CLEARPATH-IP SOFTWARE REFERENCE

VERSION 1.1 / NOVEMBER 20, 2025

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

Safety Information

Precautionary Statement

Always follow appropriate safety precautions when installing and using automated motion control equipment. Motion control systems should be designed and utilized to prevent personnel from coming into contact with moving parts and electrical contacts that could cause injury or death. Read all cautions, warnings, and notes before attempting to install or operate this device. Follow all applicable codes and standards when using this equipment. Failure to use this equipment as described may impair or neutralize protections built into the product.

General Disclaimer

The User is responsible for determining the suitability of this product for his/her application. The User must ensure that Teknic's products are installed and utilized in accordance with all local, state, federal, and private governing bodies and meet all applicable health and safety standards.

Teknic has made all reasonable efforts to accurately present the information in the published documentation and shall not be responsible for any incorrect information which may result from unintentional oversights.

Due to continuous product improvements, the product specifications as stated in the documentation are subject to change at any time and without notice. The User is responsible for consulting a representative of Teknic for detailed information and to determine any changes of information in the published documentation.

Should Teknic's products be used in an application that is safety critical, the User must provide appropriate safety testing of the products, adequate safety devices, guarding, warning notices, and machine-specific training to protect the operator and/or bystanders from injury.

Device Setup

Overview

ClearPath-IP servo motors and the I/O HUB are both configured using the ClearView 3.0 software. Refer to the ClearPath-IP User Manual for documentation about powering and interfacing with the servo motor via its diagnostic & setup port, as well as how to power the I/O HUB.

Visit Teknic's website to download the latest version of ClearView 3.0 and supporting resources for ClearPath-IP.

IP Address and Network Settings

The I/O HUB serves as the EtherNet/IP network interface for ClearPath-IP servo motors. Each I/O HUB ships from Teknic with DHCP (Dynamic Host Configuration Protocol) enabled by default. This means the I/O HUB is assigned an IP address when it connects to a DHCP server (found on most routers or office networks). DHCP can be useful during initial development and prototyping because it ensures the device is assigned a valid IP address. Afterwards, when installing ClearPath-IP on a machine, most machine builders will prefer to configure predetermined static network settings.

Setting a Static IP Address

Common EtherNet/IP tools like BOOTP or others can be used to set a static IP Address. You can also use Teknic's ClearView 3.0 setup software:

1. Apply power to the I/O HUB.
2. Open ClearView 3.0.
3. Connect a USB cable from your computer to the USB port on the I/O HUB. The I/O HUB will show up in ClearView's Device List. Note: Use a high speed, data-rated USB cable.
4. Set the I/O HUB Access Level to "Full". Access Level settings are located at upper right of the ClearView 3.0 UI.
5. Click the "Edit Network Settings..." button at bottom of UI; or, choose *Setup>Edit Network Settings*... from the main menu.
6. Follow the instructions in the Network Settings dialog window to configure for DHCP or Static IP settings.

If you'd like the I/O HUB to remain in DCHP mode, you can follow the steps above to simply view the current network settings. Ensure the I/O HUB receives power and is connected to the network via one of the two EtherNet/IP ports so it can be assigned an address.

OEM Machine Setup

OEMs that produce many iterations of the same machine or motion axis with identical load inertia should use configuration files to simplify servo setup. ClearView 3.0 allows you to save a configuration file for each I/O HUB and each servo motor. These files contain all setup parameters needed to take a factory-default device and automatically configure it to be identical to the original device. For example, the I/O HUB configuration file contains the ethernet network settings and I/O configuration, and the servo motor file contains the servo tuning parameters, safety settings, and more. Use ClearView 3.0 to load these pre-saved files instead of repeating the setup process of each device from scratch.

ClearView 3.0 is the best tool for device configuration. Contact Teknic if alternative device setup methods are needed.

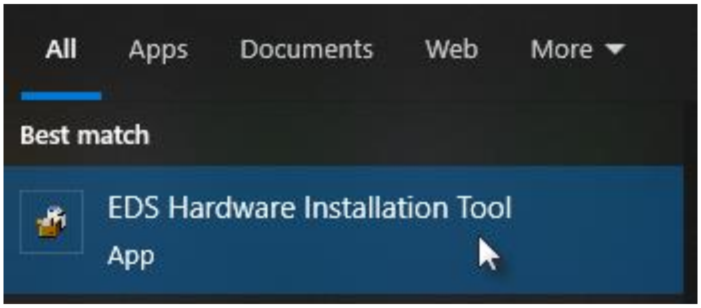
Controller Setup

ControlLogix, CompactLogix, GuardLogix (Studio 5000)

This section provides step-by-step instructions to import the Lightweight example and begin controlling ClearPath-IP using your PLC. This example will also import all Add-On Instructions (AOIs).

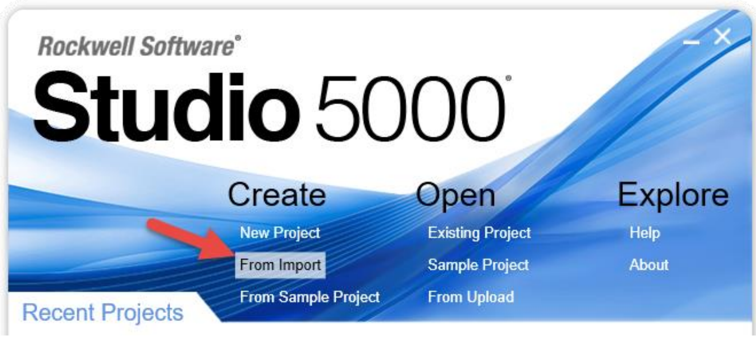
Prerequisites

- ☐ Download the ClearPath-IP EDS and examples .zip file
- ☐ Install the EDS file for your I/O HUB. Note: there are different variants of the I/O HUB; be sure you pick the correct EDS file matching your units part number.



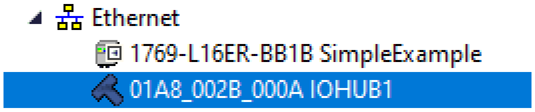
Importing the Example and AOIs

1. Open Studio 5000 Logix Designer and create a project “From Import”



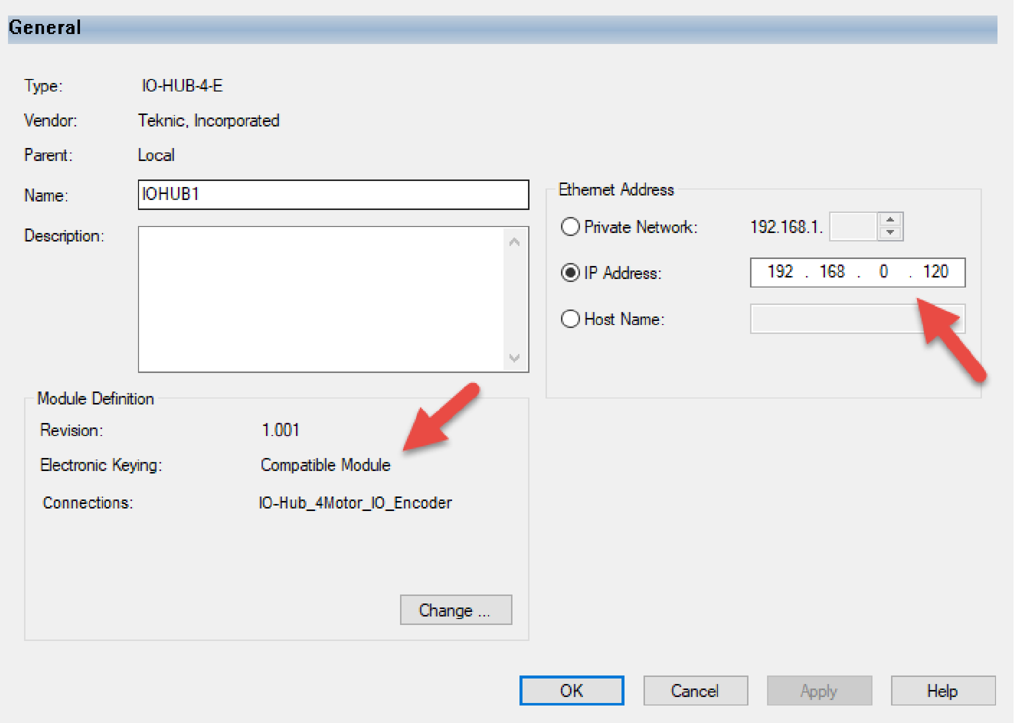
2. Select the Lightweight Example .L5K example for your I/O HUB model and Import it

3. Double click the I/O HUB module to open the Module Properties.



Ethernet
1769-L16ER-BB1B SimpleExample
01A8_002B_000A IOHUB1

4. Set the IP Address to match the real IP Address of your hub (use ClearView 3.0 to set or check the hub IP address). Ensure Electronic Keying is set to “Compatible Module”.



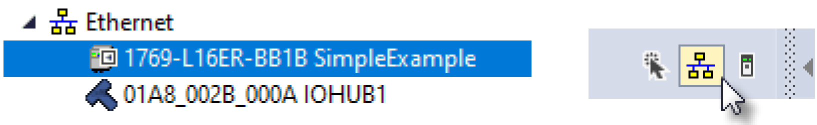
General

Type: IO-HUB-4-E
Vendor: Teknic, Incorporated
Parent: Local
Name: IOHUB1
Description:

Module Definition
Revision: 1.001
Electronic Keying: Compatible Module
Connections: IO-Hub_4Motor_IO_Encoder

Ethernet Address
○ Private Network: 192.168.1.
● IP Address: 192 . 168 . 0 . 120
○ Host Name:

5. Change Controller to match your PLC and use RSWho to set the communications path.



Ethernet
1769-L16ER-BB1B SimpleExample
01A8_002B_000A IOHUB1

Verify Communication and Run the Example

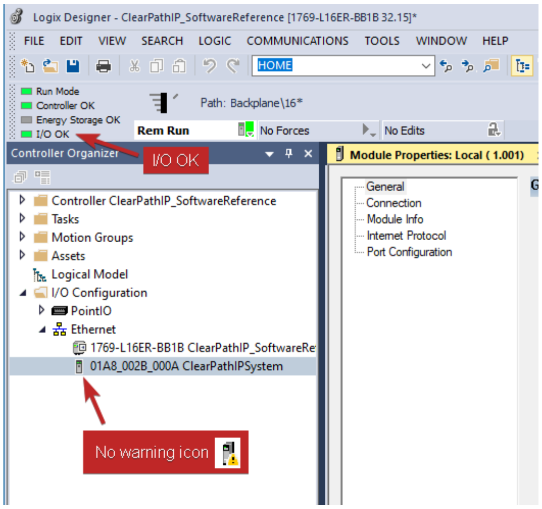
Important: The Lightweight example will automatically jog the motor back and forth. Follow all precautions outlined in the ClearPath-IP User Manual (found on Teknic's website).

Download the project to your PLC and go online with the controller.

If the device is communicating properly:

☑ The I/O status indicator will be lit green and say "OK".

☑ No warning icon will be present on the device icon.


I/O OK

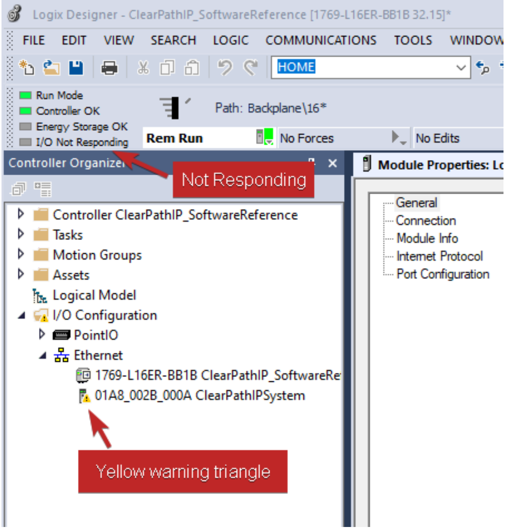
No warning icon

Logix Designer - ClearPathIP_SoftwareReference [1769-L16ER-BB1B 32.15]*
FILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS WINDOW HELP
HOME
Run Mode
Controller OK
Energy Storage OK
I/O OK
Path: Backplane\16*
Rem Run
No Forces
No Edits
Controller Organizer
Module Properties: Local (1.001)
General
Connection
Module Info
Internet Protocol
Port Configuration
Controller ClearPathIP_SoftwareReference
Tasks
Motion Groups
Assets
Logical Model
I/O Configuration
PointIO
Ethernet
1769-L16ER-BB1B ClearPathIP_SoftwareRe
01A8_002B_000A ClearPathIPSystem

If a device is NOT communicating properly:

☒ The I/O status indicator will be blinking green and say "Not Responding".

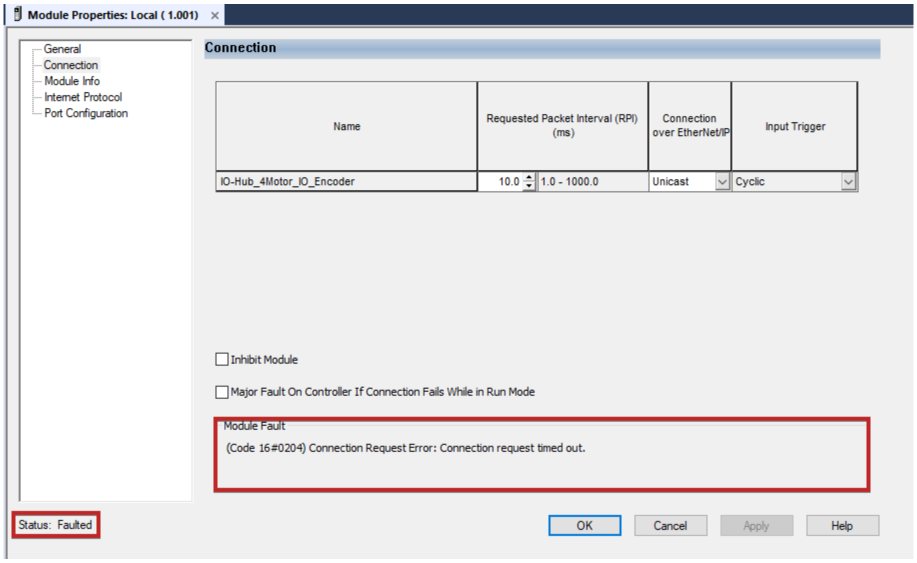
☒ A yellow triangle icon will be present on the device icon that is having communication issues.


Not Responding

Yellow warning triangle

If your device is not communicating properly, double click on it to open its Module Properties. Then go to the "Connection" section.

Observe the Status and the Module Fault error code(s):



Connection

Name	Requested Packet Interval (RPI) (ms)		Connection over EtherNet/IP	Input Trigger
IO-Hub_4Motor_IO_Encoder	10.0	1.0 - 1000.0	Unicast	Cyclic

□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

(Code 16#0204) Connection Request Error: Connection request timed out.

Status: Faulted

Use this table to help resolve communication problems depending on your error code:

Code	Description	Likely Cause
16#0204	**Connection Request Error**: Connection request timed out.	PLC can't establish a connection with your device. Check device power, network connection, and verify IP Address.
16#0116	**Electronic Keying Mismatch**: Major and/or Minor revision invalid or incorrect.	EDS File revision does not match the firmware revision of the device. The EDS file may be too old or too new.
16#0114	**Electronic Keying Mismatch**: Electronic keying product code and/or vendor ID mismatch.	EDS file does not match the device. Verify EDS file matches the device you are trying to connect to.
16#012a	**Connection Request Error**: Invalid output application path.	Keying is disabled and the EDS file does not match the device. Verify EDS file matches the device you are trying to connect to.

Add-On Instructions (AOIs)

Teknic provides a set of Add-On Instructions (AOIs) for Studio 5000 to simplify integration and control of ClearPath-IP motors. These AOIs work out of the box and mirror the structure of Allen-Bradley's native motion instructions, with select improvements. These AOIs are imported automatically with each example, or they can be imported independently.

The first instruction in your PLC logic must be the **Data Exchange** AOI. This instruction is required for any other ClearPath-IP AOI to function. It creates named motor tags (via a motor UDT) that contain named parameters—like position, velocity, and statuses—which are used as inputs and outputs for all other ClearPath-IP AOIs.

Implementation guidelines:

- ☑ Place the appropriate Data Exchange instructions at the beginning of your main routine. Each I/O HUB should have its own instance of the appropriate Data Exchange AOI.
- ☑ There is a different Data Exchange AOI for each part number I/O HUB – be sure to choose the matching AOI.
- ☑ Keep all Data Exchange AOIs enabled during the entire runtime of your program.

Lightweight Example

This basic example demonstrates the most fundamental use of ClearPath-IP's AOIs within Studio 5000. It contains these main elements:

1. A Data Exchange AOI
2. Motion Servo On (MSO) AOI – this AOI enables the servo. The servo must be enabled before it can follow any motion command.
3. Motion Axis Jog (MAJ) AOIs – jogs the motor back and forth using velocity moves
4. The example also contains Motion Axis Fault Reset (MAFR) to clear motor shutdowns when enabling (if any are present), and Motion Servo Off (MSO) to disable the servo (when the user manually toggles the EnableAndReset bit).

There is a separate Lightweight Example for each of the three different I/O HUB part numbers, as each variant uses a different Data Exchange AOI.

You may wish to expand this example by adding additional AOIs. Consider using the Integration-Ready Example if your PLC program becomes complex enough to require several different programs or routines, or for systems with multiple I/O HUBs.

Integration-Ready Example

The Integration-Ready Example can be dropped into most existing machine control programs and provide expansive and customizable functionality beyond the Lightweight Examples.

This example uses a state-based approach with several different Programs within the motion Task. Each program performs one or more functions using ClearPath-IP's AOIs. The programs are controlled using state tags in the main program. Refer to the example "README" file for more detailed instructions about using the Integration Ready Example and expanding it to meet your application requirements.

There is a separate Integration-Ready Example for each of the three different I/O HUB variants, as each variant uses a different Data Exchange AOI in the main program.

Generic EtherNet/IP Controller

Overview

ClearPath-IP is compatible with all EtherNet/IP master controllers. For ease of performance, Teknic recommends choosing a controller that supports implicit EtherNet/IP communication and whose programming environment accepts device EDS files. Controllers that only support explicit EtherNet/IP messaging can be made to work with some additional setup.

Adding an I/O HUB Device to the Controller

Most EtherNet/IP controllers use EDS files to define the target device's assembly data within the controller. If your controller does not use EDS files, you should define these assemblies manually.

Download the ClearPath-IP EDS and examples .zip file.ClearPath-IP has different EDS files, one for each part number variant of the I/O HUB. Load the appropriate EDS file of your I/O HUB variant into your PLC/Controller. Then create a module instance of the I/O HUB within your controller software. Configure the module's IP address to match the I/O HUB's address.

If your PLC does not accept EDS files, refer to the Assemblies Appendix to manually address the assemblies. PLCs that only support explicit messaging should be programmed to cyclically exchange the appropriate Input and Output assemblies.

Extract Assembly Data

The recommended way to control ClearPath-IP with any generic controller is through the same logical framework Teknic has designed for Allen Bradley's Logix 5000 controllers. To achieve this, recreate the logic of the Studio 5000 Add-on Instructions (AOIs) in your programming environment. If your PLC supports AOIs using the same implementation as Studio 5000, you may be able to recreate the AOIs in your programming environment. If not, recreate the simplified AOI logic shown in PDFs within the .zip file download.

The I/O HUB assemblies contain data for more than one ClearPath-IP motor as well as any available on-board I/O. The purpose of a "data exchange" routine or AOI is to separate out the tags for a single motor. Create an instance of this data exchange for each of your I/O HUBs. Organize a single motor's tags using a UDT (user-defined type) so other motion instructions can easily interact with each motor's data.

Implement Motion Instructions

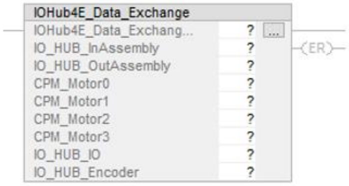
The rest of ClearPath's AOIs serve to Enable/Disable the servo, command motion, handle fault conditions, or get/set other data not contained in the assemblies. Implement the AOIs (or simplified AOI logic) needed to achieve your application requirements.

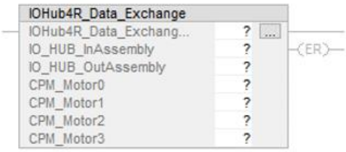
Testing communications

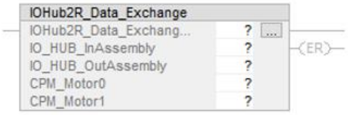
Use your implementation of the Motion Servo On (MSO) AOI to enable a servo motor. An enabled servo will actively hold its position until commanded to move.

ADD-ON INSTRUCTION (AOI) REFERENCE

DATA EXCHANGE (IOHUB_DATA_EXCHANGE)







The first instruction to include in your PLC logic is the **Data Exchange AOI**. This instruction must be present before any other ClearPath AOIs can operate. The Data Exchange AOI serves as the bridge between your PLC logic and the I/O HUB. It organizes the unstructured EtherNet/IP assembly data into custom UDT instances that can be easily used by all other ClearPath AOIs throughout your program.

There are three Data Exchange AOIs, each corresponding to a specific I/O HUB model. Be sure to use the correct one for your system:

- **IOHub4E_Data_Exchange** → IO-HUB-4E
- **IOHub4R_Data_Exchange** → IO-HUB-4R
- **IOHub2R_Data_Exchange** → IO-HUB-2R

INPUTS

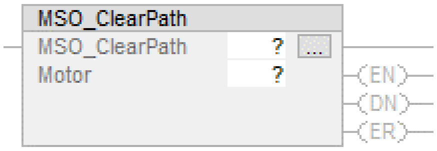
Operand	Type	Description
IOHub_ Data_Exchange	Varies	Instance of this AOI
IO_Hub_ InAssembly	Varies	The EtherNet/IP Input Assembly (tag name ends in `I:0`)
IO_Hub_ OutAssembly	Varies	The EtherNet/IP Output Assembly (tag name ends in `O:0`)
CPM_Motor0	CPM_IP_Motor	Tag representing Motor 0
CPM_Motor1	CPM_IP_Motor	Tag representing Motor 1
CPM_Motor2*	CPM_IP_Motor	Tag representing Motor 2
CPM_Motor3*	CPM_IP_Motor	Tag representing Motor 3
IO_HUB_IO**	IO_HUB_IO	Tag providing access to the I/O HUB's GPIO
IO_HUB_Encoder**	IO_HUB_Encoder	Tag providing access to the I/O HUB's External Encoder Input

**not available on the IO-HUB-2R*   ***only available on the IO-HUB-4E*

OUTPUTS

Operand	Type	Description
.ER (Error)	BOOL	There is a communication error between the PLC and the I/O HUB.

Motion Servo On (MSO_ClearPath)



Motion Servo On commands a servo to enable and verifies it has enabled successfully. Enabling the servo energizes the motor windings and allows the servo to respond to motion commands. A disabled servo is not under active control and will not respond to motion commands.

Use this instruction to enable your servo before commanding motion.

Use the Motion Servo Off (MSF_ClearPath) instruction to disable the servo.

Inputs

Operand	Type	Description
MSO_ClearPath	MSO_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.

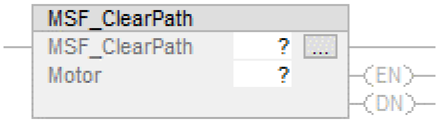
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The servo has enabled successfully.
.ER (Error)	BOOL	Another instruction commanded the motor to disable while MSO was waiting for the motor to enable.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, and the instruction logic begins executing.

Motion Servo Off (MSF_ClearPath)



Motion Servo Off commands the servo to disable and verifies the servo has disabled successfully. Disabling the servo de-energizes the motor windings. A disabled servo is not under active control and will not respond to motion commands.

By default, disabling the servo causes the motor to stop spinning by using the dynamic brake feature. This behavior can be changed to coast or decelerate to a stop (In ClearView, see Actions Upon Disable).

Use the Motion Servo On (MSO_ClearPath) instruction to enable the servo.

Safety Note: The Motion Servo Off (MSF) instruction is not designed for safety compliance. Disabling the motor does not remove power from the device.

Inputs

Operand	Type	Description
MSF_ClearPath	MSF_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.

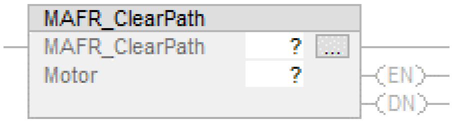
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The servo has disabled successfully.

Execution

Scan Mode	Description
Prescan	.EN, .DN are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN is true.
Rung-condition-in is TRUE	.EN is set to true, and the instruction logic begins executing.

MOTION AXIS SHUTDOWN RESET (MAFR_CLEARPATH)



Motion Axis Shutdown Reset clears shutdown conditions from a motor.

Bit 2 (Motor Shutdown Present) of the Motor Statusword will turn ON to indicate a shutdown. You can find out what caused the shutdown by using ClearView or by reading the Motor Shutdown Register.

INPUTS

Operand	Type	Description
MAFR_ClearPath	MAFR_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.

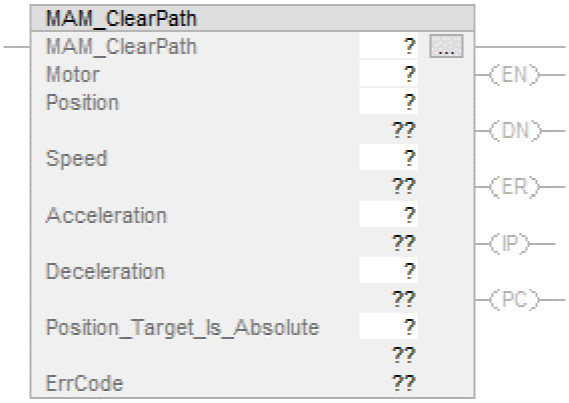
OUTPUTS

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	Shutdowns have been cleared successfully.

EXECUTION

Scan Mode	Description
Prescan	.EN, .DN are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN is true.
Rung-condition-in is TRUE	.EN is set to true, and the instruction logic begins executing.

MOTION AXIS MOVE (MAM_CLEARPATH)



Motion Axis Move commands the servo to move to a specified position. The positional move will execute on the servo each time the rung-condition-in transitions from FALSE to TRUE.

INPUTS

Operand	Type	Description
MAM_ClearPath	MAM_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.
Position	DINT	Absolute position to move to, or distance to move (depending on the Move Type selection below). **Units**: encoder counts (cnts)
Speed	DINT	Maximum velocity limit commanded during the move; must be positive. **Units:** cnts/s
Acceleration	DINT	Acceleration rate of the motor. Use only positive values. **Units:** cnts/s^2
Deceleration	DINT	Deceleration rate of the motor. Use only positive values. **Units:** cnts/s^2
Position_Target_ Is_Absolute	BOOL	**0 = Relative/Incremental Move** – Move a specified distance relative to the current commanded position. **1 = Absolute Move** – Move to a specified position relative to the home (zero) position of the motor.

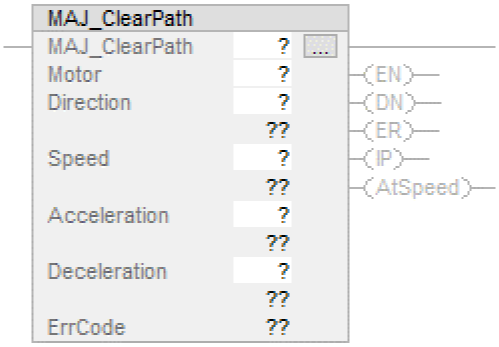
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The position command has been received successfully by the motor.
.ER (Error)	BOOL	An error occurred. Use the ErrCode output for more info.
.IP (In Process)	BOOL	The motor is actively moving toward the target position.
.PC (Process Complete)	BOOL	The motor reached the target position successfully.
ErrCode	INT	Refer to the Error Code Appendix to decode.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER, .IP, .PC are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .PC or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER, .IP, .PC are cleared to false. The instruction logic begins executing.

Motion Axis Jog (MAJ_ClearPath)



Motion Axis Jog commands the servo to rotate continuously at a specified velocity. To stop jogging, you can issue another MAJ to 0 velocity, or issue a Motion Axis Stop (MAS_ClearPath) instruction.

Caution: A jog command will continue indefinitely until it is explicitly stopped or interrupted by another move command.

Inputs

Operand	Type	Description
MAJ_ClearPath	MAJ_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.
Direction	BOOL	0 = Positive (+) / CCW 1 = Negative (-) / CW *CCW (counterclockwise) and CW (clockwise) are defined when you view the motor with its shaft pointing toward you.*
Speed	DINT	Speed to move the motor at; must be positive. **Units:** cnts/s
Acceleration	DINT	Acceleration rate of the motor. Use only positive values. **Units:** cnts/s^2
Deceleration	DINT	Deceleration rate of the motor. Use only positive values. **Units:** cnts/s^2

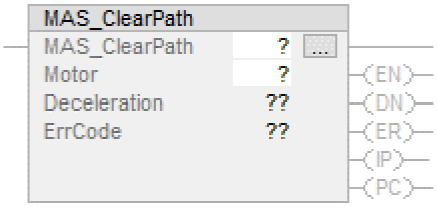
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The jog command has been received successfully by the motor.
.ER (Error)	BOOL	An error occurred. Use the ErrCode output for more info.
.IP (In Process)	BOOL	The motor is currently jogging.
ErrCode	INT	Refer to the Error Code Appendix to decode.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER, .IP are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER, .IP are cleared to false. The instruction logic begins executing.

Motion Axis Stop (MAS_ClearPath)



Motion Axis Stop commands the servo to stop all in-process motion using the specified deceleration rate. If the motor is slaved to another axis, executing MAS_ClearPath will decouple it from the master and bring it to a controlled stop. The motor remains energized during and after a Motion Axis Stop command.

Safety Note: The Motion Axis Stop (MAS) instruction is not designed for safety compliance.

Inputs

Operand	Type	Description
MAS_ClearPath	MAS_ClearPAth	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.
Deceleration	DINT	Deceleration rate of the motor. Use only positive values. A value of 0 causes the stop to use the motor's *E-Stop Deceleration Rate* set in ClearView 3.0. **Units:** cnts/s^2

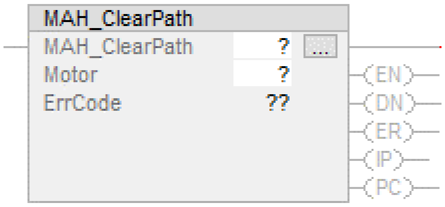
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The stop command has been received successfully by the motor.
.ER (Error)	BOOL	An error occurred. Use the ErrCode output for more info.
.IP (In Process)	BOOL	The motor is currently in the process of stopping.
.PC (Process Complete)	BOOL	The motor has stopped successfully.
ErrCode	INT	Refer to the Error Code Appendix to decode.

EXECUTION

Scan Mode	Description
Prescan	.EN, .DN, .ER, .IP are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER, .IP are cleared to false. The instruction logic begins executing.

Motion Axis Home (MAH_ClearPath)



Motion Axis Home commands the servo to execute its preconfigured homing routine. Use ClearView to configure homing.

Bit 9 (Has Homed) of the Motor Statusword can be used to determine if a motor has already been homed.

Inputs

Operand	Type	Description
MAH_ClearPath	MAH_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.

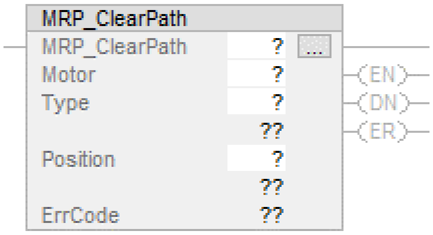
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The start homing command has been received successfully by the motor.
.ER (Error)	BOOL	An error occurred. Use the ErrCode output for more info.
.IP (In Process)	BOOL	The motor is currently in the process of homing.
.PC (Process Complete)	BOOL	The motor has successfully homed.
ErrCode	INT	Refer to the Error Code Appendix to decode.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER, .IP are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .PC or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER, .IP, .PC are cleared to false. The instruction logic begins executing.

MOTION REDEFINE POSITION (MRP_CLEARPATH)



MRP_ClearPath
MRP_ClearPath ?
Motor ?
Type ?
??
Position ?
??
ErrCode ??

The Motion Redefine Position AOI changes the active position reference, or "numberspace" of the servo. This allows you to reassign or offset the motor's current position value without physically moving.

This instruction is typically used to establish a new zero reference point or to make incremental adjustments to the current position reference during operation. It does not cause the motor to move—it only redefines how the controller interprets the current position.

INPUTS

Operand	Type	Description
MRP_ClearPath	MRP_ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Motor tag on which to perform this operation.
Type	BOOL	**Type 0 = Incremental** – Adds the Position value to the current numberspace. **Type 1 = Absolute** – Sets the current measured position equal to the Position value.
Position	DINT	

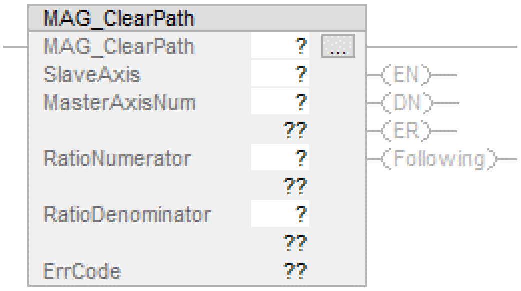
OUTPUTS

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The number space has successfully been changed.
.ER (Error)	BOOL	An error occurred. Use the ErrCode output for more info.
ErrCode	INT	Refer to the Error Code Appendix to decode.

EXECUTION

Scan Mode	Description
Prescan	.EN, .DN, .ER are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER are cleared to false. The instruction logic begins executing.

MOTION AXIS GEAR (MAG_CLEARPATH)



Motion Axis Gear configures a Slave Axis to be electronically geared to a Master Axis with a specified ratio. The ratio is defined in using a Ratio and Denominator.

For example, if the Slave Axis should move exactly half (1/2) the rotational distance and speed of the Master Axis, the Ratio Numerator should be set to 1 and the Ratio Denominator should be set to 2.

Use a negative value for Ratio Numerator to reverse the Slave Axis direction relative to the Master Axis.

Following/Gearing is unconfigured by commanding a SlaveAxis to stop using an MAS instruction.

INPUTS

Operand	Type	Description
MAG_ClearPath	MAG_ClearPath	Instance of this AOI
SlaveAxis	CPM_IP_Motor	Tag of the motor that will be geared to MasterAxis.
MasterAxisNum	SINT	Number corresponding to the motor or encoder to follow: 0 = Motor 0 (M-0) 1 = Motor 1 (M-1) 2 = Motor 2 (M-2) 3 = Motor 3 (M-3) 4 = External Encoder (Encoder Port on the I/O HUB)
Ratio Numerator	DINT	Used to specify the fractional gear ratio between the slave axis and the master axis in terms of encoder counts. *Gear Ratio = Ratio Numerator / Ratio Denominator*
Ratio Denominator	DINT	

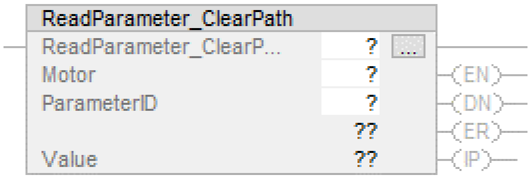
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The slave axis has successfully been geared to the master axis.
.ER (Error)	BOOL	An error occurred. Use the ErrCode output for more info.
Following	BOOL	The slave axis is currently following the master axis.
ErrCode	INT	Refer to the Error Code Appendix to decode.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER, .Following are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER, .Following are cleared to false. The instruction logic begins executing.

READ PARAMETER (READPARAMETER_CLEARPATH)



The Read Parameter instruction reads a parameter from the specified motor. One parameter can be read from each motor at any given time. If multiple instances of this instruction are executed at the same time only the last instruction will read successfully. See the list of parameters for more information.

Use this to get parameter information not otherwise available. For example, each motor's Measured Velocity is already available (from the assemblies via data exchange and motor tag), so the Read Parameter Instruction is not required to get this information.

READ PARAMETER LIST

ID (#)	Parameter Name	Units
30	Position Capture*	cnts
65	E-Stop Decel Rate	cnts/s
90	Tracking Error*	cnts
272	Tracking Error Limit	cnts
280	Motor Bus Voltage	Volts
291	Drive Temperature	°C
350	Torque Limit	0.1% of peak torque
351	Positive Torque Limit	0.1% of peak torque
353	Negative Torque Limit	0.1% of peak torque
783	RMS Percent	%

**Parameter not readable on ClearPath-IPVC*

INPUTS

Operand	Type	Description
ReadParameter_ ClearPath	ReadParameter_ ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Tag of the motor whose parameter will be read
ParameterID	INT	Which parameter will be read. (See table of Parameters)

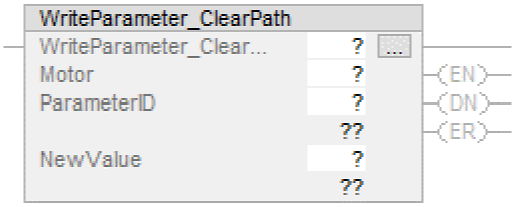
Outputs

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The parameter value has been updated
.ER (Error)	BOOL	An error occurred.
.IP (In Progress)	BOOL	The parameter value is continually being updated.
Value	DINT	The tag holding the value of the read parameter.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER, .IP are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER, .IP are cleared to false. The instruction logic begins executing.

WRITE PARAMETER (WRITEPARAMETER_CLEARPATH)



WriteParameter_ClearPath
WriteParameter_Clear... ? ...
Motor ?
ParameterID ?
??
NewValue ?
??

The Write Parameter instruction writes a new value to a parameter in the specified motor. One parameter can be written to each motor at any given time. If multiple instances of this instruction are executed at the same time only the last instruction will write successfully. See the list of parameters for more information.

WRITE PARAMETER LIST

ID (#)	Parameter Name	Units
65	E-Stop Decel Rate	cnts/s
272	Tracking Error Limit	cnts
350	Torque Limit	0.1% of peak torque
351	Positive Torque Limit	0.1% of peak torque
353	Negative Torque Limit	0.1% of peak torque

INPUTS

Operand	Type	Description
WriteParameter _ClearPath	WriteParameter _ClearPath	Instance of this AOI
Motor	CPM_IP_Motor	Tag of the motor whose parameter will be Written
ParameterID	INT	Which parameter will be read. (See table of Parameters)
NewValue	DINT	The value which is written to the parameter.

OUTPUTS

Operand	Type	Description
.EN (Enable)	BOOL	The instruction started execution.
.DN (Done)	BOOL	The parameter value has been updated
.ER (Error)	BOOL	An error occurred.

Execution

Scan Mode	Description
Prescan	.EN, .DN, .ER are cleared to false.
Rung-condition-in is FALSE	.EN is cleared to false if .DN or .ER are true.
Rung-condition-in is TRUE	.EN is set to true, .DN, .ER are cleared to false. The instruction logic begins executing.

Appendix A: AOI Error Codes

Many ClearPath-IP AOIs include an .ER boolean true/false output that indicates abnormal AOI operation. Some AOIs also include an ErrCode output that indicates one of several possible errors has occurred. Use the table below to decode any AOI ErrCode. For AOIs with only a boolean .ER output, refer to that specific AOI's documentation in the sections above.

Code	Description
0	**No error.**
100	**Invalid Move Type** – The requested move type is not recognized. Verify that the move type value matches one of the valid options listed in the Move Type table.
101	**Unsupported Move Type** – The selected move type is not supported by your motor model. Only certain move types are available for IPVC models.
102	**Motor in Shutdown** – A move command was issued while the motor is in a shutdown state. Clear all shutdowns before commanding motion.
103	**Motor Disabled** – A move command was issued while the motor is disabled. Enable the motor before commanding motion.
104	**Active Limit Switch** – The motor cannot move in the direction of an active limit switch. Command a move in the opposite direction.
105	**Software Limit Exceeded** – The commanded move would exceed a defined software limit. Adjust the target or move direction to remain within limits.
106	**Motion Stopped by Controlword** – The Controlword "Stop All Motion" bit is asserted, blocking new motion commands. Clear this bit to allow movement.
107	**Stop Sensor Active** – The stop sensor is triggered. Reset the stop sensor or clear the associated interlock before commanding motion.
108	**Invalid Motion Parameters** – One or more motion parameters are invalid. Acceleration must be non-zero, and position moves require a positive speed limit.
109	**Motor Unreachable** – The motor is not connected to the I/O HUB or is being controlled by ClearView. Verify network and device connections. Check ClearView access level; return to monitor access.
110	**Redefine Position Blocked** – A Motion Redefine Position command was issued while the motor is homing. Wait for homing to complete.
120	**Slave Axis Disabled** – The master axis cannot execute motion commands while any slave axis is disabled. Enable all slaves before proceeding.
121	**Invalid Gear Ratio** – The gear ratio contains a zero value. Both numerator and denominator must be non-zero.
122	**Invalid Master Axis** – The selected master axis does not exist or is not connected. Verify that the master corresponds to a valid motor or encoder.
123	**Move Not Allowed While Following** – This move type cannot be executed while the axis is configured as a slave (following another axis).
124	**Move Not Allowed While Being Followed** – This move type cannot be executed while the axis is configured as a master (followed by another axis).
125	**Already Following Different Axis** – The axis you are trying to follow is already configured to follow another axis. A motor cannot simultaneously be both a master and a slave. Stop following before reassigning.
126	**Already Being Followed** – The axis cannot start following a new axis because it is already being followed by another. A motor cannot simultaneously be both a master and a slave. Stop following before reassigning.
128	**Following Rejected due to Move in Progress** – Following cannot be configured if the master or slave are currently homing or if the slave was commanded to stop.

129	**Indeterminate Following Error –** Unspecified criteria to begin following has not been met. Contact Teknic for assistance.
254	**Move Canceled (Warning) –** The move was canceled due to a warning condition. The AOI parameter “MoveCanceledWarning” stores a snapshot of the motor’s warning register when the move was canceled. Refer to the Motor Warning Register appendix to interpret this code.
255	**Move Canceled (Shutdown) –** The move was canceled due to a motor shutdown. The AOI parameter “MoveCanceledShutdown” stores a snapshot of the motor’s shutdown register when the move was canceled. Refer to the Motor Shutdown Register appendix to interpret this code.

Appendix B: Motor Shutdown Register Breakdown

Motor Shutdown Register – Real-time info on what shutdowns/faults are present. Shutdowns are cleared using the Motion Axis Shutdown Reset AOI or via Controlword bit 1.

Motor Shutdown History – Stores a record of all past shutdown events in non-volatile memory until cleared via Controlword bit 2. The history can be read for each motor using the ReadParameter AOI: Parameter # = 1125 (M0), 1127 (M1), 1129 (M2), 1131 (M3).

Bit #	Description
0	**Unit Requires Repair** – Return to Teknic.
1-2	**Firmware Problem –** Unexpected error. Power cycle/clear shutdowns and contact Teknic if the error persists.
3	*Reserved*
4	**Startup Issue** – Motor failed startup sequence. Power cycle the motor and contact Teknic if the error persists.
5	**Config Load Required** – Load a config file compatible with motor's firmware version. If this fails: 1) reset the motor to factory defaults, 2) power cycle the motor, 3) load the desired config file again.
6	**Encoder Noise** – Fully power-cycle the motor. Contact Teknic if the shutdown does not clear.
7-8	**RAS Problem –** Unexpected error. Contact Teknic.
9	**Motor Phase Overload** – Phase current is beyond allowed ADC limit. Probable cause: incorrect tuning or wrong config file.
10	**RMS Torque Limit Exceeded** – Possible causes: excessive friction, mechanical misalignment, duty cycle too high, undersized motor.
11	**Tracking Error Limit Exceeded –** Possible causes: excessive friction, mechanical misalignment, vel/accel too high, low DC bus voltage.
12	**E-Stopped** – ESC Key / “E-Stop All” button was pressed in ClearView forcing a shutdown to occur.
13	*Reserved*
14	**Excessive Motor Temp** – Temperature exceeded specified limit (Additional Features > Power and Temperature). Possible causes: ambient temperature too high, low airflow, heat-sink obstructed, high load at high ambient temperature.
15	**Bus Voltage Lost / Too Low** – Probable causes: power supply disconnected (e.g. interlock open), brown out, power supply undersized.
16	**Max Bus Voltage Exceeded –** Probable cause: Large regenerated voltage upon deceleration. Possible remedy: enable the Vector Regen Shunt (VRS) feature using ClearView (under Additional Features).
17	**Excessive Bus Current** – Probable causes: bad tuning, low bus voltage, overloaded AC input/DC power supply.
18-20	*Reserved*
21	**Motor Reset Unexpectedly –** Probable causes: motor power cycled, motor control cables swapped.

22	**Motor Cable Error** – Errors detected in the connection between the motor and the I/O HUB. Probable causes: bad cable between motor and hub, electrical noise. Check cables and contact Teknic if the problem persists.
23-27	*Reserved*
28	**AC Wiring Error** – The number of AC phases detected does not match your configured AC Source Type (this setting can be found under Setup -> AC Source Type). Possible causes: the motor is wired incorrectly, an AC phase was lost, or the phase setting is wrong.
29	**AC Loss** – Possible causes: interlock open, bad wiring.
30-31	*Reserved*

Appendix C: Motor Warning Register Breakdown

Motor Warning Register – Real-time info on what non-critical warnings are present. Most warnings auto-clear when the condition is no longer present, and some warnings persist until a new command is sent.

Motor Warning History – Stores a record of all past warning events in non-volatile memory until cleared via Controlword bit 3. The history can be read for each motor using the ReadParameter AOI: Parameter # = 1124 (M0), 1126 (M1), 1128 (M2), 1130 (M3).

Bit #	Description
0	**Torque Saturation** - The motor has reached its torque limit while trying to follow the commanded motion. Probable cause: torque limit may be set too low, acceleration too high, power supply may be insufficient for application.
1	**Voltage Saturation** - Available torque is limited by DC bus voltage. Commanded speed may exceed motor spec at current power supply voltage. Try lowering speed or using a higher voltage power source (if possible).
2	**Over-Speed** - Commanded speed exceeds motor max speed limit. Lower speed to within motor spec.
3-4	*Reserved*
5	**High Temp** - Motor temperature approaching specified limit (In ClearView: Additional Features > Power and Temperature).
6	**High RMS Torque** - RMS torque is approaching the max limit.
7	**Bus Voltage Low** - Bus voltage went below user specified operating voltage (In ClearView: Additional Features > Power and Temperature).
8	*Reserved*
9	**Access Conflict** - ClearView has full access of either the motor, I/O HUB, or both. This means all commands will be rejected, including changing the state of I/O, enabling a motor, clearing faults, and commanding motion.
10	**I/O HUB Warning Present** – There is a warning present on the I/O HUB connected to this motor. Check ClearView for more information.
11	**Write Parameter Error** - Writing to parameter failed. Probable cause: invalid parameter ID.
12	**Read Parameter Error** - Read parameter failed. Probable cause: invalid parameter ID.
13	**Motor Link Errors Detected** - Errors are detected in the connection between the motor and the I/O HUB. Possible causes: bad cable, electrical noise.
14	**Motor Link Timeout** - Connection between the motor and the I/O HUB timed out. Probable cause: cable disconnected, motor lost power.
15	**Motor I/O Config Invalid** - Motor I/O incorrectly configured. Possible causes: limit switch/home sensor I/O point is an output instead of an input, brake output assigned to multiple motors.
16	**Move Canceled: (+) Limit Switch** - Positive limit switch triggered during motion or attempting to command further into the positive limit.
17	**Move Canceled: (-) Limit Switch** - Negative limit switch triggered during motion or attempting to command further into the negative limit.

18	**Move Canceled: (+) Soft Limit -** Positive software limit reached or position command outside limit.
19	**Move Canceled: (-) Soft Limit -** Negative software limit reached or position command outside limit.
20	**Move Canceled: Disabled or Shutdown -** Motor has been disabled or shutdown.
21	**Move Canceled: Stop Switch Active -** Stop switch input active.
22	**Move Canceled: Slave Axis Interrupted -** A slave axis had its motion interrupted (e.g. disabled, shutdown, reached limit).
23	**Move Canceled: Motor Disconnected -** Motor disconnected from the I/O HUB.
24	**Move Canceled: Stop All Motion -** Stop all motion Controlword bit turned on.
25	**Move Canceled: Communication Lost -** EtherNet/IP communication was lost. Possible causes: Ethernet cable disconnected or EtherNet/IP controller fault.
26-27	*Reserved*
28	**AC Wiring Error -** The number of AC phases detected does not match your configured AC Source Type (this setting can be found under Setup -> AC Source Type). Possible causes: the motor is wired incorrectly, an AC phase was lost, or the phase setting is wrong.
29	**AC Loss -** Possible causes: interlock open, bad wiring.
30-31	*Reserved*

Appendix D: Status Info & Feedback

Motor Statusword

Each ClearPath-IP motor has 32-bit value called the "Statusword" containing real-time status information. The table below defines each bit and its meaning.

Bit#	Name	1 (ON / HIGH)	0 (OFF / LOW)
0	Enabled	Motor windings energized.	Motor is disabled/de-energized.
1	Ready for Command	Servo is ready to receive motion commands.	Motor is disabled, shutdown, or startup sequence is in process.
2	Motor Shutdown Present	The motor is in a shutdown state. Refer to the Motor Shutdown Register to determine the cause.	No shutdowns present.
3	Motor Warning Present	The motor has non-critical warning(s) present. Refer to the Motor Warning Register for details.	No warnings present.
4	In Range	The motor's actual position is within the configured "In-Range Window" of its commanded position. *(Configured in ClearView)*	The motor's position error exceeds the In-Range Window.
5	Command Complete	The motor is enabled and there is no motion being commanded.	A motion command is executing or the motor is shutdown/disabled.
6	Settled	The motor is **In Range** and **Command Complete** for the "Settled Verify Time". *(Configured in ClearView)*	The motor is moving, or no longer within the In-Range Window.
7	At Speed	The motor is moving, and its actual velocity is within the configured "Velocity Window" of the target velocity. *(Configured in ClearView)*	The velocity error exceeds the Velocity Window, the motor is not moving, or the motor is homing.
8	Homing	Homing routine in progress.	The motor is not homing.
9	Has Homed	Homing has completed successfully.	Motor has not homed since power-up.
10	Brake Released	The I/O HUB output assigned to the external brake is active, releasing the brake. (*If no brake output is configured for this motor, this bit remains ON by default.)*	The external brake is de-energized and is holding the load.
11-14	Reserved	—	—
15	At Target Position	The motor is **Settled** at its target position. Indicates a position move has successfully completed. *Note: At Target Position only applies for positional moves.*	The motor is moving, or was prevented from reaching its target position.
16	Following Configured	This motor is configured to follow another axis.	Motor not configured for following.

17	Actively Following	This motor is currently following a master axis.	Motor is not following.
18	Is Followed	Other motor(s) are configured to follow this motor.	This motor has no followers.
19	Move Cancelled	The last move was cancelled. Refer to the Motor Warning Register to determine the cause.	—
20	Shutdown Reset Ack	Motor Shutdowns were cleared successfully. Used to handshake with Controlword bit “Shutdown Reset”.	—
21	In Home Sensor	Motor’s home sensor is active.	Motor’s home sensor is not configured or inactive.
22	All Motion Blocked	Motion on this axis is blocked because either the Stop Sensor is active or Controlword bit “Stop All Motion” is active.	—
23	In (+) Limit Switch	The motor’s positive limit input on the I/O HUB is active, or the Controlword bit “Ext Positive Limit” is ON.	—
24	In (-) Limit Switch	The motor’s negative limit input on the I/O HUB is active, or the Controlword bit “Ext Negative Limit” is ON.	—
25	In (+) Soft Limit	The motor’s position meets or exceeds the positive software position limit. *Note: Soft limits only apply after the motor has homed.*	Soft limit hasn’t been reached, motor has not been homed, or soft limits have not been configured.
26	In (-) Soft Limit	The motor’s position meets or exceeds the negative software position limit. *Note: Soft limits only apply after the motor has homed.*	Soft limit hasn’t been reached, motor has not been homed, or soft limits have not been configured.
27	Write Parameter Ack	A parameter write was successful. Used to handshake with Controlword bit “Write Parameter”.	—
28	Position Capture Sensor State	The motor’s position capture sensor is active.	Motor’s position capture sensor is not configured or inactive.
29	Motor Model Type	Motor model is either IPSK or IPHP.	Motor model is IPVC.
30	Reserved	—	—
31	Motor Connected	Motor is connected and communicating with the I/O HUB.	Motor is disconnected from the I/O HUB, or has communication problems.

Motor Feedback Parameters

These parameters provide real-time motion feedback and command information for each motor during operation.

Position Measured

The motor's real-time position as measured by its internal encoder.

Units: encoder counts (cnts)

Velocity Measured

The motor's real-time rotational speed based on encoder feedback.

Units: counts per second (cnts/s)

Torque Measured

The instantaneous output torque currently being produced by the motor.

Units: percent of motor's rated peak torque (%)

Position Target

The commanded position the motor is currently moving toward.

Velocity Target

The commanded velocity the motor is currently following.

Appendix E: Controlword & Move Commands

Motor Controlword

Each ClearPath-IP motor has 32-bit value called the “Controlword”, which is used to enable the servo and control other features. The table below defines each bit and its meaning.

Bit #	Name	Description
0	Enable	Turn ON to request the motor to enable. When enabled, the servo energizes its windings and produces torque to follow its commanded motion.
1	Shutdown Reset	Turn ON momentarily to clear all shutdowns and move cancelled warnings from the motor. When the shutdowns have been successfully cleared, the Shutdown Reset Ack bit (Statusword bit 20) turns ON to confirm the reset was accepted. After receiving this acknowledgment, turn Shutdown Reset back OFF to complete the handshake.
2	Clear Shutdown History	Clears the stored shutdown history. Used only for diagnostics or testing.
3	Clear Warning History	Clears the stored warning history. Used only for diagnostics or testing.
4	External Positive Limit	When ON, motion in the positive direction is stopped (acts like a positive limit switch). This bit is useful when a physical limit switch cannot be wired directly to the I/O HUB—for example, if the limit input is handled by the PLC instead.
5	External Negative Limit	When ON, motion in the negative direction is stopped (acts like a negative limit switch). This bit is useful when a physical limit switch cannot be wired directly to the I/O HUB—for example, if the limit input is handled by the PLC instead.
6	Write Parameter	Turn ON to perform a parameter write via the Generic Parameter Interface. The motor will set the Write Parameter Ack bit (Statusword bit 27) when the write operation has completed successfully. After receiving this acknowledgment, turn Write Parameter back OFF to complete the handshake and allow future writes. *For a reference implementation: refer to the WriteParameter_ClearPath AOI.*
7	Pause Parameter Reading	Turn this bit ON to stop cyclically reading a parameter via the Generic Parameter Interface.
8	Stop all Motion	When ON, overrides all motion commands and commands the axis to decelerate to a stop.
28-31	Reserved	—

Appendix F: I/O Information

Overview

ClearPath-IP provides flexible I/O capabilities for connecting sensors, switches, brakes, and other peripheral devices. All I/O configuration is performed in **ClearView**, where you define each point's function, input/output mode, and signal type. These settings are stored in the I/O HUB and are automatically applied at power-up—no PLC-side configuration is required.

The available I/O features depend on the I/O HUB model:

- On **Motor I/O** models (IO-HUB-2-R and IO-HUB-4-R), I/O points can be assigned to predefined motion-related functions. These operate automatically once configured in ClearView and are not accessed directly from the controller.
- The **Enhanced I/O HUB** (IO-HUB-4-E) adds general-purpose I/O (GPIO) that an EtherNet/IP Controller can directly monitor and control.

For IO-HUB-4-E systems, all GPIO data is organized in the IO_HUB_IO User-Defined Type (UDT).

- For **Allen-Bradley controllers using Studio 5000**, this UDT is created automatically when you import the ClearPath Add-On Instructions (AOIs).
- For **other EtherNet/IP controllers**, you should create an equivalent structure by implementing logic to extract the Assembly Data, similar to how the Data Exchange AOI operates. This ensures your controller organizes and accesses each I/O field in the same way.

IO_HUB_IO – UDT Breakdown

The **IO_HUB_IO** UDT represents all I/O data for the IO-HUB-4-E. The UDT provides dedicated data fields for each type so your controller can read or write values according to how each point is configured.

Member	Data Type	Description
DigitalInputs	INT	Bit-mapped digital inputs (I/O-0 through I/O-12). Each bit represents one input.
AnalogInputs	INT[13]	A 13-element array where each element corresponds to one analog input (I/O-0 through I/O-12). Each value is the measured voltage of that input in millivolts (mV). Points not configured as analog inputs will always report a value of zero.
DigitalOutputs	INT	Bit-mapped digital outputs (I/O-0 through I/O-11). Setting a bit to 1 energizes the corresponding output.

AnalogOut_IO12_uA	INT	Represents the commanded output current for I/O-12 when it is configured as an Analog Output in ClearView. The value is expressed in microamps (µA) and must be within the range 0-20,000 µA, corresponding to 0-20 mA of output current.
PWM_DutyCycles	INT[12]	A 12-element array where each element corresponds to one output point (I/O-0 through I/O-11). Each value sets the commanded PWM duty cycle for that point, where 0 = 0% and 255 = 100%. Only points configured for PWM operation in ClearView use this value; for all other output types, the array element is ignored.
IO_Config_ReadOnly	LINT	A read-only register that reports how each I/O point is configured in ClearView It is provided for diagnostic purposes and is not generally needed in control logic.

Using the IO-HUB-4-E Only as an I/O Expansion

If you plan to use the IO-HUB-4-E **only as an I/O expansion module** (not using any motion control functionality), the following information may help make your implementation straightforward.

For Allen-Bradley controllers using Studio 5000:

The best way to access I/O data is through the **IO_HUB_IO** UDT tag. To expose this data, use the IOHub4E_DataExchange AOI as described above.

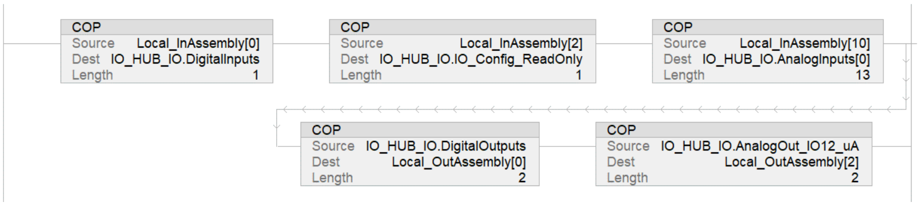
Important: The Data Exchange AOI still requires placeholder tags for the motors and external encoder. These tags are necessary for the AOI to compile, but they do not need to be used in your program.

For all other EtherNet/IP controllers:

I/O-related data must be extracted directly from the Assemblies. Depending on your application, you can parse this data into a User-Defined Type (UDT) to keep I/O fields organized, or copy the values directly into controller tags for use in your logic.

The byte ranges for all I/O related Assembly data are defined in the IO-HUB-4-E section of *Appendix H: EtherNet/IP Assemblies*. Look for the sections I/O Input Data and I/O Output Data.

A ladder diagram example showing how to copy this data from the Input and Output Assemblies is provided below.



COP
Source Local_InAssembly[0]
Dest IO_HUB_IO.DigitalInputs
Length 1

COP
Source Local_InAssembly[2]
Dest IO_HUB_IO.IO_Config_ReadOnly
Length 1

COP
Source Local_InAssembly[10]
Dest IO_HUB_IO.AnalogInputs[0]
Length 13

COP
Source IO_HUB_IO.DigitalOutputs
Dest Local_OutAssembly[0]
Length 2

COP
Source IO_HUB_IO.AnalogOut_IO12_uA
Dest Local_OutAssembly[2]
Length 2

Appendix G: Glossary of Common Terms

Electronic Data Sheet (EDS)

An EDS file provides the controller with information about a connected device and its communication structure.

ClearPath-IP includes different EDS files for each I/O HUB model:

- IO-HUB-4-E
- IO-HUB-4-R
- IO-HUB-2-R

Load the appropriate EDS file into your PLC/Controller as part of the device setup process.

Download: Included with the ClearPath-IP AOIs and Example Files here.

Implicit vs Explicit Communication

Implicit messaging is the preferred communication method for motion control. It provides high-speed, real-time updates between the controller and devices.

Explicit messaging is typically used for infrequent or configuration-related tasks. It requires more setup and overhead, making it less efficient for real-time control compared to implicit messaging.

Requested Packet Interval (RPI)

The Requested Packet Interval (RPI) defines how often data is exchanged between ClearPath-IP and the EtherNet/IP controller when an implicit connection is active.

ClearPath-IP supports RPI values of **1 to 1000 milliseconds (ms)**.

Assembly (Input / Output)

Assemblies are predefined groups of data used for cyclic communication between the EtherNet/IP controller and the I/O HUB.

The Input Assembly (often referred to as T→O, "*Target to Originator*") contains all status and feedback information sent from the I/O HUB to the controller—such as motor position, velocity, and I/O states.

The Output Assembly (often referred to as O→T, "*Originator to Target*") carries command data sent from the controller to the I/O HUB—such as motion commands, enable signals, and parameter writes.

These assemblies form the continuous data stream that allows the controller and I/O HUB to stay synchronized for real-time motion control.

Data Exchange

The Data Exchange instruction acts as the communication bridge between the PLC logic and the I/O HUB. It organizes the incoming and outgoing EtherNet/IP assembly data into structured UDTs that represent each motor and I/O point, ensuring all ClearPath-IP instructions have synchronized, readable, and writable information to operate correctly.

Add-On Instruction (AOI)

An Add-On Instruction (AOI) is a reusable block of logic in Studio 5000 that performs a specific function—similar to a subroutine or function block. AOIs allow complex operations, such as motion control or communication handling, to be encapsulated into a single instruction with defined inputs, outputs, and internal logic.

User-Defined Type (UDT)

A User-Defined Type (UDT) is a custom data structure in Studio 5000 that groups related variables—such as positions, velocities, and status bits—into a single, organized tag. UDTs simplify program design by allowing complex data to be passed between instructions as one object rather than as individual elements.

Counts (cnts)

Counts (cnts) are the native position units of a ClearPath-IP motor, representing individual increments of its internal encoder. All position, velocity, and acceleration values are expressed in counts, counts per second, and counts per second^2, respectively.

APPENDIX H: ETHERNET/IP ASSEMBLIES

OVERVIEW

The Output Assembly is sent from the master PLC to the I/O HUB to control all motion axes and output states. The Input Assembly is sent from the I/O HUB to the master PLC to convey motion status and I/O states. Most users won't need to interact with these assemblies directly because the Data Exchange logic abstracts this layer and provides tags to be used by other functions. Refer to this Appendix when recreating your own Data Exchange in unsupported programming environments.

ASSEMBLY OBJECT (04HEX)

I/O HUB	Instance	Attribute ID	Name	Size
IO-HUB-4-E	100 (0x64)	3	T2O Input Assembly	SINT[228]
	101 (0x65)	3	O2T Output Assembly	SINT[148]
IO-HUB-4-R	104 (0x68)	3	T2O Input Assembly	SINT[176]
	105 (0x69)	3	O2T Output Assembly	SINT[128]
IO-HUB-2-R	106 (0x6A)	3	T2O Input Assembly	SINT[88]
	107 (0x6B)	3	O2T Output Assembly	SINT[64]

IO-HUB-4-E

IO-HUB-4-E T2O INPUT Assembly (Instance 100 (0x64))			
	Name	Length (Bytes)	Start Byte
I/O Input Data	Digital Inputs	2	0
	I/O Configuration	7	2
	Reserved Padding	1	9
	Analog Inputs (13 inputs, 2 bytes each)	26	10
Motor 0 Input Data	Statusword	4	36
	Shutdown Register	4	40
	Warning Register	4	44
	Position Measured	4	48
	Velocity Measured	4	52
	Torque Measured	2	56
	Position Target	4	58

	Velocity Target	4	62
	Reserved Padding	2	66
	Read Parameter ID Echo	2	68
	Read Parameter Value	4	70
	Move Type Ack	1	74
	Move Number Ack	2	75
	Reserved Padding	3	77
Motor 1 Input Data	Statusword	4	80
	Shutdown Register	4	84
	Warning Register	4	88
	Position Measured	4	92
	Velocity Measured	4	96
	Torque Measured	2	100
	Position Target	4	102
	Velocity Target	4	106
	Reserved Padding	2	110
	Read Parameter ID Echo	2	112
	Read Parameter Value	4	114
	Move Type Ack	1	118
	Move Number Ack	2	119
	Reserved Padding	3	121
Motor 2 Input Data	Statusword	4	124
	Shutdown Register	4	128
	Warning Register	4	132
	Position Measured	4	136
	Velocity Measured	4	140
	Torque Measured	2	144
	Position Target	4	146
	Velocity Target	4	150
	Reserved Padding	2	154
	Read Parameter ID Echo	2	156
	Read Parameter Value	4	158
	Move Type Ack	1	162
	Move Number Ack	2	163
	Reserved Padding	3	165
Motor 3 Input Data	Statusword	4	168
	Shutdown Register	4	172
	Warning Register	4	176
	Position Measured	4	180
	Velocity Measured	4	184
	Torque Measured	2	188
	Position Target	4	190
	Velocity Target	4	194

	Reserved Padding	2	198
	Read Parameter ID Echo	2	200
	Read Parameter Value	4	202
	Move Type Ack	1	206
	Move Number Ack	2	207
	Reserved Padding	3	209
Encoder Input Data	External Encoder Position	4	212
	External Encoder Velocity	4	216
	External Encoder Index Position	4	220
	External Encoder Alarm Flag	1	224
	External Encoder Add to Position Ack	1	225
	Reserved Padding	2	226
SIZE IN BYTES = 228			

IO-HUB-4-E O2T OUTPUT Assembly (Instance 101 (0x65))

	Name	**Length (Bytes)**	**Start Byte**
I/O Output Data	Digital Outputs	2	0
	Analog Output I/O-12 µA	2	2
	PWM I/O-0 Duty Cycle	1	4
	PWM I/O-1 Duty Cycle	1	5
	PWM I/O-2 Duty Cycle	1	6
	PWM I/O-3 Duty Cycle	1	7
	PWM I/O-4 Duty Cycle	1	8
	PWM I/O-5 Duty Cycle	1	9
	PWM I/O-6 Duty Cycle	1	10
	PWM I/O-7 Duty Cycle	1	11
	PWM I/O-8 Duty Cycle	1	12
	PWM I/O-9 Duty Cycle	1	13
	PWM I/O-10 Duty Cycle	1	14
	PWM I/O-11 Duty Cycle	1	15
Motor 0 Output Data	Controlword	4	16
	Motion Parameters	16	20
	Move Type	1	36
	Move Number	2	37
	Read Parameter ID	2	39
	Write Parameter ID	2	41
	Write Parameter Value	4	43
	Reserved Padding	1	47
Motor 1 Output Data	Controlword	4	48
	Motion Parameters	16	52
	Move Type	1	68
	Move Number	2	69

	Read Parameter ID	2	71
	Write Parameter ID	2	73
	Write Parameter Value	4	75
	Reserved Padding	1	79
Motor 2 Output Data	Controlword	4	80
	Motion Parameters	16	84
	Move Type	1	100
	Move Number	2	101
	Read Parameter ID	2	103
	Write Parameter ID	2	105
	Write Parameter Value	4	107
	Reserved Padding	1	111
Motor 3 Output Data	Controlword	4	112
	Motion Parameters	16	116
	Move Type	1	132
	Move Number	2	133
	Read Parameter ID	2	135
	Write Parameter ID	2	137
	Write Parameter Value	4	139
	Reserved Padding	1	143
Encoder Output Data	Encoder Add To Position	4	144
SIZE IN BYTES = 148			

IO-HUB-4-R

IO-HUB-4-R T2O INPUT Assembly (Instance 104 (0x68))			
	Name	**Length (Bytes)**	**Start Byte**
Motor 0 Input Data	Statusword	4	0
	Shutdown Register	4	4
	Warning Register	4	8
	Position Measured	4	12
	Velocity Measured	4	16
	Torque Measured	2	20
	Position Target	4	22
	Velocity Target	4	26
	Reserved Padding	2	30
	Read Parameter ID Echo	2	32
	Read Parameter Value	4	34
	Move Type Ack	1	38
	Move Number Ack	2	39
	Reserved Padding	3	41

Motor 1 Input Data	Statusword	4	44
	Shutdown Register	4	48
	Warning Register	4	52
	Position Measured	4	56
	Velocity Measured	4	60
	Torque Measured	2	64
	Position Target	4	66
	Velocity Target	4	70
	Reserved Padding	2	74
	Read Parameter ID Echo	2	76
	Read Parameter Value	4	78
	Move Type Ack	1	82
	Move Number Ack	2	83
	Reserved Padding	3	85
Motor 2 Input Data	Statusword	4	88
	Shutdown Register	4	92
	Warning Register	4	96
	Position Measured	4	100
	Velocity Measured	4	104
	Torque Measured	2	108
	Position Target	4	110
	Velocity Target	4	114
	Reserved Padding	2	118
	Read Parameter ID Echo	2	120
	Read Parameter Value	4	122
	Move Type Ack	1	126
	Move Number Ack	2	127
	Reserved Padding	3	129
Motor 3 Input Data	Statusword	4	132
	Shutdown Register	4	136
	Warning Register	4	140
	Position Measured	4	144
	Velocity Measured	4	148
	Torque Measured	2	152
	Position Target	4	154
	Velocity Target	4	158
	Reserved Padding	2	162
	Read Parameter ID Echo	2	164
	Read Parameter Value	4	166
	Move Type Ack	1	170
	Move Number Ack	2	171
	Reserved Padding	3	173
SIZE IN BYTES = 176			

IO-HUB-4-R O2T OUTPUT Assembly (Instance 105 (0x69))			
	Name	**Length (Bytes)**	**Start Byte**
Motor 0 Output Data	Controlword	4	0
	Motion Parameters	16	4
	Move Type	1	20
	Move Number	2	21
	Read Parameter ID	2	23
	Write Parameter ID	2	25
	Write Parameter Value	4	27
	Reserved Padding	1	31
Motor 1 Output Data	Controlword	4	32
	Motion Parameters	16	36
	Move Type	1	52
	Move Number	2	53
	Read Parameter ID	2	55
	Write Parameter ID	2	57
	Write Parameter Value	4	59
	Reserved Padding	1	63
Motor 2 Output Data	Controlword	4	64
	Motion Parameters	16	68
	Move Type	1	84
	Move Number	2	85
	Read Parameter ID	2	87
	Write Parameter ID	2	89
	Write Parameter Value	4	91
	Reserved Padding	1	95
Motor 3 Output Data	Controlword	4	96
	Motion Parameters	16	100
	Move Type	1	116
	Move Number	2	117
	Read Parameter ID	2	119
	Write Parameter ID	2	121
	Write Parameter Value	4	123
	Reserved Padding	1	127
SIZE IN BYTES = 128			

IO-HUB-2-R

IO-HUB-2-R T2O INPUT Assembly (Instance 106 (0x6A))			
	Name	**Length (Bytes)**	**Start Byte**
Motor 0 Input Data	Statusword	4	0
	Shutdown Register	4	4
	Warning Register	4	8
	Position Measured	4	12
	Velocity Measured	4	16
	Torque Measured	2	20
	Position Target	4	22
	Velocity Target	4	26
	Reserved Padding	2	30
	Read Parameter ID Echo	2	32
	Read Parameter Value	4	34
	Move Type Ack	1	38
	Move Number Ack	2	39
	Reserved Padding	3	41
Motor 1 Input Data	Statusword	4	44
	Shutdown Register	4	48
	Warning Register	4	52
	Position Measured	4	56
	Velocity Measured	4	60
	Torque Measured	2	64
	Position Target	4	66
	Velocity Target	4	70
	Reserved Padding	2	74
	Read Parameter ID Echo	2	76
	Read Parameter Value	4	78
	Move Type Ack	1	82
	Move Number Ack	2	83
	Reserved Padding	3	85
SIZE IN BYTES = 88			

IO-HUB-2-R O2T OUTPUT Assembly (Instance 107 (0x6B))			
	Name	**Length (Bytes)**	**Start Byte**
Motor 0 Output Data	Controlword	4	0
	Motion Parameters	16	4
	Move Type	1	20
	Move Number	2	21
	Read Parameter ID	2	23
	Write Parameter ID	2	25
	Write Parameter Value	4	27
	Reserved Padding	1	31
Motor 1 Output Data	Controlword	4	32
	Motion Parameters	16	36
	Move Type	1	52
	Move Number	2	53
	Read Parameter ID	2	55
	Write Parameter ID	2	57
	Write Parameter Value	4	59
	Reserved Padding	1	63
SIZE IN BYTES = 64			

APPENDIX I: ETHERNET/IP OBJECT MODEL (SIMPLIFIED)

IDENTITY OBJECT (01HEX - 1 INSTANCE)

Instance	Attribute ID	Name	CIP Data Type	EEPROM	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	Y	1.1	Get
Instance 1	1	Vendor number	UINT	Y	424	Get
	2	Device type	UINT	Y	43	Get
	3	Product code number	UINT	Y	10 = IO-HUB-4-E 11 = IO-HUB-4-R 12 = IO-HUB-2-R	Get
	4	Product major revision	USINT	Y	1.1	Get
		Product minor revision	USINT	Y		
	5	Status	WORD	N	0	Get
	6	Serial number	UDINT	Y	Unique 32 bit value	Get
	7	Product name	SHORT STRING32	Y	Varies	Get

TCP OBJECT (F5HEX - 1 INSTANCE)

Instance	Attribute ID	Name	CIP Data Type	EEPROM	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	Y	4	Get
Instance 1	1	Status*	UDINT	Y	Varies	Get
	2	Configuration capability*	DWORD	Y	0x54	Get
	3	Configuration control*	DWORD	Y	Varies	Get
	4	Physical Link Object *		Y		Get
		Structure of		Y		
		Path Size	UINT	Y	Varies	

		Path	Array of Word	Y		
	5	Interface configuration*		Y		Get / Set
		Structure of		Y		
		IP Address	UDINT	Y	Varies	
		Network Mask	UDINT	Y	Varies	
		Gateway Address	UDINT	Y	Varies	
		Name Server	UDINT	Y	Varies	
		Name Server 2	UDINT	Y	Varies	
		Domain Name Size	UINT	Y	Varies	
		Domain Name	STRING	Y	Varies	
	6	Host name*		Y		Get / Set
		Structure of		Y		
		Host Name Size	UINT	Y	Varies	
		Host Name	STRING	Y	Varies	
	13	Encapsulation Inactivity Timeout	UINT	Y	Varies	Get / Set (NVRAM)

Ethernet Link Object (F6HEX - 2 Instances)

Instance	Attribute ID	Name	CIP Data Type	EEPROM	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	Y	4	Get
Instance 1 - 2	1	Interface Speed*	UDINT	Y	Varies	Get
	2	Interface Flags*	DWORD	Y	Varies	Get
	3	Physical Address	USINT Array (6)	Y	Varies	Get
	10	Interface Label	STRING	Y	Port 1/Port 2	Get
	11	Interface Capabilities	STRUCT	Y	Varies	Get

Board Object (69HEX -1 Instance)

Instance	Attribute ID	Name	Data Type	EEPROM	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	Y	2	Get
	2	Max Instance	UINT	Y	1	Get
	3	Number of Instances	UINT	Y	1	Get
Instance 1	1	IO Hub Warnings	UINT	N	Varies	Get
	2	Reset Hub Warnings	BOOL	N	Varies	Get/Set
	3	Aux Voltage level	UINT	N	Varies	Get
	4	Supply Voltage	UINT	N	Varies	Get
	5	Motor Link Timeout	USINT	Y	Varies	Get / Set
	6	Save To Hub NVM	UDINT	N	0	Get / Set
	7	Overloaded Outputs	UINT	N	0	Get
	8	Analog Overvoltage	UINT	N	0	Get
	9	Hub Filename	USINT[25]	Y	Varies	Get/ Set
	10	USER ID	USINT[25]	Y	Varies	Get/ Set
	11	Hub Part Number	USINT[25]	Y	Varies	Get
	12	Reserved	SINT	N	Varies	Get/ Set