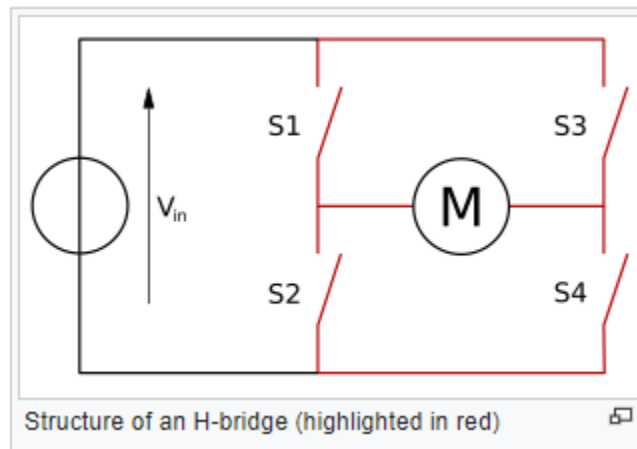# Controlling DC Motors with L293D Motor Driver

It is dangerous to hook DC motors to the BeagleBone's GPIO pins as the current required can easily exceed the maximum rating of the specifications and damage your board. Therefore we use a L293D motor driver IC to interface with the motors instead. L293D is a very commonly used motor driver, it's essentially a H-bridge circuit in a chip.

Structure of an H-bridge (highlighted in red)

via https://en.wikipedia.org/wiki/H-bridge

This document guides the user through
1. Wiring 2x DC motors with to the motor driver and an external battery source
2. Controlling the motors through beaglebone GPIO pins

Required components

- 6 free GPIO pins on BeagleBone
  (in this guide we use P8_7 to P8_12)
- 1x L293D Motor Driver
- 1x or 2x DC Motor(s)
- 1x External power source

Table of Contents

1. Wiring Guide

This section hooks cover physical wiring of all components

1.1 DC Motors and battery

Since each L293D motor driver can control up to two DC motors at the same time, this guide will assume you have two DC motors to use.

1. Wire up DC motors
   DC motors have two terminals, wire them to the breadboard. We aren't concerned with positive/negative at this stage since the motor will just turn the opposite direction if you've flipped them.

2. Wire up external power source
   Motor driver supports 4.5V to 36V. You can either use batteries or a usb battery bank, though you'll have to learn to splice a USB cable on your own.
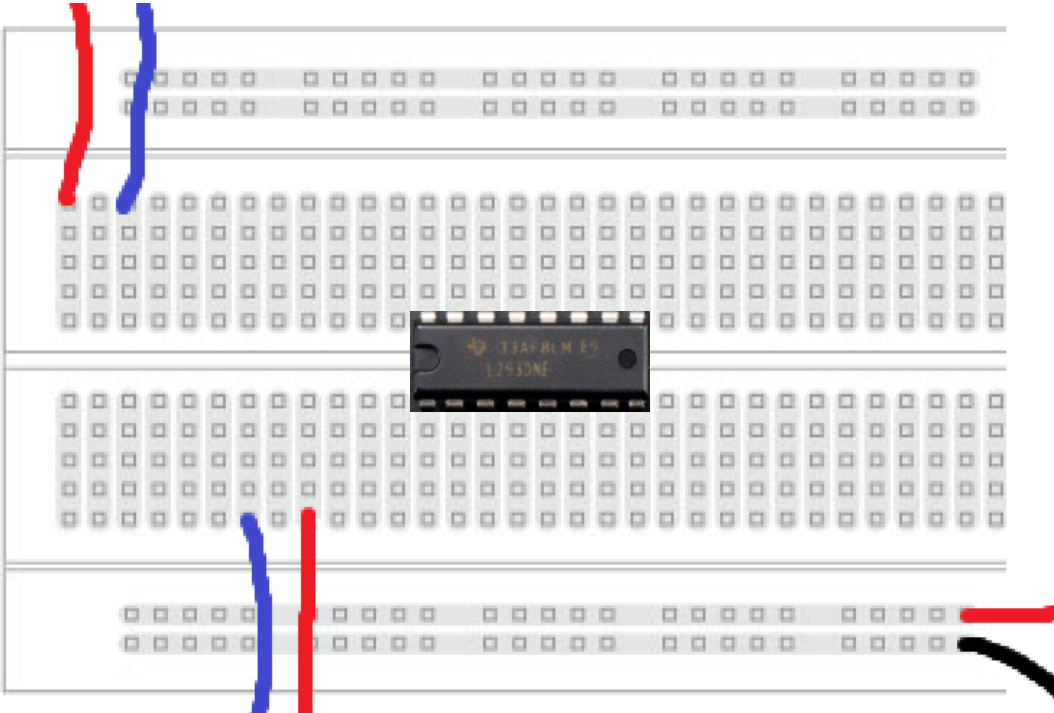   Connect the external power source to the side lanes of the breadboard.

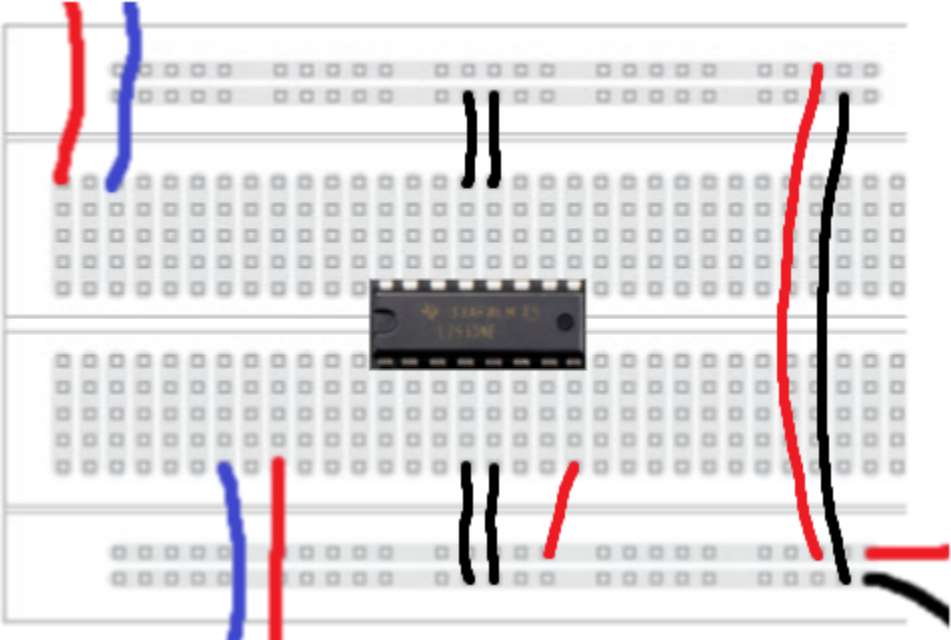At this point your setup should look something like this:
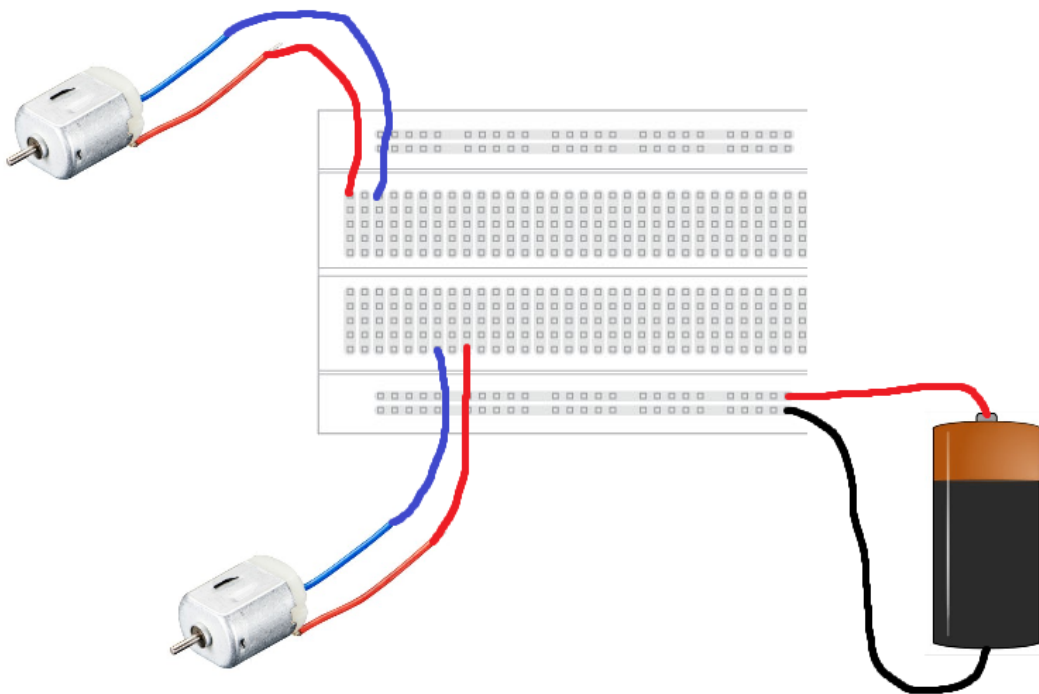
1.2 Motor Driver and BeagleBone
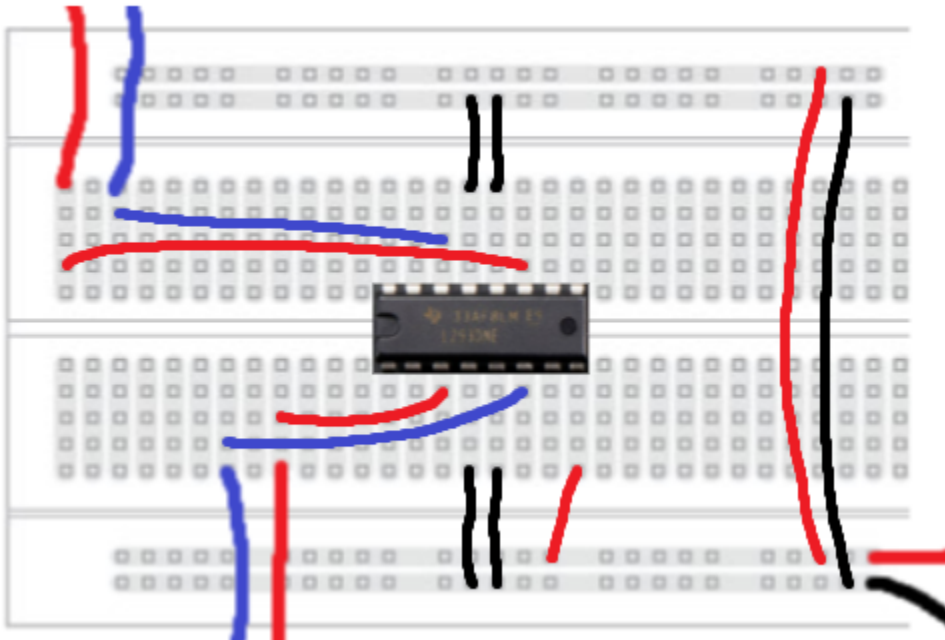
There will be a lot of wiring in this section.

1. Drop the motor driver chip in the middle of the breadboard.
    Orientation matters so make sure the notch on the side is facing left.

2. Connect external power and ground

3. Connect Motors

4. Connect to BeagleBone
Reference the below diagrams to connect all 7 motor driver terminals to the BeagleBone.

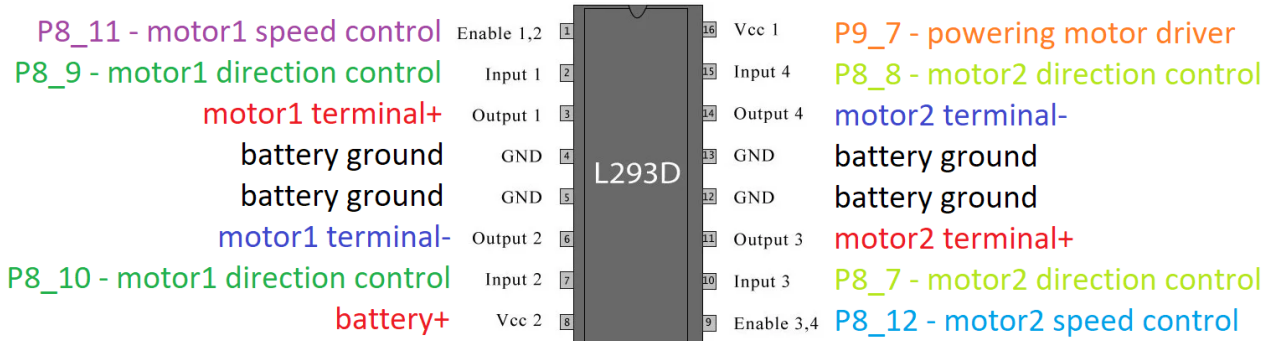First connect 7 wires to the beaglebone expansion headers.

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_113
GPIO_111	29	30	GPIO_112
GPIO_110	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

P8

DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	GPIO_63
GPIO_62	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
GPIO_86	27	28	GPIO_88
GPIO_87	29	30	GPIO_89
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
GPIO_76	39	40	GPIO_77
GPIO_74	41	42	GPIO_75
GPIO_72	43	44	GPIO_73
GPIO_70	45	46	GPIO_71

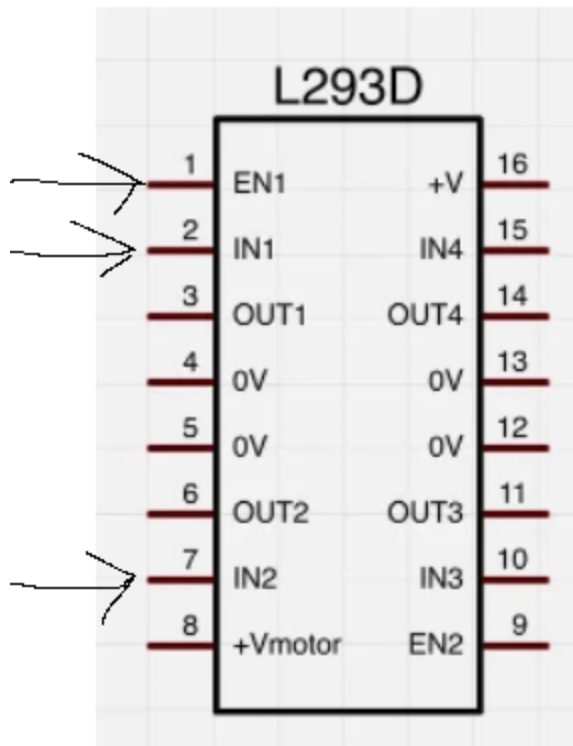Note: wiring BBG pins tend to be very loose, try holding them down when testing.

5. Now connect those wires to the corresponding lanes on the breadboard to the motor driver.

P9_7

P8_8

P8_7

P8_12

P8_10

P8_9

P8_11
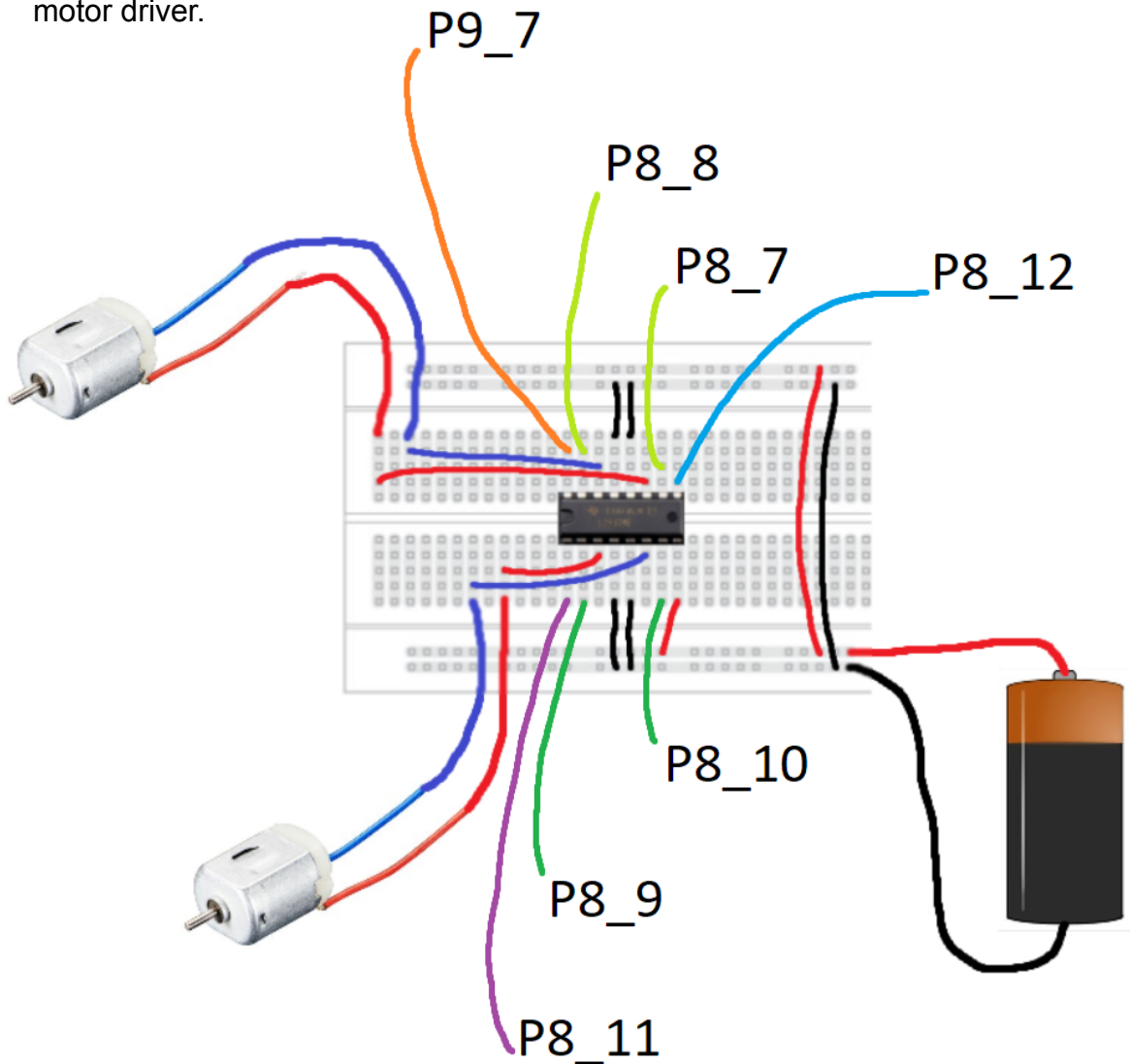
At this point, double check and make sure all 16 terminals on the motor driver are wired to something, and that no terminals are shorted. We have connected all the components in the following manner:

		L293D		
P8_11 - motor1 speed control	Enable 1,2 [1]		[16] Vcc 1	P9_7 - powering motor driver
P8_9 - motor1 direction control	Input 1 [2]		[15] Input 4	P8_8 - motor2 direction control
motor1 terminal+	Output 1 [3]		[14] Output 4	motor2 terminal-
battery ground	GND [4]		[13] GND	battery ground
battery ground	GND [5]		[12] GND	battery ground
motor1 terminal-	Output 2 [6]		[11] Output 3	motor2 terminal+
P8_10 - motor1 direction control	Input 2 [7]		[10] Input 3	P8_7 - motor2 direction control
battery+	Vcc 2 [8]		[9] Enable 3,4	P8_12 - motor2 speed control

2. Operation Reference

Taking a closer look at the L293D motor driver pin configuration, we can see the only pins controlling motor1 are pin 1, 2, and 7, or EN1, IN1, and IN2.
(similarly EN2, IN3, IN4 for motor2)

EN1 is the speed control for the motor

EN1	Motor status
Low(0)	Motor OFF
High(1)	Motor ON (100%)

You can also control the speed of the motor using PWM but we won't cover it in this guide.

IN1 and IN2 controls the direction of the motor

IN1	IN2	Motor Direction
Low(0)	Low(0)	Motor OFF
High(1)	Low(0)	Forward
Low(0)	High(1)	Backward
High(1)	High(1)	Motor OFF

You should now be able to create your own motor controller program by configuring BeagleBone's GPIO pins to the 6 speed and control pins on the motor driver.

3. Sample C program to drive motors

```c
#include <stdlib.h>
#include <stdio.h>

// P8_7 -> IN3 on L293D
#define D66 "/sys/class/gpio/gpio66/direction"
#define V66 "/sys/class/gpio/gpio66/value"
// P8_8 -> IN4 on L293D
#define D67 "/sys/class/gpio/gpio67/direction"
#define V67 "/sys/class/gpio/gpio67/value"
// P8_12 -> EN2 on L293D
#define D44 "/sys/class/gpio/gpio44/direction"
#define V44 "/sys/class/gpio/gpio44/value"

void initializeMotorDriver(void) {
    exportPin(66);
    exportPin(67);
    exportPin(44);
    writeToFile(D66, "out");
    writeToFile(D67, "out");
    writeToFile(D44, "out");
    //optional: implement pwm speed control
    writeToFile(V44, "1");
}

void setMotor(int mode) {
    // 0 = off, 1 = forward, 2 = backward
    switch (mode)
    {
    case 0:
        writeToFile(V66, "0");
        writeToFile(V67, "0");
        break;
    case 1:
        writeToFile(V66, "0");
        writeToFile(V67, "1");
        break;
    case 2:
        writeToFile(V66, "1");
        writeToFile(V67, "0");
        break;
    default:
        break;
    }
}
```