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.