Turbocharger DC Motor Driver

Features and Ratings

- Dual DC motor driver
  - ST-L298P Quad H-Bridge Motor Driver
  - Up to 48V motors and up to 2A per motor.
  - Up to 4A if both channels are used together
  - If 1A or less current is used, no heat sink or forced air cooling necessary
- Dual Quadrature Encoder Inputs
  - LS/CSI LS7366R-S Serial Encoder Counter from US Digital
  - 32-bit reconfigurable quadrature encoder counters
  - SPI enabled to reduce microcontroller utilization
- Designed for Arduino Uno
  - All communications are designed for 5V. A 3.3V board may be used if the pins are 5V tolerant. MISO (digital pin 12) is a 5V output from the encoder counters and will damage a 3.3V board.

Caution:

Before making connections:

1. Ensure your power supply is **ABLE** to have the Negative DC Output connected to the common ground for the system. There are some power supplies in the wild that create a neutral->ground fault when the negative output from the supply is connected to ground. The driver will force a common ground to all supplies.
2. Add an appropriate sized fuse between the positive terminal of the power supply and the driver board. The driver is able to handle 3A for 100us, more than this will destroy the driver. Ask for help in fuse sizing if you are inexperienced.
3. If the Motor voltage is greater than 12V, make sure the Vin_enable jumper is cleared to prevent damage to the Arduino.

Finally, check all connectors for potential shorts before applying power.

Pin Connections to Arduino

Motor Driver Arduino Pin-out

Description	Arduino Pin Number
Channel A Direction	7
Channel A PWM	9
Channel A Current Sense	A3
Channel B Direction	8
Channel B PWM	10
Channel B Current Sense	A2

Quadrature Encoder Arduino Pin-out

Description	Arduino Pin Number
MOSI	11
MISO	12
SCK	13
Encoder Counter 1 Select	4
Encoder Counter 2 Select	5
Counter Enable	6
!LFLAG with pull-up resistor	2

Pins 11-13 are SPI for the Arduino
and are handled in the SPI Library
Do not use the pins

Pin Connections to Peripherals

Headers for easier connection to communication buses and other microcontroller functions have been provided for your convince. These headers include Two Wire Interface (TWI; also known as I2C), Serial Peripheral Interface (SPI), Universal Asynchronous Receiver/Transmitter (UART), and finally two analog input enabled pins that are available in servo-standard pin-outs.

SPI

The SPI communication bus can support one additional peripheral (or more if you are experienced). The SPI bus is shared with the LS7366R-S Quadrature Encoder Counters. Digital Pin 3 has been made available for use as a chip select pin for this peripheral and is available on this header. Please note that pin 3 is arranged next to Vdd and Ground such that pin 3 can also be used apart from SPI with a servo-standard pin-out. This was done such that pin 3, which is an interrupt-enabled pin on the Arduino Uno, can be used for am interrupt-enabled limit switch.

UART

The UART communication bus is broken out above the SPI bus on the board. This is the hardware UART for the Arduino with no flow control or reset. This can be used if a peripheral needs Serial communications and the UART is not being used to talk to the computer. When programming through the USB port peripherals connected to this header must be disconnected.

TWI

PLEASE NOTE: SDA and SCL on the TWI header are labelled incorrectly and should be swapped.

The TWI has been made available for users who need the I2C communication protocol. On R3 Arduino Unos these pins are available on the digital side, but on R2 Arduino Unos these pins are only available with the analog input pins. In the interests of keeping this board inclusive, the TWI header is connected to Analog Pins A4 and A5. With this, if additional analog pins past A0 and A1 are needed and no I2C is needed, these pins can be directly used as analog inputs. Also, since all analog input pins are digital enabled as well, these pins can serve digital functions too.

A0 & A1

A0 and A1 have been made available in servo-standard pin-outs. These pins were placed for easy addition of potentiometer feedback in case encoders are not needed or for easier use input. Additionally, these pins can be used for hobby servos, buttons, limit switches, or other low-pin count needs.

Motor and Encoder Connections

Motor Connections

The DC motors can be connected to the screw terminal blocks with 18-22AWG wire. Wire outside this size range may fit but will result in a poor connection. The channels can be connected in parallel for higher current capacity. The motor power inputs are designed with Vin and Ground where ground is common to the Arduino’s ground. **Ensure your power supply can handle the DC negative output connected to earth ground.**

Encoder Connections

The encoder headers as labeled on the circuit board.

Description	PCB Label	Required
5V Encoder Power Supply	V	Yes
Encoder Channel A Output	A	Yes
Encoder Channel B Output	B	For quadrature applications
Encoder Index Output	I	For high repeatability
Encoder Power Supply Ground	G	Yes

Default Jumper Configuration

At assembly the board is configured such that both current sense channels are connected to their respective pins and the Vin Enable Jumper is opened. If you application requires more pins than were made available in the auxiliary headers (UART, SPI, TWI, A1, A0), you can cut the trace on the bottom of the board between the solder jumper pads to disable one or both current sense channels for an additional two analog input enabled pins. If your motor power supply is between 7-12V, the Vin Enable jumper can be closed by bridging the solder pads and powering the Arduino from the motor power supply. **Ensure this jumper is cleared before applying >12V to the motor Vin** otherwise the Arduino's power regulator will fail.

Motor Driver Usage

The Motor Drivers are designed with2 pins to control each motor and one feedback pin for current sensing.

The Direction pin controls the direction the motor will turn and can be reversed by reversing the polarity of motor's wires to the driver. The direction is set using the digitalWrite function.

The PWM output controls the motor's speed and is controlled through the analogWrite function.

The current sensing is used through the analogRead function on the current sensing pin. The operational amplifier is set to a gain of 15:1 across the 0.15Ω sense resistor. This gives an output of 2.25V/1A of current through the motor. This gives an ratio of 460.35/A on the analog inputs. So an analog reading of 460 is about 1A and an analog reading of 920 is about 2A. See the attached MotorTest.ino file for an example of motor usage

Encoder Counter Usage

The encoder counters are connected through SPI to the Arduino and can be accessed by pulling their respective chip select pins low. There is also a count enable pin that can be pulled low to disable the pulse counting functionality of the encoders. This would be used during setup while initializing everything then the counters would be enabled once the system is ready for them. A few functions were made to make usage of the encoder counters easier. A basic library is in the works but is not yet ready for external release. Please see the code called EncoderTest.ino to see an example of the encoder counters in use.