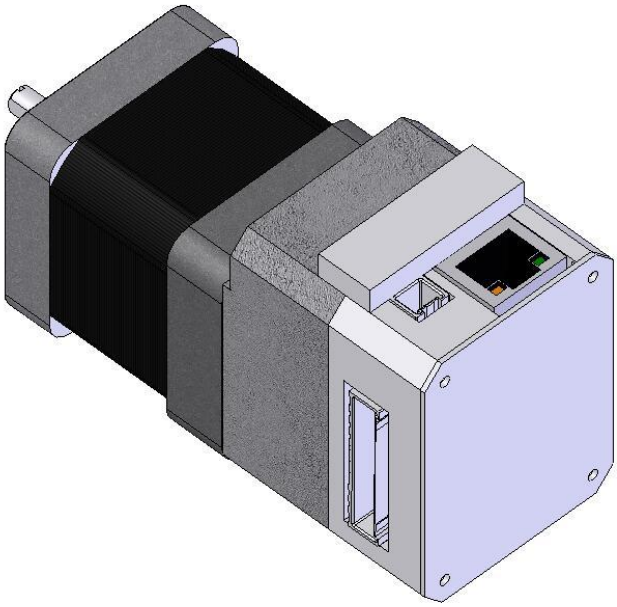
Myostat Motion Control Inc

Cool Muscle 1

EtherNet/IP Interface Manual

Document Version 1.07



Myostat Motion Control Inc
+1 905 836-4441
www.coolmuscle.com

1 Table of Contents

2 Introduction

This document details the optional Ethernet/IP Communications Interface available for use with Cool Muscle 1 (CM1) Series integrated servo motors. The interface provides connectivity between an Ethernet/IP scanner device and the stand-alone motion controller of a single CM1 device. Communications is available via Class 1 I/O Messaging (UDP) in which the scanner device establishes a connection with the CM1 device for cyclical production and consumption of I/O data. This manual serves as a reference for the EtherNet/IP communications interface only. All documentation relevant to Cool Muscle hardware and software is available at the Myostat Motion Control website http://www.coolmuscle.com.

3 Assembly Object – Class Code: 0x04

The Assembly object binds multiple input and output data into a single object. This data can then be set and received over a single I/O connection. All data is sent and received in hexadecimal format.

3.1 Input (T→O) Assembly Object – Instance 0x65 (0d101)

Motor data is transmitted to the scanner device via the standard EtherNet/IP Assembly Object Instance 0x65. Data is transmitted to the scanner device via a series of 8-bit Single Integers (SINT). The object has the following call parameters:

Service	0x0E
Class	0x04
Instance	0x65
Attribute	0x03

Data is returned as eight 8-bit SINT values in 8-bit atomic element size little endian encoding. The input data is organized as shown below:

Bytes	Data
0-1	Communication Error Code
2-3	Motor Status Code
4-7	Motor Position

Example:

Bytes	0-1	2-3	4-7
Data	00 00	08 00	A1 14 00 00
Value	Comm. Error Code = 0	Motor Status Code = 8	Motor Position = 5281

3.1.1 Communication Error Code

Bit (0=LSB)	Status Flag
0	Communication Error – motor controller is responding with communication error codes
1	Communication Error – communication with motor controller has timed out (was previously established)
2	Communication Error – communication with motor controller was never established
3-7	Reserved
8	EtherNet/IP module has defaults loaded and needs motor configuration loaded
9	EtherNet/IP module has motor configuration loaded and will initialize on reset
10-15	Reserved

The communication error code can also be read from:

Service	0x0E
Class	0x64
Instance	0x1
Attribute	0x64

3.1.2 **Motor Status Code:**

Value	*Error*	*Status*
0	NO	Motor Running
1	YES	Position Error overflow – motor disabled
2	YES	Over Speed – motor disabled
4	YES	Over Current – motor disable
8	NO	Motor is position
16	NO	Motor windings disabled
128	YES	Over Temperature Limit – motor disabled
512	NO	Emergency Stop Enabled – motor disabled

3.2 Output (O→T) Assembly Object – Instance 0x66 (0d102)

Motor run and setup data is written through the standard EtherNet/IP Assembly Object Instance 0x66. Data is received by the scanner device via a series of 8-bit Single Integers (SINT). The object has the following call parameters:

Service	0x10
Class	0x04
Instance	0x66
Attribute	0x03

Data is written as fourteen 8-bit SINT values in 8-bit atomic element size little endian encoding. All data bytes are 0 on power up. The output data is organized as follow:

Bytes	*Description*
0-1	Communication to motor controller run/idle flag
2-5	Position Word (32 bit)
6-9	Speed Word (32 bit)
10-11	Acceleration Word (16 bit)
12	Control Word
13	IN2/OUT2 control (K52) (0-unchanged, 1-communication)

Example:

Data: 01 00 40 42 0F 00 64 00 00 00 0A 00 01 00

Bytes	0-1	2-5	6-9
Data	01 00	40 42 0F 00	64 00 00 00
Value	Run/Idle flag=1 (TRUE)	Motor Position=100000	Motor Speed=100
Bytes	10-11	12	13
Data	00 0A	01	00
Value	Motor Accel=10	ControlWord=1 (Start)	IN2/OUT2 control

See functions descriptions below for value descriptions.

3.2.1 Communication To Motor Controller Flag

This flag can be used to disable the write functionality of the EtherNet/IP module to the motor controller. This should always be set to 1 (write) once the complete register has been initialized. The initial state of the Output register has all bytes as 0. Initialize all bytes and set the run/idle flag to 1.

Bytes	Value	Description
0-1	0	Idle – communication (initial power-up state)
	1	Run - communication

Note: Set this value to 1 once the complete register has been initialized.

3.2.2 Position Profile Parameters

Profile position parameters define the 4 parameters used in a move profile. They set the absolute target position, the speed, acceleration and torque used to get to that position.

Bytes	Size	Units	Description
2-5	Long Integer	pulses*	Position (P0) – Absolute target position.
6-9	Long Integer	pulses/second*	Speed (S0) – Speed defined for position move.
10-11	Short Integer	pulses/s^2	Acceleration (A0) – Acceleration defined for position move.

*Please see parameter K37 for motor resolution and further information. Default resolution is 1000 pulses/revolution and a speed unit of 100 pulses/s.

3.2.3 ControlWord

Byte	Value	Description
12	0	Do nothing (used as a transition state)
	1	Start position profile move
	2	Stop the motor (uses deceleration defined in Motion Profile – Acceleration)
	3	Enable motor (can be used to re-enable after a motor error)
	4	Disable motor
	5	Home motor – start the motor's home routine

3.2.4 **IN2/OUT2 Control**

IN2 and OUT2 can be configured as a serial port to allow direct communication to the motor. This communication port can be used for debugging and setting up the motor through standard CML functions. For example, the motor resolution can be changed from the default 1000 pulses per rev to another resolution. The use of the serial port will require a specific cable (part # CM1US7-1800).

Once the motor is in communication mode it must be changed back through the serial port in CWLite by sending K52=0.

Byte	Value	Function Description
13	0	0 – unchanged (current K52* setting)
	1	1 – communication enabled (K52=10)

*See Appendix A for further description

3.3 Configuration Assembly Object – Instance 0x80 (0d128)

The Ethernet/IP communications interface does not implement the Configuration Assembly Object. However, some Ethernet/IP scanner (client) devices require a Configuration Assembly Object to be specified when configuring server devices. In this case Instance 0x80 should be used with a data length of 0 bytes.

4 RSLogix Configuration

This section details how to add the Ethernet/IP communications module as an I/O Module to the RSLogix software environment in a step-by-step fashion. This requires the device to be first registered via the RSLinx software. Once the configuration process is complete three new global controller tags (Input, Output, Configuration) will automatically be generated.

1. Ensure RSLogix is in offline mode.
2. Right click on the 1756-ENET/B Ethernet Bridge (or comparable) node and select “Add Module”.
3. Select the module type “Generic Ethernet Module” and then click “OK”.
4. Type an identifier for the Ethernet/IP communications module in the “Name” field. This will identify controller tags.
5. Under Comm Format select “Data - SINT”
6. Click the “IP Address” radio button under the “Address/Host Name” label. Enter the module IP address.
7. Under the “Connection Parameters” label enter the following settings:

	Assembly Instance (Decimal)	**Size (8-bit)**
Input	101	8
Output	102	14
Configuration	128	0

8. Click the “Connection” tab in the Module Properties dialog box.
9. In the “Requested Packet Interval (RPI)” field enter the value 10 (ms).
10. Click “OK”. The RSLogix environment is now configured to communicate with the EIP interface module.

Appendix A – Motor K Parameters

K #	Description	Unit	Values
K21	Semi/full closed loop	0.1 deg	0: full closed loop 1-36: vector angle
K22	Time delay for semi closed loop	msec	min: 10 max: 1000
K23	Event status		0:Polling only 1: All alarm and motor status codes 2:Input status 4: Output status 8: Disable echo 16: Enable warnings and messages 32: Merge event - "Mx"
K24	Quadrature output interval (see K34)	pulses	min: 10 max: 32767
K25	Time delay for slow signal response	0.1 sec	min: 1 max:9
K26	Invert input signal		0: True 1: False
K27	Input function at logical high Syntax: K27=$N_4N_3N_2N_1$ N_4 - Input 4 digit N_3 - Input 3 digit N_2 - Input 2 digit N_1 - Input 1 digit E.g. K27=2347		0: No Action 1: General Use 2: Origin Sensor 3: Manual Feed CW 4: Manual Feed CCW 5: Output Index signal not Inposition Signal 6: CW limit Switch and origin switch 7: Emergency Stop 8: Full Stop 9: CCW Limit and origin switch
K28	Input function at the rising edge of Quick Response Signal Syntax: Same as K27		0:No Action 1:Alarm reset/Pause 2:Motor Free 3:Reset Counter 4:Execute Next Step 5:Execute Previous Step 6:Execute Bank 1 7:Go Origin 8:Jog CW (Execute Bank 2 when K36=2) 9:Jog CCW(Execute Bank 3 when K36=2)
K29	Input function at the falling edge of Quick Response Signal		Same Functions as K28 except 2: Enable Motor
K30	Input function at target voltage level of Slow Response Signal		Same Functions as K27
K31	Input function at the rising edge of Slow Response Signal		Same Functions as K28
K32	Input function at the falling edge of Slow Response Signal		Same Functions as K28 except 2: Enable Motor
K33	Output Logic		0:Normally open 1: Normally closed
K34	Output function Syntax K34=N_2N_1 N_2 - Output 2 digit N_1 - Output 1 digit E.g. K34=82		0:Command 1:Inposition 2:Alarm 3:CML O1/F1 4:CML O2/F2 5:Analog Output 6:merge motion 7:Quadrature phase output 8: Motor Free 9: Torque Limit reached - Push Mode Only

K #	Description	Unit	Values
K35	Analog output function		0: Target position 1: Target position magnified by 8 2: Current Position 3: Current Position magnified by 8 4: Position Error 5: Position Error magnified by 8 6: Current Velocity/16 7: Current Velocity /2 8:I q Real 9: Iq*8
K36	Pulse interface		0:CW/CCW 1:Step/Direction C type - 2: enables bank 2 and 3 execution
K37	Resolution and speed unit	Speed unit: 100pps Speed unit: 10pps Speed unit: 1pps	0:200, 1:400, 2:500, 3:1000, 4:2000, 5:2500, 6:5000, 7:10000, 8:25000, 10:50000, 40:300, 42:600, 43:800, 44:1200, 45:1500, 46:3000, 47:4000, 48:6000, 49:8000, 50:12000 20:200, 21:400, 22:500, 23:1000, 24:2000, 25:2500, 26:5000, 27:10000, 28:25000,, 30: 50000 60:300, 62:600, 63:800, 64:1200, 65:1500, 66:3000, 67:4000, 68:6000, 69:8000, 70:12000 100: 50000
K38	Analog interface		0:Speed Control 1:Position Control
K39	Voltage filter gain	5[rad/sec]	Min:0 Max1028
K40	Max speed for analog control (K64)	rpm	max speed at 4.8V
K41	Travel Range for analog control (K64)	Pulses	Min: -32767 Max: 32767
K42	Go origin speed	100pps	Min:1 Max: 5000
K43	Go origin/manual feed acceleration	kpps^2	Min: 1 Max: 5000
K44	Deceleration ratio	%	Min: 10 Max: 500
K45	Origin direction		0: CW 1: CCW 2: CW with reverse coordinates 3: CCW with reverse coordinates
K46	Origin search method		0:Stopper 1:Stopper(start search on power up) 2:Origin Switch 3:Origin Switch (start search on power up) 16: same as 0 but power up disabled 17: same as 1 but power up disabled 18: same as 2 but power up disabled 19: same as 3 but power up disabled
K47	Origin Stopper Voltage Level	%	Min:10 Max:100
K48	Offset distance between machine origin and mechanical origin	100 pulses	Min: -32767 Max: 32767
K49	Manual feed speed	100pps	Min:1 Max:5000
K50	Manual Jog travel distance	Pulses	Min: 1 Max: 100

K #	Description	Unit	Values
K51	Creeping speed	100pps	Min: 1 Max:1000
K52	Digital/Serial IO 1 and 2		0:Auto detect 1: Force Serial port 2:Force Digital port Note: IO1 cannot be forced to digital
K54	Quadrature output offset	pulses	Min: 0 Max: 32767
K55	Inposition tolerance	Pulses	Min: 1 Max 1000
K56	Position error overflow alarm level	Kpulses	Min: 1 Max: 32767
K57	Overload alarm time delay	msec	Min:100 Max:10000
K58	Software Limit (+)	100 pulses	Min: 0 (off) Max: 32767
K59	Software Limit (-)	100 pulses	Min: -32767 Max: 0 (off)
K60	Pushmode current level	%	Min: 10 Max: 80 NOTE: pushmode % is based on 80% of full torque
K61	Push time	msec	Min: 1 Max: 30001 (infinite push)
K63	External encoder input		0:None 1:Phase A only 2:Phase A and B 3: Enable "Fx" and "Cx" variables
K64	Analog input function		0:None, 1:S0, 2:P0, 3:S13, 4: P24 5:S14, 6:P25, 7:Speed 0-Set speed 8:Position Mulitplier 9:Analog control only (K38) NOTE: see documention on logic banks for complete control with analog input
K66	Data Streaming		0: None. 1: Send back speed target 2: Send back real position 3: Send back real speed 4: Send back real current Iq 5: Position Real 6: Velocity Real

K #	Description	Unit	Values
K67	Data Streaming sample timing	msec	0-3000
K68	S curve Function		0:S Curve with fixed timing 1:S Curve without timing
K69	S Curve Gain		0~1024
K70	Send carriage return		0: No line feed after carriage return 1: Line feed afer carriage return
K71	Temperature limit	Deg C	Min: 0 Max: 150
K72	Regeneration voltage return level	0.1V	Min: 0 Max: 391
K73	Merge motion output signal length	msec	Min: 1 Max: 1000
K74	External Torque feedback P-Gain		Min: 0 Max: 1000
K75	External Torque feedback I-Gain		Min: 0 Max: 500
K77	External Torque feedback mean value	4.88mV	Min: 0 Max: 1024
K78	External Torque feedback Gain		Min:-1024 Max: 1024
K85	Logic bank number to start on powerup		Min: 0 (no bank) Max: 30
K86	Coordiniated motion - Synchronize motors		0: Off 1: On
K87	Logic bank scan period	msec	Min: 1 Max: 32767
K88	External encoder resolution		Min: 0 Max: 50000