

# MD22 - 50Volt 5Amp H Bridge Motor Drive

## Overview

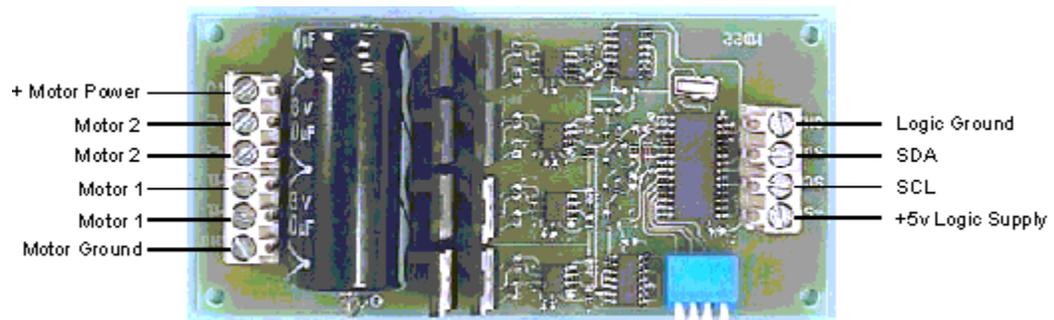
The MD22 is a robust low/medium power motor driver, designed to supply power for two motors. Main features are:

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) Motor voltage, anything up to 50vdc to suit your motors.
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 capable of much more than the 5Amp Specification.  
They are capable of 27Amp continuous with a suitable heat-sink and up to 290Amps in very brief repetitive pulses.  
Without a heat-sink they can handle 5Amps continuously, hence our specification.

## MD22 Connections

**Note - There is no fuse on the PCB. You should provide a 5A fuse in line with the +v battery terminal.**

**Don't Ignore this, High currents can be dangerous!**



The Motor Ground and the Logic Ground are internally connected on the Module. Be sure to use cable rated for at least 5A for the Battery, Fuse and Motor leads.

## 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

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.

## Differential drive

As well as controlling two motors independently, The MD22 incorporates a useful added function of differential drive, provided the motors are positioned so one is driving the left side movement and the other driving the right side. The MD22 can then use one channel to drive both motors forward/reverse and the other for turning left/right.

## 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 Mode Servo 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 through the SDA channel. Again fine adjustment to the transmitters offset may possibly be needed.

**I2C Mode**

I2C mode allows the MD22 to be connected to popular controllers such as the 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	Speed1	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 (slowest acceleration). No motor will move until directly after speed1 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, 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 speed1 control both motors speed, and speed2 becomes the turn value. Data is in the range of 0 (Full Reverse) 128 (Stop) 255 (Full Forward).

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)

### Speed1 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 2 or 3 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 speed1 but controls the operation of the motor 2. When mode is set to 2 or 3 Speed2 becomes a Turn register, and any value in speed 1 is combined with the contents of this register to steer the device.

### 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 the board changes the power (accelerates) at its slowest rate, each step taking 16.4 ms. When the acceleration register is loaded with the fastest setting 255 (default) the board will change the power every 64 us.

So to calculate the time (in seconds) for the acceleration to complete :

$$\text{time} = (256 - \text{accel reg value}) * 64\text{us}.$$

For example :

Accel reg	Time/step	Current speed	New speed	Steps	Acceleration time
0	16.4ms	0	255	255	4.18s
20	15.1ms	127	255	128	1.93s
50	13.2ms	80	0	80	1.06s
100	10ms	45	7	38	0.38s
150	6.8ms	255	5	250	1.7s
200	3.6ms	127	0	127	0.46s
255	64us	65	150	85	5.4ms

### Software Revision number

This register contains the revision number of the software in the modules PIC16F873 controller - currently 1 at the time of writing.

### 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.