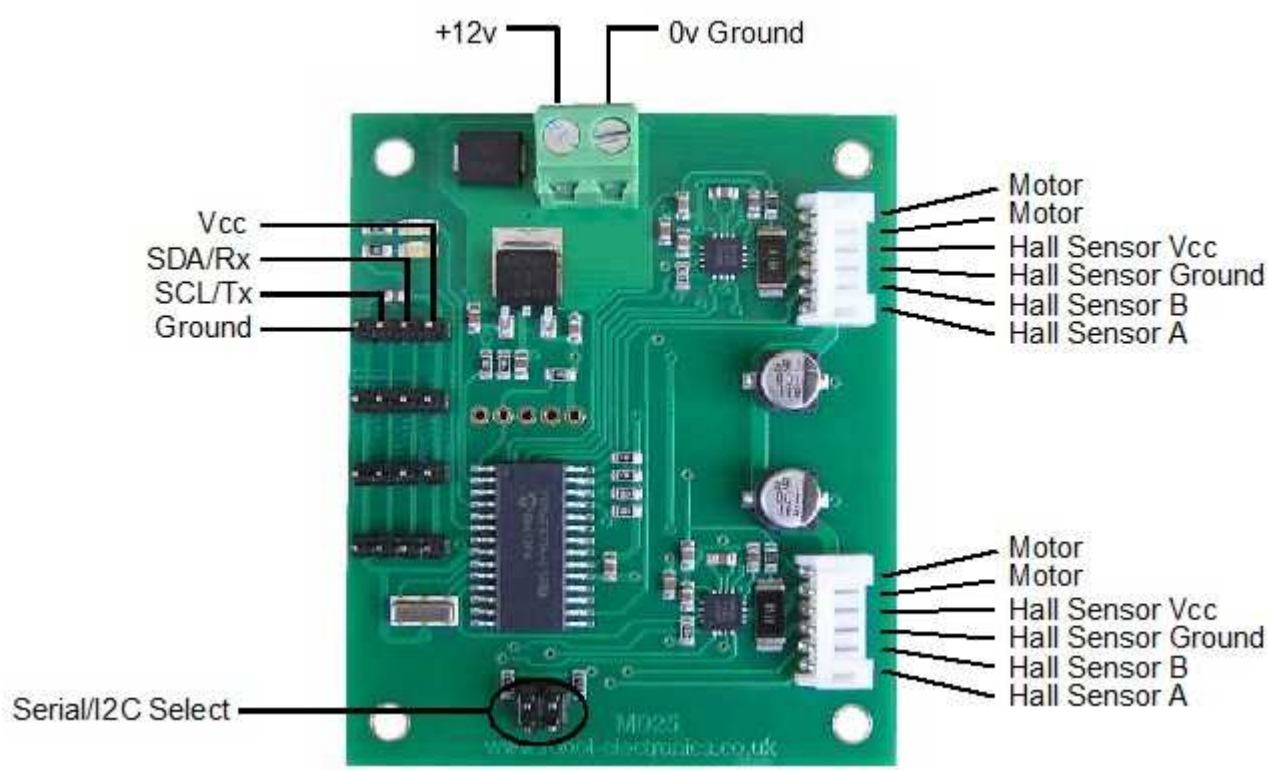
MD25 - Dual 12Volt 2.8Amp H Bridge Motor Drive

Overview

The MD25 is a robust I2C or serial, dual motor driver, designed for use with our EMG30 motors. Main features are:

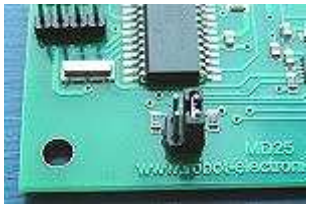
1. Reads motors encoders and provides counts for determining distance traveled and direction .
2. Drives two motors with independent or combined control.
3. Motor current is readable.
4. Only 12v is required to power the module.
5. Onboard 5v regulator can supply up to 1A peak, 300mA continuously to external circuitry
6. Steering feature, motors can be commanded to turn by sent value.
7. Variable acceleration and power regulation also included

Connections



Jumper Selection

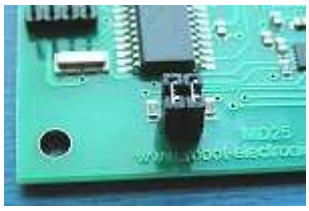
I2C mode with no jumpers installed, up to 100 khz clock.
Full Details of I2C Mode is here



Serial mode at 9600 bps, 1 start bit, 2 stop bits, no parity
Full Details of Serial Mode is here

Serial mode at 19200 bps, 1 start bit, 2 stop bits, no parity

Full Details of Serial Mode is here



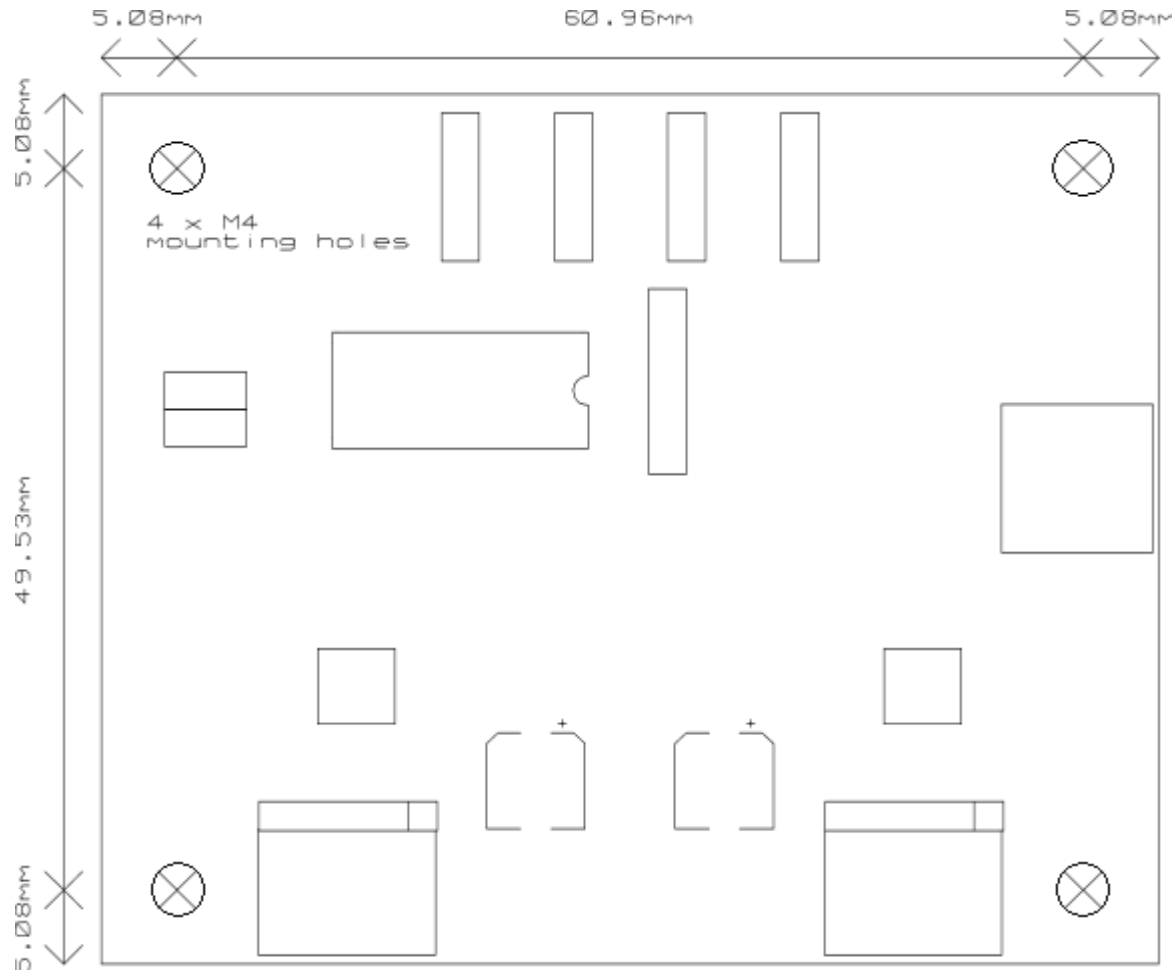
Serial mode at 38400 bps, 1 start bit, 2 stop bits, no parity
Full Details of Serial Mode is here

Motor Voltage
The MD25 is designed to work with a 12v battery. In practical terms, this means the 9v-14v swing of a flat/charging 12v battery is fine. Much below 9v and the under-voltage protection will prevent any drive to the motors.

Motor Noise Suppression
When using our EMG30 encoded motors, you will find that a 10n noise suppression capacitor has already been fitted. Other motors may require suppression. 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.

Leds
The Red Power Led indicates power is applied to the module.
A Green Led indicates communication activity with the MD25. In I2C mode the green led will also initially flash the address it has been set to. See I2C documentation for further details.

Board dimensions


5.08mm
60.96mm
5.08mm
5.08mm
49.53mm
5.08mm
4 x M4
mounting holes

MD25 - Dual 12Volt 2.8Amp H Bridge Motor Drive

I2C mode documentation

(Click here for Serial Mode)

Automatic Speed regulation
By using feedback from the encoders the MD25 is able to dynamically increase power as required. If the required speed is not being achieved, the MD25 will increase power to the motors until it reaches the desired rate or the motors reach there maximum output. Speed regulation can be turned off in the command register.

Automatic Motor Timeout
The MD25 will automatically stop the motors if there is no I2C communications within 2 seconds. This is to prevent your robot running wild if the controller fails. The feature can be turned off, if not required. See the command register.

Controlling the MD25
The MD25 is designed to operate in a standard I2C bus system on addresses from 0xB0 to 0xBE (last bit of address is read/write bit, so even numbers only), with its default address being 0xB0. This is easily changed by removing the Address Jumper or in the software see Changing the I2C Bus Address.

I2C mode allows the MD25 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 MD25 module is the same as popular EPROM's such as the 24C04. To read one or more of the MD25 registers, first send a start bit, the module address (0XB0 for example) 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 MD25 has 17 registers numbered 0 to 16 as follows;

Register	Name	Read/Write	Description
0	Speed1	R/W	Motor1 speed (mode 0,1) or speed (mode 2,3)
1	Speed2/Turn	R/W	Motor2 speed (mode 0,1) or turn (mode 2,3)
2	Enc1a	Read only	Encoder 1 position, 1st byte (highest), capture count when read
3	Enc1b	Read only	Encoder 1 position, 2nd byte
4	Enc1c	Read only	Encoder 1 position, 3rd byte
5	Enc1d	Read only	Encoder 1 position, 4th (lowest byte)
6	Enc2a	Read only	Encoder 2 position, 1st byte (highest), capture count when read
7	Enc2b	Read only	Encoder 2 position, 2nd byte
8	Enc2c	Read only	Encoder 2 position, 3rd byte
9	Enc2d	Read only	Encoder 2 position, 4th byte (lowest byte)
10	Battery volts	Read only	The supply battery voltage
11	Motor 1 current	Read only	The current through motor 1
12	Motor 2 current	Read only	The current through motor 2
13	Software Revision	Read only	Software Revision Number
14	Acceleration rate	R/W	Optional Acceleration register
15	Mode	R/W	Mode of operation (see below)
16	Command	R/W	Used for reset of encoder counts and module address changes

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 and direction of motor 1. The larger the number written to this register, the more power is applied to the motor. A mode of 2 or 3 will control the speed and direction of both motors (subject to effect of turn register).

Speed2/Turn Register
When in mode 0 or 1 this register operates the speed and direction of motor 2. When in mode 2 or 3 Speed2 becomes a Turn register, and any value in this register is combined with the contents of Speed1 to steer the device (see below).

Turn mode
Turn mode looks at the speed register to decide if the direction is forward or reverse. Then it applies 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.

Encoder registers
Each motor has its encoder count stored in an array of four bytes, together the bytes form a signed 32 bit number, the encoder count is captured on a read of the highest byte (registers 2, 6) and the subsequent lower bytes will be held until another read of the highest byte takes place. The count is stored with the highest byte in the lowest numbered register. The registers can be zeroed at any time by writing 32 (0x20) to the command register.

Battery volts
A reading of the voltage of the connected battery is available in this register. It reads as 10 times the voltage (121 for 12.1v).

Motor 1 and 2 current
A guide reading of the average current through the motor is available in this register. It reads approx ten times the number of Amps (25 at 2.5A).

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

Acceleration Rate
If you require a controlled acceleration period for the attached motors to reach there ultimate speed, the MD25 has a register to provide this. It works by using a value into the acceleration register and incrementing the power by that value. Changing 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 with an acceleration register value of 1, but 128 for a value of 2. The default acceleration value is 5, meaning the speed is changed from full forward to full reverse in 1.25 seconds. The register will accept values of 1 up to 10 which equates to a period of only 0.65 seconds to travel from full speed in one direction to full speed in the opposite direction.

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

if new speed > current speed
steps = (new speed - current speed) / acceleration register

if new speed < current speed
steps = (current speed - new speed) / acceleration register

time = steps * 25ms

For example :

Acceleration register	Time/step	Current speed	New speed	Steps	Acceleration time
1	25ms	0	255	255	6.375s
2	25ms	127	255	64	1.6s
3	25ms	80	0	27	0.675s
5 (default)	25ms	0	255	51	1.275s
10	25ms	255	0	26	0.65s

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 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)

Command register

Command		Action
Dec	Hex	
32	20	Resets the encoder registers to zero
48	30	Disables automatic speed regulation
49	31	Enables automatic speed regulation (default)
50	32	Disables 2 second timeout of motors (Version 2 onwards only)
51	33	Enables 2 second timeout of motors when no I2C comms (default) (Version 2 onwards only)
160	A0	1st in sequence to change I2C address
170	AA	2nd in sequence to change I2C address
165	A5	3rd in sequence to change I2C address

Changing the I2C Bus Address
To change the I2C address of the MD25 by writing a new address you must have only one module on the bus. Write the 3 sequence commands in the correct order followed by the address. Example; to change the address of an MD25 currently at 0xB0 (the default shipped address) to 0xB4, write the following to address 0xB0; (0xA0, 0xAA, 0xA5, 0xB4). These commands must be sent in the correct sequence to change the I2C address, additionally, no other command may be issued in the middle of the sequence. The sequence must be sent to the command register at location 16, which means 4 separate write transactions on the I2C bus. Because of the way the MD25 works internally, there MUST be a delay of at least 5mS between the writing of each of these 4 transactions. When done, you should label the MD25 with its address, however if you do forget, just power it up without sending any commands. The MD25 will flash its address out on the green communication LED. One long flash followed by a number of shorter flashes indicating its address. Any command sent to the MD25 during this period will still be received and writing new speeds or a write to the command register will terminate the flashing.

Address		Long Flash	Short Flashes
Decimal	Hex		
176	B0	1	0
178	B2	1	1
180	B4	1	2
182	B6	1	3
184	B8	1	4
186	BA	1	5
188	BC	1	6
190	BE	1	7

Take care not to set more than one MD25 to the same address, there will be a bus collision and very unpredictable results.

MD25 - Dual 12Volt 2.8Amp H Bridge Motor Drive

Serial mode documentation

(click here for I2C mode)

Automatic Speed regulation
By using feedback from the encoders the MD25 is able to dynamically increase power as required. If the required speed is not being achieved, the MD25 will increase power to the motors until it reaches the desired rate or the motors reach there maximum output. Speed regulation can be turned off with the use of the REGULATOR DISABLE command..

Automatic Motor Timeout
The MD25 will automatically stop the motors if there is no I2C communications within 2 seconds. This is to prevent your robot running wild if the controller fails. The feature can be turned off with the DISABLE TIMEOUT command

Controlling the MD25
The MD25 is designed to operate with a TTL level serial bus (5v levels). Do not connect to RS232 directly, if you wish to connect to RS232 it must be with the aid of a voltage level converter such as a ST232 or serial interface such as S13 which is available here: www.robot-electronics.co.uk/acatalog/Serial_Interface.html

Commands
An easy to use command set provides all of the functions that the MD25 has to offer. The commands are sent with a sync byte of 0 at the start and then the command followed by any data bytes. The MD25 will then respond if the command is applicable.

command	Name	Bytes sent to MD25	Bytes returned by MD25	Description
0x21	GET SPEED 1	2	1	returns the current requested speed of motor 1
0x22	GET SPEED 2	2	1	returns the current requested speed of motor 2
0x23	GET ENCODER 1	2	4	motor 1 encoder count, 4 bytes returned high byte first (signed)
0x24	GET ENCODER 2	2	4	motor 2 encoder count, 4 bytes returned high byte first (signed)
0x25	GET ENCODERS	2	8	returns 8 bytes - encoder1 count, encoder2 count
0x26	GET VOLTS	2	1	returns the input battery voltage level
0x27	GET CURRENT 1	2	1	returns the current drawn by motor 1
0x28	GET CURRENT 2	2	1	returns the current drawn by motor 1
0x29	GET VERSION	2	1	returns the MD25 software version
0x2A	GET ACCELERATION	2	1	returns the current acceleration level
0x2B	GET MODE	2	1	returns the currently selected mode
0x2C	GET VI	2	3	returns battery volts, motor1 current and then motor2 current
0x31	SET SPEED 1	3	0	set new speed1
0x32	SET SPEED 2 / TURN	3	0	set new speed2 or turn
0x33	SET ACCELERATION	3	0	set new acceleration
0x34	SET MODE	3	0	set the mode
0x35	RESET ENCODERS	2	0	zero both of the encoder counts
0x36	DISABLE REGULATOR	2	0	power output not changed by encoder feedback
0x37	ENABLE REGULATOR	2	0	power output is regulated by encoder feedback
0x38	DISABLE TIMEOUT	2	0	MD25 will continuously output with no regular commands

0x39	ENABLE TIMEOUT	2	0	MD25 output will stop after 2 seconds without communication

For example to read the battery voltage, send:
0x00 - sync byte
0x26 - READ VOLTS command
and the MD25 would respond with
0x77 - returned byte (119 decimal) 11.9v

Speed1
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 and direction of motor 1. The larger the number written to this register, the more power is applied to the motor. A mode of 2 or 3 will control the speed and direction of both motors (subject to effect of turn register).

Speed2/Turn
When in mode 0 or 1 this operates the speed and direction of motor 2. When in mode 2 or 3 Speed2 becomes a Turn value, and is combined with Speed1 to steer the device (see below).

Turn mode
Turn mode looks at the speed1 to decide if the direction is forward or reverse. Then it applies a subtraction or addition of the turn value on either motor.

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

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

GET ENCODER 1, GET ENCODER 2 or GET ENCODERS
When a read encoder command is issued the MD25 will send out 4 bytes high byte first, which should be put together to form a 32 bit signed number. For example a GET ENCODER 1 command may return 0x00,0x10,0x56,0x32.

So declare a 32 bit signed variable in your program, for C:

```
long result;
result = serin() << 24;    // (0x00 shifted 24 bits left, effectively *
16777216)
result += serin() << 16;   // (0x10 shifted 16 bits left, effectively *
65536)
result += serin() << 8;    // (0x56 shifted 8 bits left, effectively *
256)
result += serin();         /  (0x32)
```

result now equals 1070642 decimal or 0x105632 hex. If the highest bit was set then it would be -ve.
read encoders will send encoder count 1 and then encoder count 2 but is put together in exactly the same way.
The registers can be zeroed at any time by writing 0x35 to the MD25.

Battery volts
A reading of the voltage of the connected battery is available. It returns as 10 times the voltage (121 for 12.1v).

Motor 1 and 2 current

A guide reading of the average current through the motor is available. It reads approx ten times the number of Amps (25 at 2.5A).

Software Revision number
Responds with the revision number of the software in the modules PIC16F873 controller - currently 1 at the time of writing.

Acceleration Rate
If you require a controlled acceleration period for the attached motors to reach there ultimate speed, the MD25 has the ability to provide this. It works by using a sent acceleration value and incrementing the power by that value. Changing between the current speed of the motors and the new speed. 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 with an acceleration register value of 1, but 128 for a value of 2. The default acceleration value is 5, meaning the speed is changed from full forward to full reverse in 1.25 seconds. The WRITE ACCELERATION command will accept values of 1 up to 10 which equates to a period of only 0.65 seconds to travel from full speed in one direction to full speed in the opposite direction.

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

if new speed > current speed
steps = (new speed - current speed) / acceleration register

if new speed < current speed
steps = (current speed - new speed) / acceleration register

time = steps * 25ms

For example :

Acceleration register	Time/step	Current speed	New speed	Steps	Acceleration time
1	25ms	0	255	255	6.375s
2	25ms	127	255	64	1.6s
3	25ms	80	0	27	0.675s
5 (default)	25ms	0	255	51	1.275s
10	25ms	255	0	26	0.65s

Mode
The mode command changes the way the speed/turn values are used. The options being:
0, (Default Setting) If a value of 0 is written then 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 values are interpreted as signed values. The range being -128 (Full Reverse) 0 (Stop) 127 (Full Forward).

2, Writing a value of 2 to the mode 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 values are interpreted as signed values.
Data is in the range of -128 (Full Reverse) 0 (Stop) 127 (Full Forward)

GET VI or SET VI
This command instructs the MD25 to send the battery volts reading (125 = 12.5v), then the current being drawn by motor 1 (roughly 1 count per 100mA) and finally the current being drawn by motor 2.