POLOLU A4990 DUAL MOTOR DRIVER SHIELD FOR ARDUINO

USER'S GUIDE

USING THE SHIELD

The shield plugs into Arduino digital pins 6, 7, 8, 9, and 10 on one side and Arduino VIN, GND, GND, and 5V/VCC on the other. The upper-left corner of the shield partially blocks the Arduino's 3.3V pin, but this region of the board (marked with a white silkscreen box) can be removed if necessary to allow access.

In the shield's default state, the motor driver shield and Arduino are powered separately, though they share a common ground and the Arduino's 5V rail serves as the shield's logic supply. When used this way, the Arduino must be powered via USB, its power jack, or its VIN pin, and the shield must be supplied with 6 V to 32 V through its large VIN and GND pads. Attempting to power the shield from the Arduino is not recommended as this could result in large currents flowing through small traces. However, if the motor power supply is suitable, it is possible to power the Arduino from the shield. This can be accomplished by placing a jumper between the shield pins in the lower-left corner labeled VOUT and AVIN, which connects the reverse-protected motor supply voltage to the Arduino's VIN pin to power the Arduino. The Arduino's power jack must remain disconnected at all times in this configuration.

Warning: When powering the Arduino from the motor shield, you must **never** connect a different power supply to the Arduino's VIN pin or plug a power supply into the Arduino's power jack, as doing so will create a short between the shield's power supply and the Arduino's power supply that could permanently damage both the Arduino and the motor shield. In this case, it is also important that your shield power supply is an acceptable voltage for your Arduino, so the full shield operating voltage range of 6 V to 32 V probably will not be

available. For example, the recommended operating voltage of the Arduino Uno is 7 – 12 V.

The shield has integrated logic gates that simplify the control interface of the A4990 by reducing the number of PWM signals required. Each channel has a speed control input, MxPWM, and a direction control input, MxDIR. Arduino pins 9 and 7 are used to control the speed and direction, respectively, of motor 1, and pins 10 and 8 control the speed and direction of motor 2. The following truth table shows how the shield operates:

MxDIR	MxPWM	MxA	MxB	operating mode
0	PWM	PWM	L	forward/brake at speed *PWM %*
1	PWM	L	PWM	reverse/brake at speed *PWM %*
X	0	L	L	brake low (outputs shorted to ground)

The A4990 has two diagnostic pins, EF1 and EF2, that offer feedback about the state of the driver. These pins are open-drain outputs that are driven low by the chip to indicate faults (the datasheet describes what each combination of EF1 and EF2 means). Otherwise, these pins remain in a floating state. By default, EF2 is connected to Arduino digital pin 6 through a 1 kΩ protection resistor; with pin 6 configured as an input with its internal pull-up enabled, a low signal indicates an over-current, over-voltage, or over-temperature condition.

The EF1 and EF2 pins can more generally accessed via the 3-pin row of through-holes in the lower-left corner of the board. If you do not want to monitor the motor driver fault flags or would rather free up pin 6 for some other purpose, you can cut the trace that connects the EF2 pin to its neighboring (center) pin on the bottom side of the shield.

CURRENT LIMITING

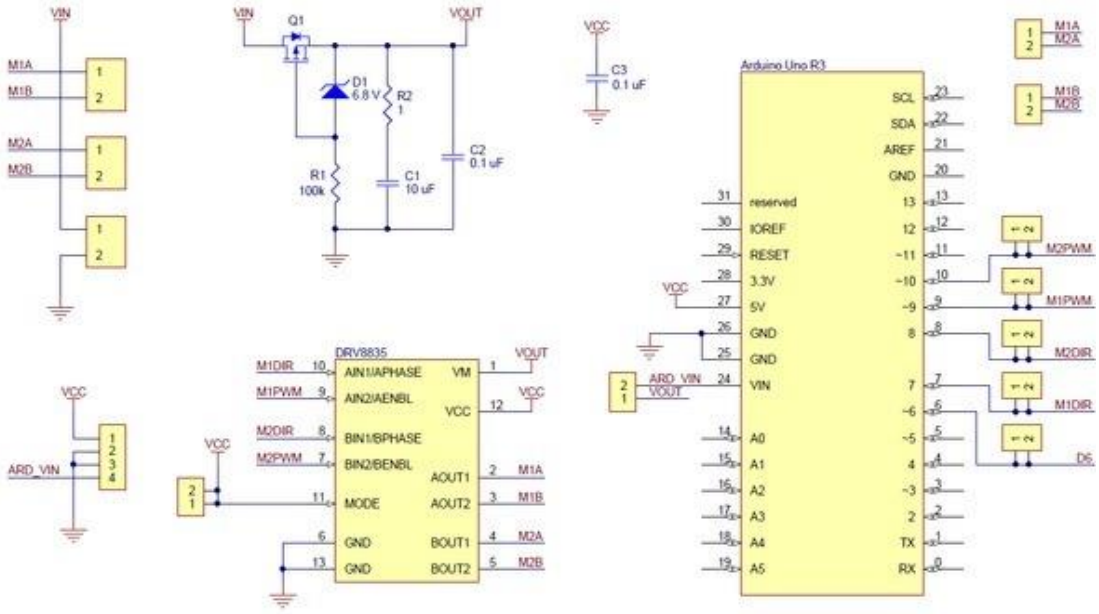
The A4990 can actively limit the current through the motors by using a fixed-frequency PWM current regulation (current chopping). This carrier board connects 0.075 Ω resistors to the current sense pins, which sets the current limit to a nominal 1 A per channel. In our tests, the board actually limited the motor current to slightly above 0.9 A.

REAL-WORLD POWER DISSIPATION CONSIDERATIONS

Even though the driver limits the motor current to about 0.9 A per channel, 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 0.9 A per channel for approximately 15 seconds before the chip's thermal protection kicked in. A continuous current of 0.65 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 inputs will introduce additional heating proportional to the frequency (unless the A4990 is already PWMing the outputs to limit the current).

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 Shield for Arduino schematic diagram.