MODBUS/TCP
Fieldbus manual
MDrive Motion Control
Products

Modbus

Schneider Electric

MODBUS/TCP Fieldbus Manual		
Date	**Revision**	**Changes**
09/14/2010	R091410	Initial Release
10/27/2010	R102710	Added registers specif c to MDrivePlus^2 Motion Control
02/01/2011	R020111	Added instructions for upgrading the Ethernet controller f rmware
04/06/2011	R040611	Added support for MODBUS/TCP f rmware version 1.0.0.2 and new manufacturer function code support for allowing execution, pause, resume and end of an MCode/TCP program.
08/13/2018	R040611	Added California Proposition 65 warning

Important information

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

Hand this manual and all other pertinent product documentation over to all users of the product.

Carefully read and observe all safety instructions and the chapter “Before you begin - safety information”.

This page intentionally left blank

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 About this manual

This manual is for use with the MODBUS/TCP Ethernet based MDrive Motion Control products.

This manual was developed from the perspective that you already have an understanding of the MODBUS protocol.

For detailed technical information on the MODBUS/TCP specif cation, please see http://www.modbus-ida.org.

1.2 Supported devices & protocols

1.2.1 Supported devices

MDrive 23 Plus² Motion Control

The MDrive 23 Plus Motion Control is supported absent the following:

- I/O points 9 - 12

All other features and functions are the same as the RS-422/485 versions.

MDrive 23 HybridMotion Control

The MDrive 23 Hybrid Motion Control is supported absent the following:

- I/O points 9 - 12

All other features and functions are the same as the RS-422/485 versions.

Protocols

The new MDrive Motion Control Ethernet products support two protocols in a single package:

1) **MCode/TCP** — Schneider Electric Motion USA's proprietary programming language for MDrive Motion Control products, adapted to utilize TCP/IP message formatting.

   If using the device using MCode/TCP, please see the MCode Programming and Reference Manual located on the web site at http://www.imshome.com/downloads/manuals.html.

2) **MODBUS/TCP** — A standard open industrial protocol supported by a variety of machine components such as programmable controllers, drives and controls, I/O modules and switches.

These protocols may be used separately or interchangeably, as is required by the constraints of the application by connecting to the port that the protocol is running on, 503 for MCode/TCP and 502 for MODBUS/TCP.

First conf guration connection will need to be over MCode/TCP using the TCP/IP Conf guration Utility to change the IP address of the device. Information on MCode is found in the MCode Programming and Software Reference available on the website at http://www.schneider-electric-motion.us.

Revision R040611

1.3 Documentation reference

The following user's manuals are available for the MODBUS devices:

- Product hardware manual, describes the technical data and installation of the product.
- Product software manual, describes the conf guration and programming of the product.
- Quick Reference, describes the basic wiring, connection and use of this product. The quick reference is shipped in printed form with the product.

This documentation is also available for download from our web site at: http://www.schneider-electric-motion.us.

1.4 Product software

1.4.1 TCP/IP Configuration Utility

TCP/IP Conf guration Utility is a software tool for setting the IP, upgrading f rmware and sending commands to the MODBUS device.

This software is required for the initial setup of the device.

Installation and usages instructions are to be found in Section 6 of this document.

This software may be downloaded from the web site at: http://www.schneider-electric-motion.us.

2 Safety

2.1 Qualification of personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product.
In addition, these persons must have received safety training to recognize and avoid hazards involved. These persons must have suff cient technical training, knowledge and experience and be able to foresee and detect potential hazards that may be caused by using the product, by changing the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used.

All persons working on and with the product must be fully familiar with all applicable standards, directives, and accident prevention regulations when performing such work.

2.2 Intended Use

The functions described in this manual are only intended for use with the basic product; you must read and understand the appropriate product manual.

The product may only be used in compliance with all applicable safety regulations and directives, the specif ed requirements and the technical data.

Prior to using the product, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety measures must be implemented.

Since the product is used as a component in an entire system, you must ensure the safety of persons by means of the design of this entire system (for example, machine design).

Operate the product only with the specif ed cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted is prohibited and can result in hazards.

Electrical equipment should be installed, operated, serviced, and maintained only by qualif ed personnel.

The product must NEVER be operated in explosive atmospheres (hazardous locations, Ex areas).and spare parts.

2.3 Hazard Categories

Safety instructions to the user are highlighted by safety alert symbols in the manual. In addition, labels with symbols and/or instructions are attached to the product that alert you to potential hazards.

Depending on the seriousness of the hazard, the safety instructions are divided into 4 hazard categories.

⚠ DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, will result in death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

CAUTION

CAUTION used without the safety alert symbol, is used to address practices not related to personal injury (e.g. **can result** in equipment damage).

2.4 Basic information

⚠ DANGER

UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

When the system is started, the drives are usually out of the operator's view and cannot be visually monitored.

• Only start the system if there are no persons in the hazardous area.

Failure to follow these instructions will result in death or serious injury.

⚠ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop, overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical functions.
- System control paths may include communication links. Consideration must be given to the implication of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines. 1)
- Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death or serious injury.

1) For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems".

CAUTION

HOT PLUGGING!

Do not connect or disconnect power, logic, or communications while the device is in a powered state.

Remove DC power by powering down at the AC side of the DC power supply.

Failure to follow these instructions can result in equipment damage.

California Proposition 65 Warning—Lead and Lead Compounds
Advertencia de la Proposición 65 de California—Plomo y compuestos de plomo
Avertissement concernant la Proposition 65 de Californie—Plomb et composés de plomb

⚠ **WARNING:** This product can expose you to chemicals including lead and lead compounds, which are known to the State of California to cause cancer and birth defects or other reproductive harm. For more information go to: www.P65Warnings.ca.gov.

⚠ **ADVERTENCIA**: Este producto puede exponerle a químicos incluyendo plomo y compuestos de plomo, que es (son) conocido(s) por el Estado de California como causante(s) de cáncer y defectos de nacimiento u otros daños reproductivos. Para mayor información, visite : www.P65Warnings.ca.gov.

⚠ **AVERTISSEMENT**: Ce produit peut vous exposer à des agents chimiques, y compris plomb et composés de plomb, identifiés par l'État de Californie comme pouvant causer le cancer et des malformations congénitales ou autres troubles de l'appareil reproducteur. Pour de plus amples informations, prière de consulter: www.P65Warnings.ca.gov.

Schneider Electric USA, Inc.
800 Federal Street
Andover, MA 01810 USA
888-778-2733
www.schneider-electric.us

Importado en México por:
Schneider Electric México, S.A. de C.V.
Av. Ejercito Nacional No. 904
Col. Palmas, Polanco 11560 México, D.F.
55-5804-5000
www.schneider-electric.com.mx

Schneider Electric Canada, Inc.
5985 McLaughlin Road
Mississauga, ON L5R 1B8 Canada
800-565-6699
www.schneider-electric.ca

3 MODBUS Implementation

3.1 MODBUS overview

MODBUS is a communications interface developed in 1979 by PLC manufacturer Modicon, Inc. (now a brand of Schneider Electric). MODBUS is designed for multidrop networks based on a master-client architecture.

The availability of devices using MODBUS has made it a de facto standard for industrial communications network. MODBUS was originally developed for use with serial communications interfaces such as RS-232 and RS-485, MODBUS/TCP communications over TCP/IP has become a standard because of the ease of interface and simpler message format.

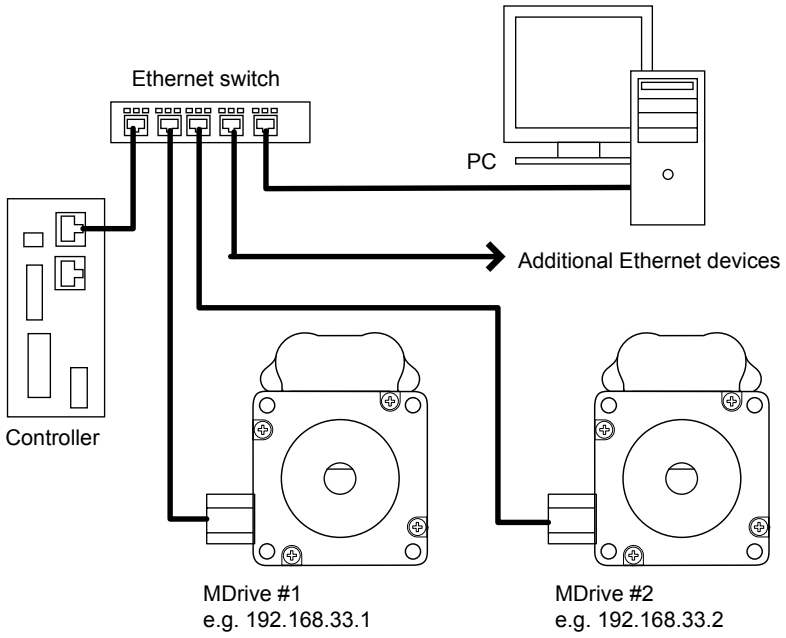


Figure 3.1: Example MODBUS network with MDrive products.

MODBUS/TCP is basically the MODBUS serial RTU encapsulated in a TCP/IP wrapper and is used for TCP/IP communications between client and server devices on an Ethernet TCP/IP network.

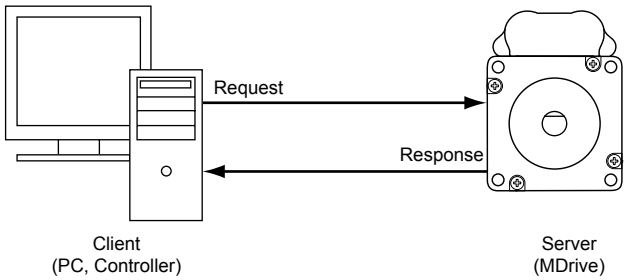


Figure 3.2: Client-server model

3.2 Message format

MODBUS/TCP uses the OSI (Open Systems Interconnection) networking model. The MODBUS ADU (Application Data Unit) makes up the OSI application layer and is wrapped inside the data array of the TCP/IP Ethernet data packet. Figure 3.3 below shows the constructiopn of a TCP/IP Ethernet data packet used tor the MODBUS/TCP protocol.

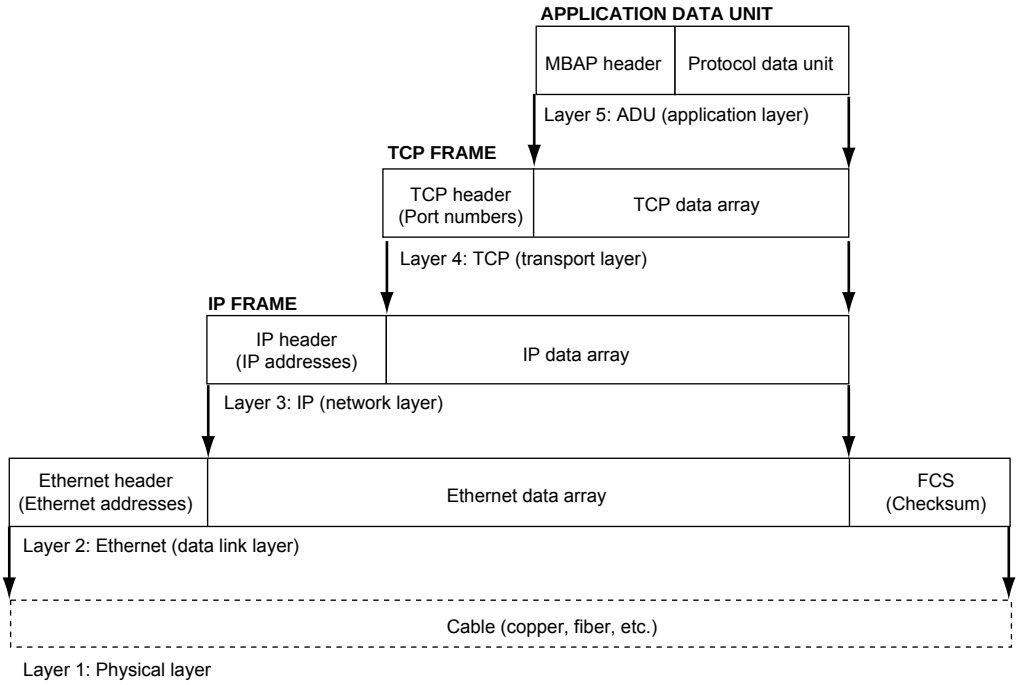


Figure 3.3: Construction of an ethernet data packet for MODBUS/TCP

3.2.1 `ADU (application data unit)

A MODBUS/TCP data packet, or Application Data Unit (ADU) consists of two components:

1) MODBUS Application Protocol (MBAP) header
2) Protocol Data Unit (PDU)

The information contained in the ADU is embedded in the data portion of the TCP frame.

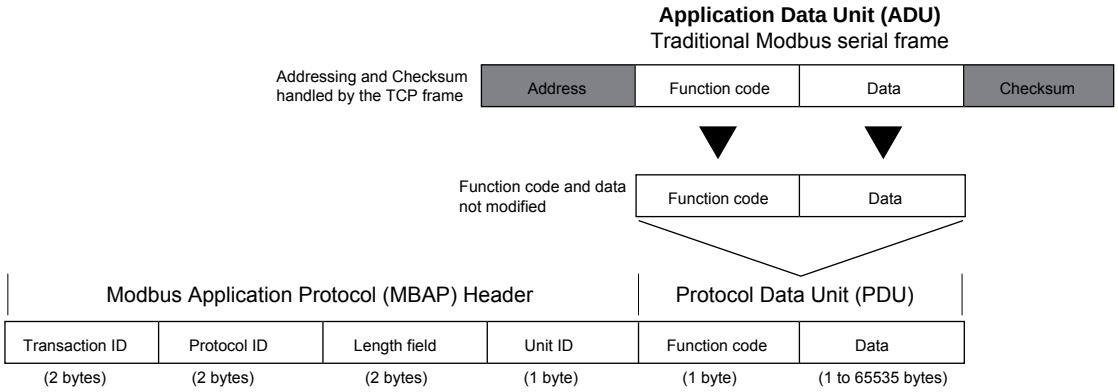


Figure 3.4: MODBUS/TCP data packet construction

MBAP header The MBAP header is 7 bytes long and consists of the following f elds made up of four f elds”

Fields	Length	Description
Transaction identifier	2 bytes	ID of a MODBUS request/response transaction. This field is used for transaction pairing, the server will copy in the Transaction ID of the request into the response.
Protocol identifier	2 bytes	0 = MODBUS protocol
Length	2 bytes	Number of bytes following, including the Unit ID and the byte length of the PDU.
Unit identifier	1 byte	ID of a remote slave. Used for intra-system communications with other buses i.e. between MODBUS/TCP and a MODBUS serial line slave through a gateway.

Table 3.1 MBAP header

Protocol Data Unit (PDU)

The PDU consists of 2 parts:

1) **Function code:** the function code identif es the action to be taken using the data bytes that will follow. These functions are covered in detail in Section 4 of this document. Basic functions are:
   Reading inputs, writing coils (digital outputs), read/write registers and manufacturer specif c conf guration functions.
2) **Data:** The data contained in the PDU, it will consist of the data and/or parameters associated with the commands to operate your MDrive product.

4 Function codes

The MDrive supports the following function codes:

Function code dec	hex	Description
Device ID		
43/14	0x2B/0x0E	Read device identification
Public		
02	0x02	Read digital inputs
01	0x01	Read coils (digital outputs)
05	0x05	Write single coil (digital output)
03	0x03	Read holding register
16	0x10	Write multiple registers
Manufacturer specific		
65	0x41	Read I/O and trip configuration and parameters
66	0x42	Write I/O and trip configuration and parameters, execute, pause, resume or end stored MCode/TCP programs.

Table 4.1: Supported function codes

Exception codes

Each function has 4 error, or exception codes that will return in case of an error with the transaction. They are:

01 – Illegal or not supported function
02 – Illegal data address
03 – Illegal data value
04 – Slave device failure

4.1 Device ID

4.1.1 Read device identification – 43 / 14 (0x2B / 0x0E)

The device type contains information about your MDrive product, importantly the part number, serial number, and f rmware version installed.

Request

	Length	Value
Function code	1 byte	0x2B
MEI* type	1 byte	0x0E
Read device ID code	1 byte	01 / 02 / 03 / 04
Object ID	1 byte	0x00 – 0x06

*MODBUS Encapsulated Interface

Response

	Length	Value
Function code	1 byte	0x2B
MEI type	1 byte	0x0E
Read device ID code	1 byte	
Conformity level	1 byte	
More follows	1 byte	
NextObjectId	1 byte	0x00
Number Of Obects	1 byte	0x06
Object Id	1 byte	0x00
Object Length	1 byte	0x03
Object Value	3 bytes	“SEM USA”
Object Id	1 byte	0x01
Object Length	1 byte	0xXX
Object Value	X bytes	“MDrive P/N”
Object Id	1 byte	0x02
Object Length	1 byte	0x08
Object Value	8 bytes	“4.0.0.0”
Object Id	1 byte	0x03
Object Length	1 byte	0x20
Object Value	32 bytes	“www.imshome.com”
Object Id	1 byte	0x04
Object Length	1 byte	0x0A
Object Value	10 bytes	MDrive Ethernet
Object Id	1 byte	0x05
Object Length	1 byte	0xXX
Object Value	X bytes	“Serial number”
Object Id	1 byte	0x06
Object Length	1 byte	0x0C
Object Value	12 bytes	ASI 4.x.x.x

4.2 Public function codes

4.2.1 Read digital inputs 02 (0x02)

Function 02 is used to read the state of the digital inputs 1 - 4 on your MDrive product. The request PDU contains the starting address of the first input specified, and the number of inputs.

In the response message the input states are packaged as 1 input per bit of the data field where status is indicated as 1 = ON and 0 = OFF. The LSB of the data byte will be the address of the input in the request.

NOTE: Digital inputs on the MDrive may also be read using the holding registers.

NOTE 2: The inputs must be configured as such using the manufacturer specific function code 66 (0x42).

Request

	Length	Value
Function code	1 byte	0x02
Starting Address	2 bytes	0x002D – 0x0030
Quantity of inputs	1 byte	1 to 4

Response

	Length	Value
Function code	1 byte	0x02
Byte Count	1 byte	1 to 4
Input status	1 to 4 bytes	

Error

	Length	Value
Error code	1 byte	0x82
Exception	1 byte	01, 02, 03 or 04
Input status	1 to 4 bytes	

Example

Example shows a read of all 4 MDrive digital inputs, the response shows input states: I1=1, I2=1, I3=0, I4=1. Input 1 is the input address and is therefore the LSB,

Request		Response	
Function	0x02	Function	0x02
Starting address Hi	0x00	Byte count	0x01
Starting address Lo	0x2D	Input status 4 – 1	0x0B
Qty of inputs Hi	0x00		
Qty of inputs Lo	0x04		

4.2.2 Read coils (digital outputs) – 01 (0x01)

Function 01 is used to read the state of the digital outputs 1 - 4 on your MDrive product. The request PDU contains the starting address of the f rst output specif ed, and the number of outputs.

In the response message the output states are packaged as 1 output per bit of the data f eld where status is indicated as 1 = ON and 0 = OFF. The LSB of the data byte will be the output of the address in the request.

NOTE: Digital outputs on the MDrive may also be read using the holding registers.

NOTE 2: The outputs must be conf gured as such using the manufacturer specif c function code 66 (0x42).

Request

	Length	Value
Function code	1 byte	0x01
Starting Address	2 bytes	0x004B – 0x004E
Quantity of outputs	1 byte	1 to 4

Response

	Length	Value
Function code	1 byte	0x01
Byte Count	1 byte	1 to 4
Input status	1 to 4 bytes	

Error

	Length	Value
Error code	1 byte	0x81
Exception	1 byte	01, 02, 03 or 04
Input status	1 to 4 bytes	

Example

Example shows a read of all 4 MDrive digital outputs, the response shows outputs states:O1=1, O2=0, O3=1, O4=0. Output 1 is the output address and is therefore the LSB,

Request		Response	
Function	0x01	Function	0x02
Starting address Hi	0x00	Byte count	0x01
Starting address Lo	0x4B	Output status 4 – 1	0x05
Qty of outputs Hi	0x00		
Qty of outputs Lo	0x04		

4.2.3 Write single coil (digital output) – 05 (0x05)

This function is used to turn a single output point ON or OFF.

The state is specif ed by a constant in the request data f eld:

- 0xFF00 – turns the output ON
- 0x0000 – turns the output OFF

All other values are illegal and will return an exception code 03: Illegal data value.

NOTE: Digital outputs on the MDrive may also be written using the holding registers.

NOTE 2: The outputs must be conf gured as such using the manufacturer specif c function code 66 (0x42).

Request

	Length	Value
Function code	1 byte	0x05
Output address	2 bytes	0x004B – 0x004E
Output value	2 bytes	0x0000 or 0xFF00

Response

	Length	Value
Function code	1 byte	0x05
Output address	2 bytes	0x004B – 0x004E
Output value	2 bytes	0x0000 or 0xFF00

Error

	Length	Value
Error code	1 byte	0x85
Exception	1 byte	01, 02, 03 or 04

Example

Example shows setting output 3 to an ON state.

Request		Response	
Function	0x05	Function	0x05
Output address Hi	0x00	Output address Hi	0x00
Output address Lo	0x4D	Output address Lo	0x4D
Output value Hi	0xFF	Output value Hi	0xFF
Output value Lo	0x00	Output value Lo	0x00

4.2.4 Read holding registers – 03 (0x03)

Tis function code is used to read a contingous block of holding registers in your MDrive. The request PDU specif es the starting register address and the number of registers.

MDrive command data mapped to the holding registers measure 1, 2 or 4 bytes in length, therefore you will not need to read more than two consecutive registers per request.

NOTE: A number of registers are marked as reserved. Use of any of these registers will return a exception code 02: illegal data address.

See Section 5 of this document for the register map.

Request

	Length	Value
Function code	1 byte	0x03
Starting address	2 bytes	0x0000 – 0x00A7*
Qty of addresses	2 bytes	1 to 4

*A number of addresses in this block are reserved for future use and will return an error.

Response

	Length	Value
Function code	1 byte	0x03
Byte count	2 bytes	1 – 4
Register value	2 bytes	0x00 to 0x04

Error

	Length	Value
Error code	1 byte	0x83
Exception	1 byte	01, 02, 03 or 04

Example

Example shows reading registers 0x008A and 0x008B (maximum velocity). The value reads as 0x00 0B B8 00 or decimal 768000 steps/second.

Request		Response	
Function	0x0F	Function	0x0F
Starting address Hi	0x00	Byte count	0x04
Starting address Lo	0x8A	Register value Hi	0x00
Qty of registers Hi	0x00	Register value Lo	0x0B
Qty of registers Lo	0x02	Register value Hi	0xB8
		Register value Lo	0x00

4.2.5 Write multiple registers – 16 (0x10)

This function code is used to write a block of registers in your MDrive. The request PDU specifies the starting register address and the number of registers to be written.

MDrive command data mapped to the registers measure 1, 2 or 4 bytes in length, therefore you will not need to write more than two consecutive registers per request.

NOTE: A number of registers are marked as reserved. Use of any of these registers will return a exception code 02: illegal data address.

See Section 5 of this document for the register map.

Request

	Length	Value
Function code	1 byte	0x10
Starting address	2 bytes	0x0000 – 0x00A7*
Qty of addresses	2 bytes	1 to 4
Byte count	1 byte	2 to 4
Registers value	to 4 bytes	value

*A number of addresses in this block are reserved for future use and will return an error.

Response

	Length	Value
Function code	1 byte	0x10
Byte count	2 bytes	1 – 4
Register value	2 bytes	0x00 to 0x04

Error

	Length	Value
Error code	1 byte	0x90
Exception	1 byte	01, 02, 03 or 04

Example

Example shows writing registers 0x008A and 0x008B (maximum velocity). The value will be set as decimal 600000 steps/second, or 0x00 09 27 C0.

Request		Response	
Function	0x10	Function	0x10
Starting address Hi	0x00	Starting address Hi	0x00
Starting address Lo	0x8A	Starting address Lo	0x8A
Qty of registers Hi	0x00	Qty of registers Hi	0x00
Qty of registers Lo	0x02	Qty of register s Lo	0x02
Byte count	0x04	Byte count	0x04
Registers value Hi	0x00		
Registers value Lo	0x09		
Registers value Hi	0x27		
Registers value Lo	0xC0		

4.3 Manufacturer specific function codes

The device supports two manufacturer specif c function codes:

- 65 (0x41) – Read specif c functions
- 66 (0x42) – Write specif c functions

Manufaturer functions

Function	R/W	Function	R/W
Setup I/O points 1 – 4	R/W	Make Up mode (MU)	R/W
Setup I/O points 7 – 8 (clock I/O)	R/W	End program (E)	R
Clock Ratio	R/W	Execute program (EX)	R
Setup Analog input (S5)	R/W	Pause program (RS)	R
Setup Capture/Trip I/O (S13)	R/W	Resume program (RS)	R
Trip on Relative Position (TR)	R/W		

4.3.1 Manufacturer specific commands using 65 (0x41) and 66 (0x42)

General purpose I/O setup

The ASCII of the command mnemonic is sent in revers order in the request PDU i.e. 1S, 2S etc. The parameter string is written or read in normal sequence.

Function		65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc											
MCode mnemonic		S1 (I/O 1)			S2 (I/O 2)			S3 (I/O 3)			S4 (I/O 4)		
Mnemonic	Hi word	0x20	0x20		0x20	0x20		0x20	0x20		0x20	0x20	
	Lo word	0x31	0x53		0x32	0x53		0x33	0x53		0x34	0x53	
Parameter data length		6 – 7 bytes											
Parameter string example	Params		A			B			C		See parameters table below for details		
	ASCII	=	0	,	0	,	0						
	Hex	3D	30	2C	30	2C	30						

Parameters

A – I/O type			B – Active Hi/Lo			C – Sinking or Sourcing		
Input functions								
dec	hex	function	dec	hex		dec	hex	
0	30	General purpose	0	30	Active Lo	0	30	Sinking
1	31	Homing	1	31	Active Hi	1	31	Sourcing
2	32	Limit +						
3	33	Limit –						
5	35	Soft stop						
7	37	Jog +						
8	38	Jog –						
11	31 31	Reset						
Output Functions								
16	31 36	General purpose						
17	31 37	Moving						
18	31 38	Fault						
20	32 30	Velocity changing						
21	32 31	Locked rotor						
23	32 33	Moving to position						
24	32 34	Hybrid circuitry active						
25	32 35	Make-up active						

Clock I/O setup (I/O 7 & 8)

The ASCII of the command mnemonic is sent in revers order in the request PDU i.e. 1S, 2S etc. The parameter string is written or read in normal sequence.

Function		**65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc**					
MCode mnemonic		**S7 (I/O 7)**				**S8 (I/O 8)**	
Mnemonic	Hi word	0x20 0x20				0x20 0x20	
	Lo word	0x37 0x53				0x38 0x53	
Parameter data length		4 – 5 bytes					
Parameter string example	Params		**A**		**B**	See parameters table below for details	
	ASCII	=	34	,	1		
	Hex	3D	33 34	2C	31		

Parameters

A – I/O type			**B – Active Hi/Lo**		
Input functions					
dec	hex	function	dec	hex	
33	33 33	Step/direction	0	30	Active Lo
34	33 34	Quadrature	1	31	Active Hi
35	33 35	Up/down			
Output Functions					
49	34 39	Step/direction			
50	35 30	Quadrature			
51	35 31	Up/down			

Set clock ratio

The ASCII of the command mnemonic is sent in revers order in the request PDU i.e. RC. The parameter string is written or read in normal sequence.

Clock Ratio value for electronic gearing. The value selected will set the ratio from the clock input on I/O 7 (Step Clock) and I/O 8

Function		**65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc**				
MCode mnemonic		**CR (Clock Ratio)**				
Mnemonic	Hi word	0x20 0x20				
	Lo word	0x52 0x43				
Parameter data length		Up to 6 bytes				
	ASCII	=	1	.	5	0	0
	Hex	3D	31	2E	35	30	30

Analog input setup

The ASCII of the command mnemonic is sent in revers order in the request PDU i.e. 5S. The parameter string is written or read in normal sequence.

The analog input may be configured for voltage or current mode with ranges of 0 to 5 VDC, 0 to 10 VDC, o to 20 mA or 4 to 20 mA.

Function		**65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc**					
MCode mnemonic		**S5 (Analog Input)**					
Mnemonic	Hi word	0x20 0x20					
	Lo word	0x35 0x53					
Parameter data length		4 – 5 bytes					
Parameter string example	Params		**A**		**B**		See parameters table below for details
	ASCII	=	0	,	0		
	Hex	3D	30	2C	30		

Parameters

A – Input mode			**B – Input range**		
dec	hex	mode	dec	hex	range
9	39	Voltage	0	30	0 – 5 VDC
			1	31	0 – 10 VDC
10	31 30	Current	0	30	0 – 20 mA
			1	31	4 – 20 mA

Capture/trip I/O setup

The ASCII of the command mnemonic is sent in revers order in the request PDU i.e. 31S. The parameter string is written or read in normal sequence.

Note the the only trip function available is trip on relative position.

Function		**65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc**					
MCode mnemonic		**S13 (Capture output/trip input)**					
Mnemonic	Hi word	0x20 0x33					
	Lo word	0x31 0x53					
Parameter data length		4 – 5 bytes					
Parameter string example	Params		**A**		**B**		See parameters table below for details
	ASCII	=	60	,	0		
	Hex	3D	30	2C	30		

Parameters

A – Input mode			**B – Active state**		
dec	hex	mode	dec	hex	range
60	36 30	Capture input	0	30	Active Lo
61	36 31	Trip output	1	31	Active Hi

Trip on relative position

The ASCII of the command mnemonic is sent in revers order in the request PDU i.e. RT. The parameter string is written or read in normal sequence.

Note the the only trip function available is trip on relative position. to re-enable the trip, use register 0x007D.

Function		**65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc**					
MCode mnemonic		**TR (Trip on relative position)**					
Mnemonic	Hi word	0x20 0x20					
	Lo word	0x52 0x54					
Parameter data length		Up to 11 bytes					
	ASCII	=	6	5	0	0	0
	Hex	3D	36	35	30	30	30

End

Stops the execution of a program.

Function		**65 (0x41) Read Mfg Spc**
MCode mnemonic		**E (Stop program execution)**
Mnemonic	Hi word	0x20 0x20
	Lo word	0x20 0x45
Parameter data length		0 bytes

Execute program

Executes the address or label of a stored MCode/TCP program.

NOTE: The program resident CANNOT have any print statements in the code.

User variables that need to be read over MODBUS/TCP during program execution must be read using Registers R1 – R4 and V1 – V8 using the associated register (See Section 5: Register Map). If using V1 – V8, they must be declared within the program using the VA (Create user variable) MCode command.

The example below shows the programmed labeled A3 being executed.

Function		**65 (0x41) Read Mfg Spc**	
MCode mnemonic		**EX (Execute program)**	
Mnemonic	Hi word	0x20 0x20	
	Lo word	0x52 0x54	
Parameter data length		Up to 4 bytes	
	ASCII	A	3
	Hex	41	33

Pause program

Pauses a running MCode/TCP program.

Function		**65 (0x41) Read Mfg Spc**
MCode mnemonic		**PS (Pause program)**
Mnemonic	Hi word	0x20 0x20
	Lo word	0x53 0x50
Parameter data length		0 bytes

Resume program

Resumes a paused MCode/TCP program.

Function		**65 (0x41) Read Mfg Spc**
MCode mnemonic		**RS (Resume program)**
Mnemonic	Hi word	0x20 0x20
	Lo word	0x53 0x52
Parameter data length		0 bytes

Make up mode (Hybrid only)

Sets the mode for Hybrid make up steps

Function		**65 (0x41) Read Mfg Spc, 66 (0x42) Write Mfg Spc**				
MCode mnemonic		**MU (Make up mode)**				
Mnemonic	Hi word	0x20 0x20				
	Lo word	0x55 0x4D				
Parameter data length		4 bytes				
Parameter string example	Params		**A**		**B**	See parameters table below for details
	ASCII	=	2	,	0	
	Hex	3D	32	2C	30	

Parameters

A – Make up mode			**B – Parameter**		
dec	hex	mode	dec	hex	range
0	30	Off	0	30	Use lead/lag
1	31	Use make up frequency	1	31	Clear lead/lag
2	32	Use system speed			

4.3.2 Read manufacturer specific – 65 (0x41)

Request

	Length	Value
Function code	1 byte	0x41
Mnemonic Hi word	2 bytes	0x2020 0x2020/2033*
Mnemonic Lo word	2 bytes	See section 4.3.1 for listing

*For capture/trip I/O point.

Response

	Length	Value
Function code	1 byte	0x41
Byte count	2 bytes	N* (quantity of characters returned)
Response	n bytes	n=N or N+1

Error

	Length	Value
Error code	1 byte	0xC1
Exception	1 byte	01, 02, 03 or 04

Example Example shows reading the setting of the trip on relative input (TR).

Request		Response	
Function	0x41	Function	0x41
Mnemonic Hi word	0x20	Byte count	0x07
	0x20	Response	0x30
Mnemonic Lo word	0x52		0x2C
	0x54		0x20
			0x30
			0x2C
			0x20
			0x30

4.3.3 Write manufacturer specific – 66 (0x42)

Request

	Length	Value
Function code	1 byte	0x42
Mnemonic Hi word	2 bytes	0x2020 0x2020/2033*
Mnemonic Lo word	2 bytes	See section 4.3.1 for listing
Byte count	1 byte	1-n bytes (28 max)
Parameter data string	n bytes	See section 4.3.1 for listing

*For capture/trip I/O point.

Response

	Length	Value
Function code	1 byte	0x42
Byte count written	1 bytes	N* (quantity of characters returned)
Mnemonic Hi word	2 bytes	0x2020 0x2020/2033*
Mnemonic Lo word	2 bytes	See section 4.3.1 for listing

Error

	Length	Value
Error code	1 byte	0xC2
Exception	1 byte	01, 02, 03 or 04

Example Example shows setting input 1 (S1). The input is shown set to a general purpose sinking input which is active when Hi, or S1=0,1,0. Note that the data string includes all of the characters, including the equal sign and the commas.

Request		Response	
Function	0x41	Function	0x41
Mnemonic Hi word	0x20	Bytes written	0x04
	0x20	Mnemonic Hi word	0x20
Mnemonic Lo word	0x31		0x20
	0x53	Mnemonic Lo word	0x31
Byte count	0x06		0x53
Parameter data string	0x3D		
	0x30		
	0x2C		
	0x31		
	0x2C		
	0x30		

5 Register map

Function	Address	Bytes	Description	Range	Default	MCode
Acceleration	0x0000 - 0x0001	4	Sets the acceleration rate in steps per second^2.	91 to 1525878997	1000000	A
Reserved	0x0002-4	—	Reserved	—	—	—
Counter 1	0x0005 - 0x0006	4	Variable contains the count of clock pulses generated by the device.	-2147483648 to +2147483647	0	C1
Counter 2	0x0007 - 0x0008	4	Variable contains the count of encoder counts read by the device.	-2147483648 to +2147483647	0	C2
Software reset enable	0x0009	1	Flag configures the device to respond (1) or not respond (0) to a CTRL+C software reset.	0/1	1	CE
Reserved	0x000A	—	Reserved	—	—	—
Clock mode enable [1]	0x000B	1	Flag to enable (1) or disable (0) clock mode (MDrivePlus only)	0/1	0	CM
Reserved	0x000C-E	—	Reserved	—	—	—
Input 1 debounce	0x000F	1	Sets digital filtering in milliseconds. Input must be stable for the set time before state change is detected.	0 – 255	0	D1
Input 2 debounce	0x0010					D2
Input 3 debounce	0x0011					D3
Input 4 debounce	0x0012					D4
Analog input filter	0x0013	1	Filter does continuos average by computing: (((X-1)/X)*current reading) + (1 / X) If X = 10, then: ((current averaged value * 9)/10) + (new reading / 10) == NEW current averaged value.	0 – 255	0	D5
Reserved	0x0014-17	—	Reserved	—	—	—
Deceleration	0x0018 - 0x0019	4	Sets deceleration rate in steps per second^2.	91 – 1525878997	1000000	D
Deadband [1]	0x001A	2	Encoder deadband (MDrivePlus only)	0 to ±65000	10	DB
Decrement variable	0x001B	1	instruction will decrement the specified variable by one.	—	—	DC
Drive enable	0x001C	1	Flag enables (1) or disables (0) the drive portion of the device.	0/1	1	DE
Reserved	0x001D	—	Reserved	—	—	—
Encode enable	0x001E	1	Enable encoder functions	0/1	0	EE
Error flag	0x001F	1	Flag indicates whether an error condition exists (1) or not (0).	0/1	0	EF
Reserved	0x0020	—	Reserved	—	—	—
Error	0x0021	2	Variable holds the error code of the last error. must be read or set to 0 to clear.	—	0	ER
Reserved	0x0022-28	—	Reserved	—	—	—
Holding current	0x0029	1	Sets the motor holding current in percent (%)	0 to 100	5	HC
Reserved	0x002A	—	Reserved	—	—	—

1 = MDrivePlus Motion Control version only

Function	Address	Bytes	Description	Range	Default	MCode
Homing mode	0x002B	1	Sets the behavior of the axis for homing routines: Mode=1 - Slew – at VM, creep + at VI Mode=2 - Slew – at VM, creep – at VI Mode=3 - Slew + at VM, creep – at VI. Mode=4 - Slew + at VM, creep + at VI	1 – 4		HM
Hold current delay time	0x002C	2	Set the time in milliseconds between the cessation of motion and shift to holding current percent. Total time is represented by the sum of 0x002C+0x0049 (motor settling delay time. The sum cannot be more than 65535 msec.	0 (no delay) or 2 – 65535	500	HT
Read input 1	0x002D	1	Read the logic state of the specified input.	0/1	—	I1
Read input 2	0x002E					I2
Read input 3	0x002F					I3
Read input 4	0x0030					I4
Read analog input	0x0031		Read the value of the analog input in counts.	0 – 1023	—	I5
Read index mark [1]	0.0032	1	This variable will read the on/off state of the Encoder Index Mark (MDrivePlus only)	0/1	—	I6
Reserved	0x0033-36	—	Reserved	—	—	—
Increment variable	0x0037	1	Increments the specified variable by one.	—	—	IC
Reserved	0x0038-39	—	Reserved	—	—	—
Read inputs 1 - 4 as BCD	0x003A	1	Reads the logic states of inputs 1-4 and returns them as a decimal value. Input 1 will represent the LSb..	0 – 15	—	IL
Read all inputs as BCD	0x003B	1	Reads the logic states of inputs 1-4 and returns them as a decimal value. Input 1 will represent the LSb..	0 – 15	—	IN
Reserved	0x003C-3E	—	Reserved	—	—	—
Jog enable	0x003F	1	Enables (1)/disables(0) jog functions when inputs are configured as Jog+ and Jog –.	0/1	0	JE
Reserved	0x0040-41	—	Reserved	—	—	—
Limit stop mode	0x0042	1	Sets the behavior of the axis upon reaching a limit switch. Mode=1 – Normal limit function with decel ramp. Mode=2 – Stops motion with decel ramp, no homing. Mode=3 – Stops motion with decel ramp, stops program. Mode=4 – Normal limit function, no decel. Mode=5 – Stops motion, no decel, no homing. Mode=6 – Stops motion, stops program, no decel.	1 – 6	1	LM
Move to absolute position	0x0043 - 0x0044	4	Point-to-point move to a ± absolute position.	—	—	MA
Moving to position	0x0045	1	Indicates that the axis is moving (1) to an absolute or relative position or stopped (0).	0/1	0	MP
Move to relative position	0x0046 - 0x0047	4	Point-to-point move to a ± position relative distance from current position.	—	—	MR

1 = MDrivePlus Motion Control version only

Function	Address	Bytes	Description	Range	Default	MCode
Microstep resolution	0x0048	1	Set the microstep resolution in microsteps per motor full step.	See table	256	MS
Motor settling delay time	0x0049	2	Specifies the motor settling delay time in milliseconds. This allows the motor to settle following a move. This variable is added to 0x002C to determine the total time before shifting to holding current. The sum cannot be more than 65535 msec.	0 – 65000	0	MT
Moving	0x004A	1	Indicates whether the axis is in motion (1) or stationary (0).	0/1	0	MV
Write output 1	0x004B	1	Write (set) the logic state of the specified output.	0/1	—	O1
Write output 2	0x004C					O2
Write output 3	0x004D					O3
Write output 4	0x004E					O4
Reserved	0x004F-54	—	Reserved	—	—	—
Set outputs 1 - 4 as a group	0x0055	1	Set outputs 1-4 as one 4 bit binary value. The value is entered in decimal, with a range of 0-15 in binary where output 1 will be the LSb	0 – 15	—	OL
Set outputs 1 - 4 as a group	0x0056	1	Set outputs 1-4 as one 4 bit binary value. The value is entered in decimal, with a range of 0-15 in binary where output 1 will be the LSb	0 – 15	—	OT
Position counter	0x0057 - 0x0058	4	Sets or reads the axis ± position in motor steps. The value of the register will be used as the reference point for absolute and relative moves.	-2147483648 – +2147483647	0	P
Position capture at trip	0x0059 - 0x005A	4	Captures axis position during a trip event. Activation will occur upon any trip function EXCEPT a position trip.	—	—	PC
Reserved	0x005B	—	Reserved	—	—	—
Position maintenance [1]	0x005C	1	Enables (1) or disables (0) position maintenance functions (MDrivePlus only)	0/1	0	PM
Reserved	0x005D-5E	—	Reserved	—	—	—
User register 1	0x005F - 0x0060	4	registers may contain up to 11 digits including the sign and may be used to store and retrieve data.	32 bit	—	R1
User register 2	0x0061 - 0x0062					R2
User register 3	0x0063 - 0x0064					R3
User register 4	0x0065 - 0x0066					R4
Run current	0x0067	1	Sets the motor run current in percent (%).	1 to 100	25	RC
Reserved	0x0068-75	—	Reserved	—	—	—

Available Microsteps Per Revolution

0x0048=	**1**	**2**	**4**	**5**	**8**	**10**	**16**	**25**	**32**	**50**
steps/rev	200	400	800	1000	1600	2000	3200	5000	6400	10000
0x0048=	**64**	**100**	**125**	**128**	**200**	**250**	**256**	**180**	**108**	**127**
steps/rev	12800	20000	25000	25600	40000	50000	51200	36000[1]	21600[2]	25400[3]

1=0.01 deg/µstep 2=1 arc minute/µstep 3=0.001 mm/µstep

1 = MDrivePlus Motion Control version only

Function	Address	Bytes	Description	Range	Default	MCode
Save	0x0076	1	Saves variables and flags in working memory to NVM.		—	S
Stall factor [1]	0x0077	2	Difference between commanded position and encoder counts at which a stall is indicated (MDrivePlus only).	0 to 65000	15	SF
Slew axis	0x0078 - 0x0079	4	Slews the axis at velocity in steps/second in the specified ± direction, Slew velocity is independent of 0x008B (maximum velocity).	±5000000	—	SL
Stall mode [1]	0x007A	1	Stall detection mode determines the response to a stall detect, either motion stops (0) or attempts to continue (1) (MDrivePlus only).	0/1	0	SM
Stall flag [1]	0x007B	1	indicates a motor stall (1) or not stalled (0) (MDrivePlus only).	0/1	0	ST
Reserved	0x007C	—	Reserved	—	—	—
Trip enable	0x007D	1	Enables/re-enables trip functions as specified by the table below. Multiple trips may be specified by adding the trip definitions i.e. 0x007D=10 will allow trip on position (2) and trip on time (8). Trips are set up using manufacturer function codes 65 (0x41) and 66 (0x42) **Trip enable definitions:** 0 – Trip functions disabled. 1 – Reserved 2 – Reserved 4 – Reserved 8 – Reserved 16 – Trip on relative position 32 – Reserved	0 – 43	0	TE
Reserved	0x007E-84	—	Reserved	—	—	—
Read axis velocity	0x0085 - 0x0086	4	Reads the current velocity in motor steps per second. NOTE: If hybrid circuitry is in make-up mode, 0x0085-86 will not return an accurate value. When the hybrid product is in torque control mode 0x0085-86 will return a zero (0). Read only variable.	—	—	V
Reserved	0x0087	—	Reserved	—	—	—
Velocity is changing	0x0088	1	Axis velocity is changing (1) or constant (0). Read only status flag.	0/1	0	VC
Set initial velocity	0x0089 - 0x008A	4	Set the initial velocity of the axis in motor steps per second.	1 to max. velocity – 1	1000	VI
Set maximum velocity	0x008B - 0x008C	4	Set the maximum velocity of the axis in motor steps per second.	Initial velocity +1 to 5000000	768000	VM
Reserved	0x008D	—	Reserved	—	—	—

1 = MDrivePlus Motion Control version only

5.1 Hybrid specific registers

Function	Address	Bytes	Description	Range	Default	MCode
Set hybrid mode	0x008E	1	Sets the hybrid operational behavior to one of four modes, detailed below: 0 Hybrid circuity disabled. 1 Fixed current mode. Current is set by the run and hold current commands, Speed is set by the system speed command. 2 Variable current mode. Current will vary as needed to position the load with the maximum current set by the run current command. 3 Torque mode, torque and speed will vary as needed to move/ position the load with the maximum torque % and speed as specified by the torque and torque-speed commands.	0 – 3	2	AS
Read hybrid status	0x008F	1	Read only status flag will return the conditions listed below, If multiple conditions exist the result is additive. i.e. At zero (64) and Calibration complete (128) AF=192 1 – Rotor lead limit reached. 2 – Rotor lag limit reached. 4 – Maximum lead/lag limit reached. 8 – Locked rotor. 16 – Hybrid mode is active. 32 – Hardware fault condition exists. 64 – At zero (0). 128 – Calibration s complete.	1 – 255	—	AF
Calibration mode	0x0090		Calibration is required to set the initial relationship between the rotor and stator and will occur automatically following these conditions: ■ Power on ■ Software reset ■ Hybrid functionality enabled ■ Bridge re0enabled after being disabled ■ Microstep resolution is changed. **Mode=0** – Timed calibration sets motor current to a percentage specified by the calibration current variable for a period of microseconds specified by the calibration time variable. **Mode=1** – Shaft snap minimalization mode: Current slowly ramps from 0 to a percent specified using the calibration current variable and holds it for a time in microseconds specified using the calibration time variable. Mode 1 is only available on power up.	0 – 1	1	CA

Function	Address	Bytes	Description	Range	Default	MCode
Set control bounds	0x0091	1	The control bounds are limits which configure the hybrid circuitry for best speed or torque performance. For torque mode operation the control bounds are preset for best torque performance. 0 – 1.1 full steps (best torque performance). 1 – 1.3 full steps (best overall performance). 2 – 1.5 full steps (best overall performance). 3 – 1.7 full steps (best speed performance).	0 – 3	1	CB
Set calibration current	0x0092	1	Sets the calibration current in percent (%).	1 – 100	100	CC
Clear locked rotor	0x0093	1	Will clear a locked rotor fault, re-enable the output bridge and initiate a timed calibration.	—	—	CF
Calibration time	0x0094	2	Sets the time in milliseconds which the hybrid circuitry will calibrate the rotor/stator relationship.	2 – 65535	200	CT
Lead limits	0x0095 – 0x0096	4	Sets the rotor lead limit in motor steps	0 – 2147483647	102400	LD
Lag limits	0x0097 – 0x0098	4	Sets the rotor lag limit in motor steps	0 – 2147483647	102400	LG
Position lead/lag	0x0099 – 0x009A	4	Represent the number of counts that the rotor leads or lags the stator. A positive value indicates position lag. A negative value indicates position lead	–2147483647 to +2147483647	—	LL
Locked rotor	0x009B	1	Indicates the state of the rotor as locked (1) or unlocked (0).	0/1	0	LR
Locked rotor timeout	0X009C – 0x009D	2	Sets the time in milliseconds in which the output bridge will disable after a locked rotor condition is detected.	2 – 65535	2000	LT
Make up frequency	0x009E – 0x009F	4	Sets the frequency in Hz at which missed steps are re-inserted into the move profile if make up mode = 1.	306 – 5000000	768000	MF
Make up	0x00A0	1	Sets the mode for make up steps. 0 = Off 1 = Make up steps at make up freq. (0x009E) 2 = Make up steps at system speed	0 – 2	0	MU
Start calibration	0x00A1	1	Start calibration process	—	—	SC
Set system speed	0x00A2	1	Sets maximum response frequency for fixed or variable current modes (AS=1 or 2) Frequency = 10 MHz / (SS+2)	0 – 255	0	SS
Reserved	0x00A3 – 0x00A4	—	Reserved	—	—	—
Torque direction	0x00A5	1	Sets the torque direction plus (1 – CW) or minus (0 – CCW) as seen facing the motor shaft.	0 – 1	1	TD
Set torque	0x00A6	1	Sets the motor torque in percent for torque mode operation.	1 – 100	25	TQ
Set torque speed	0x00A7		Determines the system speed for torque mode (AS=3) The device will perform the following calculation based upon the value of TS: Oscillator frequency = 10 MHz / (TS+2)	0 – 255	0	TS

5.2 User variable registers

The user variable registers are ONLY used to interact with MCode programs being executed using the Manufacturer Specific function code. They cannot be used for MODBUS/TCP standalone operation.

If using V1 – V8 to store or retrieve data, the variables must be declared within the MCode program using the VA (Create user variable) command.

Function	Address	Bytes	Description	Range	Default	MCode
V1	0x00A8 - 0x00A9	4	User variable 1	Variables may contain up to 11 digits including the sign and may be used to store and retrieve data.	—	**V1**
V2	0x00AA - 0x00AB	4	User variable 2		—	**V2**
V3	0x00AC - 0x00AD	4	User variable 3		—	**V3**
V4	0x00AE - 0x00AF	4	User variable 4		—	**V4**
V5	0x00B0 - 0x00B1	4	User variable 5		—	**V5**
V6	0x00B2 - 0x00B3	4	User variable 6		—	**V6**
V7	0x00B4 - 0x00B5	4	User variable 7		—	**V7**
V8	0x00B6 - 0x00B7	4	User variable 8		—	**V8**

Page intentionally left blank

6 TCP/IP Configuration Utility

The TCP/IP Conf guration Utility is specif cally for setting the Ethernet MDrive Motion Control IP address and basic functionality testing for use with MODBUS/TCP.

6.1 Installation

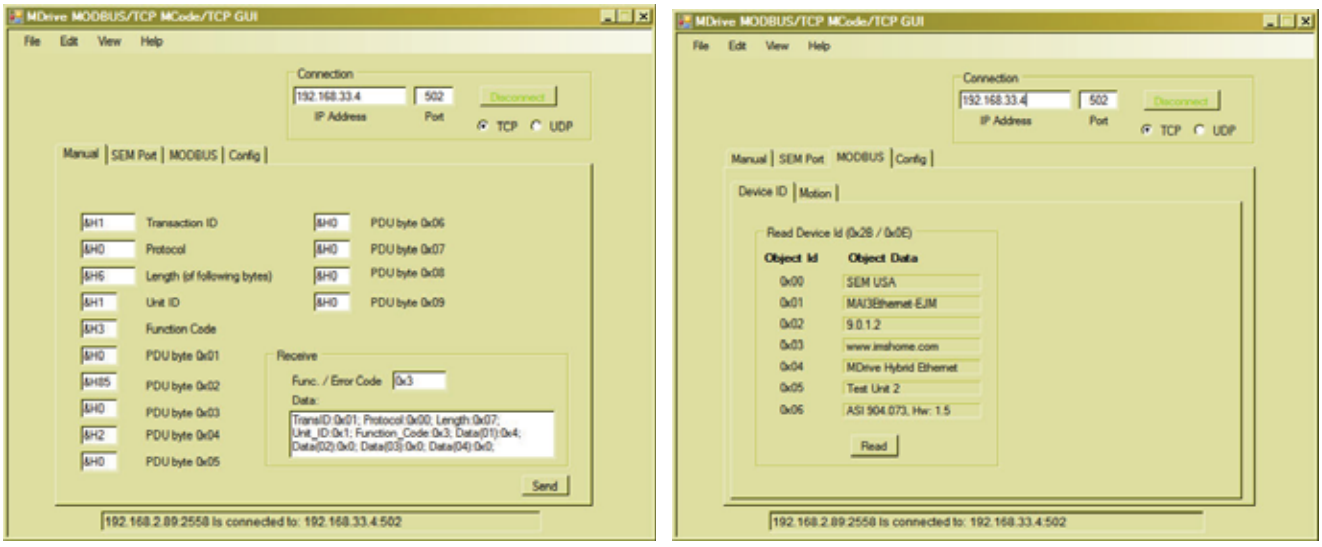
System requirements

- IBM Compatible PC.
- Windows XP Service Pack 2 or greater.
- Ethernet hub or switch with free port.

Installation

1) Download the software from the web site at http://www.imshome.com/downloads/software_interfaces.html.
2) Extract to a location on your hard drive.
3) In the folder location of the extracted f les, double click “setup. exe”
4) Follow the on-screen prompts to complete the installation.

6.2 Screen overview



Manual data entry MODBUS

Figure 6.1 TCP/IP Conf guration Utility

For MODBUS/TCP two of the 4 tabs on the GUI are available: The SEM Port and Conf g tabs will generate an error dialog if you attempt to access them in MODBUS/TCP mode.

1. Manual: Used to manually input and read function codes, register addresses and address parameters.
2. MODBUS: Has three sub-tabs for reading the device ID, reading and writing motion parameters / commands and reading / writing I/O states in decimal.

   The results are shown in hex in the Manual tab

6.3 Configuration

In order to set up the MDrive, you must f rst connect the utility in MCode/TCP mode by connecting to, the factory default IP, 192.168.33.1:503 (factory default). This is to conf gure the device IP address and Subnet Mask.

If you are on a corporate network, you may need to check with your IT department to obtain a block of private IP addresses so as not to conf ict with computers and other devices on the network.

The assigned IP address should always be within the IPv4 Private Network block (192.168.0.0 — 192.168.255.255).

MCode/TCP will always use port 503 and may communicate via TCP or UDP interchangeably.

Once conf gured, you may connect to the MODBUS/TCP port at IP.ADDRESS:502.

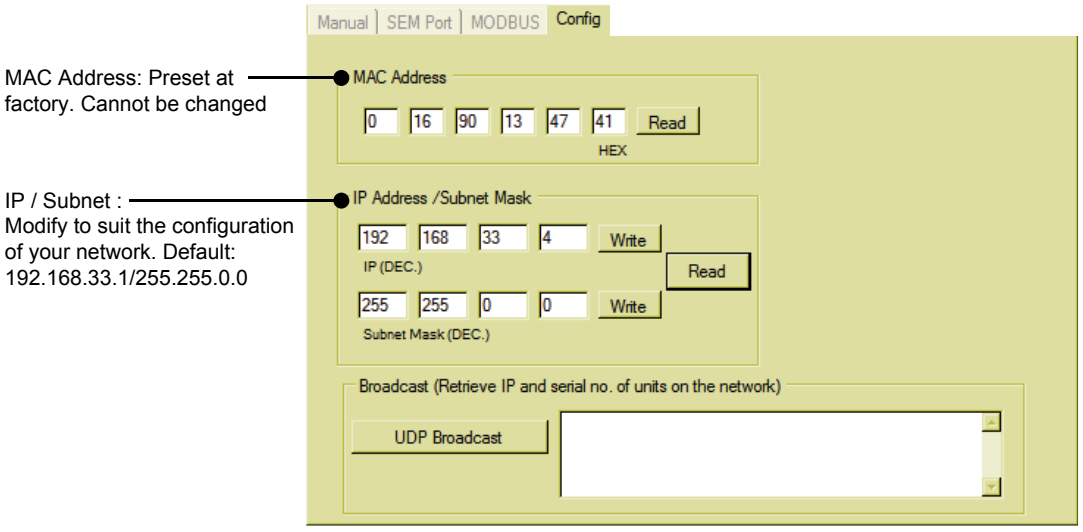


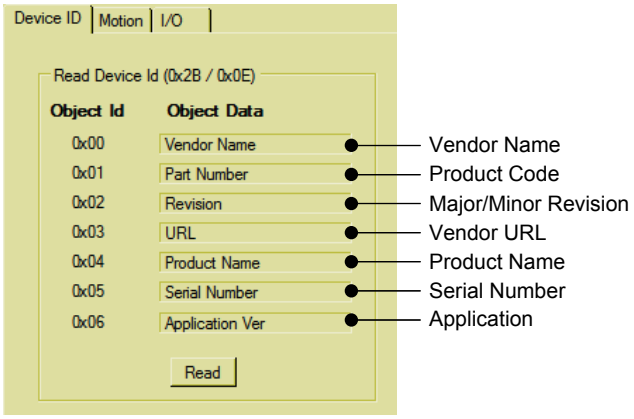
Figure 6.2: Conf guration tab

6.4 Functional testing - MODBUS tab

The MODBUS tab allows you to read the device ID information and read / write motion and I/O registers in decimal units. The last command given will show in the Manual tab as hexadecimal data.

6.4.1 Device ID sub-tab

The Device ID sub-tab allows the user to read the ID information of the device (Function code 0x2B). This information may be needed if requesting applications / technical support.



Read Device ID
Function code:0x2B
MEI Type: 0x0E

Figure 6.2: Device ID sub-tab

6.4.2 Motion sub-tab

The motion sub-tab allows access to motion variables and commands through decimal units. The MODBUS equivalent of the commands will appear on the Manual tab after the command is read or written.

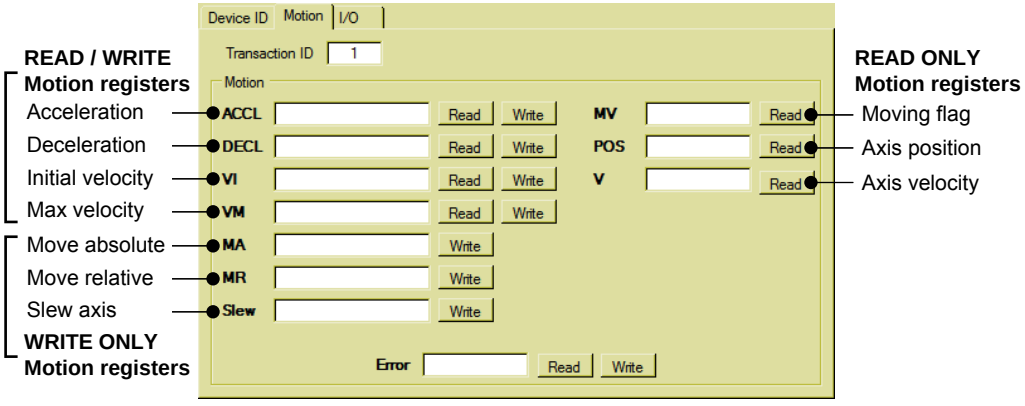


Figure 6.3 Motion sub-tab

6.4.3 I/O sub-tab

The I/O sub-tab allows the user to exercise functions 0x01 (read multiple coils), 0x05 (write single coil) and 0x0F (read multiple discrete inputs).

The read coils and inputs will display two ways:

1. The checkboxes will display as checked if the I/O point is active.
2. The total active points will display as Binary Coded Decimal (BCD) in the text f eld with a range of 0 – 15.

NOTE: The I/O points must be conf gured to the desired function, active state and sink/source using the manufacturer specif c function codes 0x41 (write specif c) and 0x42 (read specif c).

By default the I/O points are conf gured as general purpose inputs, active when low, sinking. See section 4.3 of this document.

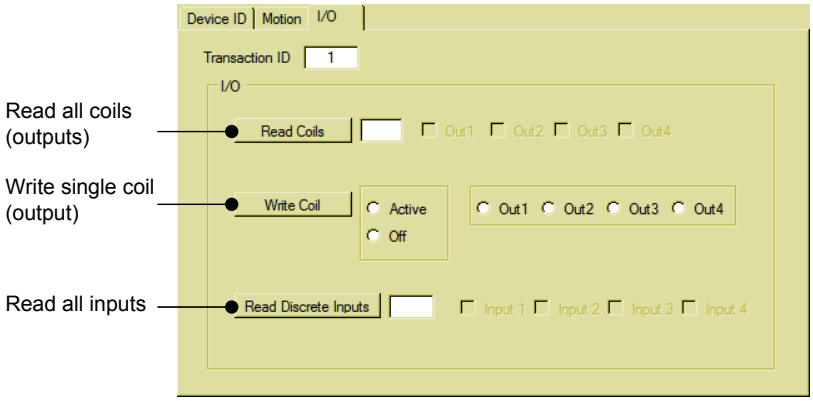


Figure 6.4: I/O sub-tab

6.5 Manual tab

The Manual tab facilitates direct entry of MODBUS functions, register addresses and parameter data. The screen itself represents the different portions of the Application Data Unit (ADU) and can be f lled out to write or read supported functions.

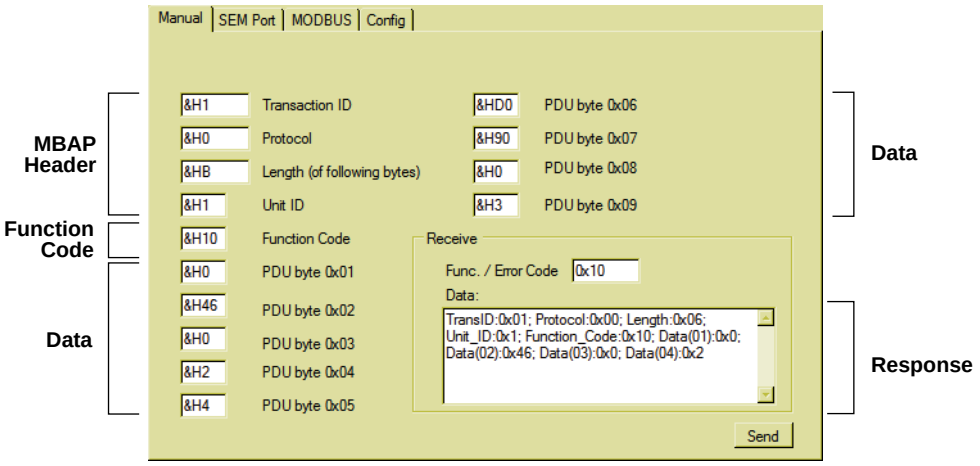


Figure 6.5: Manual entry tab

6.5.1 Using the Manual tab to write functions / data

The following illustration shows an example of writing a function, in this case a Move to Absolute position 512000.

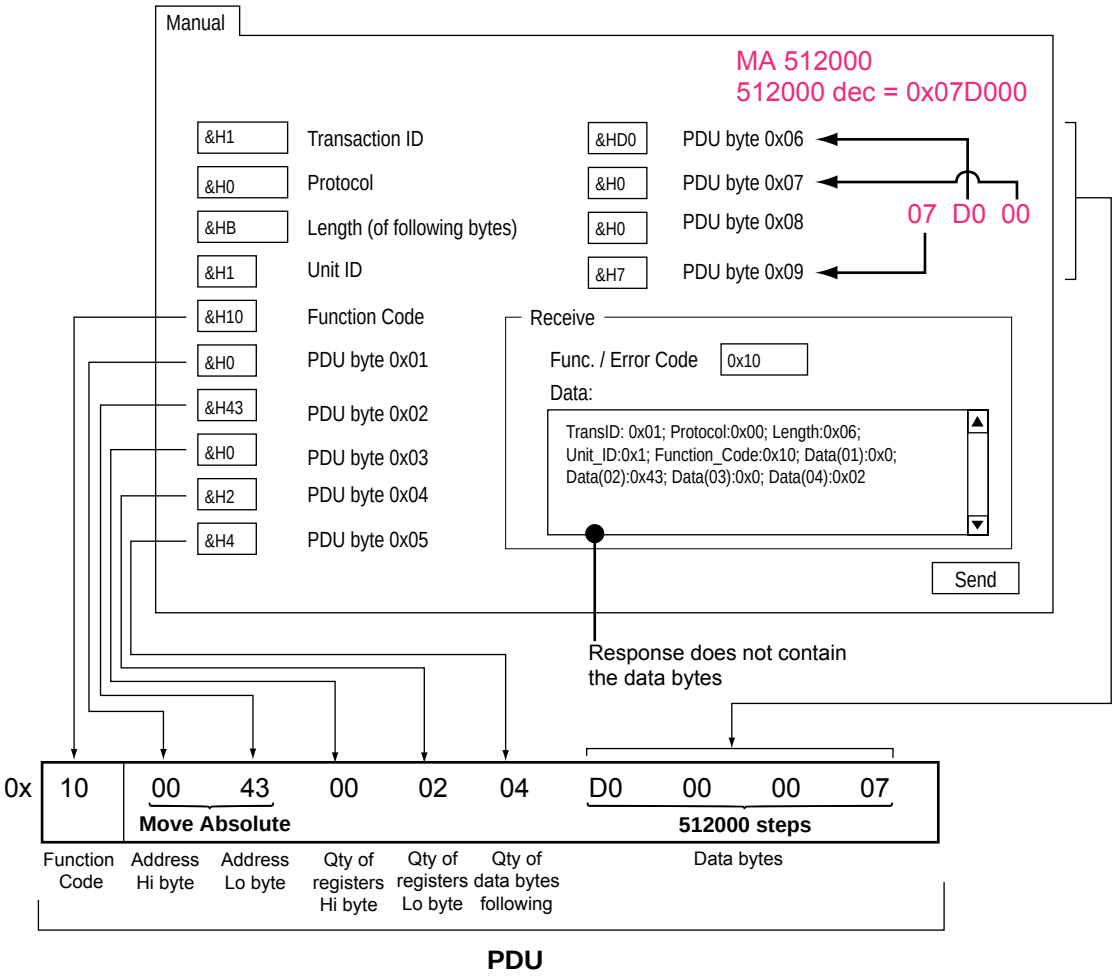


Figure 6.6: Writing a function using the Manual tab

6.5.2 Using the Manual tab to read functions / data

The following illustration shows an example of reading a function, in this case the position resulting from the absolute move in the last section.

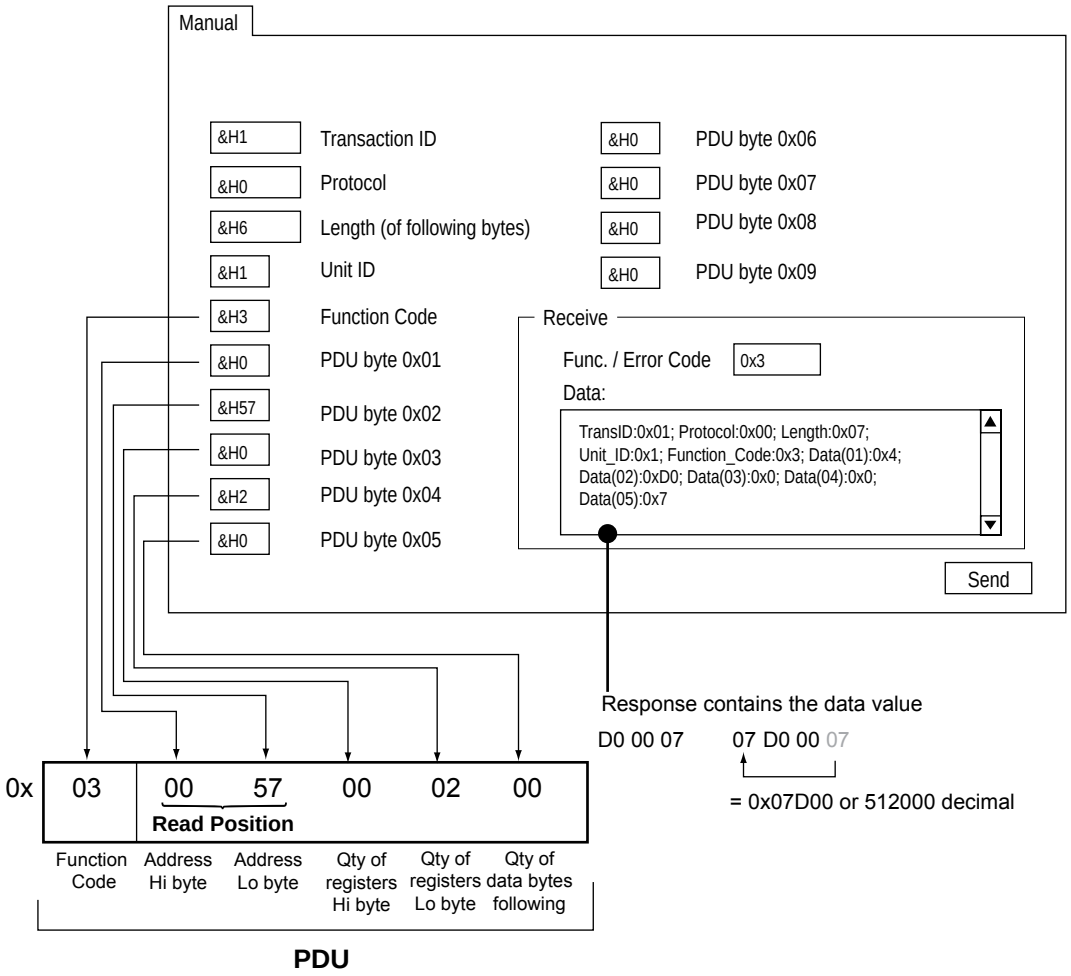


Figure 6.7: Reading a function using the Manual tab

6.6 Upgrading the Ethernet controller firmware

NOTE: This refers strictly to the controller f rmware for the Ethernet interface of Ethernet equipped MDrivePlus and MDrive Hybrid models. It is NOT an upgrade to the MDrive operating f rmware.

It is recommended that you DO NOT perform this upgrade unless so instructed by the IMS SEM Applications department.

Please review this in detail before performing the upgrade, each step must be completed in order.

Requirements

The latest versions of the software and f rmware are available on the web site under the downloads tab at http://www.schneider-electric-motion.us

1) MDrivePlus or MDrive Hybrid Motion Control with Ethernet
2) TCP/IP Conf guration Tool (Installed)
3) TFTPD Firmware Server (Installed)
4) Ethernet f rmware upgrade f le
   IMPORTANT: Unzip upgrade *.S19 f le to the installation directory of the TCP/IP Conf guration Tool

This process will utilize the f rmware upgrade area on the conf guration tab of the TCP/IP Conf guration Tool to set up.

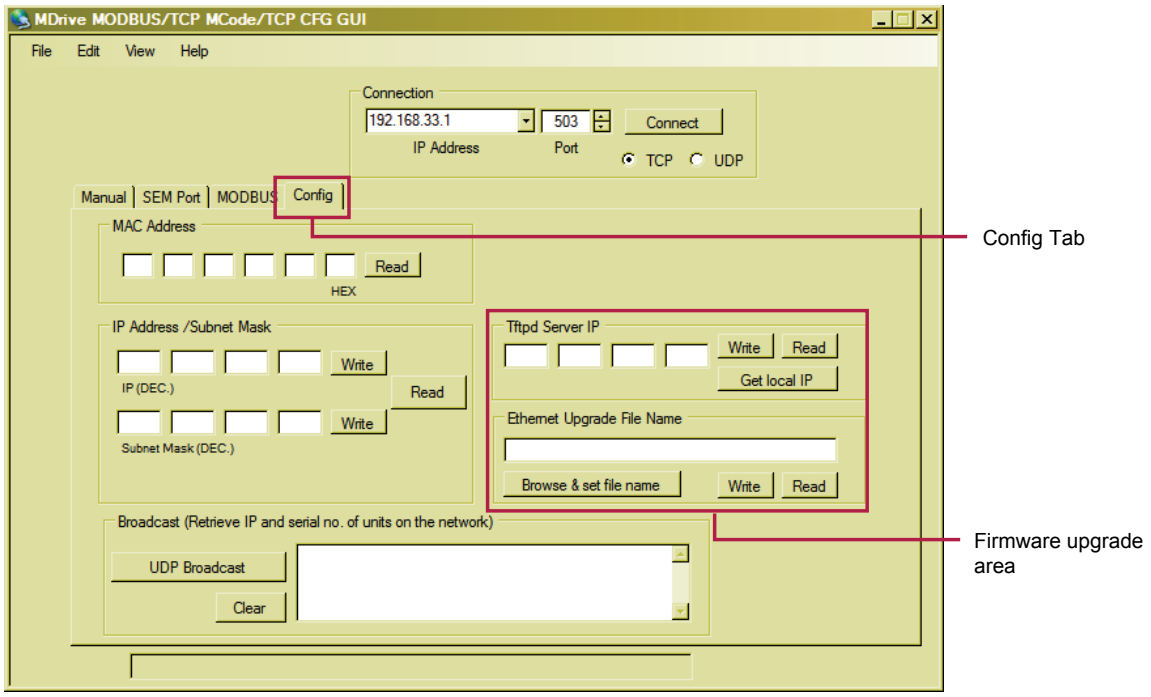


Figure 6.8: TCP/IP Conf g tab

6.6.1 To begin

1) Open the TCP/IP Conf guration Tool
2) Click the conf g tab, if not already active.
3) Connect to your Ethernet MDrive over TCP.

6.6.2 Set the Tftpd Server IP

a) Click “Get Local IP”
b) Click “Write”
c) Tftpd Server IP should read ‘OK’

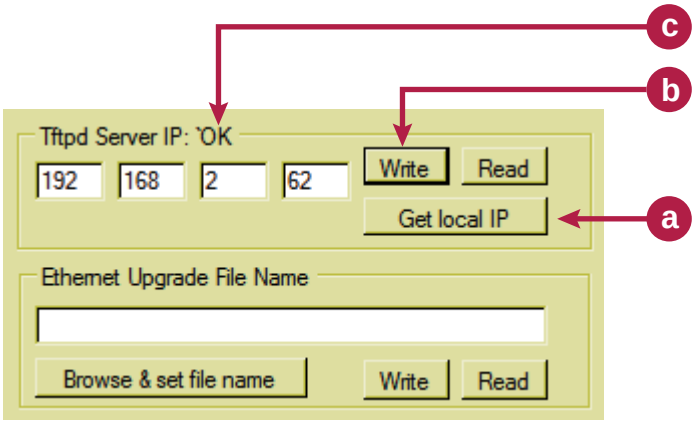


Figure 6.9: Setting the Tftpd Server IP

6.6.3 Set the Ethernet upgrade file name

a) Click “Browse & set f le name”. In the f le open window, browse to the location where you extracted the f rmware upgrade *.S19 f le. Click “OK”
b) Click “Write”
c) Ethernet Upgrade File Name should read ‘~OK’

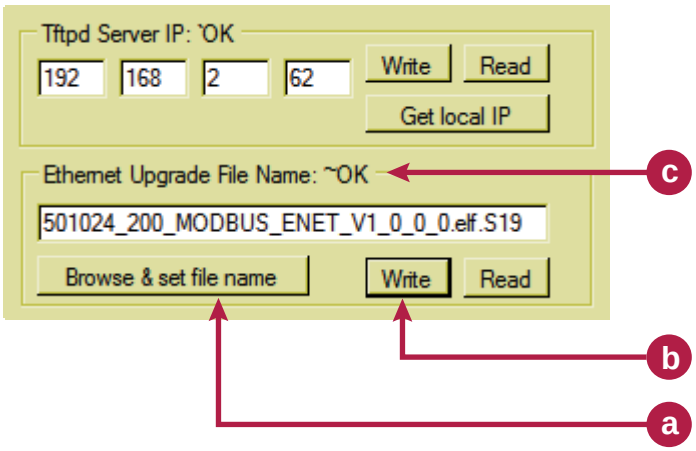


Figure 6.10: Setting the upgrade f le name

6.6.4 Enter upgrade mode

1) On the «Edit» menu, select «Enter Ethernet Firmware Upgrade Mode»
2) A dialog will open requesting verif cation of the upgrade f le-name. If the name matches, click «Yes».
3) If it does not match, click «No» and repeat step 2.
4) In the dialog, «Enter unlock code to enter upgrade mode», enter the code:

   2956102

5) The message, «Successfully entered Ethernet Firmware up-grade mode» will appear, click «OK».
6) The message «Cycle power to upgrade Ethernet f rmware via Tftpd server» will pop up. DO NOT Click OK at this point.

6.6.5 Complete upgrade process

1) Remove power from you MDrive.
2) Click “OK” on the dialog referenced in Step 3-e.
3) On the “Edit” menu, select “Select & Enter Tftpd Server”. The browse dialog should open to the install directory. If not, browse to the Tftpd_Server install directory and select “tftpd32. exe”
4) Click “Open”
5) Apply power to the MDrive
6) The upgrade should begin after a few seconds.
7) When complete, close Tftpd server. Note that there is no dialog informing you of completion. Check the tab marked “Log Viewer” to verify completion.
8) Cycle power to the MDrive.
9) Reconnect using the default IP: 192.168.33.1 and Subnet mask: 255.255.0.0.
10) Conf gure device to your system requirements.

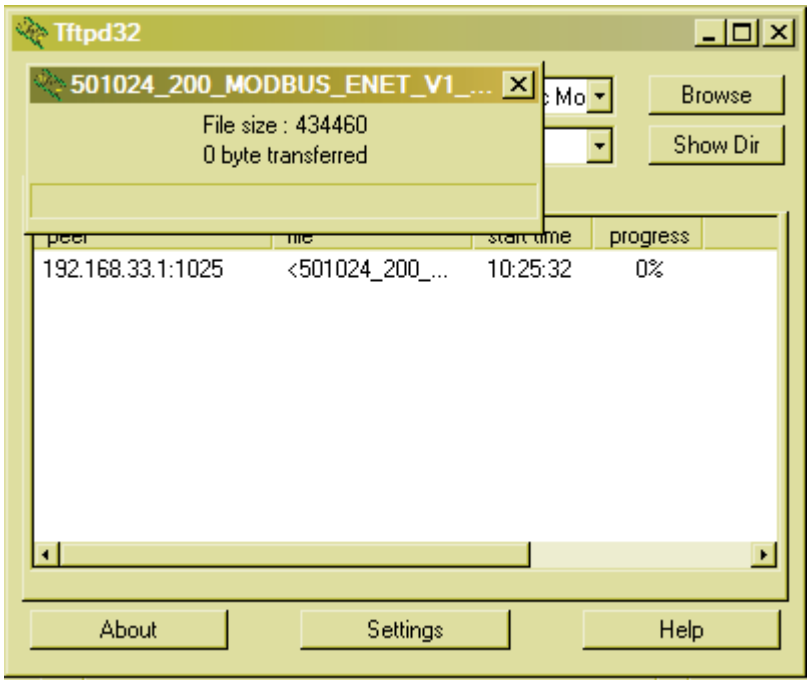


Figure 6.11: Firmware upgrading

Page intentionally left blank

WARRANTY

TWENTY-FOUR (24) MONTH LIMITED WARRANTY

IMS Schneider Electric Motion USA warrants only to the purchaser of the Product from IMS Schneider Electric Motion USA (the "Customer") that the product purchased from IMS Schneider Electric Motion USA (the "Product") will be free from defects in materials and workmanship under the normal use and service for which the Product was designed for a period of 24 months from the date of purchase of the Product by the Customer. Customer's exclusive remedy under this Limited Warranty shall be the repair or replacement, at Company's sole option, of the Product, or any part of the Product, determined by IMS Schneider Electric Motion USA to be defective. In order to exercise its warranty rights, Customer must notify Company in accordance with the instructions described under the heading "Obtaining Warranty Service".

NOTE: MDrive Motion Control electronics are not removable from the motor in the field. The entire unit must be returned to the factory for repair.

This Limited Warranty does not extend to any Product damaged by reason of alteration, accident, abuse, neglect or misuse or improper or inadequate handling; improper or inadequate wiring utilized or installed in connection with the Product; installation, operation or use of the Product not made in strict accordance with the specifications and written instructions provided by IMS; use of the Product for any purpose other than those for which it was designed; ordinary wear and tear; disasters or Acts of God; unauthorized attachments, alterations or modifications to the Product; the misuse or failure of any item or equipment connected to the Product not supplied by IMS Schneider Electric Motion USA; improper maintenance or repair of the Product; or any other reason or event not caused by IMS Schneider Electric Motion USA.

IMS SCHNEIDER ELECTRIC MOTION USA HEREBY DISCLAIMS ALL OTHER WARRANTIES, WHETHER WRITTEN OR ORAL, EXPRESS OR IMPLIED BY LAW OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. CUSTOMER'S SOLE REMEDY FOR ANY DEFECTIVE PRODUCT WILL BE AS STATED ABOVE, AND IN NO EVENT WILL IMS BE LIABLE FOR INCIDENTAL, CONSEQUENTIAL, SPECIAL OR INDIRECT DAMAGES IN CONNECTION WITH THE PRODUCT.

This Limited Warranty shall be void if the Customer fails to comply with all of the terms set forth in this Limited Warranty. This Limited Warranty is the sole warranty offered by IMS Schneider Electric Motion USA with respect to the Product. IMS Schneider Electric Motion USA does not assume any other liability in connection with the sale of the Product. No representative of IMS Schneider Electric Motion USA is authorized to extend this Limited Warranty or to change it in any manner whatsoever. No warranty applies to any party other than the original Customer.

IMS Schneider Electric Motion USA and its directors, officers, employees, subsidiaries and affiliates shall not be liable for any damages arising from any loss of equipment, loss or distortion of data, loss of time, loss or destruction of software or other property, loss of production or profits, overhead costs, claims of third parties, labor or materials, penalties or liquidated damages or punitive damages, whatsoever, whether based upon breach of warranty, breach of contract, negligence, strict liability or any other legal theory, or other losses or expenses incurred by the Customer or any third party.

OBTAINING WARRANTY SERVICE

If the Product was purchased from an IMS Schneider Electric Motion USA Distributor, please contact that Distributor to obtain a Returned Material Authorization (RMA). If the Product was purchased directly from IMS Schneider Electric Motion USA, please contact Customer Service at info@imshome.com or 860-295-6102 (Eastern Time Zone).

Customer shall prepay shipping charges for Products returned to IMS Schneider Electric Motion USA for warranty service and IMS Schneider Electric Motion USA shall pay for return of Products to Customer by ground transportation. However, Customer shall pay all shipping charges, duties and taxes for Products returned to IMS Schneider Electric Motion USA from outside the United States.

Schneider Electric Motion USA
370 North Main Street, P.O. Box 457
Marlborough, CT 06447 - U.S.A.
Tel. +00 (1) 860 295-6102 - Fax +00 (1) 860 295-6107
e-mail: info@imshome.com
http://www.motion.schneider-electric.com

 REV040611
Product Disclaimer and most recent product information at www.motion.schneider-electric.com