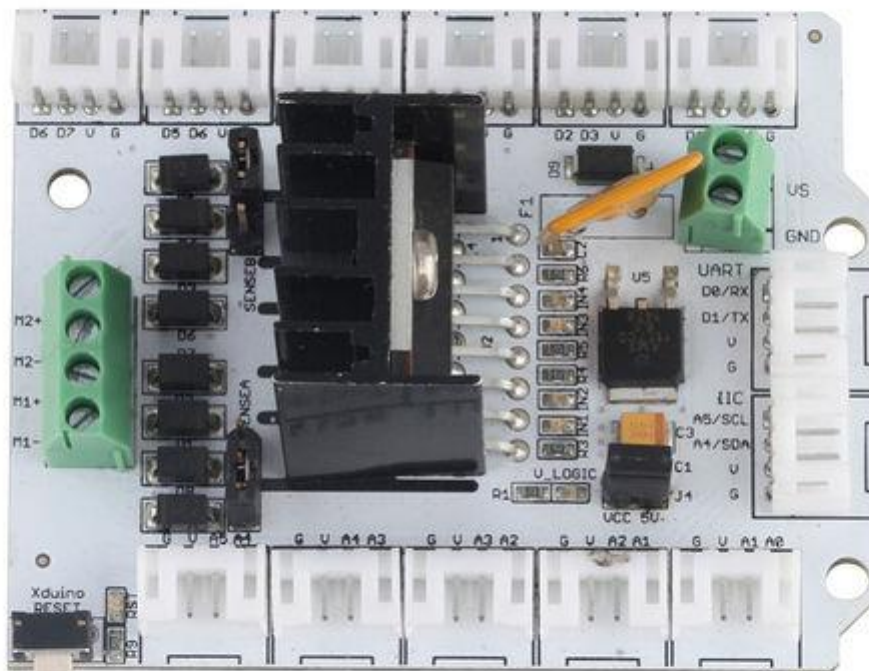
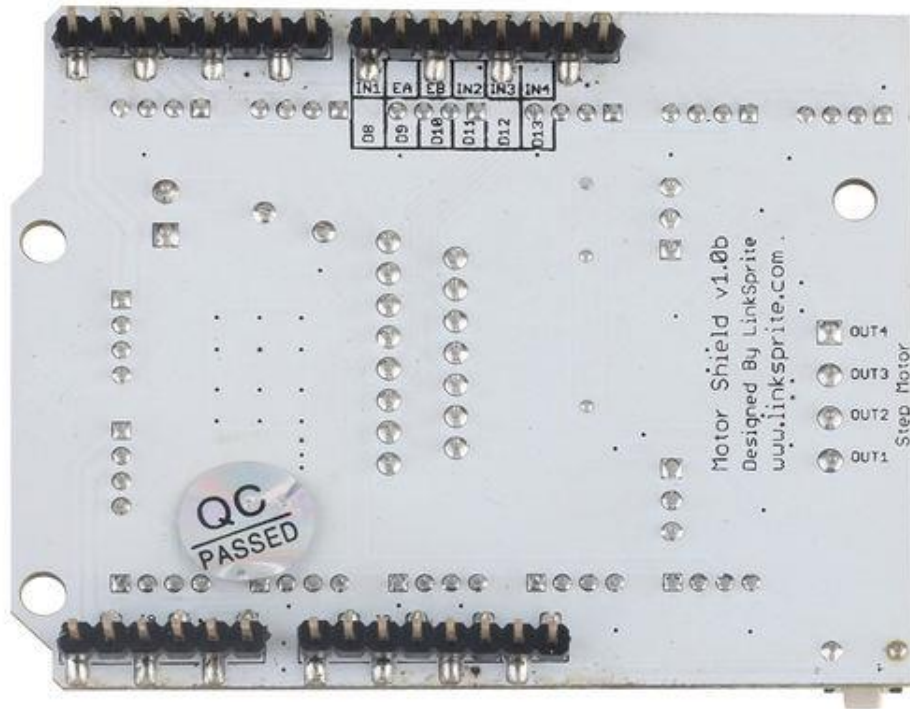


LinkSprite Arduino Motorshield

Introduction

The Motor Shield is a perfect platform for robotics and mechanical applications. It enables the Arduino to drive two brushed DC motors or one 4-wire two-phase stepper motor. Based on the H-bridge driver Chip L298N motor driver integrated circuit, it requires a 6V to 15V power supply to power the motor and also includes an on-board 5V voltage regulator for powering the main Arduino board. Additional plug and play 2.00mm spacing Grove sensors can be connected to the 13 on-board Grove connectors.





Features

- Standard Arduino compatible
- Pulse-width modulated speed control mode
- 4 direction indicator lights
- Compatible with 2.00mm spacing Grove plug and play sensors
- Extra large heat sink helps handle big loads
- Supports up to 14 servos

CAUTION:

The Driver IC and Heat sink may become very hot when working with current more than 1000mA.

Specifications

| Item | Min | Typical | Max | Unit |
|-----------------------------------|----------------|---------|--------------------------|------|
| Logic Control Voltage | 4.5 | 5 | 5.5 | V |
| Motor Supply Voltage | 6 | / | 15 | V |
| Output Voltage | 0 | / | V _{input} -1 | V |
| Output Current(For Each Channel) | / | / | 2000 * | mA |
| Output Duty range | 0%~100% | | | / |
| Dimension | 68.5x54.5x29.5 | | | mm |
| Net Weight | 37 | | | g |

* Recommend maximum duty <50% at full load

Interface Function

| Pin Name | Direction | Connection to Arduino | Description |
|----------|-----------|-----------------------|---|
| VCC | / | VCC | Power supply selector |
| VS, GND | / | / | Power Supply for Motor, 6V - 15V |
| EA | Input | D9 | TTL Compatible Enable Input of Bridge A |
| EB | Input | D10 | TTL Compatible Enable Input of Bridge B |
| IN1 | Input | D8 | TTL Compatible |

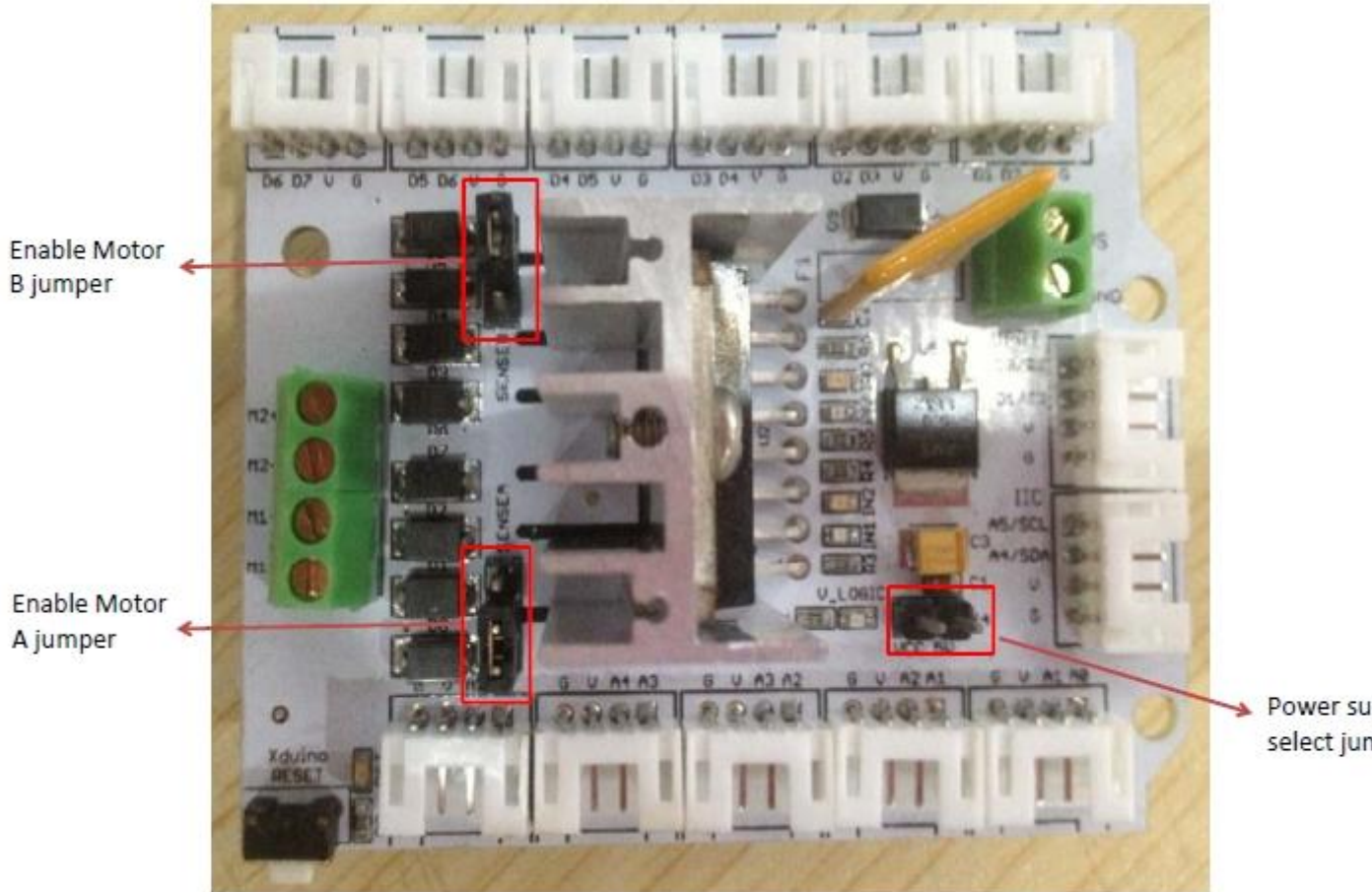
| | | | |
|----------|--------|-----|---------------------------------------|
| | | | Inputs of the Bridge A |
| IN2 | Input | D11 | TTL Compatible Inputs of the Bridge A |
| IN3 | Input | D12 | TTL Compatible Inputs of the Bridge B |
| IN4 | Input | D13 | TTL Compatible Inputs of the Bridge B |
| M1-, M1+ | Output | / | Outputs of the Bridge A |
| M2-, M2+ | Output | / | Outputs of the Bridge B |

Arduino Motor A Enable Pin: 9

Arduino Motor B Enable Pin: 10

Usage

Jumpers



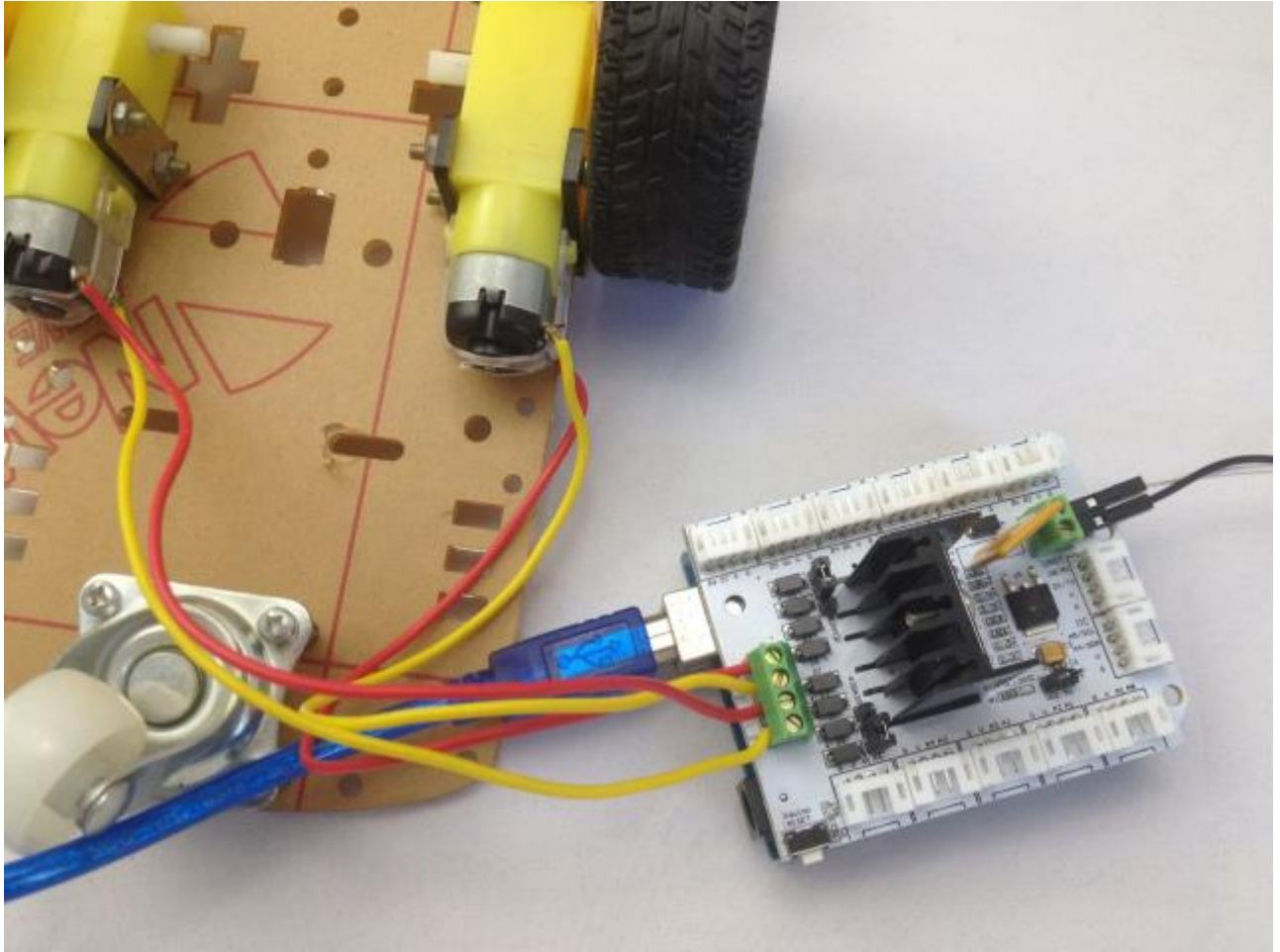
Enable Motor B jumper: Connect above two interfaces with a jumper cap when driving the DC motor A.

Enable Motor A jumper: Connect below two interfaces with a jumper cap when driving the DC motor B.

Power supply select jumper: Connect with a jumper cap, Arduino can be powered by the drive voltage via a on-board voltage regulator, which regulate the drive voltage to 5v.

Hardware Installation

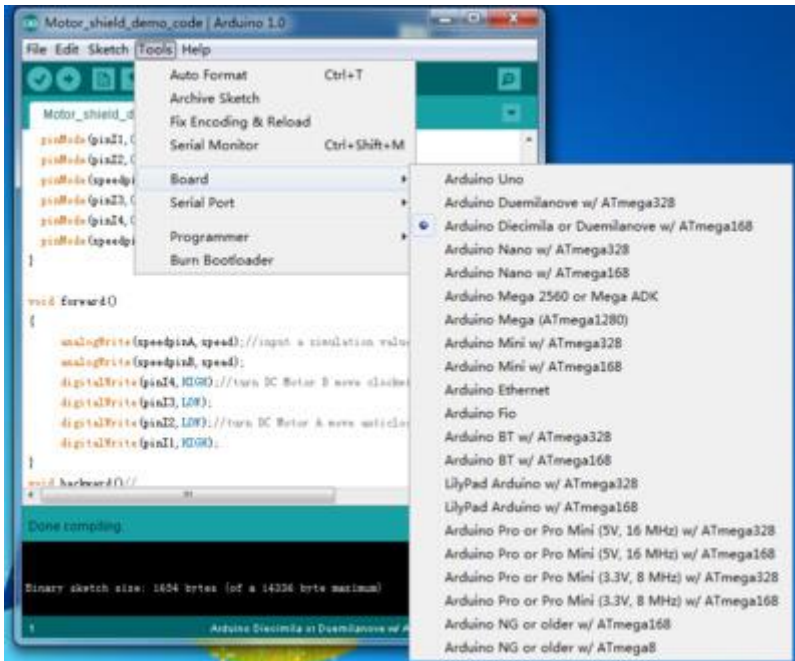
Connect DC motor to the motor shield Output Pins M1+& M1-(M2+& M2-), if using a 4-wire stepper motor, make sure the 4 Pins are correctly connected. And connect the drive voltage (The voltage needed is upon the motor used, Refer to the motor datasheet) to the Power supply terminals.



As the jumper J4 inserted, Arduino can be powered by the drive voltage via a on-board voltage regulator, which regulate the drive voltage to 5v.

Software

Download the [Motor shield Demo code](#) and open the **Motor_shield_demo_code** sketch. Select the corresponding Arduino board in **tools-boards** and serial port in **tools-serial port**.



Upload the sketch to your arduino board, then you would found the 2 motors begin to work. If you power the motor shield via the power terminal and the power jumper J4 was inserted, you can remove the USB cable now. Speed control is achieved through conventional PWM which can be obtained from Arduino's PWM output D9(EA) and D10(EB). Using the function **analogWrite()** to generate the drive PWM:

```
int speedpinA=9;//enable motor A
int speedpinB=10;//enable motor B
int speed =127;//define the speed of motor
.....
analogWrite(speedpinA,speed);//AnalogWrite to Generate PWM to control
the motor speed
analogWrite(speedpinB,speed);
```

And Set the Digital D8(IN1) and D11(IN2) to control the motor attached to OUT1/2, while digital D12(IN3) and D13(IN4) to control motor attached to OUT3/4.

```
int pinI1=8;//define I1 interface
int pinI2=11;//define I2 interface
.....
digitalWrite(pinI2,HIGH);//turn DC Motor A move clockwise
digitalWrite(pinI1,LOW);
```

Follows the logic between EA(B) and INx with motor motion:

| EA(B) | IN1(IN3) | IN2(IN4) | Motion of MotorA(B) |
|-------|----------|----------|---------------------|
| 0 | x | x | Stop |
| 1 | 0 | 0 | Stop |
| 1 | 0 | 1 | Clockwise |
| 1 | 1 | 0 | Anticlockwise |
| 1 | 1 | 1 | Stop |

```
/*
  LinkSprite Stepper Motor Control - one revolution

  Adapts the Stepper example for use with the inkSprite motor shield.
  This program drives a unipolar or bipolar stepper motor
  by using the included Stepper library of the Arduino.
  The motor is attached to the inkSprite motor shield and an Arduino.

  The digital pins 8,11,12,13 drive the L298N and are used when creating
  the stepper object
  Digital pins 9 and 10 must be high to enable the chip.

  The motor should revolve one revolution in one direction, then
  one revolution in the other direction.

  */

#include <Stepper.h>

const int stepsPerRevolution = 200; // change this to fit the number
of steps per revolution
// for your motor

// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 8,11,12,13);

void setup() {
  // set the speed at 60 rpm:
  myStepper.setSpeed(60);
  // initialize the serial port:
  Serial.begin(9600);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}
```



```
digitalWrite(9,HIGH);  
digitalWrite(10,HIGH);  
}  
  
void loop() {  
  // step one revolution in one direction:  
  Serial.println("clockwise");  
  myStepper.step(stepsPerRevolution);  
  delay(500);  
  
  step one revolution in the other direction:  
  Serial.println("counterclockwise");  
  myStepper.step(-stepsPerRevolution);  
  delay(500);  
}
```