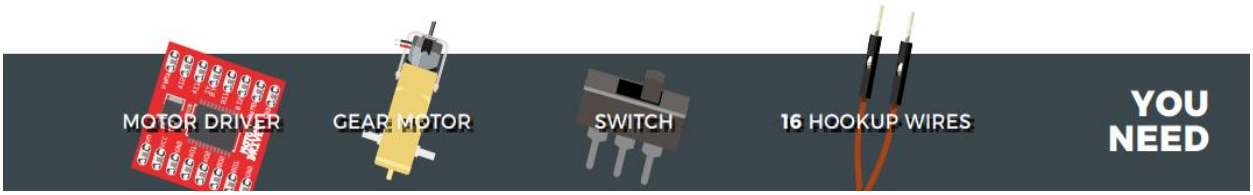
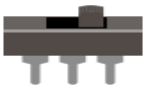
In this circuit, you will learn the basic concepts behind motor control. Motors require a lot of current, so you can't drive them directly from a digital pin on the RedBoard. Instead, you'll use what is known as a motor controller or motor driver board to power and spin the motor accordingly.

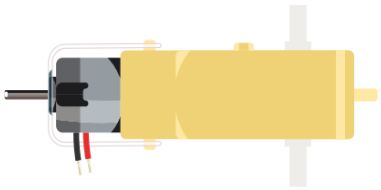
Grab the following quantities of each part listed to build this circuit:



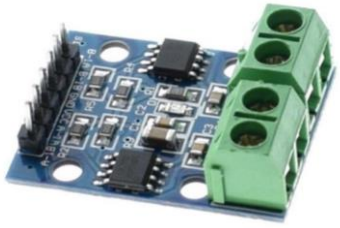
NEW COMPONENTS



SWITCHES are components that control the open-ness or closed-ness of an electric circuit. Just like the momentary buttons used in earlier circuits, this type of switch can only exist in one of two states: open or closed. However, a switch is different in that it will stay in the position it was last in until it is switched again.



THE MOTORS in your Inventor's Kit have two main parts: a small DC motor that spins quickly and a plastic gearbox that gears down the output from the hobby motor so that it is slower but stronger, allowing it to move your robot. The motors have a clever design allowing you to attach things that you want to spin fast (like a small fan or flag) to the hobby motor, and things that you want to be strong (like a wheel) to the plastic axle sticking out the side of the motor. The included wheels just so happen to fit on the plastic axles.



The L9110S 2-Channel motor driver module is a compact board that can be used to drive small robots. This module has two independent motor driver chips which can each drive up 800mA of continuous current. The boards can be operated from 2.5V to 12V enabling this module to be used with both 3.3V and 5V microcontrollers. A set of female header pins is used to connect this module to a microcontroller. The motors are attached via two sets of screw terminals. A PWM Pulse Width Modulation signal is used to control the speed of a motor and a digital output is used to change its direction. This module can also be used to drive a single four-line two phase stepper motor. Four holes make this board easy to mount onto your robot or other project

Specifications
• On-board 2 L9110 motor control chip
• Module can be driven by two dc motors at the same time or one phase 4 line 2 type stepping motor
• Input voltage: 2.5-12V DC
• Each channel has a continuous output current 800 ma
• PCB Size: 29.2mm x 23mm

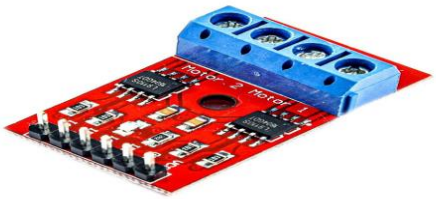
L9110S Dual-Channel Driver Module Features

Motor driver modules are very common nowadays and widely used to control the speed and direction of motors. The L9110S dual-channel module is one of them. This module can control two DC motors and one stepper motor. It is based on L9110 IC. The key features are:

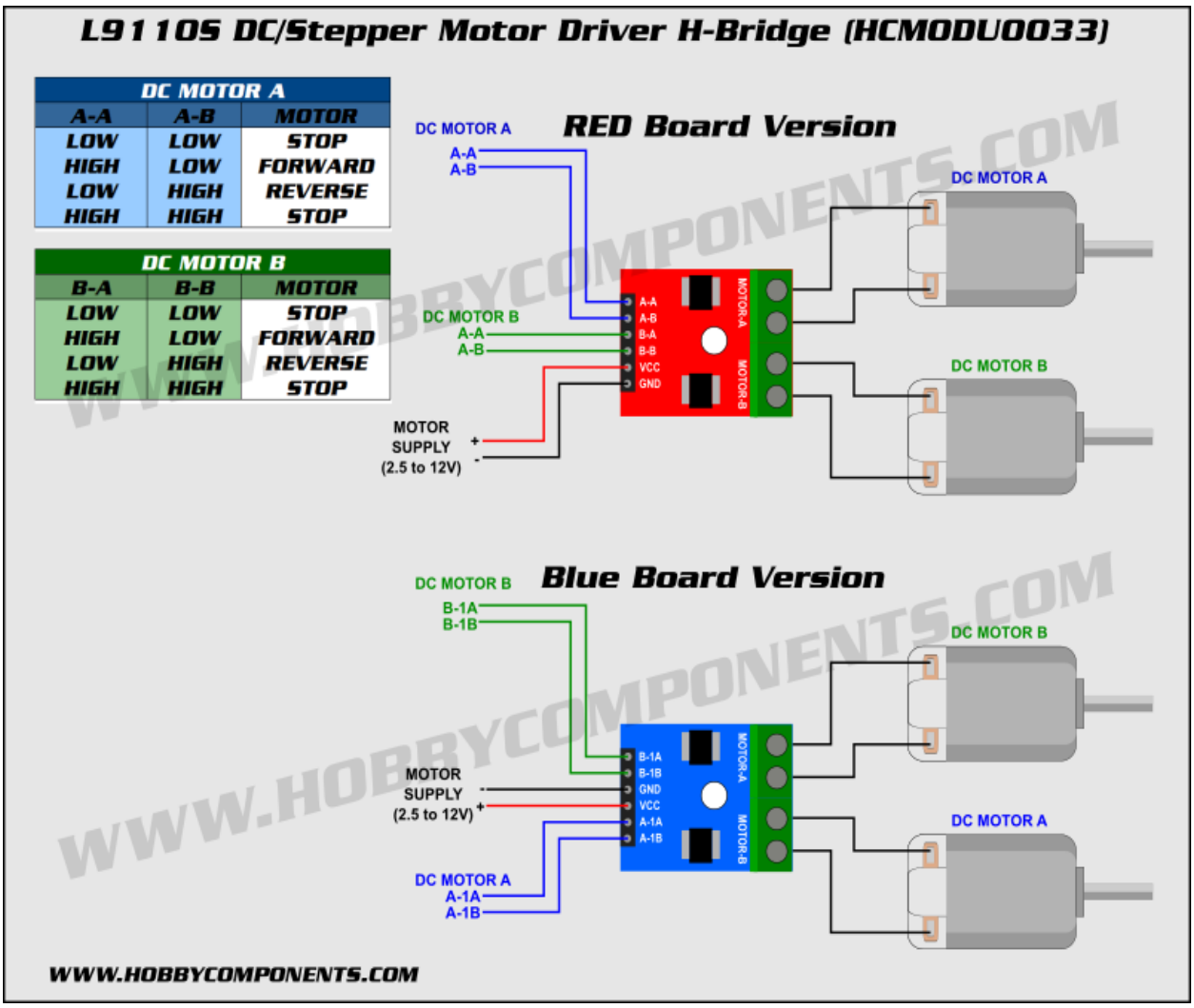
- The allowable continuous current for each channel: 800 mA
- The maximum allowable current: 1.5 A
- Power supply: 2.5V to 12V

Note

The two connected pins to each DC motor can be PWM or digital. If defined as digital, it can only control the direction of motors motion.



HOOKUP GUIDE
READY TO START HOOKING EVERYTHING UP? Check out the circuit diagram and hookup table below to see how everything is connected.

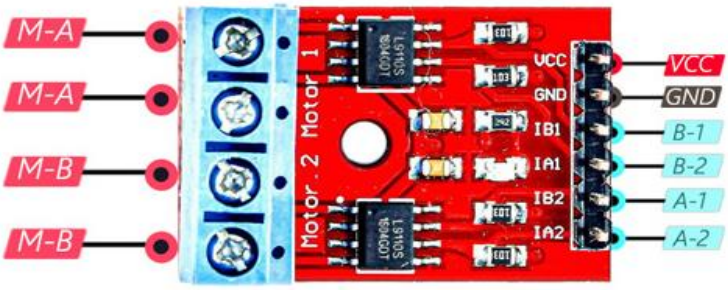


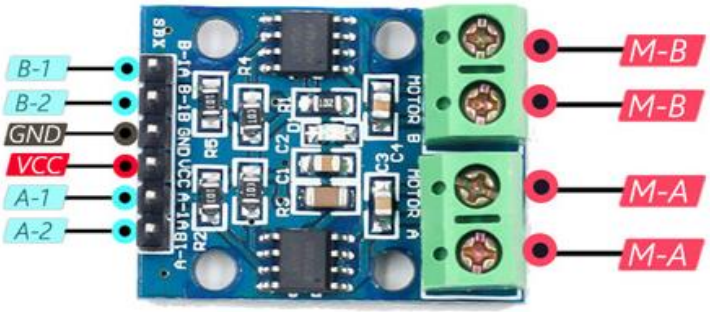
L9110S Dual-Channel Driver Module Pinout

This Module has following pins:

- **VCC**: Module power supply
- **GND**: Ground
- **M-A:** Motor A pin
- **M-B:** Motor B pin
- **A-1:** Control signal for motor A
- **A-2:** Control signal for motor A

- **B-1:** Control signal for motor B
- **B-2:** Control signal for motor B



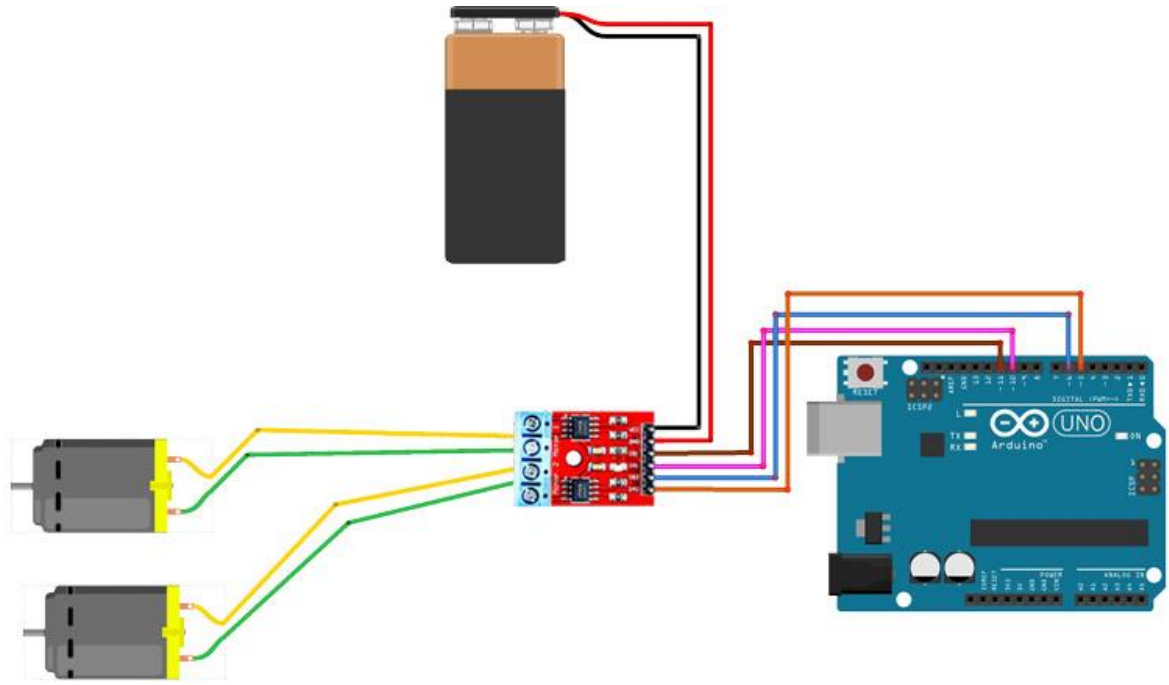


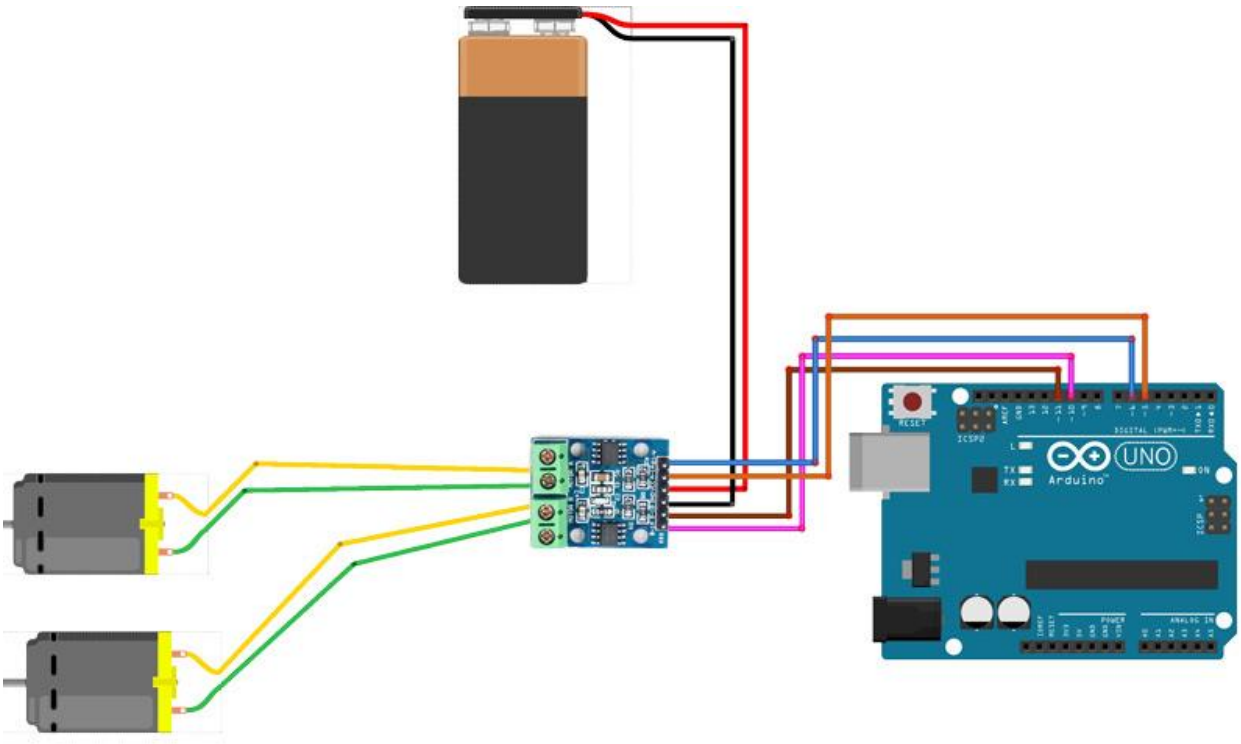
	AIN1	AIN2
Forward Direction	HIGH	LOW
Reverse Direction	LOW	HIGH
Brake / Stopped	LOW	LOW
Brake / Stopped	HIGH	HIGH

Interfacing L9110S Dual-Channel Driver Module with Arduino

Step 1: Circuit

The 2 following images show how you should connect Arduino to these modules. Connect wires accordingly.





Step 2: Code

Upload the following code to your Arduino.

SOURCE CODE:

```
#define A1 5  // Motor A pins
#define A2 6
#define B1 10 // Motor B pins
#define B2 11

int incomingByte = 0; // for incoming serial data

void setup() {

  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
  pinMode(B1, OUTPUT);
  pinMode(B2, OUTPUT);

  digitalWrite(A1, LOW);
  digitalWrite(A2, LOW);
  digitalWrite(B1, LOW);
  digitalWrite(B2, LOW);

  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps

  Serial.println("select direction of movement");
  Serial.println("1.forward");
  Serial.println("2.backward");
  Serial.println("3.stop");

}
int  input = 0;
void loop() {

  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();
    input = incomingByte - 48; //convert ASCII code of numbers to 1,2,3

  switch (input) {
    case 1:       // if input=1 ..... motors turn forward
      forward();
      break;
    case 2:       // if input=2 ..... motors turn backward
      backward();
      break;
    case 3:       // if input=1 ..... motors turn stop
```

```
    Stop();
    break;
  }
  delay(200);
  input=0;
}
}
void forward() {          //function of forward
  analogWrite(A1, 255);
  analogWrite(A2, 0);
  analogWrite(B1, 255);
  analogWrite(B2, 0);
}

void backward() {         //function of backward
  analogWrite(A1, 0);
  analogWrite(A2, 210);
  analogWrite(B1, 0);
  analogWrite(B2, 210);
}

void Stop() {             //function of stop
  digitalWrite(A1, LOW);
  digitalWrite(A2, LOW);
  digitalWrite(B1, LOW);
  digitalWrite(B2, LOW);
}
```