

EtherNet/IP with Applied Motion Drives

Jeff Kordik
CTO
Applied Motion Products, Inc.



Contents

Overview of EtherNet/IP	3
EtherNet/IP Implementation	3
Device Profile	3
Attributes	4
Explicit Message Types	4
Motion	5
I/O Commands	5
Math and Register Manipulation	6
Polling	6
Configuration Commands	7
Q Program Commands	7
EtherNet/IP And Q Programs	8
Starting a Q Segment	8
Stopping a Q Program	8
Communicating with a Q Program While It's Running	8
How to Know if a Q Program Has Stopped	9
Checking Status While a Q Program is Running	9
Downloading a Q Program over EtherNet/IP	9
EtherNet/IP Drive Models	10
ST5-IP-EN, ST5-IP-EE	10
ST10-IP-EN, ST10-IP-EE	10
STAC5-IP-N120, STAC5-IP-E120	10
STAC5-IP-N220, STAC5-IP-E220	11
Performance (ST Torque Curves)	11
Performance (STAC5 Torque Curves)	11
About Applied Motion Products	12
Further Reading	12

Overview of EtherNet/IP

Reprinted by courtesy of ODVA. Copyright 2006-2008 ODVA, Inc.

EtherNet/IP was introduced in 2001 and today is the most developed, proven and complete industrial Ethernet network solution available for manufacturing automation. EtherNet/IP is a member of a family of networks that implements the Common Industrial Protocol (CIP) at its upper layers. CIP encompasses a comprehensive suite of messages and services for a variety of manufacturing automation applications, including control, safety, synchronization, motion, configuration and information. As a truly media independent protocol that is supported by hundreds of vendors around the world, CIP provides users with a unified communication architecture throughout the manufacturing enterprise.

EtherNet/IP offers several unique advantages for manufacturing automation applications:

- Complete producer-consumer services let you simultaneously control, configure and collect data from intelligent devices over a single network or use a single network as a backbone for multiple distributed CIP networks.
- Compatible with standard Internet protocols - e.g. HTTP, FTP, SNMP and DHCP - and standard industrial protocols for data access and exchanges such as OPC
- Compliance with IEEE Ethernet standards provides users with a choice of network interface speeds - e.g. 10, 100 Mbps and 1 Gbps - and a flexible network architecture compatible with commercially available Ethernet installation options including copper, fiber, fiber ring and wireless.

EtherNet/IP Implementation

Applied Motion provides EtherNet/IP messaging class devices for controlling step motors, servo motors and I/O. All features of Applied Motion's popular Q drives are available over EtherNet/IP, including more than 100 commands and 130 registers for controlling motion, I/O, configuration, polling, math, register manipulation and Q programming.

Q program segments can also be stored in flash memory and executed automatically at power up or triggered by EtherNet/IP messages, providing a "macro customization" capability.

All devices support explicit messaging via UCMM or Class 3 Connections on a standard EtherNet/IP port.

A UDP maintenance port is provided for drive configuration and Q programming using Windows-based software tools available from Applied Motion. A bootloader is included in the EtherNet/IP communication firmware, allowing field firmware upgrades. The firmware in the motion control processor is also field upgradeable over Ethernet.

The IP address of each device can be selected by a 16 position rotary switch, by DHCP (where the device is assigned a dynamic IP address by a server on the network) or by Windows configuration software.

Device Profile

The Applied Motion EtherNet/IP implementation includes a custom device profile for accessing manufacturer specific drive features and achieving motion control.

The service code of this custom profile is 0x3C and the class code is 0x64.

In addition to the custom profile, the following standard objects and services are implemented:

- Message Router Object
- Connection Manager
- Connection Configuration
- Port
- Ethernet Link Object
- TCP/IP Object
- Assembly
- CIP Sync Object

Attributes

Name	Service	Class	Instance	Attribute
Profile Code & ARM Firmware Version	0xE	0x64	0	1

Example response message: 06.00.8E.00.00.00.69.41.

0x69 = 105, the firmware version

0x41 = ASCII 'A', the profile code

Explicit Message Types

Two types of explicit messages can be sent to Applied Motion EtherNet/IP drives. Type 1 messages include most of the buffered SCL and Q commands*, encapsulated into EtherNet/IP packets. However, unlike SCL and Q commands that are sent over RS-232, RS-485 and standard Ethernet, Type 1 messages do not support queries. "Immediate" SCL commands also cannot be encapsulated in EtherNet/IP Type 1 messages.

The response message will always echo back the opcode and register code (if present). Also contained in the response message is the drive's status code, a bit pattern that indicates useful information such as whether there is a fault or if the motor is in motion.

Type 2 messages provide additional functionality not available with Type 1 messages, including the ability to read back settings and registers. All Type 2 commands require an 8 bit opcode and an 8 bit operand. Return values include a 16 or 32 bit response, as appropriate.

The response message will always echo back the opcode and operand from the command message. Also contained in the response message is the drive's status code, unless other information is requested (e.g. parameter read command). The status code is a bit pattern that indicates useful information such as whether there is a fault or if the motor is in motion.

Both Type 1 and Type 2 messages can be sent over a Class 3 connection, or they can be sent to the Unconnected Message Manager (UCMM).

Both command message types result in a response message even when no data is requested.

All numerical values are in two's complement. Integers are sent big endian (most significant byte first).

**For detailed SCL and Q command descriptions, please see the Host Command Reference, Appendix H. When reading the command descriptions in the main part of this manual, please be advised that the EtherNet/IP encapsulation often requires that different units, and a different range of acceptable values, be used.*

Summary of Commands

More than 100 commands and 130 registers are available for motion, I/O, configuration, polling, math, register manipulation and Q programming, summarized here by category.

Motion

- Acceleration Rate
- Deceleration Rate
- Deceleration Rate, Quick Stop
- Feed And Set Output
- Feed To Double Sensor
- Feed To Length (Relative Move)
- Feed To Position (Absolute Move)
- Feed To Sensor
- Feed To Sensor With Mask Distance
- Feed To Sensor With Safety Distance
- Follow Encoder
- Hand Wheel
- Jog Accel Rate
- Jog Decel Rate
- Jog Speed
- Motor Disable
- Motor Enable
- Move Distance/Position
- Move Speed
- Secondary Speed (for segmented moves)
- Secondary Distance/Position (for segmented moves)
- Seek Home
- Start Jogging
- Stop Moving
- Wait For Input
- Wait On Move
- Wait On Position

I/O Commands

- Analog Input Deadband
- Analog Input Filter Gain
- Analog Input Offset
- Analog Input Trigger Threshold
- Analog Position Gain
- Analog Scaling / Range
- Analog Velocity Gain

Applied Motion Products

- Brake Engage Delay
- Brake Release Delay
- Define Alarm Reset Input
- Define Brake Output
- Define End of Travel Limits
- Define Fault Output
- Define Motor Enable Input
- Define Move/Tach Output
- Disable Jog On Input
- Enable Jog On Input
- Perform Auto Offset on Analog Input
- Set Digital Input Filter
- Set Output
- Test Input
- Wait For Input

Math and Register Manipulation

These commands are used to transfer data among or operate on the 68 “Q” and “user” registers.

- Compare Registers
- Register Add
- Register And
- Register Counter
- Register Decrement
- Register Divide
- Register Increment
- Register Load Immediate
- Register Move
- Register Multiply
- Register Or
- Register Read From Memory
- Register Subtract
- Register Write To Memory
- Test Register Immediate
- Time Stamp

Polling

The following can be read over EtherNet/IP in real time

- Actual Current
- Actual Speed
- Alarm Code

- Command Buffer Status
- Command Current
- Command Voltage (Ain)
- Drive Temp
- DSP Firmware Revision Letter
- Encoder Count
- Hall Pattern (Servo Only)
- In/Out 1 Input Status
- In/Out 2 Input Status [STAC5 Only, Reads As “F” On ST]
- Output Status (Reads Back Outputs)
- Position Error
- Status Word (alarm, fault, motor enabled, moving, jogging, etc)
- Sub Model (STM Integrated Drive+Motor Only)
- Supply Voltage
- Target Speed

Configuration Commands

- Absolute Position
- Acceleration Current (STM Integrated Drive+Motor Only)
- Anti-resonance Frequency (Stepper Only)
- Anti-resonance Gain (Stepper Only)
- Control Mode
- Control Loop Gains (Servo Only)
- Encoder Function (Stepper Only)
- Encoder Position
- Encoder Resolution
- Harmonic Smoothing Gain (Stepper Only)
- Harmonic Smoothing Phase (Stepper Only)
- Idle Current (Stepper Only)
- Idle Current Delay Time (Stepper Only)
- Multi Tasking On/Off
- Position Error Fault Tolerance (Servo Only)
- Restore Factory Defaults
- Running Current
- Step Filter Frequency (Stepper Only)
- Steps/Rev

Q Program Commands

- Multi Tasking On/Off
- No Operation

- On Fault Event
- On Input Event
- Queue Call
- Queue Goto
- Queue Jump
- Queue Kill
- Queue Load Execute
- Queue Repeat
- Test Input
- Wait Delay Register
- Wait On Input
- Wait On Move
- Wait On Position
- Wait Time

EtherNet/IP And Q Programs

To provide additional functionality and autonomy, Q programs can be stored in EtherNet/IP drives. These programs can be started and stopped “on demand” using explicit messaging. The *Q Programmer* application is used to compose, download and test Q programs. Please avoid sending EtherNet/IP messages to the drive while the *Q Programmer* software is running.

To start a Q program from an EtherNet/IP message, you must send a Type 1 message with opcode 0x78 (the QX command). You’ll need to specify the Q segment number. This allows you to store up to 12 Q segments, or subprograms, and operate them independently. Q segments can also call each other once one has been started.

Starting a Q Segment

The **QX** command can be used to start any of the 12 Q segments. For details, please see Table 1 of Host Command Reference, Appendix H.

Once a Q segment has begun, Type 1 messages are no longer permitted, because the CPU is busy executing the commands in the Q segment. To stop a Q program, you must use a Type 2 “SK” message (opcode 98). Q programs also stop running if they encounter a blank line in the segment. This makes it possible to launch a segment, have it complete a task, and stop by itself.

Stopping a Q Program

The **SK** command stops a Q program. This is a Type 2 message, so it can be accepted while the commands in the Q program are executing. For more detailed information, please see Table 2 of Host Command Reference, Appendix H.

Communicating with a Q Program While It’s Running

You can use Type 2 commands to read and write registers while a Q program is running. The Q program can send information to the host by changing a register that the host is polling. Registers 0 - 9 can be polled using the Type 2 User Register Read command (opcode 9A).

The host can make changes to the Q program operation by writing to parameters that the program uses. For example, you could change the motor speed sending a parameter write message that alters VE (Type 2 message,

opcode 83, operand 1D). The speed change will take effect on the next move.

Changes that affect a Q program immediately can be made using the Write Q Register command (message type 2, opcode 9E). For example, if the motor is jogging after having been sent a CJ command, writing to register J will result in an immediate speed change. *Please note that Q register writes are not range checked, so be careful before you write.*

How to Know if a Q Program Has Stopped

Since a Q program can be launched and allowed to stop itself when it encounters a blank line, you may want to know when it stops. You can do this by polling for the status word and observing bit 14. This bit is a one if the program is executing. To fetch the status word, use the Type 2 Parameter Read command with operand 0x80 as shown below.

Checking Status While a Q Program is Running

Sending a Type 2 message with opcode 0x84 (parameter read, from Table 2 of Host Command Reference, Appendix H) with operand 0x80, (status code, from Table 3 of Host Command Reference, Appendix H) will retrieve the current status word.

Typical return values:

0001	Motor enabled, Q program not running
4001	Motor enabled, Q program running
4801	Motor enabled, Q running, Wait Time command executing
4019	Motor enabled, motor moving, Q running

For more information about the status code, please read about the SC command in the main part of the Host Command Reference.

Downloading a Q Program over EtherNet/IP

The preferred method for creating, downloading and testing Q programs is to use the *Q Programmer* software from Applied Motion Products. When necessary, encoding and downloading Q segments over EtherNet/IP can be done by performing the following steps:

1. Develop and test your program using *Q Programmer* so that you know the final contents of the Q segment(s) you'll need. Any Type 1 command can be used in a Q program.
2. Encode each command into a Type 1 message, according to Table 1 of Host Command Reference, Appendix H.
3. Issue the QL (Queue Load) Type 2 command (see Table 2 of Host Command Reference, Appendix H).
4. Begin sending the encoded Q commands that you want in this segment. They will be placed into the Q buffer.
5. After sending the entire contents of a segment, issue the Type 2 "QS" command, which instructs the drive to save the Q buffer as a Q segment.
6. Repeat steps 2 - 5 if you have additional Q segments.
7. When you have completed the download process for all segments (steps 1 - 6), upload your program using *Q Programmer* to make sure that there were no mistakes.

EtherNet/IP Drive Models

ST5-IP-EN, ST5-IP-EE

- 24-48 VDC powered
- 5A/phase
- 100 MBit Ethernet
- Anti-resonance and electronic damping
- Encoder feedback (ST5-IP-EE)
- Four optically isolated digital outputs
- Eight optically isolated digital inputs
- Two 12 bit +/-10V analog inputs
- CE mark
- Drives high torque NEMA sizes 11, 14, 17, 23 and 24 step motors



ST10-IP-EN, ST10-IP-EE

- 24-80 VDC powered
- 10A/phase
- 100 MBit Ethernet
- Anti-resonance and electronic damping
- Encoder feedback (ST10-IP-EE)
- Four optically isolated digital outputs
- Eight optically isolated digital inputs
- Two 12 bit +/-10V analog inputs
- CE mark
- Drives high torque NEMA sizes 23, 24 and 34 step motors

STAC5-IP-N120, STAC5-IP-E120

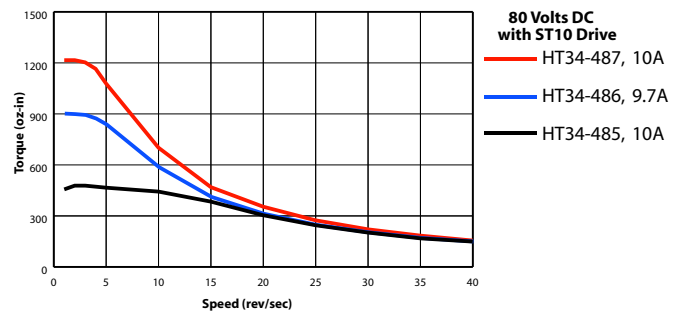
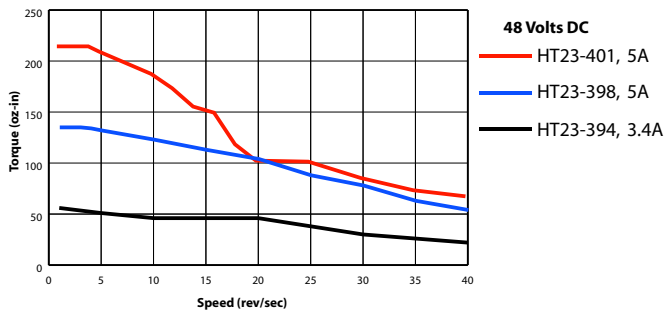
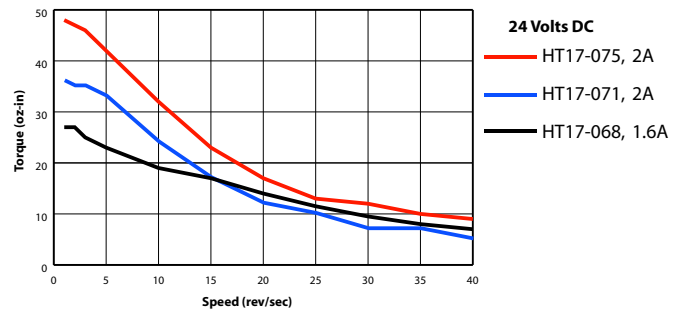
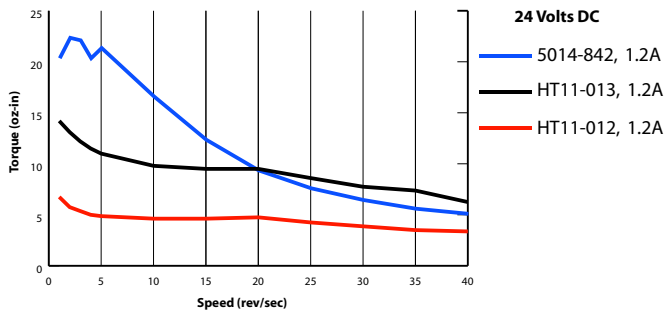
- 120 VAC powered (160 VDC bus)
- 5A/phase
- 100 MBit Ethernet
- Anti-resonance and electronic damping
- Encoder feedback (STAC5-IP-E120)
- Six optically isolated digital outputs
- 12 optically isolated digital inputs
- 12 bit +/-10V analog input
- CE mark, UL recognized
- Drives high torque NEMA sizes 23 and 34 step motors



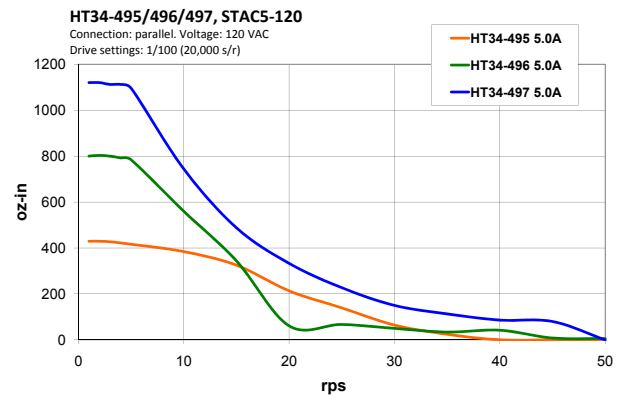
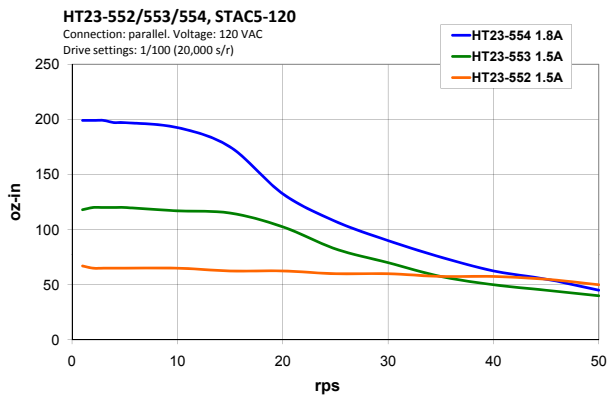
STAC5-IP-N220, STAC5-IP-E220

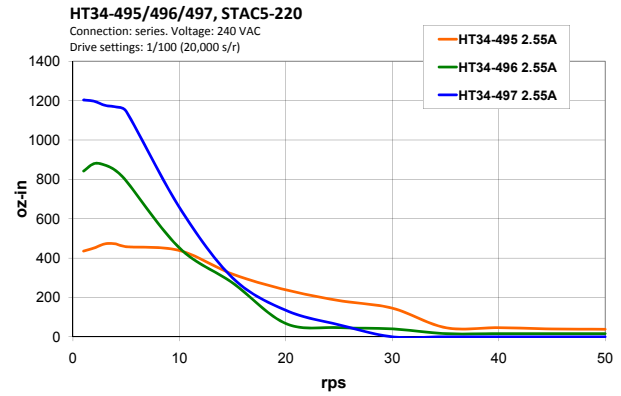
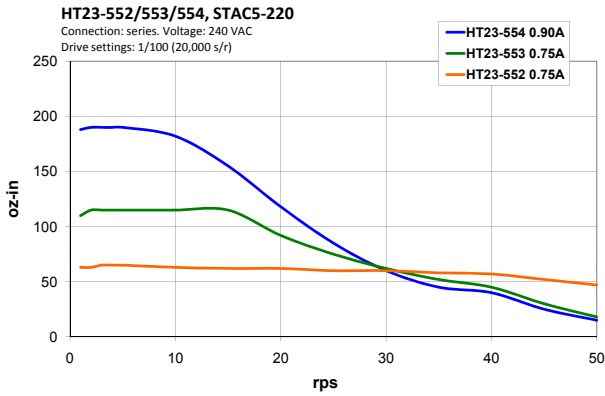
- 220 VAC powered (320 VDC bus)
- 2.6A/phase
- 100 MBit Ethernet
- Anti-resonance and electronic damping
- Encoder feedback (STAC5-IP-E220)
- Six optically isolated digital outputs
- 12 optically isolated digital inputs
- 12 bit +/-10V analog input
- CE mark, UL recognized
- Drives high torque NEMA sizes 23 and 34 step motors

Performance (ST Torque Curves)



Performance (STAC5 Torque Curves)





About Applied Motion Products

Applied Motion Products, founded in 1978, specializes in high-precision, cost-effective motion control products. We offer a full complement of stepper and servo, drives, motors, controllers, gearheads, and power supplies to serve a diverse industrial and OEM customer base.

Applied Motion offers a powerful range of standard products to meet your motion control requirements. If your application demands something unusual or unique, we can engineer a design to optimize the product's performance for your needs. This ability to offer customized products often results in improved productivity and higher reliability for our customers, often while reducing costs.

We believe the key to a successful business relationship is our commitment to fully understanding our customers' goals and the environment in which our products must perform. This is achieved by working closely with our customers, often participating in the design cycle from initial concept to finished product, all the while providing needed motor and control expertise. Our engineering experience allows us to offer customers a valuable service in the design of their products and the selection of the best motion control solution.

Applied Motion is committed to providing you with outstanding quality and the best customer service experience in the industry. Additional information, technical assistance, and off-the-shelf delivery can be provided by contacting us at 1-800-525-1609. Let us help you find the right solution for your application.

Further Reading

For more information, please read these publications from the Applied Motion web site:

- [Host Command Reference](#)
- [ST Hardware Manual](#)
- [STAC5 Hardware Manual](#)