MD22 - Dual 24Volt 5Amp H Bridge Motor Drive

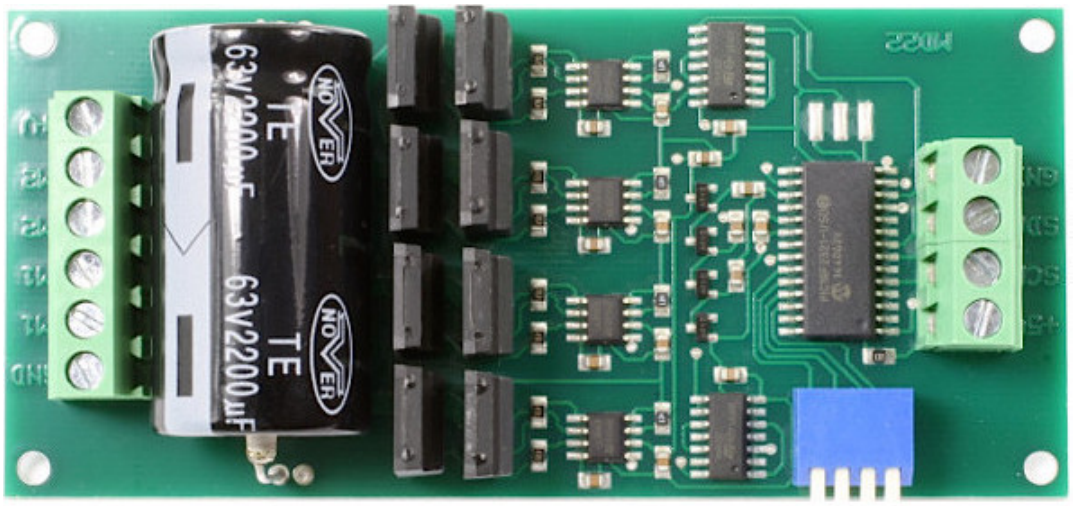
Overview

The MD22 is a robust low/medium power motor driver, designed to supply power for two motors. Main features are:

1. Drives two motors with independent control.
2. Ease of use and flexibility.
3. The 15v MOSFET drive voltage is generated onboard with a charge pump, so the module requires only two supply voltages;
   a) A standard 5V supply for the control logic, only 50mA maximum is required.
   b) The H-bridge has a rating of 60v allowing motor voltages up to 24vdc.
4. Steering feature, motors can be commanded to turn by I2C register or input (analogue + servo).
5. Control of the module can be any of;
   a) I2C bus, up to 8 MD22 modules, switch selectable addresses and 4 modes of operation including steering..
   b) 2 independent 0v-2.5v-5v analog inputs. 0v full reverse, 2.5v center stop, 5v full forward.
   c) 0v-2.5v-5v analog input for speed ,with the other channel for steering.
   d) independent channel RC mode. Motors are individually Controlled directly from the RC receiver output.
   e) RC mode with steering, allows speed control with one stick of radio control, and steering with the other.
6. Uses high current MOSFETs, making a very robust module.

MD22 Connections

Motor +v
Motor 2
Motor 2
Motor 1
Motor 1
Motor GND



Logic GND
SDA/servo/analogue
SCL/servo/analogue
Logic +5v

There is no fuse on the PCB. You should provide a fuse in line with the +24v terminal.

The motor ground and the logic ground are internally connected on the module. Be sure to use cable rated for at least 10A for the battery, fuse and motor leads.

Motor Noise Suppression

Please note that using motors with the MD22 as with any other electronic device requires suppression of noise. This is easily achieved by the addition of a 10n snubbing capacitor across the motors. The capacitor should also be capable of handling a voltage of twice the drive voltage to the motor.

Mode Switches

The 4 mode switches set the operating mode of the MD22. They are read once only when the module is powered up. You cannot switch modes while the unit is on.

Mode	Switch 1	Switch 2	Switch 3	Switch 4
I2C Bus - address 0xB0	On	On	On	On
I2C Bus - address 0xB2	Off	On	On	On
I2C Bus - address 0xB4	On	Off	On	On
I2C Bus - address 0xB6	Off	Off	On	On
I2C Bus - address 0xB8	On	On	Off	On
I2C Bus - address 0xBA	Off	On	Off	On
I2C Bus - address 0xBC	On	Off	Off	On
I2C Bus - address 0xBE	Off	Off	Off	On
0v - 2.5v - 5v Analog	On	On	On	Off
0v - 2.5v - 5v Analog + turn	Off	On	On	Off
RC Servo	On	Off	On	Off
RC Servo + turn	Off	Off	On	Off

New modes from version 3(Dec 2004)

RC Servo, timeout on	On	On	Off	Off
RC Servo + turn, timeout on	Off	On	Off	Off

New modes from version 9(Mar 2006)

Analogue turn mode 2	On	Off	Off	Off
RC Servo, turn mode 2, timeout on	Off	Off	Off	Off

Note that I2C addresses are the upper 7 bits. Bit 0 the the read/write bit, so addresses 0xB0/0xB1 are write/read respectively to the same address.
This range of I2C addresses is the same as those used by the MD03.

Analog mode - 0v-2.5v-5v
In this mode the motors are controlled independently by two 0v to 5v analog signal on the SCL (Motor1) and SDA (Motor2) lines.
0v is maximum reverse power
2.5v is the center stop position
5v is full forward power
There is a small (2.7%) dead band around 2.5v to provide a stable off position.

Analog mode - 0v-2.5v-5v with differential drive
Both motors speed is now controlled by the analogue voltage level on the SCL line. The SDA line is now responsible for offsetting the two speeds and thus controlling the degree of turn.
The voltage levels are the same as above but turn degree is:
0v is hard turn left
2.5v is the straight position
5v is hard turn right
There is the same dead band (2.7%) on the speed and the turn. .

RC servo mode
This mode allows direct connection to standard model radio control receivers. Most receivers work from a 4.8v-6v battery pack and can be powered by 5v supply that powers the MD22 logic. The control pulses (Yellow) from the receiver should be connected to the SCL (Motor1) and SDA (Motor2) terminals. Connect the receiver supply (Red) to +5v logic supply and the receiver 0v ground (Black) to the MD22 logic ground. The output from an RC receiver is a high pulse 1.5mS wide when the joystick is central. The MD22 provides full control in the range 1mS to 2mS with 1.5mS being the center off position. There is a 7uS dead zone centered on 1.5mS for the off position. The Radio Transmitter centering control should be adjusted so that the motor is off when the joystick is released.

RC servo mode with differential drive
Again uses a standard radio control receiver module output to determine speed with the addition of the extremely useful steering function. The receivers forward and reverse channel should be wired to the SCL connection and the steering (turn) through the SDA channel. Again fine adjustment to the transmitters offset may possibly be needed.

RC modes with timeout feature (from version 3)
An extra couple of modes have been added and operate in much the same way as the normal servo control. The difference is the addition of a new timeout feature. If the RC pulse is not detected on a channel 1 (SCL) for a period in excess of 200ms, then both of the motors will be stop being driven until a valid RC signal is received on channel 1.

I2C mode
I2C mode allows the MD22 to be connected to popular controllers such as the PICAXE, OOPic and BS2p, and a wide range of micro-controllers like PIC's, AVR's, 8051's etc.
I2C communication protocol with the MD22 module is the same as popular eeprom's such as the 24C04. To read one or more of the MD22 registers, first send a start bit, the module address (0XB0 for example - see mode switches) with the read/write bit low, then the register number you wish to read. This is followed by a repeated start and the module address again with the read/write bit high (0XB1 in this example). You are now able to read one or more registers. The MD22 has 8 registers numbered 0 to 7 as follows;

Register Address	Name	Read/Write	Description
0	Mode	R/W	Mode of operation (see below)
1	Speed	R/W	Left motor speed (mode 0,1) or speed (mode 2,3)
2	Speed2/Turn	R/W	Right motor speed (mode 0,1) or turn (mode 2,3)
3	Acceleration	R/W	Acceleration for i2c (mode 0,1)
4	Unused	Read only	Read as zero
5	Unused	Read only	Read as zero
6	Unused	Read only	Read as zero
7	Software Revision	Read only	Software Revision Number

The mode register defaults to 0, as does the acceleration register (fastest acceleration). No motor will move until directly after speed or speed2/turn registers are changed.

Mode Register
The mode register selects which mode of operation and I2C data input type the user requires. The options being:

0, (default setting) If a value of 0 is written to the mode register then the meaning of the speed registers is literal speeds in the range of:
0 (full reverse) 128 (stop) 255 (full forward).

1, Mode 1 is similar to Mode 0, except that the speed registers are interpreted as signed values. The meaning of the speed registers is literal speeds in the range of:
-128 (full reverse) 0 (stop) 127 (full forward).

2, Writing a value of 2 to the mode register will make speed control both motors speed. speed2 then becomes the turn value (type 1).
Data is in the range of 0 (full reverse) 128 (stop) 255 (full forward).
note - version 8+ speed controls the total power, the turn (speed 2) value is now with reference to this.

3, Mode 3 is similar to mode 2, except that the speed registers are interpreted as signed values.
Data is in the range of -128 (full reverse) 0 (stop) 127 (full forward)
note - version 8+ speed controls the total power, the turn (speed 2) value is now with reference to this.

4, (new from version 9) Alternate method of turning (type 2), the turn value being able to introduce power to the system.
Data is in the range of 0 (full reverse) 128 (stop) 255 (full forward).

5, (New from version 9) Alternate method of turning (type 2), the turn value being able to introduce power to the system. Data is in the range of -128 (full reverse) 0 (stop) 127 (full forward)

Speed register
Depending on what mode you are in, this register can affect the speed of one motor or both motors. If you are in mode 0 or 1 it will Set the speed of the motor 1. The larger the number written to this register, the more power is applied to the motor. If mode is set to a turn mode it controls the speed and direction of both motors (subject to effect of turn register).

Speed2/Turn register
Again when in mode 0 or 1 this register operates the same as speed but controls the operation of the motor 2. When a turn mode is selected Speed2 becomes a turn register, and any value in speed 1 is combined with the contents of this register to steer the device.

Turn mode (up to version 7)

In software versions up to 7, the turn modes look at the speed channel or register to decide if the direction is forward or reverse. They then apply a subtraction or addition of the turn value on either motor.

so if the direction is forward
motor speed1 = speed - turn
motor speed2 = speed + turn

else the direction is reverse so
motor speed1 = speed + turn
motor speed2 = speed - turn

If the either motor is not able to achieve the required speed for the turn (beyond the maximum output), then the other motor is automatically changed by the program to meet the required difference.

Turn mode (version 8+)

In turn mode 1 the power supplied to the motors is always with reference to the speed. Turn values are only applied with respect to the speed. The turn factor is determined by the equation below, where speed_max is a program constant equating to the maximum possible motor speed.
turn factor = turn * (speed/speed_max)

And now the power to the motors can be calculated, remembering that a turn in either direction in a forward direction is the inverse in the reverse direction so:

if we are moving forwards and require a turn then
motor speed1 = speed - turn factor
motor speed2 = speed + turn factor

else if we are moving in reverse and require a turn then
motor speed1 = speed + turn factor
motor speed2 = speed - turn factor

If the either motor is not able to achieve the required speed for the turn (beyond the maximum output), then the other motor is automatically changed by the program to meet the required difference.

Turn mode 2 (version 9+)

In turn mode 2 there is no link between the turn factor and the speed, this means when speed is at zero you can still turn. With this method is the backwards turns are inverted (left is right). The turn factor is now just :

turn factor = turn

And the power to the motor is now :

motor speed1 = speed - turn factor
motor speed2 = speed + turn factor

If the either motor is not able to achieve the required speed for the turn (beyond the maximum output), then the other motor is automatically changed by the program to meet the required difference.

Acceleration register (in direct motor control)
If you require a controlled acceleration period for the attached motors to reach there ultimate speed, the MD22 has a register to provide this. It works by inputting a value into the acceleration register which acts as a delay in the power stepping. The amount of steps is the difference between the current speed of the motors and the new speed (from speed 1 and 2 registers). So if the motors were traveling at full speed in the forward direction (255) and were instructed to move at full speed in reverse (0), there would be 255 steps.

The acceleration register contains the rate at which the motor board moves through the steps. At 0 (default) the board changes the power (accelerates) at its fastest rate, each step taking 64us. When the acceleration register is loaded with the Slowest setting of 255, the board will change the power output every 16.4ms.

So to calculate the time (in seconds) for the acceleration to complete :
time = accel reg value * 64us * steps.
For example :

Accel reg	Time/step	Current speed	New speed	Steps	Acceleration time
0	0	0	255	255	0
20	1.28ms	127	255	128	164ms
50	3.2ms	80	0	80	256ms
100	6.4ms	45	7	38	243ms
150	9.6ms	255	5	250	2.4s
200	12.8ms	127	0	127	1.63s
255	16.32ms	65	150	85	1.39s

Software revision number
This register contains the revision number of the software in the modules PIC controller.

Using the MD22 with popular controllers
One the easiest ways of connecting the MD22 to a standard controller, such as the BS2 Stamp, is to use RC Servo mode. Select normal (independent) or differential mode on the switches before powering the module. Now you can use the PULSOUT command to simulate the servo pulse and control the motors. The pulse needs to vary between 1mS (full reverse) to 2mS (full forwards) with 1.5mS being the center off position. Unlike servo's, which require the pulse to be repeated every 20mS or so, the MD22 need only be sent a new pulse when you want to change speed. With no pulses being sent it simply continues at the current speed. The timing parameter will vary depending on the controller. Here are some popular examples - all tested by us.

Controller	Pulsout Resolution	Full reverse	Center off	Full Forwards	Command example for Stop
BS2	2uS	500	750	1000	pulsout mot1, 750
BS2e	2uS	500	750	1000	pulsout mot1, 750
BS2sx	0.8uS	1250	1875	2500	pulsout mot1, 1875
BS2p	0.8uS *	1250	1875	2500	pulsout mot1, 1875
Atom	1uS	1000	1500	2000	pulsout mot1, 1500
BX-24	1.085uS	922	1382	1843	call pulseout(mot1, 1382, 1)

* BS2p resolution is 0.8uS - rather than 0.75uS or 1.18uS as specified in earlier BS2p documentation. Parallax have confirmed this to us.

Dec 2004 (from software version 3)
Pulse time in RC mode is now verified to assure it is within specified time period of 800us to 2.2ms. Pulses outside of this timing will act to stop the motor. There is also the addition of an extra two RC servo modes which will stop the motors if a valid pulse is not received on channel 1 for a period of 200ms.

Mar 2006 (from software version 9)
The MD22 now includes two ways to implement the turn, the first uses the forward and backwards channel to control the power with channel two offsetting the power levels to turn. The second method allows channel two to introduce turn without any forward or backwards movement.

MD22 schematics are md22sch1 and md22sch2