Controlling a DC pump/motor from BB using the L293D driver IC

L293D Summary:
The L293D driver IC allows us to use the Beaglebone’s 3.3v logic signal to control a DC motor. It is capable of powering motors with up to 36v. It contains an internal clamp diode to prevent damage to your MCU(BeagleBone) from inductive load, so no other discrete components are required to get your BB and motor up and running.

The L293D driver is meant for 5v logic level but treats >2.3v as high and <1.5v as low which means we can use the BB’s 3.3v logic for on/off functionality without LL conversion(pwm control may require 3.3-5 LL conversion)

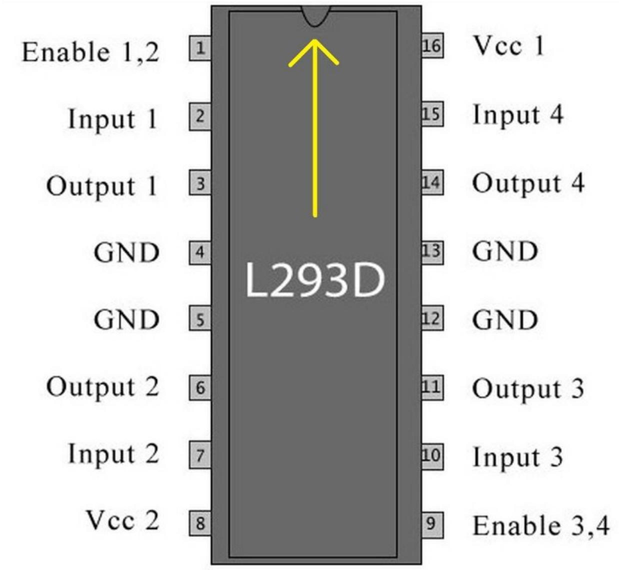
Each L293d can control either 2 motors in dual direction mode(ex RC car with a motor for both rear wheels that can drive forwards or backwards) or 4 motors in single direction mode(ex RC car with a motor for each wheel that can only drive forwards).

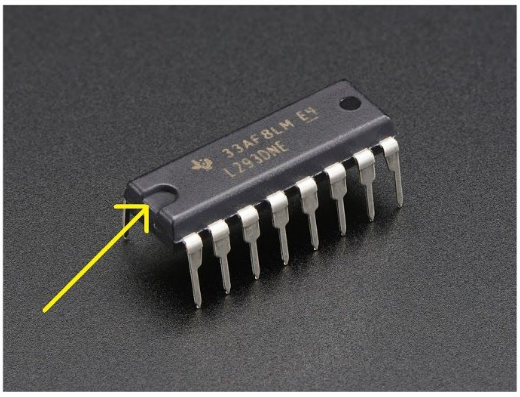
Parts needed:
1 L293d driver IC
1 BeagleBone
1 DC power supply
1-4 DC pumps/motors

Wiring up the motor circuit:
A step-by-step guide for a basic single motor dual direction setup as well as diagrams for other setups is provided below.

Beaglebone / DC PSU should not be powered while building circuit.

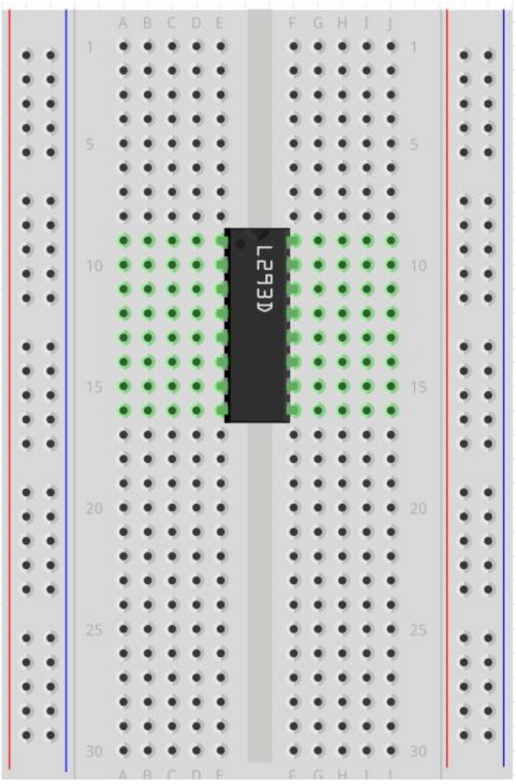
Use the groove cut in the IC and indicated on the pinout to orientate your L293d with the diagram.





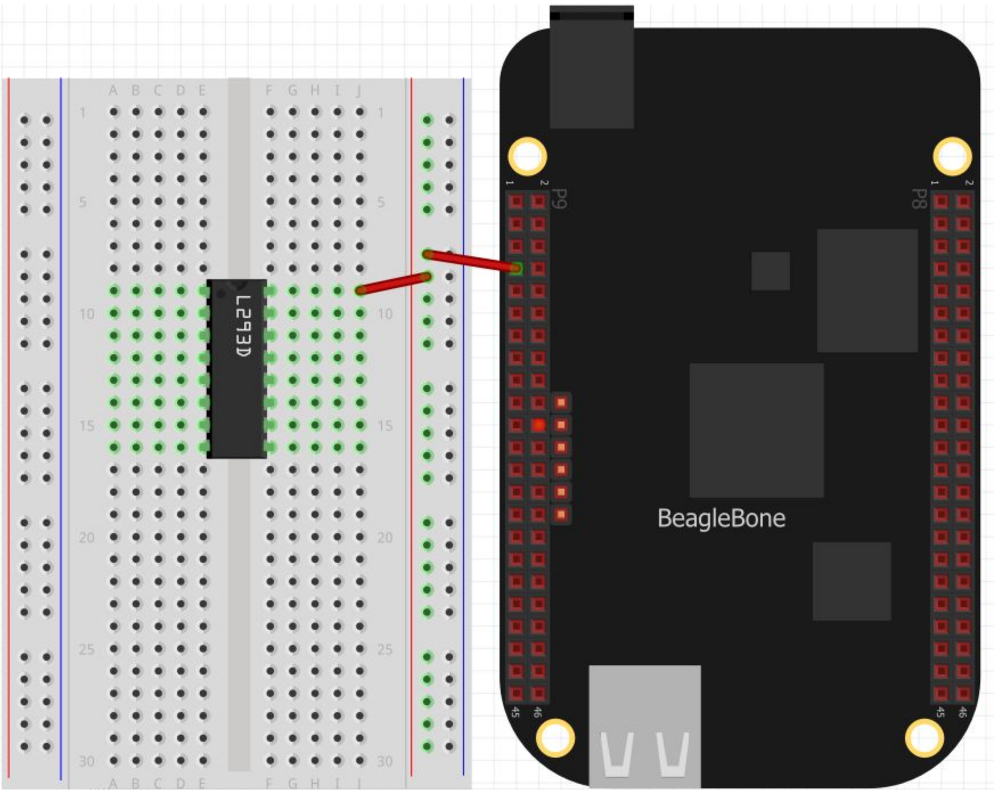
Basic L293D Setup for 1 pump, in forward/reverse mode

Step 1 (setup):
Insert the L293D into the breadboard so that it bridges the gap in the middle. The IC's 'slot' should be at the top.



Step 2 (Vcc1 / pin16):

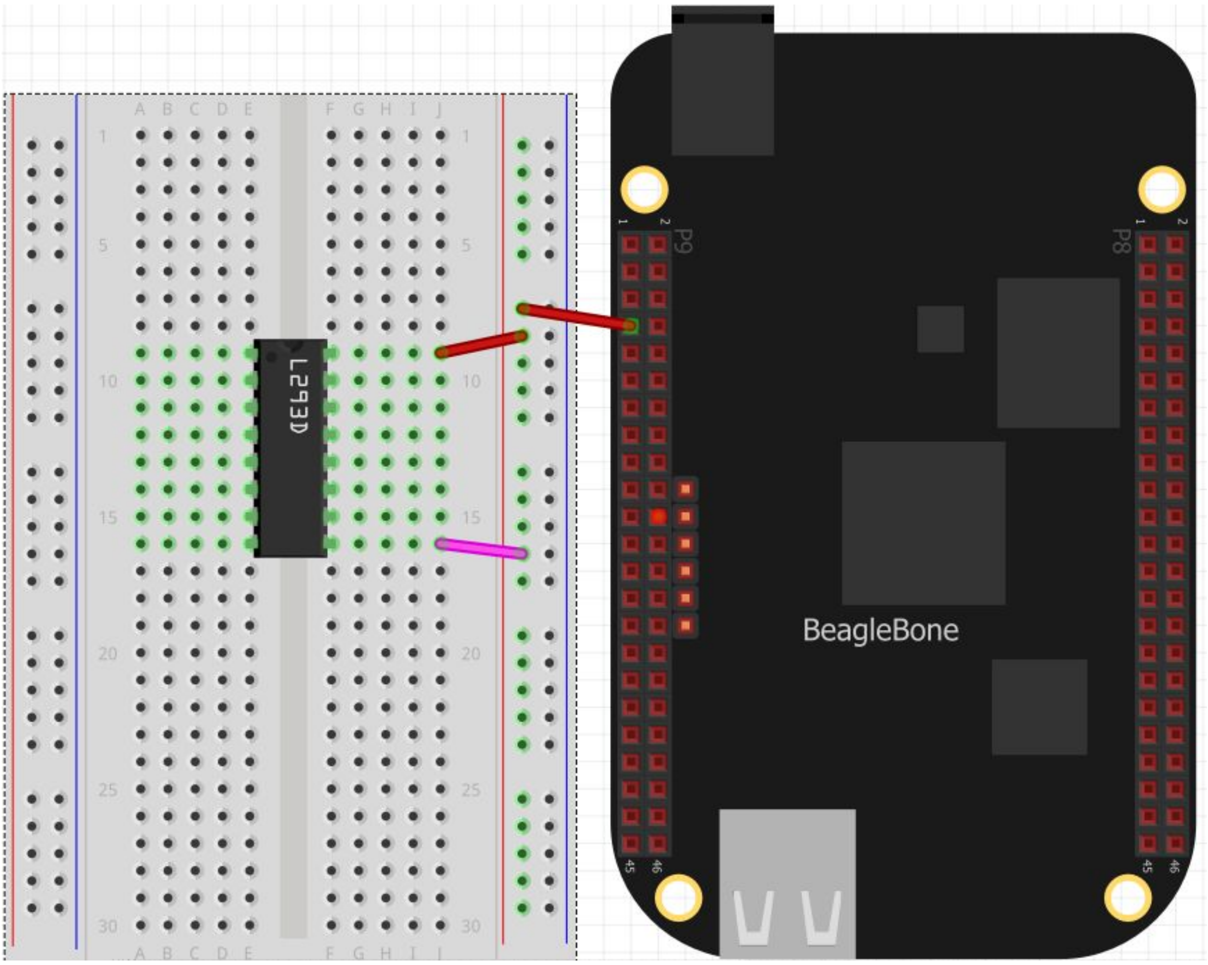
Vcc1 provides power for the L293D's internal logic. It accepts between 4.5v and 7v. We will use 5v from the beaglebone SYS_5v(P9_05).



Step 3 (Enable 3,4 / pin9):

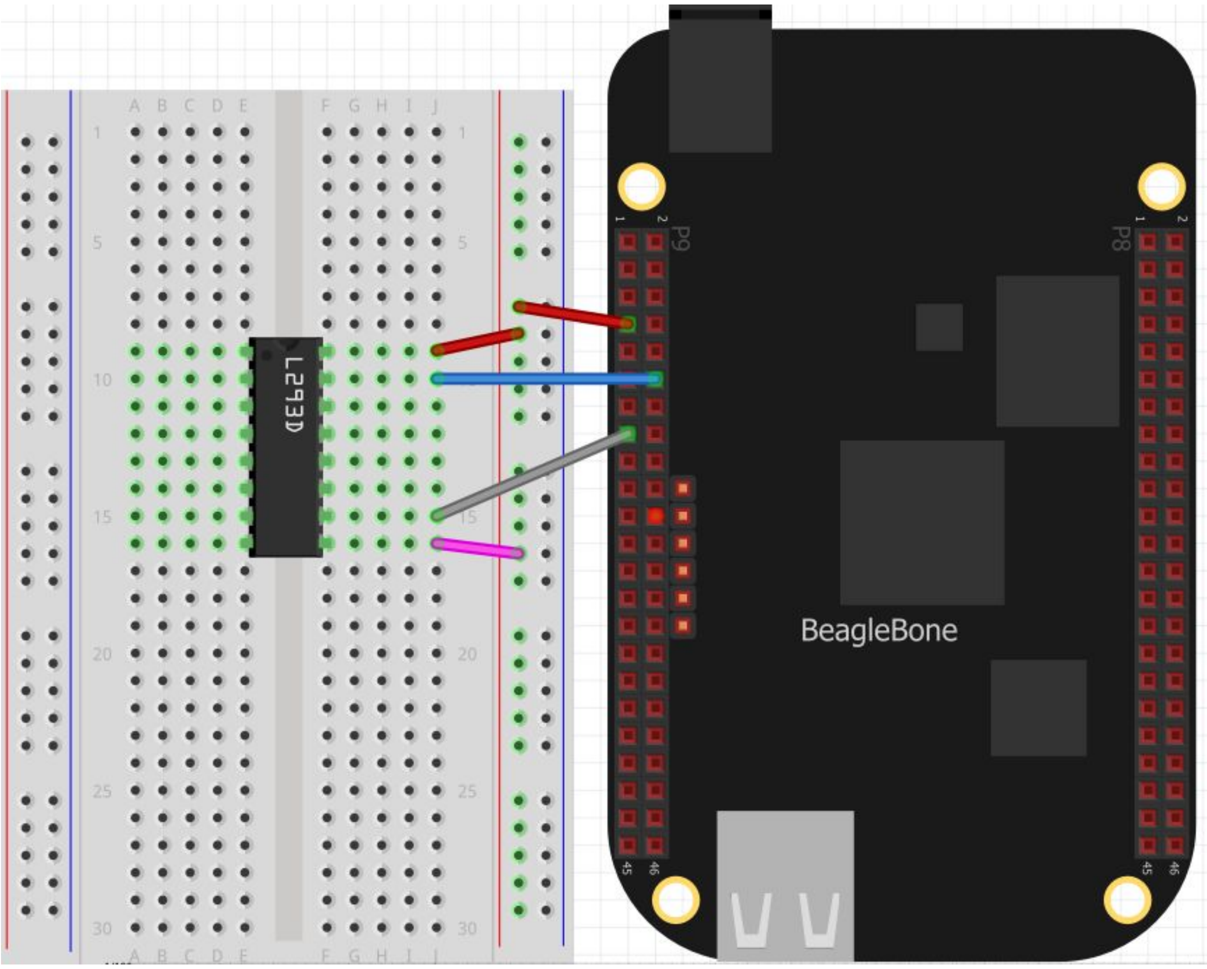
When Enable 3,4(pin9) is low, all input is ignored for inputs 3 and 4 ie pump(s) connected to the right side of the chip are disabled.

This could be useful if you wanted to add a physical 'disable' or 'emergency stop' switch, but we will simply leave it high by connecting it to the BB's 5v and rely on the BB's GPIO output to control the pumps.



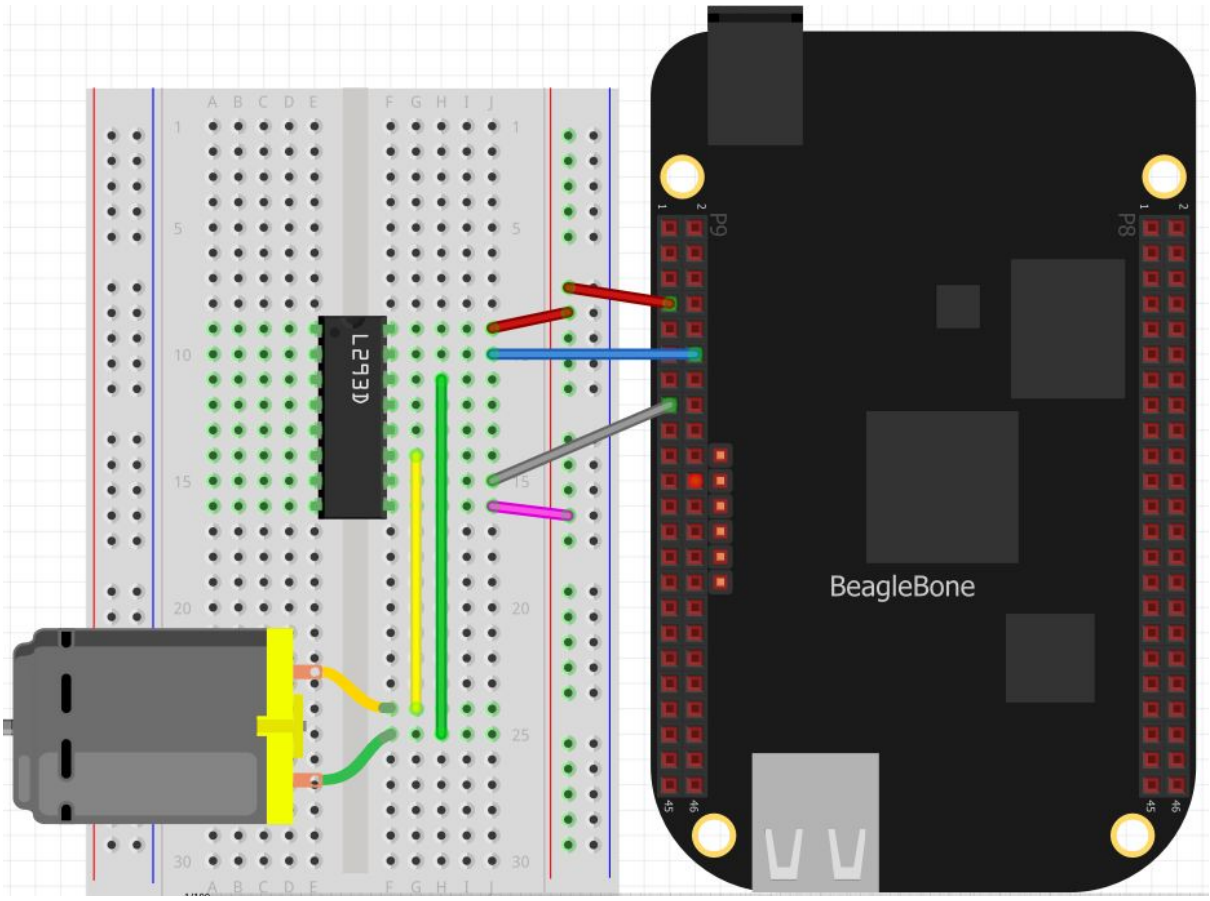
Step 4 (Input3 Input4 / pin10 pin15):

In a forward/reverse motor setup, when Input3(pin10) is set to high and and Input4(pin15) is set low the motor connected to the right side of the L293D (Output3 & Output2) will rotate. If we inverse the input(Input3 low, Input4 high), the motor will rotate in the opposite direction. You can select any of the BB's available GPIO pins to connect to the L293D's Input 3 and Input 4. We will use GPIO_48 and GPIO_60.

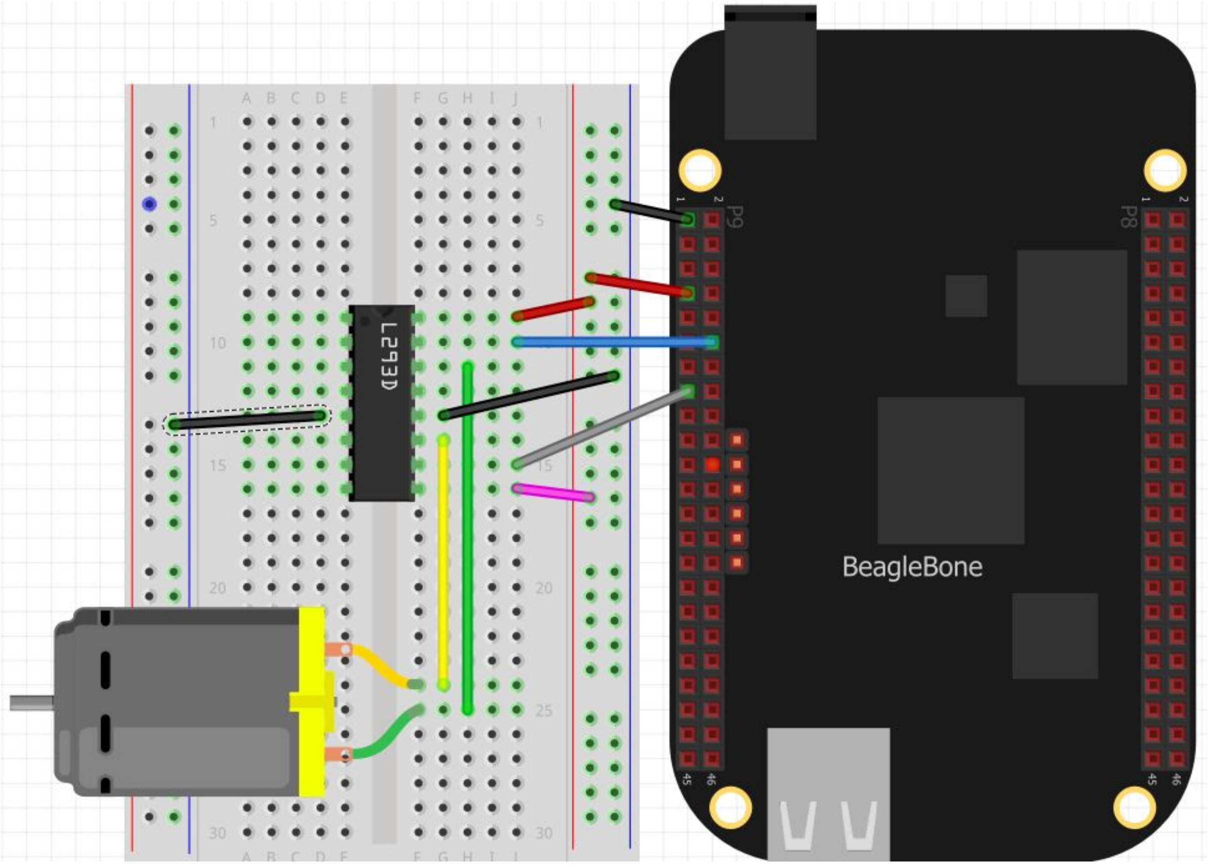


Step 5 (Output3 Output3/ pin11 pin14):

The l293d's output pins each connect to one of your motor wires. If your motor does not spin in the desired direction, simplify reverse these 2 wires(incorrect motor wire orientation will not damage anything).

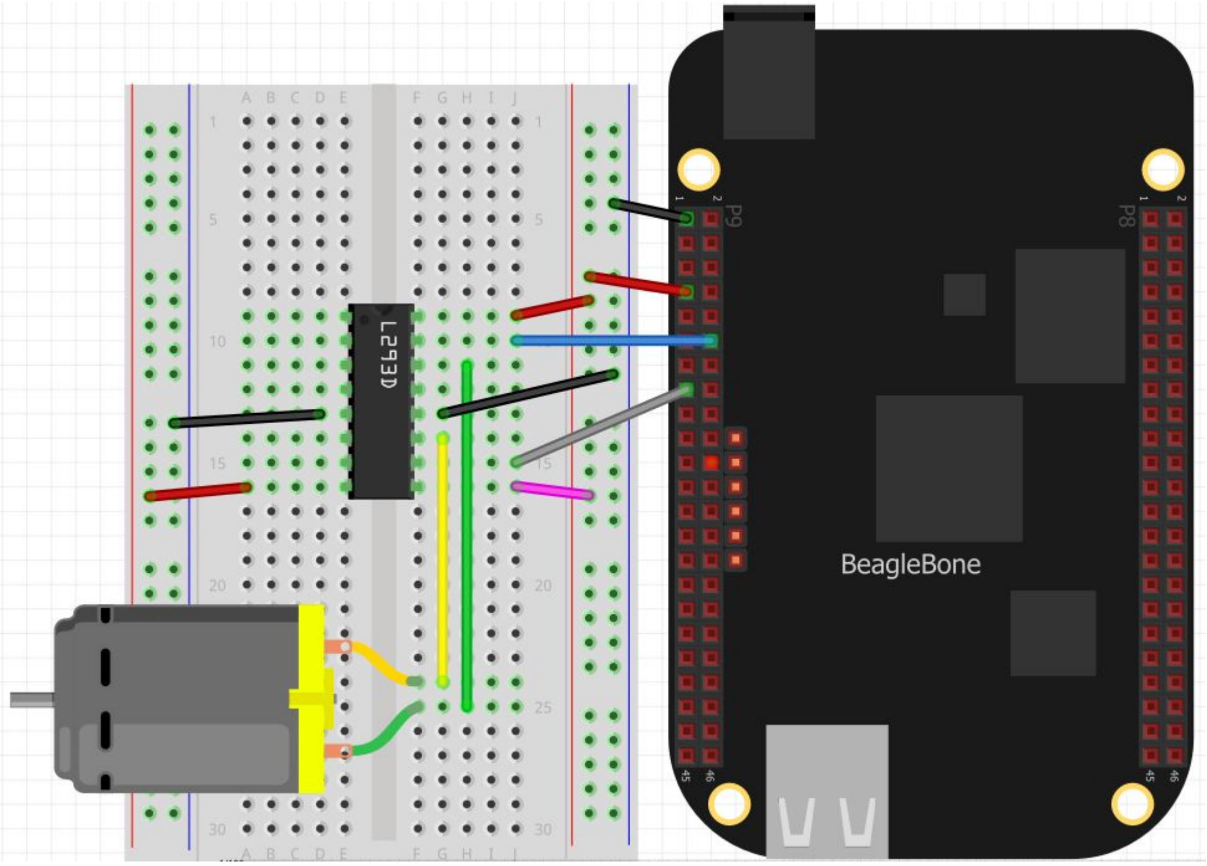


Step 6 (GND / pin4 pin5 pin12 pin13):
The L293D's ground pins are all internally connected(ie plugging into pin 4 is the same as plugging into pin 5 is the same as plugging into pin 13). We will need to connect the L293D's grounds to both the BB's ground and our DC PSU's ground. In our example we will connect L293D's pin 12 to BB's P9_01, and L293D's pin 5 to the left breadboard negative rail(to connect to PSU in later step).



Step 7 (Vcc2 / pin8):

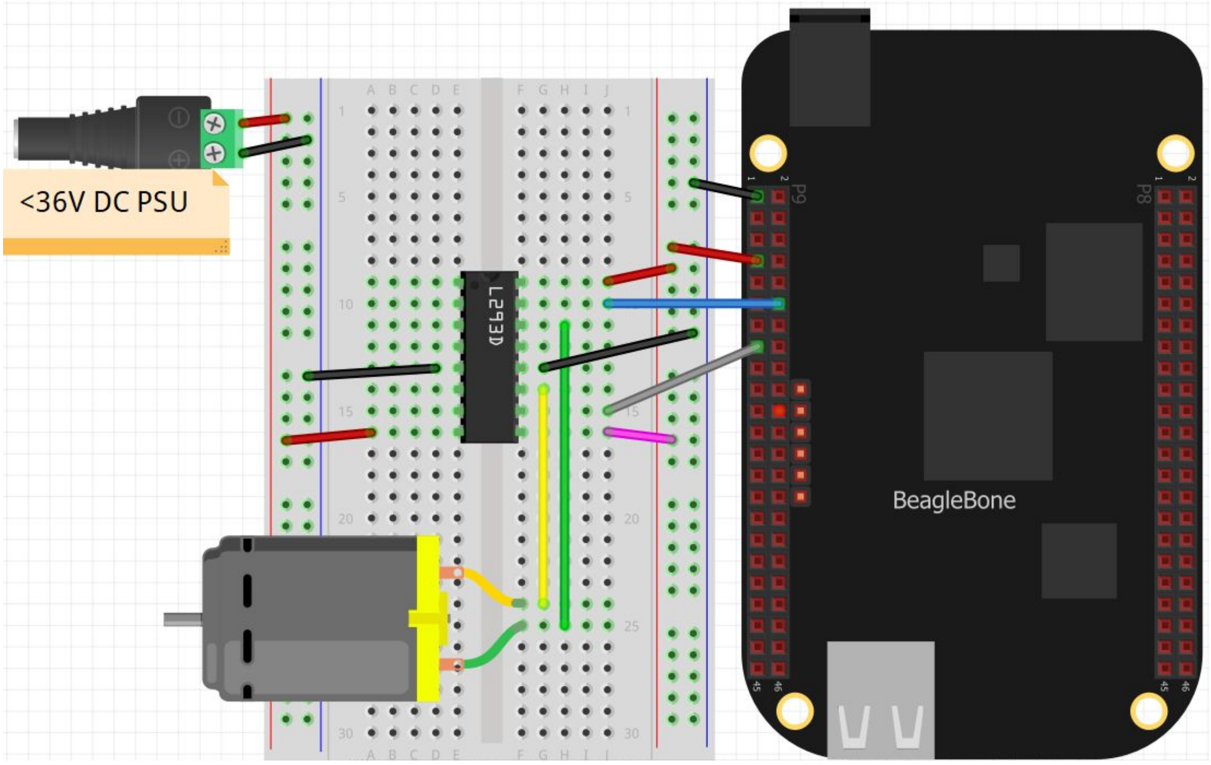
The L293D's Vcc2 pin is for supplying the voltage which will drive the motors. It will be connected to our DC PSU's positive wire in the next step but for now simply connect it to the left positive rail of the breadboard.



Step 8 (DC Power Supply):

Connect the positive wire of the DC PSU to the positive(red line) of the breadboards left power rail.

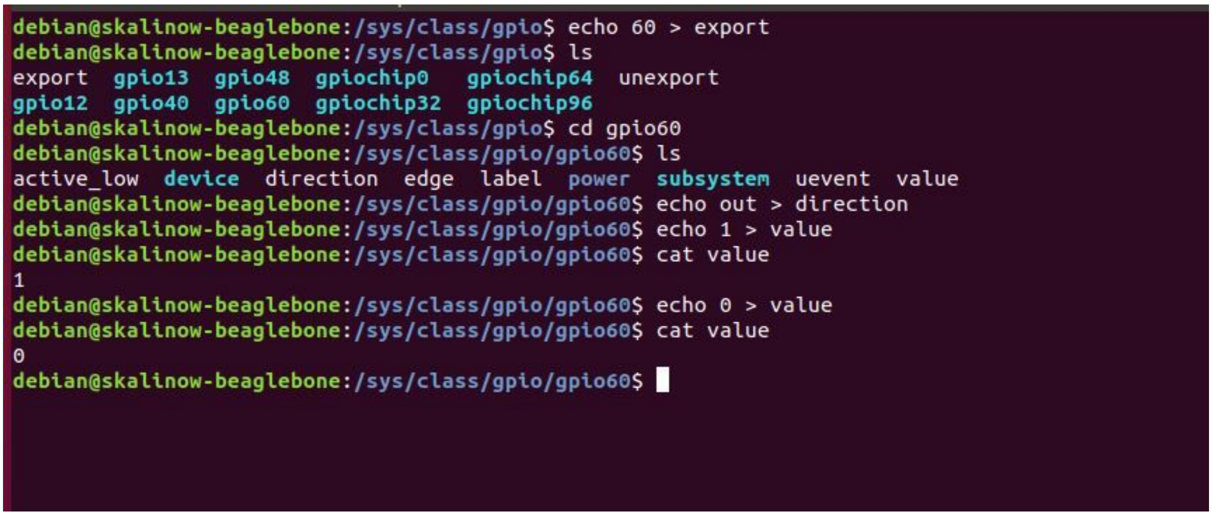
Connect the negative wire of the DC PSU to the negative(black line) of the breadboards left power rail.



Step 9:

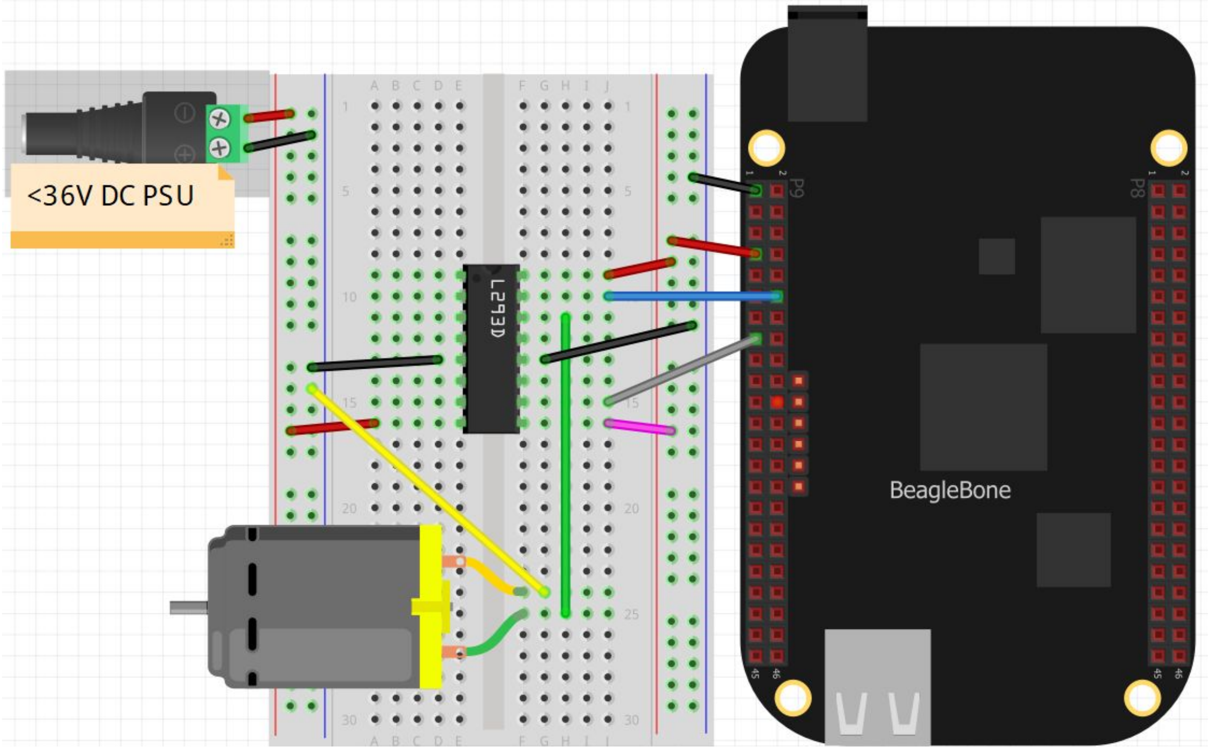
This is the final step. Triple check all wiring if you don't want to free the magic black smoke. Power up DC PSU via wall outlet. Power up BB via USB. Export GPIO pins, set one to high, and the other to low.

If beaglebone won't boot, don't panic(yet). The BB fails to boot if some pins are physically connected(but not exported) at startup. Disconnect the 2 GPIO pins from the BB(leaving the rest of the circuit as it was), boot the BB normally, export the 2 relevant gpio pins, and then reconnect the BB's 2 gpio pins to the circuit.

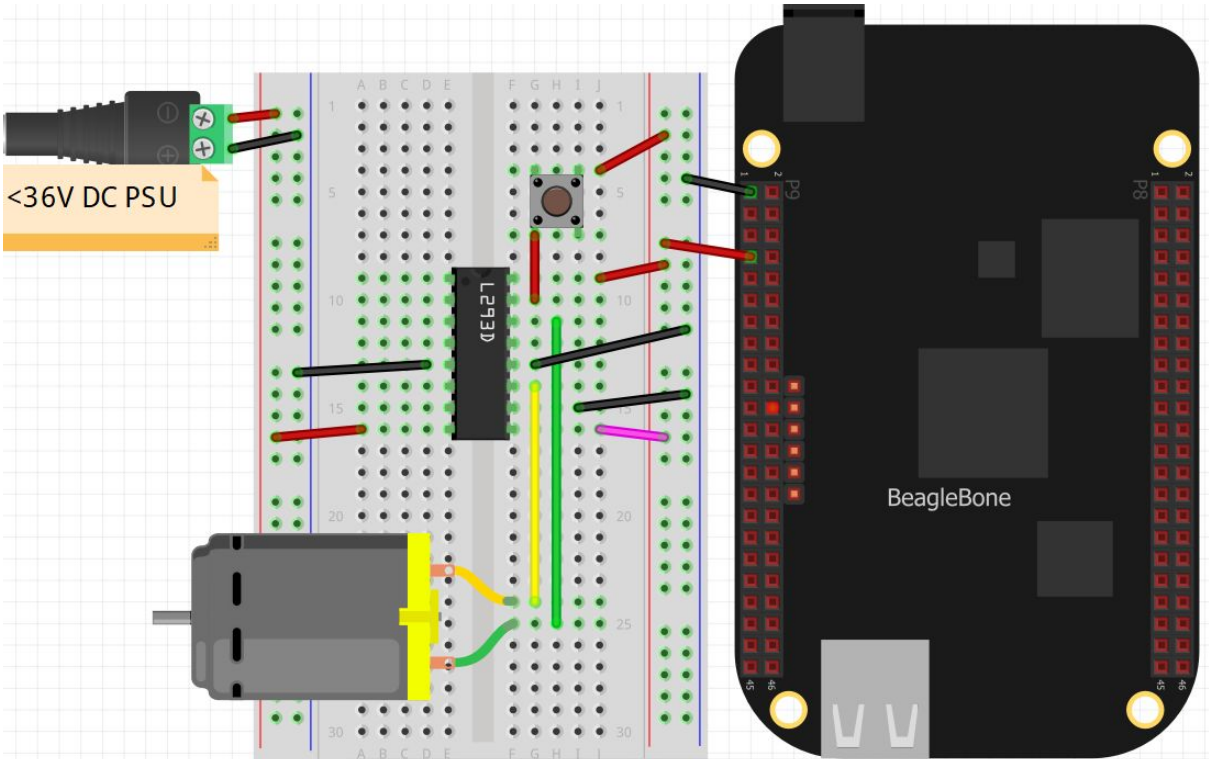


```
debian@skalinow-beaglebone:/sys/class/gpio$ echo 60 > export
debian@skalinow-beaglebone:/sys/class/gpio$ ls
export  gpio13  gpio48  gpiochip0   gpiochip64  unexport
gpio12  gpio40  gpio60  gpiochip32  gpiochip96
debian@skalinow-beaglebone:/sys/class/gpio$ cd gpio60
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ ls
active_low  device  direction  edge  label  power  subsystem  uevent  value
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ echo out > direction
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ echo 1 > value
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ cat value
1
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ echo 0 > value
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ cat value
0
debian@skalinow-beaglebone:/sys/class/gpio/gpio60$ 
```

Single direction mode allows up to 4 motors instead of 2. Connect 1 motor wire to ground.



Testing circuit. Turn motors on/off without beaglebone GPIO [Replace BB 5v/ground with other 5v source for ultimate BB safety]


<36V DC PSU
L293D
BeagleBone
P9
P8