



MD3 User Guide

V1.7



Contents

1. Introduction	4
2. Connectors and LED's	4
3. Quick Start Guide	6
3.1. MD3 Setup Program.....	7
3.2. MD3 Demo Program	9
4. MD3 Operation	14
5. MD3 Interfaces.....	14
5.1. RS485 Interface.....	15
5.2. Digital Input Interface	15
6. MD3 RS485 Serial Interface Protocol (Modbus RTU)	16
6.1. Read Holding Register (Function Code 3)	16
6.2. Write Single Register (Function Code 6)	18
7. MD3 Register Description	20
7.1. Register 0x00: DeviceAddress.....	22
7.2. Register 0x01: MotorCurrentAndMicrostepping	22
7.3. Register 0x02: CurrentCutbackSettings	23
7.4. Register 0x03: StartupDelay (unsigned, 16-bits).....	24
7.5. Register 0x04: HomingConfiguration	24
7.6. Register 0x05/0x06: PreHomeOffsetHigh / PreHomeOffsetLow (signed, 32-bits).....	25
7.7. Register 0x07/0x08: HomeOffsetHigh / HomeOffsetLow (signed, 32-bits).....	26
7.8. Register 0x09: PinFunctionAndPolarity	26
7.9. Register 0x0a: I/OPortData	31
7.10. Register 0x0b: Reserved (Do not use).....	32
7.11. Register 0x0c: QuadratureEncoderConfiguration.....	32
7.12. Register 0x0d: QuadratureEncoderMaxCount (unsigned, 16-bits)	34
7.13. Register 0x0e/0x0f: EncoderIndexCountHigh / EncoderIndexCountLow (signed, 32-bits)	34
7.14. Register 0x10: QuadratureEncoderCount (unsigned, 16-bits).....	35
7.15. Register 0x11: MotionControl.....	35
7.16. Register 0x12/0x13: MoveStepsHigh / MoveStepsLow (signed, 32-bits).....	37
7.17. Register 0x14/0x15: SpeedHigh / SpeedLow (signed, 32-bits)	37
7.18. Register 0x16/0x17: AlternateSpeedHigh / AlternateSpeedLow (unsigned, 32-bits)	38

7.19.	Register 0x18/0x19: AccelDecelHigh / AccelDecelLow (unsigned, 32-bits).....	39
7.20.	Register 0x1a: EEPROMControl (register not saved to EEPROM).....	39
7.21.	Register 0x1b: Product Information (Read Only).....	40
7.22.	Register 0x1c: SerialPortConfiguration	40
7.23.	Register 0x1d: UserRegister	41
7.24.	Register 0x1e/0x1f: SerialNumberHigh / SerialNumberLow (unsigned, 32-bits, Read Only).....	41

1. Introduction

The MD3 is a stepper motor driver with integrated motion controller that offers enhanced functionality compared to US Digital's previous generation MD2S-D and MD2S-P motor drives. The main MD3 features are:

- close compatibility with existing MD2S-D and MD2S-P motor drives
- 1/2 to 1/256 microstepping
- peak motor currents up to 7A continuous
- configurable current cutback mode to reduce motor heating when idle
- automatic thermal shutdown
- programmable acceleration, speed and move steps
- motion profiles that can be triggered by isolated digital inputs or serial commands
- switch or encoder index homing cycle with programmable startup delay and pre/post home offset movement
- quadrature counter for incremental encoder
- general purpose I/O available
- Modbus RTU protocol over RS485 bus used for configuration and serial commands
- supports multiple MD3 units on the same RS485 bus
- DC motor PWM control mode
- firmware updatable by end user
- free demo software that supports all functions
- open, register-based serial communication protocol

2. Connectors and LED's

The MD3 has 3 connectors (J2, J3, P1) and 2 status LED's as shown in the tables below.

Connector J2 Pin	Description
1	Isolator GND input (must be connected to J2 Pin 8 or external GND)
2	LOW CURRENT, SPEEDSEL, or HOME SENSOR input (isolated, 5k pullup)
3	INHIBIT or HOME input (isolated, 5k pullup. Also clears non-thermal drive fault state)
4	STEP or GO-STOP input (isolated, 5k pullup)
5	DIR input (isolated, 5k pullup)
6	Isolator +5V input (must be connected to J2 Pin 7 or external +5V source)
7	+5V out
8	GND
9	Encoder Index or digital I/O bit
10	Encoder A or digital I/O bit
11	+5V out
12	Encoder B or digital I/O bit

Connector J3 Pin	Description
1	Power supply, 9-50 VDC
2	GND
3	Stepper Phase A
4	Stepper Phase A-
5	Stepper Phase B (DC Motor +)
6	Stepper Phase B- (DC Motor -)
7	Busy output (open drain FET ground).Output is switched to GND when the motor is moving or the drive is in thermal shutdown or other fault state
8	Digital output bit (open drain FET to ground).

Caution: Do not connect or disconnect the motor leads when the MD3 is powered. Make sure that all connections are made and J3 is plugged in before applying the main power. Do not unplug J3 when the drive is powered. Failure to do so may damage the drive.

Connector P1 Pin	Description
1	GND
2	Reserved
3	Reserved
4	Reserved
5	RS485 BUS+
6	RS485 BUS-

To maintain close compatibility with the previous MD2S driver (which has 6 and 8 pin connectors), the pinout of the MD3 8 and 12 pin connectors is designed so that prewired MD2S screw terminal plugs can be plugged directly into the MD3 and function the same. See Figure 1.



Figure 1. MD2S screw terminal plug orientation in MD3

State	Heartbeat LED (Green, D6)	Current Cutback LED (Red, D5)
Boot-up	Both LED's will flash on/off twice if the MD3 registers were loaded without errors from EEPROM. If there are errors, the red LED will flash rapidly on/off for 5 seconds and the MD3 registers will be initialized with default values. The drive will then enter the "Startup delay" state.	
Startup Delay	There is a built in 2.5 second startup delay before the motor current is turned on and the drive responds to commands. Additional delay can be configured using the MD3 Demo program or with serial commands. During the Startup delay period, the Heartbeat and Current Cutback LED's will flash alternately. When this state ends, the drive will enter the "Normal operation" state.	
Normal Operation	Flashes once every ~1.5 seconds	Off
Current Cutback	Flashes once every ~1.5 seconds	On
Thermal Shutdown or other fault	Off	Flashes once every ~1.5 seconds

3. Quick Start Guide

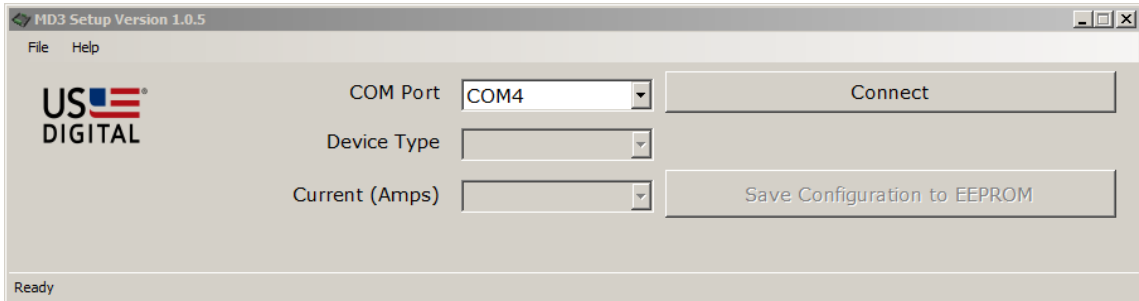
- a) Connect J2 Pin 1 to J2 Pin 8. Connect J2 Pin 6 to Pin 7. This self-powers the input isolators. For improved noise isolation, an external +5V power supply can be connected to J2 Pin 1 and J2 Pin 6 to power the isolators.
- b) Connect a two-phase stepper motor and power supply to the J3 connector.
- c) Turn on the power supply.
- d) Connect a PC's USB port to connector P1 of the MD3 using a USB-to-RS485 adapter such as US Digital's **CA-MD3** (not included) or equivalent.
- e) Download and install the MD3 Software from the following link:
<https://www.usdigital.com/support/resources/downloads/software/md3-software/>

The **MD3 Setup** program is used to configure the MD3 as a MD2S-D or MD2S-P.

The **MD3 Demo** program is used to access the full features of the MD3.

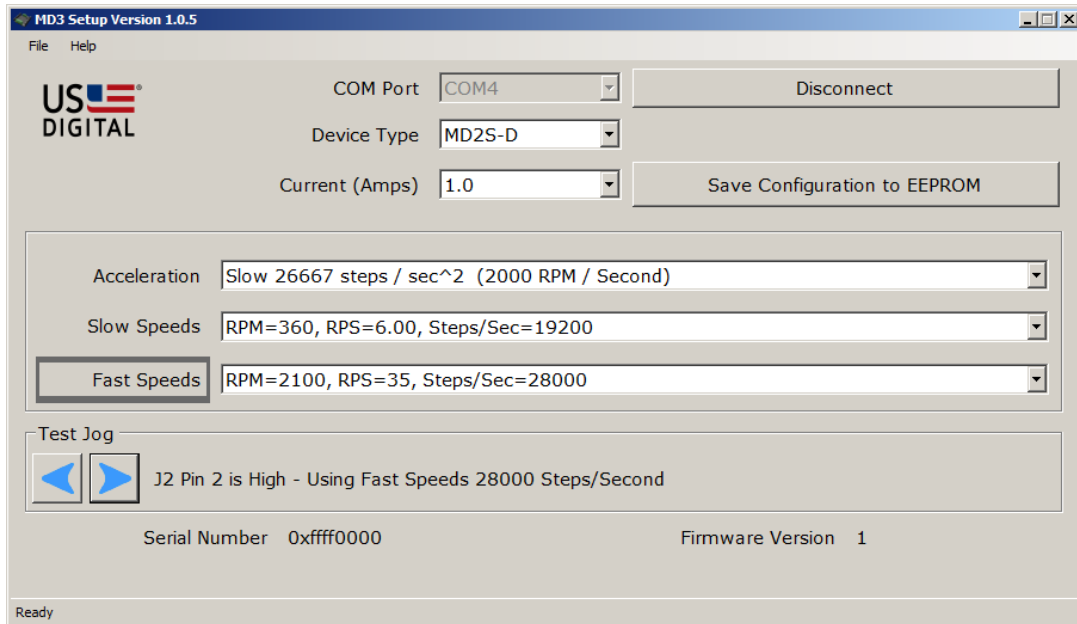
3.1. MD3 Setup Program

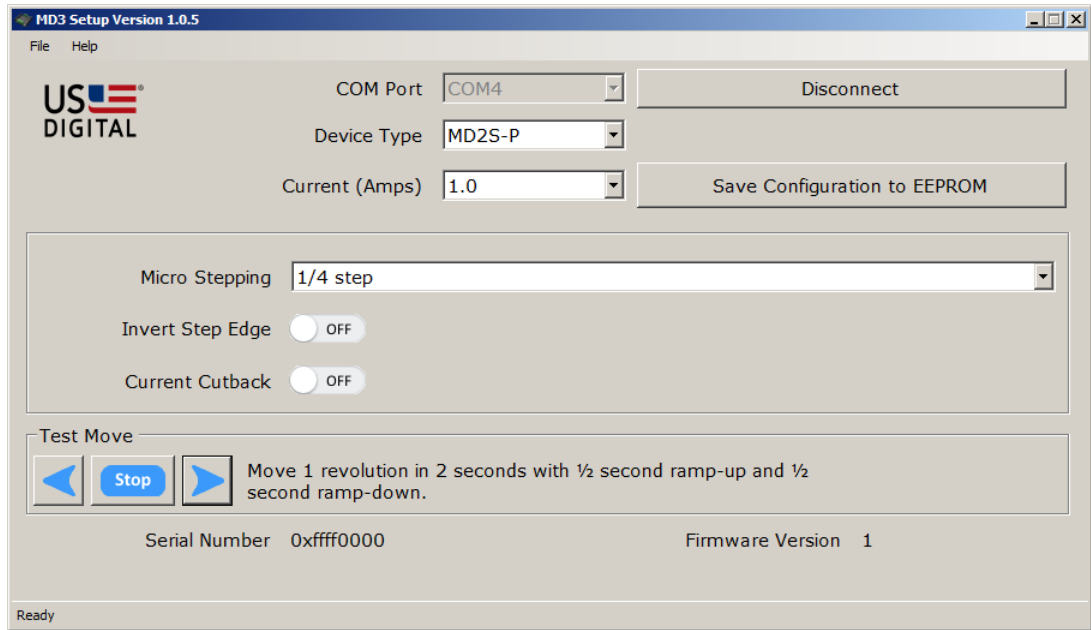
The **MD3 Setup** program presents a simple interface for customers that want to configure the MD3 to function as US Digital's previous MD2S-D or MD2S-P product. The following window will appear when the program is run:



Use the "COM Port" pulldown to choose the PC COM port the RS485 adapter is using and click on "Connect". The program will scan and connect to the MD3. Depending on the RS485 adapter used, if the program cannot connect, swap the Bus+ / Bus- leads at the output of the adapter and try again.

Once it connects, use the "Device Type" pulldown to configure the MD3 as a MD2S-D or MD2S-P. This will bring up one of the two following windows:





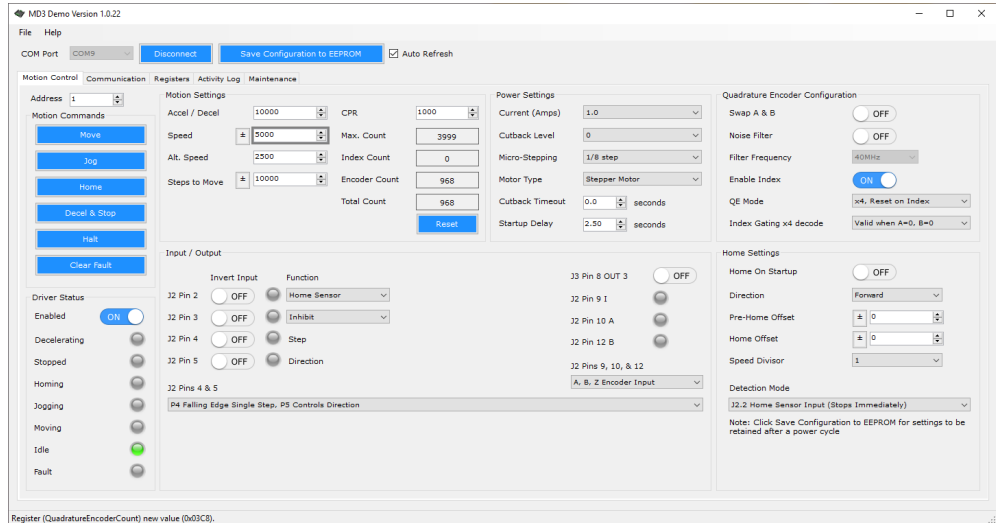
Use the pulldowns and buttons to configure the MD3 as desired. The settings are similar to the options available using the jumpers and DIP switches on the MD2S-D and MD2S-P. Click on “Save Configuration to EEPROM” to save the settings in non-volatile memory of the MD3.

If multiple MD3 units need to be configured the same way, use “Save Configuration File As...” in the “File” menu to save the current program configuration to a text .cfg file for later use. The .cfg can be loaded and programmed using the “Select Configuration File:” and “Write Configuration File to EEPROM” selections in the “File” Menu.

The left and right arrows can be used to test the movement of the motor without applying external signals to J2.

3.2. MD3 Demo Program

The **MD3 Demo** program gives the user full control of all the features of the MD3. The following window will appear when the program is run:

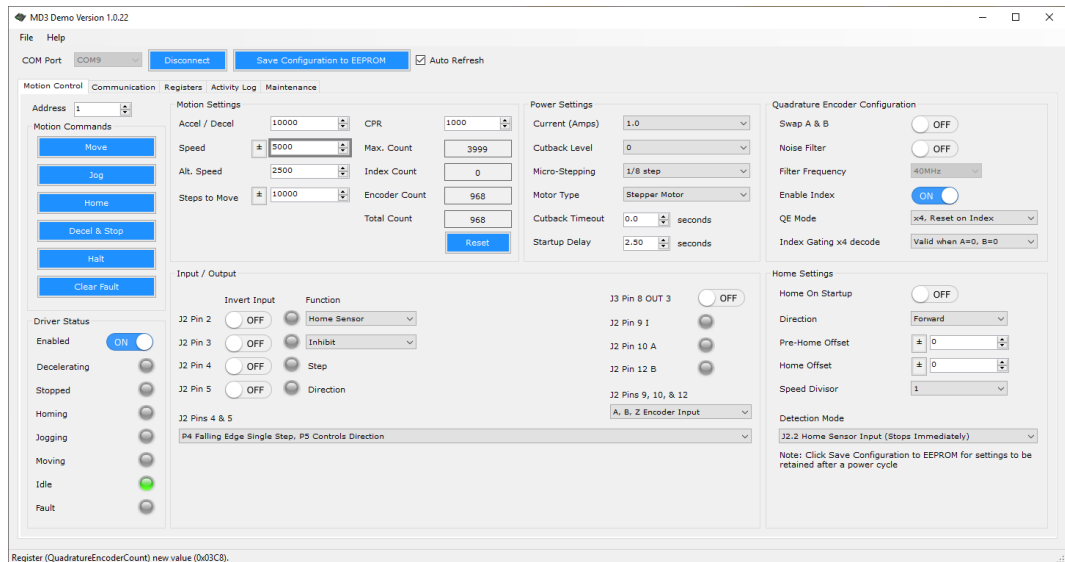


Use the “COM Port” pulldown to choose the PC COM port the RS485 adapter is using and click on “Connect”. The program will scan and connect to the MD3. Depending on the RS485 adapter used, if the program cannot connect, swap the Bus+ / Bus- leads at the output of the adapter and try again.

The 5 tabs: “Motion Control”, “Communication”, “Registers”, “Activity Log” and “Maintenance” are used to modify and display all the MD3 settings. Changes to the settings will modify the internal RAM registers of the MD3 and take effect immediately, but they will not be saved to non-volatile EEPROM on the MD3 until the “Save Configuration to EEPROM” button is pressed. If the current configuration is not wanted, the unit can be power cycled to reload the previous settings from EEPROM. Alternately, the “Reset Device” button in the “Maintenance” tab can be used to reset the MD3 and reload the settings from EEPROM.

The following sections will give a brief overview of each tab.

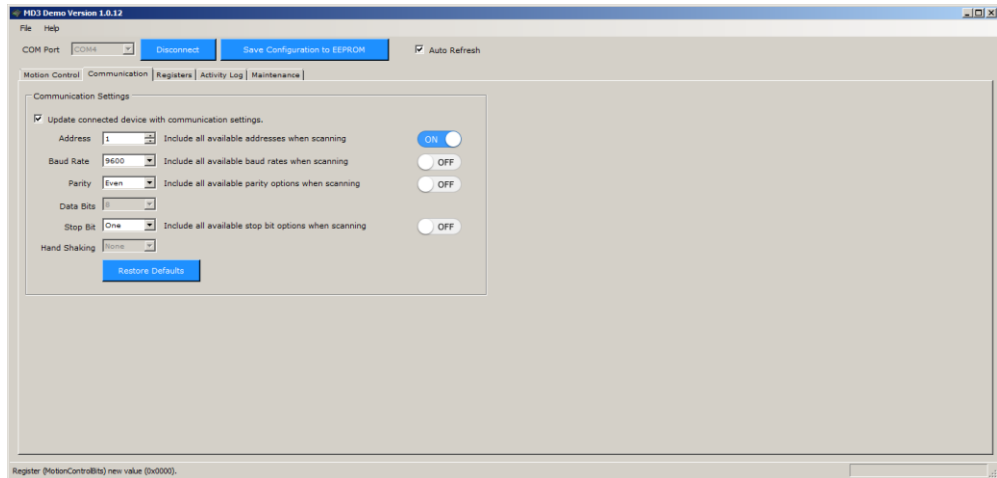
Motion Control Tab



This tab contains the motion control settings for the MD3. The user may configure the following items:

- Enable/Disable the motor drive
- Start a profile move, jog, home cycle, decelerate and stop, halt the motor
- Monitor and Clear the Fault status of the drive
- Set the Acceleration, Speed and Number of microsteps to move
- Set the motor current and microstepping fraction
- Set the current cutback timeout and current cutback level
- Configure the quadrature encoder counter
- Configure the motion control input pins on J2 Pin 4 and Pin 5.
- Configure the digital input port pins and read/write the port pins

Communication Tab

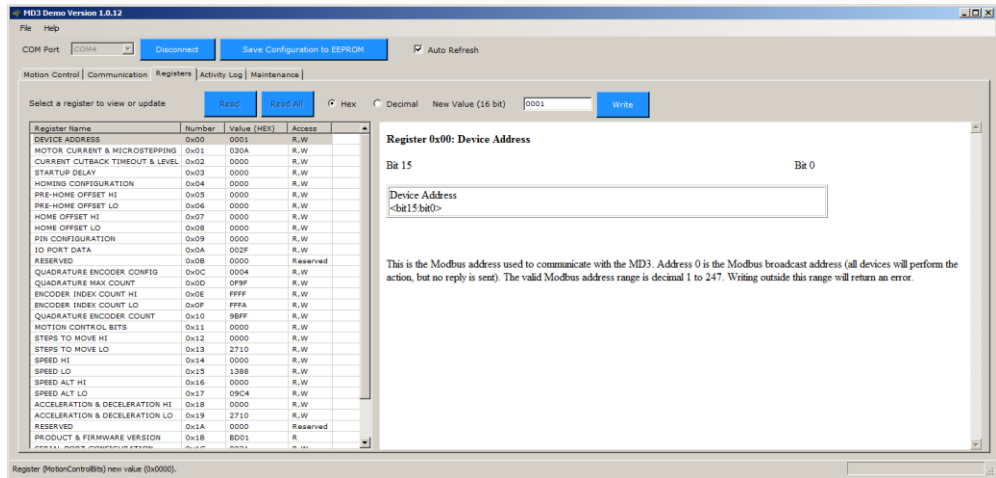


This tab contains the serial communication settings for the MD3. The pulldowns and switches are used to change the serial port settings of the host PC and the serial port scanning options.

If “Update connected device with communication settings” is checked, changing any serial port settings such as the baud rate will change the baud rate of the Host PC’s serial port **and** the MD3. If the box is unchecked, only the Host PC’s serial port parameters will be changed.

The default device address of the MD3 on the RS485 bus is Address 1. To change the address of an MD3, be sure only one device is on the bus and check “Update connected device with communication settings”. Use the “Address” dialog box to enter a new address in the range 1 to 247. Click on “Save Configuration to EEPROM” to save the settings. After different addresses are assigned to each MD3, multiple units can be connected to the same RS485 bus in parallel. To communicate with other device addresses, uncheck the “Update connected device with communication settings” box and change the address in the dialog box to match the MD3.

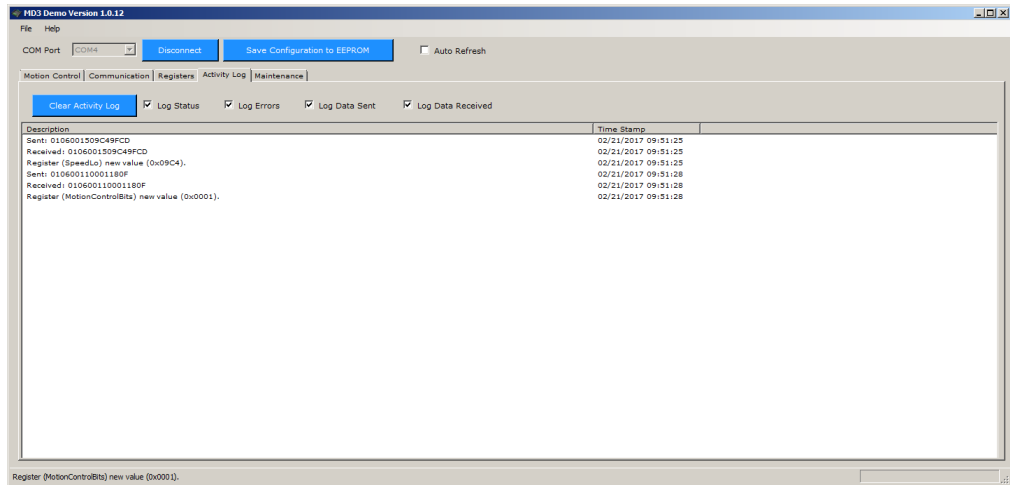
Registers Tab



This tab is used to directly read or write to the MD3 registers. The settings in the “Motion Control” and “Communication” tabs map to bits in the 32 registers in the MD3’s internal memory. These registers are accessed by sending the Modbus Read Register or Modbus Write Register commands over the RS485 bus. A host such as PLC or microcontroller can control the MD3 entirely over the RS485 bus. See Section 6 for a detailed description of the serial data format.

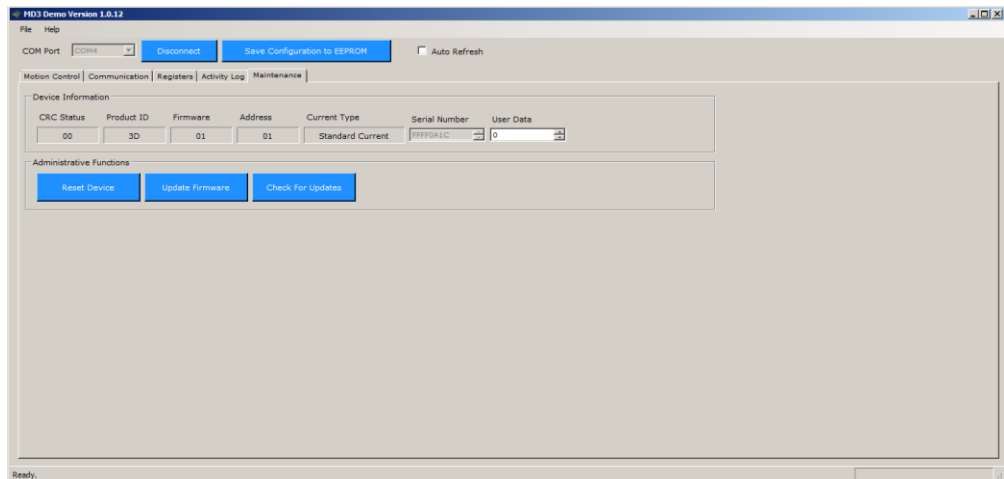
Checking “Auto Refresh” will cause the software to continuously send Modbus Register Read commands to the MD3 to update display fields such as “Driver Status” and “Digital Input” state in the “Motion Control” tab and the register contents in the “Registers” tab. This will limit performance as the drive needs to continuously process serial commands. If “Auto Refresh” is unchecked, fields such as the “Driver Status” will only update if the “Read All” button is pressed.

Activity Log Tab



The checkboxes in this tab are used to log the Modbus serial commands/replies that are exchanged between the host PC and the MD3 as the graphical user interface controls are activated. If "Auto Refresh" is checked, the log will have constant activity since the Demo software continuously reads all the registers to update the status in the "Motion Control" tab. Uncheck "Auto Refresh" to disable continuous reading of all the registers. "Log Status" is the command in human readable form. "Log Data Sent" and "Log Data Received" are the actual bytes that sent over the RS485 bus. Errors can also be logged if "Log Errors" is checked.

Maintenance Tab



This tab displays device information on the MD3 unit. "User Data" is a single 16-bit register that the user can use to store user data such as a motor axis id number. The "Reset Device" button will issue a reset command to the MD3. This will reset the drive and reload the register settings from EEPROM. The "Update Firmware" button will allow the user to choose a firmware file to update the MD3. For the fastest update, in the "Communication" tab, check the "Update connected device with communication settings" box and change the Baud Rate to 460800 bps. If the Baud Rate setting is not saved to EEPROM, after the firmware update, the MD3 will reset and the baud rate will revert to the setting in EEPROM. The baud rate in the program needs to be changed back to the value in EEPROM to communicate with the MD3 again. A firmware update takes about 6 seconds at the maximum speed. Depending on the line length and signal quality, it might not be possible to connect at 460800 bps. Check that the drive works properly at a higher speed by clicking "Read All" in the "Registers" tab and observing if any errors occur in the "Activity Log".

4. MD3 Operation

When power is applied, all MD3 16-bit registers are initialized from saved values in EEPROM memory. The LEDs will flash according to the table in Section 2. The motor current will be turned off and no input signals or serial commands will be processed until a programmable "Startup Delay" interval expires. This feature allows the MD3 to wait until other parts of the system (if applicable) are done booting. Once the startup delay expires, the MD3 can be configured to automatically start a "Home Cycle". Note that the entire homing process will take place automatically without any commands from a host.

Once the homing process is complete, the MD3 can accept Modbus commands or respond to digital inputs on connector J2.

5. MD3 Interfaces

There are 2 ways for a host to send motion commands to the MD3:

- 1) RS485 serial port using the Modbus RTU protocol on connector P1
- 2) Digital input signals on the isolated input port pins of connector J2.

5.1. RS485 Interface

The MD3 uses the Modbus RTU protocol over a RS485 serial link. The Modbus specification is available at the link: http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf . Modbus RTU is a simple command/response protocol over a serial bus. All commands to the MD3 are in the form of 16-bit register reads and writes. The MD3 has thirty-two 16-bit registers for configuration settings such as motor current and microstepping and motion commands. The following list is a simple guide to using Modbus RTU.

- The default Modbus serial parameters are: 9600 bps, 1 start bit, 8 data bits, even parity and 1 stop bit. The 8 data bits are sent LSB first. The baud rate can be changed to standard values from 4800 to 460800 bps by writing to the appropriate Modbus register
- The byte order for all 16-bit values is Big Endian (most significant byte first).
- Reading MD3 registers is done using Modbus Function Code 3 (Read Holding Registers). Writing to a register is done using Modbus Function Code 6 (Write Single Register) commands.
- Modbus device address must be in the range 0 to 247 (decimal). All MD3 units are shipped with a default address of 1. Address 0 is the Modbus broadcast address. All MD3 units on the bus will perform the register write command, but no reply is sent.
- All Modbus commands and responses have a 16-bit CRC for error detection. C source code for the CRC-16 calculation is available at the link: http://modbus.org/docs/Modbus_over_serial_line_V1_02.pdf
- Modbus RTU data is in binary format rather than ASCII, so it cannot be viewed properly on a text terminal. US Digital provides the free **MD3 Setup** and **MD3 Demo** programs which provides a graphical user interface to send configuration and motion Modbus commands to the MD3. In addition, US Digital's free **ComTool** application or the open source **Realterm** program may be used to send commands and view the serial data on a PC in hexadecimal. US Digital's ComTool application can also calculate the Modbus CRC for a command.

5.2. Digital Input Interface

The MD3 can also be controlled by digital input signals on J2 pins 2, 3, 4 and 5. These 4 input signals have isolators to prevent motor noise from coupling into the host system. The isolators should be powered by a separate +5V power supply on J2 Pin 1 and Pin 6 for the best noise isolation. Alternately, the MD3 provides a +5V output that can power the isolators.

J2 pins 4 and pin 5 are the main motion control digital inputs. These pins can be configured to be a step/direction input, go-stop/direction input, as well as other motion control modes such as starting a profile move on a falling edge and home on a rising edge, etc. There are currently a total of 10 possible motion control modes of these 2 pins. In addition to control inputs, the J2 connector also has pins to connect the A/B/index signals from a quadrature encoder. The count from the encoder can be read over the serial port by a host to monitor motor position.

6. MD3 RS485 Serial Interface Protocol (Modbus RTU)

This section describes the data format of the MD3 serial interface. There are only 2 Modbus RTU commands used on the serial bus: Read Holding Register and Write Single Register.

6.1. Read Holding Register (Function Code 3)

The “Read Holding Register” command is used to read the contents of one or more 16-bit registers in the MD3. All 32 registers can be read with a single command. The host sends the following 8 bytes to the MD3:

	Parameter	# bytes	Notes
Command	Address	1	Destination address of MD3
	0x03	1	Modbus Function Code = 0x03
	Starting register to read	2	Big-endian order. Ex. Register 0x0003 (Startup Delay) is sent 0x00 first, then 0x03
	Number of registers to read (hex)	2	0x0001 to read one register. 0x000a to read 10 consecutive registers.
	CRC-16	2	CRC-16 of the first 6 bytes in the command

The addressed MD3 will respond with the following data packet which contains the contents of the register(s):

	Parameter	# bytes	Notes
Response	Address	1	Address of responding MD3
	0x03	1	Modbus Function Code = 0x03
	Byte Count	1	2*(Number of Registers Read). Ex. 0x0a if 5 registers are read.
	Register N	2	Read Register data. 2 bytes if one register is read. 10 bytes if 5 registers are read
	Register N+1	2	
	Register N+2	2	
	...	2	
CRC-16	2	CRC-16 of all bytes in the response.	

If the CRC of the command is invalid, no action is taken. If the CRC is valid but there is an error in the command, the response is:

	Parameter	# bytes	Notes
Error Response	Address	1	Address of responding MD3
	0x83	1	Modbus Error Function Code = 0x83
	Exception Code	1	0x01: invalid function code 0x02: invalid register address
	CRC-16	2	CRC-16 of the first 3 bytes in the response.

6.2. Write Single Register (Function Code 6)

The “Write Single Register” command is used to write data to a single 16-bit register in the MD3. Some parameters in the MD3 are 32-bit quantities. In these cases, two “Write Single Register” commands in any order are needed to update the 32-bit parameter. The host sends the following 8 bytes to the MD3:

	Parameter	# bytes	Notes
Command	Address	1	Destination address of MD3
	0x06	1	Modbus Function Code = 0x06
	Register to Write	2	Big-endian order. Ex. Register 0x0003 (Startup Delay) is sent 0x00 first, then 0x03
	Data to Write	2	Big-endian order.
	CRC-16	2	CRC-16 of the first 6 bytes in the command

The addressed MD3 will respond with a duplicate of the Command packet to acknowledge the write:

	Parameter	# bytes	Notes
Response (identical to command)	Address	1	Address of responding MD3
	0x06	1	Modbus Function Code = 0x06
	Register to Write	2	Big-endian order. Ex. Register 0x0003 (Startup Delay) is sent 0x00 first, then 0x03
	Data to Write	2	Big-endian order.
	CRC-16	2	CRC-16 of the first 6 bytes in the response.

If the CRC of the command is invalid, no action is taken. If the CRC is valid but there is an error in the command, the response is:

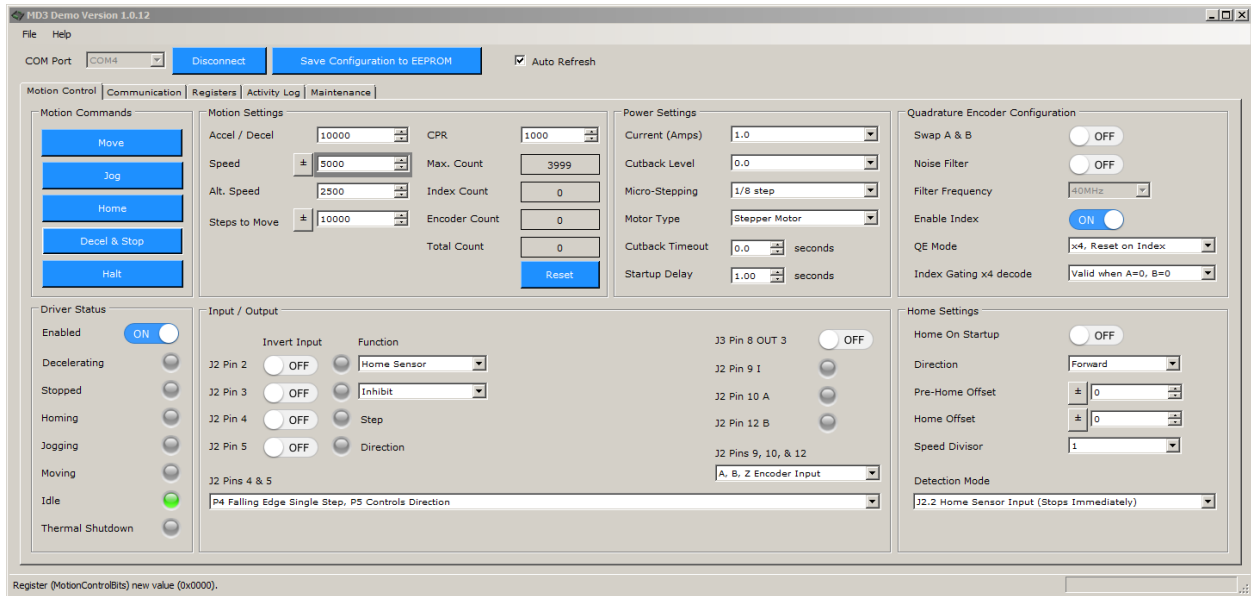
	Parameter	# bytes	Notes
Error Response	Address	1	Address of responding MD3
	0x86	1	Modbus Error Function Code = 0x86
	Exception Code	1	0x01: invalid function code 0x02: invalid register address
	CRC-16	2	CRC-16 of the first 3 bytes in the response.

7. MD3 Register Description

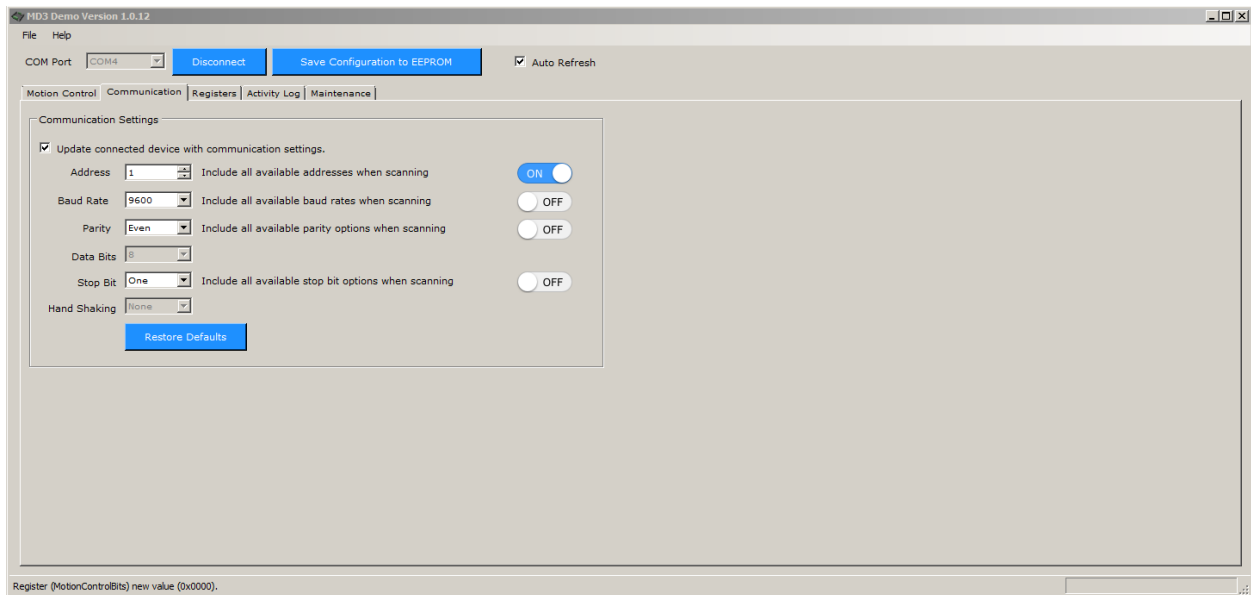
Modbus Register Address	Register Name	Initial Value
0x0000	DeviceAddress	0x0001
0x0001	MotorCurrentAndMicrostepping	0x030a
0x0002	CurrentCutbackSettings	0x0000
0x0003	StartupDelay	0x0000
0x0004	HomingConfiguration	0x0000
0x0005	PreHomeOffsetHigh	0x0000
0x0006	PreHomeOffsetLow	0x0000
0x0007	HomeOffsetHigh	0x0000
0x0008	HomeOffsetLow	0x0000
0x0009	PinFunctionAndPolarity	0x0000
0x000a	I/OPortData	0x000f
0x000b	Reserved	-
0x000c	QuadratureEncoderConfiguration	0x0005
0x000d	QuadratureEncoderMaxCount	0x0f9f
0x000e	EncoderIndexCountHigh	0x0000
0x000f	EncoderIndexCountLow	0x0000
0x0010	QuadratureEncoderCount	0x0000
0x0011	MotionControl	0x0000
0x0012	MoveStepsHigh	0x0000
0x0013	MoveStepsLow	0x2710
0x0014	SpeedHigh	0x0000
0x0015	SpeedLow	0x1388
0x0016	AlternateSpeedHigh	0x0000
0x0017	AlternateSpeedLow	0x09c4
0x0018	AccelDecelHigh	0x0000
0x0019	AccelDecelLow	0x2710
0x001a	EEPROMControl	0x9a00
0x001b	ProductInformation	0xbd01
0x001c	SerialPortConfiguration	0x0021
0x001d	UserRegister	0x0000
0x001e	SerialNumberHigh	set by manufacturer
0x001f	SerialNumberLow	set by manufacturer

Unless otherwise stated, all registers and bit fields are readable and writable by the user.

The initial configuration of the MD3 registers in GUI form in **MD3 Demo** are:



The initial serial port settings of the MD3 are:



7.1. Register 0x00: DeviceAddress

Bit 15

Bit 0

Device Address <bit15:bit0>

This is the Modbus address used to communicate with the MD3. Address 0 is the Modbus broadcast address (all devices will perform the action, but no reply is sent). The valid Modbus address range is decimal 1 to 247. Writing outside this range will return an error.

7.2. Register 0x01: MotorCurrentAndMicrostepping

Bit 15

Bit 0

Unused <bit15:14>	Motor Mode <bit13>	Motor Control Parameter <bit12:bit8>	Motor Phase Current Parameter <bit7:bit0>
----------------------	-----------------------	---	--

Motor Mode	Motor Type and control
0	Stepper Motor, open loop
1	DC Motor PWM, open loop (jog mode only)

In DC Motor PWM modes, the brushed DC motor is connected to Phase B+/B-. Leave Phase A+/A- unconnected.

Motor Control Parameter	Stepper Motor Microstepping	DC Motor PWM frequency (PWM duty cycle steps)
0	Full step, 71% current	500 Hz (± 40000)
1	half step	1 kHz (± 20000)
2	1/4 step	2 kHz (± 40000)
3	1/8 step	5 kHz (± 16000)
4	1/16 step	10 kHz (± 8000)
5	1/32 step	16 kHz (± 5000)
6	1/64 step	20 kHz (± 4000)
7	1/128 step	25 kHz (± 3200)
8	1/256 step	40 kHz (± 2000)

Due to the filtering effect of the motor's resistance/inductance and load inertia, the RPM may not change for small duty cycle changes. The DC Motor PWM mode is not recommended for pancake/printed armature motors since they have extremely low inductance (< 100 μ H). The motor may exhibit erratic behavior due to the large ripple current when driven by PWM.

Motor Phase Current Param.	High Current Version, Peak Amps	Low Current Version, Peak Amps
0x00	0	0
0x01	0.1	0.05
0x02	0.2	0.1
0x03	0.3	0.15
0x04	0.4	0.2
0x05	0.5	0.25
0x06	0.6	0.3
0x07	0.7	0.35
0x08	0.8	0.4
0x09	0.9	0.45
0x0a	1	0.5
0x0b	1.1	0.55
0x0c	1.2	0.6
0x0d	1.3	0.65
0x0e	1.4	0.7
0x0f	1.5	0.75
0x10	1.6	0.8
0x11	1.7	0.85
0x12	1.8	0.9

Motor Phase Current Param.	High Current Version, Peak Amps	Low Current Version, Peak Amps
0x13	1.9	0.95
0x14	2	1.0
0x15	2.5	1.05
0x16	3	1.1
0x17	3.5	1.15
0x18	4	1.2
0x19	4.5	1.25
0x1a	5	1.3
0x1b	5.5	1.35
0x1c	6	1.4
0x1d	6.5	1.45
0x1e	7	1.5
0x1f		1.55
0x20		1.6
0x21		1.65
0x22		1.7
0x23		1.75
0x24		1.8

The High Current Version can be set to currents below 0.5 Amps, but motor operation may be noisier. The Low Current Version is recommended for currents below 0.5A

7.3. Register 0x02: CurrentCutbackSettings

Bit 15

Bit 0

Current Cutback Timeout <bit15:bit8>	Current Cutback Level <bit7:bit0>
---	--------------------------------------

Current cutback timeout: This field is the timeout interval in 0.1 second increments. Maximum timeout is 25.5 seconds. If there is no command or input change within the timeout period, the motor current is changed from the normal current to the "Current Cutback level". Set = 0 for no current cutback. The red LED D5 will turn on when in current cutback mode. This feature is typically used to prevent motor overheating at higher current settings when the motor is not moving.

Current Cutback Level: The current cutback level uses the same table of values as the motor phase current field of Register 0x01. The current cutback level is usually set to lower than the normal operating current set in Register 0x01. It can be set to the same as the motor phase current to ignore the current cutback function, but the LED D5 will still turn on after the timeout. This field is ignored if the timeout = 0 (no current cutback).

7.4. Register 0x03: StartupDelay (unsigned, 16-bits)

Bit 15

Bit 0

Startup Delay <bit15:bit0>

Startup delay: On power up, there is an approximately 2.5 second delay before the motor current is turned on and the drive starts a home cycle (if enabled) and responds to inputs. This field specifies an additional delay in increments of 0.010 seconds. If this field is set to 0, the startup delay is 2.5 seconds. A value of 250 results in a startup delay of 2.5 seconds + 250 * .010 seconds = 5 seconds. The maximum value is 65435.

7.5. Register 0x04: HomingConfiguration

Bit 15

Bit 0

Unused <bit15:bit8>	Homing Speed Divisor <bit7:bit5>	Homing Direction <bit4>	Homing Mode <bit3:bit1>	Home on Startup <bit0>
------------------------	-------------------------------------	----------------------------	----------------------------	---------------------------

Homing speed divisor:

The speed during homing is the normal move speed set by **Register 0x14/0x15: Speed High/Low** divided by 2 ^ (Homing Speed Divisor). The acceleration is not divided down. So the homing speed is the normal move speed divided by either 1, 2, 4, 8, 16, 32, 64, or 128. 0 means home speed is the same as move speed.

Homing direction: Sets the motor direction for homing

Homing Mode	Description
0	J2 pin 2 is the active low home sensor flag input. HOMEOFFSET and PREHOMEOFFSET are in steps. Register 0x09 can be used to invert this pin so the sensor input is rising edge triggered. The switch input is sampled about every 4 microseconds. Motor will stop immediately (no deceleration) when the flag input is activated.
1	Same behavior as Value = 0 except that the motor will decelerate and stop when the flag input is activated.
2	Homing by encoder active high encoder index pulse on J2 pin 9. HOMEOFFSET and PREHOMEOFFSET are in steps. Motor will stop when the index pulse is detected (no deceleration). J2 pin 2 can be configured to be either the LOWCURRENT or SPEEDSEL input in Register 0x09: Pin Function and Polarity .

3	Same behavior as Value = 2 except that the motor will decelerate and stop when the index pulse is detected. J2 pin 2 can be configured to be either the LOWCURRENT or SPEEDSEL input in Register 0x09: Pin Function and Polarity .
---	---

Home on Startup	Description
0	Do not execute home cycle on startup. A home cycle can be manually started by Modbus command (see Register 0x11: Motion Control) or J2 pin 3 if configured as a HOME input
1	Execute home cycle on startup if "Motor Mode" field in Register 0x01: Motor Current and Microstepping is 0 (Stepper Mode, open loop) and Disable bit = 0 in Register 0x11: Motion Control . Homing on startup is never done in the DC Motor PWM modes.

7.6. Register 0x05/0x06: PreHomeOffsetHigh / PreHomeOffsetLow (signed, 32-bits)

Bit 15

Bit 0

Register 0x05: PreHomeOffsetHigh <bit15:bit0>
--

Bit 15

Bit 0

Register 0x06: PreHomeOffsetLow <bit15:bit0>

The PreHomeOffset is a signed, 32-bit integer. The upper word is in Register 0x05 and the lower word is in Register 0x06. If a home cycle is started, the PreHomeOffset is the number of steps the stepper will move before checking the flag input or encoder index signal. Set = 0 for no PreHomeOffset.

This offset is typically used with switch input homing and linear steppers. If the motor has moved beyond the flag sensor position, the motor will jam if it tries to home in the normal direction. By adding a small pre-home offset in the opposite direction, if the motor has moved past the switch input, it will be in the correct position for normal homing. The speed used for the PreHomeOffset move is the normal speed specified in **Register 0x14/0x15: Speed High/Low**. Once the PreHomeOffset move is completed, the stepper will start moving until either the sensor flag input or the encoder index pulse is detected.

7.7. Register 0x07/0x08: HomeOffsetHigh / HomeOffsetLow (signed, 32-bits)

Bit 15

Bit 0

Register 0x07: HomeOffsetHigh <bit15:bit0>

Bit 15

Bit 0

Register 0x08: HomeOffsetLow <bit15:bit0>
--

The HomeOffset is a signed, 32-bit integer. The upper word is in Register 0x07 and the lower word is in Register 0x08. The HomeOffset is the number of additional steps the motor will move after it has stopped homing due to the sensor flag input or encoder index pulse being detected. Set = 0 for no HomeOffset. The speed used for the HomeOffset move is the normal speed specified in **Register 0x14/0x15: Speed High/Low**.

7.8. Register 0x09: PinFunctionAndPolarity

This register configures the pins on the 12-pin connector, J2. The pin-out of J2 is:

J2 Pin	Description
1	Isolator GND input (must be connected to J2 Pin 8 or external GND)
2	LOW CURRENT, SPEEDSEL, or HOME SENSOR input (isolated, 5k pullup)
3	INHIBIT or HOME input (isolated, 5k pullup. Also clears non-thermal drive fault state)
4	STEP or GO-STOP input (isolated, 5k pullup)
5	DIR input (isolated, 5k pullup)
6	Isolator +5V input (must be connected to J2 Pin 7 or external +5V source)
7	+5V out
8	GND
9	Encoder Index/general purpose input or general purpose digital output
10	Encoder A/general purpose input or general purpose digital output
11	+5V out
12	Encoder B/general purpose input or general purpose digital output

The descriptions below assume that all input pins (J2 Pins 2, 3, 4, 5) are in the default mode, which is falling edge triggered with a weak internal pullup. The input pins can be individually switched to rising edge triggered by writing to the Polarity bit field of this register. The weak pullup is still present so the input will become active immediately if the pin is floating and the polarity is inverted.

The state of J2 Pins 2,3,4,5 can always be read by reading **Register 0x0a: I/O Port Data**. The general purpose outputs J2 pin 9, 10, 12 are set/cleared by writing to **Register 0x0a**.

Bit 15

Bit 0

Unused <bit15:bit13>	Enable encoder <bit12>	J2 pin 2 function <bit11:bit10>	J2 pin 3 function <bit9:bit8>	J2 pin 4/5 function <bit7:bit4>	J2 pin 2 polarity <bit3>	J2 pin 3 polarity <bit2>	J2 pin 4 polarity <bit1>	J2 pin 5 polarity <bit0>
-------------------------	---------------------------	------------------------------------	----------------------------------	------------------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Enable encoder	Description
0	J2 pins 9,10,12 are encoder Index, A, B inputs (pins can be read as general purpose inputs as well)
1	J2 pins 9,10,12 are general purpose digital outputs

J2 pin 2 function:

If the “Homing Mode” field in **Register 0x04: Homing configuration** is binary 000 or 001 (homing by sensor flag input), J2 pin 2 is always the input for the homing switch and the 2 bits in this field are ignored. If the “Homing Mode” field is binary 010 or 011 (homing by encoder index), the J2 pin 2 function bits are:

Value	Description
0	J2 pin 2 is the LOWCURRENT signal - the motor current will be reduced to the Current Cutback level (Register 0x02) on the falling edge of the input.
1	J2 pin 2 is the SPEEDSEL signal. This will select either the SPEED (input high) or SPEED_ALT (input low) register (Register 0x14/0x15, 0x16/0x17) for the move speed. If this pin changes state during a Jog move, the motor will ramp to the new speed. For a Profile move, the pin is sampled once before the move starts. Pin state changes are ignored while the Profile move is happening.
2	invalid
3	invalid

J2 pin 3 function:

This field configures J2 pin 3 as either an INHIBIT, INHIBIT_MODBUS_MOTION or a HOME input as follows:

Value	Description
0	J2 pin 3 is INHIBIT – disables drive by setting motor phase current to 0 amps on the falling edge of the input. The currently running motion profile still runs but the motor will not move again until inhibit goes high.
1	J2 pin 3 is HOME input - executes a Home cycle on the falling edge of the input.
2	J2 pin 3 is INHIBIT_MODBUS_MOTION - currently running motion is stopped and all MODBUS_CONTROL register bits except for disable/enable are ignored after the falling edge of the input. The motor current is maintained so motor position is not lost. J2 pins 2/3/4/5 functions work normally.
3	invalid

Activating the INHIBIT, INHIBIT_MODBUS_MOTION or HOME input will also clear a non-thermal Fault state of the drive. If the drive is in Fault state due to overheating, it will not clear until the MD3 cools sufficiently.

J2 pin 4, 5 function:

These 2 input pins can be configured to control one of ten different motion modes. The simplest motion mode is the step/direction mode, where Pin 4 is the motor step pulse and Pin 5 controls the step direction. Another simple motion mode is the go-stop/direction mode where Pin 4 starts and stops motor jogging and Pin 5 controls the jog direction. The other motion modes are combinations of a "Profile", "Jog", or "Home" move.

A "Profile" move is defined by the steps/speed/acceleration parameters in Registers 0x12-0x19. The motor will move the specified number of microsteps, with a linear speed ramp up from zero speed, followed by a constant speed move, and ending with a linear speed ramp down to zero speed. If the total number of microsteps is insufficient to achieve the specified speed at the specified acceleration, the motor will always move the correct number of steps, but it will ramp up to the max speed possible and then ramp down. The ramp up and ramp down time will be the same. See Figure 2.

While the MD3 is moving the stepper motor, Modbus commands can still be sent and processed. Sending many serial commands while the motor is moving will cause small timing deviations in the calculated step profile generated by the drive. In many applications, this should not be an issue. However, if the motor is accelerating or moving a heavy load at maximum step speed, the motor may stall. To avoid stalling in this case, do not flood the serial port with commands while the motor is moving. J3 pin 7 of the MD3 is the "Busy" digital output that can be used to determine when the drive is done moving instead of continuously reading the drive status over Modbus.

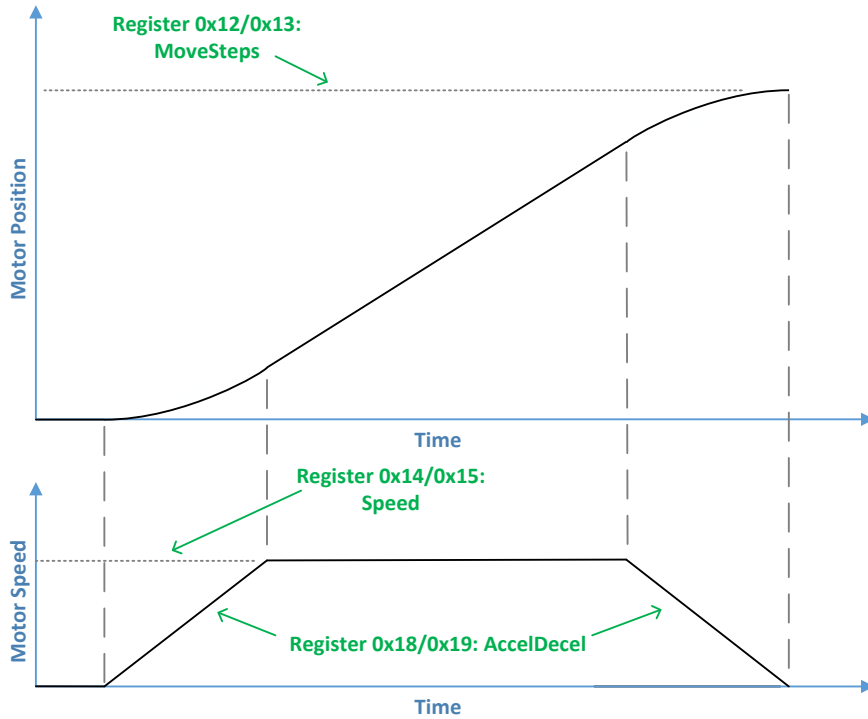


Figure 2. Profile move

A “Jog” move is defined by the speed/acceleration parameters in Registers 0x14-0x19. The motor will linearly ramp up or ramp down from the current speed to a new speed as long as the motor moves in the same direction. The motor will move continuously at the new speed until stopped. The MoveSteps parameter in Registers 0x12/0x13 is not used. See Figure 3.

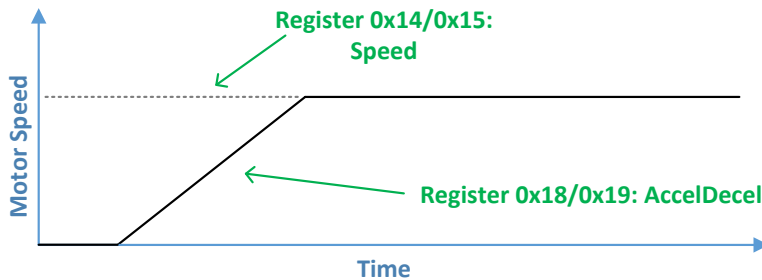


Figure 3. Jog

A “Home” cycle is defined by Registers 0x04-0x08. A Home cycle consists of 3 parts: 1) an optional pre-home offset move, 2) movement until a switch input (J2 pin 2) or encoder index pulse (J2 pin 9) is detected and 3) an optional home offset move. Once a Home cycle is started, state changes on J2 pin 4 and pin 5 are ignored until the Home cycle is complete to prevent them from interfering with the home cycle. Modbus serial commands still work during the home cycle.

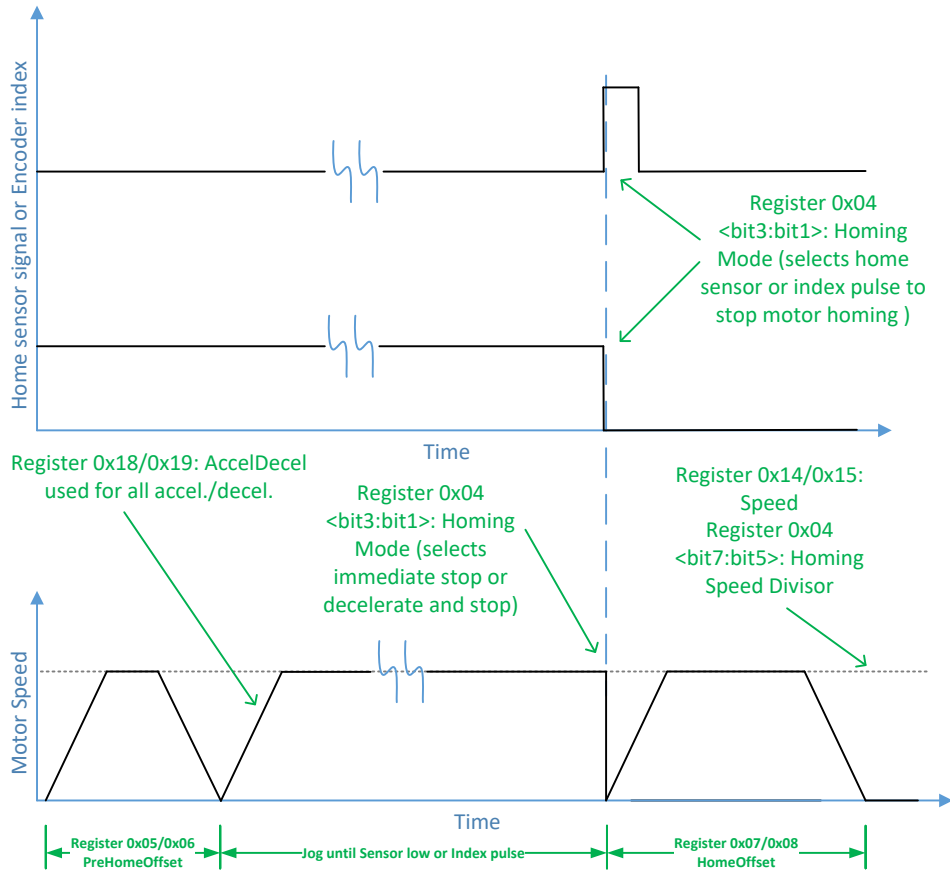


Figure 4. Home cycle

In DC motor PWM mode, J2 pin 4/5 function fields = 0110 or 0111 are the only allowed options. The motion modes are shown in the table below.

Bits	J2 pin 4	J2 pin 5	Description
0000	Step	Dir	Motor moves one step per falling edge on Pin 4. The direction is determined by the state of Pin 5 when the falling edge occurs. Maximum step frequency is 250 kHz.
0001	Move Profile	Dir	Motor will execute a Profile move on the falling edge of Pin 4. Further edges on P4 are ignored while the motor is moving. The direction of the move is determined by state Pin 5 when the falling edge occurs.
0010	Move Profile forward/backward	-	Motor will execute a Profile move on the falling edge of Pin 4. The rising edge on Pin 4 will execute the same Profile move in the opposite direction. Pin 5 is ignored.
0011	Move Profile forward/home	-	Motor will execute a Profile move on the falling edge of Pin 4. The rising edge on Pin 4 will execute a Home cycle. Pin 5 is ignored.
0100	Move Profile forward	Move Profile backward	Motor will execute a Profile move on the falling edge of Pin 4. Motor will execute a Profile move in the opposite direction on the falling edge of Pin 5.

0101	Move Profile forward	Home	Motor will execute a Profile move on the falling edge of Pin 4. Motor will execute a Home cycle on the falling edge of Pin 5.
0110	Jog	Dir	Motor will ramp up and move continuously when P4 is low. Motor will ramp down and stop when P4 is high. Pin 5 will cause the motor to ramp down and reverse the jog direction.
0111	Jog forward	Jog backward	Motor will ramp up and move continuously when P4 is low. Motor will ramp up and move in the reverse direction continuously when P5 is low. If P4 and P5 are high the motor will ramp down and stop.
1000	-	-	The motor will continuously run a Profile move forward and backward. The motion starts immediately. Stop the motion by changing the J2 pin 4/pin5 function field to step/dir (0000). The sign of the MoveStepsHigh/MoveStepsLow registers will be negative if the motion is stopped when the motor is moving backward.
1001	-	-	The motor will continuously run a Profile move forward and a Home cycle. The motion starts immediately. Stop the motion by changing the J2 pin 4/pin5 function field to step/dir (0000).
1111	-	-	No function on pin4 or pin5.

J2 pin 2, 3, 4, 5 polarity bits	Description
0	Input pins are active low. There is an internal weak pullup on all input pins
1	Input pins are active high. The internal weak pullup is still active.

7.9. Register 0x0a: I/OPortData

Bit 15

Bit 0

Unused <bit15:bit8>	J3 pin 8 output <bit7>	J2 pin 9 output <bit6>	J2 pin 10 output <bit5>	J2 pin 12 output <bit4>	J2 pin 2 input <bit3>	J2 pin 3 input <bit2>	J2 pin 4 input <bit1>	J2 pin 5 input <bit0>
------------------------	------------------------------	------------------------------	-------------------------------	-------------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

Writing to this register will write to the general purpose output pins (if enabled).

Reading from this register will return the input port state and the output port value.

7.10. Register 0x0b: Reserved (Do not use)

7.11. Register 0x0c: QuadratureEncoderConfiguration

Bit 15

Bit 0

Unused <bit15:bit10>	Noise Filter Enable <bit9>	A/B/I Input Noise Filter Frequency <bit8:bit6>	Index Gating <bit5:bit4>	SWPAB <bit3>	QE Mode <bit2:bit1>	Enable Index <bit0>
-------------------------	-------------------------------	---	-----------------------------	-----------------	------------------------	------------------------

Maximum input frequency on A/B quadrature input is 3.3 MHz.

Noise Filter Enable	Description
0	Disable noise filter
1	Enable noise filter

A/B/Z Input Noise Filter Frequency:

The noise filter on the A/B quadrature inputs will ignore pulses less than 3 clock periods in duration.

Value	Description
0	40 MHz
1	20 MHz
2	10 MHz
3	2.5 MHz
4	1.25 MHz
5	625 kHz
6	312.5 kHz
7	156.25 kHz

Index Gating:

Index gating with x4 decode:

Value	Description
0	index valid when B=0, A=0
1	index valid when B=0, A=1
2	index valid when B=1, A=0
3	index valid when B=1, A=1

Index gating with x2 decode:

Value	Description
0	index valid when A=0
1	index valid when A = 1
2	index valid when B = 0
3	index valid when B =1

SWPAB	Description
0	Encoder A/B inputs are not swapped
1	Encoder A/B inputs are swapped

QE Mode	Description
0	x2 quadrature decoding, counter reset by index pulse
1	x2 quadrature decoding, counter reset when match with Register 0x0d: Quadrature Encoder Max Count , index ignored
2	x4 quadrature decoding, counter reset by index pulse
3	x4 quadrature decoding, counter reset when match with Register 0x0d: Quadrature Encoder Max Count , index ignored

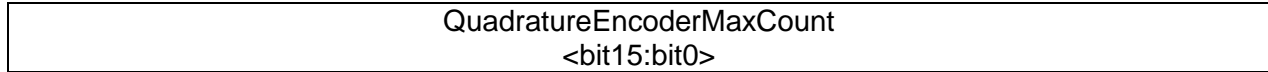
Enable Index:

Value	Description
0	Do not clear counter on index pulse
1	Clear counter on index pulse. This applies only if QE Mode = 10 or 00

7.12. Register 0x0d: QuadratureEncoderMaxCount (unsigned, 16-bits)

Bit 15

Bit 0



CPR is the Cycles Per Revolution of the external incremental encoder.

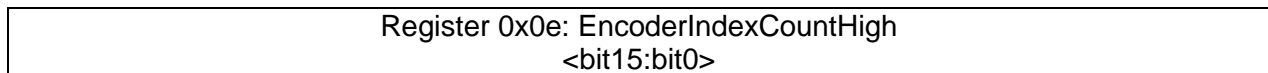
With x2 quadrature decoding, set this register equal to $(2 * \text{CPR}) - 1$

With x4 quadrature decoding, set this register equal to $(4 * \text{CPR}) - 1$

7.13. Register 0x0e/0x0f: EncoderIndexCountHigh / EncoderIndexCountLow (signed, 32-bits)

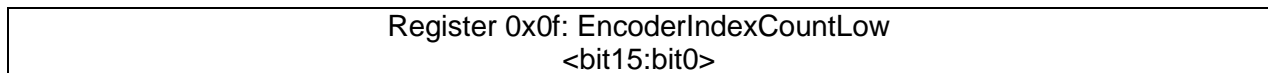
Bit 15

Bit 0



Bit 15

Bit 0



EncoderIndexCount is a signed, 32-bit integer. The upper word is in Register 0x0e and the lower word is in Register 0x0f.

This register contains the current count of index pulses. The A/B phase at the time of index is used to determine if the count should be increased or decreased. For example, 5 CW rotations will increase the count by 5 and 3 CCW rotations will decrease the count by 3.

7.14. Register 0x10: QuadratureEncoderCount (unsigned, 16-bits)

Bit 15

Bit 0

Register 0x10: QuadratureEncoderCount <bit15:bit0>

This register contains the current value of the quadrature counter.

7.15. Register 0x11: MotionControl

Bit 15

Bit 0

Fault <bit15>	Unused <bit14:bit6>	Disable <bit5>	Decel-stop <bit4>	Stop <bit3>	Home <bit2>	Jog <bit1>	Move <bit0>
------------------	------------------------	-------------------	----------------------	----------------	----------------	---------------	----------------

Setting one of the bits high in this register will command the MD3 to perform the specified function. Do not set more than one bit high at a time. All bits except Disable are self-clearing – the bit is automatically cleared when the drive completes the specified function.

Fault:

This bit will be = 1 if the MD3 is in a Fault state due to overheating, overcurrent or other error conditions. In the Fault state, all serial port commands will appear to work normally, except that the motor will not move. The open drain BUSY output signal on J3 pin 7 will be switched to GND (the output FET is on) in the Fault state. The Fault can be cleared by activating the INHIBIT or HOME signal (J2 pin 3). The Fault state can also be cleared by writing a "1" to the Fault bit (bit 15). If the Fault is due to thermal shutdown, the Fault bit will not be cleared until the drive cools sufficiently.

Disable:

If this bit is 1, the motor current is set to 0 Amps. Set this bit = 0 for normal motor operation. The state of the Disable bit is saved when registers are saved to EEPROM. All other motion bits are cleared. If Disable = 1 when registers are saved to EEPROM, the MD3 will remain disabled after boot and a home cycle will not occur even if Home on Startup = 1 in **Register 0x04:**

Homing Configuration.

Decel-stop (self-clearing) = 1 will cause the motor to decelerate and stop with motor current still active. The deceleration is specified in **Register 0x18/0x19: AccelDecelHigh / AccelDecelLow**.

Stop (self-clearing) = 1 will cause the motor to stop immediately with motor current still active.

Home (self-clearing) = 1 will execute a home cycle.

Jog = 1 will cause the stepper to perform a Jog move. A Jog move consists of a linear ramp up or ramp down from the current speed to a new speed as long as the motor moves in the same direction. The motor will continuously move at the new speed until stopped. The acceleration or deceleration is specified in **Register 0x18/0x19: AccelDecelHigh / AccelDecelLow**. The speed is specified by **Register 0x14/0x15: SpeedHigh / SpeedLow** or **Register 0x16/0x17: AlternateSpeedHigh / AlternateSpeedLow**. Negative values of speed will rotate the motor in the opposite direction.

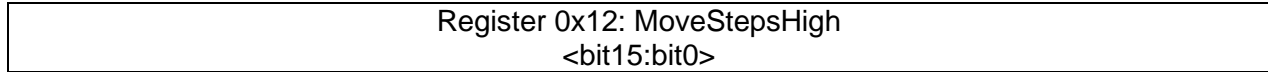
Move (self-clearing) = 1 will cause the stepper to perform a Profile move. A Profile moves the motor a specified number of microsteps, with a linear speed ramp up from zero speed, followed by a constant speed move, and ending with a linear speed ramp down to zero speed. The total number of microsteps moved is specified in **Register 0x12/0x13: MoveStepsHigh / MoveStepsLow**. Negative values of move steps will move the motor in the opposite direction. The acceleration or deceleration is specified in **Register 0x18/0x19: AccelDecelHigh / AccelDecelLow**. The speed is specified by **Register 0x14/0x15: SpeedHigh / SpeedLow** or **Register 0x16/0x17: AlternateSpeedHigh / AlternateSpeedLow**. The absolute value of the SpeedHigh / SpeedLow registers is used for the speed in a Profile move. If the total number of microsteps is insufficient to achieve the specified speed at the specified acceleration, the motor will still move the correct number of steps, but it will ramp up to the max speed possible and then ramp down. The ramp up and ramp down time will be the same.

Since we are approximating a linear ramp up of motor speed with discrete steps, the final velocity may not match the Move Speed specified in registers 0x14/0x16(SpeedHi/SpeedLo) exactly, but the Number of Steps moved will always be correct.

7.16. Register 0x12/0x13: MoveStepsHigh / MoveStepsLow (signed, 32-bits)

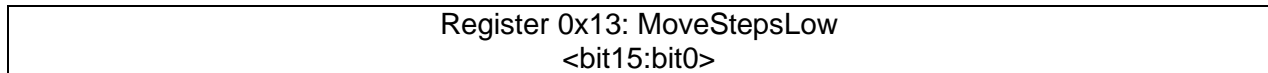
Bit 15

Bit 0



Bit 15

Bit 0



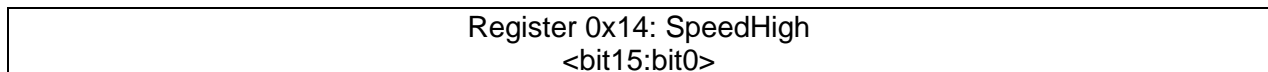
This pair of registers specifies the total number of microsteps to move in a Profile move. A Profile move consists of a linear speed ramp up from zero speed, followed by a constant speed move, and ending with a linear speed ramp down to zero speed. A negative number will rotate the motor in the opposite direction. If set to 0, the motor will not move.

If the total number of microsteps is insufficient to achieve the specified speed at the specified acceleration, the motor will still move the correct number of steps, but it will ramp up to the max speed possible and then ramp down. The ramp up and ramp down time will be the same.

7.17. Register 0x14/0x15: SpeedHigh / SpeedLow (signed, 32-bits)

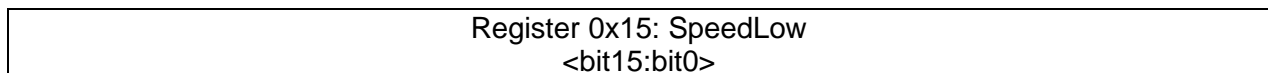
Bit 15

Bit 0



Bit 15

Bit 0



Stepper Mode:

The range of speed values in stepper mode is ± 40000 microsteps/second.

This pair of registers specifies the movement speed in microsteps/second when performing a Profile move or Jog move. A Jog move consists of a linear ramp up or ramp down from the current speed (which can be non-zero) to a new speed). In a Jog move, negative values of speed will rotate the motor opposite direction. In a Profile move, the rotation direction is

determined by the sign of the MoveStepsHigh / MoveStepsLow register. The absolute value of the SpeedHigh / SpeedLow registers is used for the speed in a Profile move.

The SpeedHigh / SpeedLow can be changed while jogging and the motor will ramp to the new speed. In a Profile move, changing the acceleration while the motor is moving will cause the motor to stop. Speed changes while the motor is performing a Profile move are ignored.

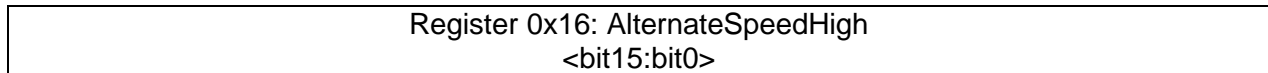
DC Motor PWM Mode:

The SpeedHigh / SpeedLow registers contain the PWM “on” time in counts. The sign of the register value determines the rotation direction. The range of duty cycle values and the PWM frequency is specified by the “Motor Control Parameter” field of **Register 0x01: Motor Current and Microstepping**. For example, if the Motor Control Parameter field is 0x06, the PWM frequency is 16 kHz and -5000 is the maximum speed (100% duty cycle) in one rotation direction and +5000 (100% duty cycle) is the maximum speed in the opposite rotation direction. A value of 2500 will result in a 50% duty cycle PWM waveform.

7.18. Register 0x16/0x17: AlternateSpeedHigh / AlternateSpeedLow (unsigned, 32-bits)

Bit 15

Bit 0



Bit 15

Bit 0



Register 0x04: Homing Configuration and **Register 0x09: Pin Function and Polarity** can be used to configure J2 pin 2 as the SPEEDSEL input. If J2 pin 2 = high, **Register 0x14/0x15: SpeedHigh / SpeedLow** is used for the speed value. If J2 pin 2 = low, **Register 0x14/0x15: AlternateSpeedHigh / AlternateSpeedLow** is used for the speed value. The absolute value of the desired alternate speed is stored in this register. The sign of the alternate speed is always the same as the sign of the SpeedHigh / SpeedLow register. The range of values for the alternate speed register is 0 to 40000 microsteps/second.

When jogging, the speed will be ramped up or down to the selected speed register setting depending on the state of J2 pin 2. For Profile moves, the state of J2 pin 2 is read at the time of

the Move command to determine which speed register to use. Changing the state of J2 pin 2 while a Profile move is in progress will change the Profile move to a Jog move.

7.19. Register 0x18/0x19: AccelDecelHigh / AccelDecelLow (unsigned, 32-bits)

Bit 15

Bit 0

Register 0x18: AccelDecelHigh <bit15:bit0>

Bit 15

Bit 0

Register 0x19: AccelDecelLow <bit15:bit0>
--

This pair of registers specifies the acceleration or deceleration in microsteps/second² for Profile or Jog moves.

7.20. Register 0x1a: EEPROMControl (register not saved to EEPROM)

Bit 15

Bit 0

CRC status <bit15:bit8>	Unused <bit7:bit2>	Read Registers from EEPROM <bit1>	Write Registers to EEPROM <bit0>
--------------------------------	---------------------------	---	---

CRC Status: This field contains CRC status (0x9a = pass, 0xf1 = fail) from the last time the registers were read from EEPROM using <bit1>, "Read Registers from EEPROM". This field is set to 0x00 after writing the registers to EEPROM using <bit0>. After power up, this field is updated with the CRC status of the EEPROM read.

Read Registers from EEPROM (self-clearing): Setting this bit = 1 will load all the registers from EEPROM, check the CRC and update the "CRC Status" field (0x9a = pass or 0xf1 = fail). This bit is automatically cleared when the action is complete.

On boot, if the EEPROM CRC check passes, LED D5/D6 will flash twice rapidly before the D5 heartbeat LED starts flashing every 2 seconds

On boot, if the EEPROM CRC check fails, the red LED D5 will flash on/off 10 times and the Modbus registers are initialized with default values. The EEPROM will not be written with default values. <bit0>, “Write Registers to EEPROM” has to be used to update the EEPROM.

Write Registers to EEPROM (self-clearing): Setting this bit = 1 will save all registers and a CRC word to EEPROM. This bit is automatically cleared when the action is complete. The “CRC Status” field will be set to 0x00. The “Enable Serial Number Update” bit is also cleared.

7.21. Register 0x1b: Product Information (Read Only)

Bit 15

Bit 0

MD3 Current Type <bit15>	b0111101 (ProductID) <bit14:bit8>	Version Number <bit7:bit0>
--------------------------	-----------------------------------	----------------------------

The “MD3 Current Type” field = 1 for standard current (7A max) and = 0 for low current (1.8A max) versions. The ProductID field is always b0111101 (0x3d). The lower byte is an integer firmware Version Number. Writing to this register will return a Modbus Write Error Message with exception code = 0x02 (invalid register address).

7.22. Register 0x1c: SerialPortConfiguration

Bit 15

Bit 0

Unused <bit15:bit6>	Parity <bit5:bit4>	Stop Bit <bit3>	Baud Rate <bit2:bit0>
---------------------	--------------------	-----------------	-----------------------

This register is used to configure the RS485 serial port. The data width is always 8 bits and is transmitted LSB first. The default for Modbus is 9600 bps, 1 start bit, 8 data bits, even parity, 1 stop bit. If a Modbus device has no parity, the Modbus specification states that 2 stop bits should be used. These default settings can be changed to the serial settings required by the Modbus host.

Parity	Description
00	No parity
01	Odd parity
10	Even parity

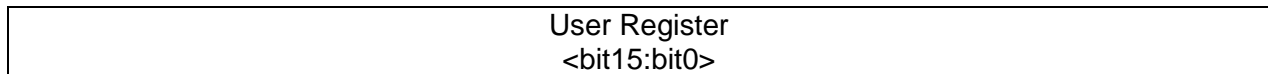
Stop Bit	Description
0	1 stop bit
1	2 stop bits

Baud Rate	Bits Per Second
0	4800
1	9600
2	19200
3	38400
4	57600
5	115200
6	230400
7	460800

7.23. Register 0x1d: UserRegister

Bit 15

Bit 0

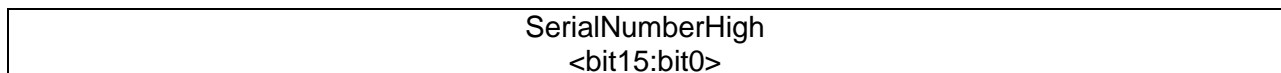


This is a general purpose register for user data. This register is saved to EEPROM.

7.24. Register 0x1e/0x1f: SerialNumberHigh / SerialNumberLow (unsigned, 32-bits, Read Only)

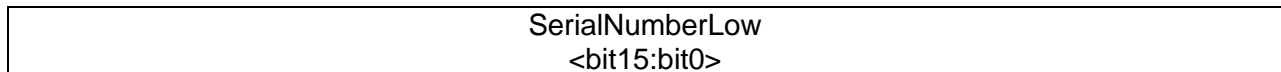
Bit 15

Bit 0



Bit 15

Bit 0



This pair of registers contains the 32-bit serial number of the MD3 unit. Writing to this register will return a Modbus Write Error Message with exception code = 0x02 (invalid register address).