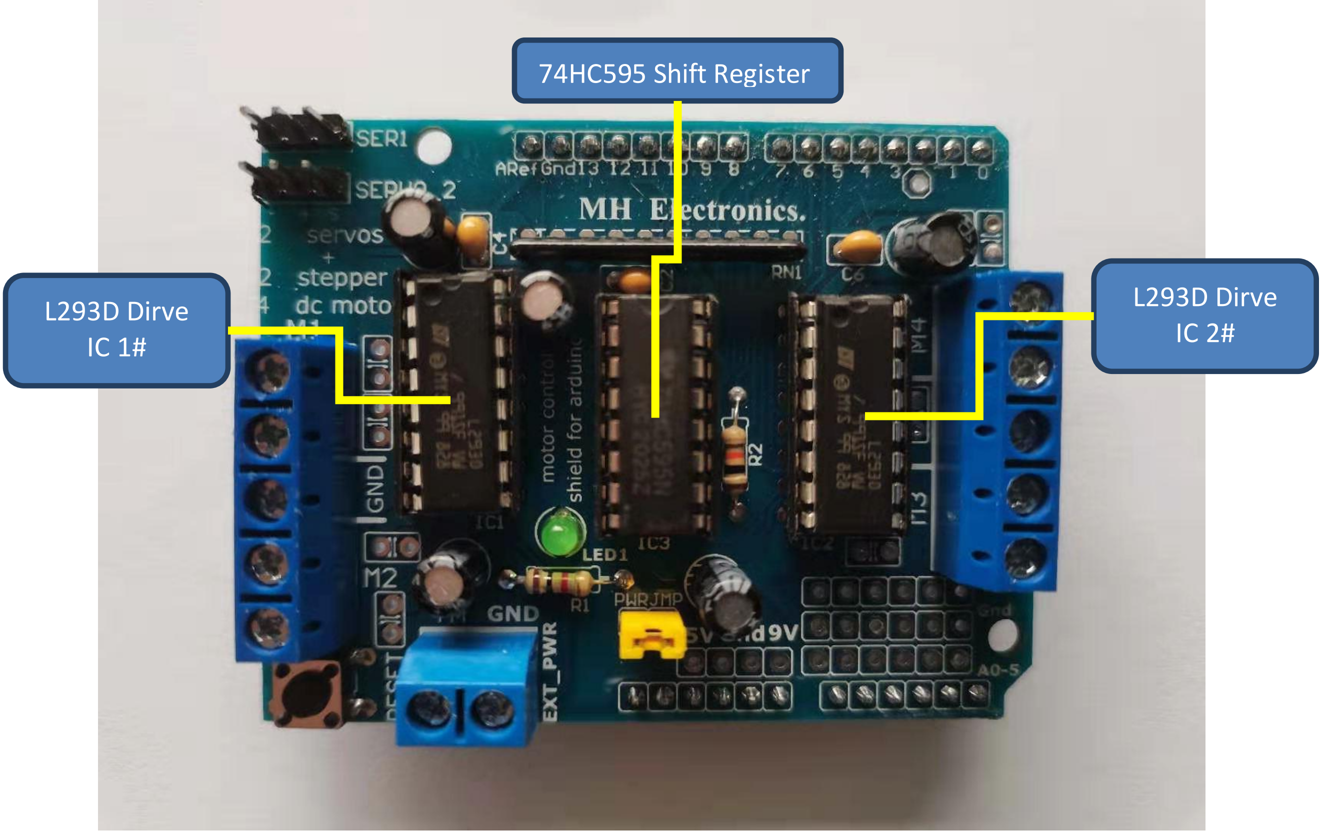




In this course, a L293D motor control shield will be used with UNO board, which can control the rotation of 4 DC motors at the same time.

L293D motor control shield introduce

The L293D motor control shield is mainly composed of 2 L293D motor driver chips and a 74HC595 shift register. Its structure is as follows



Functions of L293D motor control shield

- L293D is a dual-channel H-bridge motor driver that can drive a pair of DC motors or a single stepper motor.
- Since the shield has two L293D motor driver chipsets, which means it can drive up to four DC motors individually, it is ideal for building a four-wheeled robotic platform.
- The shield provides a total of 4 H-bridges, each of which can supply up to 0.6A to the motor.
- The shield also comes with a 74HC595 shift register that extends the 4 digital pins of the UNO board to the 8 direction control pins of the two L293D chips.
- The shield has an array of pull-down resistors to keep the motors off during power up.
- Onboard LEDs indicate that motor power is OK. If it is not lit, the motor will not run.
- The RESET button is the reset button of the UNO board (when the L293D shield is installed on the UNO board).

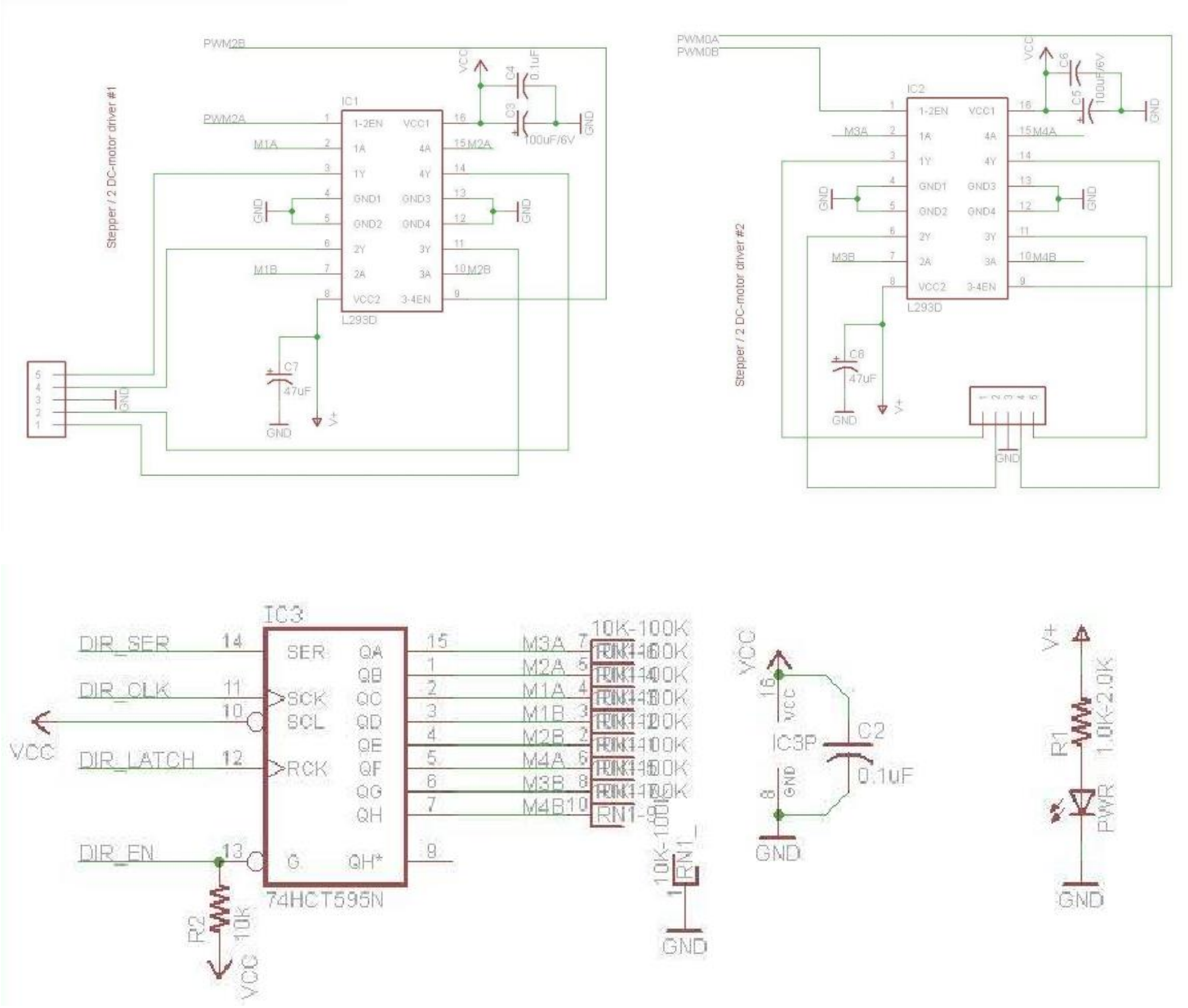
L293D motor control shield principle

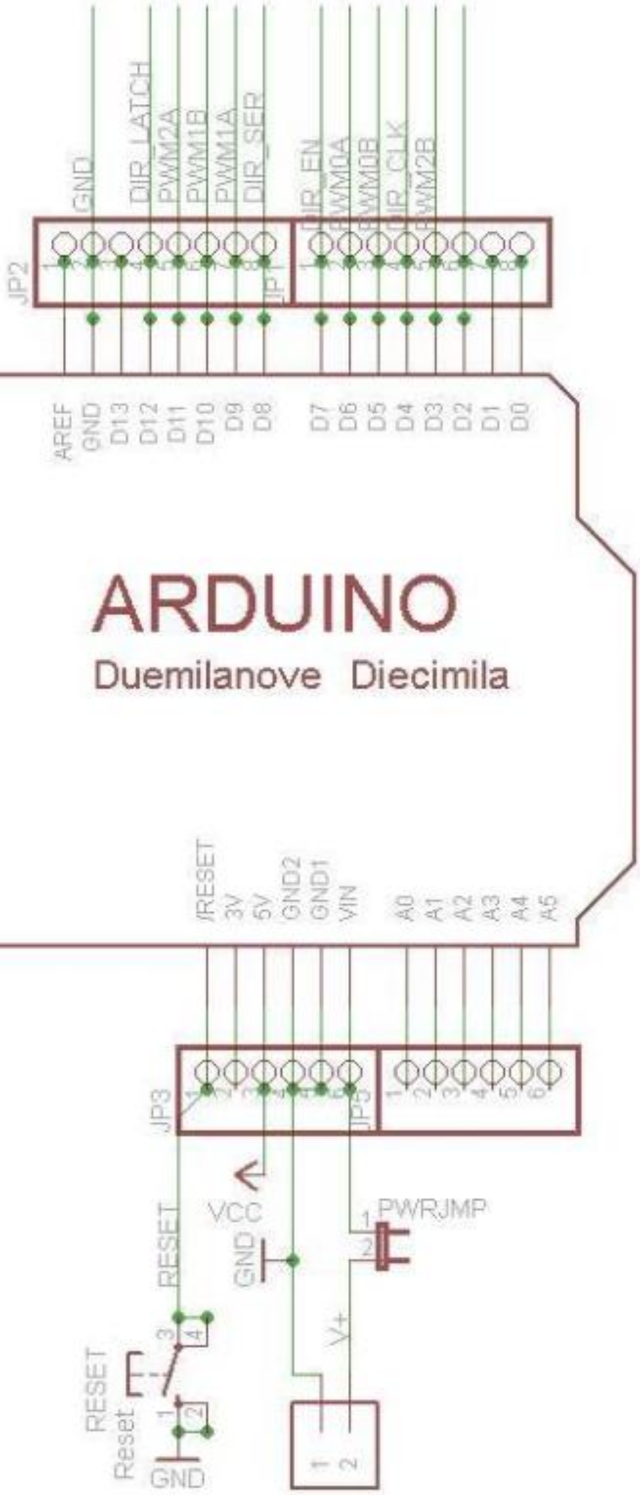
Using L293D to drive the motor, there is a 74HC595 chip in the middle that converts serial signals into parallel signals. This module is designed for Arduino, which has fewer I/O ports and requires 12 pins to control 4 DC motors. Using 74HC595 can reduce the use of 4 pins

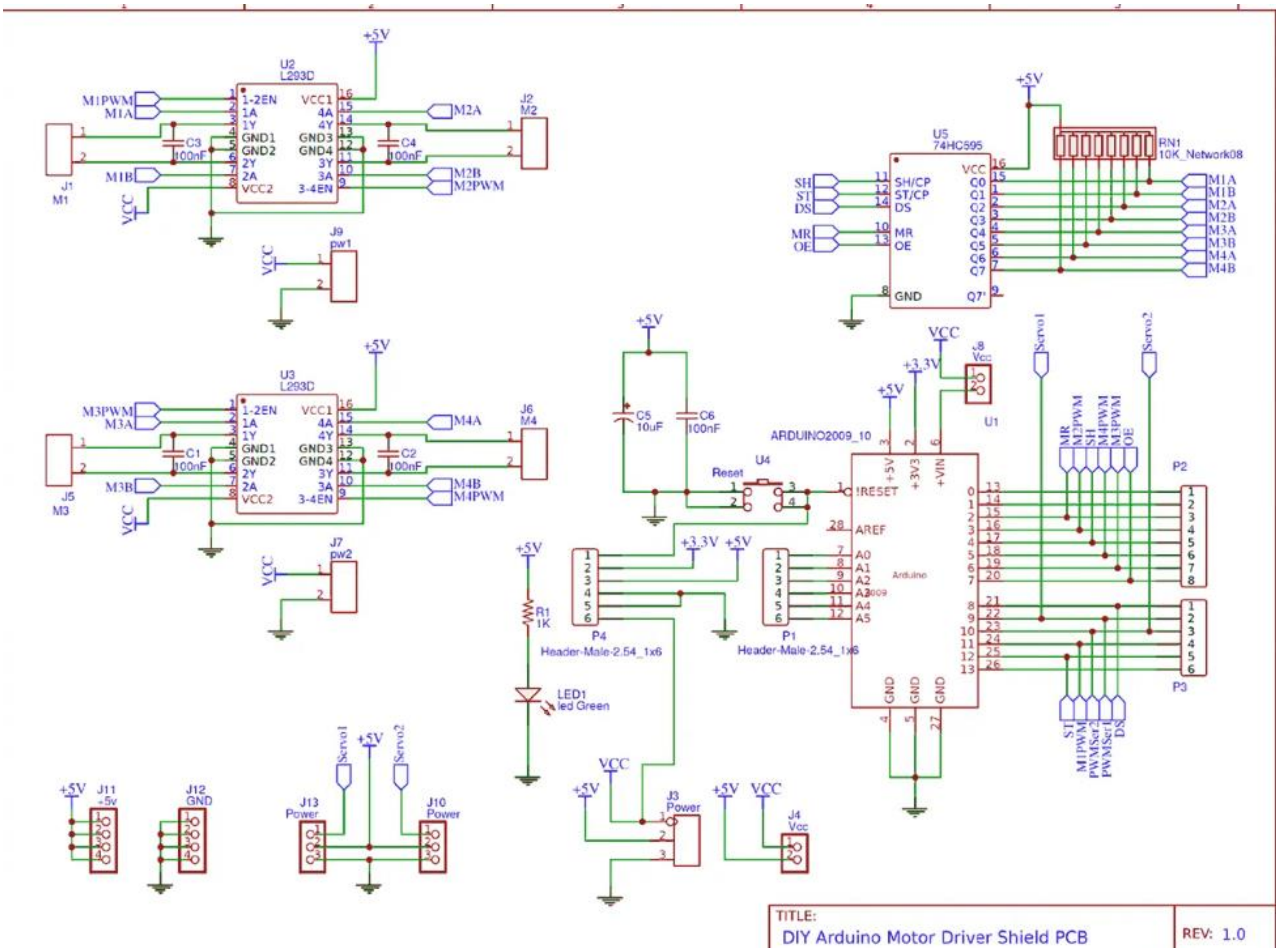
74HC595

74HC595 is an 8-bit serial input and parallel output displacement buffer: parallel output is a three state output. On the rising edge of SCK, serial data is input from SDL to the internal 8-bit displacement buffer and output from Q7 ', while parallel output stores the data from the 8-bit displacement buffer to the 8-bit parallel output buffer on the rising edge of LCK. When the control signal of the serial data input end OE is low enabled, the output value of the parallel output end is equal to the value stored in the parallel output buffer. Simply put, first set pin 7 of the module to 0, and then pin 4 of the module (clock end for data input of 74HC595 chip) receives a rising edge. Move the 8-bit data in the chip to the left by one bit, leaving the low bit to write the 0 or 1 signal of pin 8 (serial data input of 74HC595 chip) to the low bit. After writing eight times, write the 8-bit signal that controls four motors to the 74HC595 chip (M3M4M3M2M1M2M4), Then, by giving a rising edge to pin 12, the data in the chip is output to the pins of the chip (Q0~Q7)

L293D motor control shield schematic diagram




GND
DIR_LATCH
PWM2A
PWM1B
PWM1A
DIR_SER
DIR_EN
PWM0A
PWM0B
DIR_CLK
PWM2B
JP2
JP1
AREF
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
ARDUINO
Duemilanove Diecimila
/RESET
3V
5V
GND2
GND1
VIN
A0
A1
A2
A3
A4
A5
JP3
JP3
RESET
VCC
GND
PWRJMP
V+
RESET
Reset
GND



L293D motor control shield occupies Arduino pin corresponding table

L293D motor control shield	Arduino UNOR3(for example)
L293D Driver IC1#1-2EN	**D11**
L293D Driver IC1#3-4EN	**D3**
L293D Driver IC2#1-2EN	**D5**
L293D Driver IC2#3-4EN	**D6**
74HC595 DIR-SER	**D8**
74HC595 DIR-CLK	**D4**
74HC595 DIR-LATCH	**D12**
74HC595 DIR-EN	**D7**
SER1	**D10**
SERVO_2	**D9**
N/A	**D2**
N/A	**D13**
N/A	**A0**
N/A	**A1**
N/A	**A2**
N/A	**A3**
N/A	**A4**
N/A	**A5**

Power the motors through the control shield

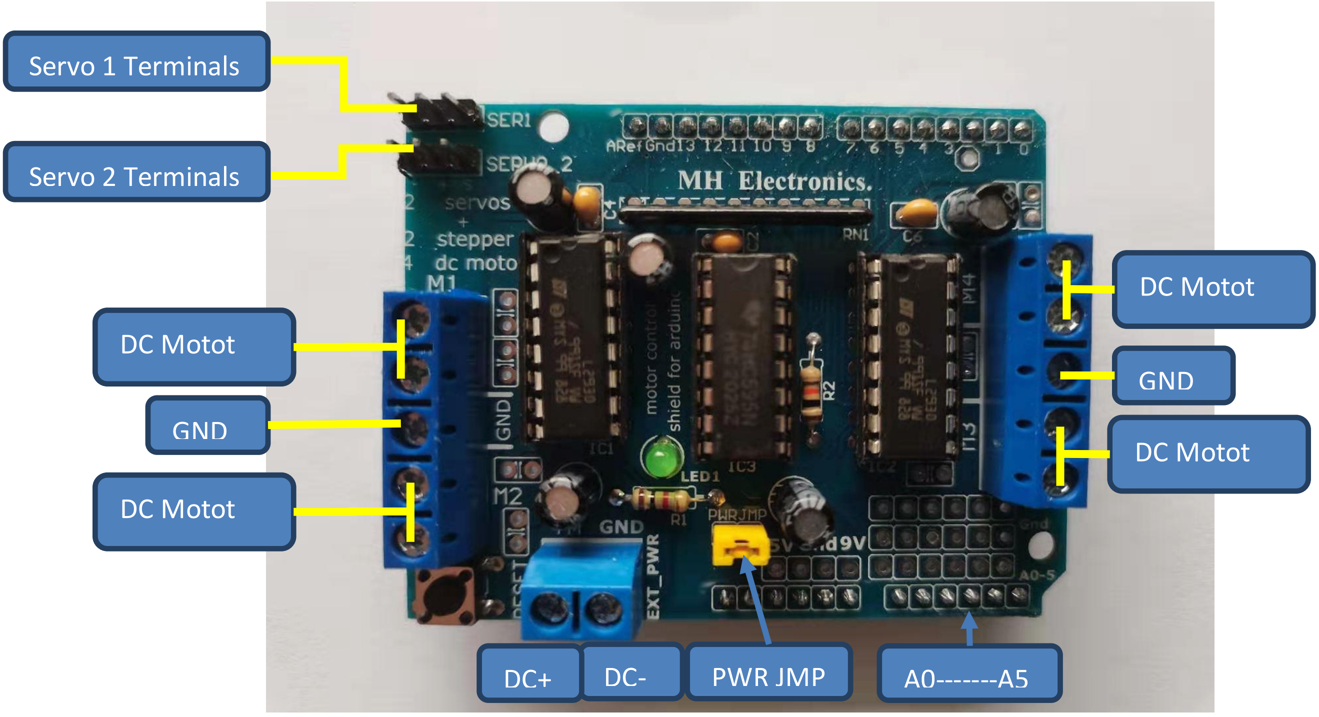
Use a single DC power supply to power both the UNO board and the control shield board, just plug the DC power supply into the DC jack of the UNO board or the 2 pin EXT_PWR on the control shield board.

Please keep the power jumper of the control shield on the board, it can only be used when the working voltage of the motor is less than 9V.

Note:

Do not supply more than 9V at the EXT_PWR input when the jumper is in place, or you may damage the Arduino UNO Board!

Output/input terminal of the L293D motor control shield



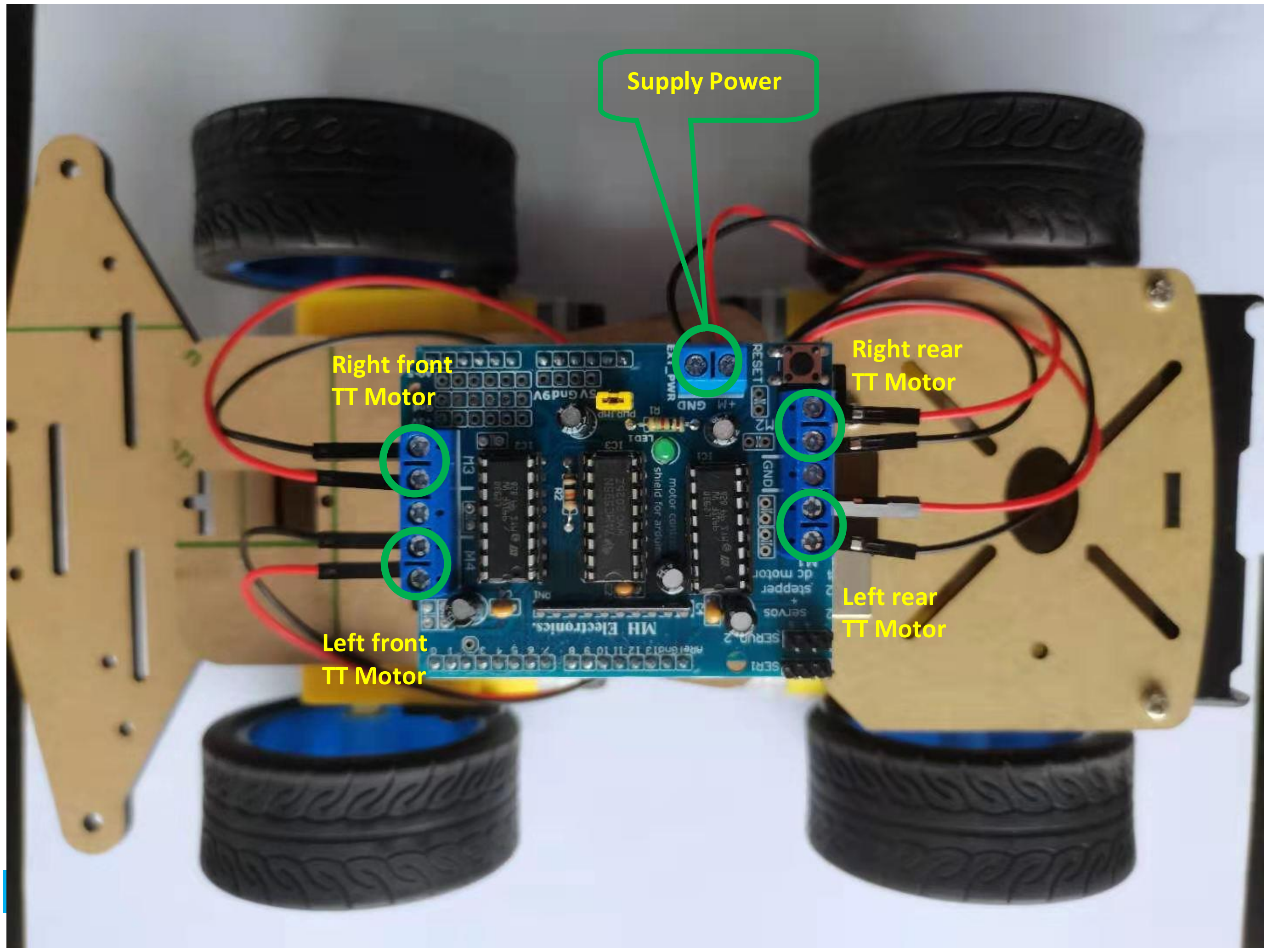
- The output channels of the two L293D chips output M1, M2, M3 and M4 through two 5-pin screw terminals. These terminals support DC motors with operating voltages between 4.5 and 25V.
- Each channel on the module can supply up to 600mA to the DC motor. However, the amount of current supplied to the motor depends on the power supply to the system.
- You can also connect two stepper motors to the output terminals. One stepper motor is connected to motor ports M1-M2 and the other is connected to M3-M4.
- If it is a unipolar stepper motor, connect the center tap of the unipolar stepper motor to the GND terminal.
- Two servos can be connected by pulling the 16-bit PWM output lines out to two 3 pin connectors.

Unused pins on L293D motor control shield

Mount the shield on the UNO board, it does not occupy the digital pins #2, #13 and analog pins A0-A5 of the UNO board. If you want to use these pins, you can connect some headers to the corresponding places.

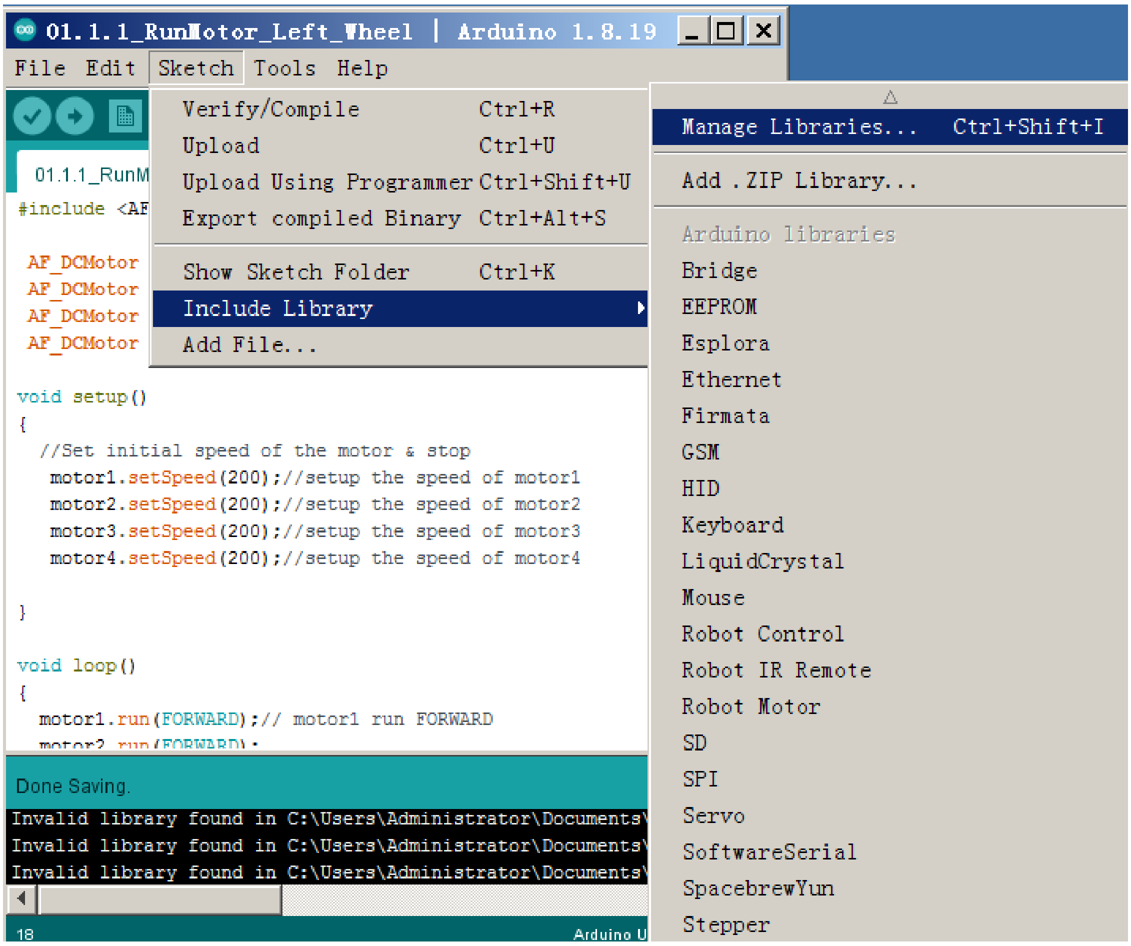
Circuit connection

First fix the UNO board on the 4WD car body, install the L293D motor control shield on the UNO board, and then connect the 4 TT motors to the L293D expansion board as shown in the figure below, and connect the power wire of the 18650 battery box to the EXT_PWR terminal of the control shield, note that the positive and negative poles cannot be connected wrongly.



Install AFMotor Library

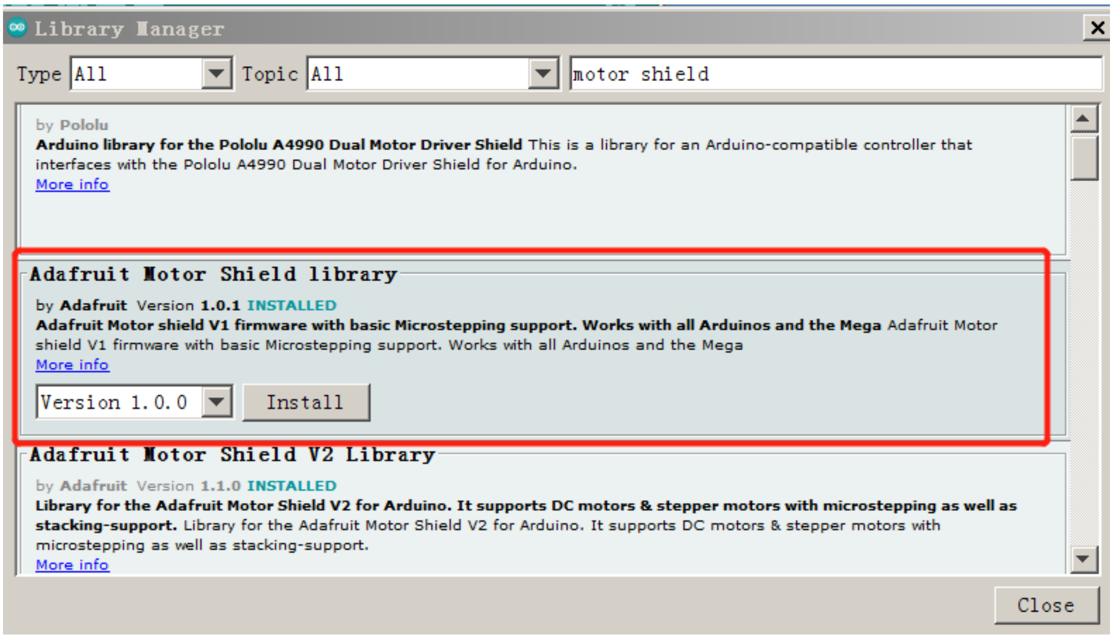
In order to establish communication with the L293D motor control shield, we need to install the **AFMotor.h** library first so that it can issue commands to control DC, stepper and servo motors. Click to open Arduino IDE， then click “Sketch”>“Include library”>“Manage Libraries...”， Wait for the library manager to download the library index and update the list of installed libraries.



```
#include <AF

 AF_DCMotor
 AF_DCMotor
 AF_DCMotor
 AF_DCMotor

void setup()
{
  //Set initial speed of the motor & stop
   motor1.setSpeed(200);//setup the speed of motor1
   motor2.setSpeed(200);//setup the speed of motor2
   motor3.setSpeed(200);//setup the speed of motor3
   motor4.setSpeed(200);//setup the speed of motor4

}

void loop()
{
  motor1.run(FORWARD);// motor1 run FORWARD
  motor2.run(FORWARD);
```

```
Done Saving.
Invalid library found in C:\Users\Administrator\Documents\
Invalid library found in C:\Users\Administrator\Documents\
Invalid library found in C:\Users\Administrator\Documents\
```

Type " **motor shield** " to filter your search. Look for the Adafruit Motor Shield library (Version 1) provided by Adafruit. Click the entry and select Install.



motor shield

by Pololu
Arduino library for the Pololu A4990 Dual Motor Driver Shield This is a library for an Arduino-compatible controller that interfaces with the Pololu A4990 Dual Motor Driver Shield for Arduino.
More info

Adafruit Motor Shield library
by **Adafruit** Version **1.0.1** **INSTALLED**
Adafruit Motor shield V1 firmware with basic Microstepping support. Works with all Arduinos and the Mega Adafruit Motor shield V1 firmware with basic Microstepping support. Works with all Arduinos and the Mega
More info

Adafruit Motor Shield V2 Library
by **Adafruit** Version **1.1.0** **INSTALLED**
Library for the Adafruit Motor Shield V2 for Arduino. It supports DC motors & stepper motors with microstepping as well as stacking-support. Library for the Adafruit Motor Shield V2 for Arduino. It supports DC motors & stepper motors with microstepping as well as stacking-support.
More info

Click "File---"Open" in the IDE interface, and select the code under the path of "4WD Car Chassis Kit Tutorial\ Sketches\ 02.1_Testing_TT_Motor".
After the code is compiled successfully, connect the UNO board on the 4WD body to the computer with a USB cable, and upload the program.
After the upload is successful, you can power on and test the TT Motor.

Code:

```
/**********************************************************************
  Product     : Cokoino 4WD Car chassis kit
  Auther      : www.cokoino.com
  Modification: 2024/01/02

           Motor4 ---- Motor3
                  |  |
                  |  |
                  |  |
           Motor1 ---- Motor2


**********************************************************************/

#include <AFMotor.h>
 AF_DCMotor motor1(1);//define motor1
 AF_DCMotor motor2(2);//define motor2
 AF_DCMotor motor3(3);//define motor3
 AF_DCMotor motor4(4);//define motor4

void setup()
{

}

void loop()
{
  motor1.setSpeed(200);//setup the speed of motor1
  motor2.setSpeed(200);//setup the speed of motor2
  motor3.setSpeed(200);//setup the speed of motor3
  motor4.setSpeed(200);//setup the speed of motor4
  //car move forward
  motor1.run(BACKWARD);//motor1 run BACKWARD
  motor2.run(FORWARD); //motor2 run FORWARD
  motor3.run(BACKWARD);//motor3 run BACKWARD
  motor4.run(FORWARD); //motor4 run FORWARD
  delay(1600);
  motor1.run(RELEASE);// motor1 stop run
  motor2.run(RELEASE);// motor2 stop run
  motor3.run(RELEASE);// motor3 stop run
```

```
motor4.run(RELEASE);// motor4 stop run
delay(1000);
 //car move backward
motor1.run(FORWARD);// motor1 run FORWARD
motor2.run(BACKWARD);//motor2 run BACKWARD
motor3.run(FORWARD);// motor3 run FORWARD
motor4.run(BACKWARD);//motor4 run BACKWARD
delay(1600);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
delay(1000);
//car move forward and turn left
motor1.setSpeed(100);//setup the speed of motor1
motor2.setSpeed(200);//setup the speed of motor2
motor3.setSpeed(200);//setup the speed of motor3
motor4.setSpeed(100);//setup the speed of motor4

motor1.run(BACKWARD);//motor1 run BACKWARD
motor2.run(FORWARD); //motor2 run FORWARD
motor3.run(BACKWARD);//motor3 run BACKWARD
motor4.run(FORWARD); //motor4 run FORWARD
delay(2000);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
delay(1000);
 //car move forward and turn right
motor1.setSpeed(200);//setup the speed of motor1
motor2.setSpeed(100);//setup the speed of motor2
motor3.setSpeed(100);//setup the speed of motor3
motor4.setSpeed(200);//setup the speed of motor4

motor1.run(BACKWARD);//motor1 run BACKWARD
motor2.run(FORWARD); //motor2 run FORWARD
motor3.run(BACKWARD);//motor3 run BACKWARD
motor4.run(FORWARD); //motor4 run FORWARD
delay(2000);
motor1.run(RELEASE);
motor2.run(RELEASE);
motor3.run(RELEASE);
motor4.run(RELEASE);
delay(1000);
 //car move rotation
motor1.setSpeed(200);//setup the speed of motor1
motor2.setSpeed(200);//setup the speed of motor2
```

```
  motor3.setSpeed(200);//setup the speed of motor3
  motor4.setSpeed(200);//setup the speed of motor4

  motor1.run(BACKWARD); //motor1 run BACKWARD
  motor2.run(BACKWARD); //motor2 run BACKWARD
  motor3.run(FORWARD);  //motor3 run FORWARD
  motor4.run(FORWARD);  //motor4 run FORWARD
  delay(2000);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(1000);

}
```

Make your suggestion and get support

THANK YOU for participating in this learning experience!

If you find errors, omissions or you have suggestions and/or questions about this lesson, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://cokoino.com/

Thank you again for choosing Cokoino products.