MD04 - 24Volt 20Amp H Bridge Motor Drive

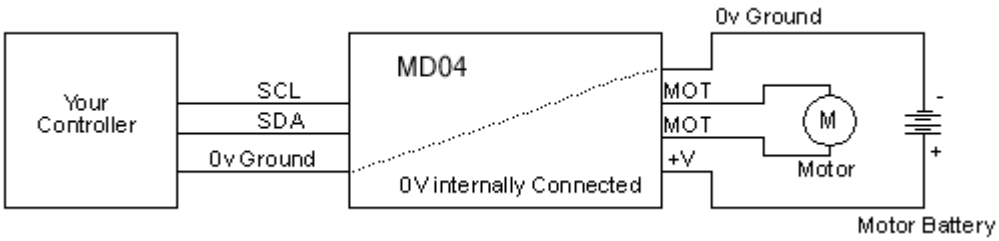
Overview

The MD04 is a medium power motor driver, designed to supply power beyond that of any of the low power single chip H-Bridges that exist. Main features are ease of use and flexibility. The motor's power is controlled by Pulse Width Modulation (PWM) of the H-Bridge at a frequency of 15khz.
Maximum voltage on the H-Bridge is 60v allowing Motor voltages up to 24vdc.
Control of the module can be any of:

1. I2C bus, up to 8 MD04 modules at unique addresses.
2. 0v-2.5-5v analog input. 0v full reverse, 2.5v center stop, 5v full forward.
3. 0v-5v analog input with separate direction control
4. RC mode. Controlled directly from the RC receiver output.
5. PWM. A simple on board filter means you can use a 0%-100% at frequencies above 20khz
6. Serial communication at TTL (5V) levels.

General usage and wiring considerations

The MD04 can handle high currents, and you will need to take a few precautions with wiring. It is very important that you do not allow motor current to flow in the logic ground path. Don't assume that just because the Battery, MD04 and Controller grounds are all together, that all is well. If at all possible, use two batteries, one for the logic and and the other for the motor power. Don't connect the battery grounds together, that is already done on the MD04. If you do, then you will only create a ground loop - and problems. The diagram below shows the general idea on keeping the logic and power sides electrically and physically separate from each other.



Motor noise suppression

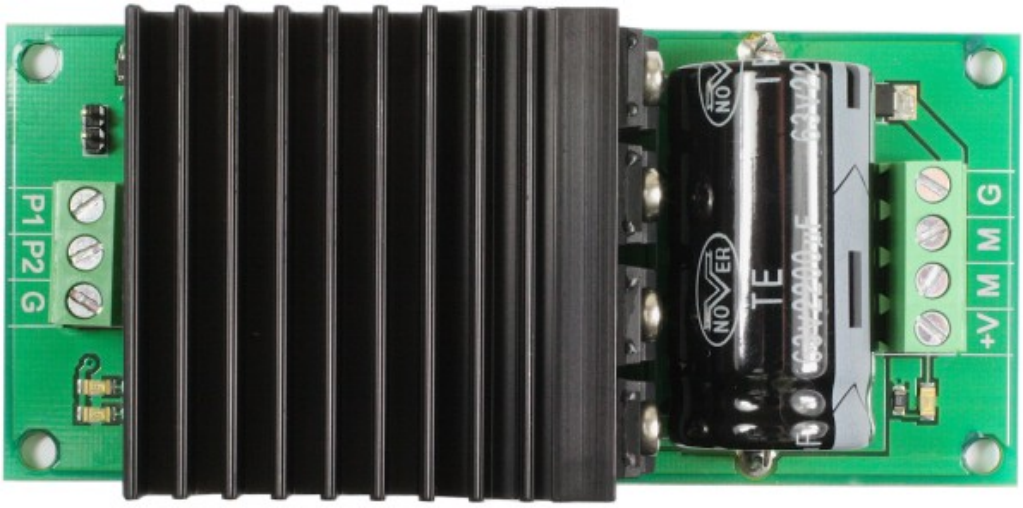
Please note that using motors with the MD04 as with any other electronic device requires suppression of noise. This is easily achieved by the addition of a 10n snubbing capacitor across the motor. The capacitor should also be capable of handling a voltage of twice the drive voltage to the motor.

Connections

Note - There is no	Two pin jumper for bootloader mode

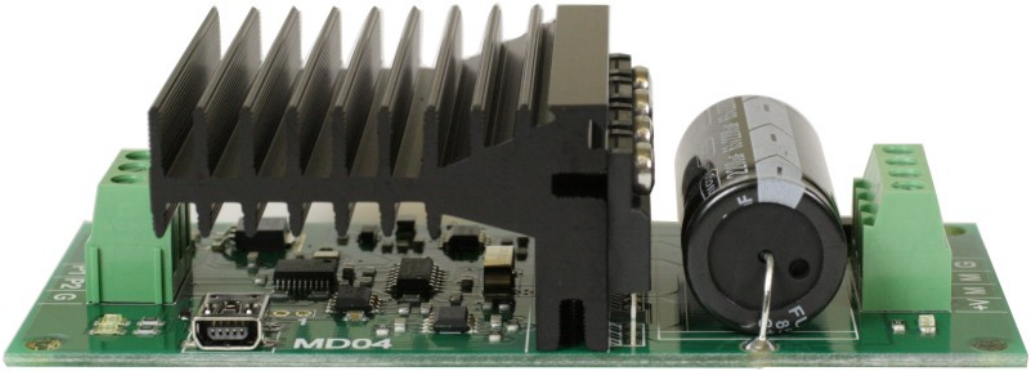
fuse on the PCB. You should provide a 25/30A fuse in line with the +v battery terminal

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



Control end screw terminal
P1 – SCL/TX/Pin 1
P2 – SDA/RX/Pin 2
G – 0V Ground

Motor end screw terminal
G – 0V ground
M – Motor connection 1
M – Motor connection 2
+V - Positive supply 8V to 24V



USB connection for setup

Be sure to use cable rated for at least 25/30A for the Battery, Fuse and Motor leads.

LED indication

There are three LED's on the MD04, the first is mounted next to the motor terminal and indicates the presence of an incoming supply. On the control end of the PCB is a green LED that will light with an incoming control signal and a red LED for problems, this LED will light when the motor is being temporarily stopped due to excess temperature or if the speed is being reduced due to over-current.

Setting the communication type

The MD04 appears as a comm port we have a simple set of commands to switch between the communication options and settings. An app to easily change between settings is available on the next page.

Command	Name	Response / additional parameters
0x24	Read Temperature	1 byte response - the PCB temperature in °C
0x25	Read Version	1 byte response - current software version
0x27	Read Mode	1 byte response - current mode of operation
0x28	Read Status	1 byte response - see status byte makeup
0x29	Read Baud	2 byte response - see baud rate calculation and table
0x34	Write Mode	Send 1 additional byte for the mode from table below
0x35	Write Baud	Send 2 additional bytes to set baud from calculation or table

Operation modes

Value	Mode
0	I2C @ 0xB0
1	I2C @ 0xB2
2	I2C @ 0xB4
3	I2C @ 0xB6
4	I2C @ 0xB8
5	I2C @ 0xBA
6	I2C @ 0xBC
7	I2C @ 0xBE
8	Analogue 0v-2.5v-5v
9	Analogue 0v-5v
10	RC mode with timeout
11	RC Mode no timeout
12	Serial

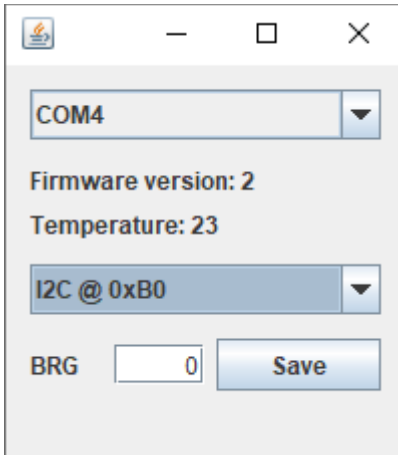
Calculated baud rates for serial mode

Baud rate	Value
300	9999
1200	2499
2400	1249
9600	311
10417	287
19.2k	155
57.6k	51
115.2k	25

Value = (48,000,000/(16* baud rate)) – 1
Because 48mhz may not divide into baud rate exactly:
Actual baud rate = 48,000,000/(16 * (Value+1))

MD04 PC setup program

We have a simple configuration program for the PC that will allow you to change the communication type and if necessary the baud rate. It operates through the USB connection on the side of the module.



Modes of operation

I2C mode

I2C mode operates with a device address set in the USB configuration. The MD04 has 8 registers:

Register	Name	Read/write	Description
0	Command	R/W	Write 1 for forwards, 2 for reverse, 0 for instant stop
1	Status	R	Acceleration, temperature and current status
2	Speed	R/W	Motor speed 0-255 (0x00 - 0xFF)
3	Acceleration	R/W	Motor acceleration 0-255 (0x00 - 0xFF)
4	Temperature	R	Module temperature in degrees centigrade
5	Motor current	R	Motor current
6	Unused	R	Read as zero
7	Version	R	Software revision

Command register

Controls the motor start, stop and direction. Write 1 to drive forwards, Write 2 to reverse, Write 0 to stop instantly. Be sure to have set the speed and acceleration before issuing these commands.
Note - The way to stop the motor is the same as other speed changes, write the new speed to the speed register and re-issue the direction command. This will cause the motor to decelerate to a stop at the rate set by the acceleration register. If you wish the motor to stop instantly and bypass the acceleration feature then write zero here, be aware that bypassing the acceleration routines does put high stresses on gearboxes of motors.

Status register

Bit 7 (msb)	6	5	4	3	2	1	Bit 0 (lsb)
Busy	-	-	-	-	over-temperature limit	over-current limiter	Acceleration in progress

Bit 0 is read as high when the drive is accelerating the motor. It will be cleared when the requested speed is achieved or the over current or over temperature limiters are active.
Bit 1 set high indicates that the current through the motor has reached 20Amps and is being limited to that value, the red LED will be illuminated when this happens.
Bit 2 set high indicates that the temperature limit has been exceeded. Above a preset threshold (80°c on the PCB), the MD04 will stop the motor drive until the temperature goes below 80°c. The MD04 will then accept instructions to drive the motor again.
Bit 7 is the busy flag. It is set high when you issue a new command to the module. It is cleared very quickly and you are unlikely ever to see it set.

Speed register
Sets the maximum speed that the motor will accelerate to. It is actually the 8-bit value sent to the modules PWM controller. Write a value of 0 to 243 (numbers from 243 to 255 are clamped to 243) . The larger the number, the more power is applied to the motor.

Acceleration register
This sets the rate at which the motor accelerates or decelerates from where it is towards the new speed set by the speed register. Write a value of 0 to 255, the larger the number the longer the module will take to reach the new speed. Writing zero to the acceleration register will allow maximum acceleration of 0 to full in 0.187 seconds. This value actually controls a timer which steps the current motor speed towards the requested motor speed. It does this every (((acceleration register)*125)+768)uS. A value of zero gives 768uS/step or 243*768=186624uS (0.187s) to accelerate from 0 to full. A value of 255 is ((255*125)+768)*243=7932249uS or just under 8 seconds.

Temperature register
Register gives a reading of the current surface temperature of the PCB in degrees centigrade.

Motor current
This is the value used internally to limit the motor current to 20A. You do not need to read or do anything with it. The value is proportional to motor current, with a value of 186 representing the 20A limit.

Software revision number
The revision number of the software in the modules PIC controller.

Serial mode

TTL level serial operating at a baud rate that is set in the USB configuration. Commands are:

Hex command value	Command	Returned byte / additional byte
0x21	Read speed	Returns 1 byte (0 - 0xff)
0x22	Read direction	Returns 1 byte (0 / 1)
0x23	Read current	Returns 1 byte (0 - 0xff)
0x24	Read temperature	Returns 1 byte (-40°c to 125°c)
0x25	Read software version	Returns 1 byte
0x26	Read acceleration	Returns 1 byte (0 - 0xff)
0x28	Read status	Returns 1 byte as described in I2C section
0x31	Write speed	Follow with desired speed (0 - 0xff)
0x32	Write command	Follow with desired direction (1/ 2) or instant stop (0)
0x33	Write acceleration	Follow with desired acceleration (0 - 0xff)

Status register

Bit 7 (msb)	6	5	4	3	2	1	Bit 0 (lsb)
Busy	-	-	-	-	over-temperature limiter	over-current limiter	Acceleration in progress

Bit 0 is read as high when the drive is accelerating the motor. It will be cleared when the requested speed is achieved or the over current or over temperature limiters are active.
Bit 1 set high indicates that the current through the motor has reached 20Amps and is being limited to that value, the red LED will be illuminated when this happens.
Bit 2 set high indicates that the over temperature limiter is active. Above a preset threshold, the motor current will be reduced in proportion to the MD04 temperature. The module can still be used but the motor power will be limited, the red LED will be illuminated when this happens. It should be noted that a few minutes of running continuously at 20A will cause the heatsink to get hot - watch your fingers!
Bit 7 is the busy flag. It is set high when you issue a new command to the module. It is cleared very quickly and you are unlikely ever to see it set.

Speed register

Sets the maximum speed that the motor will accelerate to. It is actually the 8-bit value sent to the modules PWM controller. Write a value of 0 to 243 (numbers from 243 to 255 are clamped to 243) . The larger the number, the more power is applied to the motor.

Acceleration register

This sets the rate at which the motor accelerates or decelerates from where it is towards the new speed set by the speed register. Write a value of 0 to 255, the larger the number the longer the module will take to reach the new speed. Writing zero to the acceleration register will allow maximum acceleration of 0 to full in 0.187 seconds. This value actually controls a timer which steps the current motor speed towards the requested motor speed. It does this every (((acceleration register)*125)+768)uS. A value of zero gives 768uS/step or 243*768=186624uS (0.187s) to accelerate from 0 to full. A value of 255 is ((255*125)+768)*243=7932249uS or just under 8 seconds.

Temperature register
Register gives a reading of the current surface temperature of the PCB in degrees centigrade.

Motor current
This is the value used internally to limit the motor current to 20A. You do not need to read or do anything with it. The value is proportional to motor current, with a value of 186 representing the 20A limit.

Software revision number
The revision number of the software in the modules PIC controller.

Analog mode – 0v-2.5v-5v
In this mode the motor is controlled by a 0v to 5v analog signal only on the SDA line. Pin SCL is unused and should be connected to either +5v or 0v.
0v is maximum reverse power
2.5v is the center stop position
5v is full forward power
The is a small (2.7%) dead band around 2.5v to provide a stable off position, input impedance is 47k.

Analog mode – 0v-5v
In this mode the motor is controlled by a 0v to 5v analog signal on the SDA line and direction on SCL.
0v is stop position
5v is full power
Pin SCL is the logic level (ttl) direction control. logic 0 for reverse direction and logic 1 for forward direction.

You may also use a PWM signal instead of an analog voltage on the SDA line. There is a simple resistor/capacitor filter on the module which will generate the analog voltage from the incoming PWM signal. your PWM signal should be 20khz or greater in frequency and ideally, come from a CMOS gate (0-5v) rather than ttl (0-3.5v ish). a 0% duty cycle will represent 0v and a 100% duty cycle representing 5v. This applies to both the above analog modes. Note that the PWM input is not the same as the PWM motor drive which is generated seperately.

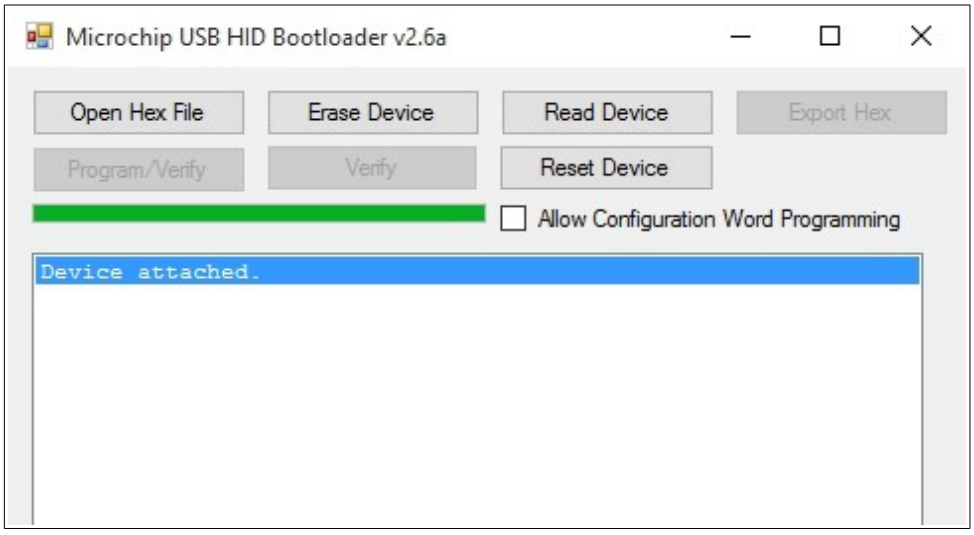
RC mode
This mode allows direct connection to standard model radio control receivers. The control pulse (Yellow) from the receiver should be connected to the SDA terminal. The SCL terminal is unused and should be connected to 0v. Connect the receiver 0v ground (Black) to the MD04 logic ground. The output from an RC receiver is a high pulse 1.5mS wide when the joystick is central. This varies down to 1.1mS and up to 1.9mS as the joystick is moved. This range can be shifted by the centering control by +/-100uS from 1mS-1.8mS to 1.2mS-2mS. The MD04 provides full control in the range 1.1mS to 1.9mS 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 with timeout
Operates in much the same way as the normal RC control with the difference being the addition of a timeout feature. If a RC pulse is not detected for a period in excess of 200ms, then the motor will be stop being driven until a valid RC signal is received

Boot loader

The MD04 has a built in boot loader meaning the firmware may be updated by the user when a new version becomes available. The boot-loader is based on the USB boot-loader provided by Microchip as part of their "Microchip Application Libraries" and can be downloaded from here.

To put the MD04 into boot loader mode you should now turn any power supply to the MD04 off and place a link across the two pin header. Now run the boot loader you have downloaded and insert the USB cable, you should now have a screen similar to that below allowing the software hex to be re-flashed in the event of a future update.



Board dimensions

The following drawing shows the MD04 mounting hole positions: