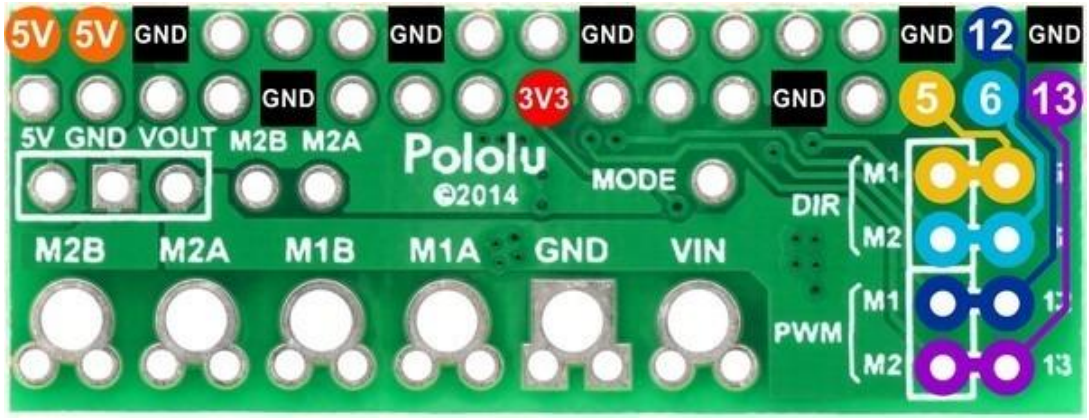
POLOLU DRV8835 DUAL MOTOR DRIVER KIT FOR RASPBERRY PI

USER'S GUIDE

USING THE MOTOR DRIVER

In the board's default state, the motor driver and Raspberry Pi are powered separately, though they share a common ground and the DRV8835 receives its logic supply voltage (VCC) from one of the Raspberry Pi's 3V3 power pins. When used this way, the Raspberry Pi must be powered via its USB Micro-B receptacle, and the motor driver board must be supplied with 1.5 V to 11 V through its large VIN and GND pads. However, the motor driver board provides a set of three through-holes where you can conveniently connect an appropriate voltage regulator, allowing the motor supply to also power the Raspberry Pi (see the *Powering the Raspberry Pi from the motor driver board* section below).



By default, the driver is configured to operate in PHASE/ENABLE mode, in which a PWM signal applied to the ENABLE pin determines motor speed and the digital state of the PHASE pin determines direction of motor rotation. GPIO 12 and 5 are used to control the speed and direction, respectively, of motor 1, and GPIO 13 and 6 control the speed and direction of motor 2. The table below shows how the inputs affect the outputs in this mode:

Drive/brake operation in default PHASE/ENABLE mode				
xPHASE	xENABLE	MxA	MxB	operating mode
1	PWM	L	PWM	reverse/brake at speed *PWM %*
0	PWM	PWM	L	forward/brake at speed *PWM %*
X	0	L	L	brake low (outputs shorted to ground)

PHASE/ENABLE mode should be suitable for most applications.

CONFIGURING THE BOARD FOR IN/IN MODE

The operating mode of the driver is controlled by the MODE pin, which the board pulls high to VCC through a 20 kΩ resistor to select PHASE/ENABLE mode by default. The pin labeled "MODE" can be driven low (or connected directly to ground) to switch the control interface to IN/IN, which allows for slightly more advanced control options as described in the table below:

Drive/coast or drive/brake operation with MODE=0 (IN/IN)				
xIN1	xIN2	MxA	MxB	operating mode
0	0	OPEN	OPEN	coast (outputs off)
0	PWM	L	PWM	reverse/coast at speed *PWM %*
PWM	0	PWM	L	forward/coast at speed *PWM %*
PWM	1	L	PWM	reverse/brake at speed *100% – PWM %*
1	PWM	PWM	L	forward/brake at speed *100% – PWM %*
1	1	L	L	brake low (outputs shorted to ground)

IN/IN mode is generally only useful if you only care about on/off control of the motors or if you can supply PWM signals to all four inputs. Since the Raspberry Pi Model B+ only has two hardware PWM outputs, additional work (such as setting up software PWM) is necessary to achieve speed control with IN/IN mode.

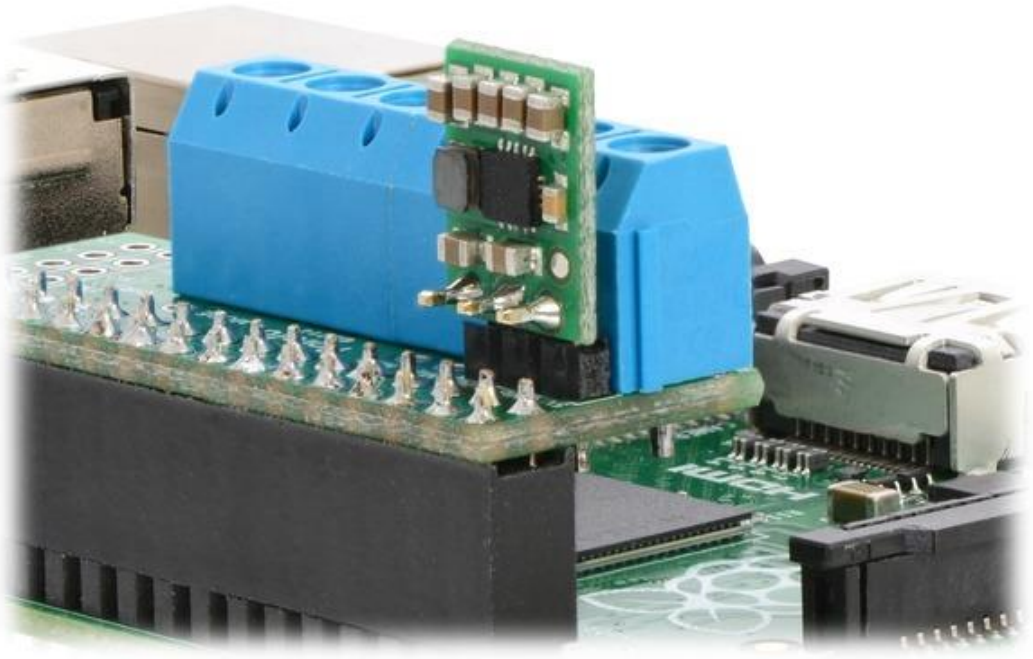
CONFIGURING THE BOARD FOR SINGLE-CHANNEL MODE (PARALLEL OUTPUTS)

In order to use the two motor channels in parallel to control a single motor, it is important to ensure that both channels will always receive the same control signals, so the reconfiguration process begins with a modification to the control inputs. First, locate the 2×4 grouping of 0.1″ through-holes along the right side of the board. The traces on the underside of the PCB between each pair of holes effectively link the Raspberry Pi's GPIO pins to the DRV8835 control pins. If you want to remap one of these control pins, you can cut the desired trace with a knife and then run a wire from the inner hole to a new GPIO pin. The remapping for single-channel mode requires you cut one PWM (12 or 13) and one DIR (5 or 6) trace. If you then solder a row of header pins along the interior row of holes, you can safely connect both PWM lines together and both DIR lines together using shorting blocks. In this configuration, the two uncut Raspberry Pi control lines determine the behavior of both motor channels.

The last step is to connect the output channels together. An easy way to do this is to solder wires to the two small holes labeled "M2A" and "M2B" above the motor outputs. You can then connect the M2A wire to the large M1A output pad and the M2B wire to the large M1B output pad, which in turn means you can get up to 3 A from the connection points for M2 (you can have your motor connected just to the M2A and M2B terminal blocks rather than trying to find a way to connect it to all four motor outputs).

POWERING THE RASPBERRY PI FROM THE MOTOR DRIVER BOARD

On the left side of the expansion board is a set of three pins surrounded by a box and labeled “5V”, “GND”, and “VOUT”. The “5V” pin is connected to the Raspberry Pi’s 5V power rail, while the VOUT pin provides access to the driver board’s motor supply voltage after reverse-voltage protection. If a suitable voltage regulator is connected to these three pins, it can generate 5 V to power the Raspberry Pi from the board’s motor supply voltage. We suggest using our S7V7F5 switching step-up/step-down regulator, which has a 2.7 V to 11.8 V input voltage range (similar to that of the DRV8835) and can supply up to 1 A of current to the Raspberry Pi.



S7V7F5 step-up/step-down regulator connected to an assembled Pololu DRV8835 Dual Motor Driver Kit for Raspberry Pi.

When adding a voltage regulator to the motor driver board, take care to orient it correctly: note that the *motor driver board’s* VOUT pin should connect to the *regulator’s* VIN pin, while the *regulator’s* VOUT pin should connect to the *motor driver board’s* 5V pin.

There are several considerations to keep in mind when powering the Raspberry Pi through a voltage regulator in this way:

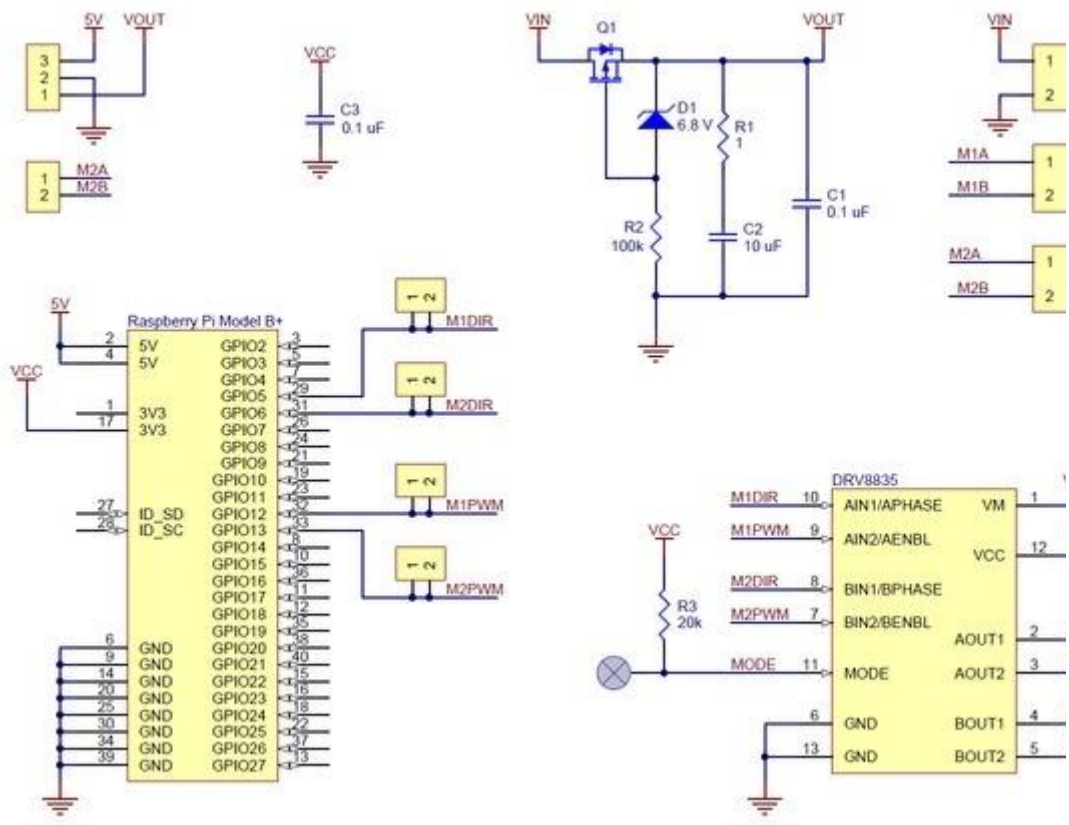
- You must **never** connect a different power supply to the Raspberry Pi (including through its USB Micro-B receptacle) while the regulator is connected, as doing so will create a short between the voltage regulator's output and the external power supply that could permanently damage the Raspberry Pi, the regulator, and/or the power supply.
- Your motor power supply must be an acceptable voltage for both your regulator and the DRV8835.
- The regulator should be able to handle the power requirements of the Raspberry Pi. The Raspberry Pi typically uses a few hundred milliamps at 5 V (depending on the specific model), although its current draw can exceed 1 A if it is also supplying power to USB devices and other peripherals. While linear regulators like a 7805 might fit in the regulator mounting location, they could generate excessive heat or shut down at higher input voltages and output currents. We recommend using a switching regulator (like our S7V7F5, as mentioned above).
-

REAL-WORLD POWER DISSIPATION CONSIDERATIONS

The DRV8835 datasheet recommends a maximum continuous current of 1.5 A per motor channel. However, the chip by itself will overheat at lower currents. For example, in our tests at room temperature with no forced air flow, the chip was able to deliver 1.5 A per channel for approximately 15 seconds before the chip's thermal protection kicked in and disabled the motor outputs, while a continuous current of 1.2 A per channel was sustainable for many minutes without triggering a thermal shutdown. The actual current you can deliver will depend on how well you can keep the motor driver cool. Our tests were conducted at 100% duty cycle; PWMing the motor will introduce additional heating proportional to the frequency.

This product can get hot enough to burn you long before the chip overheats. Take care when handling this product and other components connected to it.

SCHEMATIC DIAGRAM



Pololu DRV8835 Dual Motor Driver Kit for Raspberry Pi schematic diagram.