MagneMover LITE Ethernet Motor Configuration and Communication

Document: 10003890416; MMI-UM031E-EN-P
Version: 05

MagneMotion, Inc.
A Rockwell Automation Company
139 Barnum Road
Devens, MA 01434
USA

Phone: +1 978-757-9100
Fax: +1 978-757-9200

www.magnemotion.com

Contents

Figures

Tables

About This Document

This document is an addendum to the *Node Controller Interface User Manual* and describes the Ethernet motor communication option. Additional information describes the network topologies for wiring MagneMover® LITE Ethernet motors and for combining both RS-422 and Ethernet motors in the same transport system.

Purpose

The existing MM LITE motor uses a system architecture design where motors communicate with each other and the node controllers via RS-422 communication. The new MM LITE Ethernet motors use a new system architecture design where motors communicate with each other and the node controllers via TCP/IP over Ethernet. The differences in the two communication methods result in changes to the configuration of the transport system.

This document describes the method for creating the MagneMotion Information and Configuration Service (MICS) file for configuring a MagneMover LITE transport system with motors that use Ethernet communication. The MICS file defines the Ethernet track topology by mapping the motor MAC address and IP address to the track layout. This mapping provides a correlation from a logical track location to a motor MAC address and IP address, which is designed to simplify wiring.

Audience

This document is intended for all users of MagneMover LITE transport systems and provides information on how to install and configure MM LITE transport systems that use Ethernet connected motors.

Reference Documents

- 990000377, Node Controller Interface User Manual
- 990000410, MagneMover LITE User Manual
- 990000558, MagneMover LITE Configurator User Manual
- 10003865989, MagneMover LITE Manual, Ethernet Motors Addendum
- 1783-UM007J-EN-P, Stratix Managed Switches User Manual
- *XML Pocket Reference* - St. Laurent and Fitzgerald: O'Reilly, 2005

Definitions, Acronyms, and Abbreviations

Table 1 provides common terminology for use with XML.

Table 1: XML Terminology

Term	Definition
Attribute	An XML structural construct. A name-value pair, which is separated by an equals sign, included inside a tagged element that modifies certain features of the element. All attribute values, including things like size and width, are in fact text strings and not numbers. For XML, all values must be enclosed in quotation marks.
Document Element	The element in an XML document that contains all other elements. It is the top-level element of an XML document and must be the first element in the document. There is exactly one document element, no part of which appears in the content of any other element. The document element represents the document as a whole; every other element represents a component of the document.
Document Type Declaration	An XML structural construct. Consists of markup code that indicates the grammar rules, or Document Type Definition (DTD), for the particular class of document. The document type declaration can also point to an external file that contains all or part of the DTD. It must be included following the XML declaration and before the document element. The syntax of the document type declaration is <!DOCTYPE content>.
Element	An XML structural construct. An XML element consists of a start tag, an end tag, and the information between the tags, which is often referred to as the contents. Each element can have a set of attribute specifications. Each attribute specification has a name and a value. An instance of an element is declared using <element> </element> tags.
IP	Internet Protocol
MAC	Media Access Control (IEEE 802.3 Ethernet hardware address)
MICS	MagneMotion Information and Configuration Service
XML	Extensible Markup Language

Overview

The MICS file is used to define the Ethernet track topology for the motors in a transport system. The file also describes the interfaces and physical hardware connections within a MagneMover LITE transport system.

The MICS file works with the Node Controller Configuration File, which specifies the configuration of components within the transport system, such as; node controllers, paths, nodes, and motors. Each path that is defined in the Node Controller Configuration File defines the specific motors and their relationships to the motors on a path. The MICS file then defines the MAC address and IP address for each of those motors.

The MICS file contains the following:

- MAC address for each motor.
- IP address for each motor.
- Transport system location for each motor.
- Physical orientation of the motors.

The MICS files are written in XML format. XML allows the format to be backward and forward compatible and easily extended. A newer version of software can easily ignore older unused XML tags. An older version of software can ignore newer, unknown XML tags. The XML file format is human readable, which allows manual editing. XML files can be viewed with browsers or code editors in tree fashion (with the ability to expand or contract elements that contain other elements).

An overview of the XML format can be found in the *XML Pocket Reference*. Basically, an XML file contains a hierarchical set of elements. Each element consists of an opening tag, <Tag_name>, and a closing tag, </Tag_name>, with either data or other elements in between. In general, each tag starts with <Tag_name>, and ends with </Tag_name>. In this implementation, an element contains configuration data or other elements, but not both.

When a motor powers up, its network and topology information (from the MICS file) is provided to it dynamically by the node controller that is responsible for its path. The topology information the motor receives includes its IP address, subnet mask, default gateway, and information about its neighboring motor connections.

The MagneMotion transport system accepts any valid IPv4 address scheme to operate. The address must be on the same subnet as its node controller. It is important to keep unnecessary broadcast traffic off the transport system network as it can impact overall system performance.

For a large transport system, it is typically useful to organize the IP structure that includes the path/motor information in as shown in *Ethernet Motor Communication Recommendations* on page 24.

MICS File Format

The MICS file consists of XML elements, which are identified by their tag names, and comments. The file format consists of a declaration and a root element. The root element contains other elements. Other elements include either data or more elements. Comments can be interspersed with elements or data (but must not be within the tag).

File Naming Convention

The file can have any name for convenient reference, but must have the .xml extension. Once the file is uploaded to the node controller, it is automatically renamed to 'MICS_motor_data.xml'.

As an example:

MICS_Development_System.xml

Declaration

The XML declaration is placed on the first line of the file. It specifies the XML version and the character encoding of the document. In this case, the declaration line is:

```
<?xml version="1.0" encoding="US-ASCII"?>
```

File Identification

MICS files are identified in comment lines following the XML declaration. The revision and last change data reflect the file revision and the data and time of the last change of the file.

As an example:

```
<!-- MICS Motor Data file for a MagneMotion Node Controller -->
<!-- $Rev: 242 $ $LastChangedDate: 2018-02-15 20:06:34 (Thu, 15 Feb 2018) $ -->
```

Element

There is *only* one root element in a MICS file. In this case, the root element tag is *MICS_motor_data*. Elements can contain other elements or configuration data. Elements must be properly nested.

Tag Names

Tag names identify the function of the tag. The convention for tag names is *only* the first letter is capitalized and underscores ('_') are used in place of spaces.

Comments

Comments can be included where desired, including inside elements. Comments are identified by enclosing them in angle brackets with the first characters after the opening bracket an exclamation point and two dashes, and two dashes before the closing bracket. As an example:

```
<!-- comment -->
```

MICS File Protocol

The MICS file protocol defines the XML elements and structure that is used to identify the motors, their MAC address, IP address, and location on a path. These elements are only used in the MICS file.

MICS File XML Reference

The XML tags that are shown in Table 2 are used to define and configure the MICS file.

Table 2: MICS File XML Elements

Element	Description	Page
XML Declaration	Start of the document.	12
MICS_motor_data	Root tag for MICS motor configuration data.	13
Motor	Container tag that defines the MICS data for a motor.	14
Mac_addr	The MAC address of the master controller for the motor.	15
IP_addr	The static IPv4 address of the master controller for the motor.	16
Track_location	The motor position in the specified path.	17
Orientation	The orientation of the motor in the specified path.	19

XML Declaration

Purpose

The XML declaration is the first line in the document. It contains the XML version number and the character encoding for the file.

This element is a declaration, there is no closing tag.

Syntax

```
<?xml version="1.0" encoding="US-ASCII"?>
```

Attributes

None

Parent Tag

None

Contents

None

Example

The following example defines the version of XML used as 1.0 and the document encoding as US ASCII (see Figure 3 on page 21).

```
<?xml version="1.0" encoding="US-ASCII"?>
```


MICS_motor_data

Purpose

The root tag for MICS XML files, it contains all topology parameters for an Ethernet motor-based transport system. This tag contains no data; it only contains other tags.

Syntax

```
<MICS_motor_data>
.
.
.
</MICS_motor_data>
```

Attributes

None

Parent Tag

None (root tag).

Contents

<Motor>

Validation

Node controller requirements to accept XML upload:

- Must be the root tag of the XML file.

Example

The following example defines the enclosed tags as MICS motor data (see Figure 3 on page 21).

```
<MICS_motor_data>
.
.
.
</MICS_motor_data>
```


Motor

Purpose

Tags for MICS data for a motor. This tag contains no data; it delineates information for one motor. It contains only other tags.

Syntax

```
<Motor>
.
.
.
</Motor>
```

Attributes

None

Parent Tag

<MICS_motor_data>

Contents

All children tags.

<Mac_addr>
<IP_addr>
<Track_location>
<Orientation>

Validation

Node controller requirements for startup and initialization:

- Must contain at least one child tag.

Example

The following example defines the enclosed tags as the definition for a motor (see Figure 3 on page 21).

```
<MICS_motor_data>
<!-- PATH 1 -->
  <Motor> <!-- P1M1 -->
  .
  .
  </Motor>
```

Mac_addr

Purpose

Identifies the unique MAC address of the master controller for the motor, its network, and topology configuration, which includes the elements within this tag.

Syntax

```
<Mac_addr>MACaddr</Mac_addr>
```

Attributes

None

Parent Tag

<Motor>

Contents

MACaddr – Unique MAC address in standard IEEE-802 format (that is, six groups of two hex digits), a colon ‘:’ must be used to separate the digits (ex. hh:hh:hh:hh:hh:hh). The first three hexadecimal groups are the Organizationally Unique Identifier (OUI). The OUI for MagneMotion is c0:6c:6d. The MAC address values are set when the motor is built and the value within this tag must match the specific motor that this tag is referring to.

Validation

Node controller requirements for startup and initialization:

- Must not contain any child tags.
- Must contain a string in the hh:hh:hh:hh:hh:hh format.
- Must not be the value 00:00:00:00:00:00.
- Must not contain a value that is equal to another <Mac_addr> tag.

Example

The following example defines the MAC ID for the first motor on path 1 (see Figure 3 on page 21).

```
<MICS_motor_data>
<!-- PATH 1 -->
  <Motor> <!-- P1M1 -->
    <Mac_addr>C0:6C:6D:E0:00:20</Mac_addr>
    .
    .
```

IP_addr

Purpose

Identifies the static IPv4 address of the master controller for the motor.

Syntax

```
<IP_addr>IPaddress</IP_addr>
```

Attributes

None

Parent Tag

<Motor>

Contents

IPaddress – A 32-bit IP address in IPv4 dotted-decimal format, "ddd.ddd.ddd.ddd", where ddd is a decimal number of up to three digits in the range 0 to and 255. See *Ethernet Motor Communication Recommendations* on page 24.

Validation

Node controller requirements for startup and initialization:

- Must not contain any child tags.
- Must contain a string in IPv4 dotted-decimal notation.
- Must not be values of 0.0.0.0, 127.0.0.1, 255.255.255.255, or the node controller Subnet broadcast address.
- Must not contain a value that is equal to another <IP_addr> tag.

Example

The following example defines the IP address for the first motor on path 1 (see Figure 3 on page 21).

```
<MICS_motor_data>
<!-- PATH 1 -->
  <Motor> <!-- P1M1 -->
    .
    .
    <IP_addr>10.14.1.1</IP_addr>
```

Track_location

Purpose

Identifies the motor position in the specified path. The Node Controller Configuration File defines the paths where the motors are located.

- Straight and curve motors are represented as one <Motor> tag with one <Track_location> tag.
- Switches are represented as one <Motor> tag with two <Track_location> tags.

Syntax

```
<Track_location>Location</Track_location>
```

Attributes

None

Parent Tag

<Motor>

Contents

Location – The location is identified as P*n*M*n*. Where P*n* is the path number and M*n* is the location of the motor in the path.

Validation

Node controller requirements for startup and initialization:

- Must not contain any child tags.
- Must not contain a value that is equal to another <Track_location> tag.

Examples

The following example defines a motor that is on path 3 (see Figure 3 on page 21).

```
<MICS_motor_data>
<!-- PATH 3 -->
  <Motor> <!—P3M2 -->
    .
    .
    <Track_location>P3M2</Track_location>
```

The following example defines a switch that is at the beginning of path 2 and path 3 (see Figure 3 on page 21). The Node Controller Configuration File defines the type of switch (diverge) and the paths where the switch is located.

```
<!-- PATH 2 -->
<MICS_motor_data>
<!-- PATH 2 -->
  <Motor> <!—P2M1 P3M1)->
    .
    .
    <Track_location>P2M1</Track_location>
    <Track_location>P3M1</Track_location>
```

The following example defines a switch that is at the end path 2 and path 3 (see Figure 3 on page 21). The Node Controller Configuration File defines the type of switch (merge) and the paths where the switch is located.

```
<MICS_motor_data>
<!-- PATH 2 -->
  <Motor> <!—P2M3 P3M5 -->
    .
    .
    <Track_location>P2M3</Track_location>
    <Track_location>P3M5</Track_location>
```

Orientation

Purpose

Identifies the orientation of the motor in the specified path. For MagneMover LITE motors, the location of the power connector identifies the default upstream end of the motor. Standard orientation is when a motor location on a path is such that the upstream end of the motor is located towards the upstream end of the path.

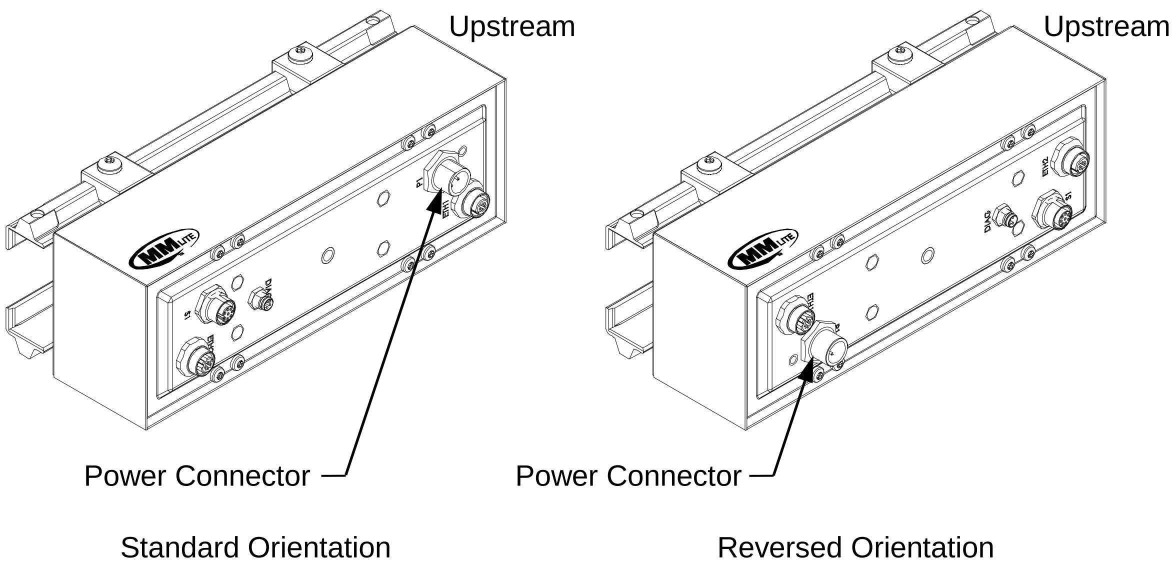

Figure 1: MagneMover LITE Motor Orientation

MagneMover LITE switches do not use this tag since the Node Controller Configuration File defines the orientation of the switches.

Syntax

`<Orientation>`*`Orientation`*`</Orientation>`

Attributes

None

Parent Tag

<Motor>

Contents

Orientation – Used with motors that can self-define their orientation.

Standard – When installed, the power connector for the motor is located toward the upstream end of the path.

Reversed – When installed, the power connector for the motor is located toward the downstream end of the path.

Validation

Node controller requirements for startup and initialization:

- Must not contain any child tags.
- Must contain a valid orientation value.

Example

The following example defines the orientation of the motor as standard, that is, the motor is positioned on the path with the power connector at the upstream end (see Figure 1on page 19 and Figure 3 on page 21).

```
<MICS_motor_data>
<!-- PATH 1 -->
  <Motor> <!-- P1M1 -->
    .
    .
    <Orientation>Standard</Orientation>
```

The following example defines the orientation of the motor as standard, that is, the motor is positioned on the path with the power connector at the downstream end (see Figure 1on page 19 and Figure 3 on page 21).

```
<MICS_motor_data>
<!-- PATH 1 -->
  <Motor> <!-- P1M1 -->
    .
    .
    <Orientation>Reversed</Orientation>
```

MICS File Example

Figure 2is an example of a MagneMover LITE system with Ethernet motors. Figure 3 is the MICS file for the system that is shown in Figure 2.

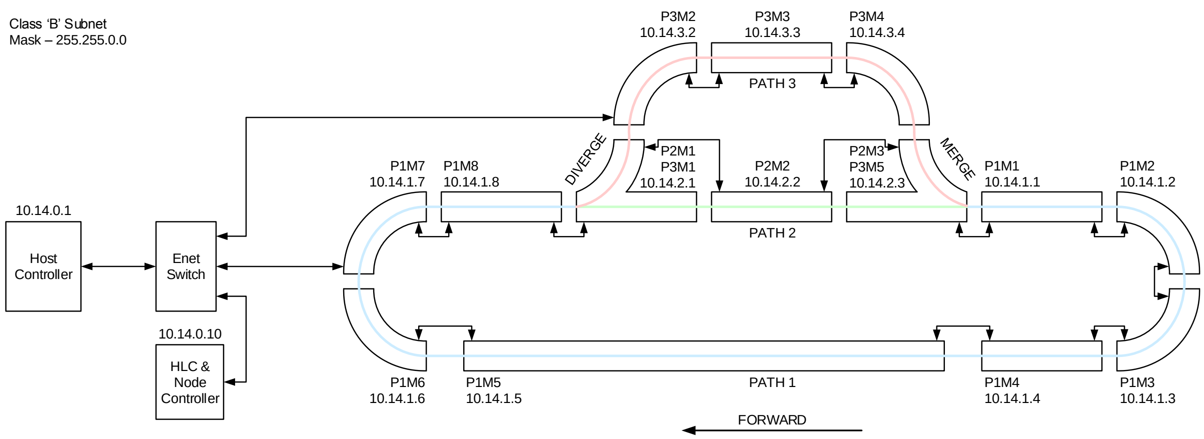

Figure 2: System Layout for MICS File Example

```
<?xml version="1.0" encoding="US-ASCII"?>

<!-- MICS Motor Data file for a MagneMotion Transport System -->
<!-- $Rev: 242 $ $LastChangedDate: 2018-02-15 20:06:34 (Thu, 15 Feb 2018) $ -->

<MICS_motor_data>
<!-- PATH 1 -->
  <Motor> <!-- P1M1 -->
    <Mac_addr>C0:6C:6D:E0:00:20</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.1</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M1</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P1M2 -->
    <Mac_addr>C0:6C:6D:E0:00:21</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.2</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M2</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P1M3 -->
    <Mac_addr>C0:6C:6D:E0:00:28</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.3</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M3</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P1M4 -->
    <Mac_addr>C0:6C:6D:E0:00:23</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.4</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M4</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>
```

Figure 3: MICS File Example

```
  <Motor> <!-- P1M5 -->
    <Mac_addr>C0:6C:6D:E0:00:24</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.5</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M5</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P1M6 -->
    <Mac_addr>C0:6C:6D:E0:00:4E</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.6</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M6</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P1M7 -->
    <Mac_addr>C0:6C:6D:E0:00:4F</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.7</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M7</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P1M8 -->
    <Mac_addr>C0:6C:6D:E0:00:26</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.8</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P1M8</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

<!-- PATH 2 -->
  <Motor> <!-- P2M1 P3M1 -->
    <Mac_addr>C0:6C:6D:E0:00:2B</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.2.1</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P2M1</Track_location> <!-- Track NC Path,Motor # -->
    <Track_location>P3M1</Track_location>
  </Motor>

  <Motor> <!-- P2M2 -->
    <Mac_addr>C0:6C:6D:E0:00:2A</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.2.2</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P2M2</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P2M3 P3M5 -->
    <Mac_addr>C0:6C:6D:E0:00:2C</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.2.3</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P2M3</Track_location> <!-- Track NC Path,Motor # -->
    <Track_location>P3M5</Track_location>
  </Motor>

<!-- PATH 3 -->
  <Motor> <!-- P3M2 -->
    <Mac_addr>C0:6C:6D:E0:00:50</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.3.2</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P3M2</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>
```

Figure 2: MICS File Example (Continued)

```
  <Motor> <!-- P3M3 -->
    <Mac_addr>C0:6C:6D:E0:00:29</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.3.3</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P3M3</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

  <Motor> <!-- P3M4 -->
    <Mac_addr>C0:6C:6D:E0:00:51</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.3.4</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P3M4</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's orientation to downstream -->
  </Motor>

</MICS_motor_data>
```

Figure 2: MICS File Example (Continued)

Wiring Ethernet Motors

The MagneMover LITE Ethernet motors can use different network style connection schemes depending on the application. When using Ethernet, all motors in a specific path must be connected to the same node controller (see Figure 4 through Figure 15). Additionally, multiple paths can be connected to the node controller using the same Ethernet chain.

Power and communications cables can be run in the cable chase under the motors to help protect them from damage and provide easy access for service.

NOTICE
Never connect or disconnect energized power lines to the MagneMover LITE transport system as damage to internal components can result.

NOTICE
Only the NC LITE supports the custom MagneMotion Power over Ethernet (PoE). Never connect the NC LITE to a standard PoE network as damage to internal components can result. The MM LITE motors, NC-12 node controller, and NC-E node controller do not support Power over Ethernet (PoE). Never connect these components to a powered Ethernet network as damage to internal components can result.

Ethernet Motor Communication Recommendations

- Recommended Ethernet addressing scheme (see Figure 2):

  Network.Path.Motor

  - Network addresses are used for network configuration
  - Path 0 addresses are used for Subnet configuration:
    x.y.0.m – *m* = Node controllers/Network devices
  - Path *p* addresses are used for motors on that path:
    x.y.p.m – *p* = path, *m* = motor
- Switches are two logical track paths, only one IP address is assigned.
- Maximum number of motors per Ethernet chain = 50.
- Factory network design must minimize extra traffic on the physical network that the transport system is using.
  - Only use Star or Chain Ethernet connection topologies (see Figure 4 through Figure 14).

  - Closed-loop (ring) Ethernet connections must be avoided (industry standard Ethernet practice) to help prevent network saturation.
  - Only pass transport system communication through the Ethernet chains in the transport system.
  - Large amounts of traffic can degrade the performance of the transport system.
- Standard IP UDP communication, low latency.
- 100BASE-TX Fast Ethernet (IEEE 802.3u) compliant.
- Minimum of CAT 5 cabling is required.
- Ethernet communication topology is independent of transport system configuration (Ethernet chaining does not have to follow the physical path layout).
- The use of Allen-Bradley® Stratix® Managed Ethernet Switches is recommended to deliver the required network performance (see Stratix Managed Switch Configuration on page 49).
- Ethernet chains can consist of multiple paths (as defined in the transport system layout drawing).
- Chains do not need to start at the beginning of a path.
- If all motors in a path are not part of the same Ethernet chain, all chains the path is a member of must connect to the same network as the node controller.

Motor Connection Examples – Straight Paths

The following figures show simplified connection diagrams for the different methods for connecting a simple string of motors using Ethernet. The specific connection method that is used depends on the application for the motors.

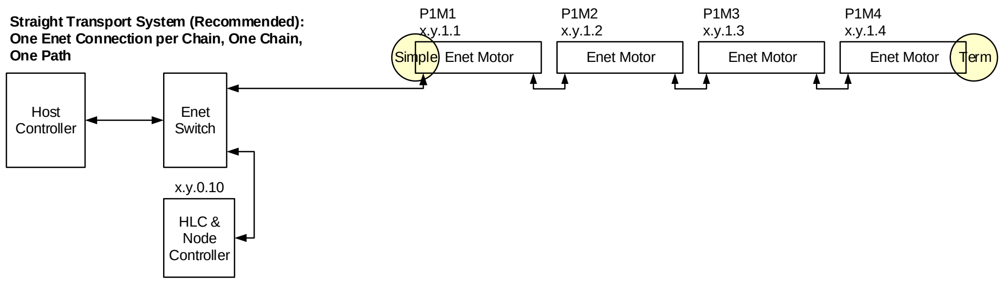

Figure 4: One Path, One Ethernet Chain

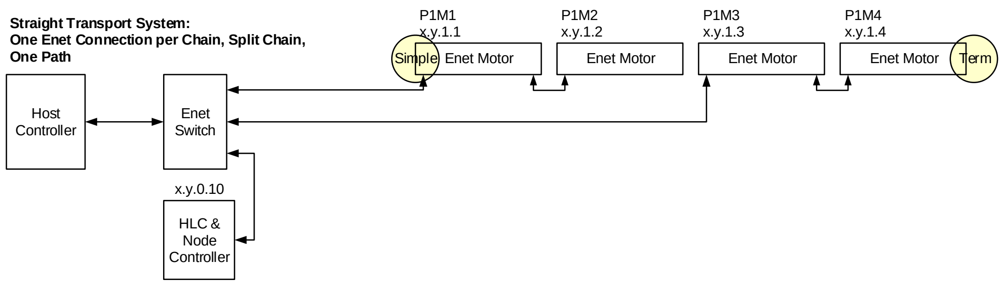

Figure 5: One Path, Two Ethernet Chains

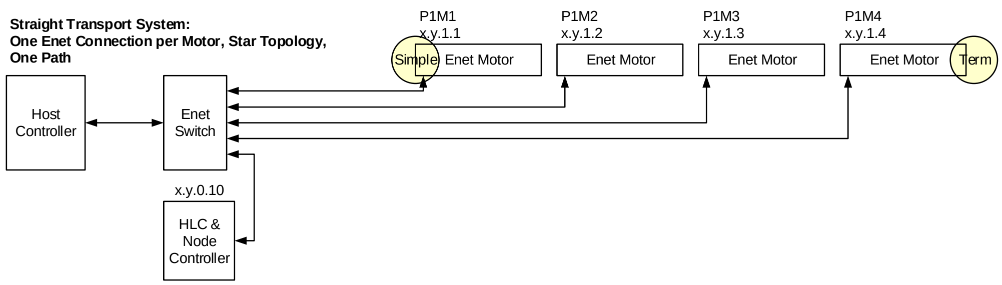

Figure 6: One Path, Ethernet Star

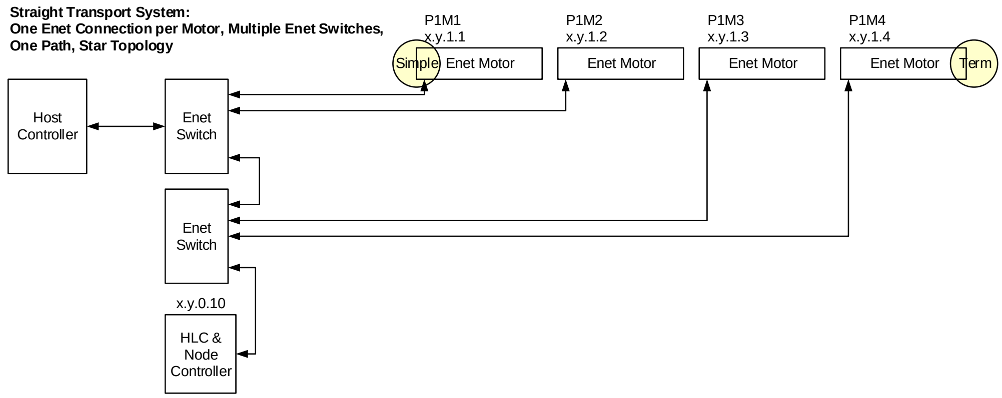

Figure 7: One Path, Ethernet Star with Multiple Ethernet Switches

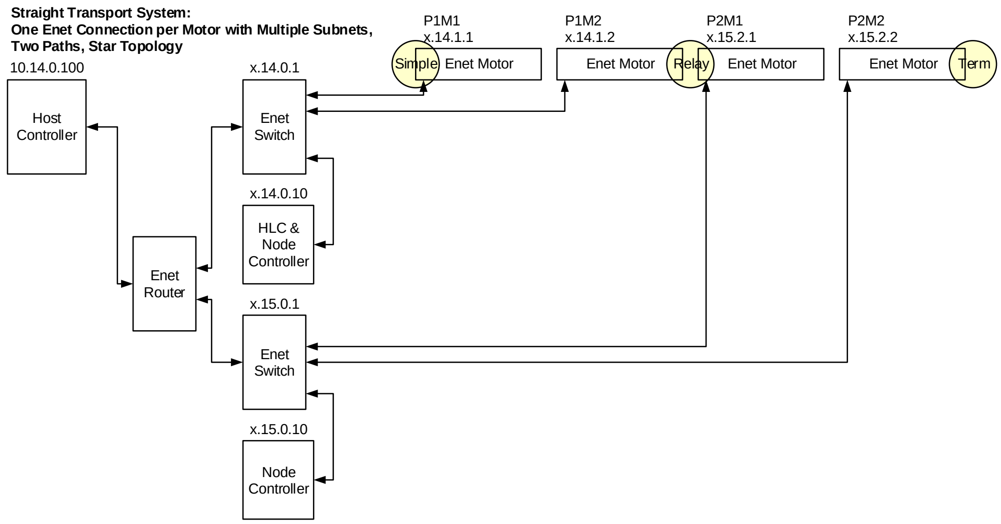

Figure 8: Two Paths, Ethernet Star with Multiple Node Controllers

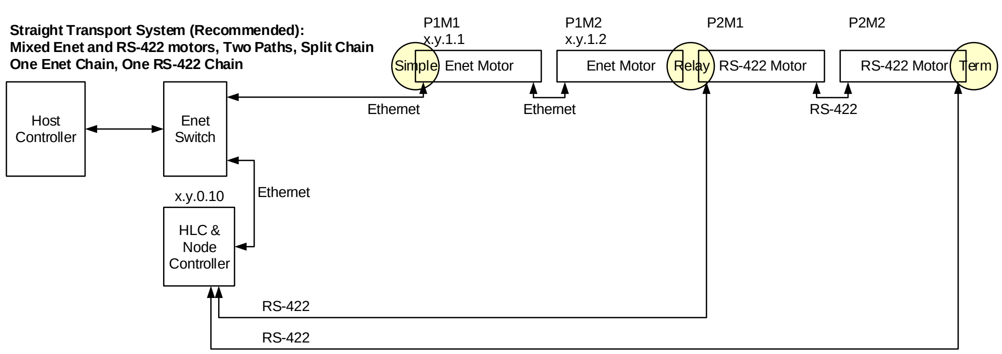

Figure 9: Two Paths, Ethernet Chain and RS-422 Chain

Motor Connection Examples – Loop Paths

The following figures show the different methods for connecting a simple loop of motors using Ethernet.

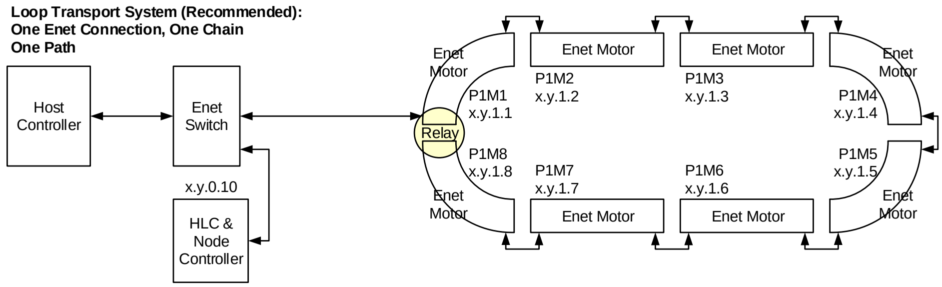

Figure 10: One Path, One Ethernet Chain

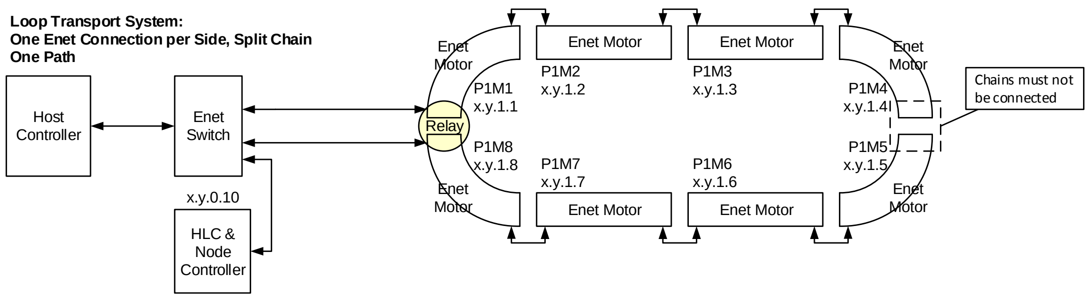

Figure 11: One Path, Two Ethernet Chains

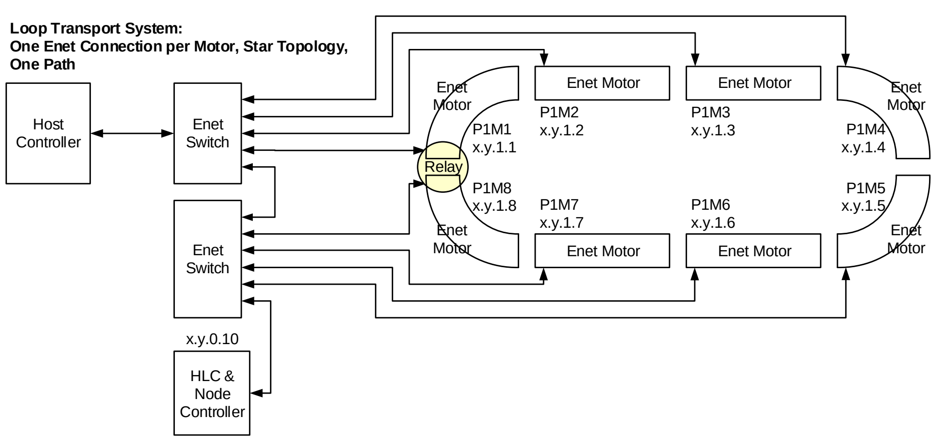

Figure 12: One Path, Ethernet Star

Motor Connection Examples – Multiple Paths

The following figures show the different methods for connecting a multiple-path transport system using Ethernet.

NOTE: The Ethernet switches only have two Ethernet ports, which are used to connect them as one item in a chain. Unlike the RS-422 switches, which have three connections, one for each path where the switch is connected.

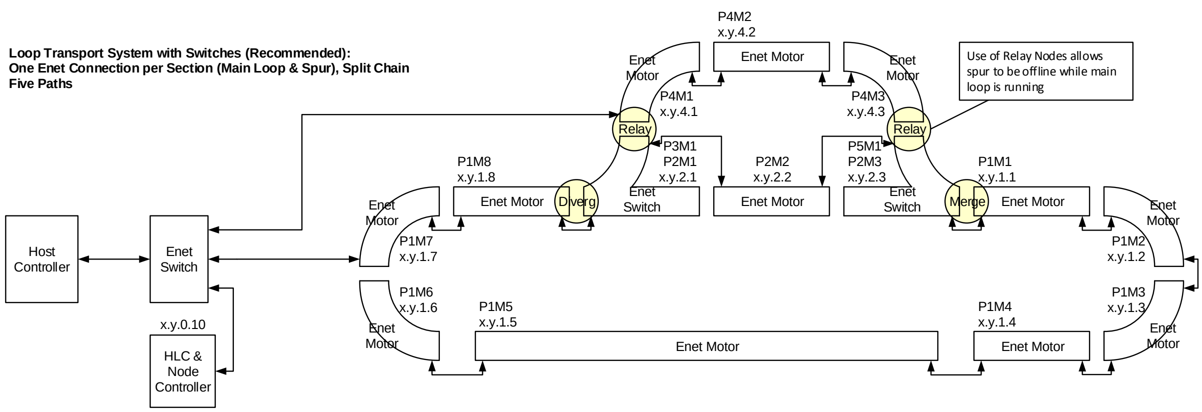

Figure 13: Five Paths, Two Ethernet Chains, Main Loop and Spur

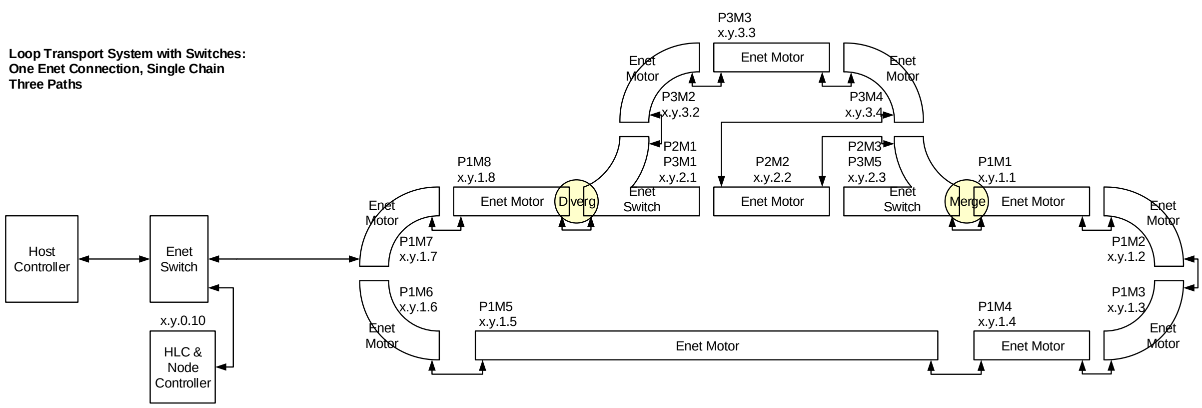

Figure 14: Three Paths, One Ethernet Chain, Main Loop and Spur

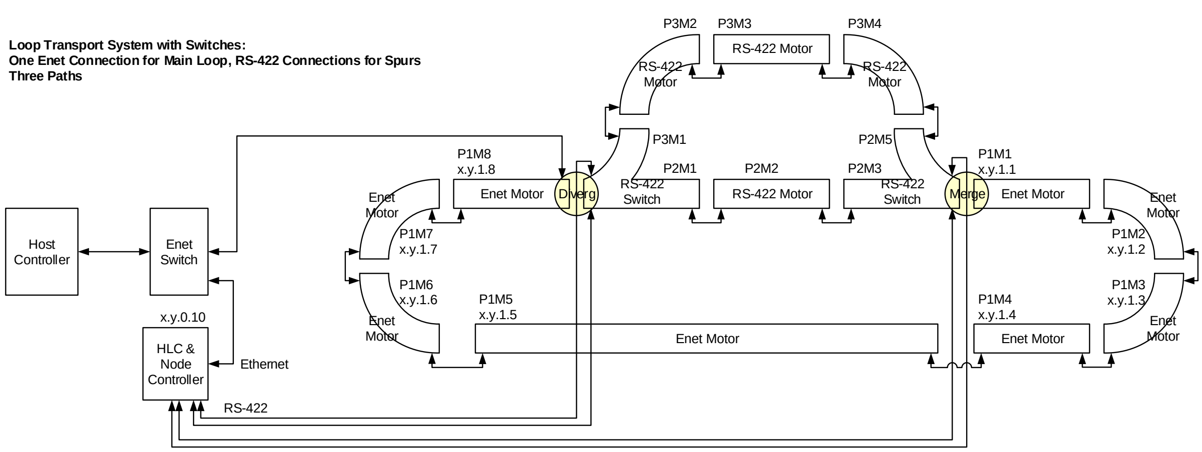

Figure 15: Three Paths, Ethernet Chain Main Loop and RS-422 Spur

Using Both RS-422 and Ethernet Motors

Transport systems can easily combine both RS-422 and Ethernet motors. Motor types are typically combined in a transport system when new Ethernet motors are added to an existing RS-422 system. When adding new motors, it is recommended that they be added as one path to simplify configuration. Relay Nodes are needed at the junction of RS-422 and Ethernet paths as shown in Figure 9 to make the transition between motor types.

Replacing RS-422 Motors with Ethernet Motors

The RS-422 MagneMover LITE motors can be replaced with Ethernet motors easily depending upon the location and mounting method for the motor. This example replaces one curve and one straight motor from the loop that is shown in Figure 16.

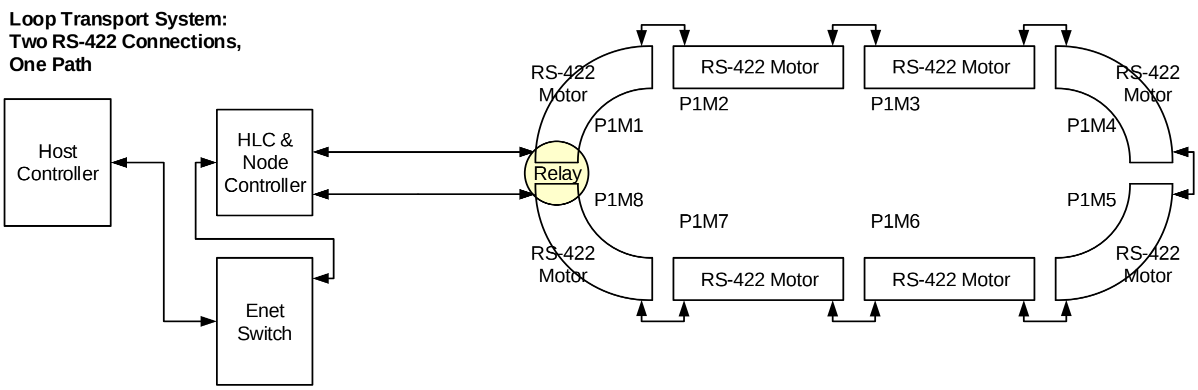

Figure 16: Basic RS-422 Loop

Remove Existing RS-422 Motors

1. Turn off all electrical and pneumatic power to the MagneMover LITE transport system where the motors are being replaced.

2. Remove the RS-422 motors from the transport system as shown in Figure 17 (see the *MagneMover LITE User Manual*).

3. Remove all communication cables for the removed motors.

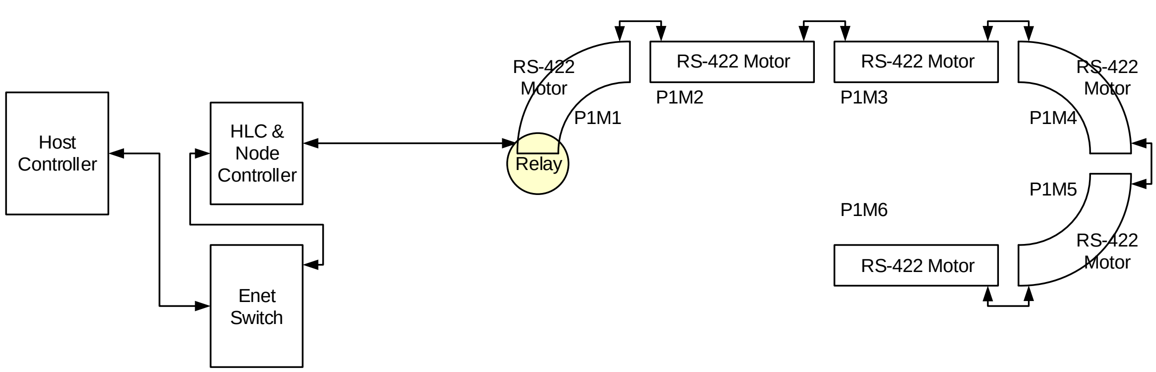

Figure 17: Basic RS-422 Loop with Motors Removed

Install New Ethernet Motors

1. Install the Ethernet motors in the transport system as shown in Figure 18 (see the *MagneMover LITE User Manual*).

2. Install an Ethernet switch if needed.

   A. Reroute the existing connection from the host controller to the node controller so it runs from the host controller to the switch.

   B. Add an Ethernet connection from the node controller to the switch.

3. Install the Ethernet cables.

   A. Add an Ethernet connection from the switch to one of the new Ethernet motors.

   B. Add an Ethernet connection from motor to motor.

4. Reinstall the RS-422 cables.

   A. Replace the connections to the ends of the RS-422 path.

5. Restore power to the MagneMover LITE transport system where the motors were replaced.

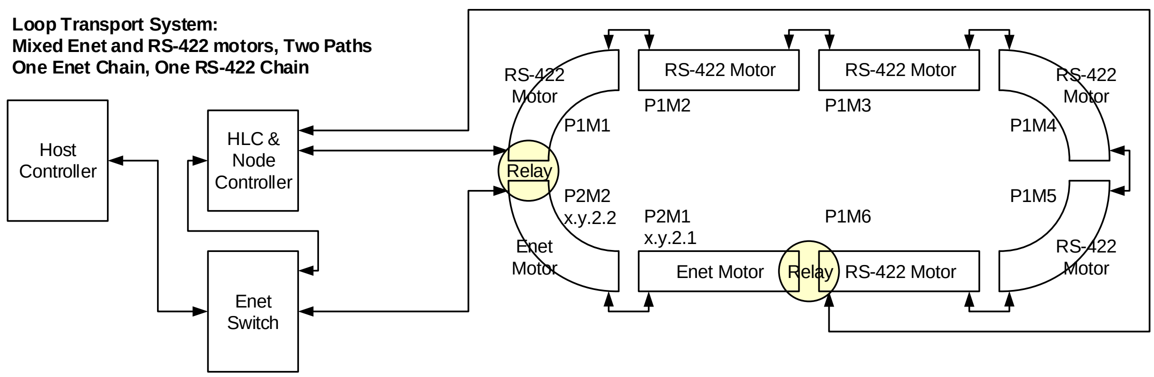

Figure 18: Basic RS-422 Loop With Ethernet Motors Added

Create/Revise Configuration Files

1. Revise the Node Controller Configuration File (see the *MagneMover LITE Configurator User Manual*).

   A. Make sure that the Ethernet motors are properly defined.

   B. Make sure that the loop has two paths that are connected by Relay Nodes as shown in Figure 18.

2. Create a MICS file for the path with the Ethernet motors (see Figure 19 and *MICS File Protocol* on page 11).

   NOTE: The MICS file only contains information about the Ethernet motors.

```
<?xml version="1.0" encoding="US-ASCII"?>

<!-- MICS Motor Data file for a MagneMotion Transport System -->
<!-- $Rev: 242 $ $LastChangedDate: 2018-07-27 13:30:00 (Fri, 27 Feb 2018) $ -->

<MICS_motor_data>
<!-- PATH 2 -->
  <Motor> <!-- P2M1 -->
    <Mac_addr>C0:6C:6D:E0:00:20</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.1</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P2M1</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's downstream orientation -->
  </Motor>

  <Motor> <!-- P2M2 -->
    <Mac_addr>C0:6C:6D:E0:00:21</Mac_addr> <!-- Motor's MAC for MICS -->
    <IP_addr>10.14.1.2</IP_addr> <!-- Motor's IP to use for MICS -->
    <Track_location>P2M2</Track_location> <!-- Track NC Path,Motor # -->
    <Orientation>Standard</Orientation> <!-- Motor's downstream orientation -->
  </Motor>

</MICS_motor_data>

</MICS_motor_data>
```

Figure 19: Example MIX File for Ethernet Motors in Loop

Upload Configuration Files

1. Make sure that the transport system software supports the use of Ethernet motors. If it does not, update all software (controller image, *motor_image*.erf, and so on) to a version that does support the Ethernet motors.
2. Upload the revised Node Controller Configuration File to the node controller and restart the node controller.
3. Upload the MICS file to the node controller and restart the node controller.
4. Upload the Motor Type files for the Ethernet Motors.
5. Program the masters and slaves in the new motors with the current Motor image files (see the *MagneMover LITE User Manual*).
6. Revise the host controller application to account for the new path in the transport system.
7. Resume motion on the MagneMover LITE transport system.

Node Controller Loading

When using RS-422 for motor communication, the load on the node controllers is physically limited by the number of nodes that can be connected to a node controller. This design helps to make sure that the node controller processor is never overloaded.

When using Ethernet for motor communication, there is no physical limit to the number of nodes that are connected to the node controller. However, the total load on the node controllers is limited by the processing power of the node controller as shown in Table 3 and Table 4.

Using Ethernet Communication

NC LITE and NC-12

The processor loading on the NC LITE and NC-12 node controllers is defined as a function of load units per node controller as shown in Table 3 and Table 4. Each multi-port node is equivalent to one load unit and results in the same amount of load on the node controller. Multiport nodes are nodes that connect multiple paths and require multiple connections to the node controller such as Relay, Diverge, or Merge Nodes. Single-port node types (Simple and Terminus Nodes) present a negligible load. When using two Terminus Nodes to pass vehicles between paths, they count as one multi-port node. When node controllers run as both a node controller and the HLC, the HLC process places an additional load of one load unit on the controller.

Table 3: NC LITE and NC-12 Load Unit Availability

NC LITE	5
NC-12	15

Table 4: NC LITE and NC-12 Load Unit Usage

Both HLC and NC	1
Multi-port Node	1

The recommended maximum number of multi-port nodes per NC-12 is 14 if the node controller is also running as an HLC. The recommended maximum number of multi-port nodes per NC LITE is 4 if the node controller is also running as an HLC.

NC-E

The processor loading on the NC-E has been determined based on the various use cases as shown in Table 5. When the node controller is run as the HLC and uses an EtherNet/IP interface to the host, the load on the node controller is greatest since the HLC continuously updates the host with status information. When the node controller is run as the HLC and uses a TCP/IP interface to the host, the load on the node controller is less since the host is only updated on request. When using multiple node controllers, the node controllers without the HLC function have no additional load and can support the maximum number of multi-port nodes.

Multi-port nodes are nodes that connect multiple paths and require multiple connections to the node controller such as Relay, Diverge, or Merge Nodes. These types of nodes result in the same amount of load on the node controller. Single-port node types (Simple and Terminus Nodes) present a negligible load. When using two Terminus Nodes to pass vehicles between paths, they count as one multi-port node.

Table 5: NC-E Loading

Configuration	Max Multi-port Nodes
NC + HLC + EtherNet/IP communication with Host	21
NC + HLC + TCP/IP communication with Host	26
NC Only	34

In systems where there are multiple node controllers, split the load evenly between all node controllers.

Using Mixed RS-422 and Ethernet Communication

When using a transport system that has both RS-422 and Ethernet motors, the loading on the NC LITE or NC-12 node controllers is still defined as a function of load units per node controller. This loading is calculated as described in Node Controller Loading on page 33.

Example 1

The example in Figure 20 shows a small system with two paths, three nodes, and one node controller. One path (shown in orange) uses Ethernet motors and one path (shown in green) uses RS-422 motors. The Ethernet motors connect to the switch the node controller connects to from the upstream end of the path at the Simple Node. The RS-422 motors connect directly to an RS-422 port on the node controller from the upstream end of the path at the Relay Node. The downstream end of the path connects directly to an RS-422 port on the node controller from the Terminus Node.

The total loading on the node controller is determined as follows:

NC and HLC	1 load unit
Relay Node	1 load unit
Total	**2 load units**

The total load on the node controller is two load units. The Simple Node and Terminus Node provide a negligible load and do not need to be counted. An NC LITE can be used as the node controller since the total of two load units is below the maximum of four load units and only two RS-422 connections are required.

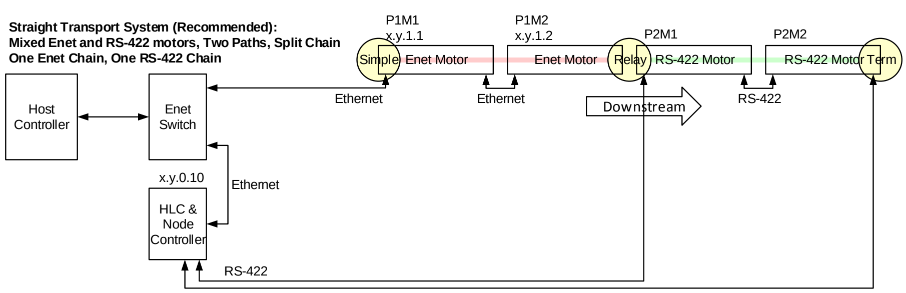

Figure 20: Ethernet Chain and RS-422 Chain (Two Paths)

Example 2

The example in Figure 21 shows a small loop system with three paths, two nodes, and one node controller. The main loop (shown in pink) uses Ethernet motors and connects from the downstream end of the path at the Diverge Node to the switch where the node controller is connected. The spur configuration uses RS-422 motors that connect directly to RS-422 ports on the node controller from the Merge Node and the Diverge Node.

The total loading on the node controller is determined as follows:

NC and HLC	1 load unit
Merge Node	1 load unit
Diverge Node	1 load unit
Total	**3 load units**

The total load on the node controller is three load units. An NC LITE can be used as the node controller since the total load is below the maximum of four load units and only four RS-422 connections are required.

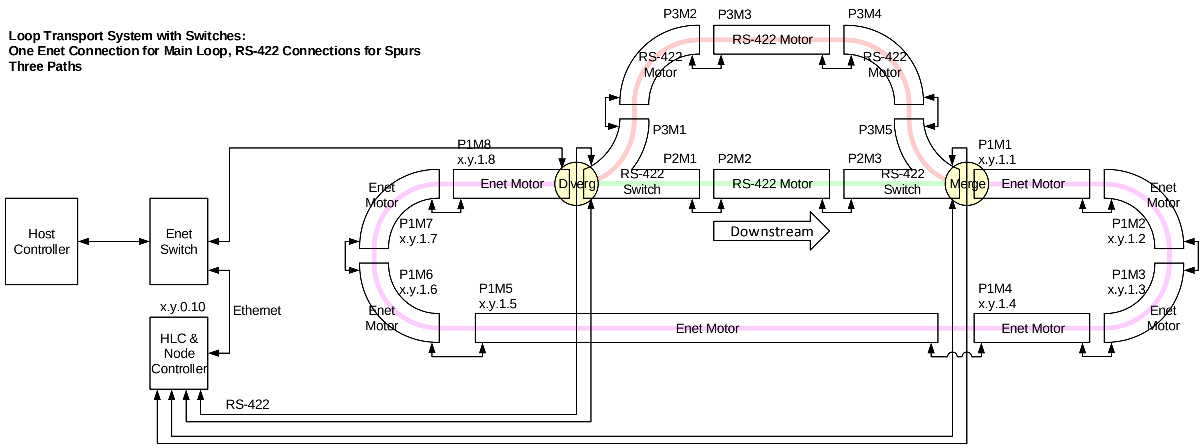

Figure 21: Ethernet Loop Track with RS-422 Spur (Three Paths)

Example 3

The example in Figure 22 shows a small loop system with six paths, four nodes, and one node controller. The main loop (shown in pink) uses Ethernet motors and connects from the downstream end of the path at the Diverge Node on the main loop to the switch where the node controller is connected. The spur configuration uses RS-422 motors that connect directly to RS-422 ports on the node controller from the Merge Node and the Diverge Node on the main loop. The Diverge Node and the Merge Node on the spur both connect directly to the RS-422 ports on the node controller.

The total loading on the node controller is determined as follows:

NC and HLC	1 load unit
2x Merge Node	2 load units
2x Diverge Node	2 load units
Total	**5 load units**

The total load on the node controller is five load units. An NC-12 must be used as the node controller since the total load exceeds the maximum of four load units for an NC LITE and ten RS-422 connections are required.

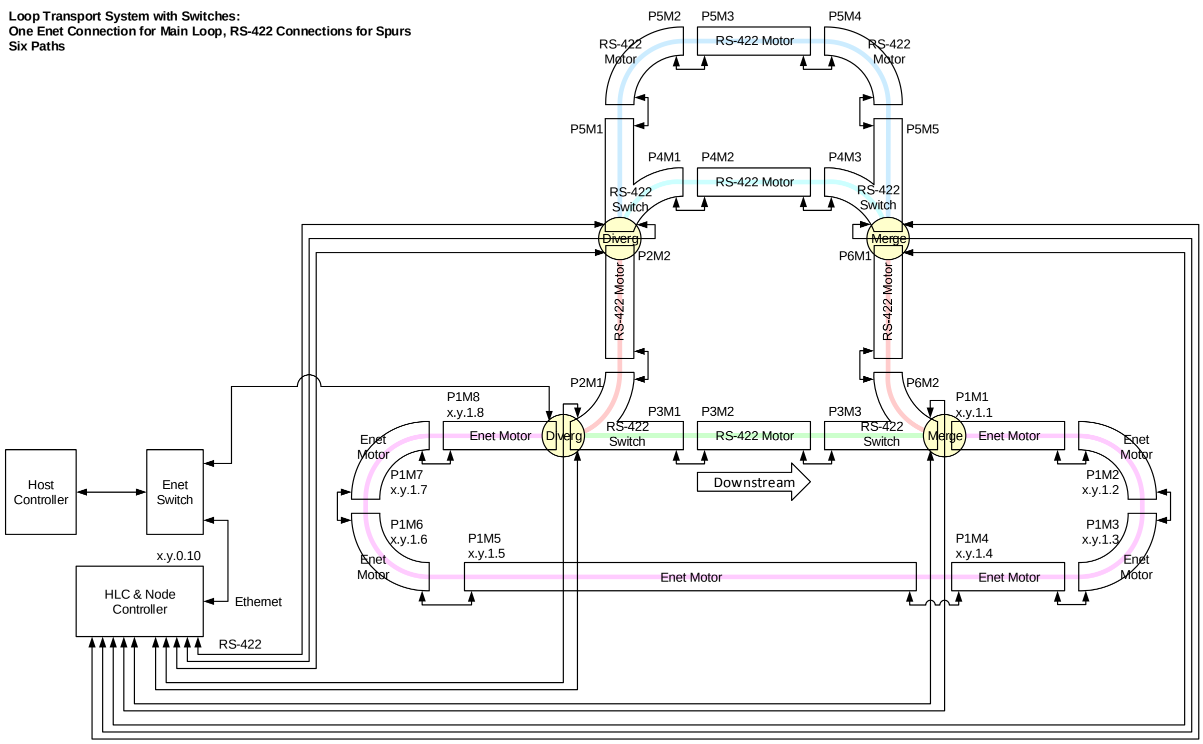

Figure 22: Ethernet Loop Track with RS-422 Spur (Six Paths)

Example 4

The example in Figure 23 shows a small loop system with six paths, four nodes, and one node controller. The main loop (shown in pink) uses RS-422 motors and connects from the upstream end of the path at the Merge Node on the main loop to an RS-422 port on the node controller. The downstream end of the path at the Diverge Node on the main loop connects to an RS-422 port on the node controller. The spur configuration uses Ethernet motors that are wired as one chain and connect to the switch the node controller is connected to.

The total loading on the node controller is determined as follows:

NC and HLC	1 load unit
2x Merge Node	2 load units
2x Diverge Node	2 load units
Total	**5 load units**

The total load on the node controller is five load units. Even though only two RS-422 connections are required, the total load exceeds the maximum of four load units for the NC LITE. Therefore, an NC-12 must be used.

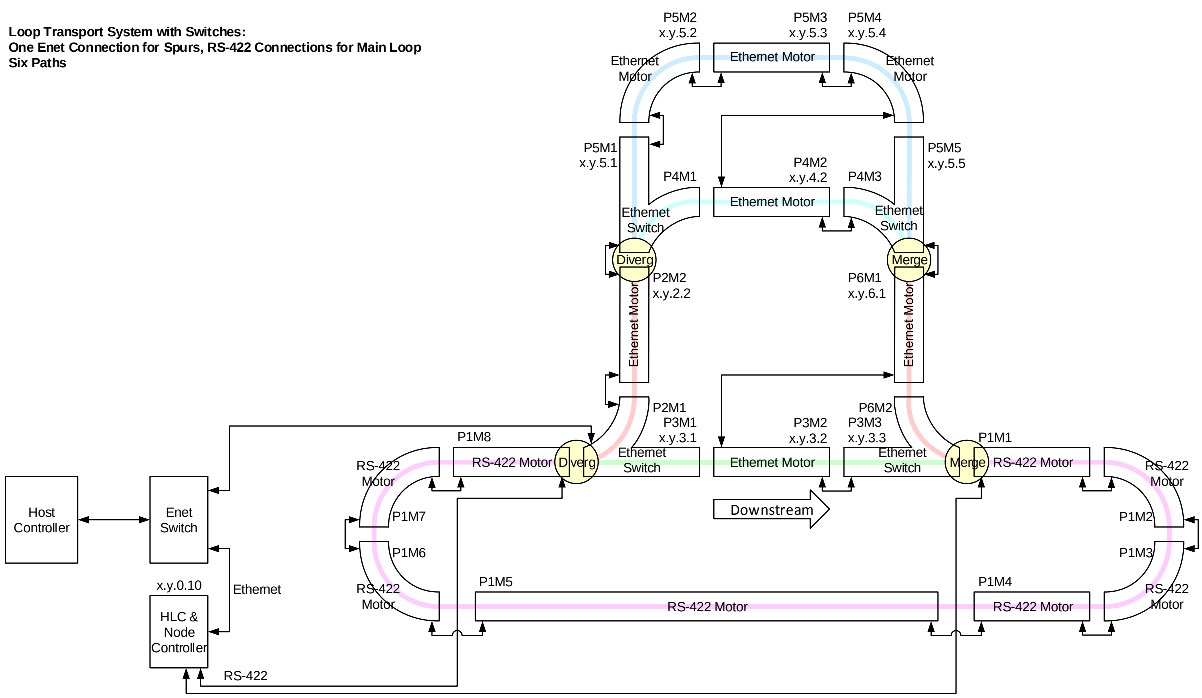

Figure 23: RS-422 Loop Track with Ethernet Spur (Six Paths)

Node Controller Interface Reference

The Node Controller Web Interface (Version 13.x.x), documented in the Node Controller Interface User Manual, 990000377, includes additional functionality to allow use with the MagneMover LITE Ethernet motors. This added functionality includes the following:

- Allows selection and loading of the MagneMotion Information and Configuration Services (MICS) file through the **Configuration Files** page.
- Displays the status of all Ethernet motor connections on the **Node Controller Interface Statistics** page.
- Displays the Ethernet MAC ID of the motors on the **Motor Information** page.

Node Controller Administration – Web

Upload a MICS File

This section describes how to upload a MICS file to all node controllers. The MICS file is created specifically for the topology configuration of the Ethernet motors in the transport system. Once created it must be uploaded to every node controller in the transport system during initial setup, if a motor is replaced, or if the configuration is changed.

When a new MICS file is uploaded to the node controller, the existing file is automatically deleted and replaced with the new file, which is renamed MICS_motor_data.xml.

1. If the transport system has multiple node controllers, connect to the node controller assigned as the high level controller (HLC). Once connected, run the Web Interface for the node controller (see the *Node Controller Interface User Manual*).
2. Select **Configuration Files** on the Main Menu.

   *The **Configuration Files** page, is displayed as shown in Figure 24.*
3. Under **Upload MICS Motor Data File**, select **Browse**.

   *The **Choose File to Upload** dialog box is displayed.*

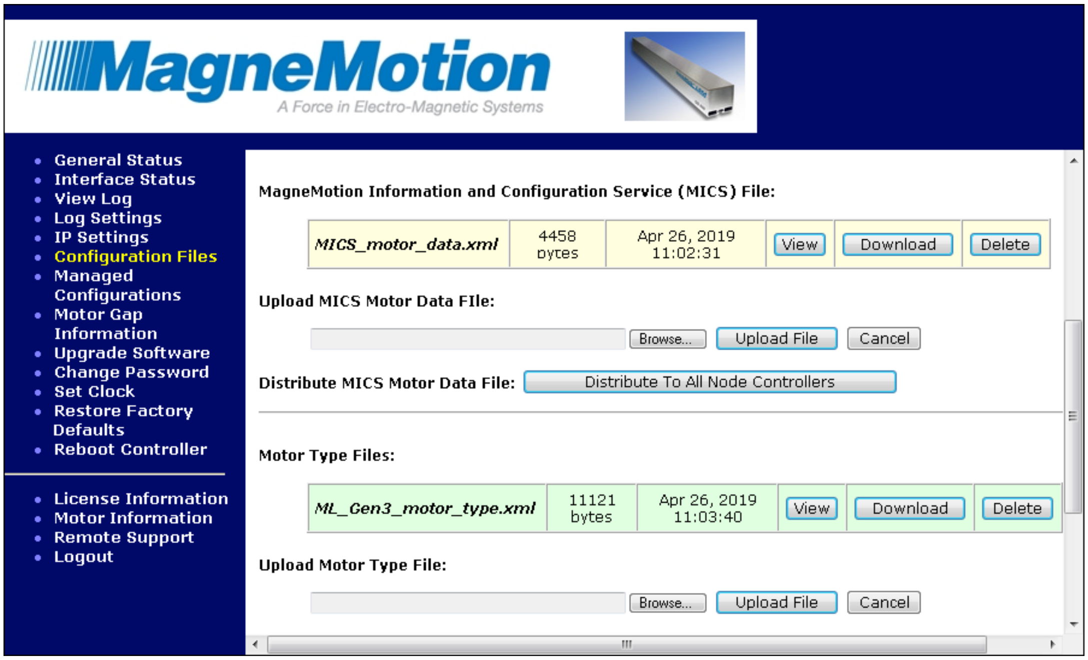

Figure 24: Configuration Files Page, MICS File Upload

4. Use the dialog box controls to select the appropriate MICS file (MICS_motor_data.xml) and select **Open**.

   *The file name is displayed in the **Browse** text field.*

5. Select **Upload File**.

   The file upload status is temporarily displayed while the file is uploaded.

   The MICS file is automatically renamed to MICS_motor_data.xml when it is uploaded.

   *The **Configuration Files** page is redisplayed showing the new MICS file with instructions to restart or reboot the node controller.*

6. For transport systems with one node controller:

   A. Restart the node controller (see the *Node Controller Interface User Manual*).

   For transport systems with multiple node controllers:

   A. Select **Distribute To All Node Controllers**.

      File distribution writes the updated file to the nonvolatile memory in the node controller. Only distribute files during transport system maintenance when there are no active vehicle commands as file distribution can interrupt normal operations.

      Files are only distributed to those node controllers identified in the Node Controller Configuration File.

   B. Restart all node controllers (see the *Node Controller Interface User Manual*).

Delete the Existing MICS File

This section describes how to delete the current MICS file from the node controller.

NOTE: Once the MICS file is deleted, a new MICS file must be uploaded to the node controller to enable communication with the Ethernet motors in the transport system.

1. Select **Configuration Files** on the Main Menu.

   *The **Configuration Files** page, is displayed as shown in Figure 24.*

2. Under MagneMotion Information and Configuration Service (MICS) File, select Delete.

   *The MICS file is deleted and the file deletion status is temporarily displayed. The **Configuration Files** page is redisplayed showing the MICS file has been deleted.*

Provision the Motors with the Network Information

During the initial configuration of the transport system, the Ethernet motors continuously broadcast a request for their network provisioning. Once the MICS file is distributed to the node controllers, they respond to each motor and provide the network provisioning information. The network provisioning information for each motor includes the Subnet Mask and gateway of the Node Controller that provisions that motor.

If changes are made to the MICS file, all power to the motors must be cycled to force the motors to request their network provisioning. Once the motors restart, the node controllers provide the network provisioning information including Subnet Mask to each motor.

If a motor is replaced, once it is powered up it broadcasts a request for its network provisioning. The node controller then provides the network provisioning information to that motor.

Node Controller Web Interface Reference

This section provides an overview of the additions to the User Interface (UI) for the node controller web interface. Examples of those pages that have been updated and descriptions of their new features are provided.

Interface Status

The Interface Status page provides an overview of the communication status for the node controller including Ethernet, Digital I/O, and Serial I/O for the node controller. The Interface Status page, which is shown in Figure 25, is accessed by selecting **Interface Status** on the Main Menu.

Node Controller Interface Statistics

Displays the status of all I/O connections for the node controller. The Node Controller Interface Statistics page, which is shown in Figure 25, is accessed by selecting **Interface Status** on the Main Menu.

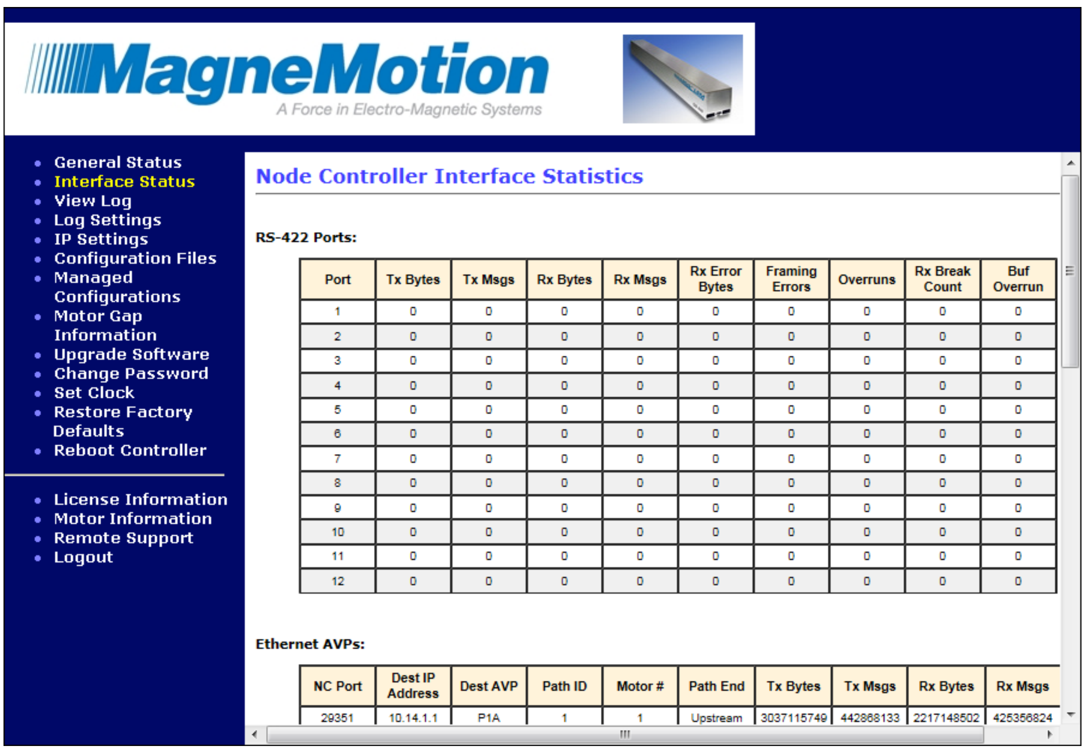

Figure 25: Interface Status Page

- **Ethernet AVPs** – Displays the status of the virtual motor connections on the node controller.
  - **NC Port** – The internal NC Port index for the connection.
  - **Dest IP Address** – The IP address of the motor connection.
  - **Dest AVP** – The logical motor connection name.

- **Path ID** – The path where the motors are located.
- **Motor #** – The motor position in the path.
- **Path End** – The upstream or downstream connection type.
- **Tx Bytes** –Number of bytes transmitted to the device connected to the port since the node controller started running.
- **Tx Msgs** – Number of valid messages that are transmitted to the device that is connected to the port since the node controller started running.
- **Rx Bytes** – Number of bytes received from the device that is connected to the port since the node controller started running.
- **Rx Msgs** – Number of valid messages that are received from the device that is connected to the port since the node controller started running.

Configuration Files

The Configuration Files page is used to identify the configuration files that are currently installed on the node controller and to upload different files if necessary. The Configuration Files page (see Figure 26) is accessed by selecting **Configuration Files** on the Main Menu (see *Upload a MICS File* on page 39 for use details).

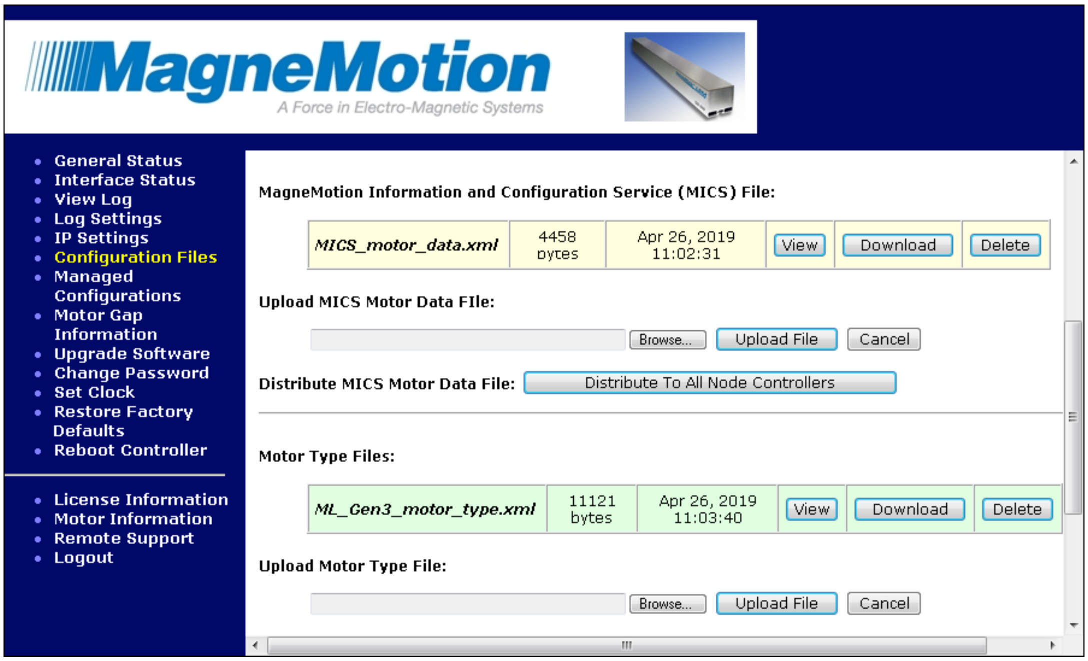

Figure 26: Configuration Files Page, MICS File

- **MagneMotion Information and Configuration Service (MICS) File** – Displays information about the current MICS file (MICS_motor_data.xml).
  - **File Information** – Name, size, date, and time uploaded.
  - **View** – Displays the contents of the MICS file. If the browser does not display the file correctly, download the file to a user-specified location and view it in a text editor.
  - **Download** – Opens a dialog box to download the MICS file from the node controller to a user-specified location.
  - **Delete** – Deletes the MICS file from the node controller.
- **Upload MICS Motor Data File** – Provides a tool to locate and upload to the node controller the MICS file.
  - **Browse...** – Opens a dialog box to locate the MICS file to be uploaded. Once selected the file name is displayed in the text field.
  - **Upload File** – Uploads the selected MICS file to the node controller.
  - **Cancel** – Clears the Browse text field.

- **Distribute MICS Motor Data File** – Provides a tool to distribute the current MICS file to all node controllers. Only shown if the node controller has been configured as the HLC and there are multiple node controllers in the transport system.
  - **Distribute To All Node Controllers** – Copies the current MICS file to all node controllers in the transport system.

Motor Information

The Motor Information page displays manufacturing data for all motors that are connected to the node controller. If the node controller is configured as the HLC, the manufacturing data for all motors in the transport system is displayed. The Motor Information page, which is shown in Figure 27, is accessed by selecting **Motor Information** on the Main Menu. If the motor uses Ethernet for communication, the IP address is displayed.

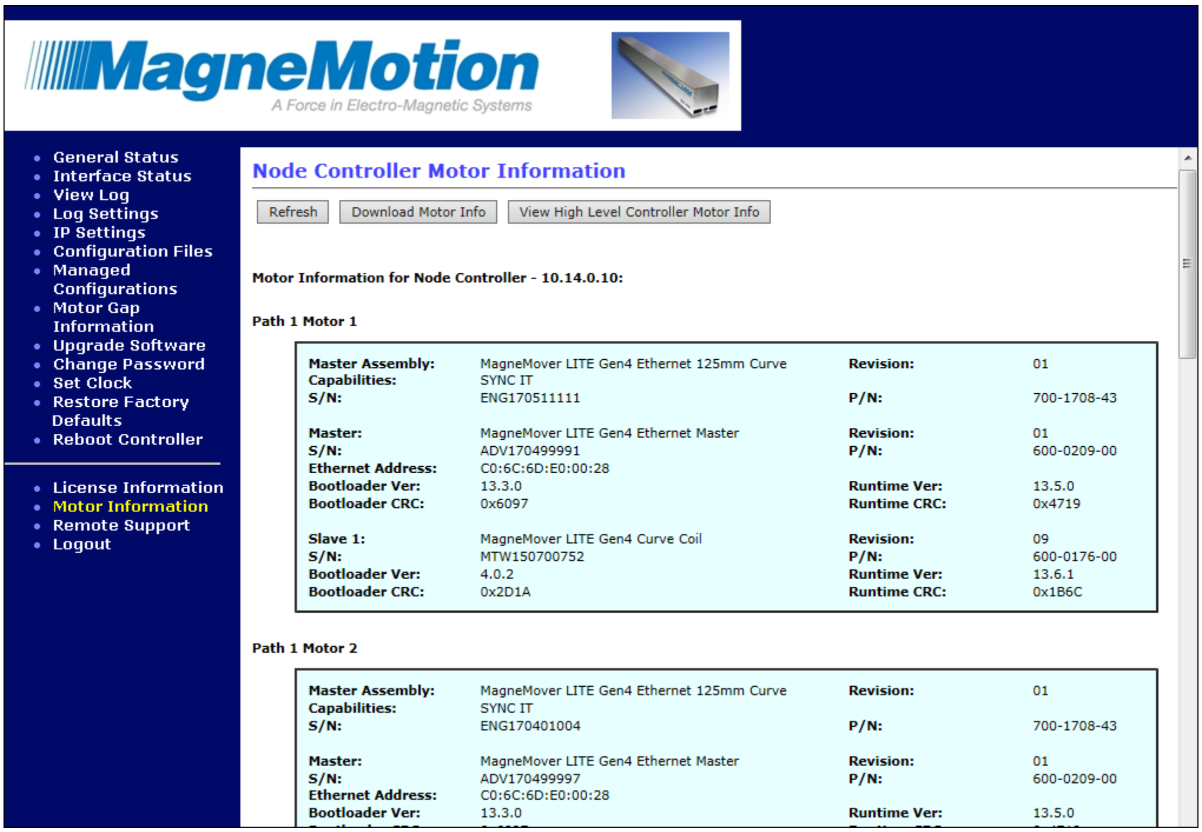

Figure 27: Motor Information Page

- **Ethernet Address** – Displays the Ethernet MAC ID of the master board for the motor.

Log Message Troubleshooting

This section describes additional messages in the log file that are related to the Ethernet motors. These descriptions provide identification of the log messages, their cause, and corrective action. The messages are organized alphabetically by type.

SYSTEM, CRITICAL:

MICS_Provision_Device: MICS PROVISION Path ID: *path* Motor Index: *index* IP: *IPaddress* MAC: *MACaddr* AVP Count: *count*

Explanation: Indicates that the specified motor was provisioned with the identified topology information.

path = Path ID number. This value is the P*n* that is identified in the Track_location XML element.
index = Motor position on the path. This value is the M*n* that is identified in the Track_location XML element.
IPaddress = IPv4 address for the motor.
MACaddr = MAC address in standard IEEE-802 format for the motor.
count = Number of virtual connections provisioned.

Transport System Limits

When using MM LITE Ethernet motors, system limits increase as shown in Table 6. When using both RS-422 and Ethernet motors, Table 6 only applies to the Ethernet motors in the transport system, refer to Transport System Limits in the *Node Controller Interface User Manual* for the limits that are related to RS-422 motors.

Table 6: MM LITE Transport System Limits – Ethernet Motors

	Ethernet Chain†	Path	Node Controller	System
Motors	50	30	3,840	3,840
Node Controllers	–	–	–	96
Nodes/Load Units	–	2	*	128
Stations	–	255	255	2,048
Vehicles	–	300	384	5,120

* Load units are used to describe node controller loading, where the NC-E supports a maximum of 34 nodes, the NC-12 supports 15 load units, and the NC LITE supports 5 load units (see Node Controller Loading on page 33).

† Ethernet chains allow devices to be connected in series with standard Ethernet cable, without the need for additional Ethernet switches. A daisy chain device has two embedded Ethernet ports that function as an Ethernet switch and an interface to the local device. This embedded switch allows information to flow to the device, or flow through the ports to other devices in the chain.

Stratix Managed Switch Configuration

The Allen-Bradley® Stratix® Ethernet switch is recommended for use with the MagneMotion Ethernet motors. For full setup and operating information see the *Stratix Managed Switches User Manual*, 1783-UM007. This section covers specific settings that are required for proper operation with the motors.

Express Setup

Use Express Setup to assign the switch an IP address and run the global macro to set initial configuration parameters.

Configure Network Settings

Once the initial configuration of the switch is complete, it can be accessed through the network by its IP address as shown in Figure 28.

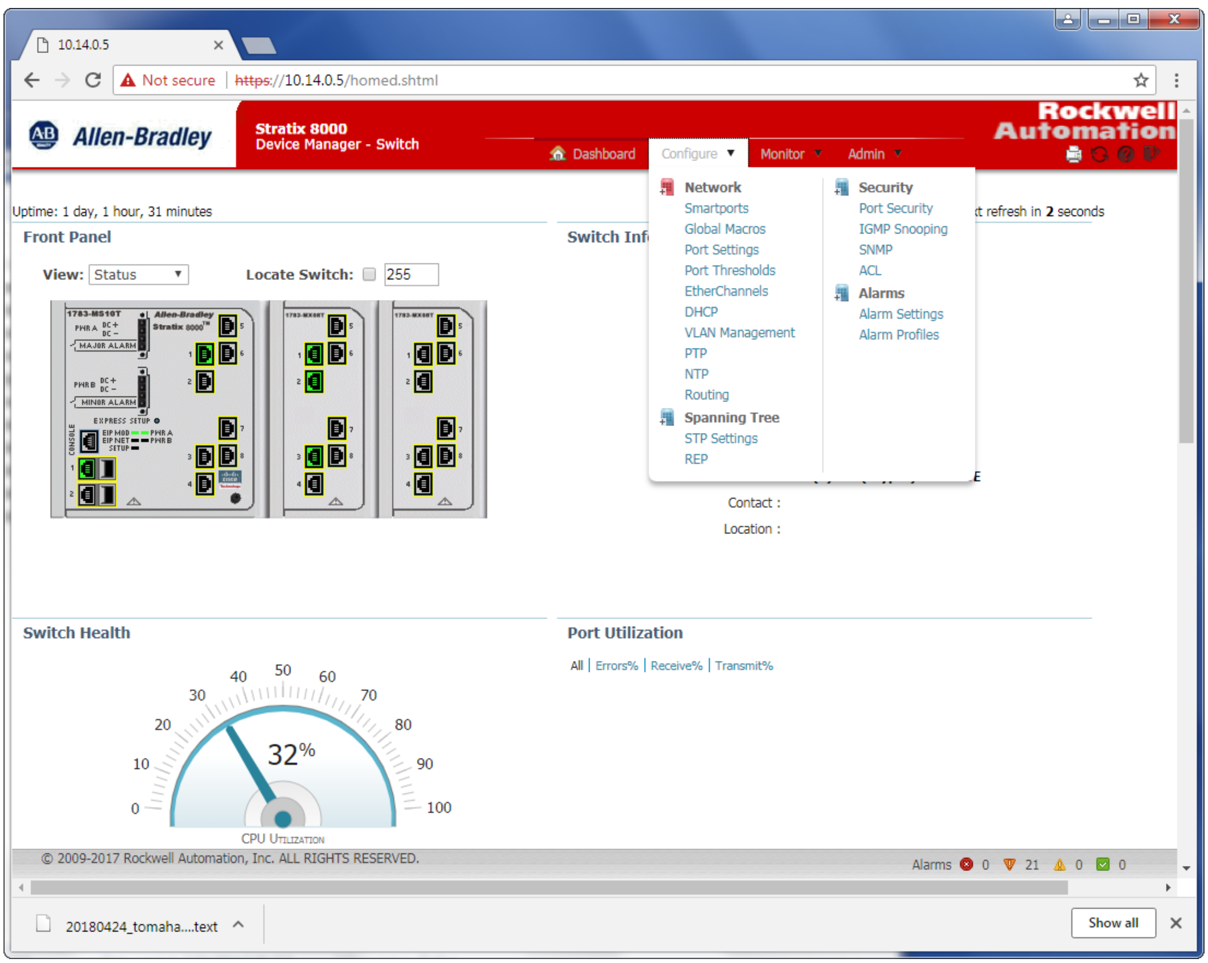

Figure 28: Stratix Switch Main Menu

Network | Smartports

Smartports are recommended configurations for switch ports. These configurations, referred to as port roles, optimize the switch connections and provide security, transmission quality, and reliability for traffic from the switch ports. Port roles also help prevent port misconfigurations.

Select the **Configure** tab on the Device Manager for the switch as shown in Figure 28. From the **Network** menu, select **Smartports** to display the **Smartport Role** configuration as shown in Figure 29. For each port where a motor is connected, select Multiport Automation Device.

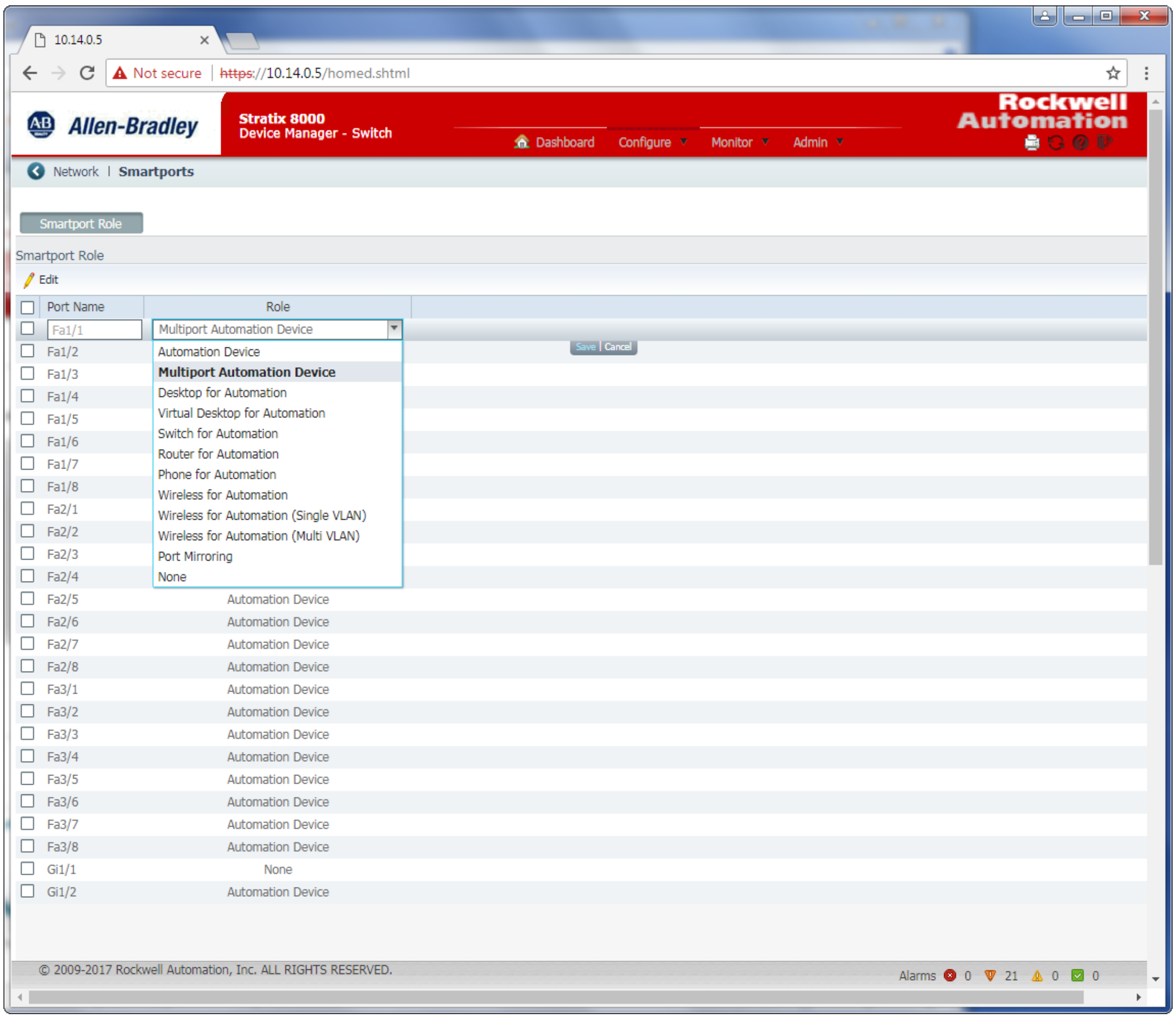

Figure 29: Smartport Setting

Spanning Tree | STP Settings

STP, the IEEE 802.1D bridge protocol, is a Layer 2 link management protocol that provides path redundancy and helps prevent loops in the network.

Select the **Configure** tab on the Device Manager for the switch as shown in Figure 28. From the **Spanning Tree** menu, select **STP Settings** to display the STP configuration as shown in Figure 30. Select the **PortFast** tab and make sure each port where a motor is connected is configured as shown in Figure 30 (Enable Port Fast is selected).

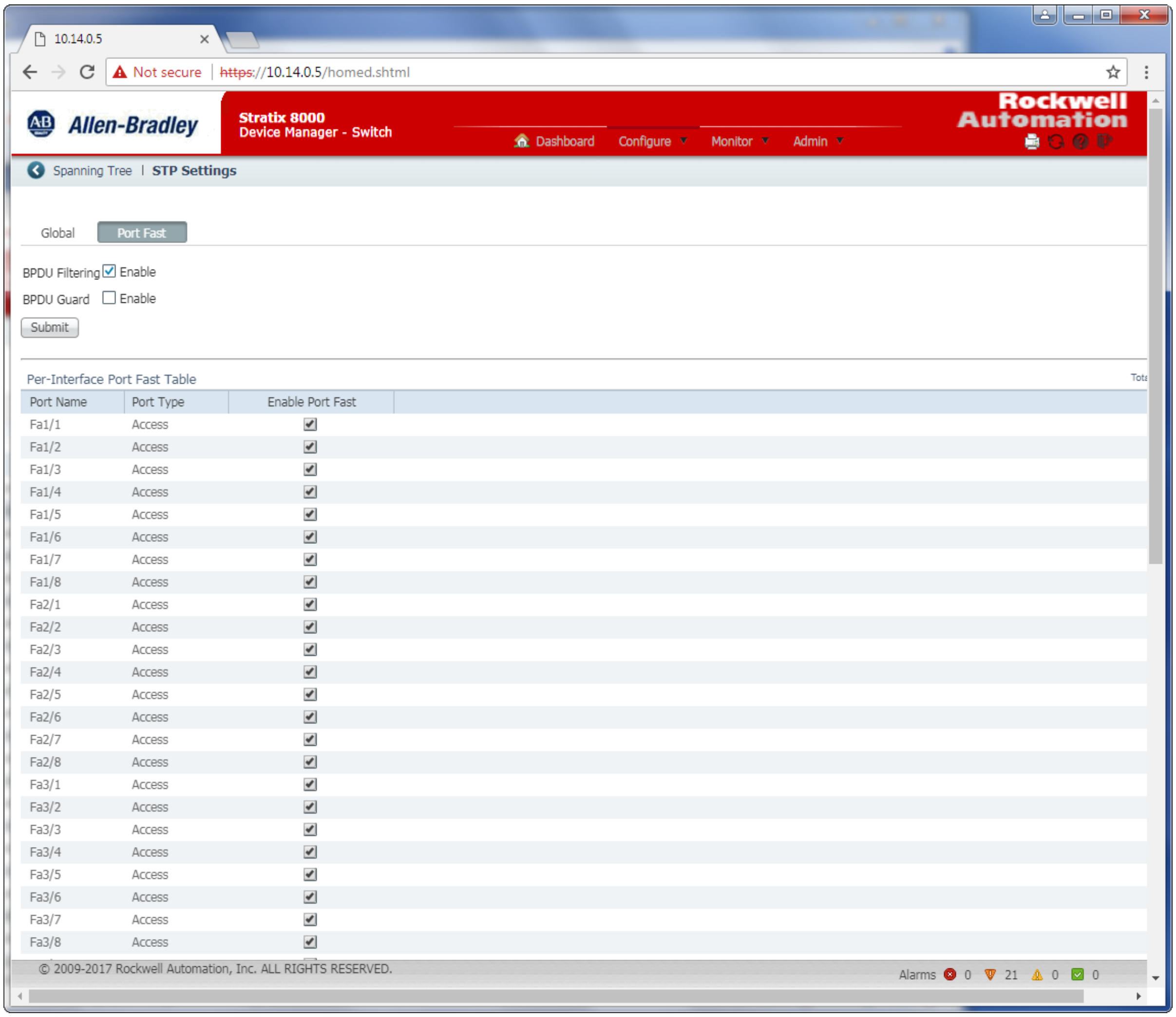

Figure 30: PortFast Setting

More Information

MagneMotion website: www.magnemotion.com

Questions and Comments: www.magnemotion.com/contact

Revision History

Ver.	Change Description
00	Initial release
01	Updated web interface screenshots, added process to delete a MICS file.
02	Updated illustrations to label motor positions on paths. Added section on using both RS-422 and Ethernet motors, including replacing RS-422 motors with Ethernet motors. Added section on node controller loading.
03	Updated node controller loading information, added NC-E node controller, updated Web Interface screenshots, updated transport system limits.
04	Updated node controller loading information, updated transport system limits. Removed description of Ethernet chain from XML Terminology glossary.
05	Corrected version number and minor formatting/grammar corrections. Added Ethernet Chain description to the transport system limits.

Rockwell Automation Support

Use the following resources to access support information.

Technical Support Center	Knowledgebase Articles, How-to Videos, FAQs, Chat, User Forums, and Product Notification Updates.	https://rockwellautomation.custhelp.com/
Local Technical Support Phone Numbers	Locate the phone number for your country.	http://www.rockwellautomation.com/global/support/get-support-now.page
Direct Dial Codes	Find the Direct Dial Code for your product. Use the code to route your call directly to a technical support engineer.	http://www.rockwellautomation.com/global/support/direct-dial.page
Literature Library	Installation Instructions, Manuals, Brochures, and Technical Data.	http://www.rockwellautomation.com/global/literature-library/overview.page
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	http://www.rockwellautomation.com/global/support/pcdc.page

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the How Are We Doing? form at http://literature.rockwellautomation.com/idc/groups/literature/documents/du/ra-du002_-en-e.pdf.

Rockwell Automation maintains current product environmental information on its website at http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page.

Product certificates are located in the Rockwell Automation Literature Library: http://www.rockwellautomation.com/global/literature-library/overview.page

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

